

9.3

*IBM MQ ' nun yapılandırılması*

**IBM**

**Not**

Bu bilgileri ve desteklediđi ürünü kullanmadan önce, "[Özel notlar](#)" sayfa 1055 bölümündeki bilgileri okuyun.

Bu basım, yeni basımlarda tersi belirtilmedikçe, IBM® MQ sürüm 9 yayın düzeyi 3 ve sonraki tüm yayınlar ve deđişiklikler için geçerlidir.

IBM'e bilgi gönderdiğinizde, IBM ' e bu bilgileri size hiçbir sorumluluk yüklemeyen uygun gördüğü yöntemlerle kullanması ya da dağıtması için münhasır olmayan bir hak verirsiniz.

© **Copyright International Business Machines Corporation 2007, 2024.**

# İçindekiler

<b>Yapılandırılıyor.....</b>	<b>7</b>
Çoklu platformlarda kuyruk yöneticileri oluşturma.....	7
Yapılandırılabilir geçici dizin.....	10
Userdata dizini.....	11
Varsayılan kuyruk yöneticisi yaratılması.....	11
Var olan bir kuyruk yöneticisinin varsayılan olarak kullanılması.....	13
Kuyruk yöneticisi yaratıldıktan sonra yapılanış kütüklerinin yedeklenmesi.....	14
İstemci ile sunucu arasındaki bağlantıların yapılandırılması.....	14
Kullanılacak iletişim tipi.....	15
IBM MQ MQI client nasıl ayarlanır.....	17
Genişletilmiş bir işlem istemcisinin yapılandırılması.....	18
MQI kanallarının tanımlanması.....	28
AMQP kanallarının oluşturulması ve kullanılması.....	29
Farklı altyapılarda sunucu bağlantısı ve istemci bağlantısı tanımlamaları yaratılması.....	34
Sunucuda sunucu bağlantısı ve istemci bağlantısı tanımlamaları yaratılması.....	40
MQI kanalları için kanal çıkış programları.....	57
İstemcinin kuyruk paylaşım grubuna bağlanması.....	61
IBM MQ ortam değişkenlerinin kullanılması.....	62
Ortam değişkenleri açıklamaları.....	63
Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerini değiştirme.....	83
IBM MQ yapılandırma dosyası, mqz.ini.....	85
Kuyruk yöneticisi yapılanış dosyaları, qm.ini.....	97
Kuruluş yapılanış dosyası, mqinst.ini.....	154
IBM MQ MQI client yapılanış dosyası, mqclient.ini.....	154
Etkinlik izleme yapılandırma dosyası, mqat.ini.....	186
Dağıtılmış kuyruğa alma yapılandırılıyor.....	189
IBM MQ dağıtılmış kuyruğa alma teknikleri.....	189
Dağıtılmış kuyruk yönetimine giriş.....	209
AIX, Linux, and Windows üzerinde kanalların izlenmesi ve denetlenmesi.....	239
IBM i üzerinde kanalların izlenmesi ve denetlenmesi.....	262
Kuyruk yöneticisi kümesinin yapılandırılması.....	283
Birörnek kümenin yapılandırılması.....	400
Yayınlama/abone olma ileti sisteminin yapılandırılması.....	421
Kuyruğa alınan yayınlama/abone olma iletisi özniteliklerini ayarlama.....	421
Kuyruğa alınan yayınlama/abone olma başlatılıyor.....	423
Kuyruğa alınan yayınlama/abone olma durduruluyor.....	423
Akış eklenmesi.....	424
Akış silinmesi.....	425
Abonelik noktası eklenmesi.....	425
Dağıtılmış yayınlama/abone olma ağlarının yapılandırılması.....	426
Birden çok kuruluşu yapılandırma.....	444
Birden çok kuruluş ortamında uygulamaların bağlanması.....	444
Birincil kuruluşun değiştirilmesi.....	451
Bir kuyruk yöneticisinin kuruluşla ilişkilendirilmesi.....	452
Sistemdeki IBM MQ kuruluşlarını bulma.....	454
Yüksek kullanılabilirliğin, kurtarmanın ve yeniden başlatmanın yapılandırılması.....	455
Otomatik istemci yeniden bağlantısı.....	456
Konsol ileti izlemesi.....	462
Yüksek kullanılabilirlikli yapılandırmalar.....	466
Günlüğe kaydetme: İletilerin kaybolmadığından emin olmak.....	626
IBM MQ kuyruk yöneticisi verilerinin yedeklenmesi ve geri yüklenmesi.....	655
Küme hatası kurtarmada yapılan değişiklikler ( z/OS dışındaki sunucularda).....	662

JMS ve Jakarta Messaging kaynaklarının yapılandırılması.....	663
JNDI ad alanında bağlantı üreticilerini ve hedefleri yapılandırma.....	665
JMS 2.0 nesnelerini IBM MQ Explorer kullanarak yapılandırma.....	668
Yönetim araçlarını kullanarak JMS ve Jakarta Messaging nesnelerini yapılandırma.....	669
WebSphere Application Server içinde JMS 2.0 kaynaklarının yapılandırılması.....	679
WebSphere Application Server ' in en son kaynak bağdaştırıcısı bakım düzeyini kullanacak şekilde yapılandırılması.....	688
JMS <b>PROVIDER</b> VERSION özelliğinin yapılandırılması.....	691
Sürekli aboneliklerin WebSphere Application Server kaldırılması.....	698
yapılandırmaManaged File Transfer.....	700
Çoklu platformlarda MFT yapılandırma seçenekleri.....	701
MFT yapılandırma seçenekleri z/OS.....	702
Redistributable Managed File Transfer components ürününü karşıdan yükleme ve yapılandırma	703
MFT Agent ya da Logger komut veri kümesi yaratılması.....	709
yapılandırmaManaged File Transfer for z/OS.....	710
IBM i üzerinde MFT ' in yapılandırılması.....	741
MFT ' nin ilk kullanım için yapılandırılması.....	742
MFT Agent kuyruk yöneticilerinin yapılandırılması.....	752
MFT günlük kaydedicisi yapılandırılması.....	763
Connect:Direct köprüsünün yapılandırılması.....	786
IBM MQ Console ve REST API ' nin yapılandırılması.....	791
mqweb sunucusu için temel yapılandırma.....	791
Bağımsız IBM MQ Web Server ' in yapılandırılması.....	795
Güvenliğin yapılandırılması.....	797
HTTP anasistem adının yapılandırılması.....	797
HTTP ve HTTPS kapılarının yapılandırılması.....	798
Yanıt zaman aşımının yapılandırılması.....	799
Otomatik başlatma yapılandırılıyor.....	800
Günlüğe kaydetmenin yapılandırılması.....	801
LTPA belirtecinin yapılandırılması.....	805
IBM MQ Console için uzak kuyruk yöneticisi bağlantı davranışının yapılandırılması.....	807
administrative REST API ağ geçidini yapılandırma.....	808
messaging REST API ' nin yapılandırılması.....	810
MFT için REST API ' in yapılandırılması.....	816
mqweb sunucusu JVM 'sinin ayarlanması.....	821
IBM MQ Console ve REST API kuruluş bileşeninin dosya yapısı.....	823
mqweb sunucusu yapılandırmanızı yedekleme ve geri yükleme.....	825
Linux ya da Windows platformlarında Aspera gateway bağlantısı tanımlanması.....	827
IBM MQ ' in IBM Cloud Private ölçüm hizmetiyle kullanılmak üzere yapılandırılması.....	832
IBM Cloud Private üzerinde ölçüm hizmeti eşgörünümüyle kullanılmak üzere bir kuyruk yöneticisinin yapılandırılması.....	833
IBM Cloud Private ölçüm hizmetine HTTP yetkili sunucusu aracılığıyla bağlanma.....	836
Ölçüm hizmeti bağlantısında sorun giderme.....	836
IBM MQ ' in Salesforce push konuları ve platform olaylarıyla kullanılmak üzere yapılandırılması.....	836
IBM MQ Bridge to Salesforce ' nin yapılandırılması.....	838
IBM MQ Bridge to Salesforce için ek yapılandırma seçenekleri.....	843
Salesforce platformu olayları için olay iletileri oluşturma.....	845
IBM MQ Bridge to Salesforce ' in çalıştırılması.....	851
IBM MQ ' nin Blockchain ile kullanılmak üzere yapılandırılması.....	853
IBM MQ Bridge to blockchain için yapılandırma dosyası oluşturma.....	855
Örnek Hyperledger Fabric ağ kimlik bilgileri dosyası.....	856
IBM MQ 9.2.0 ' den IBM MQ Bridge to blockchain için ileti biçimleri.....	858
IBM MQ Bridge to blockchain ' in çalıştırılması.....	860
IBM MQ Bridge to blockchain için ek yapılandırma seçenekleri.....	865
IBM MQ Advanced for z/OS VUE ' nin Blockchain ile kullanılmak üzere yapılandırılması.....	866
z/OS üzerinde IBM MQ Bridge to blockchain için yapılandırma dosyası oluşturma.....	868
z/OS üzerinde IBM MQ Bridge to blockchain için IBM MQ güvenlik yapılandırması.....	870
IBM MQ Bridge to blockchain ' in z/OS üzerinde çalıştırılması.....	871



z/OS üzerinde kuyruk yöneticilerini yapılandırma.....	876
z/OS üzerinde kuyruk yöneticilerini özelleştirme hazırlığı.....	877
ayarlarIBM MQ for z/OS.....	881
z/OS üzerinde bir kuyruk yöneticisinin sınanması.....	946
z/OS üzerinde diğer kuyruk yöneticileriyle iletişim kurulması.....	954
IBM MQ ' yi IMS ile kullanma.....	984
IBM MQ ' yi CICS ile kullanma.....	992
Hizmeti Dil Ortamına ya da z/OS Callable Services 'e yükseltme ve uygulama.....	992
IMS içinde OTMA çıkışlarının kullanılması.....	994
IBM z/OSMF olanağının otomatikleştirmek için kullanılması IBM MQ.....	998
Uzak z/OS kuyruk yöneticilerine MFT aracı bağlantılılığının etkinleştirilmesi.....	1009
yapılandırmaIBM MQ Internet Pass-Thru.....	1010
MQIPT içinde HTTP desteği.....	1010
MQIPT içinde SOCKS desteği.....	1012
MQIPT içinde SSL/TLS desteği.....	1013
Java security manager içinde MQIPT.....	1041
MQIPT içinde güvenlik çıkışları.....	1044
MQIPT içinde kapı numarası denetimi.....	1047
MQIPT içinde saklanan parolaları şifreleme.....	1048
MQIPT ile ilgili diğer güvenlik konuları.....	1050
MQIPT içinde bağlantı günlükleri.....	1051
IBM MQ Internet Pass-Thru ' nin kapsayıcılar kullanılarak yapılandırılması.....	1052
Akış kuyruklarının yapılandırılması.....	1053
<b>Özel notlar.....</b>	<b>1055</b>
Programlama arabirimi bilgileri.....	1056
Ticari Markalar.....	1056



# yapılandırmaIBM MQ

Bir ya da daha çok bilgisayarda bir ya da daha çok kuyruk yöneticisi yaratın ve bunları geliştirme, test ve üretim sistemlerinizde, iş verilerinizi içeren iletileri işleyecek şekilde yapılandırın.

## Bu görev hakkında

IBM MQ' i yapılandırmadan önce, IBM MQ Teknik genel bakışındaki IBM MQ kavramlarına ilişkin bilgileri okuyun. [Planning\(Planlama\)](#) belgesinde IBM MQ ortamınızı nasıl planlayacağınızı okuyun.

IBM MQ' da kuyruk yöneticilerinizi ve ilgili kaynaklarını yaratmak, yapılandırmak ve denetlemek için kullanabileceğiniz birçok farklı yöntem vardır. Bu yöntemler, komut satırı arabirimlerini, bir grafik kullanıcı arabirimini ve bir denetim API 'sini içerir. Bu arabirimler hakkında daha fazla bilgi için bkz. [Yönetme IBM MQ](#).

Bir kuyruk yöneticisinin nasıl yaratılacağına, başlatılacağına, durdurulacağına ve silineceğine ilişkin yönergeler için bkz. [“Çoklu platformlarda kuyruk yöneticileri oluşturma”](#) sayfa 7.

IBM MQ kuruluşlarınızı ve uygulamalarınızı birbirine bağlamak için gereken bileşenleri nasıl oluşturacağınıza ilişkin bilgi için bkz. [“Dağıtılmış kuyruğa alma yapılandırılıyor”](#) sayfa 189.

İstemcilerinizi farklı yöntemler kullanarak bir IBM MQ sunucusuna bağlamaya ilişkin yönergeler için bkz. [“İstemci ile sunucu arasındaki bağlantıların yapılandırılması”](#) sayfa 14.

Bir kuyruk yöneticisi kümesinin nasıl yapılandırılacağına ilişkin yönergeler için bkz. [“Kuyruk yöneticisi kümesinin yapılandırılması”](#) sayfa 283.

Yapılandırma bilgilerinizi değiştirerek IBM MQ ya da bir kuyruk yöneticisinin davranışını değiştirebilirsiniz. Daha fazla bilgi için bkz [“Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerinizi değiştirme”](#) sayfa 83. Genel olarak, bu ürün belgelerinde belirtilmediği sürece, yapılanış değişikliklerinin yürürlüğe girmesi için kuyruk yöneticisini yeniden başlatmanız gerekmez.

**z/OS** IBM MQ for z/OS' ın yapılandırılmasına ilişkin yönergeler için bkz. [“z/OS üzerinde kuyruk yöneticilerini yapılandırma”](#) sayfa 876.

## İlgili kavramlar

[IBM MQ teknik genel bakış](#)

## İlgili görevler

[Yerel IBM MQ nesnelerini yönetme](#)

[Uzak IBM MQ nesnelerini yönetme](#)

**IBM i** [IBMi Yönetimi](#)

**z/OS** [YönetmeIBM MQ for z/OS](#)

[Planlama](#)

**z/OS** [z/OS üzerinde IBM MQ ortamınızın planlanması](#)

[“z/OS üzerinde kuyruk yöneticilerini yapılandırma”](#) sayfa 876

IBM MQ for z/OSüzerinde kuyruk yöneticilerini yapılandırmak için bu yönergeleri kullanın.

Multi

## Çoklu platformlarda kuyruk yöneticileri oluşturma

İletileri ve kuyrukları kullanabilmek için önce en az bir kuyruk yöneticisi ve ilişkili nesnelere yaratmanız ve başlatmanız gerekir. Bir kuyruk yöneticisi, kendisiyle ilişkili kaynakları, özellikle de sahip olduğu kuyrukları yönetir. IBM MQ nesnelerini yaratmak, değiştirmek, görüntülemek ve silmek için İleti Kuyruklaama Arabirimi (MQI) çağrılarını ve komutlarına ilişkin uygulamalara kuyruğa alma hizmetleri sağlar.

## Başlamadan önce

**Önemli:** IBM MQ , boşluk içeren makine adlarını desteklemez. IBM MQ ürününü, alan içeren bir makine adına sahip bir bilgisayara kurarsanız, herhangi bir kuyruk yöneticisi oluşturamazsınız.

Bir kuyruk yöneticisi yaratmadan önce, özellikle bir üretim ortamında dikkate almanız gereken birkaç nokta vardır. Aşağıdaki denetim listesinde çalışmanızı sağlar:

### Kuyruk yöneticisiyle ilişkili kuruluş

Bir kuyruk yöneticisi yaratmak için IBM MQ denetim komutu **crtmqm** kullanılır. **crtmqm** komutu, bir kuyruk yöneticisini **crtmqm** komutunun verildiği kuruluşla otomatik olarak ilişkilendirir. Bir kuyruk yöneticisinde çalışan komutlar için, komutu kuyruk yöneticisiyle ilişkili kuruluştan çalıştırmanız gerekir. **setmqm** komutunu kullanarak bir kuyruk yöneticisinin ilişkili kuruluşunu değiştirebilirsiniz. Windows kuruluş programının kuruluşu gerçekleştiren kullanıcıyı **mqm** grubuna eklediğini unutmayın, daha fazla ayrıntı için bkz. [AIX, Linux® , and Windows üzerinde IBM MQ yönetimi yetkisi.](#)

### Adlandırma kuralları

Tüm platformlardaki kuyruk yöneticileriyle iletişim kurabilmeniz için büyük harfli adlar kullanın. Adların tam olarak girdiğiniz şekilde atandığını unutmayın. Çok fazla yazma zahmetinden kaçınmak için gereksiz yere uzun adlar kullanmayın.

### Benzersiz bir kuyruk yöneticisi adı belirtin

Bir kuyruk yöneticisi yarattığınızda, ağızda başka bir kuyruk yöneticisinin adı aynı olmadığından emin olun. Kuyruk yöneticisi yaratıldığında kuyruk yöneticisi adları denetlenmez ve benzersiz olmayan adlar, dağıtılmış kuyruğa alma için kanal yaratmanızı engeller. Ayrıca, yayınlama/abone olma ileti sistemi için ağı kullanıyorsanız, abonelikler bunları yaratan kuyruk yöneticisi adıyla ilişkilendirilir. Bu nedenle, kümedeki ya da sıradüzendeki kuyruk yöneticileri aynı ada sahipse, bu durum yayınların bunlara ulaşmamasına neden olabilir.

Benzersizliği sağlamanın bir yolu, her kuyruk yöneticisi adının başına kendi benzersiz düğüm adını eklemektir. Örneğin, bir düğümün adı **ACCOUNTS** ise, kuyruk yöneticinize **ACCOUNTS . SATURN . QUEUE . MANAGER** adını verebilirsiniz; burada **SATURN** , belirli bir kuyruk yöneticisini ve **QUEUE . MANAGER** , tüm kuyruk yöneticilerine verebileceğiniz bir uzantıdır. Diğer bir seçenek olarak, bunu atlayabilirsiniz, ancak **ACCOUNTS . SATURN** ve **ACCOUNTS . SATURN . QUEUE . MANAGER** ' in farklı kuyruk yöneticisi adları olduğunu unutmayın.

Diğer kuruluşlarla iletişim için IBM MQ kullanıyorsanız, örnek olarak kendi kurumsal adınızı da ekleyebilirsiniz. Bu, örneklerde gösterilmez, çünkü takip edilmesi daha zor hale getirir.

**Not:** Denetim komutlarındaki kuyruk yöneticisi adları büyük ve küçük harfe duyarlıdır. Bu, **jupiter.queue.manager** ve **JUPITER.queue.manager** adlarıyla iki kuyruk yöneticisi oluşturmanıza izin verildiği anlamına gelir. Ancak, bu tür komplikasyonlardan kaçınmak daha iyidir.

### Kuyruk yöneticisi sayısını sınırla

Kaynakların izin vereceği sayıda kuyruk yöneticisi yaratabilirsiniz. Ancak, her kuyruk yöneticisi kendi kaynaklarını gerektirdiğinden, bir düğümde 100 kuyruğu olan bir kuyruk yöneticisine sahip olmak, her biri on kuyruk olan on kuyruk yöneticisine sahip olmaktan daha iyidir.

Üretim sistemlerinde, birçok işlemci tek bir kuyruk yöneticisiyle kullanılabilir, ancak daha büyük sunucu makineleri birden çok kuyruk yöneticisiyle daha etkili çalışabilir.

### Varsayılan bir kuyruk yöneticisi belirtin

Her düğümün varsayılan bir kuyruk yöneticisi olmalıdır; ancak, IBM MQ düğümü olmadan bir düğümde yapılandırılabilir. Varsayılan kuyruk yöneticisi, uygulamaların **MQCONN** çağrısında bir kuyruk yöneticisi adı belirtmedikleri takdirde bağlandıkları kuyruk yöneticisidir. Kuyruk yöneticisi adı belirtmeden **runmqsc** komutunu çağırdığınızda **MQSC** komutlarını işleyen de kuyruk yöneticisidir.

Bir kuyruk yöneticisinin varsayılan olarak belirtilmesi, düğüm için var olan varsayılan kuyruk yöneticisi belirtimlerinin yerine geçer.

Varsayılan kuyruk yönetimini değiştirmek diğer kullanıcıları ya da uygulamaları etkileyebilir. Değişiklik, bağlı olan uygulamalar üzerinde herhangi bir etki yaratmaz; bu uygulamalar, başka MQI çağrılarında özgün bağlanma çağrılarındaki tanıtıcıyı kullanabilirler. Bu tanıtıcı, çağrıların aynı kuyruk yöneticisine yönlendirilmesini sağlar. Varsayılan kuyruk yöneticisini değiştirdiğiniz *bundan sonra* ile bağlantı kuran

uygulamalar yeni varsayılan kuyruk yöneticisine bağlanır. Niyetiniz bu olabilir, ancak varsayılanı değiştirmeden önce bunu göz önünde bulundurmanız gerekir.

Varsayılan kuyruk yöneticisi yaratılması "[Varsayılan kuyruk yöneticisi yaratılması](#)" sayfa 11 başlıklı konuda açıklanmaktadır.

### **Bir gönderilmeyen ileti kuyruğu belirtin**

Teslim edilmeyen ileti kuyruğu, iletilerin amaçladıkları hedefe yönlendirilememesi durumunda konulacağı yerel bir kuyruktur.

Ağınızdaki her kuyruk yöneticisinde bir ileti kuyruğunun olması önemlidir. Bir hata tanımlamazsanız, uygulama programlarındaki hatalar kanalların kapanmasına neden olabilir ve yönetim komutlarına yanıt alınmayabilir.

Örneğin, bir uygulama başka bir kuyruk yöneticisindeki bir kuyruğa ileti yerleştirmeyi denerse, ancak yanlış kuyruk adı verirse, kanal durdurulur ve ileti iletim kuyruğunda kalır. Diğer uygulamalar iletileri için bu kanalı kullanamazlar.

Kuyruk yöneticilerinin gitmeyen ileti kuyrukları varsa kanallar etkilenmez. Teslim edilmeyen ileti, teslim alma ucundaki gitmeyen iletiler kuyruğuna yerleştirilir ve kanal ve iletim kuyruğu kullanılabilir olur.

Bir kuyruk yöneticisi oluşturduğunuzda, gönderilmeyen iletiler kuyruğunun adını belirtmek için **-u** işaretini kullanın. MQSC komutunu kullanarak, kullanılacak gelmeyen iletiler kuyruğunu belirtmek için önceden tanımladığınız bir kuyruk yöneticisinin özniteliklerini değiştirebilirsiniz. ALTER MQSC komutunun bir örneği için [Kuyruk yöneticisi özniteliklerinin görüntülenmesi ve değiştirilmesi](#) başlıklı konuya bakın.

### **Varsayılan bir iletim kuyruğu belirleyin**

İletim kuyruğu, iletiden önce uzak kuyruk yöneticisine aktarılan iletilerin kuyruğa alındığı yerel bir kuyruktur. Varsayılan iletim kuyruğu, iletim kuyruğu belirtik olarak tanımlanmadığında kullanılan kuyruktur. Her kuyruk yöneticisine varsayılan bir iletim kuyruğu atanabilir.

Bir kuyruk yöneticisi yarattığınızda, varsayılan iletim kuyruğunun adını belirlemek için **-d** işaretini kullanın. Bu, kuyruğu gerçekten yaratmaz; bunu daha sonra belirtik olarak yapmanız gerekir. Ek bilgi için [Yerel kuyruklarla çalışma](#) başlıklı konuya bakın.

### **Gerek duyduğunuz günlüğe kaydetme değiştirmelerini belirtin**

Günlüğe kaydetme tipi ve günlük dosyalarının yolu ve büyüklüğü de içinde olmak üzere, `crtmqm` komutunda günlüğe kaydetme değiştirmelerini belirtebilirsiniz.

Bir geliştirme ortamında, varsayılan günlüğe kaydetme parametreleri yeterli olmalıdır. Ancak, aşağıdaki gibi varsayılanları değiştirebilirsiniz:

- Büyük günlükleri destekleyemeyen düşük uçlu bir sistem yapılandırmanız var.
- Aynı anda kuyruklarınızda çok sayıda uzun ileti olmasını bekliyorsunuz.
- Kuyruk yöneticisinden geçen çok sayıda kalıcı ileti bekliyorsunuz.

Günlüğe kaydetme değiştirmelerini ayarladıktan sonra, bunların bazıları yalnızca kuyruk yöneticisi silinip aynı adla ancak farklı günlüğe kaydetme değiştirmeleriyle yeniden yaratılarak değiştirilebilir.

Günlüğe kaydetme parametreleriyle ilgili daha fazla bilgi için bkz. "[Yüksek kullanılabilirliğin, kurtarmanın ve yeniden başlatmanın yapılandırılması](#)" sayfa 455.

## **AIX**

### **Yalnızca IBM MQ for UNIX sistemleri için**

`crtmqm` komutunu kullanmadan önce, ayrı bir yerel dosya sisteminde bile `/var/mqm/qmgrs/qmgr` kuyruk yöneticisi dizinini yaratabilirsiniz. `crtmqm` komutunu kullandığınızda, `/var/mqm/qmgrs/qmgr` dizini varsa, boşsa ve `mqm` dizinine aitse, bu dizin kuyruk yöneticisi verileri için kullanılır. Dizinin sahibi `mqm` değilse, yaratma işlemi First Failure Support Technology ( FFST ) ile başarısız olur. İletidir. Dizin boş değilse, yeni bir dizin yaratılır.

## Bu görev hakkında

Bir kuyruk yöneticisi yaratmak için IBM MQ denetim komutu **crtmqm** kullanılır. Daha fazla bilgi için bkz. **crtmqm**. **crtmqm** komutu, gerekli varsayılan nesnelere ve sistem nesnelere otomatik olarak yaratır (bkz. [Sistem varsayılan nesnelere](#)). Varsayılan nesnelere, oluşturduğunuz nesne tanımlamalarının temelini oluşturur; kuyruk yöneticisi işlemi için sistem nesnelere gereklidir.

**Windows** Windows sistemlerinde, **crtmqm** komutunun **sax** seçeneğini kullanarak kuyruk yöneticisinin birden çok örneğini başlatma seçeneğiniz vardır.

Bir kuyruk yöneticisi ve nesnelere yarattığınızda, kuyruk yöneticisini başlatmak için **strmqm** komutunu kullanabilirsiniz.

## Yordam

- Kuyruk yöneticilerini yaratmanıza ve yönetmenize yardımcı olacak bilgiler için aşağıdaki alt konulara bakın:
  - [“Varsayılan kuyruk yöneticisi yaratılması” sayfa 11](#)
  - [“Var olan bir kuyruk yöneticisinin varsayılan olarak kullanılması” sayfa 13](#)
  - [“Kuyruk yöneticisi yaratıldıktan sonra yapılan iş kütüklerinin yedeklenmesi” sayfa 14](#)

## İlgili kavramlar

[Kuyruk yöneticileriyle çalışma](#)

## İlgili görevler

[QM1 adlı bir kuyruk yöneticisi yaratılması](#)

[“Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerini değiştirme” sayfa 83](#)

Yapılandırma (.ini) dosyalarındaki bilgileri düzenleyerek, IBM MQ ya da tek bir kuyruk yöneticisinin davranışını kuruluşunuzun gereksinimlerine uyacak şekilde değiştirebilirsiniz. IBM MQ MQI clients için yapılandırma seçeneklerini de değiştirebilirsiniz.

[“z/OS üzerinde kuyruk yöneticilerini yapılandırma” sayfa 876](#)

IBM MQ for z/OS üzerinde kuyruk yöneticilerini yapılandırmak için bu yönergeleri kullanın.

## İlgili başvurular

[Sistem ve varsayılan nesnelere](#)

[crtmqm](#)

Linux

AIX

## Yapılandırılabilir geçici dizin

Yapılandırılabilir kısa ömürlü dizin, kuyruk yöneticisine veri kısa ömürlü olması gereken yeri tanımlar. Bu, AIX and Linux etki alanı yuvalarının Red Hat® OpenShift® ortamındaki sisteme bağlanmamış bir dosya sistemine yerleştirilmesini sağlamak için kullanılabilir.

IBM MQ 9.2.0' den önce, AIX and Linux altyapılarında bir kuyruk yöneticisi çalışırken, /var/mqm/sockets dizini altında AIX and Linux etki alanı yuvaları yaratılır. Kuyruk yöneticisini bir taşıyıcı içinde /var/mqm ile sisteme bağlanmış bir dosya sistemi olarak çalıştırırken, bazı Linux altyapıları bu etki alanı yuvalarının yaratılmasını engelleyebilir; bunun nedeni, taşıyıcının dışındaki bazı işlemlerin kap içindeki işlemlere engel olmasına izin vermesidir. Bu sorun, IBM MQ ' in varsayılan güvenlik bağlamı altında bir Red Hat OpenShift kapsayıcı platformunda çalışmasını önler.

IBM MQ 9.2.0' den **EphemeralPrefix** özniteliği, geçici dizinin konumunu yapılandırmak için kullanılabilir. Bu özniteliği kullanmazsanız, davranışta herhangi bir değişiklik görmezsiniz.

mqc.ini içinde bir kuyruk yöneticisi girdisi oluşturulduğunda (**crtmqm** ya da **addmqinf** komutları kullanılarak), aşağıdaki durumlarda **EphemeralPrefix** özniteliği eklenir:

- [“AllQueue mqc.ini dosyasının yöneticiler kısmı” sayfa 89](#) içinde **DefaultEphemeralPrefix** özniteliğini ayarlayın.
- **MQ\_EPHEMERAL\_PREFIX** ortam değişkenini ayarlayın.

- Yalnızca **addmqinf** komutu için **-v** EphemeralPrefix belirtin.

**EphemeralPrefix** özniteliğini, durdurulduğunda var olan bir kuyruk yöneticisine belirttik olarak ekleyebilirsiniz; bu, kuyruk yöneticisi yeniden başlatıldığında eklenir.



Bir **EphemeralPrefix** özniteliği belirtirseniz, kuyruk yöneticisi başlatıldığında, kuyruk yöneticisine ilişkin veri geçici değerinin olağan yeri yerine bu önek altında yaratılmasına neden olur. Yani:

- Genellikle `/var/mqm/sockets/<QM>` altında bulunan yuva dosyaları şimdi `/<EphemeralPrefix>/sockets/<QM>` altında olacaktır
- Genellikle `/<Prefix>/qmgrs/<QM>/@<Subpool>` altında bulunan alt havuz dosyaları şimdi `/<EphemeralPrefix>/qmgrs/<QM>/@<Subpool>` altında olacaktır

#### Notlar:

- `/var/mqm/sockets/@SYSTEM` sabit konumunda kalır ve **EphemeralPrefix** özniteliğinin bir parçası değildir.
- `AMQCLCHL.TAB`, `/<Prefix>/qmgrs/<QM>/@ipcc` altında kalır ve **EphemeralPrefix** özniteliğinin bir parçası değildir.

**EphemeralPrefix** özniteliğinin içerebildiği karakter sayısı altyapınıza bağlıdır:

-  Linux ve AIX AIX and Linux platformlarında 12 karakterle sınırlıdır.
-  IBM i IBM i üzerinde 24 karakterle sınırlıdır.

Çok uzun olan ya da var olmayan bir **EphemeralPrefix** özniteliği belirtirseniz, bir `AMQ7001E` iletisi alırsınız:

`AMQ7001E: Kuyruk yöneticisi için belirtilen konum geçerli değil.`

## Multi Userdata dizini

Kalıcı uygulama durumunu saklamak için `userdata` dizinini kullanabilirsiniz.

Her IBM MQ kuyruk yöneticisinin, hem kuyruk verilerini hem de kurtarma günlüğünü içeren, kalıcı durumu için özel olarak ayrılmış bir dosya sistemi vardır. Dosya sistemi, uygulamalarınıza ilişkin kalıcı durum bilgilerini saklamak için kullanabileceğiniz bir `userdata` dizini içerir. [Directory content on Unix and Linux Systems](#) ve [Directory content on Windows systems](#) başlıklı konuya bakın.

`userdata` dizini çeşitli durumlarda yararlı olabilir; örneğin:

- RDQM yapılandırmalarında, bir kuyruk yöneticisi başka bir düğüme geçtiğinde uygulama bilgilerinin de hareket etmesi için (bkz. [“Kalıcı uygulama durumu saklanıyor”](#) sayfa 568).
- Çok eşgörünümlü kuyruk yöneticileri için, uygulama durumları paylaşılan ağ dosya sisteminde kuyruk yöneticisi verileriyle birlikte bulunur.
- Daha genel olarak, uygulamaların yapılandırıldığı kuyruk yöneticisi hizmetleri.

Uygulama durumunu `userdata` dizininde saklamayı seçerseniz, bu yere yazılan verilerin kuyruk yöneticisine ayrılan kullanılabilir disk alanını kullanabileceğini bilmeniz gerekir. Kuyruk yöneticisinin kuyruk verilerini, günlüklerini ve diğer kalıcı durum bilgilerini yazması için yeterli disk yeri kaldığından emin olmanız gerekir.

`userdata` dizini `mqm` kullanıcı ve grup iyeliğine sahiptir ve kullanıcıların IBM MQ yönetici grubunda (`mqm`) olmalarına gerek kalmadan bu dizine erişebilmeleri için herkes tarafından okunabilir. `userdata` dizininin izinlerini değiştiremezsiniz, ancak gerek duyduğunuz sahiplik ve izinlerle içerik oluşturabilirsiniz.

## Multi Varsayılan kuyruk yöneticisi yaratılması

Varsayılan kuyruk yöneticisi, uygulamaların `MQCONN` çağrısında bir kuyruk yöneticisi adı belirtmedikleri takdirde bağlandıkları kuyruk yöneticisidir. Kuyruk yöneticisi adı belirtmeden **runmqsc** komutunu

çağırdığınızda MQSC komutlarını işleyen de kuyruk yöneticisidir. Bir kuyruk yöneticisi yaratmak için IBM MQ denetim komutu **crtmqm** kullanılır.

## Başlamadan önce

Varsayılan bir kuyruk yöneticisi yaratmadan önce, [“Çoklu platformlarda kuyruk yöneticileri oluşturma”](#) sayfa 7 başlıklı konuda açıklanan konuları okuyun.

**Linux** **AIX** AIX and Linux üzerinde bir kuyruk yöneticisi yaratmak için **crtmqm** kullandığınızda, `/var/mqm/qmgrs/qmgr` dizini zaten varsa, sahibi `mqm` 'dir ve boşsa, kuyruk yöneticisi verileri için kullanılır. Dizinin sahibi `mqm` değilse, kuyruk yöneticisinin yaratılması First Failure Support Technology (FFST) iletilisiyle başarısız olur. Dizin boş değilse, kuyruk yöneticisi verileri için yeni bir dizin yaratılır.

Bu önem, `/var/mqm/qmgrs/qmgr` dizini ayrı bir yerel dosya sisteminde önceden varsa da geçerlidir.

## Bu görev hakkında

**crtmqm** komutunu kullanarak bir kuyruk yöneticisi yarattığınızda, komut otomatik olarak gerekli varsayılan nesnelere ve sistem nesnelere yaratır. Varsayılan nesnelere, oluşturacağınız nesne tanımlamalarının temelini oluşturur ve kuyruk yöneticisi işlemi için sistem nesnelere gereklidir.

Komuta ilgili parametreleri ekleyerek, örneğin, kuyruk yöneticisi tarafından kullanılacak varsayılan iletim kuyruğunun adını ve teslim edilmeyen ileti kuyruğunun adını da tanımlayabilirsiniz.

**Windows** Windows sistemlerinde, kuyruk yöneticisinin birden çok eşgörünümünü başlatmak için **crtmqm** komutunun **sax** seçeneğini kullanabilirsiniz.

**crtmqm** komutu ve sözdizimi hakkında daha fazla bilgi için bkz. [crtmqm](#).

## Yordam

- Varsayılan bir kuyruk yöneticisi oluşturmak için **-q** işaretiyle **crtmqm** komutunu kullanın. Aşağıdaki **crtmqm** komutu örneği, `SATURN.QUEUE.MANAGER` adlı varsayılan bir kuyruk yöneticisi yaratır:

```
crtmqm -q -d MY.DEFAULT.XMIT.QUEUE -u SYSTEM.DEAD.LETTER.QUEUE SATURN.QUEUE.MANAGER
```

Burada:

**-q**

Bu kuyruk yöneticisinin varsayılan kuyruk yöneticisi olduğunu gösterir.

**-d MY.DEFAULT.XMIT.QUEUE**

Bu kuyruk yöneticisi tarafından kullanılacak varsayılan iletim kuyruğunun adıdır.

**Not:** IBM MQ sizin için varsayılan bir iletim kuyruğu yaratmaz; kendiniz tanımlamanız gerekir.

**-U SYSTEM.DEAD.LETTER.QUEUE**

Kuruluştaki IBM MQ tarafından oluşturulan varsayılan teslim edilmeyen iletiler kuyruğunun adıdır.

**SATURN.QUEUE.MANAGER**

Bu kuyruk yöneticisinin adıdır. Bu, **crtmqm** komutunda belirtilen son parametre olmalıdır.

## Sonraki adım

Bir kuyruk yöneticisi ve nesnelere yarattığınızda, [Kuyruk yöneticisini başlatma](#) için **strmqm** komutunu kullanın.

## İlgili kavramlar

[Yerel kuyruklarla çalışma](#)



## İlgili görevler

“Kuyruk yöneticisi yaratıldıktan sonra yapılanış kütüklerinin yedeklenmesi” sayfa 14  
IBM MQ yapılandırma bilgileri, AIX, Linux, and Windows üzerindeki yapılandırma dosyalarında depolanır. Bir kuyruk yöneticisi yarattıktan sonra, yapılanış dosyalarınızı yedekleyin. Daha sonra, sorunlara neden olan başka bir kuyruk yöneticisi oluşturursanız, sorunun kaynağını kaldırdığınızda yedekleri yeniden yürürlüğe alabilirsiniz.

[Kuyruk yöneticisi özneliklerini görüntüleme ve değiştirme](#)

## İlgili başvurular

[Sistem ve varsayılan nesnelere](#)

## Multi Var olan bir kuyruk yöneticisinin varsayılan olarak kullanılması

Var olan bir kuyruk yöneticisini, bir metin düzenleyicisini kullanarak el ile ya da Windows ve Linux sistemlerinde IBM MQ Explorer komutunu kullanarak varsayılan kuyruk yöneticisi yapabilirsiniz.

## Bu görev hakkında

Var olan bir kuyruk yöneticisini varsayılan kuyruk yöneticisi yapmak üzere bir metin düzenleyicisi kullanmak için aşağıdaki adımları izleyin.

**Linux** **Windows** Windows ve Linux (x86 ve x86-64 platformları) sistemlerinde, bu değişikliği yapmak için IBM MQ Explorer kullanmayı tercih ediyorsanız, bkz. “Bir kuyruk yöneticisini varsayılan yapmak için IBM MQ Explorer kullanılması” sayfa 13.

Varsayılan bir kuyruk yöneticisi yarattığınızda, adı IBM MQ yapılandırma dosyasındaki (mqc.ini) DefaultQueueManager öğesinin Name özneliğine eklenir. Yoksa, dörtlük ve içeriği otomatik olarak yaratılır.

## Yordam

- Var olan bir kuyruk yöneticisini varsayılan yapmak için, Name özneliğinde kuyruk yöneticisi adını yeni varsayılan kuyruk yöneticisinin adıyla değiştirin. Bunu bir metin düzenleyicisini kullanarak el ile yapabilirsiniz.
- Düşümdede varsayılan bir kuyruk yöneticiniz yoksa ve var olan bir kuyruk yöneticisini varsayılan yapmak istiyorsanız, *DefaultQueueManager* kısmı gerekli adla yaratın.
- Başka bir kuyruk yöneticisini yanlışlıkla varsayılan yaparsanız ve özgün varsayılan kuyruk yöneticisine geri dönmek isterseniz, istenmeyen varsayılan kuyruk yöneticisini istediğiniz varsayılan kuyruk yöneticisiyle değiştirerek, mqc.ini içindeki DefaultQueueManager kısmı düzenleyin.

## İlgili görevler

“Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerini değiştirme” sayfa 83  
Yapılandırma (.ini) dosyalarındaki bilgileri düzenleyerek, IBM MQ ya da tek bir kuyruk yöneticisinin davranışını kuruluşunuzun gereksinimlerine uyacak şekilde değiştirebilirsiniz. IBM MQ MQI clients için yapılandırma seçeneklerini de değiştirebilirsiniz.

## Linux Windows Bir kuyruk yöneticisini varsayılan yapmak için IBM MQ Explorer kullanılması

Windows ve Linux (x86 ve x86-64 platformları) sistemlerinde, var olan bir kuyruk yöneticisini varsayılan kuyruk yöneticisi yapmak için IBM MQ Explorer 'i kullanabilirsiniz.

## Bu görev hakkında

Var olan bir kuyruk yöneticisini Windows ve Linux (x86 ve x86-64 platformları) sistemlerinde varsayılan kuyruk yöneticisi yapmak üzere IBM MQ Explorer 'i kullanmak için aşağıdaki adımları tamamlayın.

Bu deęişiklięi el ile yapmak için bir metin düzenleyicisi kullanmayı tercih ediyorsanız, bkz. [“Var olan bir kuyruk yöneticisinin varsayılan olarak kullanılması” sayfa 13.](#)

## Yordam

1. IBM MQ Explorer uygulamasını açın.
2. **IBM MQ** öğesini farenin sağ düğmesiyle tıklatın ve **Özellikler ...** öğesini seçin. **Properties for IBM MQ** (IBM MQ Özellikleri) panosu görüntülenir.
3. **Varsayılan kuyruk yöneticisi adı** alanında, varsayılan kuyruk yöneticisinin adını yazın.
4. **Tamam**'ı tıklatın.

## ALW Kuyruk yöneticisi yaratıldıktan sonra yapılanış kütüklerinin yedeklenmesi

IBM MQ yapılandırma bilgileri, AIX, Linux, and Windows üzerindeki yapılandırma dosyalarında depolanır. Bir kuyruk yöneticisi yarattıktan sonra, yapılanış dosyalarınızı yedekleyin. Daha sonra, sorunlara neden olan başka bir kuyruk yöneticisi oluşturursanız, sorunun kaynağını kaldırdığınızda yedekleri yeniden yürürlüğe alabilirsiniz.

### Bu görev hakkında

Genel bir kural olarak, her yeni kuyruk yöneticisini yarattığınızda yapılanış dosyalarınızı yedekleyin.

İki tip yapılanış kütüğü vardır:

- Ürünü kurduğunuzda, IBM MQ yapılandırma dosyası (mqc.ini) oluşturulur. Bir kuyruk yöneticisini her yarattığınızda ya da sildiğinizde güncellenen kuyruk yöneticilerinin bir listesini içerir. Düğüm başına bir mqc.ini dosyası vardır.
- Yeni bir kuyruk yöneticisi yarattığınızda, otomatik olarak yeni bir kuyruk yöneticisi yapılanış dosyası (qm.ini) yaratılır. Kuyruk yöneticisine ilişkin yapılanış deęiştirgelerini içerir.

AMQP hizmetini kurduysanız, yedeklemeniz gereken ek bir konfigürasyon dosyası vardır:

- **Windows** Windows sistemlerinde: amqp\_win.properties
- **Linux** **AIX** AIX and Linux sistemlerinde: amqp\_unix.properties

### İlgili görevler

[“Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerini deęiştirme” sayfa 83](#)  
Yapılandırma (.ini) dosyalarındaki bilgileri düzenleyerek, IBM MQ ya da tek bir kuyruk yöneticisinin davranışını kuruluşunuzun gereksinimlerine uyacak şekilde deęiştirebilirsiniz. IBM MQ MQI clients için yapılandırma seçeneklerini de deęiştirebilirsiniz.

[“IBM MQ kuyruk yöneticisi verilerinin yedeklenmesi ve geri yüklenmesi” sayfa 655](#)

Kuyruk yöneticilerini, kuyruk yöneticilerini ve kuyruk yöneticisi verilerini yedekleyerek, yalnızca kuyruk yöneticisi yapılanışını yedekleyerek ve bir yedek kuyruk yöneticisini kullanarak, donanım hatalarından kaynaklanan olası bozulmaya karşı koruyabilirsiniz.

## İstemci ile sunucu arasındaki bağlantıların yapılandırılması

IBM MQ MQI clients ile sunucular arasındaki iletişim bağlantılarını yapılandırmak için iletişim protokolünüze karar verin, bağlantının her iki ucundaki bağlantıları tanımlayın, bir dinleyici başlatın ve kanalları tanımlayın.

### Bu görev hakkında

IBM MQ içinde, nesnelere arasındaki mantıksal iletişim bağlantılarına *kanallardır*. IBM MQ MQI clients 'i sunuculara bağlamak için kullanılan kanallara MQI kanalları denir. IBM MQ MQI client üzerindeki IBM MQ

uygulamanızın sunucudaki kuyruk yöneticisiyle iletişim kurabilmesi için, bağlantınızın her bir ucunda kanal tanımlamalarını ayarlayabilirsiniz.

MQI kanallarınızı tanımlamadan önce, hangi iletişim biçimini kullanacağınıza karar vermeli ve kanalın her sonunda bağlantıyı tanımlamalısınız.

IBM MQ MQI client ile farklı fiziksel ağlarda bulunan bir kuyruk yöneticisi arasında ya da bir güvenlik duvarı üzerinden iletişim kuran bir MQI kanalı tanımlıyorsanız, IBM MQ Internet Pass-Thru komutunu kullanarak yapılandırmayı basitleştirebilirsiniz. Daha fazla bilgi için bkz. [IBM MQ Internet Pass-Thru](#).

## Yordam

1. Kullanacağınız iletişim biçimine karar verin.

Bkz. [“Kullanılacak iletişim tipi” sayfa 15](#).

2. Kanalın her sonunda bağlantıyı tanımlayın.

Bağlantıyı tanımlamak için aşağıdakileri yapmanız gerekir:

a) Bağlantıyı yapılandırın.

b) Kanal tanımlamaları için gerek duyduğunuz parametrelerin değerlerini kaydedin.

c) Bir *dinleyici* başlatarak, sunucunun IBM MQ MQI clientürününüzden gelen ağ isteklerini algılamasını sağlayın.

## İlgili kavramlar

[“IBM MQ MQI client yapılandırma dosyası, mqclient.ini” sayfa 154](#)

Bir metin dosyasındaki öznitelikleri kullanarak istemcilerinizi yapılandırabilirsiniz. Bu öznitelikler, ortam değişkenleri ya da platforma özgü başka yollarla geçersiz kılınabilir.

## İlgili görevler

[“IBM MQ ortam değişkenlerinin kullanılması” sayfa 62](#)

Geçerli ayarları görüntülemek ya da IBM MQ ortam değişkenlerinin değerlerini sıfırlamak için komutları kullanabilirsiniz.

[IBM MQ MQI istemci uygulamalarının kuyruk yöneticilerine bağlanması](#)

## İlgili başvurular

[CHLAUTH 'U GÖRÜNTÜLE](#)

[CHLAUTH AYARLA](#)

## Kullanılacak iletişim tipi

Farklı platformlar farklı iletişim protokollerini destekler. İletim iletişim kuralı seçiminiz, IBM MQ MQI client ve sunucu platformlarının birleşimine bağlıdır.

## MQI kanalları için iletim protokolü tipleri






















İstemci ve sunucu platformlarınıza bağlı olarak, MQI kanalları için en çok dört tip iletim protokolü vardır:

- TCP/IP
- LU 6.2
- NetBIOS
- SPX.



MQI kanallarınızı tanımladığınızda, her kanal tanımlaması bir iletim protokolü (iletim tipi) özniteliği belirtmelidir. Bir sunucu tek bir protokolle sınırlı değildir, bu nedenle farklı kanal tanımları farklı protokoller belirtebilir. IBM MQ MQI clients için, farklı iletim protokolleri kullanan alternatif MQI kanallarının olması yararlı olabilir.

İletim protokolü seçiminiz, IBM MQ istemci ve sunucu platformlarının belirli birleşimine de bağlıdır. Olası birleşimler aşağıdaki çizelgede gösterilir.

Çizelge 1. İletim iletişim kuralları- IBM MQ MQI client ve sunucu platformlarının birleşimi

İletim protokolü	IBM MQ MQI client	IBM MQ sunucu
TCP/IP "1" sayfa 16	 IBM i  AIX  Linux  Windows	 IBM i  AIX  Linux  Windows  z/OS
LU 6.2	 AIX  Linux "2" sayfa 16  Windows	 IBM i  AIX  Linux "2" sayfa 16  Windows  z/OS
NetBIOS	 Windows	 Windows
SPX.	 Windows	 Windows

#### Notlar:

-   TCP/IP kullanan bir ileti kanalı, ağ verimini önemli ölçüde artıracak hızlı bir TCP/IP tüneli sağlayan bir IBM Aspera faspio Gateway' e işaret edilebilir. Bkz. [Linux ya da Windows üzerinde Aspera gateway bağlantısı tanımlanması](#).
- Linux (POWER platformu) dışında

#### İlgili kavramlar

"Windows üzerinde TCP bağlantısı tanımlanması" sayfa 250

Hedefin adresini belirlemek için gönderme ucunda bir kanal konfigürasyonu tanımlayarak ve alıcı ucunda bir dinleyici programı çalıştırarak bir TCP bağlantısı tanımlayın.

"AIX and Linux üzerinde TCP bağlantısı tanımlanması" sayfa 257

Gönderme ucundaki kanal tanımı, hedefin adresini belirler. Dinleyici ya da inet cini, alma sonunda bağlantı için yapılandırıldı.

"IBM i üzerinde TCP bağlantısı tanımlanması" sayfa 277

Bağlantı Adı alanını kullanarak kanal tanımında bir TCP bağlantısı tanımlayabilirsiniz.

"z/OS üzerinde TCP bağlantısı tanımlanması" sayfa 975

TCP bağlantısı tanımlamak için, konfigürasyonu tanımlanacak birçok ayar vardır.

"Windows üzerinde LU 6.2 bağlantısı tanımlanması" sayfa 251

SNA, iki makine arasında bir LU 6.2 etkileşimi kurulabilecek şekilde yapılandırılmalıdır.

"AIX and Linux üzerinde LU 6.2 bağlantısı tanımlanması" sayfa 261

SNA, iki makine arasında bir LU 6.2 etkileşimi kurulabilecek şekilde yapılandırılmalıdır.

"IBM i üzerinde LU 6.2 bağlantısı tanımlanması" sayfa 278

Tam olarak nitelenmiş LU 6.2 bağlantısının kip adını, TP adını ve bağlantı adını kullanarak LU 6.2 iletişim ayrıntılarını tanımlayın.

[“Windows üzerinde NetBIOS bağlantısı tanımlanması” sayfa 253](#)

NetBIOS bağlantısı yalnızca Windowsçalıştıran bir istemci ve sunucu için geçerlidir. IBM MQ , başka bir IBM MQ ürününe NetBIOS bağlantısı kurarken üç tip NetBIOS kaynağı kullanır: oturumlar, komutlar ve adlar. Bu kaynakların her birinin bir sınırı vardır; bu sınır varsayılan olarak ya da NetBIOSkuruluşu sırasında belirlenmiştir.

### İlgili görevler

[“Linux ya da Windows platformlarında Aspera gateway bağlantısı tanımlanması” sayfa 827](#)  
IBM Aspera faspio Gateway , IBM MQiçin ağ verimini önemli ölçüde artırabilen hızlı bir TCP/IP tüneli sağlar. Herhangi bir yetkili platformda çalışan bir kuyruk yöneticisi, Aspera gatewayaracılığıyla bağlanabilir. Ağ geçidinin kendisi Red Hat ya da Ubuntu Linuxya da Windowsüzerinde devreye alınır.

### İlgili başvurular

[“TCP/IP bağlantı sınırları” sayfa 17](#)

Tek bir TCP/IP kapısında kuyruğa alınabilecek işlem bekleyen bağlantı isteklerinin sayısı altyapıya bağlıdır. Sınırı ulaşırsa bir hata oluşur.







[“APPC/MVS kullanılarak z/OS için LU6.2 bağlantısı tanımlanması” sayfa 978](#)

LU6.2 bağlantısı tanımlamak için, yapılandırılacak bazı ayarlar vardır.

## TCP/IP bağlantı sınırları

Tek bir TCP/IP kapısında kuyruğa alınabilecek işlem bekleyen bağlantı isteklerinin sayısı altyapıya bağlıdır. Sınırı ulaşırsa bir hata oluşur.

Bu bağlantı sınırı, bir IBM MQ sunucusuna bağlayabileceğiniz istemci sayısı üst sınırıyla aynı değildir. Bir sunucuya, sunucu sistem kaynaklarının belirlediği düzeye kadar daha çok istemci bağlayabilirsiniz. Bağlantı isteklerine ilişkin birikim değerleri aşağıdaki çizelgede gösterilir:

Sunucu altyapısı	Bağlantı isteği sayısı üst sınırı
 AIX	100
 Linux	100
 IBM i	255
 Windows Sunucusu	100
 Windows İş İstasyonu	100
 z/OS	255

Bağlantı sınırına ulaşırsa, istemci MQCONN çağrısından MQRC\_HOST\_NOT\_ALLOWED dönüş kodunu alır ve istemci hata günlüğünde bir AMQ9202 hatası ( AIX and Linux sistemlerinde /var/mqm/errors/AMQERR0n . LOG ya da Windowsüzerinde IBM MQ istemci kuruluşunun hata alt dizininde amqerr0n . log ) bulunur. İstemci MQCONN isteğini yeniden denerse başarılı olabilir.

Yapabileceğiniz bağlantı isteklerinin sayısını artırmak ve bu sınırlama tarafından oluşturulan hata iletilerini önlemek için, her biri farklı bir kapıda dinleyen birden çok dinleyiciniz olabilir ya da birden çok kuyruk yöneticiniz olabilir.





## IBM MQ MQI client nasıl ayarlanır

Bir istemci kurmak için aşağıdaki yönergeleri izleyin.

## Başlamadan önce

Bir IBM MQ MQI client kurmak için, istemcinizin bağlanacağı, önceden kurulmuş ve çalışır durumda olan bir IBM MQ sunucunuz olmalıdır.

## Yordam

1. IBM MQ MQI istemcisi için uygun bir platformunuz olup olmadığını ve donanım ve yazılımın gereksinimleri karşılayıp karşılamadığını denetleyin.  
Platform desteği, [IBM MQ istemcileri için platform desteği](#) başlıklı konuda açıklanmaktadır.
2. IBM MQ ürününü istemci iş istasyonunuza nasıl kuracağınıza karar verin ve istemci ve sunucu altyapıları birleşiminize ilişkin yönergeleri izleyin.  
Kuruluş aşağıdaki konularda açıklanmıştır:
  -  [IBM MQ istemcisinin AIX](#)
  -  [Linux üzerinde IBM MQ istemcisi kurulması](#)
  -  [Windows üzerinde IBM MQ istemcisi kurulması](#)
  -  [IBM i üzerinde IBM MQ istemcisi kurulması](#)
3. İletişim bağlantılarınızın yapılandırıldığından ve bağlandığından emin olun.  
[İletişim bağlantılarının yapılandırılması](#) Sunucu ve istemci arasındaki bağlantıların yapılandırılması başlıklı konuda açıklanmaktadır.
4. Kuruluşunuzun doğru çalışıp çalışmadığını denetleyin.  
Kuruluşunuzun kullandığı platforma ya da platformlara ilişkin kuruluş yordamının doğrulama bölümüne bakın.
5. IBM MQ MQI client kuruluşunu doğruladığınızda, istemcinizin güvenliğini sağlamanız gerekip gerekmediğini göz önünde bulundurun.  
[İstemci güvenliği](#) [IBM MQ MQI client güvenliğinin ayarlanması](#) başlıklı konuda açıklanmıştır.
6. İstemcide çalıştırmak istediğiniz IBM MQ uygulamalarının gerektirdiği IBM MQ MQI istemcisi ve sunucusu arasındaki kanalları ayarlayın.  
Kanalların ayarlanması [MQI kanallarının tanımlanması](#) başlıklı konuda açıklanmaktadır. TLS kullanıyorsanız dikkat edilmesi gereken bazı noktalar vardır.  
Bu konular [MQI kanalının TLS kullandığını belirtme](#) başlıklı konuda açıklanmıştır. Kanalları ayarlamak için bir IBM MQ MQI client yapılandırma dosyası ya da IBM MQ ortam değişkenleri kullanmanız gerekebilir. IBM MQ ortam değişkenleri [IBM MQ ortam değişkenlerinin kullanılması](#) başlıklı konuda açıklanmıştır.
7. IBM MQ uygulamalarının tam açıklaması için [Uygulamaları geliştirme](#) başlıklı konuya bakın.
8. IBM MQ MQI client ortamında uygulama tasarlarırken, oluştururken ve çalıştırırken, kuyruk yöneticisi ortamındaki farklılıkları göz önünde bulundurmanız gerekir.  
Bu farklar hakkında bilgi için bkz:
  - [İstemci uygulamasında ileti kuyruğu arabiriminin \(MQI\) kullanılması](#)
  - [IBM MQ MQI clients için uygulama oluşturulması](#)
  - [IBM MQ MQI client uygulamalarının kuyruk yöneticilerine bağlanması](#)
  - [IBM MQ MQI clients ile sorunların çözülmesi](#)

## Genişletilmiş bir işlem istemcisinin yapılandırılması

Bu konu derlemi, hareket yöneticisinin her kategorisi için genişletilmiş işlem işlevinin nasıl yapılandırılacağını açıklar.

Her platform için, genişletilmiş hareket istemcisi aşağıdaki dış hareket yöneticileri için destek sağlar:

## XA uyumlu hareket yöneticileri

Genişletilmiş işlem istemcisi, CICS ve Tuxedo gibi XA uyumlu hareket yöneticilerini desteklemek için XA kaynak yöneticisi arabirimi sağlar.

### Windows

#### Microsoft Transaction Server (yalnızca Windows sistemleri)

Yalnızca Windows sistemlerinde, XA kaynak yöneticisi arabirimi Microsoft Transaction Server (MTS) olanağını da destekler. Genişletilmiş işlemsel istemci ile birlikte sağlanan IBM MQ MTS desteği, MTS ile XA kaynak yöneticisi arabirimi arasındaki köprüyü sağlar.

#### WebSphere Application Server

WebSphere Application Server 6 ve daha sonra bir IBM MQ ileti sistemi sağlayıcısı içerir, bu nedenle genişletilmiş hareket istemcisini kullanmanıza gerek yoktur.

## XA uyumlu hareket yöneticilerini yapılandırma

Önce IBM MQ temel istemcisini yapılandırın, daha sonra bu konulardaki bilgileri kullanarak genişletilmiş hareket işlevini yapılandırın.

**Not:** Bu bölümde, *Distributed Transaction Processing: The XA Specification* bölümünde The Open Group tarafından yayınlanan XA arabirimine ilişkin temel bir anlayışa sahip olduğunuz varsayılmıştır.

Genişletilmiş bir hareket istemcisi yapılandırmak için öncelikle IBM MQ temel istemcisini aşağıda açıklandığı şekilde yapılandırmanız gerekir:

- ▶ **AIX** [AIX üzerinde IBM MQ istemcisi kurulması](#)
- ▶ **Linux** [Linux üzerinde IBM MQ istemcisi kurulması](#)
- ▶ **Windows** [Windows üzerinde IBM MQ istemcisi kurulması](#)
- ▶ **IBM i** [IBM üzerinde IBM MQ istemcisi kurulması](#)

Altyapınıza ilişkin bilgileri kullanarak, CICS ve Tuxedo gibi XA uyumlu bir hareket yöneticisi için genişletilmiş işlem işlevini yapılandırabilirsiniz.

Hareket yöneticisi, kuyruk yöneticisine bağlı istemci uygulamasının kullandığı MQI kanalını kullanarak kaynak yöneticisi olarak bir kuyruk yöneticisiyle iletişim kurar. Hareket yöneticisi bir kaynak yöneticisi (xa\_) işlev çağrısı verdiğinde, çağrıyı kuyruk yöneticisine iletmek ve çıkışı kuyruk yöneticisinden geri almak için MQI kanalı kullanılır.

Hareket yöneticisi, kuyruk yöneticisini kaynak yöneticisi olarak açmak için bir xa\_open çağrısı yayınlayarak MQI kanalını başlatabilir ya da istemci uygulaması MQCONN ya da MQCONNX çağrısı yayınlayarak MQI kanalını başlatabilir.

- Hareket yöneticisi MQI kanalını başlatırsa ve istemci uygulaması daha sonra aynı iş parçacığında MQCONN ya da MQCONNX 'i çağırırsa, MQCONN ya da MQCONNX çağrısı başarıyla tamamlanır ve uygulamaya bir bağlantı tanıtıcısı döndürülür. Uygulama, MQRC\_ALREADY\_CONNECTED neden koduyla bir MQCC\_WARNING tamamlanma kodu almıyor.
- İstemci uygulaması MQI kanalını başlatırsa ve hareket yöneticisi daha sonra aynı iş parçacığında xa\_open yöntemini çağırırsa, xa\_open çağrısı MQI kanalı kullanılarak kuyruk yöneticisine iletilir.

Başarısızlığın ardından bir kurtarma durumunda, hiçbir istemci uygulaması çalışmıyorsa, hareket yöneticisi, hata sırasında kuyruk yöneticisinin katıldığı tamamlanmamış iş birimlerini kurtarmak için özel olarak ayrılmış bir MQI kanalını kullanabilir.

XA uyumlu bir hareket yöneticisiyle genişletilmiş bir işlem istemcisi kullanırken aşağıdaki koşulları göz önünde bulundurun:

- Tek bir iş parçacığı içinde, bir istemci uygulaması aynı anda yalnızca bir kuyruk yöneticisine bağlanabilir. Bu kısıtlama yalnızca genişletilmiş bir hareket istemcisi kullanıldığında geçerlidir; IBM MQ temel istemcisini kullanan bir istemci uygulaması, tek bir iş parçacığı içinde eşzamanlı olarak birden çok kuyruk yöneticisine bağlanabilir.
- Bir istemci uygulamasının her iş parçacığı farklı bir kuyruk yöneticisine bağlanabilir.



- Bir istemci uygulaması paylaşılan bağlantı tanıtıcılarını kullanamaz.

Genişletilmiş hareket işleme işlevini yapılandırmak için, kaynak yöneticisi görevi gören her kuyruk yöneticisi için hareket yöneticisine aşağıdaki bilgileri sağlamanız gerekir:

- xa\_open dizgisi
- Bir XA anahtar yapısına ilişkin gösterge

Hareket yöneticisi, kuyruk yöneticisini kaynak yöneticisi olarak açmak için xa\_open 'i çağırdığında, xa\_open dizgisini, çağrıda xa\_infobağımsız değişkeni olarak genişletilmiş hareket istemcisine geçirir. Genişletilmiş hareket istemcisi, xa\_open dizgisindeki bilgileri aşağıdaki şekillerde kullanır:

- Sunucu kuyruk yöneticisine bir MQI kanalı başlatmak için, istemci uygulaması henüz başlatmadıysa
- Hareket yöneticisinin kaynak yöneticisi olarak açacağı kuyruk yöneticisinin, istemci uygulamasının bağlandığı kuyruk yöneticisiyle aynı olup olmadığını denetlemek için
- Kuyruk yöneticisi dinamik kaydı kullanıyorsa, hareket yöneticisinin ax\_reg ve ax\_unreg işlevlerini bulmak için

xa\_open dizgisinin biçimi ve xa\_open dizgisindeki bilgilerin genişletilmiş bir işlem istemcisi tarafından nasıl kullanıldığıyla ilgili daha fazla ayrıntı için bkz. [“xa\\_open dizgisinin biçimi” sayfa 21.](#)

XA anahtar yapısı, hareket yöneticisinin genişletilmiş hareket istemcisi tarafından sağlanan xa \_ işlevlerini bulmasını sağlar ve kuyruk yöneticisinin dinamik kayıt kullanıp kullanmadığını belirtir. Genişletilmiş bir işlem istemcisiyle sağlanan XA anahtar yapılarıyla ilgili bilgi için bkz. [“XA anahtar yapıları” sayfa 25.](#)

Belirli bir hareket yöneticisi için genişletilmiş hareket işlevinin nasıl yapılandırılacağına ve hareket yöneticisinin genişletilmiş bir işlem istemcisiyle kullanılmasına ilişkin diğer bilgiler için aşağıdaki kısımlara bakın:

- [“CICS için genişletilmiş bir işlem istemcisinin yapılandırılması” sayfa 26](#)
- [“Tuxedo için genişletilmiş bir işlem istemcisinin yapılandırılması” sayfa 27](#)

### **İlgili kavramlar**

[“xa\\_open dizgisinin CHANNEL, TRPTYPE, CONNAME ve QMNAME parametreleri” sayfa 23](#)

Genişletilmiş hareket istemcisinin, bağlantı kurulacak kuyruk yöneticisini saptamak için bu değiştireleri nasıl kullandığını anlamak için bu bilgileri kullanın.

[“xa\\_open için ek hata işleme” sayfa 24](#)

xa\_open çağrısı belirli durumlarda başarısız olur.

### **İlgili görevler**

[“TLS kanallarıyla genişletilmiş işlem istemcisini kullanma” sayfa 25](#)

xa\_open dizgisini kullanarak TLS kanalı oluşturamazsınız. İstemci kanal tanımlama çizelgesini (ccdt) kullanmak için bu yönergeleri izleyin.

### **İlgili başvurular**

[“TPM ve AXLIB parametreleri” sayfa 24](#)

Genişletilmiş bir hareket istemcisi, hareket yöneticisinin ax\_reg ve ax\_unreg işlevlerini bulmak için TPM ve AXLIB değiştirelerini kullanır. Bu işlevler yalnızca kuyruk yöneticisi dinamik kayıt kullanıyorsa kullanılır.

[“Genişletilmiş hareket işlemede bir hata oluşması durumunda kurtarma” sayfa 24](#)

Bir hatanın ardından, hareket yöneticisinin eksik iş birimlerini kurtarabilmesi gerekir. Bunu yapmak için, hareket yöneticisinin, hata sırasında tamamlanmamış bir iş birimine katılan herhangi bir kuyruk yöneticisi olarak açılabilmesi gerekir.

## **z/OS Genişletilmiş işlemsel istemci bağlantıları için IBM MQ for z/OS ile ilgili önemli noktalar**

Bazı XA hareket yöneticileri, normalde IBM MQ for z/OS' e bağlanan istemcilerin kullanabildiği özelliklerle uyumlu olmayan hareket koordinasyonu çağrılarını dizilerini kullanır.

Uyumsuz bir sıra algılandığında, IBM MQ for z/OS bağlantı için olağandışı bir sona erdirme yayınlayıp istemciye bir hata yanıtı döndürebilir.



Örneğin, `xa_prepare`, 5C6-00D4007Dolağandışı sonunu alır; dönüş kodu -3 (XAER\_RMERR) istemciye döndürülür.

Başka bir örnek, `xa_end` 'un olağandışı sonlanma 5C6-00D40079' un alınmasıdır.

Bu durumla karşılaşan hareket yöneticileri için, hareket yöneticisinin IBM MQ for z/OS ile etkileşimde bulunmasını sağlamak üzere aşağıdaki işlemi gerçekleştirin.

IBM MQ for z/OS üzerinde XA istemci bağlantılarında, hareket yöneticisinin farklı bir bağlantıda hareket hazırlaması için gereken değişiklikleri etkinleştirdiğinizden emin olun.

#### Notlar:

- Değişiklik varsayılan olarak etkinleştirilmez. Değişikliği kullanmak için, XA istemcisi tarafından kullanılan SVRCONN kanalının açıklama alanında herhangi bir yerde CSQSERVICE1 anahtar sözcüğünü (büyük harfle) belirtmeniz gerekir.
- CSQSERVICE1 anahtar sözcüğünü içeren kanallar aşağıdaki kısıtlamalara sahiptir:
  - GROUP kurtarma atma birimine izin verilmez. Yalnızca QMGR kurtarma atma birimine izin verilir. Yok etme, `xa_open` çağrısında belirtilen ada göre belirlenir. Kuyruk paylaşım grubu adı kullanılırsa, XA bağlantısı bir grup kurtarma birimi ister.  
**xa\_info** değiştirgesinde kuyruk paylaşım grubu adını belirten bir `xa_open` çağrısı `xaer_invalile` başarısız olur.
  - `MQGMO_LOCK` ve `MQGMO_UNLOCK` seçeneklerine izin verilmez. `MQGMO_LOCK` ya da `MQGMO_UNLOCK` içeren bir MQGET çağrısı `MQRC_ENVIRONMENT_ERROR` ile başarısız oluyor.

Değişiklik, IBM MQ for z/OS 9.0 üzerinden [APAR P173410](#) aracılığıyla etkinleştirilmiştir.

#### İlgili kavramlar

[“XA uyumlu hareket yöneticilerini yapılandırma” sayfa 19](#)

Önce IBM MQ temel istemcisini yapılandırın, daha sonra bu konulardaki bilgileri kullanarak genişletilmiş hareket işlevini yapılandırın.

#### ***xa\_open*** dizgisinin biçimi

Bir `xa_open` dizgisi, tanımlı değiştirge adları ve değerleri çiftlerini içerir.

Bir `xa_open` dizgisinin biçimi şöyledir:

```
parm_name1 = parm_value1, parm_name2 = parm_value2, ...
```

Burada *parm\_name* bir parametrenin adı, *parm\_value* ise bir parametrenin değeridir. Parametrelerin adları büyük ve küçük harfe duyarlı değildir, aksi belirtilmedikçe parametrelerin değerleri büyük ve küçük harfe duyarlıdır. Parametreleri herhangi bir sırada belirtebilirsiniz.

Değiştirgelerin adları, anlamları ve geçerli değerleri şunlardır:

#### Ad

##### **Anlam ve geçerli değerler**

#### Kanal

MQI kanalının adı.

İsteğe bağlı bir değiştirgedir. Bu değiştirge belirtilirse, CONNAME değiştirgesi de belirtilmelidir.

#### İZLEMA TIPI

MQI kanalına ilişkin iletişim protokolü. Aşağıdaki protokoller geçerli değerlerdir:

##### **LU62**

SNA LU 6.2

##### **NETBIOS**

NetBIOS

##### **SPX.**

IPX/SPX

## TCP

TCP/IP

İsteğe bağlı bir deęiřtirgedir. Atlanırsa, TCP ' nin varsayılan deęeri kabul edilir. Parametrenin deęerleri büyük ve küçük harfe duyarlı deęildir.

## KONADI

MQI kanalının sunucu sonundaki kuyruk yöneticisinin aę adresi. Bu parametrenin geęerli deęerleri, TRPTYPE parametresinin deęerine baęlıdır:

## LU62

Bir CPI-C yan bilgi giriřini tanıtan sembolik hedef adı.

Ortak LU ' nun aę nitelenmiř adı geęerli bir deęer deęil, ortak LU dięeradı da deęil. Bunun nedeni, hareket programı (TP) adı ve kip adı belirtmek için ek parametre olmamasıdır.

## NETBIOS

NetBIOS adı.

## SPX.

4 baytlık bir aę adresi, 6 baytlık bir düęüm adresi ve isteğe baęlı 2 baytlık bir yuva numarası. Bu deęerlerin onaltılı gösterimde belirtilmesi gerekir. Bir nokta, aę ve düęüm adreslerini ayırmalı ve saęlanırsa yuva numarası parantez içine alınmalıdır. Örneęin:

```
0a0b0c0d.804abcde23a1(5e86)
```

Yuva numarası atlanırsa, varsayılan 5e86 deęeri varsayılır.

## TCP

Bir anasistem adı ya da IP adresi; isteğe baęlı olarak bunu parantez içinde bir kapı numarası izler. Kapı numarası atlanırsa, varsayılan deęer olan 1414 varsayılır. Bir kuyruk yöneticisi için birden çok anasistem ve kapı, noktalı virgül ayırıcı kullanılarak belirtilebilir; örneęin:

```
host1(1415);host2(1416);host3(1417)
```

İsteğe baęlı bir deęiřtirgedir. Bu parametre saęlanırsa, CHANNEL parametresi de saęlanmalıdır.

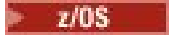
## QMNAME

MQI kanalının sunucu sonundaki kuyruk yöneticisinin adı. Ad boşluk ya da tek bir yıldız iřareti (\*) olamaz ve ad bir yıldız iřaretiyle başlayamaz. Bu, deęiřtirgenin belirli bir kuyruk yöneticisini ada göre tanımlaması gerektięi anlamına gelir.

Bu zorunlu bir parametredir.

Bir istemci uygulaması belirli bir kuyruk yöneticisine baęlandığında, herhangi bir hareket kurtarma iřlemi aynı kuyruk yöneticisi tarafından iřlenmelidir.

Uygulama bir z/OS kuyruk yöneticisine baęlanıyorsa, uygulama belirli bir kuyruk yöneticisinin adını ya da bir kuyruk paylaşım grubunun adını (QSG) belirtebilir. Kuyruk yöneticisi adını ya da kuyruk paylaşım grubu adını kullanarak uygulama, QMGR kurtarma yok etme birimiyle ya da GROUP kurtarma yok etme birimiyle bir harekete katılıp katılmayacaęını denetler. GROUP kurtarma birimi, hareketin QSG ' nin herhangi bir üyesinde iřlenmesini saęlar. GROUP kurtarma birimlerini kullanmak için **GROUPUR** kuyruk yöneticisi öznitelięi etkinleřtirilmelidir.

 GROUP kurtarma biriminin kullanılmasıyla ilgili ek bilgi için [Kuyruk paylaşım grubundaki kurtarma atma birimibařlıklı konuya](#) bakın.

## TPM

Kullanılan hareket yöneticisi. Geęerli deęerler: CICS ve TUXEDO.

Geniřletilmiř bir hareket istemcisi bu deęiřtirgeyi ve AXLIB deęiřtirgesini aynı amaç için kullanır. Bu parametrelere iliřkin ek bilgi için bkz. [TPM ve AXLIB parametreleri](#).

İsteğe baęlı bir deęiřtirgedir. Parametrenin deęerleri büyük ve küçük harfe duyarlı deęildir.

## AXLIB

Hareket yöneticisinin ax\_reg ve ax\_unreg işlevlerini içeren kitaplığın adı.

İsteğe bağlı bir değiştirgedir.

## Kullanıcı Kimliği

Kimlik doğrulaması için kuyruk yöneticisine sağlanan kullanıcı kimliği. Bu parametre sağlanırsa, **PWD** parametresi de sağlanmalıdır. Sağlanan kullanıcı kimliği ve parola doğrulanır ve hareket yöneticisinin bağlantısını tanımlamak için kullanıcı kimliği kullanılır. Kullanıcı kimliği ve parola, MQCONNX çağrısında MQCSP nesnesine veri yerleştirir.

**UID** ve **PWD** parametreleri hem istemci hem de sunucu bağ tanımları için geçerlidir.

## PWD

Kimlik doğrulaması için kuyruk yöneticisine sağlanan parola. Bu parametre sağlanırsa, **UID** parametresi de sağlanmalıdır.

**Uyarı:** Bazı durumlarda, bir istemci uygulamasına ilişkin MQCSP yapısındaki parola ağ üzerinden düz metin olarak gönderilir. İstemci uygulaması parolalarının uygun şekilde korunduğundan emin olmak için bkz. [IBM MQCSP parola koruması](#).

Aşağıda bir xa\_open dizgisi örneği verilmiştir:

```
channel=MARS.SVR,trptype=tcp,connname=MARS(1415),qmname=MARS,tpm=cics
```

## **xa\_open dizgisinin CHANNEL, TRPTYPE, CONNAME ve QMNAME parametreleri**

Genişletilmiş hareket istemcisinin, bağlantı kurulacak kuyruk yöneticisini saptamak için bu değiştirgeleri nasıl kullandığını anlamak için bu bilgileri kullanın.

xa\_open dizgisinde **CHANNEL** ve **CONNAME** değiştirgeleri belirtilirse, genişletilmiş hareket istemcisi bu değiştirgeleri ve **TRPTYPE** değiştirgesini kullanarak sunucu kuyruk yöneticisine bir MQI kanalı başlatır.

xa\_open dizgisinde **CHANNEL** ve **CONNAME** değiştirgeleri sağlanmazsa, genişletilmiş hareket istemcisi MQI kanalını başlatmak için MQSERVER ortam değişkeninin değerini kullanır. MQSERVER ortam değişkeni tanımlı değilse, genişletilmiş hareket istemcisi **QMNAME** değiştirgesiyle tanımlanan istemci kanal tanımlamasındaki girişi kullanır.

Bu durumların her birinde, genişletilmiş hareket istemcisi **QMNAME** değiştirgesinin değerinin MQI kanalının sunucu sonundaki kuyruk yöneticisinin adı olup olmadığını denetler. Değilse, xa\_open çağrısı başarısız olur ve hareket yöneticisi uygulamaya başarısızlığı bildirir.

**z/OS** Uygulama **QMNAME** parametre alanında bir kuyruk paylaşım grubu adı kullanıyorsa ve bağlı olduğu kuyruk yöneticisinde GROUPUR özelliği devre dışı bırakıldıysa, xa\_open çağrısı başarısız olur.

**z/OS** Uygulama istemcisi bir z/OS kuyruk yöneticisine bağlanıyorsa, **QMNAME** parametresi için bir kuyruk paylaşım grubu (QSG) adı belirtebilir. Bu, uygulama istemcisinin bir GROUP kurtarma birimiyle bir harekete katılmasını sağlar. Grup kurtarma atma birimiyle ilgili ek bilgi için [Kurtarma atma birimibaşlıklı konuya](#) bakın.

İstemci uygulaması daha sonra hareket yöneticisinin xa\_open çağrısını yayınlamak için kullandığı iş parçacığında MQCONN ya da MQCONNX 'i çağırdığında, uygulama xa\_open çağrısıyla başlatılan MQI kanalı için bir bağlantı tanıtıcısı alır. İkinci bir MQI kanalı başlatılmaz. Genişletilmiş hareket istemcisi, MQCONN ya da MQCONNX çağrısındaki **QMgrName** değiştirgesinin değerinin, MQI kanalının sunucu sonundaki kuyruk yöneticisinin adı olup olmadığını denetler. Değilse, MQCONN ya da MQCONNX çağrısı MQRC\_ANOTHER\_Q\_MGR\_CONNECTED neden koduyla başarısız olur. **QMgrName** değiştirgesinin değeri boşluk ya da tek bir yıldız işaretiyse (\*) ya da yıldız işaretiyle başlıyorsa, MQCONN ya da MQCONNX çağrısı MQRC\_Q\_MGR\_NAME\_ERROR neden koduyla başarısız olur.



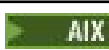

İstemci uygulaması, hareket yöneticisi aynı iş parçacığında xa\_open 'i çağırmadan önce MQCONN ya da MQCONNX çağrılarak bir MQI kanalı başlatmışsa, hareket yöneticisi bu MQI kanalını kullanır. İkinci bir MQI kanalı başlatılmaz. Genişletilmiş hareket istemcisi, xa\_open dizgisindeki **QMNAME** değiştirgesinin değerinin sunucu kuyruk yöneticisinin adı olup olmadığını denetler. Değilse, xa\_open çağrısı başarısız olur.

Bir istemci uygulaması önce bir MQI kanalını başlatırsa, MQCONN ya da MQCONNX çağrısındaki **QMgrName** değiştirgesinin değeri boş olabilir ya da tek bir yıldız işareti (\*) olabilir ya da bir yıldız işaretiyle başlayabilir. Ancak bu koşullarda, uygulamanın bağlandığı kuyruk yöneticisinin, hareket yöneticisinin daha sonra aynı iş parçacığında xa\_open komutunu çağırdığında kaynak yöneticisi olarak açmak istediği kuyruk yöneticisiyle aynı olduğundan emin olmanız gerekir. Bu nedenle, **QMgrName** parametresinin değeri, kuyruk yöneticisini belirttik olarak ada göre tanımlıyorsa, daha az sorunla karşılaşabilirsiniz.

### **TPM ve AXLIB parametreleri**

Genişletilmiş bir hareket istemcisi, hareket yöneticisinin ax\_reg ve ax\_unreg işlevlerini bulmak için TPM ve AXLIB değiştirgelerini kullanır. Bu işlevler yalnızca kuyruk yöneticisi dinamik kayıt kullanıyorsa kullanılır.

TPM parametresi bir xa\_open dizgisinde belirtilirse, ancak AXLIB parametresi sağlanmazsa, genişletilmiş hareket istemcisi TPM parametresinin değerine dayalı olarak AXLIB parametresi için bir değer varsayar. AXLIB parametresinin varsayılan değerleri için bkz. Çizelge 3 sayfa 24 .

<i>Çizelge 3. AXLIB parametresinin varsayılan değerleri</i>		
<b>TPM Değeri</b>	<b>Hizmet olarak sunulan</b>	<b>AXLIB varsayılan değeri</b>
CICS	 AIX	/usr/lpp/encina/lib/libEncServer.a(EncServer_shr.o)
CICS	 Windows Windows sistemleri	libEncSunucusu
Smokin	 AIX	/usr/lpp/tuxedo/lib/libtux.a(libtux.so.60)
Smokin	 Windows Windows sistemleri	libtux

Bir xa\_open dizgisinde AXLIB parametresi sağlanırsa, genişletilmiş hareket istemcisi TPM parametresinin değerine dayalı olarak varsayılan herhangi bir değeri geçersiz kılmak için değerini kullanır. AXLIB parametresi, TPM parametresinin belirtilmiş bir değeri olmayan bir hareket yöneticisi için de kullanılabilir.

### **xa\_open için ek hata işleme**

xa\_open çağrısı belirli durumlarda başarısız olur.

Bu bölümdeki konular, xa\_open çağrılarının başarısız olduğu durumları açıklar. Aşağıdaki durumlardan biri ortaya çıkarsa da başarısız olur:

- xa\_open dizgisinde hatalar var.
- Bir MQI kanalını başlatmak için yeterli bilgi yok.
- MQI kanalını başlatma girişimi sırasında bir sorun oluştu (örneğin, sunucu kuyruk yöneticisi çalışmıyor).

### **Genişletilmiş hareket işlemede bir hata oluşması durumunda kurtarma**

Bir hatanın ardından, hareket yöneticisinin eksik iş birimlerini kurtarabilmesi gerekir. Bunu yapmak için, hareket yöneticisinin, hata sırasında tamamlanmamış bir iş birimine katılan herhangi bir kuyruk yöneticisi olarak açılabilmesi gerekir.

Bu nedenle, yapılandırma bilgilerinde değişiklik yapmadan önce tüm tamamlanmamış iş birimlerinin çözümlendiğinden emin olmanız gerekir.

Diğer bir yöntem olarak, yapılanış değişikliklerinin, hareket yöneticisinin açması gereken kuyruk yöneticilerini açma yeteneğini etkilemediğinden emin olmanız gerekir. Aşağıda, bu tür yapılandırma değişikliklerine ilişkin örnekler verilmiştir:

- xa\_open dizgisinin içeriğini değiştirme
- MQSERVER ortam değişkeninin değerinin değiştirilmesi
- İstemci kanal tanımlama çizelgesindeki (CCDT) girişlerin değiştirilmesi

- Sunucu bağlantısı kanal tanımının silinmesi

## **XA anahtar yapıları**

Her platformda genişletilmiş işlemsel istemci ile birlikte iki XA anahtar yapısı sağlanır.

Bu anahtar yapıları şunlardır:

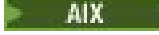


### **MQRMIASwitch**

Bu anahtar yapısı, kaynak yöneticisi olarak hareket eden bir kuyruk yöneticisi dinamik kaydı kullanmadığında bir hareket yöneticisi tarafından kullanılır.

### **MQRMIASwitchDynamic**

Bu anahtar yapısı, kaynak yöneticisi olarak hareket eden bir kuyruk yöneticisi dinamik kaydı kullandığında bir hareket yöneticisi tarafından kullanılır.

Bu anahtar yapıları, Çizelge 4 sayfa 25’inde gösterilen kitaplıklarda bulunur.

<i>Çizelge 4. XA anahtar yapılarını içeren IBM MQ kitaplıkları</i>	
<b>Hizmet olarak sunulan</b>	<b>XA anahtar yapılarını içeren kitaplık</b>
 AIX  Linux	MQ_INSTALLATION_PATH/lib/libmqcxa
 Windows sistemleri	MQ_INSTALLATION_PATH\bin\mqcxa.dll <sup>1</sup>

MQ\_INSTALLATION\_PATH , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

Her anahtar yapısındaki IBM MQ kaynak yöneticisinin adı MQSeries\_XA\_RMI 'dir, ancak birçok kuyruk yöneticisi aynı anahtar yapısını paylaşabilir.

### **İlgili kavramlar**

“Dinamik kayıt ve genişletilmiş işlemsel işleme” sayfa 25

Dinamik kaydın kullanılması, hareket yöneticisi tarafından verilen xa \_ işlev çağrılarının sayısını azaltabileceği için bir eniyileme biçimidir.

#### *Dinamik kayıt ve genişletilmiş işlemsel işleme*

Dinamik kaydın kullanılması, hareket yöneticisi tarafından verilen xa \_ işlev çağrılarının sayısını azaltabileceği için bir eniyileme biçimidir.

Bir kuyruk yöneticisi dinamik kayıt kullanmıyorsa, hareket yöneticisi her iş biriminde kuyruk yöneticisini içerir. Kuyruk yöneticisinin iş birimi içinde güncellenen kaynakları olmasa da, hareket yöneticisi bunu xa\_start, xa\_end ve xa\_prepare 'i çağırarak yapar.

Bir kuyruk yöneticisi dinamik kayıt kullanıyorsa, bir hareket yöneticisi, kuyruk yöneticisinin bir iş birimine dahil olmadığını varsayarak başlar ve xa\_start 'ı çağırır. Kuyruk yöneticisi daha sonra yalnızca kaynakları eşitleme noktası denetimi içinde güncellenirse iş birimine dahil olur. Bu durumda, genişletilmiş hareket istemcisi kuyruk yöneticisinin katılımını kaydettirmek için ax\_reg 'i çağırır.

### **TLS kanallarıyla genişletilmiş işlem istemcisini kullanma**

xa\_open dizgisini kullanarak TLS kanalı oluşturamazsınız. İstemci kanal tanımlama çizelgesini (ccdt) kullanmak için bu yönergeleri izleyin.

### **Bu görev hakkında**

xa\_open xa\_info dizgisinin sınırlı boyutu nedeniyle, bir kuyruk yöneticisine bağlanmak için xa\_open dizgi yöntemini kullanarak TLS kanalı kurmak için gereken tüm bilgiler geçirilemez. Bu nedenle, istemci kanal tanımlama çizelgesini kullanmalı ya da hareket yöneticiniz izin veriyorsa, xa\_open çağrısına başlamadan önce kanalı MQCONNX ile yaratmalısınız.

İstemci kanal tanımlama çizelgesini kullanmak için aşağıdaki adımları izleyin:

## Yordam

1. Yalnızca zorunlu qmname (kuyruk yöneticisi adı) parametresini içeren bir xa\_open dizgisi belirtin, örneğin: XA\_Open\_String=qmname=MYQM
2. Gerekli TLS parametreleriyle bir CLNTCONN (istemci bağlantısı) kanalı tanımlamak için bir kuyruk yöneticisi kullanın. Kuyruk yöneticisi adını CLNTCONN tanımlamasındaki QMNAME özniteliğine ekleyin. Bu, xa\_open dizgisindeki qmname ile eşleştirilecek.
3. CLNTCONN tanımlamasını bir istemci kanal tanımlama çizelgesinde (CCDT) ya da Windows üzerinde etkin dizinde istemci sisteminin kullanımına açın.
4. CCDT kullanıyorsanız, MQCHLLIB ve MQCHLTAB ortam değişkenlerini kullanarak CLNTCONN kanalının tanımlamasını içeren CCDT 'yi tanıtır. Bu değişkenleri hem istemci uygulaması hem de hareket yöneticisi tarafından kullanılan ortamlarda ayarlayın.

## Sonuçlar

Bu, hareket yöneticisine, CipherSpecde içinde olmak üzere, doğru kimlik doğrulaması için gereken TLS öznitilikleriyle uygun kuyruk yöneticisine bir kanal tanımlaması verir.

### **CICS için genişletilmiş bir işlem istemcisinin yapılandırılması**

Bir CICS bölgesine bir XAD kaynak tanımlaması ekleyerek CICS tarafından kullanılmak üzere genişletilmiş bir hareket istemcisi yapılandırabilirsiniz.



CICS resource definition online (RDO) **cicsadd** komutunu kullanarak XAD kaynak tanımlamasını ekleyin. XAD kaynak tanımlaması aşağıdaki bilgileri belirtir:

- xa\_open dizgisi
- Bir anahtar yükleme dosyasının tam olarak nitelenmiş yol adı

Aşağıdaki platformların her birinde CICS tarafından kullanılmak üzere bir anahtar yükleme dosyası sağlanır:

-  AIX
-  Windows

Her anahtar yükleme dosyası, dinamik kayıt için kullanılan XA anahtar yapısına (MQRMIXASwitchDynamic) bir gösterge döndüren bir işlev içerir. Her bir anahtar yükleme dosyasının tam olarak nitelenmiş yol adı için bkz. Çizelge 5 sayfa 26 .

Çizelge 5. Anahtar yükleme dosyaları	
Hizmet olarak sunulan	Yükleme dosyasını değiştir
 AIX	MQ_INSTALLATION_PATH/lib/amqczsc
 Linux	
Windows	MQ_INSTALLATION_PATH\bin\mqcc4swi.dll <sup>1</sup>

MQ\_INSTALLATION\_PATH , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

Aşağıda, Windows sistemleri için bir XAD kaynak tanımlaması örneği verilmiştir:

```
cicsadd -c xad -r REGION1 WMQXA \  
ResourceDescription="IBM MQ queue manager MARS" \  

```

```
XAOpen="channel=MARS.SVR, trptype=tcp, connname=MARS(1415), qmname=MARS, tpm=cics" \
SwitchLoadFile="C:\Program Files\IBM\MQ\bin\mqcc4swi.dll"
```




CICS bölgesine bir XAD kaynak tanımlaması ekleme hakkında daha fazla bilgi için, altyapınıza ilişkin *CICS Yönetim Başvurusu* ve *CICS Yönetim Kılavuzu* 'na bakın.

CICS 'in genişletilmiş bir işlem istemcisiyle kullanılmasına ilişkin aşağıdaki bilgileri göz önünde bulundurun:

- Bir CICS bölgesine IBM MQ için yalnızca bir XAD kaynak tanımlaması ekleyebilirsiniz. Bu, bir bölgeyle yalnızca bir kuyruk yöneticisinin ilişkilendirilebileceği ve bölgede çalışan tüm CICS uygulamalarının yalnızca o kuyruk yöneticisine bağlanabileceği anlamına gelir. Farklı bir kuyruk yöneticisine bağlanan CICS uygulamalarını çalıştırmak istiyorsanız, uygulamaları farklı bir bölgede çalıştırmamız gerekir.
- Bir bölgedeki her uygulama sunucusu, başlatılırken xa\_open ögesini çağırır ve bölgeyle ilişkilendirilmiş kuyruk yöneticisine bir MQI kanalı başlatır. Bu, bir uygulama sunucusu başlatılmadan önce kuyruk yöneticisinin başlatılması gerektiği anlamına gelir; tersi durumda, xa\_open çağırısı başarısız olur. Daha sonra uygulama sunucusu tarafından işlenen tüm IBM MQ MQI client uygulamaları aynı MQI kanalını kullanır.
- Bir MQI kanalı başlatıldığında ve kanalın istemci ucunda güvenlik çıkışı olmadığında, istemci sistemden sunucu bağlantısı MCA 'ya akan kullanıcı kimliği cics olur. Belirli koşullar altında, kuyruk yöneticisi bu kullanıcı kimliğini, sunucu bağlantısı MCA 'nın daha sonra bir istemci uygulaması adına kuyruk yöneticisi kaynaklarına erişmeye çalıştığında yetki denetimi için kullanır. Yetki denetimleri için bu kullanıcı kimliği kullanılıyorsa, bu kullanıcı kimliğinin erişmesi gereken tüm kaynaklara erişim yetkisi olduğundan emin olmanız gerekir.

Kuyruk yöneticisinin yetki denetimi için bu kullanıcı kimliğini kullanmasıyla ilgili bilgi için [Güvenlik](#) başlıklı konuya bakın.

- IBM MQ istemci sistemlerinde kullanılmak üzere sağlanan CICS görev sonlandırma çıktıları [Çizelge 6 sayfa 27](#) içinde listelenir. Bu çıktıları, IBM MQ sunucu sistemleri için karşılık gelen çıktıları yapılandırdığınız şekilde yapılandırabilirsiniz. Bu nedenle, [CICS kullanıcı çıktılarının etkinleştirilmes](#) başlıklı konuya bakın.


Çizelge 6. CICS görev sonlandırma çıktıları		
Hizmet olarak sunulan	Kaynak	Kitaplık
 AIX	amqzscgx.c	amqzscg
 Linux		
 Windows sistemleri	amqzscgn.c	mqcc1415.dll

### ***Tuxedo için genişletilmiş bir işlem istemcisinin yapılandırılması***

XAD kaynak tanımlamasını Tuxedo tarafından kullanılmak üzere yapılandırmak için UBBCONFIG dosyasını ve kaynak yöneticisi çizelgesini güncelleyin.

XAD kaynak tanımlamasını Tuxedo tarafından kullanılmak üzere yapılandırmak için aşağıdaki eylemleri gerçekleştirin:

- Bir uygulamaya ilişkin Tuxedo UBBCONFIG dosyasının GROUPS bölümünde bir xa\_open dizgisi belirtmek için **OPENINFO** değiştirgesini kullanın. Bunun nasıl yapılacağını gösteren bir örnek için, Tuxedo örnek programlarıyla kullanılmak üzere sağlanan örnek UBBCONFIG dosyasına bakın.

 Aşağıdaki altyapılarda dosyanın adı ubbstxcx.c fğolur:

– AIX



**Windows** Windows, dosyanın adı ubbstxcn.cfg.

- Tuxedo kaynak yöneticisi çizelgesindeki bir kuyruk yöneticisine ilişkin girişte, bir XA anahtar yapısının adını ve yapıyı içeren kitaplığın tam olarak nitelenmiş yol adını belirtin:
  - **AIX** AIX sistemlerinde udataobj/RMdeğerini belirtin.
  - **Windows** Windows' da udataobj\rmdeğerini belirtin.

Her platform için bunun nasıl yapılacağını gösteren bir örnek için bkz. [TUXEDO örnekleri](#). Tuxedo, bir kaynak yöneticisinin dinamik kaydını destekler ve bu nedenle MQRMIXASwitch ya da MQRMIXASwitchDynamickomutunu kullanabilirsiniz.

## **Windows** Microsoft İşlem Sunucusu

Hareket yöneticisi olarak Microsoft Transaction Server (MTS) olanağını kullanabilmeniz için ek yapılandırma gerekmez. Bununla birlikte, dikkat etmesi gereken bazı noktalar vardır.

MTS ' nin genişletilmiş işlem istemcisiyle kullanılmasına ilişkin aşağıdaki bilgileri göz önünde bulundurun:

- Bir MTS uygulaması, bir sunucu kuyruk yöneticisine bağlandığında her zaman bir MQI kanalını başlatır. MTS, hareket yöneticisi rolünde, kuyruk yöneticisiyle iletişim kurmak için aynı MQI kanalını kullanır.
- Bir hatanın ardından, MTS tamamlanmamış iş birimlerini kurtarabilmelidir. Bunu yapmak için MTS, hata sırasında tamamlanmamış bir iş birimine katılan herhangi bir kuyruk yöneticisiyle iletişim kurabilmelidir.

Bir MTS uygulaması bir sunucu kuyruk yöneticisine bağlandığında ve bir MQI kanalı başlattığında, genişletilmiş hareket istemcisi bir hata oluştuktan sonra kanalın yeniden başlatılmasını sağlamak için MQCONN ya da MQCONNX çağrısının değiştirgelerinden yeterli bilgi alır. Genişletilmiş işlem istemci bilgileri MTS ' ye iletir ve MTS bilgileri günlüğüne kaydeder.

MTS uygulaması bir MQCONN çağrısı yaparsa, bu bilgi yalnızca kuyruk yöneticisinin adıdır. MTS uygulaması bir MQCONNX çağrısı yayınlarsa ve bir kanal tanımlama yapısı (MQCD) sağlarsa, bilgiler MQI kanalının adını, sunucu kuyruk yöneticisinin ağ adresini ve kanala ilişkin iletişim protokolünü de içerir.

Bir kurtarma durumunda, MTS bu bilgileri genişletilmiş hareket istemcisine iletir ve genişletilmiş hareket istemcisi bunu MQI kanalını yeniden başlatmak için kullanır.

Herhangi bir yapılandırma bilgisini değiştirmeniz gerekirse, değişiklikleri yapmadan önce tüm tamamlanmamış iş birimlerinin çözümlendiğinden emin olun. Diğer bir seçenek olarak, yapılandırma değişikliklerinin, genişletilmiş hareket istemcisinin MTS tarafından kaydedilen bilgileri kullanarak bir MQI kanalını yeniden başlatma yeteneğini etkilemediğinden emin olun. Aşağıda, bu tür yapılandırma değişikliklerine ilişkin örnekler verilmiştir:

- MQSERVER ortam değişkeninin değerinin değiştirilmesi
  - İstemci kanal tanımlama çizelgesindeki (CCDT) girişlerin değiştirilmesi
  - Sunucu bağlantısı kanal tanımının silinmesi
- MTS ile genişletilmiş bir işlem istemcisi kullanırken aşağıdaki koşullara dikkat edin:
    - Tek bir iş parçacığı içinde, bir istemci uygulaması aynı anda yalnızca bir kuyruk yöneticisine bağlanabilir.
    - Bir istemci uygulamasının her iş parçacığı farklı bir kuyruk yöneticisine bağlanabilir.
    - Bir istemci uygulaması paylaşılan bağlantı tanıtıcılarını kullanamaz.

## **MQI kanallarının tanımlanması**

Yeni bir kanal yaratmak için, aynı kanal adını ve uyumlu kanal tiplerini kullanarak, bağlantının her bir ucu için bir tane olmak üzere **iki** kanal tanımı yaratmanız gerekir. Bu durumda, kanal tipleri *sunucu-bağlantısı* ve *istemci-bağlantısı* ' dir.



## Kullanıcı tanımlı kanallar

Sunucu kanalları otomatik olarak tanımlamadığında, kanal tanımlarını oluşturmanın ve IBM MQ MQI client makinesindeki IBM MQ uygulamasına kanala erişim vermenin iki yolu vardır.

Bu iki yöntem ayrıntılı olarak açıklanmıştır:

1. IBM MQ istemcisinde bir kanal tanımlaması, sunucuda da bir kanal tanımlaması yaratın.

Bu, IBM MQ MQI client ve sunucu platformlarının herhangi bir birleşimi için geçerlidir. Sistemde çalışmaya başladığınızda ya da kuruluşunuzu sınamak için bu programı kullanın.

Bu yöntemin nasıl kullanılacağına ilişkin ayrıntılar için bkz. [“Farklı altyapılarda sunucu bağlantısı ve istemci bağlantısı tanımlamaları yaratılması” sayfa 34](#) .

2. Sunucu makinesinde her iki kanal tanımını da yaratın.

Aynı anda birden çok kanal ve IBM MQ MQI client makinesi ayarlarken bu yöntemi kullanın.

Bu yöntemin nasıl kullanılacağına ilişkin ayrıntılar için bkz. [“Sunucuda sunucu bağlantısı ve istemci bağlantısı tanımlamaları yaratılması” sayfa 40](#) .

## Otomatik olarak tanımlanan kanallar

z/OS dışındaki platformlardaki IBM MQ ürünleri, yoksa, sunucuda otomatik olarak bir kanal tanımlaması yaratabilen bir özellik içerir.

Bir istemciden gelen ekleme isteği alınır ve o kuyruk yöneticisinde uygun bir sunucu bağlantısı tanımı bulunamazsa, IBM MQ otomatik olarak bir tanımlama yaratır ve bunu kuyruk yöneticisine ekler. Otomatik tanımlama, varsayılan sunucu bağlantısı kanalı SYSTEM.AUTO.SVRCONN. CHAD parametresiyle (ya da ChannelAutoDef parametresiyle PCF komutunu Change Queue Manager ile) ALTER QMGR komutunu kullanarak kuyruk yöneticisi nesnesini güncelleyerek, sunucu bağlantısı tanımlarının otomatik olarak tanımlanmasını geçerli kılabilirsiniz.

### İlgili kavramlar

“Kanal denetimi işlevi” sayfa 217

Kanal denetimi işlevi, kanalları tanımlamanız, izlemeniz ve denetlemeniz için olanaklar sağlar.

ALW

## AMQP kanallarının oluşturulması ve kullanılması

IBM MQ kuruluşunuza AMQP Service Component 'le ilişkin IBM MQ desteğini kurduğunuzda, bir kanalı tanımlamak, değiştirmek, silmek, başlatmak ve durdurmak için IBM MQ MQSC komutlarını (**runmqsc**) çalıştırabilirsiniz. Bir kanalın durumunu da görüntüleyebilirsiniz.

### Başlamadan önce

Bu görevde AMQP kanalını kurduğunuz varsayılır. Bunu, IBM MQ' yi kurarken AMQP SERVICE bileşenini seçerek yaparsınız. Daha fazla bilgi için platformunuza ilişkin bağlantıyı izleyin ve "AMQP Service" için tablo satırını bulun:

- **AIX** [AIX sistemleri için IBM MQ bileşenleri](#)
- **Linux** [Linux sistemleri için IBM MQ devir/dakika bileşenleri](#)
- **Linux** [IBM MQ Debian bileşenleri- Linux Ubuntu sistemleri](#)
- **Windows** [IBM MQ özellikleri](#)

**Not:** Bir SERVICE bileşeni örneği için [IBM MQ hizmetinin AMQP için yeniden başlatılması](#) başlıklı konuya bakın ve AMQP hizmetiniz düzgün çalışmayı durdurursa daha fazla bilgi edinin.

Bu görev, varolan bir kuyruk yöneticiniz olduğunu da varsayar.

Kuyruk yöneticisiyle sına bağlantısı kurmak için, OASIS AMQP 1.0 protokolünü uygulayan herhangi bir AMQP istemcisini kullanabilirsiniz; örneğin, MQ Light ve Apache Qpid istemcileri ( Apache Qpid Proton ve Apache Qpid JMS gibi).

**V 9.3.0** IBM MQ 9.3.0' den yalnızca varsayılan SYSTEM.DEF.AMQP, kuyruk yöneticisine MQ Light bağlantılarını sınamak için. Aşağıdaki yordam varsayılan kanalı kullanır.

Bu görev, MQ Light Node.js istemcisini temel alır. Ancak, IBM MQ kuyruk yöneticisiyle ilgili adımlar herhangi bir istemci için aynıdır.

**Not:** AMQP kanalları kullanıcı tanımlı AMQP hizmetlerini desteklemez. AMQP kanalları yalnızca sistemin varsayılan SYSTEM.AMQP.SERVICE hizmeti. Her kuyruk yöneticisi için bu hizmetin yalnızca bir eşgörünümünü tanımlayabilirsiniz.

## Yordam

1. **runmqsc** ürününü `mqinstall/bin/` dizininden başlatın:

```
runmqsc QMNAME
```

2. (Yalnızca kuyruk yöneticiniz IBM MQ 9.0.4 ya da daha önceki bir sürümse gereklidir.) AMQP işlevinin kurulup kurulmadığını ve düzgün çalışıp çalışmadığını denetleyin.

JVM ' yi denetleyen IBM MQ hizmetini başlatmak için **START SERVICE** komutunu kullanın:

```
START SERVICE(SYSTEM.AMQP.SERVICE)
```

**Not:** IBM MQ 9.1 SYSTEM.AMQP.SERVICE için **CONTROL** özniteliği *QMGR* olarak ayarlı. Bu, kuyruk yöneticisi başlatıldığında hizmetin otomatik olarak başlatılmasına neden olur. **CONTROL** özniteliğini *MANUAL* değerine ayarlayarak, kuyruk yöneticisi başlatıldığında hizmetin başlatılmasını önleyebilirsiniz.

Kuyruk yöneticisinin başlatılmasının ardından, AMQP hizmeti ve AMQP kanalı (tanımlıysa) otomatik olarak başlatılır.

3. MCAUSER kullanıcı kimliğini ayarlayın.

Bir AMQP istemcisi bir kanala bağlandığında, kanal, kuyruk yöneticisiyle bağlantılarda kullanılan bir MCAUSER kullanıcı kimliğini belirtir. MCAUSER varsayılan değeri boştur. Herhangi bir AMQP istemcisinin kuyruk yöneticisine bağlanabilmesi için önce, IBM MQ konularını yayınlama ve bunlara abone olma yetkisi olan geçerli bir IBM MQ kullanıcısı olması gereken MCAUSER değerini belirtmeniz gerekir.

**Not:** **Windows** Windows işletim sisteminde, IBM MQ 9.2.0' den önce, MCAUSER kullanıcı kimliği ayarı yalnızca 12 karaktere kadar olan kullanıcı kimlikleri için desteklenir. IBM MQ 9.2.0' den 12 karakter sınırı kaldırılır.

- a) MCAUSER kullanıcı kimliğini ayarlamak için **ALTER CHANNEL** komutunu kullanın:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) MCAUSER(Use ID)
```

- b) MCAUSER kullanıcı kimliğinizi yayınlamak ve konulara abone olmak üzere yetkilendirmek için aşağıdaki iki **setmqaut** komutunu kullanın:

```
setmqaut -m QMNAME -t topic -n SYSTEM.BASE.TOPIC -p MCAUSER  
-all +pub +sub
```

ve

```
setmqaut -m QMNAME -t qmgr -p MCAUSER -all +connect
```

MCAUSER kullanıcı kimliği eklenirken ya da değiştirilirken kanal çalışıyorsa, kanalı durdurup yeniden başlatmanız gerekir.

**Not:** MCAUSER kullanıcı kimliği ayarlanmazsa ya da MCAUSER kullanıcı kimliği IBM MQ konularını yayınlama ya da bunlara abone olma yetkisine sahip değilse, AMQP istemcisinde bir hata iletisi alırsınız.

4. Varsayılan SYSTEM.DEF.AMQP kanalını başlatmak için **START CHANNEL** komutunu kullanın:

```
START CHANNEL(SYSTEM.DEF.AMQP)
```

5. Kanal durumunu denetlemek istiyorsanız, **DISPLAY CHSTATUS** komutunu kullanın:

```
DISPLAY CHSTATUS(SYSTEM.DEF.AMQP) CHLTYPE(AMQP)
```

Kanal doğru çalıştığında, komut çıkışında STATUS (RUNNING) görüntülenir.

6. Varsayılan kapıyı değiştirin.

AMQP 1.0 bağlantıları için varsayılan kapı 5672 'dir. MQ Light' i önceden kurduysanız, 5672 numaralı kapıyı kullanıyorsanız, AMQP kanalınızın kullandığı kapıyı değiştirmeniz gerekir. Kapıyı değiştirmek için **ALTER CHANNEL** komutunu kullanın:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) PORT(NEW PORT NUMBER)
```

7. Kanal kimlik doğrulaması (CHLAUTH) kurallarını kullanarak AMQP kanalına yönelik bağlantıları engellemek ya da süzmek istemiyorsanız, kuyruk yöneticisinde kanal doğrulamasını aşağıdaki gibi devre dışı bırakın:

```
alter qmgr chlauth(disabled)
```

Bir üretim kuyruğu yöneticisinde bağlantı kimlik doğrulamasını devre dışı bırakmanız önerilmez. Yalnızca bir geliştirme ortamında bağlantı kimlik doğrulamasını devre dışı bırakmanız gerekir.

Diğer bir seçenek olarak, kuyruk yöneticisi kanalı kimlik doğrulama kurallarını AMQP kanalına belirli bağlantılara izin verecek şekilde yapılandırın.

8. İsteğe bağlı: Kuyruk yöneticisi için yapılandırılan anahtar havuzunu kullanarak kanalda SSL/TLS şifrelemesini etkinleştirmek istiyorsanız, kanalın SSLCIPH özniteliğini uygun bir şifre belirtimine ayarlamanız gerekir. Varsayılan olarak, şifre belirtimi boştur; bu, kanalda SSL/TLS şifrelemesinin kullanılmadığı anlamına gelir. Bir şifre belirtimi ayarlamak için **ALTER CHANNEL** komutunu kullanın. Örneğin:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLCIPH(CIPHER SPECIFICATION)
```

Ayrıca, SSL/TLS şifrelemesiyle ilişkili olarak aşağıdaki gibi ayarlayabileceğiniz birçok kanal yapılandırma seçeneği vardır:

- Varsayılan olarak, kuyruk yöneticisi anahtar havuzundaki **CERTLABL** özniteliğine karşılık gelen etikete sahip sertifika, kanal için SSL/TLS şifrelemesi tarafından kullanılan addır. **CERTLABL** ayarını tanımlayarak farklı bir sertifika seçebilirsiniz. Gerekli sertifikaya ilişkin etiketi belirtmek için **ALTER CHANNEL** komutunu kullanın:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) CERTLABL(CERTIFICATE LABEL)
```

- Kanalı, SSL/TLS istemci bağlantılarından bir sertifika gerektirecek şekilde ayarlayabilirsiniz. **SSLCAUTH** özniteliğini ayarlayarak bir SSL/TLS istemci bağlantısından sertifika gerekip gerekmediğini seçebilirsiniz. SSL/TLS istemci bağlantısından bir sertifikanın gerekip gerekmediğini ayarlamak için **ALTER CHANNEL** komutunu kullanın. Örneğin:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLCAUTH(REQUIRED or OPTIONAL)
```

- **SSLCAUTH** özneliğini **REQUIRED** olarak ayarlarsanız, istemcideki sertifikanın ayırt edici adı (DN) denetlenir. İstemciden sertifikanın Ayırt Edici Adını denetlemek için **SSLPEER** özneliğini ayarlayın. İstemciden sertifikanın ayırt edici adını denetlemek için **ALTER CHANNEL** komutunu kullanın. Örneğin:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLPEER (DN SPECIFICATION)
```

Alternatif olarak, bu yöntem **SSLPEER** özneliğinin kullanımıyla karşılaştırıldığında daha fazla ayrıntı düzeyi sunduğundan, bağlantılara izin vermek ya da bağlantıları engellemek için kanal kimlik doğrulama kayıtlarını da kullanabilirsiniz. **SSLPEER** ayarını yapma ve kanal kimlik doğrulama kayıtlarını alternatif olarak kullanma hakkında daha fazla bilgi için bkz. [SSL Eşdüzey](#).

9. Aşağıdaki komutu çalıştırarak MQ Light Node.js istemcisini kurun:

```
npm install mqlight
```

10. `node_modules/mqlight/samples` dizinine gidin ve örnek alıcı uygulamasını çalıştırın:

- Varsayılan kapı numarasını kullanıyorsanız, örnek alıcı uygulamasını çalıştırabilirsiniz:

```
node recv.js
```

- AMQP kanalınızı farklı bir kapı numarası kullanacak şekilde yapılandırdıysanız, örnek alıcı uygulamasını yeni kapı numarasını belirtmek için bir parametreyle çalıştırabilirsiniz:

```
node recv.js -s amqp://localhost:6789
```

Varsayılan kanalla başarılı bir bağlantı şu iletiyi görüntüler:

```
Connected to amqp://localhost:5672 using client-id recv_e79c55d
Subscribed to pattern: public
```

Uygulama şimdi kuyruk yöneticisine bağlı ve ileti almayı bekliyor. `public` konusuna abone oldu.

**Not:** `-i` deęiştirgesini kullanarak belirtmezseniz, `client-id` otomatik olarak üretilir.

11. Yeni bir komut penceresinde, `node_modules/mqlight/samples` dizinine gidin ve aşağıdaki komutu çalıştırarak örnek gönderen uygulamasını çalıştırın:

```
node send.js
```

Alıcı uygulamasına ilişkin komut penceresinde `Hello World` (Merhaba Dünya) iletisi görüntülenir.

12. Bir MQ Light örnek iletisi almak için **AMQSSUB** IBM MQ örneğini kullanın.

Linux ve Windows üzerinde, örnek aşağıdaki konumlarda bulunabilir:

- **Linux** Linux üzerindeki `mqinstall/samp/bin` dizini.
- **Windows** Windows üzerindeki `mqinstall/Tools/c/Samples/Bin` dizini.

- a) Aşağıdaki komutu çalıştırarak örneęi çalıştırın:

```
amqssub public QM-name.
```

- b) Aşağıdaki komutu yeniden çalıştırarak IBM MQ uygulamasına bir ileti gönderin:

```
node send.js
```

13. Daha fazla AMQP kanalı oluşturmak için **DEFINE CHANNEL** komutunu kullanın:

```
DEFINE CHANNEL(MY.AMQP.CHANNEL) CHLTYPE(AMQP) PORT(2345)
```

Bir kanal tanımladığınızda, kanal **START CHANNEL** komutu kullanılarak el ile başlatılmalıdır:

```
START CHANNEL(MY.AMQP.CHANNEL)
```

Kanalın doğru çalışıp çalışmadığını denetlemek için, yeni kanalın kapısını belirterek örnek alıcı uygulamasını çalıştırabilirsiniz:

```
node recv.js -s amqp://localhost:2345
```

## Sonraki adım

IBM MQ bağlantılarını görüntülemek, kanalı durdurmak ve kanalı silmek için aşağıdaki komutları kullanabilirsiniz:

**DISPLAY CONN(\*) TYPE(CONN) WHERE (CHANNEL EQ SYSTEM.DEF.AMQP)**

AMQP kanalının kuyruk yöneticisinde yaptığı IBM MQ bağlantısını görüntüler.

**DISPLAY CHSTATUS(\*) CHLTYPE(AMQP) CLIENTID(\*) ALL**

Belirtilen kanala bağlı AMQP istemcilerinin listesini görüntüler.

**STOP CHANNEL (MY.AMQP.CHANNEL)**

Bir AMQP kanalını durdurur ve dinlediği kapıyı kapatır.

**DELETE CHANNEL (MY.AMQP.CHANNEL)**

Oluşturduğunuz kanalları siler.

**Not:** SYSTEM.DEF.AMQP.

**runmqsc** ya da PCF kullanarak AMQP yeteneğinin IBM MQ kuruluşunda kurulu olup olmadığını ve bununla ilişkilendirilmiş bir kuyruk yöneticisi olup olmadığını belirleyebilirsiniz:

- **runmqsc** olanağını kullanarak, kuyruk yöneticisinin özneliklerini görüntüleyin ve AMQPCAP (YES) olup olmadığını denetleyin.
- PCF ' yi kullanarak **MQCMD\_INQUIRE\_Q\_MGR** komutunu kullanın ve MQIA\_AMQP\_CAPABILITY değerini onaylayın.

## İlgili görevler

[AMQP istemci uygulamalarının geliştirilmesi](#)

[AMQP istemcilerinin güvenliğinin sağlanması](#)

## İlgili başvurular

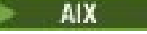
[strmqm](#)

## ALW AMQP kanalının kuyruk yöneticilerinden kaldırılması

Kuruluş dizininden klasörleri kaldırarak, AMQP kanalını kuyruk yöneticilerinden kaldırabilirsiniz.

## Yordam

1. Kuyruk yöneticisini durdurun.
2. AMBP hizmet bileşeni API ' leri için IBM MQ desteğini kaldırın:

-  AIX' da şu komutu çalıştırın:

```
installp -u mqm.amqp.rte
```

- **Linux** Linux üzerinde AMQP RPM 'yi çıkarın. RPM 'yi kurmadan önce yeniden paketlediyseniz, yeniden paketlenen RPM' nin adını belirtin.

```
rpm -e MQSeriesAMQP
```

- **Windows** Windows işletim sistemlerinde, amqp klasörünü IBM MQ kuruluşundan kaldırın. IBM MQ kuruluş yolundaki başka bir dosya ya da klasörün kaldırılmadığından emin olun.

3. Kuyruk yöneticisini yeniden başlatın.

### İlgili görevler

[AMQP istemci uygulamalarının geliştirilmesi](#)

[AMQP istemcilerinin güvenliğinin sağlanması](#)

## ALW AMQP kanal günlüğü dosyaları

AMQP kanallarına ilişkin günlük dosyaları, IBM MQ günlük dosyalarıyla aynı IBM MQ veri dizininde saklanır.

Windows üzerindeki varsayılan veri dizini şudur: C:\ProgramData\IBM\MQ.

Linux üzerindeki varsayılan veri dizini şudur: /var/mqm.

AMQP kanalı, günlük bilgilerini IBM MQ veri dizininde bulunan aşağıdaki günlük dosyalarına yazar:

- amqp.stdout, qmgrs/QM-name klasörüne yazılır.
- amqp.stderr, qmgrs/QM-name klasörüne yazılır.
- amqp\_\*.log , qmgrs/QM-name/errors klasörüne yazılır.

Bir MQ Light istemcisi bir kimlik doğrulama ya da yetkilendirme hatası alırsa, yöneticiniz amqp\_0.log dosyasında ve MQ AMQERR\*.log dosyalarında güvenlik hatasının nedenine ilişkin ayrıntılı bilgileri bulabilir.

FDC dosyaları, data-directory/errors klasörüne yazılan AMQP\*.FDC dosyaları olarak yaratılır.

Bazı yapılandırma dosyaları qmgrs/QM-name/amqp dizinine yazılır. Bu dizindeki dosyaların hiçbirini düzenlemeniz gerekmez.

### İlgili kavramlar

[AIX, Linux, and Windows üzerinde hata günlükleri](#)

### İlgili görevler

[AMQP istemci uygulamalarının geliştirilmesi](#)

[AMQP istemcilerinin güvenliğinin sağlanması](#)

## Farklı altyapılarda sunucu bağlantısı ve istemci bağlantısı tanımlamaları yaratılması

Her kanal tanımını, uygulandığı bilgisayarda yaratabilirsiniz. Ancak, istemci bilgisayarda kanal tanımlarını nasıl yaratabileceğinize ilişkin kısıtlamalar vardır.

### Bu görev hakkında

Tüm altyapılarda, sunucu makinesinde bir sunucu bağlantısı kanalı tanımlamak için IBM MQ Script (MQSC) komutlarını, programlanabilir komut biçimi (PCF) komutlarını ya da IBM MQ Explorer komutunu kullanabilirsiniz.

**z/OS** z/OS işletim ve Denetim panolarını da kullanabilirsiniz.

**IBM i** IBM i üzerinde pano arabirimini de kullanabilirsiniz.

MQSC komutları IBM MQ ' in yalnızca IBM MQ MQI client olarak kurulduğu bir makinede kullanılmadığından, istemci makinesinde istemci-bağlantı kanalı tanımlamak için farklı yollar kullanmanız gerekir.

**runmqsc**' de aşağıdaki noktalar geçerlidir:

- **-c** değiştirgesini ve isteğe bağlı olarak, yönetmek istediğiniz kuyruk yöneticisine istemci olarak **-u** değiştirgesini **runmqsc** bağlayabilirsiniz.
- Bir kullanıcı kimliği sağlamak için **-u** parametresini kullanırsanız, sizden eşleşen bir parola istenir.
- CONNAUTH AUTHINFO kaydını CHCKLOCL (REQUIRED) ya da CHCKLOCL (REQDADM) ile yapılandırdıysanız, **-u** değiştirgesini kullanmanız gerekir; tersi durumda, kuyruk yöneticinizi **runmqsc** ile denetleyemezsiniz.

## Yordam

- Sunucuda bir sunucu bağlantısı kanalı tanımlamak için bkz. [“Sunucuda sunucu bağlantısı kanalı tanımlanması” sayfa 35.](#)
- **MQSERVER** ortam değişkenini kullanarak bir IBM MQ MQI client üzerinde istemci-bağlantı kanalı yaratmak için bkz. [“MQSERVER kullanılarak IBM MQ MQI client üzerinde istemci-bağlantı kanalı yaratılması” sayfa 36.](#)
- MQCONNX çağrısında MQCNO yapısını kullanarak bir IBM MQ MQI client üzerinde istemci-bağlantı kanalı yaratmak için bkz. [“MQCNO kullanarak IBM MQ MQI client üzerinde istemci bağlantısı kanalı yaratılması” sayfa 40.](#)

## Sunucuda sunucu bağlantısı kanalı tanımlanması

Gerekliyse MQSC ' yi başlatın ve sunucu bağlantısı kanalını tanımlayın.

## Yordam

1. İsteğe bağlı: Sunucu altyapınız z/OS değilse, önce bir kuyruk yöneticisi yaratın, başlatın ve MQSC komutlarını başlatın.

- a) QM1 adlı bir kuyruk yöneticisi yaratın; örneğin:

```
crtmqm QM1
```

- b) Kuyruk yöneticisini başlat:

```
strmqm QM1
```

- c) MQSC komutlarını başlat:

```
runmqsc QM1
```

2. Seçtiğiniz ada ve *sunucu-bağlantısı* kanal tipine sahip bir kanal tanımlayın.

```
DEFINE CHANNEL(CHAN1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server-connection to Client_1')
```

Bu kanal tanımlaması, sunucuda çalışan kuyruk yöneticisiyle ilişkilendirilir.

3. Kuyruk yöneticinize gelen bağlantı erişimine izin vermek için aşağıdaki komutu kullanın:

```
SET CHLAUTH(CHAN1) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- Burada SET CHLAUTH, önceki adımda tanımlanan kanalın adını kullanır.
- Burada *'IP adresi'*, istemcinin IP adresidir.

- Burada '*userid*', hedef kuyruklara erişim denetimi için kanala sağlamak istediğiniz tanıtıcıdır. Bu alan büyük ve küçük harfe duyarlıdır.

Gelen bağlantınızı birkaç farklı öznitelik kullanarak tanımlamayı seçebilirsiniz. Örnek, IP adresini kullanır. Diğer öznitelikler arasında istemci kullanıcı kimliği ve TLS Konu Ayırt Edici Adı yer alır. Daha fazla bilgi için bkz. [Kanal kimlik doğrulama kayıtları](#)

## MQSERVER kullanılarak IBM MQ MQI client üzerinde istemci-bağlantı kanalı yaratılması

**MQSERVER** ortam değişkenini kullanarak istemci iş istasyonunda bir istemci bağlantısı kanalı tanımlayabilirsiniz.

### Bu görev hakkında

İstemci-bağlantı kanalının basit bir tanımlamasını belirtmek için **MQSERVER** ortam değişkenini kullanabilirsiniz. Bu yöntemi kullanarak kanalın yalnızca birkaç özneliğini belirtebileceğiniz bir anlamda basittir.

IBM MQ MQI client makineniz ile bir sunucu makinesi arasındaki kanalı tanımlamak için **MQSERVER** ortam değişkenini kullanırsanız, bu kanal uygulamanızın kullanabileceği tek kanaldır ve istemci kanal tanımlama çözümlerine (CCDT) başvuru yapılmaz.

MQCONN ya da MQCONNX isteği, dinleyicinin bağlı olduğu kuyruk yöneticisinden başka bir kuyruk yöneticisini belirtiyorsa ya da **MQSERVER** parametresi *TransportType* tanınmıyorsa, MQCONN ya da MQCONNX isteği MQRC\_Q\_MGR\_NAME\_ERROR dönüş koduyla başarısız olur.

**Linux** **AIX** AIX and Linux üzerinde, **MQSERVER** aşağıdaki örneklerden birinde olduğu gibi tanımlanabilir:

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56(2002)'  
export MQSERVER=CHANNEL1/LU62/BOX99
```

MQCONNX 'e sağlanan MQCNO yapısından bir MQCD yapısına gönderme yapılmadıysa, tüm MQCONN ya da MQCONNX istekleri tanımladığınız kanalı kullanmayı dener; bu durumda, MQCD yapısı tarafından belirtilen kanal **MQSERVER** ortam değişkeninin belirttiği kanallardan önceliklidir.

**MQSERVER** ortam değişkeni, **MQCHLLIB** ve **MQCHLTAB** ortam değişkenlerinin gösterdiği herhangi bir istemci kanal tanımlamasından önceliklidir.

### Yordam

- Altyapınıza bağlı olarak, kanal tanımını **MQSERVER** ile belirtmek için aşağıdaki komutlardan birini kullanın.

– **Windows** Windows' da, aşağıdaki gibi basit bir kanal tanımını belirtin:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName
```

Örneğin:

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
```

– **Linux** **AIX** AIX and Linux' da, aşağıdaki gibi basit bir kanal tanımını belirtin:

```
export MQSERVER=ChannelName/TransportType/ConnectionName
```

Örneğin:



```
SET MQSERVER=SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)
```

- **IBM i** IBM i' da, aşağıdaki gibi basit bir kanal tanımlarını belirtin:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('ChannelName/TransportType/ConnectionName')
```

Örneğin:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)')
```

#### Notlar:

- *ChannelName* , sunucuda tanımlanan adla aynı olmalıdır. Kanal adını, iletim tipini ve bağlantı adını ayırmak için bu karakter kullanıldığından, eğik çizgi (/) karakterini içeremez. Bir istemci kanalı tanımlamak için **MQSERVER** ortam değişkeni kullanıldığında, ileti uzunluğu üst sınırı olan (**MAXMSGL**) 100 MB kullanılır. Bu nedenle, kanal için geçerli olan ileti büyüklüğü üst sınırı, sunucudaki SVRCONN kanalında belirtilen değerdir.
- *TransportType* , IBM MQ istemci altyapınıza bağlı olarak LU62, TCP, NETBIOS, SPX olabilir.
- **Linux** **AIX** AIX and Linux işletim sistemlerinde *TransportType* büyük ve küçük harfe duyarlıdır ve büyük olmalıdır. MQCONN ya da MQCONNX çağrısı, iletim tipi tanınmıyorsa 2058 değerini döndürür.
- *ConnectionName* , iletişim protokolünde (*TransportType*) tanımlandığı şekilde sunucunun adıdır. Tam olarak nitelenmiş bir ağ adı olmalıdır; örneğin, AMACHINE.ACOMPANY.COM(1414).
- *ConnectionName* , bağlantı adlarının virgülle ayrılmış bir listesi olabilir. Listedeki bağlantı adları, istemci bağlantı çizelgesindeki birden çok bağlantıya benzer şekilde kullanılır. Bağlantı adı listesi, istemcinin denemesi için birden çok bağlantı belirtmek üzere kuyruk yöneticisi gruplarına alternatif olarak kullanılabilir. Çok eşgözümlü bir kuyruk yöneticisi yapılandırıyorsanız, farklı kuyruk yöneticisi yönetim ortamları belirtmek için bir bağlantı adı listesi kullanabilirsiniz.
- **MQSERVER** işlemini iptal etmek ve **MQCHLLIB** ve **MQCHLTAB** ile gösterilen istemci kanal tanımlama çizelgesine dönmek için aşağıdaki komutu girin:

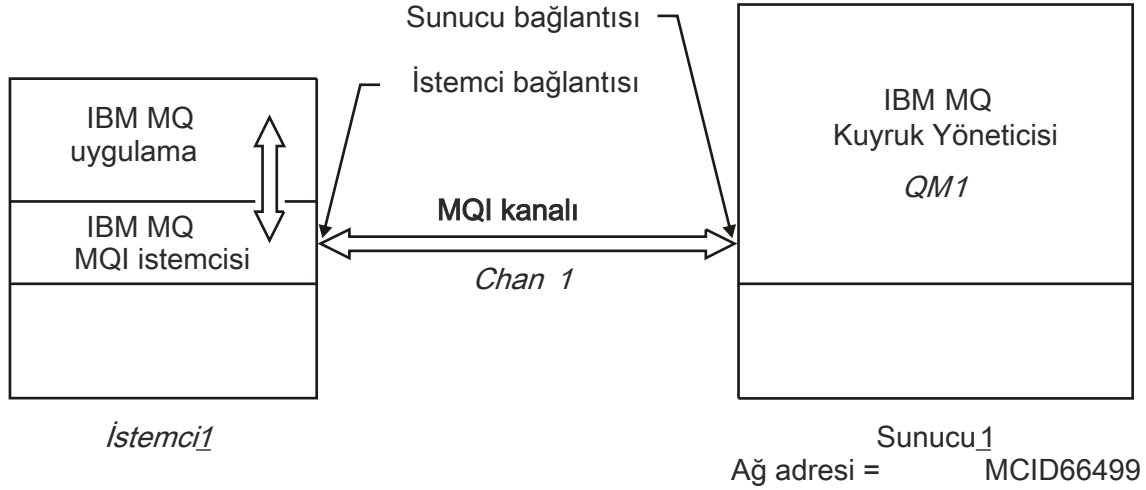
- **Linux** **AIX** AIX and Linux'ta:

```
unset MQSERVER
```

- **Windows** Windows'ta:

```
SET MQSERVER=
```

## Örnek



Şekil 1. Basit bir kanal tanımı örneği

Şekil 1 sayfa 38’inde gösterilen basit kanal tanımını yaratmak için aşağıdaki komutları kullanın:

- **Linux** **AIX** AIX and Linux'ta:

```
export MQSERVER=CHANNEL1/TCP/'MCID66499'
```

- **Windows** Windows'ta:

```
SET MQSERVER=CHANNEL1/TCP/MCID66499
```

**Not:** TCP/IP kapı numarasının değiştirilmesine ilişkin bilgi için bkz. [“TCP/IP varsayılan kapısının değiştirilmesi”](#) sayfa 39.

Basit kanal tanımlarına ilişkin diğer bazı örnekler şunlardır:

- **Windows** Windows'ta:

```
SET MQSERVER=CHANNEL1/TCP/9.20.4.56  
SET MQSERVER=CHANNEL1/NETBIOS/BOX643
```

- **Linux** **AIX** AIX and Linux'ta:

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56'  
export MQSERVER=CHANNEL1/LU62/BOX99
```

Burada BOX99 , LU 6.2 ConnectionName' dir.

- **IBM i** IBM i'ta:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('CHANNEL1/TCP/9.20.4.56(1416)')
```

IBM MQ MQI client’lerinde, tüm **MQCONN** ya da **MQCONNX** istekleri, kanal **MQCONNX**' e sağlanan MQCNO yapısından gönderme yapılan bir MQCD yapısında geçersiz kılınmadıkça, tanımladığınız kanalı kullanmayı dener.

### İlgili görevler

[“IBM MQ ortam değişkenlerinin kullanılması”](#) sayfa 62

Geçerli ayarları görüntülemek ya da IBM MQ ortam değişkenlerinin değerlerini sıfırlamak için komutları kullanabilirsiniz.

“MQCNO kullanarak IBM MQ MQI client üzerinde istemci bağlantısı kanalı yaratılması” sayfa 40  
Bir MQCONNX çağrısında MQCNO yapısını kullanarak istemci iş istasyonunda bir istemci-bağlantı kanalı tanımlayabilirsiniz.

### **TCP/IP varsayılan kapısının değiştirilmesi**

Varsayılan olarak, TCP/IP için IBM MQ , kanalın 1414 numaralı kapıya bağlanacağını varsayar. Gerekirse, varsayılan değeri değiştirebilirsiniz.

### **Bu görev hakkında**

Aşağıdaki üç seçenektan birini kullanarak kapı numarasını değiştirebilirsiniz:


- **MQSERVER** ortam değişkenini kullanarak.
- `mqclient.ini` dosyasını değiştirerek.
- Services kütüğüne IBM MQ eklenerek.

### **Yordam**

- **MQSERVER** ortam değişkenini kullanarak kapı numarasını değiştirmek için, *ConnectionName* ürünün son parçası olarak kapı numarasını köşeli ayraç içine ekleyin; örneğin:

–  AIX and Linux'ta:

```
export MQSERVER='ChannelName/TransportType/ConnectionName(PortNumber)'
```

–  Windows'ta:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(PortNumber)
```

- Kapı numarasını `mq.ini` dosyasını kullanarak değiştirmek için, protokol adına kapı numarasını ekleyin; örneğin:

```
TCP:  
port=2001
```

- IBM MQ dosyasını Services dosyasına ekleyerek kapı numarasını değiştirmek için “AIX and Linux üzerinde TCP/IP dinleyicisinin kullanılması” sayfa 258’inde açıklanan adımları tamamlayın.

### **SPX varsayılan yuvasının değiştirilmesi**

Varsayılan olarak SPX için IBM MQ , kanalın 5E86 yuvasına bağlanacağını varsayar. Gerekirse, varsayılan değeri değiştirebilirsiniz.

### **Bu görev hakkında**

Aşağıdaki seçeneklerden birini kullanarak kapı numarasını değiştirebilirsiniz:

- **MQSERVER** ortam değişkenini kullanarak.

SPX bağlantıları için, *ConnectionName* ve `network.node(socket)` biçiminde yuva belirtin. IBM MQ istemcisi ve sunucusu aynı ağdaysa, ağ belirtilmesi gerekmez. Varsayılan yuvayı kullanıyorsanız, yuvanın belirtilmesi gerekmez.

- `mqclient.ini` file.Changing `qm.ini` dosyasının değiştirilmesi.

## Yordam

- **MQSERVER** ortam deęişkenini kullanarak bir SPX baęlantısına iliřkin kapı numarasını deęiřtirmek için, ařaęıdaki örnekte gösterildięi gibi *ConnectionName* ve *network . node (socket)* formundaki yuvayı belirtin:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(SocketNumber)
```

**Not:** IBM MQ istemcisi ve sunucusu aynı aędaysa, aęı belirtmeniz gerekmez. Varsayılan yuvayı kullanıyorsanız, yuvayı belirtmenize gerek yoktur.

- Kapı numarasını *qm . ini* dosyasını kullanarak deęiřtirmek için, protokol adına kapı numarasını ekleyin; örneęin:

```
SPX:  
socket=5E87
```

## MQCNO kullanarak IBM MQ MQI client üzerinde istemci baęlantısı kanalı yaratılması

Bir MQCONNX çağrısında MQCNO yapısını kullanarak istemci iř istasyonunda bir istemci-baęlantı kanalı tanımlayabilirsiniz.

### Bu görev hakkında

Bir IBM MQ MQI client uygulaması, bir istemci-baęlantı kanalının tanımlamasını içeren bir kanal tanımlaması yapısına (MQCD) gönderme yapmak için **MQCONNX** çağrısında baęlanma seçenekleri yapısını (MQCNO) kullanabilir.

Bu şekilde, istemci uygulaması yürütme sırasında bir kanalın **ChannelName**, **TransportType** ve **ConnectionName** özniteliklerini belirleyerek istemci uygulamasının aynı anda birden çok sunucu kuyruk yöneticisine baęlanmasını saęlayabilir.

**MQSERVER** ortam deęişkenini kullanarak bir kanal tanımlarsanız, yürütme sırasında **ChannelName**, **TransportType** ve **ConnectionName** özniteliklerini belirtmenin mümkün olmadığını unutmayın.

İstemci uygulaması, **MaxMsgLength** ve **SecurityExit** gibi bir kanalın özniteliklerini de belirtir. Bu özniteliklerin belirtilmesi, istemci uygulamasının varsayılan deęerler olmayan öznitelikler için deęer belirtmesini saęlar ve kanal çıkıř programlarının bir MQI kanalının istemci ucunda çağrılmasını saęlar.

Bir kanal TLS (Transport Layer Security; İletim Katmanı Güvenlięi) kullanıyorsa, bir istemci uygulaması MQCD yapısında TLS ile ilgili bilgileri de saęlayabilir. TLS ile ilgili ek bilgiler, **MQCONNX** çağrısında MQCNO yapısı tarafından da bařvurulan TLS yapılandırma seçenekleri yapısı MQSCO ' da saęlanabilir.

MQCNO, MQCD ve MQSCO yapılarıyla ilgili daha fazla bilgi için bkz. [MQCNO](#), [MQCD](#) ve [MQSCO](#).

**Not:** MQCONNX için örnek programa **amqscnxc** adı verilir. **amqssslc** adlı başka bir örnek program, MQSCO yapısının kullanımını gösterir.

### İlgili görevler

[“MQSERVER kullanılarak IBM MQ MQI client üzerinde istemci-baęlantı kanalı yaratılması” sayfa 36](#)





**MQSERVER** ortam deęişkenini kullanarak istemci iř istasyonunda bir istemci baęlantısı kanalı tanımlayabilirsiniz.

## Sunucuda sunucu baęlantısı ve istemci baęlantısı tanımlamaları yaratılması

Sunucuda her iki tanımlı da yaratabilir ve istemci baęlantısı tanımlamasını istemcinin kullanımına açabilirsiniz.

### Bu görev hakkında

Önce bir sunucu baęlantısı kanalı tanımlarsınız ve sonra bir istemci baęlantısı kanalı tanımlarsınız:

- Tüm altyapılarda, sunucu makinesinde bir sunucu bağlantısı kanalı tanımlamak için IBM MQ Script (MQSC) komutlarını, programlanabilir komut biçimi (PCF) komutlarını kullanabilirsiniz.
-   Linux ve Windows sistemlerinde IBM MQ Explorer kullanılabilir.
-  z/OS işletim ve Denetim panolarını da kullanabilirsiniz.
-  IBM i üzerinde pano arabirimini de kullanabilirsiniz.

Sunucuda yaratılan istemci bağlantısı kanal tanımlamaları, istemci kanal tanımlama çizelgesi (CCDT) kullanılarak istemcilerin kullanımına sunulur.

## Yordam

1. Bir sunucu bağlantısı kanalı tanımlamak için bkz. [“Sunucuda sunucu bağlantısı kanalının tanımlanması” sayfa 54.](#)
2. Bir istemci-bağlantı kanalı tanımlamak için bkz. [“Sunucuda istemci-bağlantı kanalının tanımlanması” sayfa 54.](#)

## İlgili görevler

[“İkili biçim CCDT ' nin yapılandırılması” sayfa 42](#)

İstemci kanal tanımlama çizelgesi (CCDT), kuyruk yöneticisine bağlanmak için istemci uygulamaları tarafından kullanılan kanal tanımlarını ve kimlik doğrulama bilgilerini belirler. Multiplatforms üzerinde, kuyruk yöneticisi yaratıldığında varsayılan ayarları içeren ikili bir CCDT otomatik olarak yaratılır. İkili CCDT ' yi güncellemek için **runmqsc** komutunu kullanın.

[“Sunucuda sunucu bağlantısı kanalının tanımlanması” sayfa 54](#)

Kuyruk yöneticisi için bir sunucu bağlantısı kanal tanımlaması yaratın.

[“Sunucuda istemci-bağlantı kanalının tanımlanması” sayfa 54](#)

Sunucu bağlantısı kanalını tanımladıktan sonra, ilgili istemci-bağlantı kanalını tanımlayabilirsiniz.

[“İstemci bağlantısı kanal tanımlarına erişilmesi” sayfa 56](#)

İstemci kanal tanımlama çizelgesini (CCDT) kopyalayarak ya da paylaşarak istemci uygulamalarının kullanımına sunabilir, daha sonra istemci bilgisayardaki yerini ve adını belirtebilirsiniz. Bir istemci kanal tanımlama çizelgesini (CCDT) URL aracılığıyla da bulabilirsiniz.

## İstemci kanal tanımlama çizelgelerinin yapılandırılması

İstemci kanal tanımlama çizelgesi (CCDT), istemci bağlantı kanallarını ve özniteliklerini tanımlar. İstemciler, hangi kuyruk yöneticilerine bağlanacağını saptamak için bu dosyayı okur. CCDT dosyası JSON ya da ikili biçimde olabilir.


## Bu görev hakkında

Kuyruk yöneticisi CCDT dosyasını okumuyor. Yalnızca istemcilere kanal tanımları ve kimlik doğrulama bilgileri sağlamak için kullanılır.

IBM MQ 9.2.0 öncesinde, CCDT yalnızca ikili biçimde kullanılabilir. IBM MQ 9.2.0' dan JavaScript Object Notation (JSON) biçiminde bir CCDT de oluşturabilirsiniz.

Bir kuyruk yöneticisi yaratıldığında otomatik olarak ikili biçimli bir CCDT yaratılır. Bu çizelgede saklanan istemci kanalı tanımlamalarını yalnızca **runmqsc** komutunu kullanarak güncelleyebilirsiniz.

JSON biçimi CCDT, .json uzantılı düz metin dosyasıdır. **runmqsc** komutunu kullanmaktan daha az kısıtlayıcı olan bu çizelgeyi el ile oluşturur ve güncelleyebilirsiniz.

 Bir uygulama sunucusunda çalışan z/OS JMS istemcileri, uzak kuyruk yöneticisi bağlantı ayrıntılarına gönderme yapmak için bir CCDT kullanır. IBM MQ for z/OS 9.1'den IBM MQ Advanced for z/OS Value Unit Edition , JMS istemcilerinin diğer z/OS LPAR' larında kuyruk yöneticilerine uzaktan bağlanmasına izin verir. Bu nedenle, bu istemciler CCDT ' leri de kullanabilir.

CCDT ' leri istemcilerinizle çalışacak şekilde yapılandırmanıza yardımcı olmak için aşağıdaki görevlerden birini seçin:

## Yordam

- [“İkili biçim CCDT ' nin yapılandırılması” sayfa 42](#)
- [“JSON biçimi CCDT ' nin yapılandırılması” sayfa 44](#)
- [“CCDT ' ye ilişkin konular” sayfa 51](#)
- [“CCDT ' ye URL erişimi” sayfa 52](#)

## İlgili kavramlar

MQI istemcisi: [İstemci Kanal Tanımlama Çizelgesi \(CCDT\)](#)

## İlgili görevler

[“Birörnek kümenin yapılandırılması” sayfa 400](#)

Tek tip kümeler, uygulamaların ölçek ve kullanılabilirlik için tasarlanmasını sağlar ve bu tek tip küme içindeki herhangi bir kuyruk yöneticisine bağlanabilir.

## İkili biçim CCDT ' nin yapılandırılması

İstemci kanal tanımlama çizelgesi (CCDT), kuyruk yöneticisine bağlanmak için istemci uygulamaları tarafından kullanılan kanal tanımlarını ve kimlik doğrulama bilgilerini belirler. Multiplatforms üzerinde, kuyruk yöneticisi yaratıldığında varsayılan ayarları içeren ikili bir CCDT otomatik olarak yaratılır. İkili CCDT ' yi güncellemek için **runmqsc** komutunu kullanın.

## Başlamadan önce

IBM MQ 9.1.2' den, JavaScript Object Notation (JSON) biçiminde bir CCDT de yaratabilirsiniz ve bu alternatif biçimin kullanılması, ikili CCDT kullanımına göre bazı avantajlar sağlar. Bkz. [“JSON biçimi CCDT ' nin yapılandırılması” sayfa 44](#).

Tüm platformlardaki istemciler CCDT ' leri görüntüleyebilir ve kullanabilirler. Ancak ikili CCDT yalnızca IBM MQ for Multiplatforms'ta yaratılabilir ve değiştirilebilir.

## Bu görev hakkında

**Multi** Çoklu platformlar'ta:

- Kuyruk yöneticisine ilişkin veri dizini altındaki @ipcc dizininde otomatik olarak ikili bir CCDT yaratılır.
- Otomatik olarak yaratılmanın yanı sıra, bir kuyruk yöneticisiyle ilişkilendirilmiş ikili CCDT, nesne tanımlamalarıyla uyumlu olarak tutulur. Bir istemci kanal nesnesi tanımladığınızda, değiştirdiğinizde ya da sildiğinizde, hem kuyruk yöneticisi nesnesi tanımlaması, hem de CCDT ' deki giriş aynı işlemin bir parçası olarak güncellenir.

### Notlar:

- IBM MQ CCDT dosyasının tasarımı, CCDT dosyasının, yalnızca kullanıcı tarafından tanımlanan tüm istemci bağlantı kanalları gerçekten tanımlandıktan sonra küçültülmesinden ibaretti. Bir istemci-bağlantı kanalı silindiğinde, CCDT dosyasında silinmiş olarak işaretlenir, ancak fiziksel olarak kaldırılmaz.
- CCDT dosyasını daraltmaya zorlamak için, bir ya da daha çok istemci-bağlantı kanalını sildikten sonra aşağıdaki komutu verin:

```
rcrmqobj -m QM80 -t clchltab
```

- İkili CCDT ' nin yerini ve içeriğini değiştirmek için **runmqsc** komutunu kullanın.

Tüm platformlardaki istemciler ikili bir CCDT görüntüleyebilir ve kullanabilirler.

## Yordam

### • Multi

Varsayılan bir ikili CCDT yaratır.

Çoklu platformları işletim sistemi üzerinde, bir kuyruk yöneticisi yarattığınızda AMQCLCHL . TAB adlı varsayılan bir ikili CCDT yaratılır.

Varsayılan olarak, AMQCLCHL.TAB , bir sunucuda aşağıdaki dizinde bulunur:

– **IBM i** IBM işletim sisteminde, tümleşik dosya sisteminde:

```
/QIBM/UserData/mqm/qmgrs/QUEUEMANAGERNAME/&ipcc
```

– **Linux** **AIX** AIX and Linux sistemlerinde:

```
/prefix/qmgrs/QUEUEMANAGERNAME/@ipcc
```

*QUEUEMANAGERNAME* tarafından başvuru dizinin adı, AIX and Linux sistemlerinde büyük ve küçük harfe duyarlıdır. Kuyruk yöneticisi adında özel karakterler varsa, dizin adı kuyruk yöneticisi adıyla aynı olmayabilir.

– **Windows** Windows'ta:

```
MQ_INSTALLATION_PATH\data\qmgrs\QUEUEMANAGERNAME\@ipcc
```

Burada *MQ\_INSTALLATION\_PATH* , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

Ancak, kuyruk yöneticisi verileri için farklı bir dizin kullanmayı seçmiş olabilirsiniz. **crtmqm** komutunu kullandığınızda **-md DataPath** parametresini belirtebilirsiniz. Bunu yaparsanız, AMQCLCHL . TAB , belirttiğiniz *DataPath* ürününün @ipcc dizininde bulunur.

### • CCDT ' yi bulun:

- İstemci bilgisayarda
- Birden çok istemci tarafından paylaşılan bir konumda
- Sunucuda paylaşılan dosya olarak

Bkz. “CCDT ' ye ilişkin konumlar” sayfa 51.

a) Bir istemci makinesinde doğrudan ikili CCDT yaratır.

- **runmqsc** komutunu **-n** parametresiyle kullanın.
- CCDT, **MQCHLLIB** ile gösterilen konumda ve varsayılan olarak AMQCLCHL . TAB olan **MQCHLTAB** ile gösterilen dosya adıyla oluşturulur.
- **Önemli:** **-n** parametresini belirtirseniz, başka bir parametre belirtmemelisiniz.

b) Konumu değiştirin.

**MQCHLLIB** ayarını tanımlayarak CCDT yolunu değiştirebilirsiniz. Aynı sunucuda birden çok kuyruk yöneticiniz varsa, bu yöneticilerin aynı CCDT konumunu paylaştığını unutmayın.

### • CCDT ' ye Erişin

CCDT ' ye erişebilirsiniz:

- **MQCCDTURL** ortam değişkenini tanımlayarak bir dosyadan, ftp ya da http URL adresinden uzaktan.
- **MQCHLLIB** ve **MQCHLTAB** ortam değişkenlerini yerel olarak ayarlayarak.
- Yerel olarak, istemci yapılandırma dosyasında KANALLAR kısmına ilişkin **ChannelDefinitionDirectory** ve **ChannelDefinitionFile** özniteliklerini tanımlayarak.

Çeşitli örnekler için bkz. “CCDT ' ye ilişkin konumlar” sayfa 51 .

### • CCDT içeriğini görüntüleyin ya da düzenleyin.

CCDT içeriğini **runmqsc** komutuyla görüntüleyebilirsiniz:

1. Ortam değişkenlerini CCDT ' ye Erişim olarak ayarlayın
2. Komutu çalıştırır `runmqsc -n`
3. `DISPLAY CHANNEL (*)` komutunu çalıştırın; örneğin,

**Multi** Çoklu platformları işletim sistemi üzerinde, ikili CCDT içeriğini `runmqsc` komutunu kullanarak da düzenleyebilirsiniz. CCDT ' nin her girişi, belirli bir kuyruk yöneticisine istemci bağlantısını gösterir. **DEFINE CHANNEL** komutunu kullanarak bir istemci-bağlantı kanalı tanımladığınızda yeni bir giriş eklenir ve **ALTER CHANNEL** komutunu kullanarak istemci-bağlantı kanallarını değiştirdiğinizde giriş güncellenir. Komutu kullanmaya ilişkin daha fazla örnek için bkz. [runmqsc](#) .

- TLS sertifika iptali olup olmadığını denetlemek için istemcilere kimlik doğrulama bilgileri sağlayın.
  - a) Kimlik doğrulama bilgileri nesnelere içeren bir ad listesi tanımlayın.
  - b) **SSLCRNL** kuyruk yöneticisi özniteliğini ad listesi adına ayarlayın.

### İlgili kavramlar

İptal edilen sertifikalarla çalışma

### İlgili görevler

“JSON biçimi CCDT ' nin yapılandırılması” sayfa 44

İstemci kanal tanımlama çizelgesi (CCDT), kuyruk yöneticisine bağlanmak için istemci uygulamaları tarafından kullanılan kanal tanımlarını ve kimlik doğrulama bilgilerini belirler. Bir JavaScript Nesne Gösterimi (JSON) yaratmak ve güncellemek için bir metin düzenleyicisi kullanırsınız CCDT.

### **JSON biçimi CCDT ' nin yapılandırılması**

İstemci kanal tanımlama çizelgesi (CCDT), kuyruk yöneticisine bağlanmak için istemci uygulamaları tarafından kullanılan kanal tanımlarını ve kimlik doğrulama bilgilerini belirler. Bir JavaScript Nesne Gösterimi (JSON) yaratmak ve güncellemek için bir metin düzenleyicisi kullanırsınız CCDT.

### Başlamadan önce

**Multi** IBM MQ for Multiplatformskullanıyorsanız, bunun yerine bir kuyruk yöneticisi yarattığınızda otomatik olarak yaratılan ikili CCDT ' yi kullanabilirsiniz. Bkz. “İkili biçim CCDT ' nin yapılandırılması” sayfa 42.

### Bu görev hakkında

JSON biçimi için CCDT şemasının dosya adı:

#### Linux

`/opt/mqm/lib/ccdt_schema.json`

#### Windows

`C:\Program Files\IBM\MQ\bin\ccdt_schema.json`

Varsayılan JSON CCDT ' si yoktur ve IBM MQ , JSON biçiminde CCDT ' leri oluşturmak ya da düzenlemek için herhangi bir araç sağlamaz. Bununla birlikte, bir JSON CCDT ' yi el ile geliştirdiğinizde, ikili CCDT ile çalışmak için `runmqsc` komutunu kullanmaktan daha fazla yapılandırma seçeneğiniz vardır:


- JSON CCDT dosyası yaratmak ve düzenlemek için IBM MQ for Multiplatforms kullanmanıza gerek yoktur.
- JSON biçimini kullanarak, aynı adı taşıyan yinelenen kanal tanımlamaları tanımlayabilirsiniz. IBM MQ uygulamasını bulutta devreye aldığınızda, devreye alımınızı ölçeklenebilir ve yüksek düzeyde kullanılabilir hale getirmek için bunu kullanabilirsiniz.
- JSON dosyası, kuyruk yöneticisi yapılandırmasını basitleştirebilen, insan tarafından okunabilir bir dosyadır.
- Düz dosya biçimi şunlarla bütünleştirilebilir:
  - CCDT geçmişini izlemek için sürüm denetimi araçları
  - Sürekli teslimde otomasyon araçları



- CCDT dosyasını korumak için uzman araçlara gerek yoktur.
- Dosya daha küçük.
- Bu biçim, geriye ve ileriye dönük uyumluluk sağlar.

#### Notlar:

1. JSON standardı yinelenen anahtarları geçerli olarak görür, ancak JSON ayrıştırıcısı öznitelikler atanırken yalnızca yinelenen anahtarların son okuma değerini alır. Bu nedenle, yinelenen kanallar tanımlanırken, her kanal 'channel' anahtarına atanan bir dizi değerinin ögesi olmalıdır.
2. JSON CCDT 'leri, CRL (Certificate Revocation Lists; Sertifika İptal Listeleri) ve OCSP (Online Certificate Status Protocol; Çevrimiçi Sertifika Durumu İletişim Kuralı) yanıt veren konumu bilgileri için LDAP (Lightweight Directory Access Protocol; Temel Dizin Erişimi İletişim Kuralı) sunucu konumlarının depolanmasını desteklemez.

Çizelge 7. Platforma göre kodlama gereksinimleri		
Hizmet olarak sunulan	JMS istemci kodlaması	C istemcisi kodlaması
 IBM i	ASCII	EBCDIC
 AIX, Linux, and Windows	ASCII	ASCII
 z/OS	ASCII ya da EBCDIC	Geçerli değildir



**Uyarı:** JSON CCDT aracılığıyla bir kanal için herhangi bir tanımlama sağladığınızda (tüm öznitelikleri içermeyen seyrek bir kanal tanımlaması da içinde olmak üzere), JSON 'da belirtilmeyen her şey için varsayılan değerler kullanılarak, tanımlanan tüm özniteliklerle tam bir kanal tanımlaması oluşturulur.

Bu nedenle, varsayılan değerini istemediğiniz her öznitelik için belirli değerler belirtmeniz gerekir.

#### Yordam

- JSON CCDT Yarat
  - a) Sosyal metin düzenleyicisiyle .json uzantılı düz bir dosya oluşturun.
  - b) Bir CCDT tanımlayın.
 



Bkz. “JSON CCDT örnekleri” sayfa 48 ve “JSON CCDT tarafından desteklenen kanal öznitelikleri” sayfa 46.
- CCDT 'yi bulun:
  - İstemci bilgisayarda
  - Birden çok istemci tarafından paylaşılan bir konumda
  - Sunucuda paylaşılan dosya olarak


Bkz. “CCDT 'ye ilişkin konular” sayfa 51.
- JSON CCDT 'nin geçerliliğini denetle
 

CCDT 'yi JSON linter ile şemayla karşılaştırarak doğrulayın.

İki kanalla bir CCDT dosyasının nasıl yaratılacağına ve çalışıp çalışmadığına ilişkin bilgi için [IBM MQ CCDT JSON dosyasının şemayla karşılaştırılarak doğrulanması](#) başlıklı konuya bakın.

Ürün ve istemci paketlerinde CCDT şeması bulunur:

  -   AIX and Linux sistemlerinde:
 

Sırasıyla ürün ve istemci paketlerinde \$MQ\_INSTALLATION\_PATH/lib ve /lib .
  -  Windows'ta:

Sırasıyla ürün ve istemci paketlerinde %MQ\_INSTALLATION\_PATH%\bin ve \bin .

#### Notlar:

- JSON göstergeleri çevrimiçi olarak kullanılabilir.
  - Şema, zorunlu öznitelikleri 'required' anahtarıyla tanımlar.
  - Şema, 'type' anahtarıyla öznitelik veri tiplerini tanımlar.
  - CCDT ' ye Erişin  
CCDT ' ye erişebilirsiniz:
    - **MQCCDTURL** ortam değişkenini tanımlayarak bir dosyadan, ftp ya da http URLadresinden uzaktan.
    - **MQCHLLIB** ve **MQCHLTAB** ortam değişkenlerini yerel olarak ayarlayarak.
    - Yerel olarak, istemci yapılandırma dosyasında KANALLAR kısmına ilişkin **ChannelDefinitionDirectory** ve **ChannelDefinitionFile** özniteliklerini tanımlayarak.
- Çeşitli örnekler için bkz. “CCDT ' ye ilişkin konumlar” sayfa 51 .
- CCDT içeriğinin görüntülenmesi ya da düzenlenmesi  
CCDT ' nin her girişi, belirli bir kuyruk yöneticisine istemci bağlantısını gösterir. CCDT içeriğini bir metin düzenleyicisiyle görüntüleyebilir ya da düzenleyebilirsiniz.  
Yalnızca CCDT ' yi görüntülemek istiyorsanız, bunu **runmqsc** komutunu aşağıdaki gibi kullanarak da yapabilirsiniz:
    1. Önceki adımda açıklandığı gibi CCDT ' ye erişmenizi sağlamak için ortam değişkenlerini ayarlayın.
    2. **runmqsc -n** komutunu çalıştırın. Daha fazla bilgi için bkz. [runmqsc](#).
    3. **DISPLAY CHANNEL** komutunu çalıştırın. Örneğin, **DISPLAY CHANNEL (\*)** komutunu çalıştırın.

#### İlgili kavramlar

[İptal edilen sertifikalarla çalışma](#)

#### İlgili görevler

[“İkili biçim CCDT ' nin yapılandırılması” sayfa 42](#)

İstemci kanal tanımlama çizelgesi (CCDT), kuyruk yöneticisine bağlanmak için istemci uygulamaları tarafından kullanılan kanal tanımlarını ve kimlik doğrulama bilgilerini belirler. Multiplatforms üzerinde, kuyruk yöneticisi yaratıldığında varsayılan ayarları içeren ikili bir CCDT otomatik olarak yaratılır. İkili CCDT ' yi güncellemek için **runmqsc** komutunu kullanın.

[“Birörnek kümenin yapılandırılması” sayfa 400](#)

Tek tip kümeler, uygulamaların ölçek ve kullanılabilirlik için tasarlanmasını sağlar ve bu tek tip küme içindeki herhangi bir kuyruk yöneticisine bağlanabilir.

*JSON CCDT tarafından desteklenen kanal öznitelikleri*

JSON CCDT tarafından desteklenen istemci bağlantı kanalı özniteliklerinin listesi. Bu liste, ikili CCDT tarafından desteklenen özniteliklerin bir alt kümesidir.

## Öznitelik eşlemesi

Bu öznitelikler aşağıdaki kanal nesnesine eklenir:

```
{ "channel": [ { $CHANNEL_1_KEY_VALUE_LIST }, ..., { $CHANNEL_N_KEY_VALUE_LIST } ] }
```

Burada \$CHANNEL\_X\_KEY\_VALUE\_LIST , aşağıdaki çizelgede listelenen özniteliklerin virgülle ayrılmış listesidir.

Temel kullanım senaryoları için bkz. “JSON CCDT örnekleri” sayfa 48 .

JSON şeması /opt/mqm/lib/ccdt\_schema.json içinde gönderilir. Özniteliklerin her biri için geçerli olan değerleri keşfetmek için JSON şemasına bakın.

Aşağıdaki çizelge, ilgili ikili kanal özniteliği tanımlamasıyla birlikte JSON nesnesini, anahtarını ve veri tipini listeler.



**Uyarı:** Gerekli öznitelikler, kanal **name** ve kanal **type**' dir. **portRange**değerini de tanımlarsanız, **düşük** ve **yüksek** öznitelikler de gereklidir.

JSON nesnesi	JSON anahtarı	JSON veri tipi	İkili öznitelik tanımlaması
kanal (dizi)	Ad	STRING	Kanal
kanal (dizi)	tip	STRING	CHLTYPE
channel.clientConnection	queueManager	STRING	QMNAME
channel.clientConnection.connection (dizi)	anasistem	STRING	KONADı
channel.clientConnection.connection	kapı	TAMSAYI	KONADı
channel.compression.header (dizi)	üstbilgi	STRING	TAMAMLANDI
channel.compression.message (dizi)	ileti	STRING	ŞİRKET
channel.connectionManagement	benzerlik	STRING	SONRAKALIK
channel.connectionManagement	clientWeight	TAMSAYI	KLNTWGHT
channel.connectionManagement	defaultReconnect	STRING	DEFRECON
channel.connectionManagement	disconnectInterval	TAMSAYI	İNDIRIM
channel.connectionManagement	heartInterval	TAMSAYI	HBINT (HBINT
channel.connectionManagement	keepAliveAralığı	TAMSAYI	KAINT
channel.connectionManagement	sharingConversations	TAMSAYI	SHARECNV
channel.connectionManagement.localAddress (dizi)	anasistem	STRING	LOCLADDR
channel.connectionManagement.localAddress (dizi)	kapı	TAMSAYI	LOCLADDR
channel.connectionManagement.localAddress.portRange	yüksek	TAMSAYI	LOCLADDR
channel.connectionManagement.localAddress.portRange	düşük	TAMSAYI	LOCLADDR
channel.exits.receive (dizi)	Ad	STRING	RCVEXIT
channel.exits.receive (dizi)	userData	STRING	RCVDATA
channel.exits.security	Ad	STRING	SCYEXIT
channel.exits.security	userData	STRING	SCYDATA
channel.exits.send (dizi)	Ad	STRING	SENDEXIT
channel.exits.send (dizi)	userData	STRING	SENDDATA
channel.general	açıklama	STRING	TANIMLAMA
channel.general	maximumMessageUzunluğu	TAMSAYI	MAXMSGSL
channel.timestamps	değiştirildi	STRING	ALTDATe ve ALTTIME
channel.transmissionSecurity	certificateLabel	STRING	CERTLABL
channel.transmissionSecurity	certificatePeerAdı	STRING	SSLPEER

JSON nesnesi	JSON anahtarı	JSON veri tipi	İkili öznitelik tanımlaması
channel.transmissionSecurity	cipherSpecification	STRING	SSLCIPH

#### Notlar:

- channel.connectionManagement.localAddress , aşağıdaki tuş birleşimlerinden biri olarak tanımlanabilir:
  - Anasistem ve kapı
  - anasistem ve portRange
  - kapı
  - portRange
- channel.timestamps altered JSON anahtarı isteğe bağlıdır ve tanımlı değilse, değer varsayılan olarak JSON CCDT dosyasının son değiştirme zamanına ayarlanır. Ancak, ortam CCDT 'yi bir URL' den getirecek şekilde yapılandırıldıysa, varsayılan değer dosyanın en son karşıdan yüklendiği zamandır.
- channel.clientConnection.connection , hem anasistem hem de kapı tuşlarını içermelidir.
- Değiştirilen anahtar, ALTDATE ve ALTIME özniteliklerini içeren tek bir dizedir.
- İletim tipi yalnızca TCP olabilir; bu nedenle, şemada şu öznitelikler tanımlı değil:
  - **TRPTYPE**
  - **USERID**
  - **PASSWORD**
  - **MODENAME**
  - **TPNAME**

#### İlgili başvurular

Kanal tipleri için kanal öznitelikleri

#### JSON CCDT örnekleri

Gereksinimlerinize temel olarak bu konuda listelenen örnekleri kullanın.

Soysal bir metin düzenleyicisi açın ve aşağıdaki örneklerden birini kopyalayın:

- [“Basit bir istemci bağlantısı tanımlayın” sayfa 48](#)
- [“TLS kullanarak bir kanal ve bir kuyruk yöneticisi tanımlayın” sayfa 49](#)
- [“TLS kullanmayan bir kanal ve bir kuyruk yöneticisi tanımlayın” sayfa 49](#)
- [“Aynı adla iki kanal tanımla” sayfa 49](#)
- [“Bir istemci bağlantı kanalına ilişkin CCDT kanal özniteliği tanımlarının tam listesi” sayfa 50](#)

#### Basit bir istemci bağlantısı tanımlayın

```
{
  "channel":
  [
    {
      "general":
      {
        "description": "a channel"
      },
      "name": "channel",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "localhost",
            "port": 1414
          }
        ]
      }
    }
  ]
}
```

```

    }
  ],
  "queueManager": "QM1"
},
"type": "clientConnection"
}
]
}

```

## TLS kullanarak bir kanal ve bir kuyruk yöneticisi tanımlayın

```

{
  "channel": [
    {
      "name": "SSL.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "aztlan1.fyre.ibm.com",
            "port": 1419
          }
        ],
        "queueManager": "QM92TLS"
      },
      "transmissionSecurity": {
        "cipherSpecification": "TLS_AES_128_GCM_SHA256",
        "certificateLabel": "ibmwebspheremqadministrator"
      },
      "type": "clientConnection"
    }
  ]
}

```

## TLS kullanmayan bir kanal ve bir kuyruk yöneticisi tanımlayın

```

{
  "channel": [
    {
      "name": "SYSTEM.DEF.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "aztlan1.fyre.ibm.com",
            "port": 1414
          }
        ],
        "queueManager": "QM92"
      },
      "type": "clientConnection"
    }
  ]
}

```

## Aynı adla iki kanal tanımla

Her kanal iki ayrı kuyruk yöneticisine bağlanır:

```

{
  "channel": [
    {
      "general": {
        "description": "First channel"
      },
      "name": "channel",
      "clientConnection": {
        "connection": [
          {
            "host": "localhost",

```

```

        "port": 1414
      }
    ],
    "queueManager": "QM1"
  },
  "type": "clientConnection"
},
{
  "general":
  {
    "description": "Second channel"
  },
  "name": "channel",
  "clientConnection":
  {
    "connection":
    [
      {
        "host": "localhost",
        "port": 1415
      }
    ],
    "queueManager": "QM2"
  },
  "type": "clientConnection"
}
]
}

```

## Bir istemci bağlantı kanalına ilişkin CCDT kanal özneteliği tanımlarının tam listesi

```

{
  "channel":
  [
    {
      "compression":
      {
        "header": [ "system" ],
        "message": [ "zlibfast" ]
      },
      "connectionManagement":
      {
        "sharingConversations": 10,
        "clientWeight": 1,
        "affinity": "none",
        "defaultReconnect": "yes",
        "heartbeatInterval": 600,
        "keepAliveInterval": -1,
        "localAddress":
        [
          {
            "portRange":
            {
              "low": 2020,
              "high": 3030
            }
          }
        ]
      },
      "exits":
      {
        "receive":
        [
          {
            "name": "",
            "userData": ""
          }
        ],
        "security":
        {
          "name": "",
          "userData": ""
        },
        "send":
        [
          {
            "name": "",
            "userData": ""
          }
        ]
      }
    }
  ]
}

```

```

    ]
  },
  "general":
  {
    "description": "First channel",
    "maximumMessageLength": 4194304
  },
  "name": "the_channel",
  "clientConnection":
  {
    "connection":
    [
      {
        "host": "localhost",
        "port": 1414
      }
    ],
    "queueManager": "QM1"
  },
  "timestamps":
  {
    "altered": "2018-12-04T15:37:22.000Z"
  },
  "transmissionSecurity":
  {
    "cipherSpecification": "",
    "certificateLabel": "",
    "certificatePeerName": ""
  },
  "type": "clientConnection"
}
]
}

```

## İlgili başvurular

[Kanal tipleri için kanal öznitelikleri](#)

[Alfabetik sırada kanal öznitelikleri](#)

## CCDT 'ye ilişkin konular

IBM MQ , bir dosyadan, FTP 'den ya da HTTP URL' den CCDT alınmasını destekler. CCDT 'yi sunucuda bulunmaya devam ederken, istemcinin paylaşılan bir dosya olarak erişmesine olanak verebilirsiniz. Diğer bir seçenek olarak, CCDT 'yi tek tek istemci bilgisayarlara kopyalayarak ya da CCDT' yi birden çok istemci tarafından paylaşılan bir yere kopyalayarak CCDT 'yi dağıtabilirsiniz.

Dosyayı kopyalamak için FTP kullanıyorsanız, ikili kipi ayarlamak için bin seçeneğini kullanın; varsayılan ASCII kipini kullanmayın. CCDT 'yi kullanılabilir kılmak için hangi yöntemi seçerseniz seçin, kanallarda yetkisiz değişiklikleri önlemek için konum güvenli olmalıdır.

## CCDT dosyasını sunucuda barındırma

IBM MQ 9.0adresinden CCDT, URL aracılığıyla erişilebilen merkezi bir konumda barındırılabilir ve konuşlandırılan her istemci için CCDT 'yi ayrı ayrı güncelleme gereksinmesi ortadan kaldırılabilir. IBM MQ 9.0 , yerel dosya, FTP ya da HTTP kaynağı olsun, yerel (C/C + +, COBOL ve RPG) ve yönetilmeyen .NET uygulamaları için CCDT 'yi URL' den çekme yeteneğini ekledi.

IBM MQ istemcilerinin varsayılan önbelleğe alma davranışı, bir CCDT dosyasının yalnızca dosya değiştirme zamanı son alındığından farklıysa aşağı çekilmesidir. Çoğu istemci yapılandırma seçeneğinde olduğu gibi, URL konumunun sağlanabileceği çeşitli yollar vardır:

- MQCONNX MQI çağrısına geçirilen MQCNO yapısıyla **CCDTUr1Ptx** ve **CCDTUr1Offset**
- **MQCCDTURL** ortam değişkeni
- **ChannelDefinitionDirectory** özniteliği mqclient . ini ile ilgili kanal (Channel) kısmı

Kimliği doğrulanmış ve doğrulanmamış URL ' ler desteklenir. Bazı örnekler:

```
export MQCCDTURL=ftp://myuser:password@myhost.sample.com//var/mqm/qmgrs/QMGR/@ipcc/AMQCLCHL.TAB
```

```
export MQCCDTURL=http://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc/AMQCLCHL.TAB
```

Bu desteği FTP ya da HTTP ile kullanmak istiyorsanız, yine de CCDT dosyasını bir sunucuda bulundurmanız gerekir; ancak IBM MQ 9.0 adresindeki destek eklenerek, tüm istemci uygulamalarınız, güncellemeleri el ile göndermeden ya da her istemciye bir ağ dosya sistemi bağlamanız gerekmeden kanal tanımlarında yapılan değişiklikleri otomatik olarak alabilir. Daha fazla bilgi için bkz [“CCDT ' ye URL erişimi” sayfa 52.](#)

## İstemci CCDT ' nin yerinin belirlenmesi

Bir istemci sisteminde, CCDT ' nin yerini aşağıdaki şekillerde belirtebilirsiniz:

- Çizelgenin bulunduğu dizini belirtmek için ortam değişkenlerini **MQCHLLIB** ve çizelgenin dosya adını belirtmek için **MQCHLTAB** kullanın.
- İstemci yapılandırma kütüğü kullanılıyor. KANAL kısmında, çizelgenin bulunduğu dizini belirtmek için **ChannelDefinitionDirectory** özniteliğini ve dosya adını belirtmek için **ChannelDefinitionFile** özniteliğini kullanın.
- Daha önce açıklandığı gibi merkezi bir konumda bulunan bir CCDT için URL (dosya, FTP ya da HTTP) sağlayarak.

Konum hem istemci yapılandırma dosyasında hem de ortam değişkenleri kullanılarak belirtilirse, ortam değişkenleri öncelikli olur. İstemci yapılandırma dosyasında standart bir konum belirtmek ve gerektiğinde ortam değişkenlerini kullanarak bu özelliği geçersiz kılmak için bu özelliği kullanabilirsiniz.

CCDT ' nin yerini belirtmek için bir URL kullanırsanız, yerel istemci uygulamasının istemci kanalı tanımlamasını bulması için öncelik sırası [“CCDT ' ye URL erişimi” sayfa 52](#) içinde açıklanmıştır.

## CCDT ' ye URL erişimi

Bir istemci kanal tanımlama çizelgesini (CCDT), URL yoluyla erişilebilen merkezi bir yerde barındıracak ve konuşlandırılan her istemci için CCDT ' yi tek tek güncelleme gereksinmesini kaldırabilirsiniz.

IBM MQ 9.0 adresinden, bir istemci kanal tanımlama çizelgesi URL yoluyla aşağıdaki yollardan biriyle bulunabilir:

- MQCNO kullanarak programlamaya göre
- Ortam değişkenlerini kullanarak



**Uyarı:** URL ' yi yalnızca istemci olarak bağlanan yerel programlar (C, COBOL ya da C+ + uygulamaları) için sağlamak üzere ortam değişkeni seçeneğini kullanabilirsiniz. Ortam değişkenlerinin Java, JMS ya da yönetilen .NET uygulamaları için etkisi yoktur.

IBM MQ , bir dosyadan, ftp ya da http URL adresinden CCDT alınmasını destekler.

- `mqclient.ini` dosyası Channel kısmı kullanılarak.

**MQCCDTURL** ortam değişkeni, istemci kanal tanımlama çizelgesinin elde edilebileceği tek bir değer olarak bir dosya, ftp ya da http URL belirtmenizi sağlar.

**MQCHLLIB** ortam değişkeniyle belirtilen dizin yolunu (ya da [“İstemci yapılandırma kütüğünün kanal kısmı” sayfa 171](#) içindeki **ChannelDefinitionDirectory** özniteliğiyle belirtilen yolu), var olan yerel dosya sistemi dizinine ( /var/mqm) ek olarak bir CCDT dosyasını dosya, ftp ya da http URL yoluyla bulmak için de kullanabilirsiniz. **MQCHLLIB** değerinin bir dizin kökü olduğunu ve tam olarak nitelenmiş URL ' yi türetmek için **MQCHLTAB** ile birlikte çalıştığını unutmayın.

Bağlantılarda temel kimlik doğrulaması, URL ' de kodlanan kimlik bilgileri aracılığıyla desteklenir:

## Kimliği doğrulanmış bağlantılar

```
export MQCHLLIB=ftp://myuser:password@myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=http://myuser:password@myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
```

## Kimliği doğrulanmamış bağlantılar

```
export MQCHLLIB=ftp://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=http://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=file:///var/mqm/qmgrs/QMGR/@ipcc
```



**Not:** Kimliği doğrulanmış bağlantıları kullanmak istiyorsanız, JMS ile olduğu gibi, URL içinde kodlanmış kullanıcı adını ve parolayı belirtmeniz gerekir.

Yerli bir istemci uygulaması için istemci kanalı tanımlamasını bulma sırası:

1. MQCNO içinde **ClientConnOffset** ve **ClientConnPtr** tarafından sağlanan MQCD.
2. MQCNO içinde **CCDTUrlOffset** ve **CCDTUrlPtr** tarafından sağlanan URL .
3. **MQSERVER** ortam değişkeni.
4. Bir mqclient.ini dosyası tanımlanmışsa ve Kanallar kısmı bir **ServerConnectionParms** özniteliği içeriyorsa, tanımladığı kanal kullanılır. Daha fazla bilgi için bkz. “IBM MQ MQI client yapılandırma dosyası, mqclient.ini” sayfa 154 ve “İstemci yapılandırma kütüğünün kanal kısmı” sayfa 171.
5. **MQCCDTURL** ortam değişkeni.
6. **MQCHLLIB** ve **MQCHLTAB** ortam değişkeni.
7. “İstemci yapılandırma kütüğünün kanal kısmı” sayfa 171 içinde **ChannelDefinitionDirectory** ve **ChannelDefinitionFile** .

**Önemli:** URL kullanarak bir CCDT dosyasına erişim, file:// iletişim kuralını kullanırken bile her zaman dosyanın salt okunur bir kopyasını açar.

Yazma erişimi için bir CCDT dosyasını açma girişi; örneğin, bir istemciden **DEFINE CHANNEL** MQSC komutunu kullanırken, dosyanın yazma erişimi için açılmadığını belirten bir hata iletisi döndürür.

Ancak, **runmqsc** kullanılarak kanal ve kimlik doğrulama bilgileri tanımları okunabilir.

### İlgili görevler

“İstemci bağlantısı kanal tanımlarına erişilmesi” sayfa 56

İstemci kanal tanımlama çizelgesini (CCDT) kopyalayarak ya da paylaşarak istemci uygulamalarının kullanımına sunabilir, daha sonra istemci bilgisayardaki yerini ve adını belirtebilirsiniz. Bir istemci kanal tanımlama çizelgesini (CCDT) URL aracılığıyla da bulabilirsiniz.

“İkili biçim CCDT ' nin yapılandırılması” sayfa 42

İstemci kanal tanımlama çizelgesi (CCDT), kuyruk yöneticisine bağlanmak için istemci uygulamaları tarafından kullanılan kanal tanımlarını ve kimlik doğrulama bilgilerini belirler. Multiplatforms üzerinde, kuyruk yöneticisi yaratıldığında varsayılan ayarları içeren ikili bir CCDT otomatik olarak yaratılır. İkili CCDT ' yi güncellemek için **runmqsc** komutunu kullanın.

IBM MQ classes for JMS ile CCDT kullanılması

### İlgili başvurular

[CCDTURL](#)

[MQCNO-Bağlantı seçenekleri](#)

[XMSC\\_WMQ\\_CCDTURL](#)

## Windows Active Directory ' deki istemci bağlantısı kanalları

Active Directory'yi destekleyen Windows sistemlerinde IBM MQ , dinamik istemci-sunucu bağ tanımları sağlamak için Active Directory ' de istemci bağlantı kanallarını yayınlar.

İstemci bağlantı kanalı nesnelere tanımlandığında, bunlar varsayılan olarak AMQCLCHL . TAB adlı bir istemci kanal tanımlama dosyasına yazılır. İstemci bağlantı kanalları TCP/IP iletişim kuralını kullanıyorsa, IBM MQ sunucusu bunları Active Directory' de de yayınlar. IBM MQ istemcisi sunucuya nasıl bağlanacağını belirlediğinde, aşağıdaki arama sırasını kullanarak ilgili bir istemci bağlantı kanalı nesne tanımlamasını arar:

1. [MQCONN](#) MQCD veri yapısı
2. **MQSERVER** ortam değişkeni
3. İstemci kanal tanımlama dosyası
4. Active Directory

Bu sipariş, geçerli uygulamaların herhangi bir değişiklikten etkilenmediği anlamına gelir. Bu girdileri Active Directory ' de istemci kanal tanımlama dosyasındaki kayıtlar olarak düşünebilirsiniz ve IBM MQ istemcisi bunları aynı şekilde işler. Active Directory' de istemci bağlantısı kanal tanımlamalarını yayınlama desteğini yapılandırmak ve denetlemek için, [setmqsc](#) içinde açıklandığı gibi setmqscp komutunu kullanın.

## Sunucuda sunucu bağlantısı kanalının tanımlanması

Kuyruk yöneticisi için bir sunucu bağlantısı kanal tanımlaması yaratın.

### Yordam

1. Sunucu makinesinde, seçtiğiniz adla ve *sunucu-bağlantısı* kanal tipiyle bir kanal tanımlayın. Örneğin:

```
DEFINE CHANNEL(CHAN2) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
DESCR('Server-connection to Client_2')
```

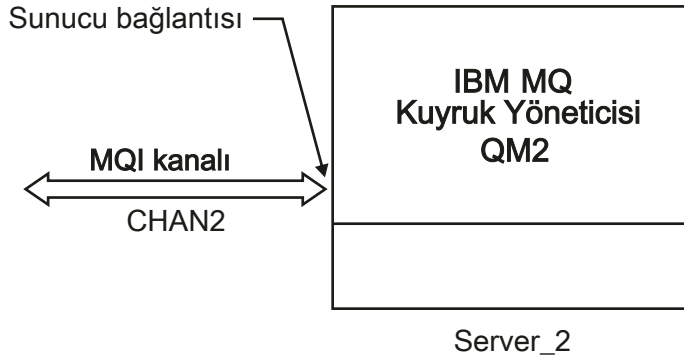
2. Kuyruk yöneticinize gelen bağlantı erişimine izin vermek için aşağıdaki komutu kullanın:

```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- Burada **SET CHLAUTH** , önceki adımda tanımlanan kanalın adını kullanır.
- Burada '*IP adresi*' IP adresi, istemcinin IP adresidir.
- Burada '*userid*' , hedef kuyruklara erişim denetimi için kanala sağlamak istediğiniz tanıtıcıdır. Bu alan büyük ve küçük harfe duyarlıdır.

Gelen bağlantınızı birkaç farklı öznitelik kullanarak tanımlamayı seçebilirsiniz. Örnek, IP adresini kullanır. Diğer öznitelikler arasında istemci kullanıcı kimliği ve TLS Konu Ayırt Edici Adı yer alır. Daha fazla bilgi için bkz. [Kanal kimlik doğrulama kayıtları](#)

Bu kanal tanımlaması, sunucuda çalışan kuyruk yöneticisiyle ilişkilendirilir.



Şekil 2. Sunucu bağlantısı kanalının tanımlanması

### İlgili görevler

“Sunucuda istemci-bağlantı kanalının tanımlanması” sayfa 54

Sunucu bağlantısı kanalını tanımladıktan sonra, ilgili istemci-bağlantı kanalını tanımlayabilirsiniz.

## Sunucuda istemci-bağlantı kanalının tanımlanması

Sunucu bağlantısı kanalını tanımladıktan sonra, ilgili istemci-bağlantı kanalını tanımlayabilirsiniz.

### Başlamadan önce

Sunucu bağlantısı kanalını tanımlayın. Daha fazla bilgi için bkz [“Sunucuda sunucu bağlantısı kanalının tanımlanması” sayfa 54.](#)

## Yordam

1. Sunucu bağlantısı kanalı ile aynı ada sahip, ancak *istemci-bağlantısı* kanalı tipinde bir kanal tanımlayın. Bağlantı adını (CONNNAME) belirtmeniz gerekir. TCP/IP için bağlantı adı, sunucu makinesinin ağ adresi ya da anasistem adıdır. İstemci ortamında çalışan IBM MQ uygulamanızın bağlanmasını istediğiniz kuyruk yöneticisi adını (QMNAME) belirtmeniz de önerilir. Kuyruk yöneticisi adını değiştirerek, farklı kuyruk yöneticilerine bağlanmak için bir kanal kümesi tanımlayabilirsiniz.

```
DEFINE CHANNEL(CHAN2) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNNAME(9.20.4.26) QMNAME(QM2) DESCR('Client-connection to Server_2')
```

2. Kuyruk yöneticinize gelen bağlantı erişimine izin vermek için aşağıdaki komutu kullanın:

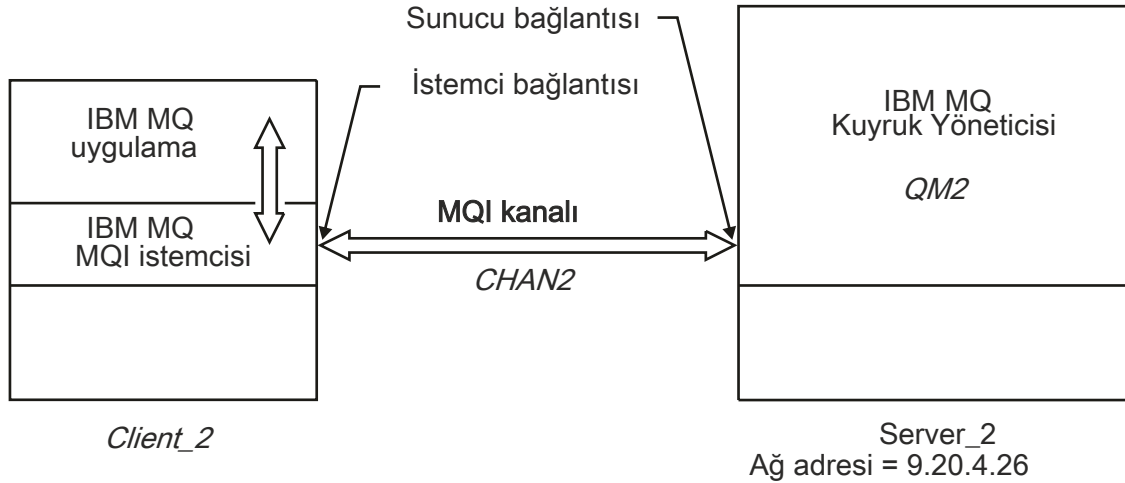
```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP-address') MCAUSER('userid')
```

- Burada **SET CHLAUTH** komutu, önceki adımda tanımlanan kanalın adını kullanır.
- Burada '*IP adresi*', istemcinin IP adresidir.
- Burada '*userid*', hedef kuyruklara erişim denetimi için kanala sağlamak istediğiniz tanıtıcıdır. Bu alan büyük ve küçük harfe duyarlıdır.

Gelen bağlantınızı birkaç farklı öznitelik kullanarak tanımlamayı seçebilirsiniz. Örnek, IP adresini kullanır. Diğer öznitelikler arasında istemci kullanıcı kimliği ve TLS Konu Ayırt Edici Adı yer alır. Daha fazla bilgi için bkz. [Kanal kimlik doğrulama kayıtları](#)

## Sonuçlar

**Multi** Çoklu platformları işletim sisteminde bu kanal tanımlaması, kuyruk yöneticisiyle ilişkilendirilmiş istemci kanal tanımlama çizelgesi (CCDT) adı verilen bir dosyada saklanır. İstemci kanal tanımlama çizelgesi birden çok istemci-bağlantı kanalı tanımlaması içerebilir. İstemci kanal tanımlama çizelgesine ilişkin ek bilgi ve istemci-bağlantı kanalı tanımlamalarının z/OS üzerinde nasıl saklandığına ilişkin bilgi için bkz. "İkili biçim CCDT 'nin yapılandırılması" sayfa 42.



Şekil 3. İstemci-bağlantı kanalının tanımlanması

### İlgili başvurular

[DEFINE CHANNEL](#) (yeni bir kanal tanımlayın)

[SET CHLAUTH](#) (kanal kimlik doğrulama kaydı oluşturma ya da değiştirme)

## İstemci bağlantısı kanal tanımlarına erişilmesi

İstemci kanal tanımlama çizelgesini (CCDT) kopyalayarak ya da paylaşarak istemci uygulamalarının kullanımına sunabilir, daha sonra istemci bilgisayardaki yerini ve adını belirtebilirsiniz. Bir istemci kanal tanımlama çizelgesini (CCDT) URLaracılığıyla da bulabilirsiniz.

### Başlamadan önce

Bu görev, gerek duyduğunuz istemci bağlantısı kanallarını CCDT ' de tanımladığınızı varsayar. Bkz. [“İstemci kanal tanımlama çizelgelerinin yapılandırılması” sayfa 41.](#)

### Bu görev hakkında

Bir istemci uygulamasının istemci kanal tanımlama çizelgesini (CCDT) kullanabilmesi için, CCDT ' yi kullanılabilir kılmamız ve yerini ve adını belirtmemiz gerekir. Bunu yapmanın birkaç yolu vardır:

- CCDT ' yi istemci bilgisayara kopyalayabilirsiniz.
- CCDT ' yi birden çok istemci tarafından paylaşılan bir konuma kopyalayabilirsiniz.
- CCDT ' yi sunucuda bulunmaya devam ederken, istemcinin paylaşılan bir dosya olarak erişmesine olanak verebilirsiniz.

IBM MQ yerel (C/C + +, COBOL ve RPG) ve yönetilmeyen .NET uygulamaları, merkezi bir konumda barındırılan CCDT 'yi URL' den (yerel dosya, ftp ya da http kaynağı olsun) çekebilir.

### Yordam

1. CCDT ' yi istemci uygulamaları için aşağıdaki yollardan biriyle kullanılabilir kılın:
  - a) İsteğe bağlı: CCDT ' yi istemci bilgisayara kopyalayın.
  - b) İsteğe bağlı: CCDT ' yi birden çok istemci tarafından paylaşılan bir yere kopyalayın.
  - c) İsteğe bağlı: CCDT ' yi sunucuda bırakın, ancak istemci tarafından paylaşılabilir yapın.
  - d) İsteğe bağlı: Yerel (C/C + +, COBOL ve RPG) ve yönetilmeyen .NET uygulamalarının CCDT 'yi bu URL' den alabilmesi için merkezi bir konumda barındırılan bir CCDT için yerel bir dosya, ftp ya da http URL tanımlayın.
2. İstemcide, CCDT ' yi içeren dosyanın yerini ve adını üç yoldan biriyle belirtin:
  - a) İsteğe bağlı: İstemci yapısını kütüğünün KANAL (KANAL) kısmı kullanılır. Daha fazla bilgi için bkz [“İstemci yapısını kütüğünün kanal kısmı” sayfa 171.](#)
  - b) İsteğe bağlı: **MQCHLLIB** ve **MQCHLTAB** ortam değişkenlerini kullanın.

Örneğin, aşağıdakileri yazarak ortam değişkenlerini ayarlayabilirsiniz:

-  **Linux**  **AIX** AIX and Linux sistemlerinde:

```
export MQCHLLIB= MQ_INSTALLATION_PATH/qmgrs/ QUEUEMANAGERNAME /@ipcc
export MQCHLTAB=AMQCLCHL.TAB
```

-  **IBM i** IBM i sistemlerinde:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE('/QIBM/UserData/mqm/qmgrs/QUEUEMANAGERNAME/@ipcc')
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(AMQCLCHL.TAB)
```

Burada *MQ\_INSTALLATION\_PATH* , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

- c) İsteğe bağlı: Yalnızca Windows sistemlerinde, istemci bağlantısı kanal tanımlarını Active Directory' de yayınlamak için **setmqscp** denetim komutunu kullanın.

d) MQCNO kullanarak programlama yaparak, ortam deęişkenlerini kullanarak ya da mqclient.ini dosya stanzlarını kullanarak URL aracılığıyla merkezi olarak barındırılan bir CCDT ' nin konumunu sağlayın. Daha fazla bilgi için bkz. “CCDT ' ye ilişkin konular” sayfa 51 ve “CCDT ' ye URL erişimi” sayfa 52.

**MQSERVER** ortam deęişkeni ayarlanırsa, bir IBM MQ istemcisi, istemci kanal tanımlama çizelgesindeki tanımlamaları tercih etmek için **MQSERVER** tarafından belirtilen istemci-baęlantı kanalı tanımlamasını kullanır.

### İlgili görevler

“İkili biçim CCDT ' nin yapılandırılması” sayfa 42

İstemci kanal tanımlama çizelgesi (CCDT), kuyruk yöneticisine bağlanmak için istemci uygulamaları tarafından kullanılan kanal tanımlarını ve kimlik doğrulama bilgilerini belirler. Multiplatforms üzerinde, kuyruk yöneticisi yaratıldığında varsayılan ayarları içeren ikili bir CCDT otomatik olarak yaratılır. İkili CCDT ' yi güncellemek için **runmqsc** komutunu kullanın.

### İlgili başvurular

MQI istemcisi: İstemci Kanal Tanımlama Çizelgesi (CCDT)

## ALW

## MQI kanalları için kanal çıkış programları

AIX, Linux, and Windows üzerindeki IBM MQ MQI client ortamı için üç tip kanal çıkışı vardır.

Bunlar:

- Çıkış gönder
- Alma çıkışı
- Güvenlik Çıkışı

Bu çıkışlar, kanalın hem istemci hem de sunucu ucunda bulunur. MQSERVER ortam deęişkenini kullanıyorsanız, uygulamanız çıkışları kullanmaz. Kanal çıkışları, [İleti sistemi kanalları için kanal çıkış programları](#) içinde açıklanır.

Gönderme ve alma çıkışları birlikte çalışır. Bunları kullanabilmenin birkaç yolu vardır:

- İletiyi bölme ve yeniden birleştirme
- Bir iletideki verileri sıkıştırma ve açma (bu işlev IBM MQ' in bir parçası olarak sağlanır, ancak farklı bir sıkıştırma teknięi kullanmak isteyebilirsiniz)
- Kullanıcı verilerinin şifrenmesi ve şifresinin çözülmesi (bu işlev IBM MQ' in bir parçası olarak sağlanır, ancak farklı bir şifreleme teknięi kullanmak isteyebilirsiniz)
- Gönderilen ve alınan her iletiyi günlüğe kaydetme

IBM MQ istemcisi ve sunucusunun doğru tanımlandığından emin olmak ve erişimi denetlemek için güvenlik çıkışını kullanabilirsiniz.

Kanal eşgörünümünün sunucu baęlantısı tarafındaki gönderme ya da alma çıkışlarının, ilişkilendirildikleri baęlantıda MQI çağruları gerçekleştirilmesi gerekiyorsa, bunlar MQCXP Hconn alanında sağlanan baęlantı tanıtıcısını kullanır. İstemci baęlantısı gönderme ve alma çıkışlarının MQI çağruları yapamayacağına farkında olmalısınız.

### İlgili kavramlar

“İstemci baęlantısında güvenlik çıkışları” sayfa 58

Bir kanalın dięer ucundaki ortaęın gerçek olduğunu doğrulamak için güvenlik çıkış programlarını kullanabilirsiniz. Bir istemci baęlantısına güvenlik çıkışı uygulandığında dikkat edilmesi gereken özel noktalar vardır.

[Kullanıcı çıkışları, API çıkışları ve IBM MQ kurulabilir hizmetleri](#)

### İlgili görevler

[Kuyruk yöneticisi olanaklarını genişletme](#)

### İlgili başvurular

“Çıkış yolu” sayfa 58

İstemci yapılanış dosyasında, kanal çıkışlarının yerinin varsayılan yolu tanımlanır. Kanal çıkışları, bir kanal başlatıldığında yüklenir.

[“Bir gönderme ya da alma çıkış programında API çağrısını tanımlama” sayfa 60](#)

İstemciler için MQI kanallarını kullandığınızda, aracı arabelleğinin 10. baytı, bir gönderme ya da alma çıkışı çağrıldığında kullanılmakta olan API çağrısını belirler. Bu, hangi kanal akışlarının kullanıcı verilerini içereceğini belirlemek için kullanışlıdır ve şifreleme ya da dijital imzalama gibi işlemlerin gerçekleştirilmesini gerektirebilir.

## ALW Çıkış yolu

İstemci yapılanış dosyasında, kanal çıkışlarının yerinin varsayılan yolu tanımlanır. Kanal çıkışları, bir kanal başlatıldığında yüklenir.

AIX, Linux, and Windows sistemlerinde, IBM MQ MQI clientkuruluşu sırasında sisteminize bir istemci yapılandırma dosyası eklenir. İstemcideki kanal çıkışlarının yeri için varsayılan yol, bu dosyada şu kısmı kullanarak tanımlanır:

```
ClientExitPath:  
ExitsDefaultPath= string  
ExitsDefaultPath64= string
```

Burada *dizgi* , altyapıya uygun biçimde bir dosya yeridir.

Bir kanal kullanıma hazırlandığında, bir MQCONN ya da MQCONNX çağrısından sonra istemci yapılandırma dosyası aranır. ClientExitYol kısmı okunur ve kanal tanımında belirtilen tüm kanal çıkışları yüklenir.

## ALW İstemci bağlantısında güvenlik çıkışları

Bir kanalın diğer ucundaki ortağın gerçek olduğunu doğrulamak için güvenlik çıkış programlarını kullanabilirsiniz. Bir istemci bağlantısına güvenlik çıkışı uygulandığında dikkat edilmesi gereken özel noktalar vardır.

[Şekil 4 sayfa 59](#) içinde, bir kullanıcının kimliğini doğrulamak için IBM MQ nesne yetki yöneticisi kullanılarak istemci bağlantısında güvenlik çıkışlarının kullanılması gösterilir.

MQCNO yapısındaki SecurityParmsPtr ya da SecurityParmsOffset alanı istemci tarafından ayarlanır ve kanalın her iki ucunda da güvenlik çıkışları vardır. Olağan güvenlik iletisi alışverişi sona erdikten ve kanal çalışmaya hazır olduktan sonra, MQCSP yapısı istemci güvenlik çıkışına geçirilir. Çıkış, MQCXP yapısındaki SecurityParms alanını kullanarak MQCSP yapısına erişebilir. Çıkış tipi MQXR\_SEC\_PARMS olarak ayarlandı. Güvenlik çıkışı, MQCSP yapısındaki kimlik bilgilerini değiştirebilir ya da değiştirmeden bırakabilir.

Çıkıştan döndürülen veriler, kanalın sunucu bağlantısı sonuna gönderilir. MQCSP yapısı, kanalın sunucu bağlantısı ucunda yeniden oluşturulur ve sunucu bağlantısı güvenlik çıkışına geçirilir. Çıkış, MQCXP yapısındaki SecurityParms alanını kullanarak MQCSP yapısına erişebilir. Güvenlik çıkışı bu verileri alır ve işler. Bu işlem tipik olarak, kuyruk yöneticisi bağlantısını yetkilendirmek için kullanılan istemci çıkışı tarafından MQCSP yapısındaki kimlik bilgilerinde yapılan değişiklikleri tersine çevirir. Sonuçtaki MQCSP yapısına, kuyruk yöneticisi sistemindeki MQCNO yapısında SecurityParmsPtr kullanılarak gönderme yapılıyor.

MQCXP yapısının SecurityParms alanıyla döndürülen bellek adresi, MQXR\_TERM değiştirinceye kadar adreslenebilir ve değişmez olarak kalmalıdır. MQXR\_TERM için çıkış çağrılmadan önce bir çıkış, belleği geçersiz kılmamalı ya da sistemi serbest bırakmamalıdır.

MQCNO yapısındaki SecurityParmsPtr ya da SecurityParmsOffset alanı ayarlıysa ve kanalın yalnızca bir ucunda bir güvenlik çıkışı varsa, güvenlik çıkışı MQCSP yapısını alır ve işler. Tamamlayıcı eylemi gerçekleştirmek için çıkış olmadığından, şifreleme gibi eylemler tek bir kullanıcı çıkışı için uygun değildir.

MQCNO yapısındaki SecurityParmsPtr ve SecurityParmsOffset alanları ayarlanmazsa ve kanalın her iki ucunda da bir güvenlik çıkışı varsa, güvenlik çıkışı ya da çıkışları çağrılır. Her iki güvenlik çıkışı da SecurityParmsPtr alanı tarafından adreslenen kendi MQCSP yapısını döndürebilir. Güvenlik

çıkışı sonlandırılıncaya kadar yeniden çağrılmaz (MQXR\_TERM içinExitReason ). Çıkış yazıcısı, o aşamada MQCSP için kullanılan belleği boşaltabilir.

Bir sunucu bağlantısı kanal eşgörünümü birden fazla etkileşim paylaşırken, güvenlik çıkışına yapılan çağrı kalıbı ikinci ve sonraki etkileşimde kısıtlanır.

İlk etkileşim için örüntü, kanal eşgörünümü etkileşimleri paylaşmıyor gibi aynıdır. İkinci ve sonraki etkileşimler için, güvenlik çıkışı hiçbir zaman MQXR\_INIT, MQXR\_INIT\_SEC ya da MQXR\_SEC\_MSG ile çağrılmaz. MQXR\_SEC\_PARMs ile çağrılır.

Paylaşım sohbetleri olan bir kanal eşgörünümünde, MQXR\_TERM yalnızca çalışan son etkileşim için çağrılır.

Her etkileşim, MQCD ' yi değiştirmek için çıkışın MQXR\_SEC\_PARMs çağrısına olanak sağlar; örneğin, kanalın sunucu bağlantısı ucunda bu özellik, kuyruk yöneticisiyle bağlantı kurulmadan önce MCAUserIdentifier ya da LongMCAUserIntPtr değerlerini değiştirmek için yararlı olabilir.

Server-connection exit	Client-connection exit
	Invoked with MQXR_INIT Responds with MQXCC_OK
Invoked with MQXR_INIT Responds with MQXCC_OK	
	Invoked with MQXR_INIT_SEC Responds with MQXCC_OK
Invoked with MQXR_INIT_SEC Responds with MQXCC_OK	
	Invoked with MQXR_SEC_PARMs Responds with MQXCC_OK
Invoked with MQXR_SEC_PARMs Responds with MQXCC_OK	
Data transfer begins	
Invoked with MQXR_TERM Responds with MQXCC_OK	Invoked with MQXR_TERM Responds with MQXCC_OK

Şekil 4. Güvenlik değiştirgelerini kullanarak istemci bağlantısı için sözleşmeyle istemci bağlantısı tarafından başlatılan değiş tokuş

**Not:** IBM WebSphere MQ 7.1 yayın düzeyinden önce oluşturulan güvenlik çıkışı uygulamalarının güncellenmesi gerekebilir. Daha fazla bilgi için bkz. [Kanal güvenliği çıkış programları](#).

## Bir gönderme ya da alma çıkış programında API çağrısını tanımlama

İstemciler için MQI kanallarını kullandığınızda, aracı arabelleğinin 10. baytı, bir gönderme ya da alma çıkışı çağrıldığında kullanılmakta olan API çağrısını belirler. Bu, hangi kanal akışlarının kullanıcı verilerini içereceğini belirlemek için kullanışlıdır ve şifreleme ya da dijital imzalama gibi işlemlerin gerçekleştirilmesini gerektirebilir.

Aşağıdaki tablo, bir API çağrısı işlenirken kanal akışının 10 bayt 'ında görüntülenen verileri göstermektedir.

**Not:** Bunlar, bu baytın tek değerleri değildir. Başka **ayrılmış** değerler vardır.

<i>Çizelge 8. API çağrılarını belirleme</i>		
<b>API çağrısı</b>	<b>İstek için bayt 10 değeri</b>	<b>Yanıt için bayt 10 değeri</b>
MQCONN " <u>1</u> " sayfa 61, " <u>2</u> " sayfa 61	X'81 '	X' 91 '
MQDISC " <u>1</u> " sayfa 61	X'82 '	X' 92 '
MQOPEN " <u>3</u> " sayfa 61	X'83 '	X' 93 '
MQCLOSE	X'84 '	X' 94 '
MQGET " <u>4</u> " sayfa 61	X'85 '	X' 95 '
MQPUT " <u>4</u> " sayfa 61	X'86 '	X' 96 '
MQPUT1 isteği " <u>4</u> " sayfa 61	X'87 '	X' 97 '
MQSET isteği	X'88 '	X' 98 '
MQINQ isteği	X'89 '	X' 99 '
MQCMIT isteği	X'8A'	X'9A'
MQBACK isteği	X'8B'	X'9B'
MQSTAT isteği	X'8D'	X'9D'
MQSUB isteği	X'8E'	X'9E'
MQSUBRQ isteği	X'8F'	X'9F'
xa_start isteği	X'A1'	X'B1'
xa_end isteği	X'A2'	X'B2'
xa_open isteği	X'A3'	X'B3'
xa_close isteği	X'A4'	X'B4'
xa_prepare isteği	X'A5'	X'B5'
xa_commit isteği	X'A6'	X'B6'
xa_rollback isteği	X'A7'	X'B7'
xa_unut isteği	X'A8'	X'B8'
xa_recover isteği	X'A9'	X'B9'
xa_complete isteği	X'AA'	X'BA'

**Notlar:**



1. İstemci ile sunucu arasındaki bağlantı, istemci uygulaması tarafından MQCONN kullanılarak başlatılır. Bu nedenle, özellikle bu komut için, başka birkaç ağ akışı vardır. Aynı durum, ağ bağlantısını sonlandıran MQDISC için de geçerlidir.
2. MQCONNX, istemci-sunucu bağlantısı amacıyla MQCONN ile aynı şekilde işlenir.
3. Büyük bir dağıtım listesi açılırsa, gerekli tüm verileri SVRCONN MCA ' ya geçirmek için MQOPEN çağrısı başına birden fazla ağ akışı olabilir.
4. Büyük iletiler, iletim kesimi büyüklüğünü aşabilir. Bu gerçekleşirse, tek bir API çağrısından kaynaklanan birçok ağ akışı olabilir.

z/OS

## İstemcinin kuyruk paylaşım grubuna bağlanması

Bir kuyruk paylaşım grubunun üyesi olan bir sunucudaki bir istemci ile kuyruk yöneticisi arasında bir MQI kanalı yaratarak bir istemciyi kuyruk paylaşım grubuna bağlayabilirsiniz.

### Bu görev hakkında

Bir kuyruk paylaşım grubu, aynı paylaşılan kuyruk kümesine erişebilen bir kuyruk yöneticileri kümesi tarafından oluşturulur. Paylaşılan kuyruklar hakkında daha fazla bilgi için bkz. [Paylaşılan kuyruklar ve kuyruk paylaşım grupları](#).

Paylaşılan kuyruğa konan bir istemci, kuyruk paylaşım grubunun herhangi bir üyesine bağlanabilir. Bir kuyruk paylaşım grubuna bağlanmanın yararları, ön uç ve arka uç kullanılabilirliği ve artırılmış kapasitede olası artışlardır. Belirli bir kuyruk yöneticisine ya da soysal arabirime bağlanabilirsiniz.

Kuyruk paylaşım grubundaki bir kuyruk yöneticisine doğrudan bağlanma, iletileri paylaşılan bir hedef kuyruğa koyabilmenizi sağlar ve bu da arka uç kullanılabilirliğini artırır.

Bir kuyruk paylaşım grubunun soysal arabirimine bağlanma, gruptaki kuyruk yöneticilerinden biriyle bir oturum açar. İstemci kuyruk yöneticisi gruptaki herhangi bir kuyruk yöneticisine bağlanabildiğinden bu, ön uç kullanılabilirliğini artırır. Kuyruk paylaşım grubu içindeki belirli bir kuyruk yöneticisine bağlanmak istemediğinizde, soysal arabirimi kullanarak gruba bağlanırsınız.

Soysal arabirim, bir Sysplex Distributor VIPA adresi ya da VTAM soysal kaynak adı ya da kuyruk paylaşım grubu için başka bir ortak arabirim olabilir. Soysal bir arabirim ayarlanmasıyla ilgili daha fazla ayrıntı için [Kuyruk paylaşım gruplarını kullanarak IBM MQ for z/OS için iletişimi ayarlamabaşlıklı konuya](#) bakın.

### Yordam

Bir kuyruk paylaşım grubunun soysal arabirimine bağlanmak için, gruptaki herhangi bir kuyruk yöneticisi tarafından erişilebilen kanal tanımlamaları yaratmanız gerekir. Bunu yapmak için, gruptaki her kuyruk yöneticisinde aynı tanımlamalara sahip olmanız gerekir.

1. SVRCONN kanalını aşağıdaki örnekte gösterildiği gibi tanımlayın:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
QSGDISP(GROUP)
```

Sunucudaki kanal tanımlamaları paylaşılan bir Db2 havuzunda saklanır. Kuyruk paylaşım grubundaki her kuyruk yöneticisi, tanımlamanın yerel bir kopyasını oluşturur ve bir MQCONN ya da MQCONNX çağrısı yayınladığınızda her zaman doğru sunucu bağlantısı kanalına bağlanmanızı sağlar.

2. CLNTCONN kanalını aşağıdaki örnekte gösterildiği gibi tanımlayın:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME( VIPA address ) QMNAME(QSG1) +
DESCR('Client-connection to Queue Sharing Group QSG1') QSGDISP(GROUP)
```

## Sonuçlar

Kuyruk paylaşım grubunun soysal arabirimi istemci-bağlantı kanalındaki CONNAME alanında saklandığı için, artık gruptaki herhangi bir kuyruk yöneticisine bağlanabilir ve o grubun sahip olduğu paylaşılan kuyruklara yerleştirebilirsiniz.

## IBM MQ ortam değişkenlerinin kullanılması

Geçerli ayarları görüntülemek ya da IBM MQ ortam değişkenlerinin değerlerini sıfırlamak için komutları kullanabilirsiniz.

### Bu görev hakkında

Ortam değişkenlerini aşağıdaki şekillerde kullanabilirsiniz:

- Kalıcı bir değişiklik yapmak üzere sistem profilinizdeki değişkenleri ayarlamak için
- Yalnızca bu oturum için değişiklik yapmak üzere komut satırından bir komut vermek için
- Bir ya da daha çok değişkene çalışmakta olan uygulamaya bağlı olarak belirli bir değer vermek için, uygulama tarafından kullanılan bir komut dosyası dosyasına komut ekleyin

Her ortam değişkeni için, komutları kullanarak yürürlükteki ayarı görüntüleyebilir ya da ortam değişkeninin değerini ilk durumuna getirebilirsiniz. Bu komutlar, tersi belirtilmedikçe, desteklenen tüm platformlarda kullanılabilir. Komutun biçimi altyapınıza bağlıdır. Örneğin:

- **Linux** **AIX** AIX and Linux'ta:

```
export [environment variable]=value
```

- **Windows** Windows'ta:

```
Set [environment variable]=value
```

- **IBM i** IBM i'ta:

```
ADDENVVAR ENVVAR(environment variable) VALUE(xx)
```

- **MQ Appliance** IBM MQ Appliance için, IBM MQ Appliance belgelerinde [Configuring environment variable on IBM MQ Appliance](#) başlıklı konuya bakın.

Uygulanabilir olduğunda, IBM MQ ayarlamadığınız ortam değişkenleri için varsayılan değerleri kullanır.

**Not:** **z/OS** IBM MQ for z/OS , herhangi bir IBM MQ ortam değişkenlerini desteklemez. Bu altyapıyı sunucunuz olarak kullanıyorsanız, z/OS üzerinde istemci kanal tanımlama çizelgesinin nasıl oluşturulduğu hakkında bilgi için [İstemci kanal tanımlama çizelgesi](#) başlıklı konuya bakın. İstemci altyapınızda IBM MQ ortam değişkenlerini kullanmaya devam edebilirsiniz.

### Yordam

- **Windows**

Windows' ta, her ortam değişkeni için, geçerli ayarı görüntülemek ya da bir değişkenin değerini sıfırlamak için aşağıdaki komutları kullanın:

- Bir ortam değişkeninin değerini kaldırmak için aşağıdaki komutu kullanın:

```
SET MQSERVER=
```

- Bir ortam değişkeninin yürürlükteki ayarını görüntülemek için aşağıdaki komutu kullanın:

```
SET MQSERVER
```

- Oturuma ilişkin tüm ortam değişkenlerini görüntülemek için aşağıdaki komutu kullanın:

```
set
```

- **Linux** **AIX**

AIX and Linux' ta, her ortam değişkeni için, geçerli ayarı görüntülemek ya da bir değişkenin değerini sıfırlamak için aşağıdaki komutları kullanın:

- Bir ortam değişkeninin değerini kaldırmak için aşağıdaki komutu kullanın:

```
unset MQSERVER
```

- Bir ortam değişkeninin yürürlükteki ayarını görüntülemek için aşağıdaki komutu kullanın:

```
echo $MQSERVER
```

- Oturuma ilişkin tüm ortam değişkenlerini görüntülemek için aşağıdaki komutu kullanın:

```
set
```

### İlgili görevler

[IBM MQ classes for JMS/Jakarta Messaging için ortam değişkenlerini ayarlama](#)

[IBM MQ classes for Java ile ilgili ortam değişkenleri](#)

[service.env dosyasında ek ortam değişkenlerinin tanımlanması](#)

“Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerini değiştirme” sayfa 83

Yapılandırma (.ini) dosyalarındaki bilgileri düzenleyerek, IBM MQ ya da tek bir kuyruk yöneticisinin davranışını kuruluşunuzun gereksinimlerine uyacak şekilde değiştirebilirsiniz. IBM MQ MQI clients için yapılandırma seçeneklerini de değiştirebilirsiniz.

### İlgili başvurular

[MFT özelliklerinde ortam değişkenlerinin kullanımı](#)

## Ortam değişkenleri açıklamaları

Müşteri kullanımı için tasarlanan sunucu ve istemci ortam değişkenlerinin açıklamaları.

### Kullanım örnekleri

- **Linux** **AIX** AIX and Linux sistemlerinde şu biçimi kullanın: `export [environment variable]=value`.
- **Windows** Windows sistemlerinde şu biçimi kullanın: `Set [environment variable]=value`.
- **IBM i** IBM i sistemlerinde şu biçimi kullanın: `ADDENVVAR ENVVAR(environment variable) VALUE(xx)`.
- **MQ Appliance** IBM MQ Appliance için, IBM MQ Appliance belgelerinde [Configuring environment variable on IBM MQ Appliance](#) başlıklı konuya bakın.

### AMQ\_ALLOWED\_CIPHERS

**Multi**

IBM MQ 9.2.0' den, Multiplatforms üzerinde IBM MQ kanallarıyla kullanılmak üzere etkinleştirilen özel bir CipherSpecs listesi belirtmek için **AMQ\_ALLOWED\_CIPHERS** ortam değişkenini kullanabilirsiniz. Ortam değişkeni, .ini dosyasının **AllowedCipherSpecs** SSL kısmı özniteliğiyle aynı değerleri alır:

- Tek bir CipherSpec adı ya da
- Yeniden etkinleştirilecek IBM MQ CipherSpec adlarının virgülle ayrılmış bir listesi ya da
- Tüm CipherSpecs 'i gösteren ALL özel değeri (önerilmez).

**Not:** **ALL** CipherSpecs özelliğinin etkinleştirilmesi, SSL 3.0 ve TLS 1.0 iletişim kurallarını ve çok sayıda zayıf şifreleme algoritmasını etkinleştireceği için önerilmez.

Daha fazla bilgi için bkz. [TLS el sıkışmasında CipherSpec siparişi içinde Çoklu platformlarda etkinleştirilen CipherSpecs için özel bir liste sağlama](#) .

## AMQ\_BAD\_COMMS\_DATA\_FDCS

**AMQ\_BAD\_COMMS\_DATA\_FDCS** ortam değişkeni, herhangi bir değere ayarlandığında geçerlidir.

IBM MQ ' in TCP/IP üzerinden anasistemden aldığı veriler yanlış biçimdeyse (örneğin, bir ağ istemcisi bir IBM MQ dinleyici kapisına bağlanmış ve desteklenmeyen bir uygulama protokolüyle iletişim kurmaya çalışmışsa), kuyruk yöneticisi kuyruk yöneticisi hata günlüklerine bir **AMQ9207E** hata iletisi yazar. IBM MQ dinleyicileri, kuyruk yöneticisi ileti kanalı araçlarından (MCA) ve MQI, JMS ve XMS istemci uygulamalarından TCP/IP bağlantılarını destekler.

**Not:** IBM MQ dinleyicileri, AMQP ve MQTT istemcileri tarafından kullanılan uygulama protokolünü desteklemez; bunun yerine, bu istemciler geçerli AMQP kanalında ya da MQXR telemetri hizmetinde yapılandırılan ağ kapılarına bağlanmalıdır.

IBM MQ ' in aldığı geçersiz verileri içeren bir başarısızlık verileri yakalama (FDC) kaydı da yazılabilir. Ancak, bu uzak tarafla bir etkileşimin başlangıcıydı ve biçim, HTTP web tarayıcısından GET isteği gibi bilinen basit bir biçimse, FFST dosyası oluşturulmaz. FFST dosyalarının bilinen basit biçimler de içinde olmak üzere hatalı veriler için yazılmasını sağlamak üzere bunu geçersiz kılmak istiyorsanız, **AMQ\_BAD\_COMMS\_DATA\_FDCS** ortam değişkenini herhangi bir değere (örneğin, TRUE) ayarlayabilir ve kuyruk yöneticisini yeniden başlatabilirsiniz.

## AMQ\_CONVEBCDICNEWLINE

Multi

**AMQ\_CONVEBCDICNEWLINE** ortam değişkenini kullanarak, IBM MQ ' un bir EBCDIC NL karakterini ASCII biçimine nasıl dönüştüreceğini belirleyebilirsiniz. Ortam değişkeni, `mqs.ini` dosyasının **ConvEBCDICNewLine** özniteliğiyle (NL\_TO\_LF, TABLEya da ISO) aynı değerleri alır (bkz. `mqs.ini` dosyasının tüm kuyruk yöneticileri bölümü). Örneğin, `mqs.ini` dosyasının kullanılmadığı durumlarda istemci tarafında **ConvEBCDICNewLine** işlevselliği sağlamak için **ConvEBCDICNewLine** kısmı özniteliği yerine **AMQ\_CONVEBCDICNEWLINE** ortam değişkenini kullanabilirsiniz. Hem stanza özniteliği hem de ortam değişkeni ayarlanırsa, stanza özniteliği öncelikli olur.

Daha fazla bilgi için bkz. [Kodlanmış karakter takımları arasında veri dönüştürme](#) .

## AMQ\_DIAGNOSTIC\_MSG\_ÖNEM DÜZEYİ

**AMQ\_DIAGNOSTIC\_MSG\_SEVERITY** ortam değişkeni bir IBM MQ işlemi için 1 olarak ayarlanırsa, IBM MQ işlemi bir hata günlüğüne ya da konsola bir ileti yazdığı anda ileti önem derecesine tek bir büyük harf alfabetik karakter olarak eklenmesine neden olur.

**AMQ\_DIAGNOSTIC\_MSG\_SEVERITY** ' in etkinleştirdiği davranış varsayılan olarak ayarlanır. Ortam değişkenini 0 olarak ayarlayarak bu davranışı kapatabilirsiniz.

Daha fazla bilgi için bkz. [Hata günlüklerini kullanma](#) .

## AMQ\_DISABLE\_CLIENT\_AMS

Ürünün önceki bir sürümünden bir kuyruk yöneticisine bağlanmaya çalışırken 2085 (MQRC\_UNKNOWN\_OBJECT\_NAME) hatası bildirilirse ve aşağıdaki istemcilerden birini kullanıyorsanız, istemcide IBM MQ Advanced Message Security (AMS) öğesini geçersiz kılmak için

**AMQ\_DISABLE\_CLIENT\_AMS** ortam değişkenini kullanabilirsiniz:

- IBM Java runtime environment (JRE) dışında bir Java runtime environment (JRE)
- Bir IBM MQ IBM MQ classes for JMS ya da IBM MQ classes for Java istemcisi.

**Not:** C istemcileri için **AMQ\_DISABLE\_CLIENT\_AMS** ortam deęişkenini kullanamazsınız. Bunun yerine **MQS\_DISABLE\_ALL\_INTERCEPT** ortam deęişkenini kullanmanız gerekir.

Daha fazla bilgi için bkz. [İstemcide Gelişmiş İleti Güvenliğinin Devre Dışı Bırakılması](#).

## AMQ\_DMPMQCFG\_QSGDISP\_DEFAULT

**dmpmqcfg** komutu tarafından kullanılan bir kuyruk yöneticisinin yok edilmesiyle ilgili sorgular varsayılan olarak yalnızca QSGDISP (QMGR) tanımlarını sorgular. Aşağıdaki deęerlerden birine ayarlanabilen **AMQ\_DMPMQCFG\_QSGDISP\_DEFAULT** ortam deęişkenini kullanarak ek tanımlamaları sorgulayabilirsiniz:

### Canlı

Yalnızca, QSGDISP (QMGR) ya da QSGDISP (COPY) ile tanımlanan nesnelere ekleyin.

### TÜMÜ

QSGDISP (QMGR) ve QSGDISP (COPY) ile tanımlanan nesnelere içer. Kuyruk yöneticisi bir kuyruk paylaşım grubunun üyesiye, QSGDISP (GROUP) ve QSGDISP (SHARED) de içerilir.

### Kopyala

Yalnızca QSGDISP (COPY) ile tanımlanan nesnelere içer

### GRUP

Yalnızca QSGDISP (GROUP) ile tanımlanan nesnelere içer; hedef kuyruk yöneticisi bir kuyruk paylaşım grubunun üyesi olmalıdır.

### QMGR

Yalnızca QSGDISP (QMGR) ile tanımlanan nesnelere dahil edin. Bu ortam deęişkenini, var olan **dmpmqcfg** davranışını eşleştirmek için kullanırsanız, bu varsayılan davranıştır.

### ÖZEL

Yalnızca, QSGDISP (QMGR) ya da QSGDISP (COPY) ile tanımlanan nesnelere ekleyin.

### PAYLAŞILAN

Yalnızca, QSGDISP (SHARED) ile tanımlanan nesnelere içer.

## AMQ\_IODELAY, AMQ\_IODELAY\_INMS ve AMQ\_IODELAY\_FFST

Multi

V 9.3.4

IBM MQ , günlük okuma ve yazma ya da giriş ve çıkış işlemlerinin beklenenden daha uzun sürmesini algılar. Bunun nedeni, işletim sistemi ya da depolama sistemiyle ilgili sorunlar olabilir ve kuyruk yöneticisi başarımını etkileyebilir. IBM MQ 9.3.4' den, kuyruk yöneticisi günlüğü ve depolama dosyası sisteminize ilişkin giriş ve çıkış yavaş olduğunda tanılama ve zaman ayarlarını yapmak için **AMQ\_IODELAY** ortam deęişkenlerini kullanabilirsiniz. Kuyruk yöneticisi hata günlüğünde [AMQ6729W](#) Günlük G/Ç işlemi eşliği aşıldı iletisini görürseniz, nedeni araştırın ve ayarlamaları uygun şekilde yapın. Aşağıdaki örneklerde gösterildiği gibi deęişkenleri kullanın:

### AMQ\_IODELAY

Eşik süresi (saniye), varsayılan deęer 1 saniyedir. Bir G/Ç işlemi bu eşikten uzun sürerse, IBM MQ günlük dosyalarında AMQ6729W hata iletisi bildirilir. Gecikmeler devam ederse uyarı iletisi en fazla 10 saniyede bir yinelenir. Hataları engellemek için bunu artırabilir ya da belirli performans sorunlarını araştırmak için azaltabilirsiniz. Örneğin,

```
export AMQ_IODELAY=200000
```

### AMQ\_IODELAY\_INMS

Zaman ölçüsünü saniye yerine mikrosaniye olarak deęiştirin. Kuyruk yöneticisi günlüğünde AMQ6729 iletisini almadan önce daha düşük bir eşik belirlemek için bu seçeneği kullanın.

```
export AMQ_IODELAY_INMS=YES
```

### AMQ\_IODELAY\_FFST

Hata günlüğündeki uyarı iletisine ek olarak, eşik aşıldığında tanılama bilgilerini içeren bir FFST dosyası oluşturulur.

```
export AMQ_IODELAY_FFST=YES
```

Kuyruk yöneticisinin bu örnekte olduğu gibi başlatılması, bir giriş/çıkış işleminin 200000 mikrosaniyeden (0.2s) uzun sürmesi durumunda bir FDC ya da FFST dosyasının yazılmasına neden olur; bu işlem hala görece cömert bir eşiktir.

Daha fazla bilgi için bkz. [Kuyruk yöneticisi sağlıklı işletim denetimi davranışı](#).

## AMQ\_LDAP\_TRACE

**AMQ\_LDAP\_TRACE** ortam değişkeni boş olmayan bir değere ayarlanırsa, kuyruk yöneticisini durdurmadan ya da başlatmadan LDAP istemcisi izlemesi açılıp kapanabilir.

Daha fazla bilgi için [LDAP istemcisi kitaplık kodunun dinamik izlemesini etkinleştirme](#) başlıklı konuya bakın.

## AMQ\_LICENSING\_METRIC

Multi

**AMQ\_LICENSING\_METRIC=VPCMonthlyPeak** ortam değişkeninin ayarlanması, kuyruk yöneticisinin saatlik kapsayıcı tabanlı lisanslarla ilgili verileri karşıya yükleme varsayılan davranışı yerine aylık VPC lisans tipleriyle ilgili verileri karşıya yüklemesine neden olur.

IBM MQ ' in IBM Cloud Private ölçüm hizmetiyle birlikte kullanılmak üzere yapılandırılmasına ilişkin ek bilgi için IBM Cloud Private belgelerinde [IBM Cloud Private ölçüm hizmeti](#) konusuna bakın.

## AMQ\_MQS\_INI\_LOCATION

Linux

AIX

AIX and Linux sistemlerinde, **AMQ\_MQS\_INI\_LOCATION** ortam değişkeninde `mqs.ini` kütüğünün yerini ayarlayarak `mqs.ini` kütüğü için kullanılan yeri değiştirebilirsiniz. Bu ortam değişkeni sistem düzeyinde ayarlanmalıdır.

Dizin konumları da içinde olmak üzere `mqs.ini` dosyasıyla ilgili daha fazla bilgi için bkz. [IBM MQ yapılandırma dosyası, mqs.ini](#).

## AMQ\_NO\_BAD\_COMMS\_DATA\_FDCS

**AMQ\_NO\_BAD\_COMMS\_DATA\_FDCS** ortam değişkeni, herhangi bir değere ayarlandığında geçerlidir.

IBM MQ ,IBM MQ dışı bir istemciyi IBM MQ TCP/IP dinleyicisine bağlamaya çalışırken ilk veri iletimini tanımazsa, kuyruk yöneticisinin kuyruk yöneticisi hata günlüklerine bir [AMQ9207E](#) hata ileti yazmasına neden olur. Hata verileri yakalama (FDC) kaydı da yazılır.

**AMQ\_NO\_BAD\_COMMS\_DATA\_FDCS** ortam değişkeniyle bu tanılama dosyalarının oluşturulmasını engelleyebilirsiniz. **AMQ\_NO\_BAD\_COMMS\_DATA\_FDCS** herhangi bir değere ayarlandığında (örneğin, TRUE), bu işlem IBM MQ ilk iletişim akışında [AMQ9207E](#) hata iletilerini oluşturmasını bildirir. Geçerli olması için, kuyruk yöneticisi ve dinleyici işlemleri başlatılmadan önce ortam değişkeni ayarlanmalıdır.

FDC, bir istemcinin kuyruk yöneticisine geçerli bir IBM MQ protokolü gönderdiği ve daha sonra geçersiz veriler gönderdiği durumlarda oluşturulmaya devam eder; bu durum, daha fazla araştırma yapılmasını garanti eden bir istemci sorununun göstergesidir.

**Not:** IBM MQ 9.2.0' den [AMQ9207E](#) ilk iletişim akışlarında hata iletileri raporlanırken FFSTlerin yakalanması varsayılan olarak engellenir.

## AMQ\_NO\_IPV6

**AMQ\_NO\_IPV6** ortam değişkeni, herhangi bir değere ayarlandığında geçerlidir. Bu ortam değişkeni ayarlandığında, bağlantı girişimi sırasında IPv6 kullanımı devre dışı bırakılır.

## AMQ\_REVERSE\_COMMIT\_ORDER

**AMQ\_REVERSE\_COMMIT\_ORDER** ortam değişkeni, bir kuyruk yöneticisini yapılandırır; böylece, bir XA hareketinde IBM MQ kuyruk yöneticisi değişikliği, ilgili veritabanı güncellemesi tamamlandıktan sonra kesinleştirilir. Kuyruklardan ileti okuyan uygulamalar, yalnızca ilgili veritabanı güncellemesi tamamlandıktan sonra ileti görürler.

**Not:** Yalıtım düzeyi içinde açıklanan senaryoyu okumadan ve anlamadan **AMQ\_REVERSE\_COMMIT\_ORDER** ayarını tanımlamayın.

## AMQ\_SSL\_ALLOW\_DEFAULT\_CERT

**AMQ\_SSL\_ALLOW\_DEFAULT\_CERT** ortam değişkeni ayarlanmadığında, bir uygulama istemci anahtar deposunda kişisel sertifikayla bir kuyruk yöneticisine bağlanabilir; ancak, sertifika `ibmwebspheremquserid` etiket adını içerdiğinde. **AMQ\_SSL\_ALLOW\_DEFAULT\_CERT** ortam değişkeni ayarlandığında, sertifika `ibmwebspheremquserid` etiket adını gerektirmez. Bir kuyruk yöneticisine bağlanmak için kullanılan sertifika, anahtar havuzunda varsayılan bir sertifika bulunması ve anahtar havuzunun `ibmwebspheremquserid` önekiğine sahip bir kişisel sertifika içermemesi koşuluyla, varsayılan bir sertifika olabilir.

1 değeri, varsayılan sertifika kullanımını etkinleştirir.

Bir uygulama, **AMQ\_SSL\_ALLOW\_DEFAULT\_CERT** ortam değişkenini kullanmak yerine `mqclient.ini` dosyasındaki SSL kısmına ilişkin **CertificateLabel** ayarını kullanabilir. Daha fazla bilgi için bkz. [Sayısal sertifika etiketleri, gereksinimleri anlama ve istemci yapılandırma dosyasının SSL kısmı](#).

## AMQ\_SSL\_LDAP\_SERVER\_VERSION

**AMQ\_SSL\_LDAP\_SERVER\_VERSION** ortam değişkeni, CRL sunucularının LDAP iletişim kuralının belirli bir sürümünün kullanılmasını gerektirdiği durumlarda IBM MQ şifreleme bileşenleri tarafından LDAP v2 ya da LDAP v3 kullanılmasını sağlamak için kullanılabilir.

Ortam değişkenini, kuyruk yöneticisini ya da kanalı başlatmak için kullanılan ortamda uygun değere ayarlayın:

- LDAP v2 'nin kullanılmasını istemek için `AMQ_SSL_LDAP_SERVER_VERSION=2` değerini ayarlayın.
- LDAP v3 kullanılmasını istemek için `AMQ_SSL_LDAP_SERVER_VERSION=3` değerini ayarlayın.

Bu ortam değişkeni, kullanıcı kimlik doğrulaması ya da kullanıcı yetkilendirmesi için IBM MQ kuyruk yöneticisi tarafından kurulan LDAP bağlantılarını etkilemez.

## AMQ\_USE\_ZLIBNX



AIX işletim sistemlerinde **AMQ\_USE\_ZLIBNX** ortam değişkeni, ileti kanalı araçlarının (MCA) ZLIBFAST ya da ZLIBHIGH tekniklerini kullanırken ileti verilerinin sıkıştırılması ve sıkıştırmasının açılması için donanım hızlandırılmalı `zlibNX` kitaplığını kullanmasını sağlamak üzere kullanılabilir.

**İpucu:** 2 KB 'nin üzerinde boyuta sahip yüksek oranda sıkıştırılabilir iletiler, CPU kullanımını azaltarak `zlibNX` kitaplığını kullanmayı seçmekten yararlanabilir.

`zlibNX` kitaplığı, Technology Level 4 Expansion Pack ve sonraki sürümleriyle IBM AIX 7.2 içinde bulunur. Ortam değişkeni ayarlıysa ve `zlibNX` kitaplığı (`/usr/opt/zlibNX/lib/libz.a`) kurulu değilse, ileti kanalı araçları IBM MQ for AIX kuruluşunda sağlanan standart `zlib` kitaplığını kullanır.

## GİRİŞ SAYFASI



AIX, Linux ve IBM i sistemlerinde **HOME** ortam değişkeni, `mqclient.ini` dosyası için aranan dizinin adını belirtir. Bu dosya, IBM MQ MQI client tarafından kullanılan yapılandırma bilgilerini içerir.



Daha fazla bilgi için bkz. [IBM MQ MQI istemcisi yapılandırma dosyası](#), [mqclient.ini](#) ve [İstemci yapılandırma dosyasının konumu](#).

## HOMEDRIVE ve HOMEPATH

Windows

Kullanılabilmesi için hem **HOMEDRIVE** hem de **HOMEPATH** ortam değişkenlerinin ayarlanması gerekir. Bunlar, Windows sistemlerinde `mqclient.ini` dosyası için aranan dizinin adını belirtmek için kullanılır. Bu dosya, IBM MQ MQI client tarafından kullanılan yapılandırma bilgilerini içerir.

Daha fazla bilgi için bkz. [IBM MQ MQI istemcisi yapılandırma dosyası](#), [mqclient.ini](#) ve [İstemci yapılandırma dosyasının konumu](#).

## LDAP\_BASEDN

**LDAP\_BASEDN** , bir LDAP örnek programını çalıştırmak için gerekli ortam değişkenidir. Dizin aramasına ilişkin temel Ayırt Edici Adı belirler.

## LDAP\_HOST

**LDAP\_HOST** , LDAP örnek programını çalıştırmak için isteğe bağlı bir ortam değişkenidir. LDAP sunucusunun çalıştığı anasistemin adını belirler; belirtilmezse, varsayılan olarak yerel anasisteme ayarlanır.

## LDAP\_SÜRÜMÜ

**LDAP\_VERSION** , LDAP örnek programını çalıştırmak için isteğe bağlı bir ortam değişkenidir. Kullanılacak LDAP protokolünün sürümünü belirtir ve 2 ya da 3 olabilir. Çoğu LDAP sunucusu artık protokolün 3. sürümünü desteklemektedir; hepsi eski sürüm 2 'yi desteklemektedir. Bu örnek, protokolün her iki sürümüyle de aynı şekilde çalışır ve belirtilmezse, varsayılan olarak sürüm 2 'ye ayarlanır.

## MQ\_CHANNEL\_SUPPRESS\_INTERVAL

**MQ\_CHANNEL\_SUPPRESS\_INTERVAL** ortam değişkeni, **MQ\_CHANNEL\_SUPPRESS\_MSGS** ile tanımlanan iletilerin, engellenmeden önce belirtilen zaman aralığında bir iletinin kaç kez geçmesine izin verileceği ile birlikte hata günlüğüne yazılması engelleneceği zaman aralığını saniye cinsinden belirtir. Varsayılan değer 60,5 'tir; bu, belirli bir iletinin ilk beş geçişinden sonra 60 saniyelik bir aralıkta daha fazla tekrarının engelleneceği anlamına gelir. Daha fazla bilgi için [Multiplatforms üzerindeki hata günlüklerinden kanal hata iletilerini engelleme başlıklı konuya](#) bakın.

**MQ\_CHANNEL\_SUPPRESS\_INTERVAL** ortam değişkeni, `qm.ini` dosyasındaki `SuppressInterval` ile karşılaştırılabilir.

## MQ\_CHANNEL\_SUPPRESS\_MSGS

**MQ\_CHANNEL\_SUPPRESS\_MSGS** ortam değişkeni, hata günlüğündeki kanal hata iletilerini gizler. Gizlenen iletilerin bir listesini belirtebilirsiniz. **MQ\_CHANNEL\_SUPPRESS\_MSGS** , her iletinin gizlenmeden önce kaç kez görüneceğini ve iletilerin gizleneceği süreyi belirten **MQ\_CHANNEL\_SUPPRESS\_INTERVAL** ile birlikte kullanılır. Daha fazla bilgi için [Multiplatforms üzerindeki hata günlüklerinden kanal hata iletilerini engelleme başlıklı konuya](#) bakın.

**MQ\_CHANNEL\_SUPPRESS\_MSGS** ortam değişkeni, `qm.ini` dosyasındaki `SuppressMessage` ile karşılaştırılabilir, ancak ortam değişkenini kullanarak herhangi bir kanal iletilisini engelleyebilirsiniz, `qm.ini` yöntemi için kısıtlayıcı bir liste vardır.

## MQ\_BAĞLANTI TIPI

Multi



Multiplatforms üzerinde, bir MQCONNX çağrısında kullanılan MQCNO yapısının Options (Seçenekler) alanında belirtilen bağ tanımı tipiyle birlikte **MQ\_CONNECT\_TYPE** ortam değişkenini kullanabilirsiniz. **MQ\_CONNECT\_TYPE** ' in yalnızca STANDARD bağ tanımları için etkisi vardır. Diğer bağlamalar için **MQ\_CONNECT\_TYPE** yoksayılr.

Daha fazla bilgi için bakınız: [Use of MQCONNX call options with MQ\\_CONNECT\\_TYPE](#).

## MQ\_CROSS\_QUEUE\_ORDER\_ALL

**MQ\_CROSS\_QUEUE\_ORDER\_ALL** ortam değişkenini sıfır dışında bir değere ayarladığınızda, ileti koyma sırası bir iş biriminde tutulur. Bu, bir İş Birimi 'ndeki (UoW) iletiler birden çok kuyruğa (örneğin, Q1, Q2) konursa, bir MQCMIT yayınlandığında iletilerin teslim edildiği " Direktör0 çok çok lerindeki gönderildi çok ede çok yorsa: (LAR ne lar?).

Çok kuyruklu bir yönetici ortamında, her kuyruk yöneticisi başlatılmadan önce, **MQ\_CROSS\_QUEUE\_ORDER\_ALL** ' in hem gönderme hem de alma tarafında boş olmayan bir değeri olmalıdır.

## MQ\_EPHEMERAL\_PREFIX

**MQ\_EPHEMERAL\_PREFIX** ortam değişkeni, kuyruk yöneticisi çalışırken geçici kuyruk yöneticisi verilerinin alıkonduğu kuyruk yöneticisinin geçici dizininin yolunu belirtir.

mqs.ini dosyasının AllQueueManager kısmındaki **DefaultEphemeralPrefix** özniteliğinde **EphemeralPrefix** özniteliğini değiştirerek geçici öneki değiştirmenin bir alternatifi olarak, **crtmqm** komutu için **EphemeralPrefix** ' yi geçersiz kılmak üzere **MQ\_EPHEMERAL\_PREFIX** ortam değişkenini kullanabilirsiniz. Daha fazla bilgi için bkz. [Yapılandırılabilir geçici dizin](#).

## MQ\_FILE\_PATH

Windows

**MQ\_FILE\_PATH** ortam değişkeni, Windows platformunda çalıştırma zamanı paketinin kuruluğu sırasında yapılandırılır. Bu ortam değişkeni, Windows kaydında aşağıdaki anahtarla aynı verileri içerir:

```
HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName\FilePath
```

Daha fazla bilgi için bkz. [setmqenv \(set IBM MQ environment\)](#) ve [crtmqenv \(create IBM MQ environment\)](#).

## MQ\_JAVA\_DATA\_PATH

**MQ\_JAVA\_DATA\_PATH** ortam değişkeni, IBM MQ classes for JMS , IBM MQ classes for Jakarta Messaging ve IBM MQ classes for Java için günlük ve izleme çıkışına ilişkin dizini belirtir. IBM MQ classes for JMS , IBM MQ classes for Jakarta Messaging ve IBM MQ classes for Java ile birlikte sağlanan komut dosyaları tarafından kullanılır.

Daha fazla bilgi için [IBM MQ classes for JMS/Jakarta Messaging ve IBM MQ classes for Java ile ilgili ortam değişkenleri için ortam değişkenlerinin ayarlanması](#) başlıklı konuya bakın.

## MQ\_JAVA\_KURULUŞ\_YOLU

**MQ\_JAVA\_INSTALL\_PATH** ortam değişkeni, JMS için IBM MQ sınıfları için kurulu olanaçinde gösterildiği gibi IBM MQ classes for JMS ve IBM MQ classes for Jakarta Messaging 'in ve IBM MQ classes for Java kurulu dizinleri içinde gösterildiği gibi IBM MQ classes for Java ' in kurulu olduğu dizini belirtir.

Daha fazla bilgi için [IBM MQ classes for JMS/Jakarta Messaging ve IBM MQ classes for Java ile ilgili ortam değişkenleri için ortam değişkenlerinin ayarlanması](#) başlıklı konuya bakın.

## MQ\_JAVA\_LIB\_YOLU

**MQ\_JAVA\_LIB\_PATH** ortam deęiřkeni, IBM MQ classes for JMS ve IBM MQ classes for Jakarta Messaging ve IBM MQ classes for Java kitaplıklarının saklandıęı dizini belirtir. IBM MQ classes for JMS ve IBM MQ classes for Jakarta Messaging ile verilen bazı komut dosyaları (örneęin, IVTRun) ya da IBM MQ classes for Java bu ortam deęiřkenini kullanır.

Daha fazla bilgi için [IBM MQ classes for JMS/Jakarta Messaging ve IBM MQ classes for Java](#) ile ilgili ortam deęiřkenleri için ortam deęiřkenlerinin ayarlanması bařlıklı konuya bakın.

## MQ\_OVERRIDE\_DATA\_PATH

IBM MQ veri yolunun varsayılan dizinini deęiřtirmek için **MQ\_OVERRIDE\_DATA\_PATH** ortam deęiřkenini kullanabilirsiniz.

## MQ\_SET\_NODELAYACK

AIX

**MQ\_SET\_NODELAYACK** ortam deęiřkeni, AIX üzerinde TCP gecikmeli onayını kapatır.

Bu ortam deęiřkenini ayarladıęınızda, TCP\_NODELAYACK seęeneęiyle iřletim sisteminin setsockopt çağrısıyla TCP gecikmeli alındı bildirimini kapatır. Bu iřlevi yalnızca AIX destekler; bu nedenle **MQ\_SET\_NODELAYACK** ortam deęiřkeninin yalnızca AIX üzerinde bir etkisi vardır.

## MQ\_USER\_NAME

Linux

Linux üzerinde kayıtlı olmayan bir kuruluřun adlandırılmamıř bir kullanıcının adını seęmesine izin vermek için **MQ\_USER\_NAME** ortam deęiřkenini kullanabilirsiniz. Bu, örneęin, OpenShift'te yayınlama/abone olma sıradüzenlerini kullanmak için gereklidir.

**MQ\_USER\_NAME** deęeri, sistemde önceden var olan bir kullanıcıyla eřleřmemelidir ve 12 bayttan küçük ya da 12 bayta eřit olmalıdır.

## MQAPI\_TRACE\_LOGFILE

Örnek API çıkıř programı, **MQAPI\_TRACE\_LOGFILE** ortam deęiřkeninde tanımlı bir önekle kullanıcı tarafından belirtilen bir dosyaya yönelik MQI izlemesi oluřturur.

Daha fazla bilgi için bkz. [API çıkıř örneę programı](#).

## MQAPPLNAME

ALW

Uygulama adı henüz seęilmediyse, kuyruk yöneticisine yönelik baęlantıyı tanımlamak için kullanılacak ad olarak **MQAPPLNAME** ortam deęiřkenini kullanabilirsiniz. Yalnızca ilk 28 karakter kullanılır ve bunların tümü boş ya da boş olmamalıdır.

Daha fazla bilgi için [Desteklenen programlama dillerinde uygulama adını kullanma](#) bařlıklı konuya bakın.

## MQCCSID

**MQCCSID** ortam deęiřkeni, kullanılacak kodlanmış karakter takımı numarasını belirler ve sunucunun konfigürasyonunun tanımlandıęı CCSID deęerini geçersiz kılar. **MQCCSID**, bir uygulamanın yerel CCSID 'sini geçersiz kılmak ve kullanılacak kodlanmış karakter takımı numarasını (örneęin, yerel CCSID desteklenmeyen bir CCSID ise ya da gerekli CCSID deęilse) belirlemek için kullanılabilir.

**MQCCSID** ayarını tanımlamak için ařaęıdaki komutlardan birini kullanın:

- **Linux** **AIX** AIX and Linux'ta:

```
export MQCCSID=number
```

- **Windows** Windows'ta:

```
SET MQCCSID=number
```

- **IBM i** IBM i'ta:

```
ADDENVVAR ENVVAR(MQCCSID) VALUE(number)
```

Ek bilgi için [İstemci ya da sunucu CCSID 'sinin seçilmesibaşlıklı konuya](#) bakın.

## MQCCDTURL

**MQCCDTURL** ortam değişkeni, **MQCHLLIB** ve **MQCHLTAB** ortam değişkenlerinin bir birleşimini ayarlamaya ilişkin eşdeğer bir yetenek sağlar. İstemci olarak bağlanan yerel programlar (C, COBOL ya da C++ uygulamaları) için istemci kanal tanımlama çizelgesinin edinilebileceği tek bir değer olarak bir dosya, ftp ya da http URL sağlamanızı sağlar.

**Not:** URL ' yi sağlamak için ortam değişkenlerinin kullanılması Java, JMS ya da yönetilen .NET uygulamaları için bir etki yaratmaz.

IBM MQ , bir dosyadan, ftp ya da http URL ' den CCDT alınmasını destekler. Ancak **MQCCDTURL** yalnızca bir URL değerini kabul eder. Varolan yerel dosya sistemi dizin biçimini kabul etmez.

Yerel bir dosyayla birlikte **MQCHLLIB** ve **MQCHLTAB** yerine **MQCCDTURL** protokolünü kullanmak için 'file://' iletişim kuralını kullanabilirsiniz. Bu nedenle, AIX ve Linux için bu örnekte gösterildiği gibi:

```
export MQCCDTURL=file:///var/mqm/qmgrs/QMGR/@ipcc/MYCHL.TAB
```

şuna eşdeğerdir:

```
export MQCHLLIB=/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLTAB=MYCHL.TAB
```

Windows için bu örnekte gösterildiği gibi bir JSON dosyası da belirtebilirsiniz:

```
set MQCCDTURL=file:/c:/mq-channels/CCDT-QMGR1.json
```

şuna eşdeğerdir:

```
set MQCHLLIB=C:\mq-channels
set MQCHLTAB=CCDT-QMGR1.json
```

Daha fazla bilgi için bkz. [CCDT ' ye URL erişimi](#).

## MQCERTLABL

**MQCERTLABL** ortam değişkeni, TLS anlaşması sırasında gönderilen kişisel sertifikayı bulmak için IBM MQ ' in kullanacağı kanal tanımlamasının sertifika etiketini tanımlar.

Daha fazla bilgi için bkz. [Sayısal sertifika etiketleri, gereksinimlerin anlaşılması](#).

## MQCERTVPOL

**MQCERTVPOL** ortam değişkeni, kullanılacak sertifika doğrulama ilkesinin tipini belirtir. Bu ortam değişkeni, istemci yapılandırma dosyasının SSL kısmına ilişkin **CertificateValPolicy** özneliğini geçersiz kılar.

**MQCERTVPOL** iki değerden birine ayarlanabilir:

## Fark Etmez

Temeldeki güvenli yuva kitaplığı tarafından desteklenen herhangi bir sertifika geçerlilik denetimi ilkesini kullanın. Bu ayar varsayılan ayardır.

## RFC5280

Yalnızca RFC 5280 standardına uyan sertifika geçerlilik denetimini kullanın.

**MQCERTVPOL** ayarını tanımlamak için aşağıdaki komutlardan birini kullanın:

- **Linux** **AIX** AIX and Linux sistemleri için:

```
export MQCERTVPOL= value
```

- **Windows** Windows sistemleri için:

```
SET MQCERTVPOL= value
```

- **IBM i** IBM i sistemleri için:

```
ADDENVVAR ENVVAR(MQCERTVPOL) VALUE(value)
```

Daha fazla bilgi için bkz. [IBM MQ içindeki sertifika doğrulama ilkeleri](#) ve [IBM MQ içinde sertifika doğrulama ilkelerini yapılandırma](#).

## MQCHLLIB

**MQCHLLIB** ortam değişkeni, istemci kanal tanımlama çizelgesini (CCDT) içeren kütüğün dizin yolunu belirtir. Dosya sunucuda yaratılır, ancak IBM MQ MQI client iş istasyonuna kopyalanabilir.

**MQCHLLIB** ayarını tanımlamak için aşağıdaki komutlardan birini kullanın:

- **Windows** Windows'ta:

```
SET MQCHLLIB=pathname
```

Örneğin:

```
SET MQCHLLIB=C:\wmqtest
```

- **Linux** **AIX** AIX and Linux sistemleri için:

```
export MQCHLLIB=pathname
```

- **IBM i** IBM için:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE(pathname)
```

**MQCHLLIB** ayarlanmazsa, istemcinin yolu varsayılan olarak şu olur:

- **Linux** **AIX** AIX and Linux işletim tarihinde: `/var/mqm/`
- **Windows** Windows işletim tarihinde: `MQ_INSTALLATION_PATH`
- **IBM i** IBM işletimlerinde: `/QIBM/UserData/mqm/`

**crtmqm** ve **strmqm** komutları için, yol varsayılan olarak iki yol kümesinin birine ayarlanır. *datapath* ayarlanırsa, yol varsayılan olarak ilk kümenin birine ayarlanır. *datapath* ayarlanmazsa, yol varsayılan olarak ikinci kümenin birine ayarlanır.

- **Linux** **AIX** AIX and Linux işletim tarihinde: *datapath/@ipcc*
- **Windows** Windows işletim tarihinde: *datapath\@ipcc*
- **IBM i** IBM işletim tarihinde: *datapath/&ipcc*

Ya da:

- **Linux** **AIX** AIX and Linux işletim tarihinde: */prefix/qmgrs/qmgrname/@ipcc*
- **Windows** Windows işletim tarihinde: *MQ\_INSTALLATION\_PATH\data\qmgrs\qmgrname\@ipcc*
- **IBM i** IBM işletim tarihinde: */prefix/qmgrs/qmgrname/&ipcc*

Burada:

- *MQ\_INSTALLATION\_PATH* , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.
- Varsa, *datapath* , kuyruk yöneticisi kıtasında tanımlanan DataPath değeridir.
- *prefix* , kuyruk yöneticisi kıtasında tanımlanan Önek değeridir. Önek genellikle aşağıdaki değerlerden biridir:
  - **Linux** **AIX** AIX and Linux sistemlerinde */var/mqm* .
  - **IBM i** */QIBM/UserData/mqm/* açık IBM i.
- *qmgrname* , kuyruk yöneticisi kıtasında tanımlanan Directory özniteliğinin değeridir. Değer, gerçek kuyruk yöneticisi adından farklı olabilir. Değer, özel karakterlerin yerine geçecek şekilde değiştirilmiş olabilir.
- Kuyruk yöneticisinin tanımlandığı yer, altyapıya bağlıdır:
  - **Linux** **IBM i** **AIX** IBM i, AIX and Linux üzerindeki *mqsc.ini* dosyasında.
  - **Windows** Windows üzerindeki kayıta.

**Notlar:**

1. **z/OS** Sunucunuz olarak IBM MQ for z/OS kullanıyorsanız, dosyanın IBM MQ istemci iş istasyonunda tutulması gerekir.
2. Ayarlanırsa, *MQCHLLIB CCDT* ' yi bulmak için kullanılan yolu geçersiz kılar.
3. *MQCHLLIB* , *MQCHLTAB* ortam değişkeniyle birlikte çalışan bir URL içerebilir (bkz. “*CCDT* ' ye URL erişimi” sayfa 52).
4. *MQCHLLIB* gibi ortam değişkenlerinin kapsamı, platforma özgü bir şekilde bir süreç ya da iş ya da sistem çapında olabilir.
5. *MQCHLLIB* sistem genişliğinde bir sunucuyu ayarlarsanız, sunucudaki tüm kuyruk yöneticileri için aynı yolu *CCDT* dosyasına ayarlar. *MQCHLLIB* ortam değişkenini ayarlamazsanız, yol her kuyruk yöneticisi için farklıdır. Kuyruk yöneticileri, ayarlanmışsa, **crtmqm** ya da **strmqm** komutunda *MQCHLLIB* değerini okur.
6. Bir sunucuda birden çok kuyruk yöneticisi yaratırsanız, ayırım aşağıdaki nedenden ötürü önemlidir. Sistem genelinde *MQCHLLIB* ayarını tanımlarsanız, her kuyruk yöneticisi aynı *CCDT* dosyasını günceller. Dosya, sunucudaki tüm kuyruk yöneticilerinden gelen istemci bağlantısı tanımlamalarını içerir. Aynı tanımlama birden çok kuyruk yöneticinde varsa, örneğin *SYSTEM.DEF.CLNTCONN* , dosya en son tanımlamayı içerir. Bir kuyruk yöneticisi yarattığınızda, *MQCHLLIB* ayarlanırsa, *CCDT* ' de *SYSTEM.DEF.CLNTCONN* güncellenir. Güncelleme, farklı bir kuyruk yöneticisi tarafından yaratılan *SYSTEM.DEF.CLNTCONN* ' in üzerine yazılır. Önceki tanımlamayı değiştirdiyse, değişiklikleriniz kaybolur.

Bu nedenle, **MQCHLLIB** ayarını sunucuda sistem çapında bir ortam değişkeni olarak belirlemeniz için alternatifler bulmanız gerekir.

7. Bir istemci-bağlantı tanımlamasındaki MQSC ve PCF NOREPLACE seçeneği CCDT dosyasının içeriğini denetlemez. NOREPLACE seçeneğinden bağımsız olarak, önceden yaratılmış, ancak bu kuyruk yöneticisi tarafından yaratılmamış aynı ada sahip bir istemci bağlantısı kanal tanımlaması değiştirilir. Tanımlama daha önce aynı kuyruk yöneticisi tarafından yaratıldıysa, tanımlama başkasıyla değiştirilmez.
8. **rcrmqobj** -t clchltab komutu CCDT dosyasını siler ve yeniden yaratır. Dosya, yalnızca komutun çalıştırıldığı kuyruk yöneticisinde yaratılan istemci bağlantısı tanımlamalarıyla yeniden yaratılır.
9. CCDT 'yi güncelleyen diğer komutlar yalnızca aynı kanal adına sahip istemci bağlantısı kanallarını değiştirir. Dosyadaki diğer istemci-bağlantı kanalları değiştirilmez.
10. **MQCHLLIB** yolu için tırnak işareti gerekmez.

Daha fazla bilgi için bkz. [CCDT 'ye ilişkin yerler](#), [CCDT' ye URL erişimive İstemci uygulamalarının ortam değişkenlerini kullanarak kuyruk yöneticilerine bağlanması](#).

## MQCHLTAB

**MQCHLTAB** ortam değişkeni, istemci kanal tanımlama çizelgesini (CCDT) içeren kütüğün adını belirtir. Varsayılan dosya adı AMQCLCHL .TAB' dir.

**MQCHLTAB** ayarını tanımlamak için aşağıdaki komutlardan birini kullanın:

- **Linux** **AIX** AIX and Linux'ta:

```
export MQCHLTAB=filename
```

- **Windows** Windows'ta:

```
SET MQCHLTAB=filename
```

- **IBM i** IBM i'ta:

```
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(filename)
```

Örneğin:

```
SET MQCHLTAB=ccdf1.tab
```

İstemciyle aynı şekilde, sunucudaki **MQCHLTAB** ortam değişkeni istemci kanal tanımlama çizelgesinin adını belirtir.

Daha fazla bilgi için bkz. [CCDT 'ye ilişkin yerler](#), [CCDT' ye URL erişimive İstemci uygulamalarının ortam değişkenlerini kullanarak kuyruk yöneticilerine bağlanması](#).

## MQCLNTCF

**MQCLNTCF** ortam değişkeni, IBM MQ MQI client yapılandırma kütüğünün yerini belirtir. Bu dosya, IBM MQ MQI client tarafından kullanılan yapılandırma bilgilerini içerir.

mqclient.ini dosyasının dosya yolunu değiştirmek için **MQCLNTCF** ortam değişkenini kullanabilirsiniz.

Bu ortam değişkeninin biçimi tam URL ' dir. Bu, dosya adının mqclient.iniolması gerekmediği ve dosyanın ağa bağlı bir dosya sistemine yerleştirilmesini kolaylaştırdığı anlamına gelir. Daha fazla bilgi için bkz. [IBM MQ MQI istemcisi yapılandırma dosyası](#), [mqclient.ini](#) ve [İstemci yapılandırma dosyasının konumu](#).

## MQDOTNET\_TRACE\_ON

**MQDOTNET\_TRACE\_ON** ortam değişkeni, yeniden dağıtılabilen IBM MQ .NET istemcileri için izlemeyi etkinleştirmek üzere kullanılır. 0 'a eşit ve 0 'dan küçük değerler izlemeyi etkinleştirmez, 1 varsayılan izlemeyi etkinleştirir ve 1 'den büyük değerler ayrıntı izlemeyi etkinleştirir.

Daha fazla bilgi için [IBM MQ .NET uygulamalarını izleme](#) ve [Ortam değişkenlerini kullanarak IBM MQ .NET uygulamalarını izleme](#) başlıklı konuya bakın.

## MQIPADDRV

**MQIPADDRV** ortam değişkeni, bir kanal bağlantısı için hangi IP protokolünün kullanılacağını belirler. Olası "MQIPADDR\_IPV4" ya da "MQIPADDR\_IPV6" dizgi değerlerini içeriyor. Bu değerler, **ALTER QMGR IPADDRV** içindeki IPv4 ve IPv6 ile istemci yapılandırma dosyasının TCP ' nin **IPAddressVersion** özniteliği ile aynı anlamlara sahiptir. Ortam değişkeni ayarlanmazsa, "MQIPADDR\_IPV4" varsayılır.

**MQIPADDRV** ayarını tanımlamak için aşağıdaki komutlardan birini kullanın:

- Linux AIX AIX and Linux'ta:

```
export MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6" />
```

- Windows Windows'ta:

```
SET MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
```

- IBM i IBM i'ta:

```
ADDENVVAR ENVVAR(MQIPADDRV) VALUE(MQIPADDR_IPV4|MQIPADDR_IPV6)
```

## MQKEYRPWD

V 9.3.0 V 9.3.0

**MQKEYRPWD** ortam değişkenini ayarladığınızda, kullanıcıya ait sayısal sertifikayı tutan anahtar havuzunun parolasını belirtir. **MQKEYRPWD** kullanılırsa, ortam değişkeninin değerini ayarlamadan önce parolayı şifrelemelisiniz.

**MQKEYRPWD** ' yi ayarlamak için aşağıdaki komutlardan birini kullanın:

- Linux AIX AIX and Linux sistemlerinde:

```
export MQKEYRPWD=passphrase
```

- Windows Windows sistemlerinde:

```
SET MQKEYRPWD=passphrase
```

- IBM i IBM i'ta:

```
ADDENVVAR ENVVAR(MQKEYRPWD) VALUE(passphrase)
```

Bu ortam değişkeni için varsayılan değer yok.

V 9.3.0 V 9.3.0 Daha fazla bilgi için bkz.

- ALW AIX, Linux, and Windows üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması ve Anahtar havuzu parolasını şifreleme

- **IBM i** IBM i üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması ve Anahtar havuzu parolasını şifreleme.

## MQLICENSE

Linux

Linux sistemlerinde, ürünü kurduktan sonra bir IBM MQ lisansını kabul etmek ya da görüntülemek için **MQLICENSE** ortam değişkenini kullanabilirsiniz.

Bunu neden yapmak istediğinize ya da yapmanız gerekebileceğine ilişkin daha fazla bilgi için bkz. Linux için lisansı kabul etme IBM MQ

**MQLICENSE** ortam değişkeni iki değerden birine ayarlanabilir:

### kabul et

Kuruluş sonrası lisansı kabul edin.

### görünüm

Lisans kabul edildiyse, lisansı görüntüleyin.

Lisans kuruluştan sonra kabul etmek için şu komutu kullanın:

```
export MQLICENSE=accept
```

Lisansı görüntülemek için şu komutu kullanın:

```
export MQLICENSE=view
```

**Not:** Lisansı kabul etmek ve görüntülemek için aşağıdaki komutları da kullanabilirsiniz:

- mqlicense (kuruluştan sonra lisans kabul et)
- dspmqlc ( IBM MQ lisansını görüntüle)

## MQMAXERRORLOGSIZE

Multi

**MQMAXERRORLOGSIZE** ortam değişkeni, yedeklemeye kopyalanan kuyruk yöneticisi hata günlüğünün boyutunu belirtir.

Daha fazla bilgi için bkz. Hata günlüklerini kullanma.

## MQNAME

Windows

**MQNAME** ortam değişkeni, IBM MQ işlemlerinin kullanabileceği yerel NetBIOS adını belirtir. NetBIOS bağlantısı yalnızca Windows çalıştıran bir istemci ve sunucu için geçerlidir.

**MQNAME** ayarını tanımlamak için şu komutu kullanın:

```
SET MQNAME=Your_env_Name
```

Örneğin:

```
SET MQNAME=CLIENT1
```

Bazı NetBIOS uygulamaları, IBM MQ MQI client üzerinde eşzamanlı olarak birden çok IBM MQ uygulaması çalıştırıyorsanız, her uygulama için **MQNAME** tarafından ayarlanan benzersiz bir ad gerektirir.

Daha fazla bilgi için IBM MQ yerel NetBIOS adını tanımlamabaşlıklı konuya bakın.



## MQNOREMPPOOL

**MQNOREMPPOOL** ortam deęişkenini ayarladığınızda, kanal havuzlamayı kapatır ve kanalların dinleyicinin iş parçacıkları olarak çalışmasına neden olur.

Daha fazla bilgi için bkz. [MCATYPE \(Message channel agent type\)](#).

## MQPSE\_TRACE\_LOGFILE

Bir yayını aboneye teslim edilmeden önce engellemek için çıkışın örnek bir C programı olan Yayınlama Çıkışı Örnek Programı AMQSPSE0' nı çalıştırırken **MQPSE\_TRACE\_LOGFILE** ortam deęişkenini kullanırsınız. İzlenecek uygulama işleminde, bu ortam deęişkeni izleme dosyalarının nereye yazılması gerektiğini açıklar.

Daha fazla bilgi için bkz. [Yayınlama Çıkışı örnek programı](#).

## MQS\_AMSCRED\_KEYFILE

**MQS\_AMSCRED\_KEYFILE** ortam deęişkenini kullanarak, IBM MQ Advanced Message Security (AMS) uygulamalarının çalıştırma zamanında ya da **runamscred** komutunu kullanarak bir anahtar deposu yapılandırma dosyasını korurken kullanılacak ilk anahtar dosyasını geçersiz kılabilir ya da sağlayabilirsiniz.

Daha fazla bilgi için bkz. [Anahtar depolarının ve sertifikaların AMS ile kullanılması ve IBM MQ bileşeni yapılandırma dosyalarındaki parolaları koruma](#).

## MQS\_DISABLE\_ALL\_INTERCEPT

Ürünün önceki bir sürümünden bir kuyruk yöneticisine bağlanmaya çalışırken 2085 (MQRC\_UNKNOWN\_OBJECT\_NAME) hatası raporlandıysa ve IBM MQ ' u yerel C istemcileriyle kullanıyorsanız, **MQS\_DISABLE\_ALL\_INTERCEPT** ortam deęişkenini devre dışı bırakmak için kullanabilirsiniz IBM MQ Advanced Message Security (AMS).

**Not:** **MQS\_DISABLE\_ALL\_INTERCEPT** ortam deęişkenini yalnızca C istemcileri için kullanabilirsiniz. Java istemcileri için bunun yerine **AMQ\_DISABLE\_CLIENT\_AMS** ortam deęişkenini kullanmanız gerekir.

Daha fazla bilgi için bkz. [İstemcide Gelişmiş İleti Güvenliğinin Devre Dışı Bırakılması](#).

## MQS\_IPC\_HOST

IPC dosya sistemi nesnelerinin sistem tarafından ayırt edilmesi gerektiğinden, kuyruk yöneticisinin üzerinde çalıştığı her sisteme ilişkin bir altdizin, dizin yoluna eklenir. Anasistem adının oluşturulan değeri bir sorun yaratırsa, **MQS\_IPC\_HOST** ortam deęişkenini kullanarak anasistem adını ayarlayabilirsiniz.

Daha fazla bilgi için bakınız: [Sharing IBM MQ files on Multiplatforms](#).

## MQS\_KEYSTORE\_CONF

**MQS\_KEYSTORE\_CONF** ortam deęişkeni, dosya *home\_directory/.mq/keystore.conf* varsayılan konumunda değilse, IBM MQ Advanced Message Security (AMS) için anahtar deposu yapılanış kütüğünün yerini belirtir.

Daha fazla bilgi için bkz. [AMS ile anahtar depolarının ve sertifikaların kullanılması](#).

Managed File Transfer ile ilgili sorunlarınız varsa, [MFT AMS ' ye ilişkin anahtar deposu özelliklerini okumadığınızda sorun giderme başlıklı konuya bakın](#).

## MQS\_MQI\_KEYFILE



**MQS\_MQI\_KEYFILE** ortam deęişkenini ayarladığınızda, parola koruma işlemleri için kullanılacak ilk anahtarı içeren ilk anahtar dosyasının konumunu belirtir. İlk anahtar dosyası belirtilmezse, varsayılan ilk anahtar IBM MQ parola koruma sistemi tarafından kullanılır.

**MQS\_MQI\_KEYFILE** ayarını tanımlamak için aşağıdaki komutlardan birini kullanın:

- Linux AIX AIX and Linux sistemlerinde:

```
export MQS_MQI_KEYFILE=key file location
```

- Windows Windows sistemlerinde:

```
SET MQS_MQI_KEYFILE=key file location
```

- IBM i IBM i'ta:

```
ADDENVVAR ENVVAR(MQS_MQI_KEYFILE) VALUE(key file location)
```

V9.3.0 V9.3.0 Daha fazla bilgi için bkz. [AIX, Linux, and Windows üzerinde IBM MQ MQI client için başlangıç anahtarı sağlanması](#) ve [IBM i üzerinde IBM MQ MQI client için başlangıç anahtarı sağlanması](#).

## MQS\_SSLCRYP\_KEYFILE

V9.3.0

**MQS\_SSLCRYP\_KEYFILE** ortam değişkeni, parolayı PKCS #11 şifreleme donanımı yapılandırma dizisinde şifrelemek için kullanılan ilk anahtarı içeren dosyanın tam yolunu ve adını belirtmenin alternatif bir yoludur; bunun yerine, `qm.ini`'nin SSL kısmı 'ndaki **SSLCryptoHardwareKeyFile** özniteliğiyle belirtmeniz gerekir. **MQS\_SSLCRYP\_KEYFILE**, `qm.ini` dosyasından daha yüksek bir önceliğe sahiptir, bu nedenle değeri diğer herhangi bir değerden önceliklidir. Daha fazla bilgi için bkz. [IBM MQ Client using cryptographic hardware](#).

## MQS\_TRACE\_OPTIONS

AIX

AIX üzerinde seçmeli bileşen izlemesi için, **MQS\_TRACE\_OPTIONS** ortam değişkenini kullanarak yüksek ayrıntı ve parametre izleme işlevlerini tek tek etkinleştirin.

**Not:** **MQS\_TRACE\_OPTIONS** ortam değişkenini yalnızca IBM Destek tarafından yapmanız istendiye ayarlayın.

Daha fazla bilgi için bkz. [AIX and Linux üzerinde izleme](#).

## MQSERVER

**MQSERVER** ortam değişkeni, en küçük bir kanal tanımlamak için kullanılır. **MQSERVER**, IBM MQ sunucusunun yerini ve kullanılacak iletişim yöntemini belirtir.

**Not:** TLS kanalı ya da kanal çıkışları olan bir kanal tanımlamak için **MQSERVER** kullanamazsınız. TLS kanalının nasıl tanımlanacağı hakkında daha fazla bilgi için [TLS ile kanalları korumabaşlıklı konuya](#) bakın.

Aşağıdaki örneklerde **MQSERVER** ayarının nasıl ayarlandığı gösterilmektedir:

- Linux AIX AIX and Linux'ta:

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
```

- Windows Windows'ta:

```
SET MQSERVER=SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)
```

- **IBM i** IBM i'ta:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)')
```

#### Not:

- Kanal adını, iletim tipini ve bağlantı adını ayırmak için bu karakter kullanıldığından kanal adı eğik çizgi (/) karakterini içeremez. Bir istemci kanalı tanımlamak için **MQSERVER** ortam değişkeni kullanıldığında, ileti uzunluğu üst sınırı (MAXMSGL) 100 MB kullanılır. Bu nedenle, kanal için geçerli olan ileti büyüklüğü üst sınırı, sunucudaki SVRCONN kanalında belirtilen değerdir.
- İletim tipi, IBM MQ istemci altyapınıza bağlı olarak LU62 , TCP , NETBIOS, SPX olabilir.
- Bağlantı adı tam olarak nitelenmiş bir ağ adı olmalıdır., örneğin, AMACHINE . ACOMPANY . COM(1414) .
- Bağlantı adı, bağlantı adlarının virgülle ayrılmış bir listesi olabilir. Listedeki bağlantı adları, istemci bağlantı çizelgesindeki birden çok bağlantıya benzer şekilde kullanılır. Bağlantı adı listesi, istemcinin denemesi için birden çok bağlantı belirtmek üzere kuyruk yöneticisi gruplarına alternatif olarak kullanılabilir. Çok eşgörünümlü bir kuyruk yöneticisi yapılandırıyorsanız, farklı kuyruk yöneticisi yönetim ortamları belirtmek için bir bağlantı adı listesi kullanabilirsiniz.

IBM MQ MQI client makineniz ile bir sunucu makinesi arasındaki kanalı tanımlamak için **MQSERVER** ortam değişkenini kullanırsanız, bu kanal uygulamanızın kullanabileceği tek kanaldır ve istemci kanal tanımlama çizelgesine (CCDT) başvuru yapılmaz.

Daha fazla bilgi için bakınız: [Creating a client-connection channel on IBM MQ MQI client using MQSERVER.](#)

## MQSNOAUT



**Uyarı:** Bu işlev önerilmez.

**MQSNOAUT** ortam değişkenini herhangi bir değere ayarladığınızda, nesne yetki yöneticisi (OAM) devre dışı bırakılır ve güvenlik denetimi önlenir. Bu, bir test ortamı için uygun olabilir. Bu, hem yetkilendirme hem de bağlantı kimlik doğrulama işlevselliğini içerir. TLS, Kanal Kimlik Doğrulama Kayıtları ve Güvenlik Çıkışları etkilenmez.

**MQSNOAUT** ortam değişkeni yalnızca bir kuyruk yöneticisi yaratıldığında geçerli olur.



**Uyarı:** OAM ' yi etkinleştirmek için kuyruk yöneticisini silmeniz, ortam değişkenini silmeniz ve ardından **MQSNOAUT** belirtmeden kuyruk yöneticisini yeniden yaratmanız gerekir.

Daha fazla bilgi için [AIX, Linux ve Windows sistemlerinde güvenlik erişimi denetimlerinin engellenmesi başlıklı konuya](#) bakın.

## MQSPREFIX

Varsayılan önceki değıştirmenin bir alternatifi olarak, **MQSPREFIX** ortam değışkenini kullanarak **crtmqm** komutuna ilişkin **DefaultPrefix** değeri geçersiz kılabilirsiniz.

Daha fazla bilgi için bkz. [IBM MQ dosya adları](#) ve **mq.s.ini** dosyasının [AllQueueYöneticileri](#) bölümü.

## MQSSLCRYP

**ALW**

**MQSSLCRYP** ortam değışkeni, sistemde var olan şifreleme donanımını yapılandırmak için kullanabileceğiniz bir parametre dizisini içerir.

**V 9.3.0**

İzin verilen değerler, istemci yapılandırma dosyasının SSL kısmına ilişkin [SSLCryptoHardware](#) alanıyla aynıdır.

**MQSSLCRYP** parametresini tanımlamak için aşağıdaki komutlardan birini kullanın:

- **Linux** **AIX** AIX and Linux sistemlerinde:

```
export MQSSLCRYP=string
```

- **Windows** Windows sistemlerinde:

```
SET MQSSLCRYP=string
```

Daha fazla bilgi için bkz. [IBM MQ bileşeni yapılandırma dosyalarındaki parolaları korumadaki AIX, Linux, and Windows üzerinde şifreleme donanımı için yapılandırma ve Şifreleme donanımını kullanan IBM MQ clients](#) .

## MQSSLFIPS

**MQSSLFIPS** ortam değişkeni, şifreleme IBM MQ içinde gerçekleştirilirse yalnızca FIPS onaylı algoritmaların kullanılıp kullanılmayacağını belirtir. Bu ortam değişkenini YES ya da NO olarak ayarlayabilirsiniz. Varsayılan değer NO' dur. Bu değerler, [ALTER QMGR](#) komutunun **SSLFIPS** parametresiyle aynıdır.

**MQSSLFIPS** ayarını tanımlamak için aşağıdaki komutlardan birini kullanın:

- **Linux** **AIX** AIX and Linux sistemlerinde:

```
export MQSSLFIPS=YES|NO
```

- **Windows** Windows sistemlerinde:

```
SET MQSSLFIPS=YES|NO
```

- **IBM i** IBM i'ta:

```
ADDENVVAR ENVVAR(MQSSLFIPS) VALUE(YES|NO)
```

FIPS sertifikalı algoritmaların kullanımı, şifreleme donanımının kullanımından etkilenir. Daha fazla bilgi için [MQI istemcisinde çalıştırma zamanında yalnızca FIPS onaylı CipherSpecs ' in kullanılmasının belirtilmesibaşlıklı konuya bakın.](#)

## MQSSLKEYR

**MQSSLKEYR** ortam değişkeni, kullanıcıya ait sayısal sertifikayı bulduran anahtar havuzunun yeriniuzantısı olmadan içerdiği anlamına gelir.

**V9.3.0** **V9.3.0** Anahtar havuzunun tam yolunu ve dosya adını belirtin. Dosya soneki belirtilmezse, bunun .kdbolduğu varsayılır.

**MQSSLKEYR** ayarını tanımlamak için aşağıdaki komutlardan birini kullanın:

- **Linux** **AIX** AIX and Linux sistemlerinde:

```
export MQSSLKEYR=pathname
```

- **Windows** Windows sistemlerinde:

```
SET MQSSLKEYR=pathname
```

- **IBM i** IBM i'ta:

```
ADDENVVAR ENVVAR(MQSSLKEYR) VALUE(pathname)
```

Bu ortam deęişkeni için varsayılan deęer yok.

Daha fazla bilgi için, **ALTER QMGR** komutunun **SSLKEYR** parametresine bakın.

## MQSSLPROXY

**MQSSLPROXY** ortam deęişkeni, OCSP denetimleri için GSKit tarafından kullanılacak HTTP yetkili sunucusunun anasistem adını ve kapı numarasını belirtir.

**MQSSLPROXY** ayarını tanımlamak için aşağıdaki komutlardan birini kullanın:

- **Linux** **AIX** AIX and Linux sistemlerinde:

```
export MQSSLPROXY="string"
```

- **Windows** Windows sistemlerinde:

```
SET MQSSLPROXY= string
```

**MQSSLPROXY** ile belirttięiniz dizgi, OCSP denetimleri için GSKit tarafından kullanılacak HTTP Yetkili sunucusunun anasistem adı ya da aę adresi olabilir. Bu adresi parantez içine alınmış isteęe baęlı bir kapı numarası izleyebilir. Kapı numarasını belirtmezseniz, varsayılan HTTP kapısı olan 80 kullanılır.

**Linux** **AIX** Örneęin, AIX and Linux sistemlerinde aşağıdaki komutlardan birini kullanabilirsiniz:

- ```
export MQSSLPROXY="proxy.example.com(80)"
```
- ```
export MQSSLPROXY="127.0.0.1"
```

Daha fazla bilgi için bkz. [Online Certificate Status Protocol \(OCSP\) ile çalıřma](#).

## MQSSLRESET

**MQSSLRESET** ortam deęişkeni, TLS gizli anahtarı yeniden anlařılmadan önce bir TLS kanalında gönderilen ve alınan řifrelenmemiş bayt sayısını belirtir. 0-999 999 999 999 aralıęında bir tamsayıya ayarlanabilir. Varsayılan deęer, gizli anahtarların hiçbir zaman yeniden anlařılmadıęını gösteren 0 'dır. 1-32 KB aralıęında bir TLS gizli anahtar sıfırlama sayısı belirtirseniz, TLS kanalları 32 KB gizli anahtar sıfırlama sayısını kullanır. Bu gizli sıfırlama sayısı, küçük TLS gizli anahtar sıfırlama deęerleri için ortaya çıkabilecek çok fazla anahtar sıfırlamasını önlemektir.

**MQSSLRESET** ayarını tanımlamak için aşağıdaki komutlardan birini kullanın:

- **Linux** **AIX** AIX and Linux sistemlerinde:

```
export MQSSLRESET=integer
```

- **Windows** Windows sistemlerinde:

```
SET MQSSLRESET=integer
```

- **IBM i** IBM i'ta:

```
ADDENVVAR ENVVAR(MQSSLRESET) VALUE(integer)
```

Daha fazla bilgi için [SSL ve TLS gizli anahtarlarını sıfırlamabaşlıklı konuya](#) bakın.

## MQSUIEB

**ALW**

IBM MQ ürününü, AIX, Linux, and Windows platformlarında NSA Suite B standardına uygun çalışacak şekilde yapılandırabilirsiniz.

**MQSUIEB** ortam değişkeni, Suite B uyumlu şifrelemenin kullanılıp kullanılmayacağını belirtir. Suite B şifrelemesi kullanılacaksa, **MQSUIEB** değerini aşağıdakilerden birine ayarlayarak şifreleme gücünü belirtebilirsiniz:

- YOK
- 128\_BIT, 192\_BIT
- 128\_BIT
- 192\_BIT

Virgülle ayrılmış bir liste kullanarak birden çok değer belirtebilirsiniz. NONE değerinin başka bir değerle kullanılması geçersizdir.

Daha fazla bilgi için [Suite B için IBM MQ 'yi yapılandırmabaşlıklı konuya](#) bakın.

## MQTCPTIMEOUT

**MQTCPTIMEOUT** ortam değişkeni, IBM MQ 'un TCP bağlantı çağrısı için ne kadar süre beklediğini belirtir.

## ODQ\_MSG

**runmqdlq**' dan farklı bir teslim mektubu kuyruk işleyicisi kullanıyorsanız, örneğin kaynağı olan amqsd1q, temel olarak kullanabilirsiniz. Örnek, ürün içinde sağlanan ölü harf işleyicisi gibidir, ancak izleme ve hata raporlaması farklıdır. Hata ve bilgi iletileri içeren dosyanın adını ayarlamak için **ODQ\_MSG** ortam değişkenini kullanın. Sağlanan dosyanın adı amqsd1q.msg.

Daha fazla bilgi için bkz. [Ölü mektup kuyruğu işleyicisi örneği](#).

## ODQ\_İZLEME

**runmqdlq**' dan farklı bir teslim mektubu kuyruk işleyicisi kullanıyorsanız, örneğin kaynağı olan amqsd1q, temel olarak kullanabilirsiniz. Örnek, ürün içinde sağlanan ölü harf işleyicisi gibidir, ancak izleme ve hata raporlaması farklıdır. İzlemeyi etkinleştirmek için **ODQ\_TRACE** ortam değişkenini YES ya da yesolarak ayarlayın.

Daha fazla bilgi için bkz. [Ölü mektup kuyruğu işleyicisi örneği](#).

## WCF\_TRACE\_ON

WCF özel kanalı için iki farklı izleme yöntemi vardır. Bu iki izleme yöntemi bağımsız olarak ya da birlikte etkinleştirilir. Her yöntem kendi izleme dosyasını üretir; bu nedenle, her iki izleme yöntemi de etkinleştirildiğinde, iki izleme çıkış dosyası oluşturulur. İki farklı izleme yöntemini etkinleştirmek ve devre dışı bırakmak için dört birleşim vardır. WCF izlemesini etkinleştirmek için bu birleşimlerin yanı sıra, XMS .NET izlemesi **WCF\_TRACE\_ON** ortam değişkeni kullanılarak etkinleştirilebilir.

Daha fazla bilgi için bakınız: [Tracing the WCF custom channel for IBM MQ](#).

## WMQSOAP\_HOME

**WMQSOAP\_HOME** ortam değişkeni, .NET SOAP over JMS hizmet barındırma ortamı IBM MQ'inde doğru kurulduktan ve yapılandırıldıktan sonra ek yapılandırma adımları tamamlanırken kullanılır. Yerel bir kuyruk yöneticisinden erişilebilir.

Daha fazla bilgi için bkz. [WCF Client to a .NET service hosted by IBM MQ sample](#) ve [WCF client to a Axis Java hizmeti IBM MQ sample](#) tarafından barındırılır.

## XMS\_TRACE\_ON, XMS\_TRACE\_FILE\_PATH, XMS\_TRACE\_FORMAT ve XMS\_TRACE\_SPECIFICATION

IBM MQ classes for XMS .NET Framework kullanıyorsanız, XMS ortam değişkenlerinin yanı sıra bir uygulama yapılandırma dosyasından da izleme yapılandırabilirsiniz. IBM MQ classes for XMS .NET (.NET Standard ve .NET 6 kitaplıkları) kullanıyorsanız, XMS ortam değişkenlerinden izlemeyi yapılandırmanız gerekir. İzleme olağan durumda IBM Destek rehberliğinde kullanılır.

Bir XMS .NET uygulaması için izlemeyi etkinleştirmek ve yapılandırmak üzere uygulamayı çalıştırmadan önce aşağıdaki ortam değişkenlerini ayarlayın:

### XMS\_TRACE\_ON

**XMS\_TRACE\_ON** ortam değişkeni ayarlanırsa, varsayılan olarak tüm izleme etkinleştirilir.

### XMS\_TRACE\_DOSYA\_YOLU

**XMS\_TRACE\_FILE\_PATH** ortam değişkeni, bu kayıtların yürürlükteki çalışma dizininden başka bir yere yazılmasını istiyorsanız, izleme ve FFDC kayıtlarının yazıldığı dizinin tam olarak nitelenmiş yol adını belirtir.

### XMS\_TRACE\_FORMAT

**XMS\_TRACE\_FORMAT** ortam değişkeni, gerekli izleme biçimini belirtir; bu biçim BASIC ya da ADVANCED olabilir.

### XMS\_TRACE\_SPECIFICATION

**XMS\_TRACE\_SPECIFICATION** ortam değişkeni, bir uygulama yapılandırma kütüğünün İzleme kısmından tanımlanan izleme ayarlarını geçersiz kılar. **XMS\_TRACE\_SPECIFICATION** yalnızca IBM MQ classes for XMS .NET Framework için geçerlidir.

Daha fazla bilgi için bkz. [XMS .NET uygulamalarını izleme](#) ve [XMS .NET uygulamalarının XMS ortam değişkenlerini kullanarak izlenmesi](#).

## Multi Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerini değiştirme

Yapılandırma (.ini) dosyalarındaki bilgileri düzenleyerek, IBM MQ ya da tek bir kuyruk yöneticisinin davranışını kuruluşunuzun gereksinimlerine uyacak şekilde değiştirebilirsiniz. IBM MQ MQI clients için yapılandırma seçeneklerini de değiştirebilirsiniz.

### Bu görev hakkında

IBM MQ' u yöneten bir yapılandırma öznitelikleri kümesinde (ya da değiştiricilerinde) belirtilen değerleri değiştirerek, düğüm ya da kuyruk yöneticisi düzeyinde IBM MQ yapılandırma bilgilerini değiştirebilirsiniz.

Bir yapılandırma kütüğü (ya da kısım kütüğü), .ini kütüğünde ortak bir işlevi olan ya da sistemin günlük işlevleri, kanal işlevleri ve kurulabilir hizmetler gibi bir kısmını tanımlayan satır grupları olan bir ya da daha çok kısım içerir. Aşağıdaki yapılandırma dosyaları içinde IBM MQ yapılandırma özniteliklerini değiştirebilirsiniz:


### IBM MQ yapılandırma dosyası, mqs.ini

mqs.ini dosyası, düğümdeki değişiklikler bir bütün olarak etkiler. Her IBM MQ kuruluşu için bir mqs.ini dosyası vardır.

IBM MQ yapılandırma kütüğü kuyruk yöneticileriyle ilişkili verileri bulmak için kullanıldığından, var olmayan ya da yanlış bir yapılandırma kütüğü bazı ya da tüm MQSC komutlarının başarısız olmasına neden

olabilir. Ayrıca, uygulamalar IBM MQ yapılandırma dosyasında tanımlı olmayan bir kuyruk yöneticisine bağlanamaz.

### Kuruluş yapılandırma dosyası, mqinst.ini

 AIX and Linux sistemlerinde kuruluş yapılandırma dosyası mqinst.ini, tüm IBM MQ kuruluşlarına ilişkin bilgileri içerir. Biçimi düzeltilmediği ve değiştirildiği için mqinst.ini dosyası doğrudan düzenlenmemeli ya da dosyaya başvurulmamalıdır. Bunun yerine, komutları kullanarak düzenlemeniz gerekir.

### Kuyruk yöneticisi yapılandırma dosyası, qm.ini

qm.ini dosya etkileri belirli kuyruk yöneticileri için değişir. Düğümdeki her kuyruk yöneticisi için bir qm.ini dosyası vardır.

### IBM MQ MQI client yapılandırma dosyası, mqclient.ini

IBM MQ MQI clients için yapılandırma seçenekleri, genellikle mqclient.iniolarak adlandırılan istemci yapılandırma dosyasında ayrı olarak tutulur.

### Etkinlik izleme yapılandırma dosyası, mqat.ini


mqat.ini dosyası, etkinlik izleme davranışını yapılandırmak için kullanılır.

Aşağıdaki gibi bir yapılandırma dosyasını düzenlemeniz gerekebilir:

- Bir yapılandırma dosyasını kaybedersiniz. (Yapabilirsiniz yedekten kurtarın.)
- Bir ya da daha çok kuyruk yöneticisini yeni bir dizine taşımanız gerekir.
- Varsayılan kuyruk yöneticinizi değiştirmeniz gerekir. Var olan kuyruk yöneticisini yanlışlıkla silerseniz bu durum oluşabilir.
- Bunu IBM Destek tarafından yapmanız önerilir.

**Önemli:** Bir yapılandırma dosyasında yaptığınız değişiklikler genellikle, kuyruk yöneticisi yeniden başlatılıncaya kadar yürürlüğe girmez.

### Yapılandırma dosyalarının düzenlenmesiyle ilgili not almak için işaret eder:

- Bir yapılandırma dosyasının özniteliklerinin değerleri aşağıdaki önceliklere göre ayarlanır:
  - Komut satırında girilen parametreler, yapılandırma dosyalarında tanımlanan değerlerden önceliklidir.
  - qm.ini dosyalarında tanımlanan değerler, mqs.ini dosyasında tanımlanan değerlerden önceliklidir.
- Kuruludan sonra, IBM MQ yapılandırma dosyalarındaki varsayılan değerleri düzenleyebilirsiniz.
- Bir kuyruk yöneticisini yedeklerken, hem yapılandırma kütüğünü (qm.ini), hem de merkezi IBM MQ yapılandırma kütüğünü (mqs.ini) içermeyi unutmayın.
- Bir yapılandırma dosyası özniteliğinde yanlış bir değer ayarlarsanız, etki, özniteliğin tamamen eksik olmasıyla aynıdır. Değer yoksayılr ve sorunu belirtmek için bir işleç iletisi yayınlanır.
-  IBM işletim sistemi üzerinde, .ini dosyaları IFS ' de bulunan akış dosyalarıdır.
- mqat.ini dosyasının biçimi için bir dizi sözdizimi kuralı vardır. Daha fazla bilgi için bkz. Uygulama etkinliği izlemesi [mqat.inikullanarak etkinlik izleme davranışını yapılandırma](#).

## Yordam

1. Bir yapılandırma dosyasını düzenlemeden önce, gerekirse geri döndürebileceğiniz bir kopyanızı yedekleyin.
2. .ini yapılandırma dosyasını aşağıdaki yollardan biriyle düzenleyin:
  - Standart bir metin düzenleyicisini kullanarak el ile. Açıklamalar, açıklama metninden önce bir ";" ya da "#" karakteri eklenerek yapılandırma dosyalarına eklenebilir. Bir yorumu göstermeden ";" ya da "#" karakterini kullanmak istiyorsanız, karaktere bir "\" karakteriyle önek olarak önek verebilirsiniz. Daha sonra karakter, yapılandırma verilerinin bir parçası olarak kullanılır.
  - Otomatik olarak, düğümdeki kuyruk yöneticilerinin yapılandırmasını değiştiren komutlar kullanılarak. Daha fazla bilgi için bkz. [Komutlar başvurusu](#).



**Windows** Örneğin, Windows özel komut **amqmdain** , qm . ini özelliklerinin bir alt kümesini otomatik olarak günceller. Daha fazla bilgi için bkz. [amqmdain](#).

- **Linux** **Windows** Linux (x86 ve x86-64) ve Windows sistemlerinde, IBM MQ Explorer komutunu kullanarak qm . ini özelliklerinin bir alt kümesini güncelleyebilirsiniz. Daha fazla bilgi için bakınız: [Configuring IBM MQ using MQ Explorer](#).

**Not:** Kurulabilir hizmetleri ve bunların bileşenlerini değiştirmenin önemli etkileri olduğu için, kurulabilir hizmetler IBM MQ Explorer içinde salt okunur olur. Bu nedenle, qm . ini dosyasını düzenleyerek kurulabilir hizmetlerde değişiklik yapmanız gerekir. Daha fazla bilgi için bkz [“qm.ini dosyasının hizmet kısmı” sayfa 138](#).

## İlgili görevler

Yönetme IBM MQ

## Multi IBM MQ yapılandırma dosyası, mqs.ini

IBM MQ yapılandırma kütüğü ( mqs . ini), düğümdeki tüm kuyruk yöneticileriyle ilgili bilgileri içerir. Kuruluş sırasında otomatik olarak yaratılır.

**Not:** mqs . ini dosyasının nasıl ve ne zaman düzenleneceğine ve dosyada yaptığınız herhangi bir değişikliğin ne zaman geçerli olacağına ilişkin daha fazla bilgi için bkz. [“Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerini değiştirme” sayfa 83](#).

## Dizin konumları

**Linux** **AIX** AIX and Linux sistemlerinde, veri dizini ve günlük dizini her zaman sırasıyla /var/mqm ve /var/mqm/log olur.

**Windows** Windows sistemlerinde, mqs . ini veri dizininin konumu ve günlük dizininin konumu kayıt dosyasında saklanır; bu dizinlerin yerleri değişebilir. AIX and Linux sistemlerinde mqinst . ini içinde bulunan kuruluş yapılandırma bilgileri de kayıt defterindedir; Windows dosyasında mqinst . ini dosyası yoktur (bkz. [“Kuruluş yapılandırma dosyası, mqinst.ini” sayfa 154](#)).

**Windows** Windows sistemlerine ilişkin mqs . ini dosyası, HKLM\SOFTWARE\IBM\IBM MQ anahtarında belirtilen WorkPath ile verilir. Şunları içerir:

- Kuyruk yöneticilerinin adları
- Varsayılan kuyruk yöneticisinin adı
- Her biriyle ilişkili dosyaların konumu

**IBM i** IBM sistemlerinde mqs . ini dosyası /QIBM/UserData/mqm içinde saklanır. Dosya şunları içerir:

- Kuyruk yöneticilerinin adları.
- Varsayılan kuyruk yöneticisinin adı.
- Her kuyruk yöneticisiyle ilişkili dosyaların konumu.
- API çıkışlarını tanımlayan bilgiler (daha fazla bilgi için [API çıkışlarını yapılandırma](#) başlıklı konuya bakın).

Özellikle, mqs . ini dosyası her bir kuyruk yöneticisiyle ilişkili verileri bulmak için kullanılır.

## AIX and Linux için örnek mqs . ini dosyası

**Linux** **AIX**

```
#####  
#* Module Name: mqs.ini *#  
#* Type : IBM MQ Machine-wide Configuration File *#  
#* Function : Define IBM MQ resources for an entire machine *#
```

```

#####
#* Notes : *#
#* 1) This is the installation time default configuration *#
#* *#
#####
AllQueueManagers:
#####
#* The path to the qmgrs directory, below which queue manager data *#
#* is stored *#
#####
DefaultPrefix=/var/mqm

LogDefaults:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=0
  LogDefaultPath=/var/mqm/log

QueueManager:
  Name=saturn.queue.manager
  Prefix=/var/mqm
  Directory=saturn!queue!manager
  InstallationName=Installation1

QueueManager:
  Name=pluto.queue.manager
  Prefix=/var/mqm
  Directory=pluto!queue!manager
  InstallationName=Installation2

DefaultQueueManager:
  Name=saturn.queue.manager

ApiExitTemplate:
  Name=OurPayrollQueueAuditor
  Sequence=2
  Function=EntryPoint
  Module=/usr/ABC/auditor
  Data=123

ApiExitCommon:
  Name=MQPoliceman
  Sequence=1
  Function=EntryPoint
  Module=/usr/MQPolice/tmqp
  Data=CheckEverything

```

## Windows için örnek mqz .ini dosyası



```

#####
#* Module Name: mqz.ini *#
#* Type : IBM MQ Machine-wide Configuration File *#
#* Function : Define IBM MQ resources for an entire machine *#
#####
#* Notes : *#
#* 1) This is the installation time default configuration *#
#* *#
#####
AllQueueManagers:
#####
#* The path to the qmgrs directory, below which queue manager data *#
#* is stored *#
#####
DefaultPrefix=C:\ProgramData\IBM\MQ

LogDefaults:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=0
  LogDefaultPath=C:\ProgramData\IBM\MQ\log

QueueManager:

```

```

Name=saturn.queue.manager
Prefix=C:\ProgramData\IBM\MQ
Directory=saturn!queue!manager
InstallationName=Installation1

QueueManager:
  Name=pluto.queue.manager
  Prefix=C:\ProgramData\IBM\MQ
  Directory=pluto!queue!manager
  InstallationName=Installation2

DefaultQueueManager:
  Name=saturn.queue.manager

ApiExitTemplate:
  Name=OurPayrollQueueAuditor
  Sequence=2
  Function=EntryPoint
  Module=C:\usr\ABC\auditor
  Data=123

ApiExitCommon:
  Name=MQPoliceman
  Sequence=1
  Function=EntryPoint
  Module=C:\usr\MQPolice\tmpq
  Data=CheckEverything

```

## IBM i için örnek mqsc.ini dosyası

IBM i

```

#*****#
#* Module Name: mqsc.ini                                     **#
#* Type       : IBM MQ Configuration File                 **#
#* Function   : Define IBM MQ resources for the node     **#
#*           **#
#*****#
#* Notes      :                                           **#
#* 1) This is an example IBM MQ configuration file       **#
#*           **#
#*****#
AllQueueManagers:
#*****#
#* The path to the qmgrs directory, within which queue manager data **#
#* is stored                                           **#
#*****#
DefaultPrefix=/QIBM/UserData/mqm

QueueManager:
Name=saturn.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMSATURN.Q
Directory=saturn!queue!manager

QueueManager:
Name=pluto.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMPLUTO.QU
Directory=pluto!queue!manager

DefaultQueueManager:
Name=saturn.queue.manager

```

### Notlar:

1. Dügümdeki IBM MQ , kuyruk yöneticileri ve günlükler için varsayılan konumları kullanır.
2. saturn.queue.manager kuyruk yöneticisi, düğümün varsayılan kuyruk yöneticisidir. Bu kuyruk yöneticisiyle ilişkili dosyaların dizini otomatik olarak dosya sistemi için geçerli bir dosya adına dönüştürüldü.
3. Kuyruk yöneticileriyle ilişkili verileri bulmak için IBM MQ yapılandırma dosyası kullanıldığından, var olmayan ya da yanlış bir yapılandırma dosyası, bazı ya da tüm IBM MQ komutlarının başarısız

olmasına neden olabilir. Ayrıca, uygulamalar IBM MQ yapılandırma dosyasında tanımlı olmayan bir kuyruk yöneticisine bağlanamaz.

## mqs.ini stanzas



**Uyarı:** Bu konuda, mqs.ini dosyasındaki stanzas ile ilgili daha fazla bilgi yer alır. Her bir kıta, o kıtadaki parametrelerle ilgili bilgi içerir.

### Multi mqs.ini dosya stanzalarının ve özniteliklerinin özeti

Daha fazla bilgi bağlantılarıyla birlikte IBM MQ yapılandırma dosyasının (mqs.ini) stanzas özniteliklerinin bir özeti.

Çizelge 9. mqs.ini dosyasının stanzas dosyası	
Kıta ve öznitelikler	Özniteliklerin açıklaması
<b>AllQueueYöneticiler kısmı</b>	
<u>DefaultPrefix</u>	Kuyruk yöneticisi verilerinin saklandığı qmgrs dizininin yolu.
<b>ALW</b> <u>DefaultEphemeralÖnek</u>	Kuyruk yöneticisi geçici verilerinin saklandığı dizininin yolu.
<b>Multi</b> <u>ConvEBCDICNewline</u>	IBM MQ , EBCDIC NL karakterini ASCII biçimine dönüştürmeyi
<b>ApiExitOrtak kısmı ve ApiExitŞablon kısmı</b>	
<u>NAME</u>	API çıkışının açıklayıcı adı, MQAXP yapısının ExitInfoAd alanında iletildi.
<u>function</u>	API çıkış kodunu içeren modüldeki işlev giriş noktasının adı.
<u>Modül</u>	API çıkış kodunu içeren modül.
<u>Veriler</u>	MQAXP yapısının ExitData alanında API çıkışına geçirilecek veriler.
<u>Sequence</u>	Bu API çıkışının diğer API çıkışlarıyla görel olarak çağrıldığı sıra.
<b>DefaultQueueManager kısmı</b>	
<u>NAME</u>	Kuyruk yöneticisi adının belirtik olarak belirtilmediği komutları işleyen kuyruk yöneticisinin adı.
<b>ExitProperties kısmı</b>	
<u>CLWLMode</u>	Küme çalışma alanı (CLWL) çıkış çıkışının FAST kipinde mi, yoksa SAFE kipinde mi çalıştığını belirler.
<b>LogDefaults kısmı</b>	
<u>LogPrimaryDosyaları</u>	Kuyruk yöneticisi yaratıldığında ayrılan günlük dosyaları.
<u>LogSecondaryDosyaları</u>	Birincil dosyalar tükendiğinde ayrılan günlük dosyaları.
<u>LogFileSayfaları</u>	Günlük dosyası sayfalarının sayısı. (Günlük dosyası büyüklüğü 4 KB sayfa birimi olarak belirtilir.)
<u>LogType</u>	Kuyruk yöneticisi tarafından kullanılacak günlüğe kaydetme tipi (dairesele ya da doğrusal).

Çizelge 9. mqs.ini dosyasının stanzas dosyası (devamı var)	
Kıta ve öznitelikler	Özniteliklerin açıklaması
<u>LogBufferSayfaları</u>	Arabellek kayıtlarının yazılması için ayrılan bellek miktarı, arabelleklerin 4 KB sayfa birimi cinsinden büyüklüğünü belirtir.
<u>LogDefaultYol</u>	Bir kuyruk yöneticisine ilişkin günlük dosyalarının bulunduğu dizin.
<u>LogWriteBütünlük</u>	Günlük kaydedicinin günlük kayıtlarını güvenilir bir şekilde yazmak için kullandığı yöntem.
<b>QueueManager kısmı</b>	
<u>NAME</u>	Kuyruk yöneticisinin adı.
<u>Önek</u>	Kuyruk yöneticisi dosyalarının saklandığı yer.
<u>Dizin</u>	Kuyruk yöneticisi dosyalarının saklandığı <code>prefix\QMGRS</code> dizini altındaki alt dizinin adı.
<u>DataPath</u>	Kuyruk yöneticisi yaratıldığında sağlanan belirtik bir veri yolu, kuyruk yöneticisi verilerinin yolu olarak Öneki ve Dizini geçersiz kılar.
<u>InstallationName</u>	Bu kuyruk yöneticisiyle ilişkili IBM MQ kuruluşunun adı.
<u>EphemeralPrefix</u>	Kuyruk yöneticisi geçici verilerinin saklandığı yer.

## Multi AllQueue mqs.ini dosyasının yöneticiler kısmı

AllQueueYöneticileri kısmı, bir kuyruk yöneticisiyle ilişkili dosyaların saklandığı `qmgrs` dizininin yolunu, yürütülür kitaplığın yolunu ve EBCDIC biçimli verileri ASCII biçimine dönüştürme yöntemini belirtebilir.

Tüm kuyruk yöneticilerine ilişkin bilgileri belirtmek için `mqs.ini` dosyasındaki AllQueueManager olanağını kullanın.

**Linux** **Windows** Alternatif olarak, Linux (x86 ve x86-64) ve Windows üzerinde IBM MQ Explorer General ve Extended IBM MQ özellikleri sayfasını kullanın.

### DefaultPrefix= *dizin\_adi*

Bu öznitelik, kuyruk yöneticisi verilerinin saklandığı `qmgrs` dizininin yolunu belirtir.

Kuyruk yöneticisine ilişkin varsayılan öneki değiştirirseniz, kuruluş sırasında yaratılan dizin yapısını eşler. Özellikle, `qmgrs` yapısını oluşturmalsınız. Varsayılan öneki değiştirmeden önce IBM MQ önekini durdurun ve IBM MQ ögesini yalnızca yapıları yeni konuma taşıdıktan ve varsayılan öneki değiştirdikten sonra yeniden başlatın.

**Not:** **ALW** AIX and Linux sistemlerinde `/var/mqm/errors` dizinini ya da Windows sistemlerinde `\errors` dizinini silmeyin.

Varsayılan öneki değiştirmenin bir alternatifi olarak, `crtmqm` komutuna ilişkin **DefaultPrefix** değerini geçersiz kılmak için **MQSPREFIX** ortam değişkenini kullanabilirsiniz.

İşletim sistemi kısıtlamaları nedeniyle, sağlanan yolu, yol uzunluğu ve herhangi bir kuyruk yöneticisi adının toplamı en çok 70 karakter uzunluğunda olacak şekilde yeterince kısa tutun.

## Multi DefaultEphemeralPrefix = *dizin\_adi*

Bu öznitelik, IPC uvaları gibi kuyruk yöneticisi geçici verilerinin alıkonduğu dizinin yolunu belirtir ve yalnızca bir kuyruk yöneticisi yaratıldığında kuyruk yöneticisinin **EphemeralPrefix** değerini ayarlamak için kullanılır. Ayrıca, varsayılan değeri değiştirirseniz dizini kendiniz yaratmanız gerekir. IBM MQ grubunun o dizine yazmasına izin veren izinlerle geçici veri dizinini oluşturmanız gerekir.

mqs.ini dosyasını deęiřtirmenin bir alternatifi olarak, **crtmqm** komutu için **DefaultEphemeralPrefix** deęerini geersiz kılmak üzere **MQ\_EPHEMERAL\_PREFIX** ortam deęiřkenini kullanabilirsiniz.

İřletim sistemi kısıtlamaları nedeniyle, varsayılan geici önek řunlarla sınırlıdır:

- **Linux** **AIX** AIX and Linux platformlarında 12 karakter.
- **IBM i** IBM üzerinde 24 karakter.

**MQ Appliance** **DefaultEphemeralPrefix**, IBM MQ Appliance üzerinde desteklenmez.

#### **Multi** **ConvEBCDICNewline= NL\_TO\_LF | TABLE | ISO**

EBCDIC kod sayfaları, ASCII kod sayfaları tarafından desteklenmeyen bir yeni satır (NL) karakteri içerir (ASCII 'nin bazı ISO çeřitlemeleri eřdeęeri içerir). IBM MQ 'un EBCDIC NL karakterini ASCII biçimine nasıl dönüřtüreceęini belirtmek için **ConvEBCDICNewline** öznitelięini kullanın.

**IBM i** IBM MQ for IBM üzerinde, CCSID 1253 bir ISO CCSID olarak kabul edilir ve NL\_TO\_LF hem ISO hem de ASCII dönüřtürmelerini etkiler.

**z/OS** **ConvEBCDICNewline** öznitelięi z/OS üzerinde kullanılamaz. z/OS üzerindeki davranıř, ConvEBCDICNewline=TABLE ile eřdeęerdir. Dięer platformlardaki varsayılan deęerin farklı olabileceęini unutmayın.

#### **NL\_TO\_LF**

EBCDIC NL karakterini (X'15 '), tüm EBCDIC 'den ASCII 'ye dönüřtürmeler için ASCII satır besleme karakterine (LF) (X'0A') dönüřtürün.

NL\_TO\_LF varsayılan deęerdir.

#### **Çizelge**

EBCDIC NL karakterini, tüm EBCDIC 'den ASCII' ye dönüřtürmeler için altyapınızda kullanılan dönüřtürme çizelgelerine göre dönüřtürün.

Bu tür bir dönüřtürmenin etkisi platformdan platforma ve dilden dile deęiřebilir; aynı platformda bile, farklı CCSID 'ler kullanırsanız davranıř deęiřebilir.

#### **ISO**

Dönüřtür:

- TABLE yöntemini kullanan ISO CCSID 'leri
- NL\_TO\_CF yöntemini kullanan dięer tüm CCSID 'ler

Olası ISO CCSID 'leri [Çizelge 10 sayfa 90](#) içinde gösterilir.

<b>CCSID</b>	<b>Kod Kümesi</b>
819	ISO8859-1
912	ISO8859-2
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7
916	ISO8859-8
920	ISO8859-9
1051	roman8

ASCII CCSID 'si bir ISO altkümümesi değilse, **ConvEBCDICNewline** varsayılan olarak NL\_TO\_LF kullanılır.

IBM MQ 9.1.0 Fix Pack 2 ve IBM MQ 9.1.2olanağından, **ConvEBCDICNewline** kısmı özniteliği yerine **AMQ\_CONVEBCDICNEWLİNE** ortam değişkenini kullanabilirsiniz; örneğin, mqsc.ini dosyasının kullanılmadığı durumlarda istemci tarafında **ConvEBCDICNewline** işlevselliğini sağlamak için. Ortam değişkeni, **ConvEBCDICNewline** özniteliğiyle aynı değerleri (NL\_TO\_LF, TABLEya da ISO) alır. Hem öznitelik hem de ortam değişkeni ayarlanırsa, stanza özniteliği önceliklidir.

Multi

## ApiExit mqsc.ini dosyasının ortak ve ApiExitŞablon kısmı

ApiExitŞablonu ve ApiExitOrtak kısmı, tüm kuyruk yöneticileri için API çıkış yordamlarını tanımlar.

Tüm kuyruk yöneticilerine ilişkin API çıkış yordamlarını tanımlamak için mqsc.ini dosyasındaki ApiExitŞablonu ve ApiExitCommon stanzlarını kullanın. (Tek tek kuyruk yöneticilerine ilişkin API çıkış yordamlarını tanımlamak için, “[ApiExit qm.ini dosyasının yerel kısmı](#)” sayfa 109’ünde açıklandığı gibi ApiExitYerel kısmı kullanılır.)

Linux

Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer Exits IBM MQ özellikler sayfasını kullanın.

Windows

Windows üzerinde, API çıkışlarına ilişkin girdileri değiştirmek için **amqmdain** komutunu da kullanabilirsiniz.

Bu öznitelikleri kullanma hakkında daha fazla bilgi için [API çıkışlarını yapılandırma](#) başlıklı konuya bakın.

### Ad=ApiExit\_name

API çıkışının açıklayıcı adı, MQAXP yapısının ExitInfoAd alanında iletilir.

Bu ad benzersiz olmalı, 48 karakterden uzun olmamalı ve yalnızca IBM MQ nesnelere adları (örneğin, kuyruk adları) için geçerli karakterler içermelidir.

### İşlev=işlev\_adi

API çıkış kodunu içeren modüldeki işlev giriş noktasının adı. Bu giriş noktası MQ\_INIT\_EXIT işlevidir.

Bu alanın uzunluğu MQ\_EXIT\_NAME\_LENGTH ile sınırlıdır.

### Module=modül\_adi

API çıkış kodunu içeren modül.

Bu alan modülün tam yol adını içeriyorsa, olduğu gibi kullanılır. Bu alan yalnızca modül adını içeriyorsa, modül, qm.ini dosyasının ExitPath kısmında **ExitsDefaultPath** özniteliği kullanılarak bulunur.

Ayrı iş parçacıklı kitaplıkları destekleyen platformlarda, API çıkış modülünün hem iş parçacıklı olmayan hem de iş parçacıklı bir sürümünü sağlamanız gerekir. İş parçacıklı sürümün bir \_x soneki olmalıdır. IBM MQ uygulama sınırlı kod öbeğinin iş parçacıklı sürümü, yüklenmeden önce belirtilen modül adının sonuna \_x örtük olarak eklenir.

Bu alanın uzunluğu, altyapının desteklediği yol uzunluğu üst sınırıyla sınırlıdır.

### Veri=veri\_adi

MQAXP yapısının ExitData alanında API çıkışına geçirilecek veriler.

Bu özniteliği eklerse, baştaki ve sondaki boşluklar kaldırılır, kalan dizgi 32 karaktere kesilir ve sonuç çıkışa geçirilir. Bu özniteliği atlarsanız, çıkışa varsayılan değer olan 32 boşluk geçirilir.

Bu alanın uzunluk üst sınırı 32 karakterdir.

### Sequence=sequence\_number (Sıralama\_numarası)

Bu API çıkışının diğer API çıkışlarıyla görelilik olarak çağrıldığı sıra. Düşük sıra numarasına sahip bir çıkış, daha yüksek sıra numarasına sahip bir çıkıştan önce çağrılır. Çıkışların sıra numaralandırmasının bitişik olması gerekmez. 1, 2, 3 dizisinin sonucu 7, 42, 1096 ile aynıdır. İki çıkış aynı sıra numarasına sahipse, kuyruk yöneticisi hangisinin önce aranacağına karar verir. MQAXP ' de ExitChainAreaPtr ile gösterilen ExitChainAlanına zamanı ya da bir imleyiciyi koyarak ya da kendi günlük dosyanızı yazarak, olaydan sonra hangisinin çağrıldığı anlayabilirsiniz.

Bu öznitelik, işaretli bir sayısal değerdir.

Multi

## DefaultQueue mqs.ini dosyasının Manager kısmı

DefaultQueueManager kısmı, düğüme ilişkin varsayılan kuyruk yöneticisini belirtir.

Varsayılan kuyruk yöneticisini belirtmek için mqs . ini dosyasındaki DefaultQueueManager kısmına bakın.

Linux

Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer General IBM MQ özellikler sayfasını kullanın.

### Ad = varsayılan\_kuyruk\_yöneticisi

Varsayılan kuyruk yöneticisi, kuyruk yöneticisi adının açık olarak belirtilmediği komutları işler. Yeni bir varsayılan kuyruk yöneticisi yaratırsanız, **DefaultQueueManager** özniteliği otomatik olarak güncellenir. Yanlışlıkla yeni bir varsayılan kuyruk yöneticisi yaratır ve özgün kuyruk yöneticisine geri dönmek isterseniz, **DefaultQueueManager** özniteliğini el ile değiştirin.

Multi

## mqs.ini dosyasının ExitProperties kısmı

ExitProperties kısmı, kuyruk yöneticisi çıkış programları tarafından kullanılan yapılandırma seçeneklerini belirtir.

Kuyruk yöneticisi çıkış programları tarafından kullanılan yapılandırma seçeneklerini belirtmek için mqs . ini dosyasındaki ExitProperties (ExitProperties) kısmına bakın.

Linux

Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer Extended IBM MQ özellikler sayfasını kullanın.

### CLWLMode = SAFE (varsayılan) | HIZLI

Küme iş yükü (CLWL) çıkışı, bir MQI çağrısına yanıt olarak hangi küme kuyruğunun açılacağını belirtmenizi sağlar (örneğin, MQOPEN, MQPUT). CLWL çıkışı, **CLWLMode** özniteliğinde belirttiğiniz değere bağlı olarak FAST kipinde ya da SAFE kipinde çalışır. **CLWLMode** özniteliğini atlarsanız, küme iş yükü çıkışı SAFE kipinde çalışır.

#### GÜVENLİ

CLWL çıkışını kuyruk yöneticisinden ayrı bir işlemde çalıştırın. Bu varsayılandır.

SAFE kipinde çalışırken kullanıcı tarafından yazılan CLWL çıkışıyla ilgili bir sorun ortaya çıkarsa, aşağıdaki durum oluşur:

- CLWL sunucusu işlemi (amqzlw0) başarısız oldu.
- Kuyruk yöneticisi CLWL sunucusu işlemini yeniden başlatır.
- Hata, hata günlüğünde size bildirilir. Bir MQI çağrısı devam ediyorsa, dönüş kodu biçiminde bildirim alırsınız.

Kuyruk yöneticisinin bütünlüğü korunur.

**Not:** CLWL çıkışının ayrı bir işlemde çalıştırılması performansı etkileyebilir.

#### Hızlı

Kuyruk yöneticisi işleminde küme çıkışını satır içinde çalıştırın.

Bu seçeneğin belirtilmesi, SAFE kipinde çalıştırmayla ilişkili süreç geçiş maliyetlerinden kaçınarak performansı artırır, ancak kuyruk yöneticisi bütünlüğü pahasına bunu yapar. CLWL çıkışını yalnızca CLWL çıkışınızla ilgili bir sorun olmadığından eminseniz ve özellikle performans konusunda endişeliyseniz, CLWL çıkışını FAST kipinde çalıştırmamız gerekir.

CLWL çıkışı FAST kipinde çalışırken bir sorun ortaya çıkarsa, kuyruk yöneticisi başarısız olur ve kuyruk yöneticisinin bütünlüğü tehlikeye atılır.

Multi

## mqs.ini dosyasının LogDefaults kısmı

LogDefaults kısmı, tüm kuyruk yöneticilerine ilişkin günlük varsayılanlarıyla ilgili bilgileri belirtir.

Tüm kuyruk yöneticilerine ilişkin günlük varsayılanlarına ilişkin bilgileri belirtmek için mqs . ini dosyasındaki LogDefaults (Günlük Varsayılanları) kısmı kullanılır.



Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer Default log settings IBM MQ özellikler sayfasını kullanın.

Varsayılan olmayan bir değere gereksiniminiz varsa, bu değeri LogDefaults (LogDefaults) kısmına belirttik olarak belirtmeniz gerekir.

LogDefaults kısmı yoksa, IBM MQ varsayılanları kullanılır. Günlük öznitelikleri, bir kuyruk yöneticisi yarattığınızda varsayılan değerler olarak kullanılır, ancak **crtmqm** komutunda günlük özniteliklerini belirtirseniz geçersiz kılınabilir. Bu komutla ilgili daha fazla bilgi için bkz. **crtmqm**.

Bir kuyruk yöneticisi yaratıldıktan sonra, o kuyruk yöneticisine ilişkin günlük öznitelikleri "[qm.ini dosyasının günlük kısmı](#)" sayfa 129'de açıklanan ayarlardan alınır.

**Not:** Yeni bir IBM MQ kuruluşu için sağlanan LogDefaults kısmı, öznitelikler için belirttik bir değer içermiyor. Öznitelik olmaması, yeni bir kuyruk yöneticisi yaratıldığı sırada bu değer için varsayılan değer kullanıldığı anlamına gelir. LogDefaults kısmı için varsayılan değerler "[AIX and Linux için örnek mq.s.ini dosyası](#)" sayfa 85 ve "[Windows için örnek mq.s.ini dosyası](#)" sayfa 86'de gösterilir. LogBufferPages özniteliği için sıfır değeri 512 anlamına gelir.

"[AllQueue mq.s.ini dosyasının yöneticiler kısmı](#)" sayfa 89'de belirtilen varsayılan örnek ve "[qm.ini dosyasının günlük kısmı](#)" sayfa 129'de belirtilen kuyruk yöneticisi için belirtilen günlük yolu, kuyruk yöneticisinin ve günlükünün farklı fiziksel sürücülerde olmasına izin verir. Bu, varsayılan olarak aynı sürücüde olmalarına rağmen, önerilen yöntemdir.

Günlük boyutlarını hesaplama hakkında bilgi için bkz. "[Günlüğün boyutunun hesaplanması](#)" sayfa 633.

**Not:** Aşağıdaki parametre listesinde belirtilen sınırlar, IBM MQ tarafından ayarlanan sınırlardır. İşletim sistemi sınırları, olası günlük büyüklüğü üst sınırını azaltabilir.

#### **LogPrimaryFiles = 3 (varsayılan) | 2-254 (Windows) | 2-510 (AIX and Linux)**

Kuyruk yöneticisi yaratıldığında ayrılan günlük dosyaları.

Sahip olabileceğiniz birincil günlük dosyası sayısı alt sınırı 2 'dir ve üst sınırı Windows için 254, AIX and Linux için 510 'dur. Varsayılan 3'tür.

Birincil ve ikincil günlük dosyalarının toplam sayısı, Windows üzerinde 255 ya da AIX and Linux üzerinde 511 'i geçmemelidir ve 3 'ten küçük olmamalıdır.

Değer, kuyruk yöneticisi yaratıldığında ya da başlatıldığında incelenir. Kuyruk yöneticisi yaratıldıktan sonra değiştirebilirsiniz. Ancak, kuyruk yöneticisi yeniden başlatılıncaya kadar değerdeki bir değişiklik yürürlüğe girmez ve etki hemen başlamayabilir.

#### **LogSecondaryFiles = 2 (varsayılan) | 1-253 (Windows) | 1-509 (AIX and Linux)**

Birincil dosyalar tükendiğinde ayrılan günlük dosyaları.

İkincil günlük kütüğü sayısı alt sınırı 1 'dir ve üst sınırı Windows için 253 ya da AIX and Linux için 509 'dur. Varsayılan sayı 2 'dir.

Birincil ve ikincil günlük dosyalarının toplam sayısı, Windows üzerinde 255 ya da AIX and Linux üzerinde 511 'i geçmemelidir ve 3 'ten küçük olmamalıdır.

Değer, kuyruk yöneticisi başlatıldığında incelenir. Bu değeri değiştirebilirsiniz, ancak değişiklikler kuyruk yöneticisi yeniden başlatılıncaya kadar yürürlüğe girmez ve bu durumda bile etki hemen başlamayabilir.

#### **LogFileSayfalar = sayı**

Günlük verileri, günlük dosyaları adı verilen bir dizi dosyada tutulur. Günlük dosyası büyüklüğü 4 KB 'lık sayfa birimlerinde belirtilir.

Günlük dosyası sayfalarının varsayılan sayısı 4096 'dır ve günlük dosyası boyutu 16 MB 'dir.

AIX and Linux işletim sistemlerinde, günlük kütüğü sayfası sayısı alt sınırı 64 'tür ve Windows işletim sistemlerinde günlük kütüğü sayfası sayısı alt sınırı 32 'dir; her iki durumda da üst sınır değeri 65 535 'tir.

**Not:** Kuyruk yöneticisi yaratılırken belirtilen günlük dosyalarının büyüklüğü bir kuyruk yöneticisi için değiştirilemez.

## LogType = CIRCULAR (varsayılan) | LINEAR

Kullanılacak günlüğün tipi. Varsayılan değer CIRCULAR' dır.

### Dairesel

Sistem durdurulduğunda devam eden hareketleri geri almak için günlüğü kullanarak yeniden başlatmayı başlatın.

Döngüsel günlüğe kaydetmeye ilişkin daha ayrıntılı açıklamalar için bkz. "[Günlüğe kaydetme tipleri](#)" sayfa 628 .

### Doğrusal

Hem yeniden başlatma kurtarması hem de ortam ya da ileriye işleyerek kurtarma için (günlüğün içeriğini yeniden oynatarak kayıp ya da hasarlı veri yaratılması).

Doğrusal günlüğe kaydetmenin tam açıklaması için bkz. "[Günlüğe kaydetme tipleri](#)" sayfa 628 .

Varsayılanı değiştirmek istiyorsanız, LogType özneliğini düzenleyebilir ya da **crtmqm** komutunu kullanarak doğrusal günlük kaydını belirtebilirsiniz.

IBM MQ 9.1.0' den bir kuyruk yöneticisi yaratıldıktan sonra günlüğe kaydetme yöntemini değiştirebilirsiniz. Daha fazla bilgi için bkz. [migmqlog](#).

## LogBufferPages=0 (varsayılan) |0-4096

Arabellek kayıtlarının yazılması için ayrılan bellek miktarı, arabelleklerin 4 KB sayfa birimi cinsinden büyüklüğünü belirtir.

Arabellek sayfası sayısı alt sınırı 18 ve üst sınırı 4096 'dır. Daha büyük arabellekler, özellikle daha büyük iletiler için daha yüksek verim sağlar.




0 (varsayılan) değerini belirlerseniz, kuyruk yöneticisi 512 (2048 KB) olan boyutu seçer.

1-17 aralığında bir sayı belirtirseniz, kuyruk yöneticisi varsayılan olarak 18 (72 KB) değerini alır. 18 ve 4096 aralığında bir sayı belirtirseniz, kuyruk yöneticisi ayrılan belleği ayarlamak için belirtilen sayıyı kullanır.

## LogDefaultYol = *dizin\_adi*

Bir kuyruk yöneticisine ilişkin günlük dosyalarının bulunduğu dizin. Dizin, kuyruk yöneticisinin yazabileceği ve tercihen ileti kuyruklarından farklı bir sürücüde bulunduğu yerel bir aygıtta bulunur. Farklı bir sürücü belirtilmesi, sistem arızası durumunda ek koruma sağlar.

Varsayılan:

-  *DefaultPrefix* \log for IBM MQ for Windows ; burada *DefaultPrefix* , All Queue Managers IBM MQ özellikler sayfasındaki *DefaultPrefix* özneliğinde belirtilen değerdir. Bu değer, kuruluş sırasında ayarlanır.
-   AIX and Linux sistemleri için /var/mqm/log .

Diğer bir seçenek olarak, **-ld** işaretini kullanarak **crtmqm** komutunda bir dizinin adını belirtebilirsiniz. Bir kuyruk yöneticisi yaratıldığında, kuyruk yöneticisi dizini altında da bir dizin yaratılır ve bu, günlük dosyalarını tutmak için kullanılır. Bu dizinin adı, kuyruk yöneticisi adını temel alır. Bu, günlük dosyası yolunun benzersiz olmasını ve dizin adı uzunluklarıyla ilgili sınırlamalara uymasını sağlar.

**crtmqm** komutunda **-ld** belirtmezseniz, *mqc.ini* dosyasındaki **LogDefaultPath** özneliğinin değeri kullanılır.

Kuyruk yöneticisi adı, birden çok kuyruk yöneticisinin farklı günlük dizinleri kullandığından emin olmak için dizin adının sonuna eklenir.

Kuyruk yöneticisi yaratıldığında, yapılanış bilgilerindeki günlük özneliklerinde bir **LogPath** değeri yaratılır ve kuyruk yöneticisinin günlüğüne ilişkin tam dizin adı belirlenir. Bu değer, kuyruk yöneticisi başlatıldığında ya da silindiğinde günlüğün yerini belirlemek için kullanılır.

## LogWriteIntegrity =SingleWrite|DoubleWrite|TripleWrite (varsayılan)

Günlük kaydedicinin günlük kayıtlarını güvenilir bir şekilde yazmak için kullandığı yöntem.

## TripleWrite (varsayılan)

DoubleWrites seçeneğini belirleyebilirsiniz, ancak bunu yaparsanız sistem bunu TripleWrite olarak yorumlar.

## SingleWrite

SingleWrite komutunu kullanmanız gerekir; ancak, IBM MQ kurtarma günlüğünü barındıran dosya sistemi ve aygıt 4KB 'nin atomlarına yazılmasını açık bir şekilde güvenceye alıyorsa.

Başka bir deyişle, herhangi bir nedenle 4KB sayfasının yazılması başarısız olduğunda, yalnızca iki olası durum işlem öncesi görüntüsü ya da sonraki görüntüdür. Hiçbir ara durum mümkün olmamalıdır.

**Not:** Kalıcı iş yükünde yeterli eşzamanlılık varsa, varsayılan değer olan TripleWrite dışında bir değer ayarlanmasında en düşük potansiyel avantaj vardır.

Daha fazla bilgi için bkz "[LogWriteIntegrity- SingleWrite ya da TripleWrite](#)" sayfa 132.

Multi

## mqs.ini dosyasının QueueManager kısmı

QueueManager kısmı, kuyruk yöneticisi dizininin yerini belirtir.

Her kuyruk yöneticisi için bir QueueManager kısmı vardır. Bu koşulun öznitelikleri, kuyruk yöneticisi adını ve o kuyruk yöneticisiyle ilişkili dosyaları içeren dizinin adını belirtir. Dizinin adı kuyruk yöneticisi adına dayalıdır, ancak kuyruk yöneticisi adı geçerli bir dosya adı değilse dönüştürülür. Ad dönüşümüne ilişkin ek bilgi için [IBM MQ dosya adlarını anlamabaşlıklı konuya](#) bakın.

### Ad = kuyruk\_yöneticisi\_adı

Kuyruk yöneticisinin adı.

### Önek = önek

Kuyruk yöneticisi dosyalarının saklandığı yer. Varsayılan olarak bu değer, mqs.ini dosyasındaki [Tüm Kuyruk Yöneticileri](#) ögesinin **DefaultPrefix** özneliğinde belirtilen değerle aynıdır.

### Dizin = ad

Kuyruk yöneticisi dosyalarının saklandığı *prefix*\QMGRS dizini altındaki alt dizinin adı. Bu ad kuyruk yöneticisi adına dayalıdır, ancak yinelenen bir ad varsa ya da kuyruk yöneticisi adı geçerli bir dosya adı değilse dönüştürülebilir.

### DataPath= yol

Kuyruk yöneticisi yaratıldığında sağlanan belirtik bir veri yolu, kuyruk yöneticisi verilerinin yolu olarak **Prefix** ve **Directory** 'yi geçersiz kılar.

### InstallationName= ad

Bu kuyruk yöneticisiyle ilişkili IBM MQ kuruluşunun adı. Bu kuruluştan gelen komutlar, bu kuyruk yöneticisiyle etkileşimde bulunurken kullanılmalıdır.

IBM i

### Kitaplık = ad

Bu kuyruk yöneticisiyle (örneğin, günlükler ve günlük nesnelere) ilgili IBM i nesnelere saklandığı kitaplığın adı. Bu ad kuyruk yöneticisi adına dayalıdır, ancak yinelenen bir ad varsa ya da kuyruk yöneticisi adı geçerli bir kitaplık adı değilse dönüştürülebilir.

### EphemeralPrefix= ad

Kuyruk yöneticisi geçici verilerinin saklandığı yer.

Varsayılan olarak bu değer yoktur; bu, verilerin Önek konumu altında saklandığı anlamına gelir.

Değer, kuyruk yöneticisi yaratıldığında, mqs.ini dosyasındaki **MQ\_EPHEMERAL\_PREFIX** ortam değişkeninin değerinden ya da [AllQueueManager](#) olanağının **DefaultEphemeralPrefix** özneliğinden ayarlanır.

IBM i

İşletim sistemi kısıtlamaları nedeniyle, varsayılan geçici önek IBM üzerinde 24 karakterle sınırlıdır.

## İlgili görevler

["Bir kuyruk yöneticisinin kuruluşla ilişkilendirilmesi"](#) sayfa 452

Bir kuyruk yöneticisi oluşturduğunuzda, **crtmqm** komutunu yayınlayan kuruluşla otomatik olarak ilişkilendirilir. AIX, Linux, and Windows' da, **setmqm** komutunu kullanarak bir kuyruk yöneticisiyle ilişkili kuruluşu değiştirebilirsiniz.

## Windows Gelişmiş Yapılandırma ve Güç Arabirimi (ACPI)

Windows , ACPI (Advanced Configuration and Power Interface; Gelişmiş Yapılandırma ve Güç Arabirimi) standardını destekler. Bu, ACPI etkinleştirilmiş donanıma sahip Windows kullanıcılarının, sistem askıya alma kipine girdiğinde ve bu kipten devam ettiğinde kanalları durdurmalarını ve yeniden başlatmalarını sağlar.

Sistem bir askıya alma isteği aldığı anda IBM MQ ' in nasıl davranacağını belirlemek için IBM MQ Explorer içindeki ACPI IBM MQ özellikleri sayfasını kullanın.

ACPI IBM MQ özellikler sayfasında belirtilen ayarların yalnızca Uyarı İzleyicisi çalışırken uygulandığını unutmayın. Uyarı İzleyicisi çalışıyorsa, görev çubuğunda Uyarı İzleyicisi simgesi vardır.

### DoDialog= Y | N

Askıya alma isteği sırasında iletişim kutusunu görüntüler.

### DenySuspend= Y | N

Askıya alma isteğini reddeder. Bu, DoDialog= N ise ya da DoDialog= Y ise ve not defteri kapakınız kapalı olduğu için bir iletişim kutusu görüntülenemiyorsa kullanılır.

### CheckChannelsRunning=Y | N

Kanalların çalışıp çalışmadığını denetler. Sonuç, diğer ayarların sonucunu belirleyebilir.

Aşağıdaki çizelge, bu değiştiregelerin her birleşiminin etkisini özetler:

DoDialog	DenySuspend	CheckChannels Çalışıyor	Eylem
N	N	N	Askıya alma isteğini kabul edin.
N	N	Y	Askıya alma isteğini kabul edin.
N	Y	N	Askıya alma isteğini reddet.
N	Y	Y	Çalışan herhangi bir kanal varsa, askıya alma isteğini reddet; isteğin kabul edilmemesi durumunda.
Y	N	N	İletişim kutusunu görüntüle (bkz. <a href="#">Not</a> ; askıya alma isteğini kabul edin). Bu varsayılandır.
Y	N	Y	Çalışan bir kanal yoksa, askıya alma isteğini kabul edin; iletişim kutusu görüntüleniyorsa (bkz. <a href="#">Not</a> ; isteği kabul edin).
Y	Y	N	İletişim kutusunu görüntüle ( <a href="#">Not</a> ; askıya alma isteğini reddetme).
Y	Y	Y	Çalışan kanal yoksa, askıya alma isteğini kabul edin; iletişim kutusu görüntüleniyorsa ( <a href="#">Not</a> ; isteği reddetme).

**Not:** İşlemin iletişim kutusunu görüntüleyeceği durumlarda, iletişim kutusu görüntülenemiyorsa (örneğin, not defteri kapakınızın kapalı olması nedeniyle), askıya alma isteğinin kabul edilip edilmediğini belirlemek için DenySuspend seçeneği kullanılır.

## Kuyruk yöneticisi yapılandırma dosyaları, qm.ini

Kuyruk yöneticisi yapılandırma dosyası ( `qm.ini`), belirli bir kuyruk yöneticisiyle ilgili bilgileri içerir. Tek bir kuyruk yöneticisinin yapılandırmasını değiştirmek için kullanabileceğiniz öznitelikler, IBM MQ ile ilgili ayarları geçersiz kılar.

Her kuyruk yöneticisi için bir kuyruk yöneticisi yapılandırma dosyası vardır. `qm.ini` dosyası, ilişkilendirildiği kuyruk yöneticisi yaratıldığında otomatik olarak yaratılır.

**Not:** Bir `qm.ini` dosyasının nasıl ve ne zaman düzenleneceğine ve dosyada yaptığınız herhangi bir değişikliğin ne zaman geçerli olacağına ilişkin daha fazla bilgi için bkz. "[Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerini değiştirme](#)" sayfa 83.

IBM MQ 9.0.4 ve IBM MQ 9.0.0 Fix Pack 2' den **strmqm** komutu, kuyruk yöneticisini tam olarak başlatmadan önce `qm.ini` dosyasındaki KANAL ve SSL stanzalarının sözdizimini denetler; bu, sorunun ne olduğunu görmeyi çok daha kolay hale getirir ve **strmqm**, `qm.ini` dosyasında hata olduğunu bulursa bunu hızlı bir şekilde düzeltin. Daha fazla bilgi için bkz. [strmqm](#).

### qm.ini dosyalarının konumu

Linux

AIX

AIX and Linux sistemlerinde, kuyruk yöneticisinin kapladığı dizin ağacının kökünde bir `qm.ini` dosyası tutulur. Örneğin, QMNAME adlı bir kuyruk yöneticisine ilişkin konfigürasyon dosyasının yolu ve adı:

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

Windows

Windows sistemlerinde, `qm.ini` kütüğünün yeri HKLM\SOFTWARE\IBM\WebSphere MQ anahtarında belirtilen WorkPath ile verilir. Örneğin, QMNAME adlı bir kuyruk yöneticisine ilişkin konfigürasyon dosyasının yolu ve adı aşağıdaki gibidir:

```
C:\ProgramData\IBM\MQ\qmgrs\QMNAME\qm.ini
```

IBM i

Bir `qm.ini` kütüğü `mqmdata directory/QMNAME/qm.ini` içinde tutulur; burada `mqmdata directory` varsayılan olarak `/QIBM/UserData/mqm` ve `QMNAME` kullanıma hazırlama kütüğünün uygulandığı kuyruk yöneticisinin adıdır.

**Not:** `mqm.ini` dosyasındaki `mqmdata directory` dosyasını değiştirebilirsiniz.

Kuyruk yöneticisi adı en çok 48 karakter uzunluğunda olabilir. Ancak bu, adın geçerli ya da benzersiz olduğunu garanti etmez. Bu nedenle, kuyruk yöneticisi adına dayalı olarak bir dizin adı oluşturulur. Bu süreç *ad dönüşümü* olarak bilinir. Açıklama için bkz. [IBM MQ dosya adları](#) ve [IBM i üzerindeki nesne adları](#).

### qm.ini stanzas



#### Uyarı:

- Bu konuda, `qm.ini` dosyasındaki stanzas ile ilgili daha fazla bilgi yer alır. Her bir kıta, uygun olduğu durumlarda bir örnek de dahil olmak üzere, o kitedeki parametrelerle ilgili bilgileri içerir.
- Her bir kıta, o kıta için geçerli olan IBM MQ for Multiplatforms platformunu veya platformlarını gösterir.

Multi

### Başlangıçta qm.ini otomatik yapılandırması

IBM MQ 9.2.0' den kuyruk yöneticinizi, her kuyruk yöneticisi başlangıcında `qm.ini` geçersiz kılma değerlerini içeren bir dosyanın ya da dosya kümesinin içeriğini otomatik olarak uygulaması için yapılandırabilirsiniz.

Bir sonraki kuyruk yöneticisi yeniden başlatıldığında değiştirilebilen ve otomatik olarak yeniden yürütülebilir bir yapılandırmaya sahip olmak için bu seçeneği kullanabilirsiniz. Örneğin, `qm.ini` geçersiz kılma

değerleri takılı bir sürücüde bulunuyorsa, en son sürümün başlatıldığı her kuyruk yöneticisine uygulandığı merkezi bir yapılandırmaya sahip olmak mümkündür.

Otomatik küme işlevselliğini kullanarak tek tip küme yaratmayı basitleştirmek için bu işlevselliği kullanabilirsiniz. Bunun bir örneği için bkz. [“Yeni bir örnek küme oluşturma” sayfa 415.](#)

**Not:** Bu geçersiz kılma değerleri yalnızca kuyruk yöneticisi başlangıcında uygulanır ve kuyruk yöneticisinin yaratılmasını etkileyemez. Örneğin, bu işlevle birincil günlük dosyalarının sayısını ayarlayamazsınız.

## Başlamadan önce

Aşağıdakileri kullanabilirsiniz:

1. Tek bir dosya ve `qm.ini` dosyasında yapılan değişiklikleri içeren bir metin dosyası oluşturun.
2. `qm.ini` biçim dosyaları kümesi:
  - Yapılandırmaların var olduğu bir dizini tanımlamak için, ve
  - Bu dizinde, her biri `.ini` uzantılı dosyalar oluşturun; örneğin, `qminisettings.ini`.

Dosya ya da dosyalar, yalnızca değişen öğelere ilişkin kıta ve **attribute=değer** ayarlarını içermelidir. Örneğin, Kanal ızgarasında **MaxChannels** özneliğini güncellemek için dosya şunları içerebilir:

```
Channels:  
MaxChannels=1234
```

`qm.ini` geçersiz kılma dosyalarında, önceki `#` olan satırın bir açıklama olarak kabul edildiğini unutmayın.

## qm.ini dosya özneliklerinin otomatik yapılandırmasını etkinleştirme

**strmqm** komutunun **-ii** işaretini kullanarak ve belirli bir dosyayı ya da dizini göstererek yeni bir kuyruk yöneticisi yapılandırabilirsiniz. Sağlanan değer, `qm.ini` dosyasında **AutoConfig** kısmı altında, öznelik olarak **Iniconfig** olarak saklanır.

Geçerli bir dosya ya da dizini gösteren **AutoConfig** stanza özneliği **Iniconfig** ekleyerek, var olan bir kuyruk yöneticisini otomatik MQSC yapılandırmasını etkinleştirecek şekilde yapılandırabilirsiniz. Örneğin:

```
AutoConfig:  
Iniconfig=C:\MQ_Configuration\uniclus.ini
```

## Otomatik yapılandırma nasıl çalışır?

Kuyruk yöneticisi başlatılırken, geçerli sözdizimini sağlamak için **AutoConfig** kısmı özneliği **Iniconfig** tarafından tanımlanan yapılandırma doğrulanır ve ardından kuyruk yöneticisi veri ağacında `autocfg` dizininde tek bir `cached.ini` dosyası olarak saklanır.

Bir dizindeki birden çok dosya işlendiğinde, bunlar alfabetik sırayla işlenir.

Kuyruk yöneticisi ilk başlatılırken, dosya ya da dizin okunamadı, kuyruk yöneticisinin başlatılmasını önler ve hem konsola hem de kuyruk yöneticisi hata günlüğüne uygun bir hata iletisi gönderilir.

Sonraki yeniden başlatmalarda, gösterilen dosya ya da dizin okunamıyorsa, önceden önbelleğe alınan dosya kullanılır ve kuyruk yöneticisinin hata günlüğüne yazılan bir ileti bunu vurgular.

**strmqm** komutunu kullandığınızda, kuyruk yöneticisi çağrılmadan önce `cached.ini` dosyasının içeriği `qm.ini` dosyasına geçersiz kılma olarak uygulanır.

Bu, yedek kuyruk yöneticisine ilişkin ayarların kuyruk yöneticisi etkin duruma geldiğinde değil, **strmqm** komutu işlendiğinde okunduğu anlamına gelir.

## Yeni qm.ini dosyası nasıl oluşturulur?

Otomatik kullanıma hazırlama yapılanışı ilk kez yapılandırıldığında ve kuyruk yöneticisi başlatıldığında, yürürlükteki qm . ini kütüğünün bir kopyası, kuyruk yöneticisinin veri dizinindeki autoconfig altdizinine base\_qm . iniolarak kopyalanır. Bu, buradan itibaren taban çizgisi olarak kabul edilir.

Her kuyruk yöneticisi başlangıcında, yani **strmqm** zaman, şu anda etkin olan qm . ini dosyası atılır ve base\_qm . inikopyasıyla değiştirilir. Daha sonra cached . ini dosyasındaki yapılandırma bu dosyaya uygulanır.

Bir kuyruk yöneticisi otomatik yapılandırma denetimi altına alındıktan sonra, qm . ini dosyasında yapılan tüm değişiklikler, AutoConfig (Otomatik Yapılandırma) kütüphanesindeki **IniConfig** özniteliğinin kullanılmasını gösteren dosya ya da dosyalar aracılığıyla gerçekleştirilmelidir.

Var olan qm . ini dosyası kuyruk yöneticisi başlangıcında kaldırıldığından, kuyruk yöneticisi temel satırına yalnızca **IniConfig** özniteliğini kullanan sağlanan qm . ini dosyasındaki yapılandırma uygulanır.

Kuyruk yöneticisinin önceki başlangıçlarında otomatik kullanıma hazırlama yapılanışı yoluyla bir bölüm ya da öznitelik değiştirildiyse, bu değişiklikler **IniConfig** özniteliğiyle tanımlanan dosyada ya da dosyalarda hala tanımlanmadıkları sürece kaldırılır.

Kuyruk yöneticisi başlangıcında qm . ini dosyasının yeniden yaratılması nedeniyle, bu, qm . ini dosyasında el ile yapılan değişikliklerin kaybedileceği anlamına gelir. Bir değişikliği kalıcı yapmanız gerekiyorsa ve bu değişikliği yapmak için **IniConfig** özniteliğini kullanamazsanız, aşağıdakilerden birini yapabilirsiniz:

- base\_qm . ini dosyasının kendisinde değişiklik yapın.
- base\_qm . ini dosyasını silin.

Bu dosyayı silerseniz, base\_qm . ini , qm . ini dosyasının yürürlükteki içeriğine dayalı olarak sonraki kuyruk yöneticisi başlangıcında yeniden oluşturulur. Bu, gelecekteki yeni temel çizgisi başladıkça geçerli tüm değişiklikleri *güçlendirir* .

### İlgili kavramlar

“qm.ini dosya stanzas ve özniteliklerinin özeti” sayfa 99

Daha fazla bilgi bağlantılarıyla birlikte, kuyruk yöneticisi yapılanış dosyasının ( qmi . ini) bölümlerinin özniteliklerinin özeti.

### Multi qm.ini dosya stanzas ve özniteliklerinin özeti

Daha fazla bilgi bağlantılarıyla birlikte, kuyruk yöneticisi yapılanış dosyasının ( qmi . ini) bölümlerinin özniteliklerinin özeti.

Çizelge 11. qm.ini dosyasının stanzas dosyası	
Kıta ve öznitelikler	Özniteliklerin açıklaması
<b>Windows</b> AccessMode kısmı	
<b>Windows</b> erişim grubu <sup>1</sup>	Üyeleri tüm kuyruk yöneticisi veri dosyalarına tam erişim yetkisi verilen bir Windows güvenlik grubu.
<b>ApiExitYerel parça</b>	
<u>NAME</u>	API çıkışının açıklayıcı adı, MQAXP yapısının ExitInfoAd alanında iletildi.
<u>function</u>	API çıkış kodunu içeren modüldeki işlev giriş noktasının adı.
<u>Modül</u>	API çıkış kodunu içeren modül.
<u>Veriler</u>	MQAXP yapısının ExitData alanında API çıkışına geçirilecek veriler.






Çizelge 11. qm.ini dosyasının stanzas dosyası (devamı var)

<b>Kıta ve öznitelikler</b>	<b>Özniteliklerin açıklaması</b>
<a href="#">Sequence</a>	Bu API çıkışının diğer API çıkışlarıyla görece olarak çağrıldığı sıra.
<b>Linux AIX V 9.3.4 AuthToken kısmı</b>	
<a href="#">KeyStore</a>	Güvenilen sertifika verenin genel anahtar sertifikalarını ya da simetrik anahtarlarını içeren anahtar deposu için dosya yolu.
<a href="#">KeyStorePwdFile</a>	Anahtar deposu için şifrelenmiş parolayı içeren dosyanın dosya yolu.
<a href="#">CertLabel</a>	Kimlik doğrulama belirteçlerini doğrulamak için kullanılan anahtar deposundaki bir genel anahtar sertifikasına ya da simetrik anahtara ilişkin sertifika etiketi.
<a href="#">UserClaim</a>	Kuyruk yöneticisinin yetki denetimi için kabul edebileceği kullanıcı kimliği bilgilerini içeren simge içindeki talep.
<a href="#">AllowOSGroups</a>	Bu öznitelik, benimsenen kullanıcı için grup üyeliğinin denetlenip denetlenmeyeceğini belirler.
<b>AutoCluster (Otomatik Küme) kısmı</b>	
<a href="#">Tip</a>	Otomatik kümenin tipi. Tek geçerli seçenek, tek biçimli bir kümeyi gösteren tek biçimli bir seçenektir.
<a href="#">ClusterName</a>	Otomatik kümenin adı.
<a href="#">RepositoryName1</a>	Otomatik kümedeki ilk tam havuz için kuyruk yöneticisi adı.
<a href="#">Repository1Conname</a>	Otomatik küme üyelerinin kuyruk yöneticisine nasıl bağlanması gerektiğine ilişkin bağlantı adı (CONNNAME) değeri.
<a href="#">RepositoryName2</a>	Otomatik kümedeki ikinci tam havuzun kuyruk yöneticisi adı.
<a href="#">Repository2Conname</a>	Otomatik küme üyelerinin kuyruk yöneticisine nasıl bağlanması gerektiğine ilişkin bağlantı adı (CONNNAME) değeri.
<b>AutoConfig kısmı</b>	
<a href="#">MQSCConfig</a>	Tam dosya yolu ya da tüm *.mqsc dosyalarının her kuyruk yöneticisi başlangıcında kuyruk yöneticisine uygulandığı bir dizinin yolu.
<a href="#">IniConfig</a>	Her kuyruk yöneticisi başlangıcında qm.ini dosyasına tüm *.ini dosyalarının uygulandığı bir dizinin yolu ya da tam dosya yolu.
<b>Kanallar kısmı</b>	
<a href="#">MaxChannels</a>	İzin verilen yürürlükteki kanal sayısı üst sınırı.
<a href="#">MaxActiveKanalları</a>	Herhangi bir zamanda etkin olmasına izin verilen kanal sayısı üst sınırı.
<a href="#">MaxInitiators</a>	Başlatıcı sayısı üst sınırı.



Çizelge 11. qm.ini dosyasının stanzas dosyası (devamı var)

<b>Kıta ve öznitelikler</b>	<b>Özniteliklerin açıklaması</b>
<a href="#">MQIBindType</a>	Uygulamalara ilişkin bağ tanımı.
<a href="#">PipeLineUzunluğu</a>	Bir kanalın kullanacağı koşut zamanlı iş parçacığı sayısı üst sınırı.
<a href="#">AdoptNewMCA</a>	IBM MQ bir kanal başlatma isteği aldığı anda yeni bir kanal eşgörünümü başlatılabilmesi için var olan kanal türlerinin durdurulabileceği, ancak kanalın bir örneğinin çalışmakta olduğunu bulabileceği kanallar.
<a href="#">AdoptNewMCATimeout</a>	Yeni kanal örneğinin eski kanal örneğinin sona ermesini bekleyeceği süre (saniye olarak).
<a href="#">AdoptNewMCACheck</a>	<b>AdoptNewMCA</b> özniteliği etkinleştirilirken gereken denetim tipi.
<a href="#">ChlauthEarlyBenimseme</a>	Bağlantı kimlik doğrulaması ve kanal kimlik doğrulama kurallarının işlendiği sıra.
<a href="#">PasswordProtection</a>	Kanal TLS şifrelemesini kullanmıyorsa, bir uygulama tarafından belirtilen kimlik bilgilerinin MQCSP parola korumasıyla korunmasının gerekip gerekmediği.
<a href="#">IgnoreSeqNumberMismatch</a>	Kuyruk yöneticisinin kanal başlatma sırasında bir sıra numarası uyumsuzluğu nasıl işleyeceğini denetler.
<b>Bağlantı kısmı</b>	
<a href="#">DefaultBindTip</a>	Ayrı işlemlerde çalışan uygulamalar ve kuyruk yöneticisi, bazı kaynakları paylaşır ya da paylaşmaz.
<b>DiagnosticMessages kısmı</b>	
<a href="#">NAME</a>	Bir kıta adı.
<a href="#">Hizmet</a>	Bu kıta tarafından etkinleştirilmekte olan bir hizmet.
<a href="#">ExcludeMessage</a>	Kuyruk yöneticisi hata günlüğüne yazılmayacak iletiler.
<a href="#">SuppressMessage</a>	Kuyruk yöneticisi hata günlüğüne yalnızca belirli bir zaman aralığında bir kez yazılacak iletiler.
 <a href="#">SuppressInterval</a>	<b>SuppressMessage</b> içinde belirtilen iletilerin yalnızca bir kez kuyruk yöneticisi hata günlüğüne yazıldığı saniye cinsinden zaman aralığı.
<a href="#">Önem Derecesi</a>	Önem düzeylerinin virgülle ayrılmış listesi.
<a href="#">FilePath</a>	Günlük dosyalarının yazılacağı yol. (Yalnızca Hizmet özniteliği Dosya olarak ayarlandığında desteklenir.)
<a href="#">FilePrefix</a>	Günlük dosyalarının öneki. (Yalnızca Hizmet özniteliği Dosya olarak ayarlandığında desteklenir.)
<a href="#">FileSize</a>	Günlüğün yuvarlanacağı büyüklük. (Yalnızca Hizmet özniteliği Dosya olarak ayarlandığında desteklenir.)
<a href="#">Biçim</a>	Dosyanın biçimi. (Yalnızca Hizmet özniteliği Dosya olarak ayarlandığında desteklenir.)
  <a href="#">Syslog</a>	JSON biçimi tanımlama iletileri belirtimini kullanarak syslog'a süzgeç uygulanmamış iletiler gönderen Syslog hizmeti.

Çizelge 11. qm.ini dosyasının stanzas dosyası (devamı var)


<b>Kıta ve öznitelikler</b>	<b>Özniteliklerin açıklaması</b>
<span style="background-color: #4F81BD; color: white; padding: 2px;">Linux</span> <span style="background-color: #4F81BD; color: white; padding: 2px;">AIX</span> <a href="#">Girinti</a>	Syslog girişleriyle ilişkili ident değeri. (Yalnızca Hizmet özniteliği Syslog olarak ayarlandığında desteklenir.)
<b>ExitPath kısmı</b>	
<a href="#">ExitsDefaultYolu</a>	Kuyruk yöneticisi sistemindeki (32 bit) kullanıcı çıkış programlarının yolu.
<a href="#">ExitsDefaultPath64</a>	Kuyruk yöneticisi sistemindeki kullanıcı çıkış programlarının yolu (64 bit).
<b>ExitPropertiesYerel kısmı</b>	
<a href="#">CLWLMode</a>	Küme çalışma alanı (CLWL) çıkış çıkışının FAST kipinde mi, yoksa SAFE kipinde mi çalıştığını belirler.
<span style="background-color: #4F81BD; color: white; padding: 2px;">Linux</span> <span style="background-color: #4F81BD; color: white; padding: 2px;">IBM i</span> <span style="background-color: #4F81BD; color: white; padding: 2px;">AIX</span>	<b>Dosya sistemi kısmı</b>
<span style="background-color: #4F81BD; color: white; padding: 2px;">Linux</span> <span style="background-color: #4F81BD; color: white; padding: 2px;">IBM i</span> <span style="background-color: #4F81BD; color: white; padding: 2px;">AIX</span>	<a href="#">ValidateAuth</a>
	mqm grubunun üyesi olmayan kullanıcıların hata dizinlerine ve dosyalarına erişmesine izin verir.
<b>Günlük kısmı</b>	
<a href="#">LogPrimaryDosyaları</a>	Kuyruk yöneticisi yaratıldığında ayrılan günlük dosyaları.
<a href="#">LogSecondaryDosyaları</a>	Birincil dosyalar tükendiğinde ayrılan günlük dosyaları.
<a href="#">LogFileSayfaları</a>	Günlük dosyası sayfalarının sayısı. (Günlük dosyası büyüklüğü 4 KB sayfa birimi olarak belirtilir.)
<a href="#">LogType</a>	Kuyruk yöneticisi tarafından kullanılacak günlüğe kaydetme tipi (dairesel ya da doğrusal).
<a href="#">LogBufferSayfaları</a>	Arabellek kayıtlarının yazılması için ayrılan bellek miktarı, arabelleklerin 4 KB sayfa birimi cinsinden büyüklüğünü belirtir.
<a href="#">LogPath</a>	Bir kuyruk yöneticisine ilişkin günlük dosyalarının bulunduğu dizin.
<a href="#">LogWriteBütünlük</a>	Günlük kaydedicinin günlük kayıtlarını güvenilir bir şekilde yazmak için kullandığı yöntem.
<a href="#">LogManagement</a>	Günlük kapsamlarını el ile ya da kuyruk yöneticisi tarafından yönetmek için kullanılan yöntem.
<span style="background-color: #4F81BD; color: white; padding: 2px;">Windows</span> <b>LU62 kısmı</b>	
<span style="background-color: #4F81BD; color: white; padding: 2px;">Windows</span>	<a href="#">TPName</a>
	Uzak sitede başlatılacak TP adı.
<span style="background-color: #4F81BD; color: white; padding: 2px;">Windows</span>	<a href="#">Library1</a>
	APPC DLL ' nin adı.
<span style="background-color: #4F81BD; color: white; padding: 2px;">Windows</span>	<a href="#">Library2</a>
	Kod iki ayrı kitaplıkta saklandıysa, Library1 ile aynı.
<span style="background-color: #4F81BD; color: white; padding: 2px;">CP4I</span> <b>NativeHAInstance kısmı</b>	
<a href="#">"Ad" sayfa 133</a>	Kuyruk yöneticisi yönetim ortamı yaratıldığında kullanılan yönetim ortamı adı.

Çizelge 11. qm.ini dosyasının stanzas dosyası (devamı var)	
Kıta ve öznitelikler	Özniteliklerin açıklaması
<a href="#">"ReplicationAddress" sayfa 133</a>	Eşgörünümün anasistem adı, IPv4 noktalı onlu ya da IPv6 onaltılı biçim adresi.
<b>CP4I NativeHALocalEşgörünüm kısmı</b>	
<a href="#">"LocalName" sayfa 134</a>	Yerel HA kuyruk yöneticisi yaratıldığında belirtilen günlük eşlemesi eşgörünümü adından alınan NativeHALocalyönetim ortamı kısmı adı.
<a href="#">"KeyRepository" sayfa 134</a>	Günlük eşleme trafiğinin korunması için kullanılacak sayısal sertifikayı bulduran anahtar havuzunun konumu.
<a href="#">"CertificateLabel" sayfa 135</a>	Günlük eşleme trafiğinin korunması için kullanılacak dijital sertifikayı tanımlayan sertifika etiketi.
<a href="#">"CipherSpec" sayfa 135</a>	Günlük eşleme trafiğini korumak için kullanılacak MQ CipherSpec .
<a href="#">"LocalAddress" sayfa 135</a>	Günlük eşleme trafiğini kabul eden yerel ağ arabirimi adresi.
<a href="#">"HeartbeatInterval" sayfa 135</a>	Sağlıklı işletim bildirim aralığı, Yerel HA kuyruk yöneticisinin etkin bir eşgörünümün ağ sağlıklı işletim bildirim gönderme sıklığını milisaniye cinsinden tanımlar.
<a href="#">"HeartbeatTimeout" sayfa 135</a>	Sağlıklı işletim bildirim zamanaşımı, bir Yerel HA kuyruk yöneticisinin etkin eşgörünümün yanıt vermediğine karar vermeden önce eşlemenin ne kadar bekleyeceğini tanımlar.
<a href="#">"RetryInterval" sayfa 135</a>	Yeniden deneme aralığı, bir Yerel HA kuyruk yöneticisinin başarısız bir eşleme bağlantısını ne sıklıkta yeniden denemesi gerektiğini milisaniye cinsinden tanımlar.
<b>Windows NETBIOS kısmı</b>	
<b>Windows</b> <a href="#">LocalName</a>	Bu makinenin LAN üzerinde bilindiği ad.
<b>Windows</b> <a href="#">AdapterNum</a>	LAN bağdaştırıcısının numarası.
<b>Windows</b> <a href="#">NumSess</a>	Ayrılacak oturum sayısı.
<b>Windows</b> <a href="#">NumCmds</a>	Ayrılacak komutların sayısı.
<b>Windows</b> <a href="#">NumNames</a>	Ayrılacak ad sayısı.
<b>Windows</b> <a href="#">Library1</a>	NetBIOS DLL ' inin adı.
<b>QMErrorLog kısmı</b>	
<a href="#">ErrorLogBüyüküğü</a>	Yedeklemeye kopyalandığı kuyruk yöneticisi hata günlüğünün boyutunu belirtir.
<a href="#">ExcludeMessage</a>	Kuyruk yöneticisi hata günlüğüne yazılmayacak iletileri belirtir.
<a href="#">SuppressMessage</a>	Kuyruk yöneticisi hata günlüğüne yalnızca belirli bir zaman aralığında bir kez yazılan iletileri belirtir.



Çizelge 11. qm.ini dosyasının stanzas dosyası (devamı var)

Kıta ve öznitelikler	Özniteliklerin açıklaması
<u>SuppressInterval</u>	SuppressMessage içinde belirtilen iletilerin kuyruk yöneticisi hata günlüğüne yalnızca bir kez yazılacağı zaman aralığını saniye cinsinden belirtir.
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <span style="background-color: #4F81BD; color: white; padding: 2px 5px;">Linux</span> <span style="background-color: #4F81BD; color: white; padding: 2px 5px;">AIX</span> </div> <div> <b>Sınırlı Kip kısmı</b> <sup>2</sup> </div> </div>	
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <span style="background-color: #4F81BD; color: white; padding: 2px 5px;">Linux</span> <span style="background-color: #4F81BD; color: white; padding: 2px 5px;">AIX</span> </div> <div> <u>ApplicationGroup</u> </div> </div>	Uzak iletilerin yerleştirildiği yerel iletim kuyruğunun adı, bir iletim kuyruğu hedef için açık olarak tanımlanmamışsa.
<b>Güvenlik kısmı</b>	
<u>ClusterQueueAccessControl</u>	Küme kuyruklarının ya da küme kuyruğu yöneticilerinin bulundurduğu tam olarak nitelenmiş kuyrukların erişim denetimini denetleyin.
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <span style="background-color: #C00000; color: white; padding: 2px 5px;">Windows</span> </div> <div> <u>GroupModel</u> </div> </div>	Windows üzerinde bir kullanıcının grup üyeliğini belirlerken Nesne Yetkilisi Yöneticisi 'nin (OAM) genel grupları denetleyip denetlemeyeceğini belirler.
<b>Hizmet kısmı</b>	
<u>NAME</u>	Gerekli hizmetin adı.
<u>EntryPoints</u>	Hizmet için tanımlanan giriş noktalarının sayısı.
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <span style="background-color: #C00000; color: white; padding: 2px 5px;">Windows</span> </div> <div> <u>SecurityPolicy</u> </div> </div>	Windows üzerinde, her kuyruk yöneticisine ilişkin güvenlik ilkesi
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <span style="background-color: #4F81BD; color: white; padding: 2px 5px;">Linux</span> <span style="background-color: #4F81BD; color: white; padding: 2px 5px;">AIX</span> </div> <div> <u>SecurityPolicy</u> </div> </div>	AIX and Linux' da, kuyruk yöneticisinin kullanıcı tabanlı ya da grup tabanlı yetki kullanıp kullanmadığını belirler.  <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <span style="background-color: #4F81BD; color: white; padding: 2px 5px;">V 9.3.0</span> </div> <div>             IBM MQ 9.3.0' den işletim sistemi olmayan bir kullanıcı adı da oluşturabilirsiniz.           </div> </div>
<u>SharedBindingsUserId</u>	Yalnızca paylaşılan bağ tanımları için, MQZ_AUTHENTICATE_USER işlevindeki IdentityContext yapısındaki UserIdentifier alanının etkin kullanıcı kimliği mi, yoksa gerçek kullanıcı kimliği mi olduğu.
<u>FastpathBindingsUserId</u>	Yalnızca fastpath bağ tanımları için, IdentityContext yapısındaki MQZ_AUTHENTICATE_USER işlevindeki UserIdentifier alanının geçerli kullanıcı kimliği mi, yoksa gerçek kullanıcı kimliği mi olduğu.
<u>IsolatedBindingsUserId</u>	Yalnızca yalıtılmış bağ tanımları için, IdentityContext yapısındaki MQZ_AUTHENTICATE_USER işlevindeki UserIdentifier alanının geçerli kullanıcı kimliği ya da gerçek kullanıcı kimliği olup olmadığını belirler.
<b>ServiceComponent kısmı</b>	
<u>Hizmet</u>	Gerekli hizmetin adı.
<u>NAME</u>	Hizmet bileşeninin açıklayıcı adı.
<u>Modül</u>	Bu bileşene ilişkin kodu içerecek modülün adı.
<u>ComponentDataBoyutu</u>	Her çağrıda bileşene geçirilen bileşen veri alanının bayt cinsinden boyutu.

Çizelge 11. qm.ini dosyasının stanzas dosyası (devamı var)

Kıta ve öznitelikler	Özniteliklerin açıklaması
<b>Windows</b> SPX kısmı	
<b>Windows</b> Yuva	Onaltılı gösterimde SPX yuva numarası.
<b>Windows</b> BoardNum	LAN bağdaştırıcısı numarası.
<b>Windows</b> KeepAlive	KeepAlive işlevini açın ya da kapatın.
<b>Windows</b> Library1	SPX DLL ' nin adı.
<b>Windows</b> Library2	Kod iki ayrı kitaplıkta saklandıysa, LibraryName1ile aynı.
<b>Windows</b> ListenerBacklog	SPX dinleyicisi için bekleyen isteklerin varsayılan sayısını geçersiz kılın.
<b>SSL kısmı</b>	
<b>V 9.3.0</b> OutboundSNI	SNI yetenekli istemcilerin, bir TLS bağlantısı ya da anasistem adı başlatılırken uzak sisteme SNI ' yi hedef IBM MQ kanal adına ayarlayıp ayarlamayacağını belirler.
AllowOutboundSNI	SNI özellikli istemcilerin, bir TLS bağlantısı başlatılırken uzak sisteme SNI ' yi hedef IBM MQ kanal adına ayarlayıp ayarlamayacağını belirler.  <b>Uyarı:</b> <b>V 9.3.0</b> <b>Deprecated</b> IBM MQ 9.3.0 ' dan bu özellik kullanımdan kaldırılmıştır. Bunun yerine <b>OutboundSNI</b> kullanın.
AllowedCipherBelirtileri	Çoklu Platformlar üzerinde IBM MQ kanallarıyla kullanılmak üzere sipariş edilen ve etkinleştirilen özel bir CipherSpecs listesini belirtir.
AllowTLSV13	Bir kuyruk yöneticisinin TLS 1.3 CipherSpecs ' i kullanıp kullanamayacağı.
CDPCheckExtensions	Bu kuyruk yöneticisindeki TLS kanallarının CrLDistributionPoint sertifika uzantılarında adı belirtilen CDP sunucularını denetlemeye çalışıp çalışmadığını belirler.
MinimumRSAKeyBoyutu	RSA sertifikalarının kabul edilebilmesi için sahip olması gereken anahtar büyüklüğü alt sınırını belirler.
OCSPAAuthentication	Bir OCSP sunucusundan iptal durumu saptanamadığında yapılacak işlem.
OCSPCheckExtensions	Bu kuyruk yöneticisindeki TLS kanallarının AuthorityInfoErişim sertifikası uzantılarında adı belirtilen OCSP sunucularını denetlemeye çalışıp çalışmadığını belirler.
OCSPTimeout	İptal denetimi gerçekleştirilirken OCSP yanıtlayıcısının bekleneceği saniye sayısı.
<b>ALW</b> PeerCertChainValidation	IBM Global Security Kit (GSKit) sertifika geçerlilik denetimi ayarı.

Çizelge 11. qm.ini dosyasının stanzas dosyası (devamı var)

<b>Kıta ve öznitelikler</b>	<b>Özniteliklerin açıklaması</b>
<u>SSLHTTPProxyName</u>	OCSP denetimleri için GSKit tarafından kullanılacak HTTP Yetkili sunucusunun anasistem adı ya da ağ adresi.
<u>SSLHTTPConnectTimeout</u>	İptal denetimi gerçekleştirilirken bir HTTP sunucusuyla ağ bağlantısının başarıyla kurulması için beklenecek saniye sayısı.
<b>Alt havuz kısmı</b> "3" sayfa 108	Bu kıta IBM MQ tarafından oluşturulur. Bunu değiştirme.
<u>ShortSubpoolAdı</u> "3" sayfa 108	IBM MQ ' un çalışan işlemleri arasında iç iletişim için kullandığı /var/mqm/sockets dizininde yaratılan bir dizine ve simgesel bağlantıya karşılık gelen ad.
<b>TCP kısmı</b>	
<u>Kapı</u>	TCP/IP oturumları için ondalık gösterimde varsayılan kapı numarası.
 <u>Library1</u>	TCP/IP yuvaları DLL 'sinin adı.
<u>KeepAlive</u>	KeepAlive işlevini açın ya da kapatın.
<u>ListenerBacklog</u>	TCP/IP dinleyicisine ilişkin bekleyen isteklerin varsayılan sayısını geçersiz kılın.
<u>Connect_Timeout</u>	Yuvayı bağlama girişiminin zamanaşımına uğraması için geçen saniye sayısı.
<u>SndBuffBoyut</u>	Kanalların gönderilmesi tarafından kullanılan TCP/IP gönderme arabelleğinin bayt cinsinden büyüklüğü.
<u>RcvBuffBoyutu</u>	Alıcı kanallar tarafından kullanılan TCP/IP alma arabelleğinin bayt cinsinden büyüklüğü.
<u>RcvSndBuffSize</u>	Alıcı kanalının gönderen ucu tarafından kullanılan TCP/IP gönderme arabelleğinin bayt cinsinden büyüklüğü.
<u>RcvRcvBuffSize</u>	Bir alıcı kanalının alıcı sonu tarafından kullanılan TCP/IP alma arabelleğinin bayt cinsinden büyüklüğü.
<u>SvrSndBuffSize</u>	Bir istemci bağlantısı sunucu bağlantısı kanalının sunucu sonu tarafından kullanılan TCP/IP gönderme arabelleğinin bayt cinsinden büyüklüğü.
<u>SvrRcvBuffSize</u>	İstemci-bağlantı sunucusu bağlantı kanalının sunucu sonu tarafından kullanılan TCP/IP alma arabelleğinin bayt cinsinden büyüklüğü.
 <u>SecureComms</u>	Düz metin iletişimine izin verilip verilmeyeceğini, varsayılan değere izin verilip verilmeyeceğini belirtir.
<b>Ayarlama değiştirgeleri kısmı</b>	
<u>SuppressDspAuthFail</u>	Bir yetki denetimi başarısız olduğunda, bağlantı bir nesne için + dsp yetkisine sahip değilse, kuyruk yöneticisinin yetki olaylarının oluşturulmasını ve AMQ8077 hata iletilerinin hata günlüğüne yazılması engellenip engellenmeyeceğini belirler.

Çizelge 11. qm.ini dosyasının stanzas dosyası (devamı var)

<b>Kıta ve öznitelikler</b>	<b>Özniteliklerin açıklaması</b>
<u>ImplSyncOpenOutput</u>	Örtülü bir eşitleme noktası, eşitleme noktasının dışında kalıcı bir koyma için etkinleştirilmeden önce, kuyruğa konmak üzere açık olan uygulama sayısı alt sınırı.
<u>UniformClusterAdı</u>	Tek tip küme olarak kullanmakta olduğunuz IBM MQ kümesinin adı.
<u>OAMLdapConnectZamanaşımı</u>	LDAP istemcisinin sunucuyla TCP bağlantısı kurmak için bekleyeceği saniye cinsinden süre üst sınırı.
<u>OAMLdapQueryTimeLimit</u>	LDAP istemcisinin sunucudan bir LDAP isteğine yanıt almak için bekleyeceği saniye cinsinden süre üst sınırı.
<b>V 9.3.2</b> <u>OAMLdapResponseWarningTime</u>	Bir LDAP sunucusuna yönelik bağlantı <b>OAMLdapResponseWarningTime</b> parametresi tarafından belirlenen eşik sayısından daha uzun sürerse, hata günlüğüne bir <u>AMQ5544W</u> iletisi yazılır.
<u>ExpiryInterval</u>	Kuyruk yöneticisinin, diğer kuyruk etkinlikleri tarafından henüz temizlenmemiş olan süresi dolmuş iletileri arayan kuyrukları tarama sıklığını belirtir. Saniye cinsinden bir zaman aralığıdır.
<b><u>LivenessHeartBeatLen</u></b>	Kuyruk yöneticisinin günlüğe yazma işlemlerini makul bir hızda denetleme sıklığını yapılandırır.
<b><u>ECHearBeatLen</u></b>	Kuyruk yöneticisi genel sağlıklı işletim denetimlerinin sıklığını yapılandırır.
<b><u>FileLockHeartBeatLen</u></b>	Dosya kilidine ilişkin varsayılan değeri, yürütme denetleyicisinin birincil çok eşgörümlü dosyada dışlayıcı kilidi tutmaya devam ettiğinden emin olmak için belirli aralıklarla gerçekleştirdiği çok eşgörümlü bir kuyruk yöneticisi olup olmadığını denetler.
<b>Değişkenler kısmı</b>	
<u>öznitelik=değer</u>	MQSC tanımlamaları sırasında araya ekleme olarak kullanılacak ad ve ilişkili değer.
<b>XAResourceManager kısmı</b>	
<u>NAME</u>	Kaynak yöneticisi yönetim ortamı.
<u>SwitchFile</u>	Kaynak yöneticisinin XA anahtar yapısını içeren yükleme dosyasının tam olarak nitelenmiş adı.
<u>XAOpenString</u>	Kaynak yöneticisinin xa_open giriş noktasına geçirilecek veri dizgisi.
<u>XACloseString</u>	Kaynak yöneticisinin xa_close giriş noktasına geçirilecek veri dizgisi.
<u>ThreadOfDenetim İş Parçacığı</u>	Kuyruk yöneticisinin kaynak yöneticisini kendi çok iş parçacıklı işlemlerinden birinden çağırması gerektiğinde diziselleştirme için kullandığı değer. Windows için zorunludur.

**Notlar:**

1. AccessMode kısmı, **crtmqm** komutundaki **-a [r]** seçeneği tarafından ayarlanır. Kuyruk yöneticisi yaratıldıktan sonra AccessMode (Erişim Kipi) kısmı değiştirilmez.
2. RestrictedMode kısmı, **crtmqm** komutundaki **-g** seçeneği tarafından ayarlanır. Kuyruk yöneticisi yaratıldıktan sonra bu kısmı değiştirmeyin. **-g** seçeneğini kullanmazsanız, **qm.ini** dosyasında stanza oluşturulmaz.
3. Bir kuyruk yöneticisi yarattığınızda, Althavuz kısmı ve o bölüm içindeki ShortSubpoolözneteliği IBM MQ tarafından otomatik olarak yazılır. IBM MQ , ShortSubpoolAdı için bir değer seçer. Bu değeri değiştirmeyin.

## Windows **qm.ini** dosyasının AccessMode kısmı

Erişim Kipi yalnızca Windows sunucuları için geçerlidir. **qm.ini** dosyasının AccessMode kısmı, **crtmqm** komutundaki **-a [ r]** seçeneği tarafından ayarlanır. Kuyruk yöneticisi yaratıldıktan sonra AccessMode (Erişim Kipi) kısmı değiştirilmez.

Erişim grubunu kullan ( **-a [ r]** ) Üyeleri tüm kuyruk yöneticisi veri dosyalarına tam erişim verilecek bir Windows güvenlik grubu belirtmek için **crtmqm** komutu seçeneği. Grup, kullanılan sözdizimine bağlı olarak yerel ya da genel bir grup olabilir. Grup adı için geçerli sözdizimi şöyledir:

```
LocalGroup
Etki alanı adı\GlobalGroup adı
GlobalGroup adı @ Etki alanı adı
```

**crtmqm** komutunu **-a [ r]** seçeneğiyle çalıştırmadan önce ek erişim grubunu tanımlamanız gerekir.

Grubu **-ayerine -ar** kullanarak belirtirseniz, yerel **mqm** grubuna kuyruk yöneticisi veri dosyalarına erişim izni verilmez. Kuyruk yöneticisi veri dosyalarını barındıran dosya sistemi, yerel olarak tanımlanan gruplar için erişim denetimi girişlerini desteklemiyorsa bu seçeneği kullanın.

Grup genellikle, çok eşgörümlü kuyruk yöneticilerinin paylaşılan bir kuyruk yöneticisi verilerine ve günlüklerine erişmelerini sağlamak için kullanılan genel bir güvenlik grubudur. Klasörde okuma ve yazma izinlerini ayarlamak ya da kuyruk yöneticisi verilerini ve günlük dosyalarını içeren paylaşım için ek güvenlik erişimi grubunu kullanın.

Ek güvenlik erişimi grubu, kuyruk yöneticisi verilerini ve günlüklerini içeren klasör için izinleri ayarlamak üzere **mqm** adlı yerel grubu kullanmanın alternatifidir. **mqmyerel** grubundan farklı olarak, ek güvenlik erişimi grubunu yerel ya da genel bir grup yapabilirsiniz. Çok eşgörümlü kuyruk yöneticileri tarafından kullanılan veri ve günlük dosyalarını içeren paylaşılan klasörlere ilişkin izinleri ayarlamak için bu bir genel grup olmalıdır.

Windows işletim sistemi, kuyruk yöneticisi verilerini ve günlük dosyalarını okumak ve yazmak için erişim izinlerini denetler. Kuyruk yöneticisi işlemlerini çalıştıran kullanıcı kimliğinin izinlerini denetler. Denetlenen kullanıcı kimliği, kuyruk yöneticisini hizmet olarak mı başlattığınıza, yoksa etkileşimli olarak mı başlattığınıza bağlıdır. Kuyruk yöneticisini hizmet olarak başlattıysanız, Windows sistemi tarafından denetlenen kullanıcı kimliği, **Hazırla** IBM MQ sihirbazıyla yapılandırıldığınız kullanıcı kimliğidir. Kuyruk yöneticisini etkileşimli olarak başlattıysanız, Windows sistemi tarafından denetlenen kullanıcı kimliği, **stmqm** komutunu çalıştıran kullanıcı kimliğidir.

Kuyruk yöneticisini başlatmak için kullanıcı kimliği yerel **mqm** grubunun bir üyesi olmalıdır. Kullanıcı kimliği ek güvenlik erişimi grubunun üyesiyse, kuyruk yöneticisi grubu kullanarak izin verilen dosyaları okuyabilir ve yazabilir.

**Sınırlama:** Yalnızca Windows işletim sisteminde ek bir güvenlik erişimi grubu belirtebilirsiniz. Diğer işletim sistemlerinde ek bir güvenlik erişimi grubu belirtirseniz, **crtmqm** komutu bir hata döndürür.

## Örnek kısmı

```
AccessMode:
SecurityGroup=wmq\wmq
```



## İlgili kavramlar

[“Windows üzerinde paylaşılmayan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğini sağlayın” sayfa 528](#)

[“Windows üzerinde paylaşılan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğinin sağlanması” sayfa 525](#)

## İlgili görevler

[“Windows üzerinde etki alanı iş istasyonlarında ya da sunucularda çok eşgörunümlü kuyruk yöneticisi yaratılması” sayfa 499](#)

## İlgili başvurular

[crtmqm \(kuyruk yöneticisi yarat\)](#)

Multi

## ApiExit qm.ini dosyasının yerel kısmı

ApiExitYerel kısmı, bir kuyruk yöneticisine ilişkin API çıkış yordamlarını belirtir.

Bir sunucu için, qm. ini dosyasının ApiExitYerel kısmı, bir kuyruk yöneticisine ilişkin API çıkış yordamlarını tanıttak şekilde deęiřtirin.

Linux

Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer Exits kuyruk yöneticisi özellikleri sayfasını kullanın.

Bir istemci için, mqclient. ini dosyasındaki ApiExitLocal (Yerel) kısmı, bir kuyruk yöneticisine ilişkin API çıkış yordamlarını tanıttak şekilde deęiřtirin.

## Genel Bakış

ApiExitLocal kısmı yalnızca tek bir Module belirtilmesine izin verir; ancak, ařaęıdaki gibi dört modül sağlanmalıdır:

- 32 bit diřsiz
- 32 bit iş parçacığı
- 64 bit iş parçacıklı deęil
- 64 bit iş parçacığı

Çıkışın iş parçacıklı sürümünü tanımlamak için sağlanan modül adının sonuna IBM MQ eklendiğini \_r , ancak IBM MQ ' nin 32 bit ve 64 bit deęiřkeleri için doğrudan eşdeęer bir düzenek sağlamadığını unutmayın.

*prefix*/mqm/samp/bin içinde gönderilen amqsaxe0 ve amqsaxe0\_r sürümleri, oluşturuldukları platformda kuyruk yöneticisinin yerel boyutu (řimdi 64 bit) için oluşturulur ve yalnızca aynı yerel boyutta çalışın uygulamalar tarafından kullanılabilir.

Nitelenmemiş bir modül adı sağlanırsa, 32 bit deęiřkeleri için IBM MQ /var/mqm/exits ve 64 bit deęiřkeleri için /var/mqm/exits64 içine bakar

Örneğin, module=amqsaxe ařaęıdakileri belirtir:

```
/var/mqm/exits/amqsaxe - 32 bit unthreaded variant
/var/mqm/exits/amqsaxe_r - 32 bit threaded variant
/var/mqm/exits64/amqsaxe - 64 bit unthreaded variant
/var/mqm/exits64/amqsaxe_r - 64 bit threaded variant
```

Windows

Windows sistemlerinde, API çıkışlarına ilişkin girişleri deęiřtirmek için **amqmdain** komutunu da kullanabilirsiniz. (Tüm kuyruk yöneticilerine ilişkin API çıkış yordamlarını tanımlamak için, [“ApiExit mqm.ini dosyasının ortak ve ApiExitŞablon kısmı” sayfa 91](#) içinde açıklandığı gibi ApiExitCommon ve ApiExitTemplate kısımları kullanılır.)

API çıkışının doğru çalışması için, sunucudan gelen iletinin istemciye dönüřtürölmemiş olarak gönderilmesi gerektiğini unutmayın. API çıkışı iletiyi işledikten sonra, iletinin istemcide dönüřtürölmesi gerekir. Bu nedenle, istemcide tüm dönüřtürme çıkışlarını kurmuş olmanız gerekir.

Bu öznitelikleri kullanma hakkında daha fazla bilgi için [API çıkışlarını yapılandırmabaşlıklı](#) konuya bakın.

## Parametreler

### Ad=ApiExit\_name

API çıkışının açıklayıcı adı, MQAXP yapısının ExitInfoAd alanında iletilir.

Bu ad benzersiz olmalı, 48 karakterden uzun olmamalı ve yalnızca IBM MQ nesnelерinin adları (örneğin, kuyruk adları) için geçerli karakterler içermelidir.

### İşlev=işlev\_adi

API çıkış kodunu içeren modüldeki işlev giriş noktasının adı. Bu giriş noktası MQ\_INIT\_EXIT işlevidir.

Bu alanın uzunluğu MQ\_EXIT\_NAME\_LENGTH ile sınırlıdır.

### Module=modül\_adi

API çıkış kodunu içeren modül.

Bu alan modülün tam yol adını içeriyorsa, olduğu gibi kullanılır. Bu alan yalnızca modül adını içeriyorsa, modül, qm.ini dosyasının ExitPath kısmında **ExitsDefaultPath** özneliği kullanılarak bulunur.

Ayrı iş parçacıklı kitaplıkları destekleyen platformlarda, API çıkış modülünün hem iş parçacıklı olmayan hem de iş parçacıklı bir sürümünü sağlamanız gerekir. İş parçacıklı sürümün bir \_r soneki olmalıdır. IBM MQ uygulama sınırlı kod öbeğinin iş parçacıklı sürümü, yüklenmeden önce belirtilen modül adının sonuna \_r örtük olarak eklenir.

Bu alanın uzunluğu, altyapının desteklediği yol uzunluğu üst sınırıyla sınırlıdır.

### Veri=veri\_adi

MQAXP yapısının ExitData alanında API çıkışına geçirilecek veriler.

Bu özneliği eklerse, baştaki ve sondaki boşluklar kaldırılır, kalan dizgi 32 karaktere kesilir ve sonuç çıkışa geçirilir. Bu özneliği atlarsanız, çıkışa varsayılan değer olan 32 boşluk geçirilir.

Bu alanın uzunluk üst sınırı 32 karakterdir.

### Sequence=sequence\_number (Sıralama\_numarası)

Bu API çıkışının diğer API çıkışlarıyla görel olarak çağrıldığı sıra. Düşük sıra numarasına sahip bir çıkış, daha yüksek sıra numarasına sahip bir çıkıştan önce çağrılır. Çıkışların sıra numaralandırmasının bitişik olması gerekmez. 1, 2, 3 dizisinin sonucu 7, 42, 1096 ile aynıdır. İki çıkış aynı sıra numarasına sahipse, kuyruk yöneticisi hangisinin önce aranacağına karar verir. MQAXP 'de ExitChainAreaPtr ile gösterilen ExitChainAlanına zamanı ya da bir imleyiciyi koyarak ya da kendi günlük dosyanızı yazarak, olaydan sonra hangisinin çağrıldığı anlayabilirsiniz.

Bu öznelik, işaretsiz bir sayısal değerdir.

## Örnek kısmı

```
ApiExitLocal:  
Name=ClientApplicationAPIchecker  
Sequence=3  
Function=EntryPoint  
Module=/usr/Dev/ClientAppChecker  
Data=9.20.176.20
```

### Linux AIX V9.3.4 qm.ini dosyasının AuthToken kısmı

Kuyruk yöneticisini, uygulamalara bağlanarak sağlanan kimlik doğrulama belirteçlerini doğrulayacak şekilde yapılandırmak için **AuthToken** kısmı kullanın.

## AuthToken kısmı

### KeyStore= dizgi

Güvenilen sertifika veren genel anahtar sertifikalarını ve simetrik anahtarlarını içeren anahtar deposu için dosya yolu. Anahtarları var olan bir anahtar deposuna ekleyebilir ya da yeni bir anahtar deposu yaratabilirsiniz. Daha fazla bilgi için [Kimlik doğrulama belirteçlerini kabul etmek](#)

için kuyruk yöneticisini yapılandırmabaşlıklı konuya bakın. Kuyruk yöneticisi, uygulamanın sunduğu kimlik doğrulama simgesinin güvenilir sertifika veren tarafından imzalandığını doğrulamak için anahtar deposundaki anahtarları kullanır.

Dosya uzantısı .kdb olan bir CMS anahtar deposunu ya da dosya uzantısı .p12olan bir PKCS#12 anahtar deposunu kullanabilirsiniz. Anahtar deposu dosyası yoksa ya da dosyaya erişilemiyorsa, bir AMQ7076E: ini dosyasında öznitelik için geçersiz değer hatası, kuyruk yöneticisinin hata günlüğüne yazılır.

Anahtar deposu tipinin, anahtar deposuna ilişkin dosya adı uzantısıyla eşleştiğini doğrulayın. IBM MQ anahtar deposunun doğru biçimini saptar; ancak, anahtar deposu tipi ve dosya adı uzantısı eşleşmezse, tutarsızlıklar diğer yönetim sorunlarına neden olabilir.

Anahtar deposu dosya yolu uzunluğu üst sınırı 256 karakterdir.

### **KeyStorePwdFile= dizgi**

Anahtar deposu için şifrelenmiş parolayı içeren dosyanın dosya yolu. Dosya, şifrelenmiş parolayı tek bir metin satırı olarak içermelidir. Düz metin parolaları kabul edilmez.

Parolayı anahtar deposu parola dosyasına kaydetmeden önce şifrelemek için **runqmcrcd** komutunu kullanın. Anahtar deposu parola dosyası yalnızca **runqmcrcd** komutu çalıştırılarak oluşturulan şifrelenmiş parolayı içermelidir.

Şifrelenmeden önce düz metin parolası uzunluğu üst sınırı 1024 karakterdir.

Bu parametre isteğe bağlıdır. Sağlanmazsa, kuyruk yöneticisi, parola aynı dizinde ve anahtar deposuyla aynı ada sahip, ancak .sthdosya uzantısına sahip bir parola saklama dosyası arar. Saklama dosyası bulunamazsa, yapılandırma reddedilir ve AMQ7006E hata iletisi kuyruk yöneticisinin hata günlüğüne yazılır. Anahtar deposu parolalarını saklama seçenekleriyle ilgili ek bilgi için [Anahtar havuzu parolalarının şifrelenmesibaşlıklı konuya](#) bakın.

Parola dosyası yolunun uzunluk üst sınırı 256 karakterdir.

### **CertLabel= dizgi**

Kimlik doğrulama belirteçlerini doğrulamak için kullanılan anahtar deposundaki bir genel anahtar sertifikasına ya da simetrik anahtara ilişkin sertifika etiketi. **CertLabel** özneliğini yineleyerek en çok 32 sertifika etiketi sağlayabilirsiniz.

Kuyruk yöneticisi anahtar deposuna sertifika eklediğinizde, bunlara anlamlı etiketler verin. Sertifika etiketleri büyük ve küçük harfe duyarlıdır. Bunlar alfasayısal karakterler, noktalama işaretleri ve boşluklar içerebilir. Geçersiz bir karakter algılanırsa, bir hata döndürülür ve IBM MQ hata günlüğüne bir hata iletisi yazılır.

Güvenilir belirteç veren birden çok genel anahtar sertifikası ve simetrik anahtar sağlayabilir. Örneğin, genel anahtar sertifikalarının geçerlilik dönemleri vardır. Süre bitimine yaklaştıklarında belirteç veren, yeni bir süre bitim tarihi içeren yeni bir sertifika sağlar. Bir süre için, her iki sertifika da geçerli olabilir.

Uygulamalar kimlik doğrulaması için belirteç sunduğunda, simgeyi imzalamak için kullanılan geçerli bir anahtar bulununcaya kadar **CertLabels** listesi denetlenir. Eşleşme bulunursa, belirteç imzası doğrulanır.

**CertLabel** belirtilmezse, simgeyi gösteren uygulamadaki bağlantı 2063

MQRC\_SECURITY\_ERRORneden koduyla başarısız olur ve AMQ5786E: Kimlik doğrulama simgesi yapılandırma hatası iletisi kuyruk yöneticisinin hata günlüğüne yazılır.

Sertifika etiketi uzunluğu üst sınırı 64 karakterdir.

Örneğin,

```
AuthToken:  
KeyStore=/var/mqm/qmgrs/qmgrs/qm1/tokenissuer/key.kdb  
KeyStorePwdFile=/var/mqm/qmgrs/qm1/tokenissuer/key.pw  
CertLabel=token  
CertLabel=rsaakey  
CertLabel=mark  
... up to 32 CertLabel fields
```

### **UserClaim= dizgi**

Kuyruk yöneticisinin yetki denetimi için benimsediği kullanıcı kimliğini içeren simge içindeki talep.

Kuyruk yöneticisi **ADOPTCTX(NO)** ile yapılandırıldıysa, bu parametre isteğe bağlıdır. **ADOPTCTX(YES)** kullanılırsa, bu parametre gereklidir. **ADOPTCTX** , kuyruk yöneticisinin **CONNAUTH** özneliği tarafından başvuru kimlik doğrulama bilgileri (AUTHINFO) nesnesinde bulunan bir özneliktir.

Bir kimliği benimsemek için belirteç, **AuthToken** stanza 'nın **UserClaim** özneliğinde belirtilen ada sahip bir talep içermelidir ve **ADOPTCTX(YES)** kullanılmalıdır.

Örneğin, belirteciniz bir talep içeriyorsa "AppUser": "MyUserName", yetkilendirme için qm.ini dosyasının AuthToken (Kimlik Doğrulama Simgesi) kısmı içinde UserClaim=AppUser belirtmeniz gerekir "MyUserName" .

**UserClaim** öznelik değerinin uzunluk üst sınırı 128 karakterdir.

**Not:** IBM MQ 9.3.4' den AuthToken kısmı belirtilirse, Hizmet kısmına ilişkin **SecurityPolicy** özneliğinin geçerli değeri UserExternalolarak ayarlanır. **SecurityPolicy** , Hizmet kısmı içinde belirttik olarak Grup olarak ayarlanırsa belirteç kimlik doğrulaması kullanılamaz. **SecurityPolicy** Grupolarak ayarlanırsa, **SecurityPolicy** özneliğini Hizmet dörtünden kaldırın ve kuyruk yöneticisini yeniden başlatın. Daha fazla bilgi için bkz. [SecurityPolicy](#).

**Not:** Kimlik doğrulama bilgileri nesnesinin **ADOPTCTX** özneliğini kullanarak, simgedeki kullanıcı kimliğinin yetkilendirme denetimleri için kullanılıp kullanılmayacağını denetleyin. Kuyruk yöneticisini yarattığınızda, bu öznelik **ADOPTCTX(YES)** olarak ayarlanır. Bu değer, simgedeki kullanıcı kimliğinin benimsenmesine neden olur. Kullanıcı kimliği, kimlik doğrulama belirteçlerinde kullanıcı kimliklerine ilişkin gereksinimleri karşılamalıdır. Daha fazla bilgi için bkz. [Kimlik doğrulama belirteçlerinde kullanıcı kimlikleri](#). Belirteç kullanıcı talebi, gereksinimleri karşılamayan bir kullanıcı kimliği içeriyorsa, bağlantı **2035 MQRC\_NOT\_AUTHORIZED** neden koduyla reddedilir. **ADOPTCTX(NO)** ayarlanırsa, belirteç yalnızca kimlik doğrulaması için kullanılır ve yetkilendirme için başka bir kullanıcı kullanılmalıdır.

#### **AllowOSGroups=NO (varsayılan) |YES**

Varsayılan değer Hayır'dır. Bir simgeden benimsenen bir kimliğin işletim sistemi (OS) kullanıcısı olarak kabul edilip edilmeyeceğini ve eşleşen işletim sistemi kullanıcısının grup üyeliklerinin yetkilendirme sırasında kabul edilip edilmeyeceğini belirler.

#### **AllowOSGroups= NO | N**

Yetki denetimleri yalnızca, simgeden benimsenen kullanıcının adını temel alır.

#### **AllowOSGroups= YES | Y**

Yetki denetimleri kullanıcının adına bağlıdır ve ait olabileceği gruplar da denetlenir.

## **Örnek kısmı-yalnızca kimlik doğrulaması**

**AuthToken** kısmı yalnızca gerekli en az iki parametreyle geçerli olabilir:

- **KeyStore** dosya yolu ve
- **CertLabel** adı.

```
AuthToken:
  KeyStore=/var/mqm/qmgrs/qmgrs/qm1/tokenissuer/key.kdb
  CertLabel=token
  ... up to 32 CertLabel fields
```

En az iki parametre eklediyseniz, o zaman:

- Anahtar deposu parola dosyasının gerekli olmaması için, şifrelenmiş anahtar deposu parolasıyla birlikte key.sth adlı bir parola saklama dosyası var olmalıdır.
- Belirteç, yetkilendirme için IBM MQ ' e geçirilecek bir kullanıcı adı içermiyor. Uygulama bağlanabilir ve kimliği doğrulanabilir, ancak uygulamanın bağlandıktan sonra çalışması için yetki sağlayabilmesi için farklı bir mekanizma bulunmalıdır.

Kuyruk yöneticinizin yapılandırmasına bağlı olarak, yetkilendirme için kullanılan kullanıcı adı, MCA kuralları aracılığıyla kanalda tanımlanan kullanıcı adı ya da istemci uygulamasının sunucunuzda var olabileceği şekilde çalıştırdığı kullanıcı adı olabilir ve yetkililere sahip gruplara ait olabilir. Belirteçleri kullanırken şunu unutmayın:

- Kuyruk yöneticiniz **UserExternal** kipine geçiriliyor; bu, kuyruk yöneticisinin çalıştığı işletim sisteminde varolmayan kullanıcıların kimlik doğrulaması için kullanılabilmesi anlamına geliyor.
- **AuthToken** qm.ini kısmına **AllowOSGroups** seçeneğini eklemesiniz de, varsayılan değer Hayır olarak ayarlanır. Bu nedenle, **UserClaim** değerini eklersiniz, ancak **AllowOSGroups=Evet** değerini belirtmesiniz, yetki için benimsenen simge kullanıcısı, kuyruk yöneticisinin çalıştığı işletim sisteminde ait olabileceği gruplar için denetlenmez.

## Örnek kıta-kimlik doğrulama ve yetkilendirme

Tüm **AuthToken** değiştirgelerini tanımlayabilirsiniz:

- **KeyStore** dosya yolu,
- **KeyStorePwdFile** dosya yolu,
- **CertLabel** adı,
- **UserClaim** adı ve
- **AllowOSGroups** seçeneğiyle aynı.

```
AuthToken:
  KeyStore=/var/mqm/qmgrs/qmgrs/QMJWT/ssl/key.kdb
  KeyStorePwdFile=/var/mqm/qmgrs/QMJWT/ssl/key.pw
  CertLabel=token
  CertLabel=rsaakey
  CertLabel=mark
  ... up to 32 CertLabel fields
  UserClaim=AppUser
  AllowOSGroups=Y
```

Kullanılabilir tüm parametreleri eklediyseniz, o zaman:

- **runqmcrcd** komutunu kullanarak anahtar deposu parolasını şifreleyin. Dosyayı bir dosyaya kaydedin ve dosya yolunu **AuthToken** kısmı içine ekleyin.
- Kimlik doğrulama belirteci kullanıcı talebindeki kullanıcı adı hem kimlik doğrulama hem de yetkilendirme için kullanılır.
  - Simge kullanıcısı, kuyruk yöneticisinin çalıştığı işletim sisteminde bir kullanıcı olarak var olabilir.
  - Kullanıcı denetimini etkinleştirmek için bir kimlik doğrulama bilgileri nesnesi tanımladınız.
  - Kanal kimlik doğrulaması ya da MCA kurallarına dayalı olarak, IBM MQ nesneleriyle etkileşim yetkisi olan bir kullanıcıyı benimseyecek kanal kimlik doğrulama kayıtlarını ayarlarsanız.

Belirteç kullanıcılarını doğrulama ve yetkilendirme stratejiniz, gereksinimlerinize ve IBM MQ kuyruk yöneticilerinizin önceden nasıl yapılandırıldığına bağlıdır. Daha fazla bilgi için [Kimlik doğrulama belirteçleriyle çalışmabaşlıklı konuya](#) bakın.

### İlgili kavramlar

[Belirteçlerle çalışma](#)

### İlgili görevler

[AuthToken](#) kuyruk yöneticisini kabul edecek şekilde yapılandırma

[Uygulamada kimlik doğrulama belirteçlerini kullanma](#)

## Multi qm.ini dosyasının AutoCluster kısmı

AutoCluster (Otomatik Küme) kısmı, kuyruk yöneticisi kümenin otomatik bir kümenin üyesi olup olmadığını tanımlamaya başladığında kullanılır ve kümenin tam havuzlarını tanımlayabilir.

AutoCluster (Otomatik Küme) kısmı için aşağıdaki öznitelikler zorunludur:

### Tip = Tek biçimli

Otomatik kümenin tipini belirtir ve geçerli tek seçenek tek bir kümeyi gösteren *Tek biçimli* seçeneğidir.

### ClusterName=< Dizi >

Kümenin adı; otomatik küme adıdır.

AutoCluster (Otomatik Küme) kısmı için aşağıdaki öznitelikler isteğe bağlıdır, ancak bunları çift olarak sağlamanız gerekir:

**HavuzAdı1 =< Dizgi>**

Bu, otomatik kümedeki ilk tam havuzun kuyruk yöneticisi adıdır. Bu, bu kuyruk yöneticisinin adı ya da başka bir ad olabilir.

**Repository1Conname=< Bağlantı Adı Dizesi >**

Bu, otomatik küme üyelerinin bu kuyruk yöneticisine nasıl bağlanmaları gerektiğine ilişkin bağlantı adı (CONNAME) değeridir.

**Havuz2Adı=< Dizgi>**

Bu, otomatik kümedeki ikinci tam havuzun kuyruk yöneticisi adıdır. Bu, bu kuyruk yöneticisinin adı ya da başka bir ad olabilir.

**Repository2Conname=< Bağlantı adı dizgisi >**

Bu, otomatik küme üyelerinin bu kuyruk yöneticisine nasıl bağlanmaları gerektiğine ilişkin bağlantı adı (CONNAME) değeridir.

## Örnek kısmı

```
AutoCluster:
  Repository1Name=QM1
  Repository2Name=QM2
  Repository1Conname=127.0.0.1(1414)
  Repository2Conname=127.0.0.1(1415)
  ClusterName=UNIFORMCLUSTER1
  Type=Uniform
```

### İlgili kavramlar

[“Otomatik uygulama dengeleme” sayfa 403](#)

Otomatik uygulama dengelemesi, IBM MQ tek tip kümesinin rasgele hale getirme ya da uygulamaların belirli kuyruk yöneticilerine el ile sabitlenmesi yerine küme genelinde uygulama dağıtımını yakından yönetmesini sağlayarak uygulama dağıtımını ve kullanılabilirliğini büyük ölçüde geliştirir.

### İlgili görevler

[“Yeni bir birörnek küme oluşturma” sayfa 415](#)

Yeni bir birörnek kümeyi nasıl oluşturduğunuz.

### İlgili başvurular

[“Otomatik küme yapılandırmasının kullanılması” sayfa 419](#)

IBM MQ yapılandırmasını, qm . ini yapılandırma bilgilerini değiştirerek otomatik yapılandırmayı etkinleştirecek şekilde yapılandırın.

## Multi qm.ini dosyasının AutoConfig kısmı

AutoConfig kısmının öznitelikleri, tek biçimli kümelerin ayarlanmasının bir parçası olarak sık sık kullanılır.

**Not:** Tek biçimli kümeler için yalnızca AutoCluster (Otomatik Küme) kılımını kullanabilirsiniz.

**MQSCConfig=< Yol>**

Yol, tam dosya yolu ya da tüm \* .mqsc dosyalarının her kuyruk yöneticisi başlangıcında kuyruk yöneticisine uygulandığı bir dizinin yoludur.

Daha fazla bilgi için bkz. [Başlangıçta MQSC komut dosyasından otomatik yapılandırma](#).

**IniConfig=< Yol>**

Yol, tam dosya yolu ya da tüm \* .ini dosyalarının her kuyruk yöneticisi başlangıcında qm . ini dosyasına uygulandığı bir dizinin yoludur.

Daha fazla bilgi için bkz. [“Başlangıçta qm.ini otomatik yapılandırması” sayfa 97](#).

## V 9.3.0 ConfigTimeout

Kuyruk yöneticisinin otomatik yapılandırmanın tamamlanmasını beklediği değer (saniye cinsinden). Bu süre bittikten sonra, kuyruk yöneticisi başlatılmaya devam eder ve uygulamaların bağlanması için kullanılabilir.

Varsayılan davranış zamanaşımı olmamaktır. Bu, tüm otomatik yapılandırma komutları tamamlanmaya kadar, uygulamaların bağlanması için kuyruk yöneticisinin kullanılmadığı anlamına gelir.

Bu özneliği, yapılandırma uzun sürdüğü için yapılandırmamalısınız; çünkü uygulamalar, kendileri için geçerli olan yapılandırmadan (örneğin, uygulamanın gerektirdiği kuyrukların yaratılması) önce bağlanabilir.

## Örnek kısmı

```
AutoConfig:
MQSCConfig=/tmp/auto.mqsc
IniConfig=/tmp/auto.ini
ConfigTimeout=120
```

### İlgili kavramlar

[“Otomatik uygulama dengeleme” sayfa 403](#)

Otomatik uygulama dengelemesi, IBM MQ tek tip kümesinin rasgele hale getirme ya da uygulamaların belirli kuyruk yöneticilerine el ile sabitlenmesi yerine küme genelinde uygulama dağıtımını yakından yönetmesini sağlayarak uygulama dağıtımını ve kullanılabilirliğini büyük ölçüde geliştirir.

### İlgili görevler

[“Yeni bir birörnek küme oluşturma” sayfa 415](#)

Yeni bir birörnek kümeyi nasıl oluşturduğunuz.

### İlgili başvurular

[“Otomatik küme yapılandırmasının kullanılması” sayfa 419](#)

IBM MQ yapılandırmasını, qm.ini yapılandırma bilgilerini değiştirerek otomatik yapılandırmayı etkinleştirecek şekilde yapılandırın.

### Multi

## qm.ini dosyasının kanal kısmı

Kanal kısmı öznelikleri, bir kanalın yapılandırmasını belirler.

### z/OS

Bu bilgiler IBM MQ for z/OS için geçerli değildir.

Kanallara ilişkin bilgileri belirtmek için qm.ini dosyasındaki CHANNELS kısmı kullanın.

### Linux

### Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer Channels kuyruk yöneticisi özellikleri sayfasını kullanın.

### MaxChannels = 100 (varsayılan) |sayı

İzin verilen yürürlükteki kanal sayısı üst sınırı.

Varsayılan değer 100 değeridir.

Gerekliyse, geçerli kanal sayısı üst sınırını sınırlamak için **MaxChannels** değerini farklı bir değere ayarlayabilirsiniz. IBM MQ Appliance için, varsayılan değer 999 999 999 'dur ve değiştirilmemelidir.

### MaxActiveKanalları = MaxChannels\_value

Herhangi bir zamanda *etkin* olmasına izin verilen kanal sayısı üst sınırı. Varsayılan değer, **MaxChannels** özneliği için belirtilen değerdir.

### MaxInitiators = 3 (varsayılan) |sayı

Başlatıcı sayısı üst sınırı. Varsayılan değer ve üst sınır değeri 3 'tür.



## **MQIBindType= FASTPATH | STANDARD**

Uygulamalara ilişkin bağ tanımı:

### **FastPath**

Kanallar MQCONNX FASTPATH kullanılarak bağlanır; aracı işlemi yoktur.

### **Standart**

Kanallar STANDARD kullanılarak bağlanır.

## **PipeLineLength = 1 |sayı**

Bir kanalın kullanacağı koşut zamanlı iş parçacığı sayısı üst sınırı. Varsayılan değer 1'dir. 1 'den büyük herhangi bir değer 2 olarak değerlendirilir.

Boruhattı kullandığınızda, kanalın her iki ucundaki kuyruk yöneticilerini 1 'den büyük bir

**PipeLineLength** olacak şekilde yapılandırın.

**Not:** Boru hattı yalnızca TCP/IP kanalları için geçerlidir.

Ek bilgi için [Birden çok iş parçacığı desteği-veri bağlantısı](#) konusuna bakın.

## **AdoptNewMCA = NO (varsayılan) | SVR | SDR | RCVR | CLUSRCVR | ALL | FASTPATH**

IBM MQ bir kanalı başlatma isteği alırsa, ancak kanalın bir örneğinin çalışmakta olduğunu bulursa, bazı durumlarda yenisinin başlatılabilmesi için var olan kanal örneğinin durdurulması gerekir.

**AdoptNewMCA** özneliği, hangi kanal tiplerinin bu şekilde sonlandırılabilceğini denetlemenizi sağlar.

Belirli bir kanal tipi için **AdoptNewMCA** özneliğini belirtirseniz, ancak eşleşen bir kanal örneği çalışmakta olduğu için yeni kanal başlatılamazsa:

1. Yeni kanal, sonlandırmayı isteyerek önceki kanalı durdurmaya çalışır.
2. Önceki kanal sunucusu **AdoptNewMCATimeout** bekleme aralığı sona erdiğinde bu isteğe yanıt vermezse, önceki kanal sunucusuna ilişkin iş parçacığı ya da işlem sona erdirilir.
3. Önceki kanal sunucusu 2. adımdan sonra sona ermediyse ve **AdoptNewMCATimeout** bekleme aralığı ikinci kez sona erdikten sonra IBM MQ , kanalı CHANNEL IN USE hatasıyla sona erdirir.

**AdoptNewMCA** işlevselliği sunucu, gönderen, alıcı ve küme alıcı kanalları için geçerlidir. Gönderen ya da sunucu kanalı söz konusu olduğunda, alan kuyruk yöneticisinde belirli bir ada sahip bir kanalın yalnızca bir eşgörünümü çalıştırılabilir. Bir alıcı ya da küme alıcı kanalı söz konusu olduğunda, alan kuyruk yöneticisinde belirli bir ada sahip bir kanalın birden çok eşgörünümü çalışıyor olabilir, ancak belirli bir uzak kuyruk yöneticisinden aynı anda yalnızca bir eşgörünüm çalıştırılabilir.

**Not: AdoptNewMCA** , istekte bulunan ya da sunucu bağlantısı kanallarında desteklenmez.

Aşağıdaki listeden virgüllerle ya da boşluklarla ayrılmış olarak bir ya da daha çok değer belirleyin:

### **HAYIR**

AdoptNewMCA özelliği gerekli değildir. Bu varsayılandır.

### **SVR**

Sunucu kanallarını benimseyin.

### **SDR**

Gönderen kanallarını benimseyin.

### **RCVR**

Alıcı kanallarını benimseyin.

### **CLUSRCVR**

Küme alıcı kanallarını benimseyin.

### **TÜMÜ**

FASTPATH kanalları dışındaki tüm kanal tiplerini benimseyin.

### **FastPath**

FASTPATH kanalından biriye kanalı benimseyin. Bu yalnızca uygun kanal tipi de belirtildiyse gerçekleşir, örneğin: AdoptNewMCA=RCVR , SVR , FASTPATH.

**Dikkat!:** AdoptNewMCA özneliği, FASTPATH kanallarıyla öngörülemez bir şekilde davranabilir. FASTPATH kanalları için AdoptNewMCA özneliğini etkinleştirirken çok dikkatli olun.



### **AdoptNewMCATimeout= 60 (varsayılan) | 1-3600**

Yeni kanal örneğinin eski kanal örneğinin sona ermesini bekleyeceği süre (saniye olarak). 1-3600 aralığında bir değer belirtin. Varsayılan değer 60'tır.

### **AdoptNewMCACheck = QM | ADDRESS | NAME | ALL**

AdoptNewMCA özneliği etkinleştirilirken gereken denetim tipi. Mümkünse, kanallarınızın yanlışlıkla ya da kötü niyetle kapatılmasını önlemek için tam denetim gerçekleştirin. En azından kanal adlarının eşleştiğini kontrol edin.

QM, NAME ya da ALL durumunda virgüllerle ya da boşluklarla ayrılmış olarak aşağıdaki değerlerden birini ya da daha fazlasını belirtin:

#### **QM**

Kuyruk yöneticisi adlarının eşleştiğini denetleyin.

Kuyruk yöneticisi adının QMID ile değil, kendisinin eşleştiğini unutmayın.

#### **ADRES**

İletişim kaynağı IP adresini denetleyin. Örneğin, TCP/IP adresi.

**Not:** Virgülle ayrılmış CONNAME değerleri hedef adresler için geçerlidir ve bu nedenle bu seçenekle ilgili değildir.

Çok eşgörünümlü bir kuyruk yöneticisinin hosta 'den hostb' e geçmemesi durumunda, o kuyruk yöneticisindeki giden kanallar hostbkaynak IP adresini kullanır. Bu, hosta' den farklıysa, AdoptNewMCACheck=ADDRESS eşleşmez.

Bir saldırganın var olan bir çalışan kanalı bozmasını önlemek için karşılıklı kimlik doğrulamasıyla SSL ya da TLS kullanabilirsiniz. Alternatif olarak, çok eşgörünümlü kuyruk yöneticileri yerine IP devralma özelliği olan bir HACMP tipi çözüm kullanın ya da kaynak IP adresini maskeleyen için bir ağ yük dengeleyici kullanın.

#### **AD**

Kanal adlarının eşleştiğini kontrol edin.

#### **TÜMÜ**

Eşleşen kuyruk yöneticisi adlarını, iletişim adresini ve eşleşen kanal adlarını denetleyin.

Varsayılan değer AdoptNewMCACheck=NAME , ADDRESS , QM' dir.

### **ChlauthEarlyEvlut Edinme = Y (varsayılan) | N**

Bağlantı kimlik doğrulaması ve kanal kimlik doğrulama kurallarının işlendiği sıra, IBM MQ istemci uygulaması bağlantıları için güvenlik bağlamının belirlenmesinde önemli bir etmendir.



**Uyarı:** qm.ini dosyasında **ChlauthEarlyAdopt** yoksa varsayılan değer N'dir, ancak IBM MQ 9.0.4 ' den tüm kuyruk yöneticileri **ChlauthEarlyAdopt**=Y ile oluşturulur ve qm.ini dosyasına otomatik olarak eklenir.

**ChlauthEarlyAdopt** yalnızca, kuyruk yöneticisindeki bağlantı kimlik doğrulaması AUTHINFO nesnesinde ADOPTCTX (YES) ayarlandıysa, bağlantı doğrulaması için bir kuyruk yöneticisine sağlanan kullanıcı kimliklerini benimser.

**ChlauthEarlyAdopt** için geçerli değerler şunlardır:

#### **Y**

Kanal, kanal kimlik doğrulama kurallarını uygulamadan önce kuyruk yöneticisi bağlantısı kimlik doğrulamasını kullanan bir uygulama tarafından sağlanan kullanıcı kimliği ve parola kimlik bilgilerini doğrular ve kullanır. Bu işlem kipinde, kanal kimlik doğrulama kuralları, bağlantı kimlik doğrulama denetimlerinden kaynaklanan kullanıcı kimliğiyle eşleşir.

#### **N**

Kanal, kanal kimlik doğrulama kuralları uygulanıncaya kadar bir uygulama tarafından sağlanan kullanıcı kimliği ve parola kimlik bilgilerinin bağlantı kimlik doğrulamasını geciktirdi. Bu işlem kipinde, kanal kimlik doğrulamasını engelleme ve eşleme kurallarının kullanıcı kimliği ve parola geçerlilik denetimi sonuçlarını dikkate almayacağını unutmayın.

Örneğin, varsayılan kimlik doğrulama bilgileri nesnesi **ADOPTCTX(YES)** olarak ayarlanır ve kullanıcı fred oturum açar. Aşağıdaki iki CHLAUTH kuralı yapılandırıldı:

```
SET CHLAUTH('MY.CHLAUTH') TYPE(ADDRESSMAP) DESCR('Block all access by
default') ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
SET CHLAUTH('MY.CHLAUTH') TYPE(USERMAP) DESCR('Allow user bob and force
CONNAUTH') CLNTUSER('bob') CHCKCLNT(REQUIRED) USERSRC(CHANNEL)
```

bobkullanıcısının benimsenen güvenlik bağlamı olarak komutu doğrulamak amacıyla aşağıdaki komut verilir:

```
runmqsc -c -u bob QMGR
```

Gerçekte, kuyruk yöneticisi bobdeğil, fredgüvenlik bağlamını kullanır ve bağlantı başarısız olur. bobgüvenlik bağlamını kullanmak için **ChlauthEarlyAdopt** , Yolarak ayarlanmalıdır.

### **PasswordProtection = Uyumlu (varsayılan) |her zaman|isteğe bağlı|uyarı**

Bağlantı TLS şifrelemesini kullanmıyorsa, IBM MQ 8.0' den IBM MQ client uygulamalarının bir kuyruk yöneticisine bağlandığında belirttiği kimlik doğrulama kimlik bilgileri IBM MQ MQCSP parola koruma özelliği kullanılarak korunabilir.

MQCSP parola koruması, test ve geliştirme amaçlarıyla yararlıdır; MQCSP parola korumasının kullanılması TLS şifrelemesi ayarlanmasından daha basittir, ancak güvenli değildir.

MQCSP yapısındaki kimlik bilgilerinin ve bu öznitelik için ayarlanabilecek değerlerin korunması hakkında daha fazla bilgi için bkz. [MQCSP parola koruması](#).

### **IgnoreSeqNumberMismatch = NO (default) | YES**

Bir kanalın iki ucundaki İleti Kanalı Aracıları (MCA), eşitlemeyi sürdürmek için kanal üzerinden gönderilen iletilerin sayısını korur. Eşitleme kaybedilebilir; örneğin, bir uçta kanal tanımlaması silinip yeniden oluşturulursa. Bu koşullarda, senkronizasyon verilerinin kaybolduğunu onaylamak ve kanalın başlatmaya devam etmesine izin vermek için bir RESET CHANNEL gerekebilir.

Alıcı kuyruk yöneticisinde **IgnoreSeqNumberMismatch** öznitelik ayarlanmalıdır.

Bu öznitelik, alıcı kanalda kanal sıfırlama komutunu etkili bir şekilde gerçekleştirir.

Bu öznitelik, aşağıdaki değerleri kullanarak, kuyruk yöneticisinin kanal başlatılırken bir sıra numarası uyuşmazlığı nasıl işleyeceğini denetler:

#### **HAYIR**

Kanal yeniden eşzamanlama sırasında kanal sıra numaraları denetlenir; iki MCA aynı sıra numarası üzerinde anlaşmazsa, AMQ9526 hata iletisi bildirilir ve kanal başlatılamaz.

#### **EVET**

Kanal yeniden eşzamanlama sırasında kanal sıra numaraları denetlenir, ancak iki MCA aynı sıra numarası üzerinde anlaşmazsa, AMQ9703 uyarı iletisi bildirilir ve kanal başlatma işlemi devam eder. Bu öznitelik değeri normal koşullar altında gerekli olmamalıdır. Örneğin, olağanüstü durumdan kurtarma sırasında eşzamanlama verilerinin kaybolduğu bilindiğinde, bu seçenek her sıra numarası uyuşmazlığı için el ile onay verilmesine gerek kalmaz. Bu değer belirtilmesi, her sıra numarası uyuşmazlığı için otomatik olarak bir **RESET CHANNEL** yayınlanması ile benzer bir etkiye sahiptir.

### **ChlauthIgnoreUserCase = N (varsayılan) | Y**

Bir kuyruk yöneticisinin CHLAUTH kuralları içinde kullanıcı adı eşleştirmesini büyük ve küçük harfe duyarsız hale getirmesini sağlar. Bu seçenek aşağıdakileri sağlar:

- CHLAUTH TYPE (USERMAP) kurallarındaki CLNTUSER büyük ve küçük harfe duyarsız olarak eşleştirilecek
- CHLAUTH TYPE (BLOCKUSER) kurallarındaki USERLIST büyük ve küçük harfe duyarsız olarak eşleştirilecek

**ChlauthIgnoreUserCase** için geçerli değerler şunlardır:

## N

Kanal kimlik doğrulama kuralları, istemci kullanıcı kimliğini büyük ve küçük harfe duyarlılıkla eşleştirmeyi dener; örneğin, CLNTUSER 'i (' Fred ') belirten bir kural, ' fred 'ya da 'XX\_ENCODE\_CASE\_ONE fred' ile eşleşmez; yalnızca 'Fred' kullanıcı kimliğiyle eşleşir. Bu varsayılan değerdir.

## Y

Kanal kimlik doğrulama kuralları, istemci kullanıcı kimliğini büyük ve küçük harfe duyarsız olarak eşleştirmeyi dener; örneğin, TYPE (USERMAP) ya da TYPE (USERBLOCK) ile CLNTUSER ('Fred') belirtilirse, 'Fred', 'FRED' ve 'fred' gibi kullanıcı tanıtıcılarının tümü eşleşir.

Kanal kimlik doğrulama kurallarını eşleştirirken kullanıcı tanıtıcılarının büyük/küçük harf kullanımı yoksayılırken, birden çok kuralın eşleşmesinin mümkün olduğunu unutmayın. Bu durumda, eşleştirilen kural tanımsız olur. Örneğin, 'Fred' adlı kullanıcı CLIENT kanalı aracılığıyla bir kuyruk yöneticisine bağlanırsa, 'mquser1' ya da 'mquser2' ile eşlenebilir:

```
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('fred') USERSRC(MAP) MCAUSER('mquser1')
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('FRED') USERSRC(MAP) MCAUSER('mquser2')
```

ChlauthIgnoreUserCase=Y kullanırken belirsizliği önlemek için, büyük ve küçük harfe duyarsız bir eşleşme kullanırken çıkışacak ve farklı davranışlarla sonuçlanacak CHLAUTH kurallarını tanımlamaktan kaçınınız.

### ChlauthIssueWarn = y

**SET CHLAUTH** komutunda **WARN = YES** özneliğini ayarladığınızda AMQ9787 iletisinin oluşturulmasını istiyorsanız bu özneliği ayarlayınız.

## Örnek kısım

```
Channels:
  MaxChannels=200
  MaxActiveChannels=100
  MQIBindType=STANDARD
  PipelineLength=2
```

## İlgili kavramlar

[“Kanal durumları” sayfa 219](#)

Bir kanal herhangi bir zamanda birçok eyaletten birinde olabilir. Bazı eyaletlerin de alt devletleri vardır. Belirli bir durumdan bir kanal diğer durumlara geçebilir.

### Multi

### qm.ini dosyasının bağlantı kısmı

Bağlantı kısmı varsayılan bağ tanımlama tipini tanımlar.

Varsayılan bağ tanımlama tipini belirtmek için qm.ini kütüğündeki Bağlantı kısmı kullanılır.

### Linux

### Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer Extended kuyruk yöneticisi özellikleri sayfasını kullanınız.

**Not:** Gerekiyorsa bir Bağlantı kısmı oluşturmalısınız.

### DefaultBindType = SHARED (varsayılan) | YALITILMIŞ

**DefaultBindType** YALITILMIŞ olarak ayarlanırsa, uygulamalar ve kuyruk yöneticisi ayrı işlemlerde çalışır ve bunlar arasında kaynak paylaşılmaz.

**DefaultBindType** SHARED olarak ayarlanırsa, uygulamalar ve kuyruk yöneticisi ayrı işlemlerde çalışır, ancak bazı kaynaklar bunlar arasında paylaşılır.

Varsayılan değer SHARED değeridir.



**Uyarı:** **DefaultBindType** , tüm MQCONN çağrıları ve MQCNO\_STANDARD\_BINDING ile MQCONNX kullanan tüm çağrılar için geçerlidir.

**DefaultBindType** ' in deđiřtirilmesi, bazı uygulamaların bařarımında dūřmesine neden olabilir.

## Örnek kısmı

```
Connection:  
DefaultBindType=SHARED
```

### Multi Tanılama iletilerini günlüđe kaydetme

IBM MQ tanılama ileti günlükleri, IBM MQ sisteminin çeřitli bileřenlerinin IBM MQ yapılandırması ve alıřtırma zamanı durumu deđiřiklikleri ve sorunlarıyla ilgili tanılama iletilerini raporlamasını sađlayan bir mekanizmadır.

Bu günlüklere bazen IBM MQ *hata günlüklerinden*, ancak her zaman IBM MQ bilgi ve uyarı iletilerinin yanı sıra hata iletilerini de ierir. IBM MQ ' in bu günlüklere rapor veren üç birincil bileřeni řunlardır:

- Kuyruk Yöneticileri
- IBM MQ müřterileri
- IBM MQ sisteminin geri kalanı

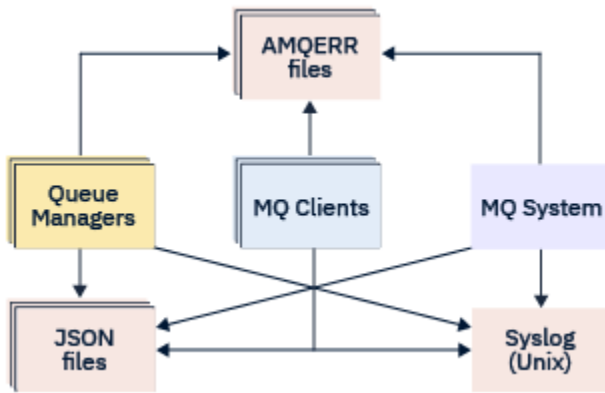
IBM MQ , *tanılama ileti hizmetleri* olarak bilinen bir dizi farklı yöntem aracılıđıyla tanılama iletilerini raporlamayı destekler ve bu bilgileri kaydetmeye ve tüketmeye yönelik uyarlanmış bir yaklařım sađlar:

- AMQERRnn günlük dosyaları
- JSON biçimli günlük dosyaları
- **AIX** JSON biçiminde Syslog

IBM MQ tarafından JSON ıkıřı, JSON günlüđünün ya da Syslog kaydının her bir satırının geerli bir JSON nesnesini göstermesi gibi tek satırlı JSON nesnelere olarak biçimlendirilir. Bir bütün olarak günlük, tek bir JSON nesnesi olarak kapsüllenmez.

Ařađıdaki řekilde, kuyruk yöneticilerinin, IBM MQ istemcilerinin ve IBM MQ sisteminin, aıklanan yöntemleri kullanarak tanılama iletilerini *raporlayabileceđi* gösterilmektedir.

řekil 5. IBM MQ ' un farklı kısımları tanılama iletilerini nasıl bildirebilir?



### IBM MQ tanılama günlükleri nasıl yapılandırılır:

Tanılama günlükleri, `qm.ini` dosyasındaki stanzas kullanılarak, bunları gerektiren IBM MQ bileřenine özel olarak tanımlanır ve özelleřtirilebilir. Her benzersiz günlük kaydı uç noktası, `ini` dosyası iindeki kendi kısmı bařlıđı altında ve iinde tanımlanan özelleřtirmelerle birlikte tanımlanır. Özelleřtirmeler řunları ierebilir:

- Baştan başlama gerçekleşmeden önce kaydırılacak günlük dosyalarının boyutu; Syslog için geçerli değildir
- Günlük iletilerinin önem derecesine göre süzme ve
- Gizlenecek herhangi bir özel ileti kodu.

IBM MQ , üç tip günlük kaydı uç noktasının herhangi birine ya da tümüne yazacak şekilde yapılandırılabilir; böylece belirli günlük kısmı belirli rolleri yerine getirebilir. Benzer şekilde, birden çok dosya hizmeti tanımlanabilir. Örneğin:

- JSON biçimi, yerel ve Bulut ortamlarında otomatikleştirilmiş araçlar aracılığıyla ayrıştırmayı kolaylaştırır.
- Syslog çıkışı, IBM MQ bileşenlerinin sistemdeki diğer ürünlerle uyumlu olarak tanılama bilgilerini ortak bir işletim sistemi günlük kaydı konumuna bütünleştirmesini sağlar.
- Belirli günlük dosyalarının kaydedilmesine izin veren önem derecesine göre süzülen günlük uç noktaları; örneğin, yalnızca sistemdeki önemli hatalar.

Yapılandırılan tanılama günlüğü stiline bakılmaksızın, IBM MQ sistem günlüğü dizini (/var/mqm/errors/AMQERRnn.log) ve belirli kuyruk yöneticisi günlük dizini (/var/mqm/qmgrs/<qmgr\_name>/errors/AMQERRnn.log) altında tutulan geleneksel tanılama dosyaları, kullanılan diğer günlük kaydı yapılandırmasına ek olarak her zaman yazılır.

Yalnızca kuyruk yöneticileri için, bu zorunlu günlüklerin isteğe bağlı yapılandırması [“Tanılama ileti hizmeti stanzas” sayfa 123](#) öznitelikleri belirtilerek gerçekleştirilebilir.

## Farklı kıta alanları

Ek kısaltmalar, IBM MQ' in farklı alanlarına uygulanabilir.

### Qmgr (qm.ini)

Kuyruk yöneticisi tarafından oluşturulan günlük iletileri için geçerlidir

### Sistem (mqc.ini)

Sistem tarafından oluşturulan günlük iletileri için geçerlidir. Bir kuyruk yöneticisinin kendi günlüklerine erişememesi ya da kendi günlüklerine yazamaması dışında, bu seçenek bir kuyruk yöneticisine özgü değildir.

### Şablonlar (mqc.ini)

Bir kuyruk yöneticisi yaratıldığında qm . ini ' e kopyalanan bir ya da daha çok bölüm şablon olarak.

### İstemci (mqclient.ini)

İstemci işlemi için geçerlidir; örneğin, istemci kipinde uzak bir kuyruk yöneticisine **runmqsc** .

## JSON biçimli ve geleneksel olarak biçimlendirilmiş günlükler arasında dönüştürme

mqrc komutu, JSON ile geleneksel olarak biçimlendirilen günlükler arasında ve farklı diller arasında bir dizi dönüştürmeye izin verecek şekilde geliştirilmiştir.

### İlgili başvurular

[“Tanılama ileti hizmeti stanzas” sayfa 123](#)

Kullanılabilir tanılama iletileri hizmeti seçenekleri, günlük çıkışının farklı IBM MQ bileşenlerinden farklı günlük uç noktalarına yönlendirilebilmesi için IBM MQ tanılama günlüklerinizin özelleştirilmesini sağlar.

[“QMErrorLog kısmı” sayfa 122](#)

IBM MQ hata günlüklerinin işlem ve içeriğini uyarlamak için qm . ini dosyasındaki QMErrorLog kuyruk yöneticisi hata günlüğü kısmı kullanılır.

[“Tanılama iletileri hizmetleri” sayfa 125](#)

Yapılandırma dosyalarınızın DiagnosticSystemMessages, DiagnosticMessages ve DiagnosticMessagesTemplate (Tanılama İletileri) altında belirtilen aşağıdaki tanılama iletileri hizmetleri ve bunların hizmete özgü öznitelikleri tanımlanabilir:

IBM MQ hata günlüklerinin işlem ve içeriğini uyarlamak için `qm.ini` dosyasındaki QMErrorLog kuyruk yöneticisi hata günlüğü kısmı kullanılır.

QMErrorLog hizmeti, kuyruk yöneticisiyle ilgili tanılama iletilerini çıkışta kullanılan geleneksel IBM MQ tanılama günlüğü hizmetidir. QMErrorLog hizmeti sürekli olarak çalışır ve kapatılamaz, ancak bir dereceye kadar uyarlanabilir.

Belirli iletilerin kuyruk yöneticisi hata günlüğüne yazılmasını dışlamak için `qm.ini` dosyasındaki QMErrorLog kısmı kullanılabilir. Belirli bir zaman aralığı için hata günlüğüne ileti yazılmasını da engelleyebilirsiniz.

Diğer bir seçenek olarak, `qm.ini` dosyasını doğrudan düzenlemek yerine,

**Dışlanan iletiler, Engellenen iletiler ve Engellenen iletiler aralığı** öznitelikleriyle iletileri dışlamak ve gizlemek için IBM MQ Explorer içindeki [Genişletilmiş Kuyruk Yöneticisi özellikleri sayfasını](#) kullanabilirsiniz.



#### Uyarı:

- **Windows** IBM MQ Explorer yalnızca Windows altyapısında yerel bir kuyruk yöneticisi kullanıyorsanız değişiklik yapmak için kullanabilirsiniz.
- QMErrorLog kısmı, IBM MQ sistem yapılandırma dosyası `mqc.ini`ya da genellikle `mqclient.ini`olarak adlandırılan istemci yapılandırma dosyası için geçerli değildir.

QMErrorLog kısmına aşağıdaki öznitelikler eklenebilir:

#### **ErrorLogBüyükLük = büyüklük üst sınırı**

Yedeklemeye kopyalanan kuyruk yöneticisi hata günlüğünün boyutunu belirtir. *büyükLük üst sınırı* , 32768-2147483648 byte aralığında olmalıdır. **ErrorLogSize** belirtilmezse, 33554432 bayt (32 MB) varsayılan değeri kullanılır.

Gerekirse, boyut üst sınırını önceki üst sınır olan 2 MB ' ye düşürmek için bu özneteliği kullanabilirsiniz.

**MQMAXERRORLOGSIZE** ortam değişkenini kullanarak günlüğün boyutunu ayarlayabilirsiniz.

#### **ExcludeMessage= msgIds**

Kuyruk yöneticisi hata günlüğüne yazılmayacak iletileri belirtir.

Daha fazla bilgi için "Tanılama ileti hizmeti stanzas" sayfa 123 içindeki [ExcludeMessage](#) bölümüne bakın.

#### **SuppressMessage= msgIds**

Kuyruk yöneticisi hata günlüğüne yalnızca belirli bir zaman aralığında bir kez yazılan iletileri belirtir. Aynı ileti tanıtıcısı hem [SuppressMessage](#) hem de [ExcludeMessage](#) içinde belirtilirse, ileti dışlanır.

Bu seçenek, `mqclient.ini` içinde tanımlanan tanılama iletileri hizmetleri için geçerli değildir. Daha fazla bilgi için "Tanılama ileti hizmeti stanzas" sayfa 123 içindeki [SuppressMessage](#) başlıklı konuya bakın.

#### **SuppressInterval= uzunluk**

[SuppressMessage](#) içinde belirtilen iletilerin kuyruk yöneticisi hata günlüğüne yalnızca bir kez yazıldığı saniye cinsinden zaman aralığını belirtir. *uzunluk* , 1-86400 saniye aralığında olmalıdır. [SuppressInterval](#) (Engelleme Aralığı) belirtilmezse, 30 saniyelik varsayılan değer kullanılır.

## Örnek kısmı

```
QMErrorLog:
  ErrorLogSize=262144
  ExcludeMessage=7234
  SuppressMessage=9001,9002,9202
  SuppressInterval=30
```

## İlgili kavramlar

“Kuyruk yöneticisi yapılanış dosyaları, qm.ini” sayfa 97

Kuyruk yöneticisi yapılanış dosyası ( qm . ini), belirli bir kuyruk yöneticisiyle ilgili bilgileri içerir. Tek bir kuyruk yöneticisinin yapılandırmasını değiştirmek için kullanabileceğiniz öznitelikler, IBM MQ ile ilgili ayarları geçersiz kılar.

## İlgili başvurular

“Tanılama iletisi hizmeti stanzas” sayfa 123

Kullanılabilir tanılama iletisi hizmeti seçenekleri, günlük çıkışının farklı IBM MQ bileşenlerinden farklı günlük uç noktalarına yönlendirilebilmesi için IBM MQ tanılama günlüklerinizin özelleştirilmesini sağlar.

### **Tanılama iletisi hizmeti stanzas**

Kullanılabilir tanılama iletisi hizmeti seçenekleri, günlük çıkışının farklı IBM MQ bileşenlerinden farklı günlük uç noktalarına yönlendirilebilmesi için IBM MQ tanılama günlüklerinizin özelleştirilmesini sağlar.

Aşağıdaki adlardan birini içeren bir bölümü kullanarak ek tanılama iletisi hizmetlerini etkinleştirebilirsiniz:

#### • **DiagnosticSystemMessages**

Sistem hata günlüğüne giden bir tanılama iletisi oluşturulduğunda kullanılan hizmetleri tanımlar. mqs . ini ya da mqclient . ini dosyalarında geçerlidir.

İstemci uygulamaları, mqclient.ini dosyasında ve mqs.ini' de bir **DiagnosticSystemMessages** kısmı kullanır; **DiagnosticSystemMessages** kısmı, kuyruk yöneticisi bağlamı olmayan bir sunucu uygulamasına ilişkin iletileri denetler.

Syslog hizmetine ek olarak tüm iletileri yazan bir kuyruk yöneticisi ve uygulamaları da yapılandırabilirsiniz.

#### • **DiagnosticMessages**

Kuyruk yöneticisi hata günlüğüne giden bir tanılama iletisi oluşturulduğunda kullanılan hizmetleri tanımlar. Yalnızca qm . ini dosyasında geçerlidir.

#### • **DiagnosticMessagesTemplate**

Bir kuyruk yöneticisi yaratıldığında, qm . ini dosyasında mqs . ini dosyasından **DiagnosticMessages** dosyasına kopyalanan bir bölüm.

Tanılama iletilerini görüntülemek için [mqrc](#) komutunu kullanın.

## Stanzaların öznitelikleri



**Uyarı:** Hizmetve bir kıta adı zorunludur.

**ad= < stanzaname >**

Bir kıta adı. Değer bir ini dosyasında benzersiz olmalıdır.

**Hizmet = hizmet tipi**

Bu öznitelik, hizmetin adının büyük ve küçük harfe duyarlı olmadığı, bu kıta tarafından etkinleştirilmekte olan bir hizmeti tanımlar.

Örneğin, syslog ' i ek hizmet olarak etkinleştirmek için aşağıdakileri girin:

```
Service=syslog
```

Bkz. “Tanılama iletisi hizmetleri” sayfa 125 ve tanılama iletisi hizmet kısmı ile kullanılmak üzere sağlanan özel öznitelikleri.

Aşağıdaki isteğe bağlı öznitelikleri stanzlara ekleyebilirsiniz:


- [ExcludeMessage](#)
- [SuppressMessage](#)
- [SuppressInterval](#)



- “Önem düzeyleri” sayfa 125


#### **ExcludeMessage= msgIds**

Kuyruk yöneticisi hata günlüğüne yazılmayacak iletileri belirtir. IBM MQ sisteminiz yoğun bir şekilde kullanılıyorsa, birçok kanal durduruluyor ve başlatılıyorsa, z/OS konsoluna ve basılı günlüğe çok sayıda bilgi iletileri gönderilir. IBM MQ - IMS köprüsü ve arabellek yöneticisi de çok sayıda bilgi iletileri üretebilir; bu nedenle, iletileri dışlamanız gerekirse çok sayıda ileti almanızı önler. *msgIds* , aşağıdaki ileti tanıtıcılarının virgülle ayrılmış bir listesini içerir:

- 5211-Özellik adı uzunluğu üst sınırı aşıldı.
- 5973-Dağıtılmış yayınlama/abone olma aboneliği engellendi
- 5974-Dağıtılmış yayınlama/abone olma yayını engellendi
- 6254-Sistem paylaşılan kitaplığı dinamik olarak yükleyemedi
-  7163-İş başlatıldı iletileri (yalnızcaIBM i )
- 7234-Yüklenmiş ileti sayısı
- 8245-Varlık, nesneyi görüntülemek için yeterli yetkiye sahip değil
- 9001-Kanal programı olağan şekilde sona erdi
- 9002-Kanal programı başlatıldı
- 9202-Uzak anasistem kullanılamıyor
- 9208-Anasistemden alma sırasında hata oluştu
- 9209-Bağlantı kapatıldı
- 9228-Kanal yanıtlayıcı başlatılamıyor
- 9489-SVRCONN eşgörünüm sayısı üst sınırı aşıldı
- 9490-İstemci başına SVRCONN eşgörünüm sayısı üst sınırı aşıldı
- 9508-Kuyruk yöneticisiyle bağlantı kurulamıyor
- 9524-Uzak kuyruk yöneticisi kullanılamıyor
- 9528-Kullanıcı, kanalın kapatılmasını istedi
- 9545-Bağlantı kesme aralığının süresi doldu
- 9558-Uzak Kanal kullanılamıyor
- 9637-Kanalda sertifika yok
- 9776-Kanal, kullanıcı kimliği tarafından engellendi
- 9777-Kanal, NOACCESS işlemi tarafından engellendi
- 9782-Bağlantı adres tarafından engellendi
- 9999-Kanal programı olağandışı sona erdi

#### **SuppressMessage= msgIds**

Kuyruk yöneticisi hata günlüğüne yalnızca belirli bir zaman aralığında bir kez yazılan iletileri belirtir. IBM MQ sisteminiz yoğun bir şekilde kullanılıyorsa, birçok kanal durduruluyor ve başlatılıyorsa, z/OS konsoluna ve basılı günlüğe çok sayıda bilgi iletileri gönderilir. IBM MQ - IMS köprüsü ve arabellek yöneticisi de çok sayıda bilgi iletileri üretebilir; bu nedenle, iletilerin engellenmesi, gerekirse yinelenen ileti almanızı önler. Zaman aralığı SuppressInterval ile belirtilir. *msgIds* , aşağıdaki ileti tanıtıcılarının virgülle ayrılmış bir listesini içerir:

- 5211-Özellik adı uzunluğu üst sınırı aşıldı.
- 5973-Dağıtılmış yayınlama/abone olma aboneliği engellendi
- 5974-Dağıtılmış yayınlama/abone olma yayını engellendi
- 6254-Sistem paylaşılan kitaplığı dinamik olarak yükleyemedi
-  7163-İş başlatıldı iletileri (yalnızcaIBM i )
- 7234-Yüklenmiş ileti sayısı
- 8245-Varlık, nesneyi görüntülemek için yeterli yetkiye sahip değil
- 9001-Kanal programı olağan şekilde sona erdi
- 9002-Kanal programı başlatıldı
- 9202-Uzak anasistem kullanılamıyor
- 9208-Anasistemden alma sırasında hata oluştu



9209-Bağlantı kapatıldı  
9228-Kanal yanıtlayıcı başlatılmıyor  
9489-SVRCONN eşgörünüm sayısı üst sınırı aşıldı  
9490-İstemci başına SVRCONN eşgörünüm sayısı üst sınırı aşıldı  
9508-Kuyruk yöneticisiyle bağlantı kurulamıyor  
9524-Uzak kuyruk yöneticisi kullanılmıyor  
9528-Kullanıcı, kanalın kapatılmasını istedi  
9545-Bağlantı kesme aralığının süresi doldu  
9558-Uzak Kanal kullanılmıyor  
9637-Kanalda sertifika yok  
9776-Kanal, kullanıcı kimliği tarafından engellendi  
9777-Kanal, NOACCESS eylemi tarafından engellendi  
9782-Bağlantı adres tarafından engellendi  
9999-Kanal programı olağandışı sona erdi

Aynı ileti tanıtıcısı hem SuppressMessage hem de ExcludeMessage içinde belirtilirse, ileti dışlanır.  
Bu seçenek, MQ client.ini içinde tanımlanan tanılama iletisi hizmetleri için geçerli değildir.

### **SuppressInterval= uzunluk**

**SuppressMessage** içinde belirtilen iletilerin yalnızca bir kez kuyruk yöneticisi hata günlüğüne yazılacağı zaman aralığını saniye cinsinden belirtir. *uzunluk* , 1-86400 saniye aralığında olmalıdır. **SuppressInterval** belirtilmezse, 30 saniyelik varsayılan değer kullanılır.

### **Önem düzeyleri**

Önem düzeyi adının büyük ve küçük harfe duyarlı olmadığı önem düzeylerinin virgülle ayrılmış listesi. İzin verilen değerler şunlardır:

- I (veya Bilgi veya 0)
- W (ya da Uyarı ya da 10)
- E (ya da Hata ya da 20 ve 30)
- S (ya da Durdur ya da 40)
- T (ya da Sistem ya da 50)

### **Notlar:**

1. Varsayılan değer a11 değeridir.
2. Hizmete yalnızca seçilen önem düzeyindeki iletiler sunulur.

Diğer bir seçenek olarak, belirtilen hata düzeyini ve tüm daha yüksek düzeyleri görüntüleyen artı karakterini (+) kullanabilirsiniz. Örneğin, tüm hataları görüntülemek için:

Severities=E+

### **İlgili başvurular**

[“QMErrorLog kısmı” sayfa 122](#)

IBM MQ hata günlüklerinin işlem ve içeriğini uyarlamak için qm.ini dosyasındaki QMErrorLog kuyruk yöneticisi hata günlüğü kısmı kullanılır.

[“Tanılama iletisi hizmetleri” sayfa 125](#)

Yapılandırma dosyalarınızın DiagnosticSystemMessages, DiagnosticMessages ve DiagnosticMessagesTemplate (Tanılama İletileri) altında belirtilen aşağıdaki tanılama iletisi hizmetleri ve bunların hizmete özgü öznitelikleri tanımlanabilir:

**Multi** Tanılama iletisi hizmetleri

Yapılandırma dosyalarınızın DiagnosticSystemMessages, DiagnosticMessages ve DiagnosticMessagesTemplate (Tanılama İletileri) altında belirtilen aşağıdaki tanılama iletisi hizmetleri ve bunların hizmete özgü öznitelikleri tanımlanabilir:

Aşağıdaki tanımlama iletisi hizmetleri tanımlandı:

### Dosya

Bu hizmet, süzgeç uygulanmamış iletileri QMErrorLog hizmetine benzer şekilde bir dosyaya gönderir. Belirtilen **Format** biçimine bağlı olarak, var olan metin biçimi ya da belirtilen JSON biçimi kullanılır. Varsayılan olarak, **Format** özelliğine bağlı olarak AMQERR01.LOG, AMQERR02.LOG ve AMQERR03.LOG ya da AMQERR01.json, AMQERR02.json ve AMQERR03.json adlı üç dosya ve yapılandırılan boyuta dayalı olarak bu baştan başlama dosyaları vardır.

Yalnızca bir Dosya kısmı için aşağıdaki öznitelikler desteklenir:

### FilePath

Günlük dosyalarının yazılacağı yol. Varsayılan değer, AMQERR01.log dosyalarıyla (sistem ya da kuyruk yöneticisi) aynı konumdur. Yol mutlak olmalıdır, ancak değiştirilebilir eklemeler içerebilir. Örneğin:

#### + MQ\_Q\_MGR\_DATA\_PATH +

Kuyruk yöneticisi tanımlama iletileri dizininin üst öğesinin tam yolu. Varsayılan değerler şunlardır:

- **Linux** **AIX** AIX and Linux platformlarında: /var/mqm/qmgrs/<QM\_name>
- **Windows** Windows üzerinde, C:\Program Data\IBM\MQ\qmgrs\<QM\_name>

#### + MQ\_DATA\_PATH +

Sistem tanımlama iletileri dizininin üst öğesinin tam yolu. Varsayılan değerler şunlardır:

- **Linux** **AIX** AIX and Linux platformlarında: /var/mqm
- **Windows** Windows işletim tarihinde: C:\Program Data\IBM\MQ

Var olan hatalar dizinini kullanmıyorsa, bu yolu uygun izinlerle yaratmalısınız.

### FilePrefix

Günlük dosyalarının öneki. Varsayılan değer AMQERR'dir.

### FileSize

Günlüğün yuvarlanacağı büyüklük. Varsayılan değer, anlamsal olarak aynı olan "QMErrorLog kısmı" sayfa 122 ürününün **ErrorLogSize** özelliğinde olduğu gibi 32MB'dir.

**Not: ErrorLogSize** özelliği, özel tanımlama hizmetleri için değil, yalnızca varsayılan hata günlüğü hizmeti için geçerlidir.

**MQMAXERRORLOGSIZE** ortam değişkenini kullanarak günlüğün boyutunu ayarlayabilirsiniz.

### Format

Dosyanın biçimi. Değer *text* (ek QMErrorLog biçim hizmetleri için) ya da *json* (varsayılan) olabilir.

Dosyanın soneki, bu özneliğin ayarına dayalı olarak .LOG ya da .json olabilir.

Örneğin, kuyruk yöneticisinin qm.ini dosyasını düzenleyin ve aşağıdaki kısmı ekleyin:

```
DiagnosticMessages:  
  Service = File  
  Name = JSONLogs  
  Format = json  
  FilePrefix = AMQERR
```

Yeniden başlatıldıktan sonra, kuyruk yöneticisinin ERRORS dizininde AMQERR0x.json dosyaları olur.

Birden çok Dosya hizmeti tanımlayabilirsiniz. Bu, farklı etiketlere sahip iletilerin farklı günlük kümelerine bölündüğü aşağıdaki örneklerde gösterildiği gibi yapılandırmaya izin verir:

```
DiagnosticMessages:  
  Name=ErrorsToFile  
  Service=File
```

```
Severities=E+
FilePrefix=OnlyErrors

DiagnosticMessages:
Name=NonErrorstoFile
Service=File
Severities=1 W
FilePrefix=Information
```

Linux

AIX

## Sistem günlüğü

Syslog hizmeti Windows ya da IBM i üzerinde kullanılamaz

Yalnızca bir Syslog hizmeti tanımlayabilirsiniz ve Syslog hizmeti, [JSON biçimi tanılama iletileri](#) belirtimini kullanarak syslog 'a süzgeç uygulanmamış iletiler gönderir. Bilgiler, msgID ve eklemelerle başlayarak, çizelgede gösterilen sırayla syslog 'a eklenir.

İletinin önem derecesi aşağıdaki şekilde syslog düzeyiyle eşlenir:

Önem Derecesi	Seviye
0	GüNLük BILGISI
10	GüNLük_UYARI
20	GüNLük_ERR
30	GüNLük_ERR
40	LOG_UYARISI
50	LOG_UYARISI

Aşağıdaki öznitelik yalnızca syslog kısmı için desteklenir:

### Ident

Syslog girişleriyle ilişkili **ident** değerini tanımlar. Varsayılan değer *ibm-mq'* dur.

Aşağıdaki örnek, Syslog 'a gönderilen hata iletilerini göstermektedir:

```
DiagnosticMessages:
Name=ErrorsToSyslog
Ident=mq
Service=Syslog
Severities=E+
```

Soysal dörtlük öznitelikleriyle ilgili ek bilgi için bkz. [“Tanılama ileti hizmeti stanzas” sayfa 123](#) .

### Notlar:

1. Yalnızca Dosya hizmeti için, her biri farklı bir ada sahip birden çok bölge olabilir. Yalnızca, dizideki son adı kullanan tanımlama geçerli olur.
2. Bir kıta değerinde yapılan değişiklikler yalnızca kuyruk yöneticisi yeniden başlatıldığında yürürlüğe girer.

Multi

## qm.ini dosyasının ExitPath kısmı

ExitPath kısmı, kuyruk yöneticisi sistemindeki kullanıcı çıkış programlarının yolunu belirtir.

Kuyruk yöneticisi sistemindeki kullanıcı çıkış programlarının yolunu belirtmek için qm . ini dosyasındaki ExitPath kısmı kullanın.

Linux

Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer Exits kuyruk yöneticisi özellikleri sayfasını kullanın.

### ExitsDefaultYol = *dizgi*

ExitsDefaultYol özniteliği şunların konumunu belirtir:

- İstemciler için 32 bit kanal çıkışları
- Sunucular için 32 bit kanal çıkışları ve veri dönüştürme çıkışları
- Nitelenmemiş XA anahtarı yükleme dosyaları

### ExitsDefaultPath64= *dizgi*

ExitsDefaultPath64 özniteliği aşağıdakilere ilişkin yeri belirtir:

- İstemciler için 64 bit kanal çıkışları
- Sunucular için 64 bitlik kanal çıkışları ve veri dönüştürme çıkışları
- Nitelenmemiş XA anahtarı yükleme dosyaları

## Örnek kısmı

```
ExitPath:  
ExitsDefaultPath=/var/mqm/exits  
ExitsDefaultPath64=/var/mqm/exits64
```

Multi

### ExitProperties qm.ini dosyasının yerel kısmı

ExitPropertiesYerel kısmı, bir kuyruk yöneticisindeki çıkış özellikleriyle ilgili bilgileri belirtir.

Bir kuyruk yöneticisindeki çıkış özellikleriyle ilgili bilgileri belirtmek için qm.ini dosyasındaki ExitPropertiesYerel kısmı kullanın.

Linux

Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer Cluster kuyruk yöneticisi özellikleri sayfasını kullanın.

Windows

Diğer bir seçenek olarak, Windows üzerinde **amqmdain** komutunu kullanarak bu bilgileri belirtebilirsiniz.

Varsayılan olarak bu ayar, makine genelinde yapılandırmanın ExitProperties kısmında bulunan **CLWLMode** özniteliğinden devralınır ( "mqm.ini dosyasının ExitProperties kısmı" sayfa 92 içinde açıklanmıştır). Bu ayarı yalnızca bu kuyruk yöneticisini farklı bir şekilde yapılandırmak istiyorsanız değiştirin. Bu değer, Küme kuyruğu yöneticisi özellikler sayfasındaki küme iş yükü kipi özniteliği kullanılarak tek tek kuyruk yöneticileri için geçersiz kılınabilir.

Kuyruk yöneticisi çıkış programları tarafından kullanılan yapılandırma seçeneklerini belirtmek için mqm.ini dosyasındaki ExitProperties (ExitProperties) kısmına bakın.

Linux

Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer Extended IBM MQ özellikler sayfasını kullanın.

### CLWLMode = SAFE (varsayılan) | Hızlı

Küme iş yükü (CLWL) çıkışı, bir MQI çağrısına yanıt olarak hangi küme kuyruğunun açılacağını belirtmenizi sağlar (örneğin, MQOPEN, MQPUT). CLWL çıkışı, **CLWLMode** özniteliğinde belirttiğiniz değere bağlı olarak FAST kipinde ya da SAFE kipinde çalışır. **CLWLMode** özniteliğini atlarsanız, küme iş yükü çıkışı SAFE kipinde çalışır.

### GÜVENLİ

CLWL çıkışını kuyruk yöneticisinden ayrı bir işlemde çalıştırın. Bu varsayılandır.

SAFE kipinde çalışırken kullanıcı tarafından yazılan CLWL çıkışıyla ilgili bir sorun ortaya çıkarsa, aşağıdaki durum oluşur:

- CLWL sunucusu işlemi (amqzlw0) başarısız oldu.
- Kuyruk yöneticisi CLWL sunucusu işlemini yeniden başlatır.

- Hata, hata günlüğünde size bildirilir. Bir MQI çağrısı devam ediyorsa, dönüş kodu biçiminde bildirim alırsınız.

Kuyruk yöneticisinin bütünlüğü korunur.

**Not:** CLWL çıkışının ayrı bir işlemde çalıştırılması performansı etkileyebilir.

### Hızlı

Kuyruk yöneticisi işleminde küme çıkışını satır içinde çalıştırın.

Bu seçeneğin belirtilmesi, SAFE kipinde çalıştırmayla ilişkili süreç geçiş maliyetlerinden kaçınarak performansı artırır, ancak kuyruk yöneticisi bütünlüğü pahasına bunu yapar. CLWL çıkışını yalnızca CLWL çıkışınızla ilgili bir sorun olmadığından eminseniz ve özellikle performans konusunda endişeliyseniz, CLWL çıkışını FAST kipinde çalıştırmanız gerekir.

CLWL çıkışı FAST kipinde çalışırken bir sorun ortaya çıkarsa, kuyruk yöneticisi başarısız olur ve kuyruk yöneticisinin bütünlüğü tehlikeye atılır.

## Örnek kısım

```
ExitPropertiesLocal:  
  CLWLMode=SAFE
```

Linux

IBM i

AIX

### qm.ini dosyasının dosya sistemi kısmı

Dosya sistemi kısmı, kuyruk yöneticisi hata günlüklerinde ayarlanan izinlerin değişmeden mi kalacağını yoksa varsayılan değerlerine geri mi çevrileceğini belirtir.

Hata günlüğü dosyalarında ayarlanan varsayılan izinlerin çoğu durumda yararlı olması beklenir ve bu nedenle çoğu IBM MQ denetimcilerinin bunları değiştirmesine gerek yoktur.

Ancak, IBM MQ yöneticiniz hata günlüğü dosyalarındaki izinleri değiştirmek isteyebilir; bu durumda, Filesystem stanza option **ValidateAuth=No** değerini ayarlamalıdır; bu, kuyruk yöneticisinin daha sonra izinleri değiştirmeden bırakmasına neden olur.

Varsayılan davranış (**ValidateAuth=Hayırolmadan**), kuyruk yöneticisinin kuyruk yöneticisi hata günlüklerinin dosya izinlerini denetleyip bunları varsayılan değerlerine geri çevirmesidir. Bu denetim, kuyruk yöneticisi sona erdirmeye ya da başlatma işlemi de içinde olmak üzere, istediğiniz zaman gerçekleştirilebilir.

## Örnek kısım

```
Filesystem:  
  ValidateAuth=No
```

Multi

### qm.ini dosyasının günlük kısmı

Günlük kısmı, bir kuyruk yöneticisinde oturum açmaya ilişkin bilgileri belirtir.

Bir kuyruk yöneticisinde oturum açmaya ilişkin bilgileri belirtmek için qm.ini dosyasındaki Log (Günlük) kısmına bakın.

Linux

Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer **Günlük kuyruğu yöneticisi** özellikler sayfasını kullanın.

Varsayılan olarak, bu ayarlar kuyruk yöneticisine ilişkin varsayılan günlük ayarları için belirtilen ayarlardan edinilir ( "[mq5.ini dosyasının LogDefaults kısmı](#)" sayfa 92 içinde açıklanmıştır). Bu ayarları yalnızca bu kuyruk yöneticisini farklı bir şekilde yapılandırmak istiyorsanız değiştirin.

Günlük boyutlarının hesaplanmasıyla ilgili daha fazla bilgi için bkz. [“Günlüğün boyutunun hesaplanması” sayfa 633.](#)

**Not:** Aşağıdaki değiştirge listesinde belirtilen sınırlar IBM MQ tarafından belirlenir. İşletim sistemi sınırları, olası günlük büyüklüğü üst sınırını azaltabilir.

#### **LogPrimaryFiles = 3 (varsayılan) |2-254 ( Windows ) |2-510 ( AIX and Linux sistemleri)**

Kuyruk yöneticisi yaratıldığında ayrılan günlük dosyaları.

Sahip olabileceğiniz birincil günlük kütüğü sayısı alt sınırı 2 'dir ve üst sınırı Windowssistemlerinde 254 ya da AIX and Linux sistemlerinde 510 'dur. Varsayılan 3'tür.

Birincil ve ikincil günlük dosyalarının toplam sayısı, Windowssistemlerinde 255 ya da AIX and Linux sistemlerinde 511 'i geçmemelidir ve 3 'ten az olmamalıdır.

Değer, kuyruk yöneticisi yaratıldığında ya da başlatıldığında incelenir. Kuyruk yöneticisi yaratıldıktan sonra değiştirebilirsiniz. Ancak, kuyruk yöneticisi yeniden başlatılıncaya kadar değerdeki bir değişiklik yürürlüğe girmez ve etki hemen başlamayabilir.

#### **LogSecondaryFiles = 2 (varsayılan) |1-253 ( Windows ) |1-509 ( AIX and Linux sistemleri)**

Birincil dosyalar tükendiğinde ayrılan günlük dosyaları.

İkincil günlük kütüğü sayısı alt sınırı 1 'dir ve üst sınırı Windowssistemlerinde 253 ya da AIX and Linux sistemlerinde 509 'dur. Varsayılan sayı 2 'dir.

Birincil ve ikincil günlük dosyalarının toplam sayısı, Windowssistemlerinde 255 ya da AIX and Linux sistemlerinde 511 'i geçmemelidir ve 3 'ten az olmamalıdır.

Değer, kuyruk yöneticisi başlatıldığında incelenir. Bu değeri değiştirebilirsiniz, ancak değişiklikler kuyruk yöneticisi yeniden başlatılıncaya kadar yürürlüğe girmez ve bu durumda bile etki hemen başlamayabilir.

#### **LogFileSayfalar = sayı**

Günlük verileri, günlük dosyaları adı verilen bir dizi dosyada tutulur. Günlük dosyası büyüklüğü 4 KB 'lik sayfa birimlerinde belirtilir.

Günlük dosyası sayfalarının varsayılan sayısı 4096 'dır ve günlük dosyası boyutu 16 MB 'dir.

AIX and Linux sistemlerinde günlük kütüğü sayfası sayısı alt sınırı 64, Windows sistemlerinde günlük kütüğü sayfası sayısı alt sınırı 32 'dir; her iki durumda da üst sınır 65 535 'tir.

**Not:** Kuyruk yöneticisi yaratılırken belirtilen günlük dosyalarının büyüklüğü, kuyruk yöneticisi için değiştirilemez.

#### **LogType = CIRCULAR (varsayılan) | LINEAR| REPLICATED**

Kuyruk yöneticisi tarafından kullanılacak günlük kaydı tipi. Varsayılan değer CIRCULAR' dır. Gerek duyduğunuz günlük kaydı tipiyle bir kuyruk yöneticisi yaratılmasıyla ilgili daha fazla bilgi için [“mq.s.ini dosyasının LogDefaults kısmı” sayfa 92](#) içindeki **LogType** özneliğinin tanımına bakın.

#### **Dairesel**

Sistem durduğunda devam eden hareketleri geri almak için günlüğü kullanarak yeniden başlatma işlemi başlatın.

Döngüsel günlüğe kaydetmeye ilişkin daha ayrıntılı açıklamalar için bkz. [“Günlüğe kaydetme tipleri” sayfa 628 .](#)

#### **Doğrusal**

Hem yeniden başlatma kurtarması hem de ortam ya da ileriye işleyerek kurtarma için (günlüğün içeriğini yeniden oynatarak kayıp ya da hasarlı veri yaratılması).

Doğrusal günlüğe kaydetmenin tam açıklaması için bkz. [“Günlüğe kaydetme tipleri” sayfa 628 .](#)

#### **CP4I REPLİKE EDİLEN**

Bir Yerel HA grubu tarafından etkin eşgörünümünden eşleme eşgörünümlerine günlük verilerini eşlemek için kullanılır.

Eşlenmiş günlüğe kaydetmenin tam açıklaması için bkz. [“Günlüğe kaydetme tipleri” sayfa 628 .](#)

**Not:** Bir kuyruk yöneticisinin **LogType** değeri, qm . ini dosyasında bu öznitelik değiştirilerek değiştirilemez. Bir kuyruk yöneticisinin **LogType** değerini değiştirmek için **migmqllog** komutunu kullanmanız gerekir.

### **LogBufferPages=0 (varsayılan) |0-4096**

Arabellek kayıtlarının yazılması için ayrılan bellek miktarı, arabelleklerin 4 KB sayfa birimi cinsinden büyüklüğünü belirtir.

Arabellek sayfası sayısı alt sınırı 18 ve üst sınırı 4096 'dır. Daha büyük arabellekler, özellikle daha büyük iletiler için daha yüksek verim sağlar.

0 (varsayılan) belirlerseniz, kuyruk yöneticisi büyüklüğü seçer.

1-17 aralığında bir sayı belirtirseniz, kuyruk yöneticisi varsayılan olarak 18 (72 KB) değerini alır. 18-4096 aralığında bir sayı belirtirseniz, kuyruk yöneticisi ayrılan bellek miktarını ayarlamak için belirtilen sayıyı kullanır.

Değer, kuyruk yöneticisi başlatıldığında incelenir. Değer, belirtilen sınırlar içinde artırılabilir veya azaltılabilir. Ancak, değerdeki bir değişiklik, kuyruk yöneticisinin bir sonraki başlatılışına kadar etkili olmaz.

### **LogPath= *dizin\_adi***

Bir kuyruk yöneticisine ilişkin günlük dosyalarının bulunduğu dizin. Bu, kuyruk yöneticisinin yazabileceği ve tercihen ileti kuyruklarından farklı bir sürücüye yazabileceği yerel bir aygıtta bulunmalıdır. Farklı bir sürücü belirtilmesi, sistem arızası durumunda ek koruma sağlar.

Varsayılan:

- **Windows** Windows içinde C : \ProgramData\IBM\MQ\log .
- **Linux** **AIX** AIX and Linux sistemlerinde /var/mqm/log .

-ld işaretini kullanarak **crtmqm** komutunda bir dizinin adını belirtebilirsiniz. Bir kuyruk yöneticisi yaratıldığında, kuyruk yöneticisi dizini altında da bir dizin yaratılır ve bu, günlük dosyalarını tutmak için kullanılır. Bu dizinin adı, kuyruk yöneticisi adını temel alır. Bu, günlük dosyası yolunun benzersiz olmasını ve dizin adı uzunluklarıyla ilgili sınırlamalara uymasını sağlar.

**crtmqm** komutunda -ld değerini belirtmezseniz, **LogDefaultPath** özneliğinin değeri kullanılır.

**Linux** **AIX** AIX and Linux sistemlerinde, kullanıcı kimliği mqm ve grup mqm, günlük dosyaları için tam yetkilere sahip olmalıdır. Bu dosyaların yerlerini değiştirirseniz, bu yetkililere kendiniz vermeniz gerekir. Günlük dosyaları ürünle birlikte sağlanan varsayılan konumlardaysa bu gerekli değildir.

### **LogWriteIntegrity =SingleWrite|DoubleWrite|TripleWrite (varsayılan)**

Günlük kaydedicinin günlük kayıtlarını güvenilir bir şekilde yazmak için kullandığı yöntem.

#### **TripleWrite (varsayılan)**

DoubleWrites seçeneğini belirleyebilirsiniz, ancak bunu yaparsanız sistem bunu TripleWrite olarak yorumlar.

#### **SingleWrite**

SingleWrite komutunu kullanmanız gerekir; ancak, IBM MQ kurtarma günlüğünü barındıran dosya sistemi ve aygıt 4KB ' nin atomlarına yazılmasını açık bir şekilde güvenceye alıyorsa.

Başka bir deyişle, herhangi bir nedenle 4KB sayfasının yazılması başarısız olduğunda, yalnızca iki olası durum işlem öncesi görüntüsü ya da sonraki görüntüdür. Hiçbir ara durum mümkün olmamalıdır.

**Not:** Kalıcı iş yükünde yeterli eşzamanlılık varsa, varsayılan değer olan TripleWrite dışında bir değer ayarlanmasında en düşük potansiyel avantaj vardır.

Daha fazla bilgi için bkz "[LogWriteIntegrity- SingleWrite ya da TripleWrite](#)" sayfa 132.

## LogManagement = E1 ile (varsayılan) | Otomatik | Arşiv

Günlük kapsamlarını el ile ya da kuyruk yöneticisi tarafından yönetmek için kullanılan yöntem. Varsayılan değer E1 ile' dir.

Öznitelik yalnızca **LogType** LINEAR olduğunda geçerlidir.

**LogManagement** değerini değiştirirseniz, kuyruk yöneticisi yeniden başlatılincaya kadar değişiklik yürürlüğe girmez.

Öznitelik için tanınmayan bir değer bulunursa, değer düzeltilinceye kadar kuyruk yöneticisi başlatılmaz.

**IBM i** **LogManagement** özelliği IBM üzerinde geçerli değil.

### E1 ile (varsayılan)

Günlük kapsamlarını el ile yönetebilirsiniz. Bu seçeneğin belirtilmesi, kuyruk yöneticisinin artık kurtarma için gerekli olmadığına bile günlük kapsamlarını yeniden kullanmadığı ya da silmediği anlamına gelir.

### Otomatik

Günlük kapsamı, kuyruk yöneticisi tarafından otomatik olarak yönetilir. Bu seçeneğin belirtilmesi, kuyruk yöneticisinin artık kurtarma için gerekli olmadığına anda günlük kapsamlarını yeniden kullanabileceği ya da silbileceği anlamına gelir. Arşivleme için izin verilmez.

### Arşivleme

Günlük kapsamı kuyruk yöneticisi tarafından yönetilir, ancak her günlük kapsamının arşivlenmesi tamamlandığında kuyruk yöneticisine bildirmeniz gerekir.

Bu seçeneğin belirtilmesi, kuyruk yöneticisine kurtarma için artık gerekli olmayan bir kapsamın arşivlendiği bildirilir bildirilmez, kuyruk yöneticisinin bir günlük kapsamını yeniden kullanmakta ya da silmekte serbest olduğu anlamına gelir.

Bu bildirim, **RESET QMGR MQSC** komutunu ya da Kuyruk Yöneticisini Sıfırla PCF komutunu kullanarak gerçekleştirilebilir.

## Örnek kısmı

```
Log:
LogPrimaryFiles=3
LogSecondaryFiles=2
LogFilePages=4096
LogType=CIRCULAR
LogBufferPages=0
LogPath=/var/mqm/log/saturn!queue!manager/
```

**Not: LogBufferPages** için sıfır değeri 512 değerini verir.

### **Multi** **LogWriteIntegrity- SingleWrite ya da TripleWrite**

**LogWriteIntegrity** seçeneğinin ayarı, qm. ini dosyasının Log (Günlük) kısmına, günlük kayıtlarını kurtarma günlüğüne yazmak için IBM MQ içinde kaydedici tarafından kullanılan algoritmayı belirler. Varsayılan ayar TripleWrite ' dir ve bu ayar neredeyse her olası senaryoda güvenlidir.

**LogWriteIntegrity** ayarı, yalnızca kısmi bir günlük sayfası yazılacağı zaman herhangi bir etkiye sahip olur. Makul miktarda eşzamanlı etkinlik içeren bir kuyruk yöneticisi için bu senaryo nadiren ortaya çıkar.

### SingleWrite

SingleWrite , çok olağandışı koşullarda varsayılan TripleWrite ayarından daha iyi performans gösterebilecek bir algoritma seçer. SingleWrite ayarı güvenlidir; ancak temeldeki depolama platformu, her koşulda IBM MQ kurtarma günlüğüne eşzamanlı olarak yazılan 4KB sayfalarının atomik olarak yazıldığını garanti edebiliyorsa.



SingleWrite ayarını kullanmanız gerekir; ancak, IBM MQ kurtarma günlüğünü barındıran dosya sistemi ya da aygıt 4KB 'nin atomlarına yazılmasını açık bir şekilde garanti ederse. Başka bir deyişle, 4KB sayfasının yazılması başarısız olduğunda, olası iki durum ya işlem öncesi görüntü ya da sonraki görüntü olmalıdır ve ara durum mümkün olmamalıdır. Diğer tüm durumlarda TripleWritekomutunu kullanmanız gerekir.

Yeterli koşutzamanlılığı olan bir sistemde kuyruk yöneticisi yalnızca günlük verilerinin tüm sayfalarını yazar ve tam sayfaların yüksek bir yüzdesi elde edilirse, SingleWrite ile TripleWritearasında önemli bir performans farkı yoktur.

Çok az eşzamanlılık içeren bir sistemde SingleWriteiçin önemli bir performans avantajı olabilir, ancak tercih edilen çözüm genellikle SingleWritekullanmak yerine koşutzamanlılığı artırmaktır.

4KB yazmanın atomizitesini güvenilir bir şekilde belirlemenin zor olabileceğini ve temel yazılımda ya da donanımda yapılan değişikliklerin bu tür bir garantiyi geçersiz kılabileceğini unutmayın.

Depolama altyapınızın gerekli garantileri şimdi ve gelecekte herhangi bir zamanda vereceğinden herhangi bir şüpheniz varsa, TripleWritekomutunu kullanmanız gerekir.

### Windows qm.ini dosyasının LU62 kısmı (yalnızcaWindows )

LU62 kısmı, SNA LU 6.2 protokolü yapılanış değiştirgelerini belirtiyor. Bu parametreler, kanallar için varsayılan öznitelikleri geçersiz kılar.

SNA LU 6.2 protokolü yapılanış değiştirgelerini belirtmek için qm . ini kütüğündeki LU62 kısmına bakın. Kanallar için varsayılan öznitelikleri geçersiz kılar.

Linux Windows Alternatif olarak, Linux (x86 ve x86-64) ve Windowsistemlerinde IBM MQ Explorer LU6 . 2 kuyruk yöneticisi özellikleri sayfasını kullanın.

#### TPAdı

Uzak sitede başlatılacak TP adı.

#### Library1= DLLName 1

APPC DLL 'nin adı.

Varsayılan değer WCPIC32' dir.

#### Library2= DLLName2

Kod iki ayrı kitaplıkta saklandıysa, Library1ile aynı.

Varsayılan değer WCPIC32' dir.

### CP4I NativeHAInstance qm.ini dosyasının kısmı

Kapsayıcılarda IBM MQ için NativeHAInstance kısmı, Yerel HA yapılandırmasındaki üç düğümün birbiriyle nasıl iletişim kurabileceğini belirtir.

**Not:** Bu bilgiler yalnızca kapsayıcı ortamlar için geçerlidir. IBM MQ İşletmeni kullanılarak Yerel HA 'nın yapılandırılması ya da Kendi kapsayıcılarınızı oluşturuyorsanız Yerel HA grubunun yaratılması başlıklı konuya bakın.

Yerel HA grubundaki (yerel yönetim ortamı da içinde olmak üzere) her kuyruk yöneticisi yönetim ortamı için bir tane olmak üzere üç NativeHAInstance kısmı eklersiniz. Aşağıdaki öznitelikleri ekleyin:

#### Ad

Kuyruk yöneticisi yönetim ortamını yaratırken kullandığınız yönetim ortamı adını belirtin.

#### ReplicationAddress

Eşgörünümün anasistem adını, IPv4 noktalı onlu ya da IPv6 onaltılı biçim adresini belirleyin.

Adresi anasistem adı, IPv4 noktalı onlu ya da IPv6 onaltılı biçim adresi olarak belirleyebilirsiniz.

Eşleme adresi, gruptaki her eşgörünümünden çözülebilir ve yönlenebilir olmalıdır. Günlük eşleme için kullanılacak kapı numarası köşeli ayraç içinde belirtilmelidir; örneğin:

```
ReplicationAddress=host1.example.com(4444)
```

## Örnek kısım

Aşağıdaki örnekte, `qm.ini` dosyasında yerel bir HA yapılandırmasının üç düğümünü belirtmek için kullanılan `NativeHAInstance` kısmı gösterilmektedir.

```
NativeHAInstance:
  Name=node-1
  ReplicationAddress=host1.example.com(4444)
NativeHAInstance:
  Name=node-2
  ReplicationAddress=host2.example.com(4444)
NativeHAInstance:
  Name=node-3
  ReplicationAddress=host3.example.com(4444)
```

## İlgili kavramlar

“`NativeHALocalInstance` `qm.ini` dosyasının kısmı” sayfa 134

Kapsayıcılarda IBM MQ için `NativeHALocalInstance` kısmı, Yerel HA yapılandırmasının çalışmasını denetler.

### CP4I

## NativeHALocalInstance `qm.ini` dosyasının kısmı

Kapsayıcılarda IBM MQ için `NativeHALocalInstance` kısmı, Yerel HA yapılandırmasının çalışmasını denetler.

**Not:** Bu bilgiler yalnızca kapsayıcı ortamlar için geçerlidir. [IBM MQ İşletmeni](#) kullanılarak Yerel HA 'nın yapılandırılması ya da [Kendi kapsayıcılarınızı oluşturuyorsanız Yerel HA grubunun yaratılması](#) başlıklı konuya bakın.

`NativeHALocalInstance` kısmı, bir Yerel HA yapılandırması oluşturduğunuzda her bir düğümdeki `qm.ini` dosyasına otomatik olarak eklenir. Daha sonra `qm.ini` dosyasını düzenleyebilir ve `NativeHALocalInstance` kıtasındaki öznitelikleri özelleştirebilirsiniz.

### LocalName

Yerel HA kuyruk yöneticisi yaratıldığında belirtilen günlük eşlemesi eşgörünümü adından alınan `NativeHALocalInstance` kısmı adı.

İsteğe bağlı olarak, `NativeHALocalInstance` kısmı için aşağıdaki öznitelikleri ekleyebilirsiniz:

### KeyRepository

#### V 9.3.0

#### V 9.3.0

Günlük eşleme trafiğini korumak için kullanılan sayısal sertifikayı tutan anahtar havuzunun tam yolu ve dosya adı. Dosya uzantısı belirtilmezse, dosya uzantısının `.kdb` olduğu varsayılır.

`KeyRepository` kısmı özniteliği atlanırsa, günlük eşleme verileri düz metindeki eşgörünümler arasında değiştirilir.

#### V 9.3.2

### KeyRepositoryPassword

Anahtar havuzu, hassas bilgiler içerdiği için bir parolayla korunmuştur. Anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını alabilmelidir. Parola bir anahtar havuzu parola saklama dosyasında saklanmıyorsa, parolayı `KeyRepositoryPassword` özniteliğinde verebilirsiniz. Örneğin:

```
KeyRepositoryPassword=passw0rd
```



**Uyarı:** Parolayı bu özniteliği kullanarak girdiyerseniz, parolayı IBM MQ parola koruma sistemiyle şifreleyin. Daha fazla bilgi için bkz. [“Anahtar havuzu parolasını şifreleme”](#) sayfa 136.

#### V 9.3.2

### InitialKeyFile

`KeyRepositoryPassword` özniteliğiyle belirtilen anahtar havuzu parolası belirli bir başlangıç anahtarıyla şifrelenmişse bu özniteliği belirtin. İlk anahtarı içeren dosyanın adı, anahtar havuzu parolasını şifrelemek için `runmqicred` komutu kullanıldığında `-sf` parametresi kullanılarak belirtilebilir.

Bu özneliğin deęerini, parolayı şifrelemek için kullanılan ilk anahtarı içeren dosyanın adına ayarlayın. Örneęin, mykey . key adlı bir dosya ilk anahtarı içeriyorsa:

```
InitialKeyFile=/mykey.key
```

Daha fazla bilgi için bkz “Anahtar havuzu parolasını şifreleme” sayfa 136.

### **CertificateLabel**

Günlük eşleme trafięinin korunması için kullanılacak dijital sertifikayı tanımlayan sertifika etiketi. KeyRepository sağlanıyorsa, ancak CertificateLabel atlanırsa, varsayılan deęer olan ibmwebspheremqueue\_manager kullanılır.

### **CipherSpec**

Günlük eşleme trafięini korumak için kullanılacak CipherSpec . Bu dörtlük öznelięi sağlanırsa, KeyRepository da sağlanmalıdır. KeyRepository sağlanıyorsa, ancak CipherSpec atlanırsa, varsayılan deęer olan ANY kullanılır.

### **LocalAddress**

Günlük eşleme trafięini kabul eden yerel ağ arabirimi adresi. Bu öznelik sağlanırsa, "[ addr] [ (port)]" biçimini kullanarak yerel ağ arabirimini ve/veya kapıyı tanımlar. Ağ adresi anasistem adı, IPv4 noktalı onlu ya da IPv6 onaltılı biçimi olarak belirlenebilir. Bu öznelik atlanırsa, kuyruk yöneticisi tüm ağ arabirimlerine bağlanmayı dener, yerel eşgörünüm adıyla eşleşen NativeHAInstances kısmı içindeki ReplicationAddress içinde belirtilen kapıyı kullanır.

### **HeartbeatInterval**

Saęlıklı işletim bildirim aralıęı, Yerel HA kuyruk yöneticisinin etkin bir eşgörünümün ağ saęlıklı işletim bildirim gönderme sıklıęını milisaniye cinsinden tanımlar. Saęlıklı işletim bildirim aralıęı deęerinin geçerli aralıęı 500 (0.5 saniye) ile 60000 (1 dakika) arasındadır; bu aralıęın dışındaki bir deęer kuyruk yöneticisinin başlatılmamasına neden olur. Bu öznelik atlanırsa, 5000 (5 saniye) varsayılan deęeri kullanılır. Her eşgörünüm aynı saęlıklı işletim bildirim aralıęını kullanmalıdır.

### **HeartbeatTimeout**

Saęlıklı işletim bildirim zaman aşımı, bir Yerel HA kuyruk yöneticisinin etkin eşgörünümün yanıt vermedięine karar vermeden önce eşlemenin ne kadar bekleyeceęini tanımlar. Saęlıklı işletim bildirim zaman aşımı deęerinin geçerli aralıęı 500 (0.5 saniye) ile 120000 (2 dakika) arasındadır. Saęlıklı işletim bildirim zaman aşımının deęeri, saęlıklı işletim bildirim aralıęından büyük ya da bu aralıęa eşit olmalıdır.

Geçersiz bir deęer, kuyruk yöneticisinin başlatılmasına neden olur. Bu öznelik atlanırsa, eşleme işlemi başlatmadan önce yeni bir etkin eşgörünüm seçmek için 2 x HeartbeatInterval bekler. Her eşgörünüm aynı saęlıklı işletim bildirim zaman aşımını kullanmalıdır.

### **RetryInterval**

Yeniden deneme aralıęı, bir Yerel HA kuyruk yöneticisinin başarısız bir eşleme bağlantısını ne sıklıkta yeniden denemesi gerektięini milisaniye cinsinden tanımlar. Yeniden deneme aralıęının geçerli aralıęı 500 (0.5 saniye) ile 120000 (2 dakika) arasındadır. Bu öznelik atlanırsa, eşleme başarısız bir eşleme bağlantısını yeniden denemeden önce 2 x HeartbeatInterval bekler.

### **SSLFipsRequired**

Günlük eşleme trafięinin gönderilmesinde şifreleme kullanılıyorsa yalnızca FIPS onaylı algoritmaların kullanılıp kullanılmayacaęını belirtir. Yes ya da Noolarak ayarlayın.

### **EncryptionPolicySuiteB**

Günlük eşleme trafięinin Suite-B uyumlu şifreleme kullanıp kullanmadıęını ve hangi güvenlik düzeyinin kullanılıp kullanılmadıęını belirtir. Aşaęıdaki deęerlerden birine ayarlayın:

#### **NONE**

Suite-B uyumlu şifreleme kullanılmaz. Bu ayar varsayılan ayardır.

#### **128\_BIT, 192\_BIT**

Güvenlik düzeyini hem 128 bit hem de 192 bit düzeylerine ayarlar.

#### **128\_BIT**

Güvenlik düzeyini 128 bit düzeyine ayarlar.

## 192\_BIT

Güvenlik düzeyini 192 bit düzeyine ayarlar.

### Anahtar havuzu parolasını şifreleme

V 9.3.2

Anahtar havuzu parolası, IBM MQ parola koruma sistemi ya da bir anahtar havuzu parola saklama dosyası kullanılarak korunabilir. Bu iki yöntemle ilgili daha fazla bilgi için [Anahtar havuzu parolalarının şifrelenmesi](#) başlıklı konuya bakın.

Havuz parolası NativeHALocalInstance kısmı içindeki KeyRepositoryPassword özneliği kullanılarak belirtilirse, IBM MQ parola koruma sistemini kullanarak parolayı şifreleyin. Parolayı şifrelemek için **runmqicred** komutunu kullanın. Komut, KeyRepositoryPassword özneliğinde belirtilebilir şifrelenmiş parolayı döndürür.

Parolayı güvenli bir şekilde şifrelemek için benzersiz bir başlangıç anahtarı kullanın. İlk anahtarı içeren dosyanın adı, **runmqicred** komutunda **-sf** parametresi kullanılarak belirtilebilir. Benzersiz bir anahtar belirtmezseniz, varsayılan anahtar kullanılır.

Anahtar havuzu parolasını benzersiz bir başlangıç anahtarıyla şifrelediyseniz, aynı ilk anahtarı NativeHALocalInstance kısmına InitialKeyFile özneliğini kullanarak da sağlamanız gerekir.

### Örnek kısmı

Aşağıdaki örnekte, qm.ini dosyasında bir düğümün yerel adını belirtmek için kullanılan NativeHALocalInstance kısmı gösterilmektedir.

```
NativeHALocalInstance:  
LocalName=node-1
```

### İlgili kavramlar

“NativeHAInstance qm.ini dosyasının kısmı” sayfa 133

Kapsayıcılarda IBM MQ için NativeHAInstance kısmı, Yerel HA yapılandırmasındaki üç düğümün birbiriyle nasıl iletişim kurabileceğini belirtir.

### İlgili başvurular

[runmqicred \(IBM MQ Client parolalarını koru\)](#)

### Windows qm.ini dosyasının NETBIOS kısmı (yalnızca Windows)

qm.ini dosyasındaki NETBIOS kısmı, NetBIOS iletişim kuralı yapılandırma parametrelerini belirtir. Bu parametreler, kanallar için varsayılan öznelikleri geçersiz kılar.

NetBIOS protokolü yapılanış değiştirgelerini belirtmek için qm.ini kütüğündeki NETBIOS kısmına bakın. Kanallar için varsayılan öznelikleri geçersiz kılar.

Linux

Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer Netbios kuyruk yöneticisi özellikleri sayfasını kullanın.

### LocalName= ad

Bu makinenin LAN üzerinde bilindiği ad.

### AdapterNum = 0 (varsayılan) | bağdaştırıcı\_no

LAN bağdaştırıcısının numarası. Varsayılan değer bağdaştırıcı 0 'dir.

### NumSess = 1 (varsayılan) | number\_of\_sessions

Ayrılacak oturum sayısı. Varsayılan değer 1'dir.

### NumCmds = 1 (varsayılan) | number\_of\_commands

Ayrılacak komutların sayısı. Varsayılan değer 1'dir.

### NumNames = 1 (varsayılan) | number\_of\_names

Ayrılacak ad sayısı. Varsayılan değer 1'dir.

## Library1= *DLLName1*

NetBIOS DLL 'inin adı.

Varsayılan değer NETAPI32' dir.

## İlgili kavramlar

“IBM MQ Yerel NetBIOS adının tanımlanması” sayfa 254

IBM MQ kanal işlemleri tarafından kullanılan yerel NetBIOS adı üç şekilde belirtilebilir.

Linux

AIX

## qm.ini dosyasının RestrictedMode kısmı

RestrictedMode kısmı, MQI uygulamalarını çalıştırmasına, tüm IPCC kaynaklarını güncellemesine ve bazı kuyruk yöneticisi dizinlerinin içeriğini değiştirmesine izin verilen üyeleri içeren grubun adını belirtir. Bu kıta yalnızca AIX and Linux sistemleri için geçerlidir.

RestrictedMode kısmı, **crtmqm** komutundaki **-g** seçeneği tarafından ayarlanır. **-g** seçeneğini kullanmazsanız, qm . ini dosyasında stanza oluşturulmaz.

IBM MQ uygulamalarının kuyruk yöneticisi veri dizini içindeki kuyruk yöneticisine bağlı olarak dosya oluşturdukları bazı dizinler vardır. Uygulamaların bu dizinlerde dosya oluşturmaları için bunlara dünya çapında yazma erişimi verilir:

- /var/mqm/sockets/QMgrName/@ipcc/ssem/hostname/
- /var/mqm/sockets/QMgrName/@app/ssem/hostname/
- /var/mqm/sockets/QMgrName/zsocketapp/hostname/

Burada QMGRNAME , kuyruk yöneticisinin adı, hostname ise anasistem adıdır.

Bazı sistemlerde, tüm kullanıcılara bu dizinlere yazma erişimi verilmesi kabul edilemez. Örneğin, kuyruk yöneticisine erişmesi gerekmeyen kullanıcılar. Kısıtlı kip, kuyruk yöneticisi verilerini saklayan dizinlerin izinlerini değiştirir. Daha sonra dizinlere yalnızca belirtilen uygulama grubunun üyeleri erişebilir. Kuyruk yöneticisiyle iletişim kurmak için kullanılan System V IPC paylaşılan belleğine ilişkin izinler de aynı şekilde değiştirilir.

Uygulama grubu, aşağıdaki işlemleri yapma iznine sahip üyeleri olan grubun adıdır:

- MQI uygulamalarını çalıştır
- Tüm IPCC kaynaklarını güncelle
- Bazı kuyruk yöneticisi dizinlerinin içeriğini değiştir

Bir kuyruk yöneticisi için kısıtlı kipi kullanmak için:

- Kuyruk yöneticisini yaratan kişi, mqm grubunda ve uygulama grubunda olmalıdır.
- mqm kullanıcı kimliği uygulama grubunda olmalıdır.
- Kuyruk yöneticisini denetlemek isteyen tüm kullanıcılar, mqm grubunda ve uygulama grubunda olmalıdır.
- IBM MQ uygulamalarını çalıştırmak isteyen tüm kullanıcılar uygulama grubunda olmalıdır.

Uygulama grubunda olmayan bir kullanıcı tarafından verilen herhangi bir MQCONN ya da MQCONNX çağırısı, MQRC\_Q\_MGR\_NOT\_KULLANILABİLİR neden koduyla başarısız olur.

**Önemli:** Birçok işletim sisteminde, bir kullanıcının bir gruba eklenmesinin tanınması için söz konusu kullanıcının oturumu kapatması ve yeniden oturum açması gerekir.

Kısıtlı kip, IBM MQ yetkilendirme hizmetiyle çalışır. Bu nedenle, kullanıcılara IBM MQ ' e bağlanma ve IBM MQ yetkilendirme hizmetini kullanarak gereksinim duydukları kaynaklara erişme yetkisi de vermeniz gerekir.

ALW

IBM MQ yetkilendirme hizmetinin yapılandırılmasına ilişkin ek bilgi için [AIX, Linux, and Windows sistemlerinde güvenliğin ayarlanması](#) kısmına bakın.

IBM MQ kısıtlı kipini yalnızca, yetkilendirme hizmeti tarafından sağlanan denetim kuyruk yöneticisi kaynaklarının yeterli yalıtımını sağlamadığında kullanın.

## İlgili başvurular

[crtmqm](#) (kuyruk yöneticisi yarat)

Multi

### qm.ini dosyasının güvenlik kısmı

Güvenlik kısmı, Nesne Yetkilisi Yöneticisi 'ne (OAM) ilişkin seçenekleri belirtir.

#### **ClusterQueueAccessControl=RQMName | Xmitq**

Küme kuyruğu yöneticilerinin bulunduğu küme kuyruklarının ya da tam olarak nitelenmiş kuyrukların erişim denetimini denetlemek için bu özneliği ayarlayın.

#### **RQMName**

Uzaktan barındırılan kuyrukların erişim denetimi için denetlenen tanımlar, adlandırılmış kuyruk yöneticisi tanımları ya da adlandırılmış kuyruk yöneticisi tanımlarıdır.

#### **Xmitq.**

Uzaktan barındırılan kuyrukların erişim denetimi için denetlenen tanımlar SYSTEM.CLUSTER.TRANSMIT.QUEUEolarak çözülür.

Xmitq varsayılan değerdir.

Windows

### GroupModel=GlobalGroups

Bu öznelik, Windowsüzerinde bir kullanıcının grup üyeliğini belirlerken OAM ' in genel grupları denetleyip denetmeyeceğini belirler.

Varsayılan değer, genel grupların denetlenmemesidir.

#### **GlobalGroups**

OAM, genel grupları denetler.

GlobalGroups ayarlandığında, yetkilendirme komutları, **setmqaut**, **dspmqaut**ve **dmpmqaut** genel grup adlarını kabul eder; bkz. **setmqaut -g** parametresi.

**Not:** ClusterQueueAccessControl=RQMName 'in ayarlanması ve MQZAS\_VERSION\_6 ' den daha az bir yetki hizmetinin özel bir somutlaması olması kuyruk yöneticisinin başlatılmamasına neden olur. Bu örnekte ClusterQueueAccessControl=Xmitq ayarını tanımlayın ya da özel Yetki Hizmeti 'ni MQZAS\_VERSION\_6 ya da daha sonraki bir sürüme yükseltin.

## Örnek kısmı

```
Security:
  ClusterQueueAccessControl=Xmitq
  GroupModel=GlobalGroups
```

Multi

### qm.ini dosyasının hizmet kısmı

Hizmet kısmı, kurulabilir hizmetlerde değişiklik yapmak için kullanılır. Bu kıta, hizmetin adını ve hizmet için tanımlanan giriş noktalarının sayısını içerir.

**Not:** Linux Windows Kurulabilir hizmetleri ve bunların bileşenlerini değiştirmenin önemli etkileri vardır. Bu nedenle, kurulabilir hizmetler IBM MQ Exploreriçinde salt okunur olur.

Bir hizmet içindeki her bileşen için, o bileşene ilişkin kodu içeren modülün adını ve yolunu da belirtmeniz gerekir. Bunun için [ServiceComponent](#) kısmı kullanın.

**Service** ve **ServiceComponent** kısmı herhangi bir sırada olabilir ve bunların altındaki kıta anahtarları da herhangi bir sırada olabilir. Bu iki kıta için de tüm kıta anahtarları mevcut olmalı. Bir kıta anahtarı yinelenirse, sonuncusu kullanılır.

Başlatma sırasında kuyruk yöneticisi, yapılandırma dosyasındaki her hizmet bileşeni girişini sırayla işler. Daha sonra, belirtilen bileşen modülünü yükler ve bileşenin giriş noktasını çağırıp (bileşenin kullanıma hazırlanması için giriş noktası olması gerekir), bir yapılandırma tanıtıcısı geçirilir.

## Ad = AuthorizationService (varsayılan) |NameService

Gerekli hizmetin adı.

### AuthorizationService

IBM MQ için, **AuthorizationService** bileşeni nesne yetki yöneticisi ya da OAM olarak bilinir. **Service** kısmı ve ilişkili **ServiceComponent** kısmı, kuyruk yöneticisi yaratıldığında otomatik olarak eklenir, ancak *MQSNOAUT* ortam değişkeni tarafından geçersiz kılınabilir. Diğer **ServiceComponent** stanzalarını el ile ekleyin.

**Linux** **AIX** `qm.ini` dosyasındaki aşağıdaki stanzas örnekleri, IBM MQ for AIX üzerinde iki yetkilendirme hizmeti bileşenini tanımlar. *MQ\_INSTALLATION\_PATH*, IBM MQ 'in kurulu olduğu üst düzey dizini gösterir.

```
Service:
  Name=AuthorizationService
  EntryPoints=13

ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.UNIX.auth.service
Module=MQ_INSTALLATION_PATH/lib/amqzfu
  ComponentDataSize=0

ServiceComponent:
  Service=AuthorizationService
  Name=user.defined.authorization.service
  Module=/usr/bin/udas01
  ComponentDataSize=96
```

**Linux** **AIX** `ServiceComponent` kısmı `MQSeries.UNIX.auth.service` varsayılan yetkilendirme hizmeti bileşenini (OAM) tanımlar. Bu kısmı kaldırır ve kuyruk yöneticisini yeniden başlatırsanız, OAM devre dışı bırakılır ve yetki denetimi yapılmaz.

**Windows** **SecurityPolicy** özniteliğini IBM MQ hizmetlerini kullanarak da ekleyebilirsiniz. **SecurityPolicy** özniteliği, yalnızca Hizmet kısmı için belirtilen hizmet yetkilendirme hizmeti (varsayılan OAM) ise geçerlidir. **SecurityPolicy** özniteliği, her kuyruk yöneticisi için güvenlik ilkesini belirtmenizi sağlar. Olası değerler şunlardır:

### Varsayılan

Varsayılan güvenlik ilkesinin geçerli olmasını istiyorsanız `Varsayılan` seçeneğini belirleyin. Belirli bir kullanıcı kimliği için OAM 'ye Windows güvenlik tanıtıcısı (NT SID) geçirilmezse, ilgili güvenlik veritabanlarında arama yapılarak uygun SID' yi alma girişiminde bulunulur.

### NTSIDsRequired

Güvenlik denetimleri gerçekleştirilirken OAM ' ye bir NT SID iletilmesini gerektirir.

**Windows** `ServiceComponent` kısmı `MQSeries.WindowsNT.auth.service` varsayılan yetkilendirme hizmeti bileşenini (OAM) tanımlar. Bu kısmı kaldırır ve kuyruk yöneticisini yeniden başlatırsanız, OAM devre dışı bırakılır ve yetki denetimi yapılmaz.

### NameService

Varsayılan olarak bir ad hizmeti sağlanmaz. Bir ad hizmetine gereksinim duyarsanız, `NameService` kısmı el ile eklenmelidir.

**Linux** **AIX** Ad hizmeti için aşağıdaki AIX and Linux `qm.ini` dosya kısmı örnekleri, ABC şirketi tarafından sağlanan bir ad hizmeti bileşenini belirtir.

```
# Stanza for name service
Service:
  Name=NameService
  EntryPoints=5

# Stanza for name service component, provided by ABC
ServiceComponent:
  Service=NameService
```



```
Name=ABC.Name.Service
Module=/usr/lib/abcname
ComponentDataSize=1024
```

**Not:** **Windows** Windows sistemlerinde NameService kısmı bilgileri Kayıt Defteri 'nde saklanır.

### **EntryPoints= giriş sayısı**

Hizmet için tanımlanan giriş noktalarının sayısı.

Bu, kullanıma hazırlama ve sonlandırma giriş noktalarını içerir.

### **Windows SecurityPolicy= Varsayılan |NTSIDsRequired**

Windows sistemlerinde **SecurityPolicy** özneliği yalnızca, belirtilen hizmet varsayılan yetkilendirme hizmeti (OAM) ise geçerlidir. **SecurityPolicy** özneliği, her kuyruk yöneticisi için güvenlik ilkesini belirtmenizi sağlar.

Olası değerler şunlardır:

#### **Varsayılan**

Geçerli olmak için varsayılan güvenlik ilkesini kullanın. Belirli bir kullanıcı kimliği için OAM 'ye Windows güvenlik tanıtıcısı (NT SID) geçirilmezse, ilgili güvenlik veritabanlarında arama yapılarak uygun SID' yi alma girişiminde bulunulur.

#### **NTSIDsRequired**

Güvenlik denetimleri gerçekleştirirken OAM ' ye bir NT SID geçirin.

Daha fazla bilgi için bkz. [Windows güvenlik tanıtıcıları \(SID\)](#).

Ayrıca bkz. [Configuring authorization service stanzas: Windows systems](#).

### **Linux AIX SecurityPolicy=user|group|UserExternal|default**

AIX and Linux sistemlerinde bu değer, kuyruk yöneticisinin kullanıcı tabanlı ya da grup tabanlı yetki kullanıp kullanmadığını belirtir. Değerler büyük ve küçük harfe duyarlı değildir.

Değer aşağıdaki değerlerden biri olabilir:

#### **grup**

Kuyruk yöneticisi, grup tabanlı yetki kullanır. Bir kaynağa erişim yetkisi bir gruba verilir.

Bir kullanıcı, ait olduğu her gruba verilen tüm yetkilerin toplamını alır.

Kullanıcı kimlikleri ve gruplar yerel işletim sistemine tanımlanmalıdır.

#### **kullanıcı**

Kuyruk yöneticisi, kullanıcı tabanlı yetkilendirilen üyünü göstermek (esasir) için kullanılır. Bir kaynağa erişim yetkisi bir gruba ya da belirli bir kullanıcı kimliğine verilebilir.

Bir kullanıcı aşağıdaki yetkilerin toplamını alır:

- Belirli kullanıcıya verilen yetkiler.
- Kullanıcının ait olduğu her gruba verilen yetkiler.

Kullanıcı kimlikleri ve gruplar yerel işletim sistemine tanımlanmalıdır.

### **V 9.3.0 UserExternal**

Kuyruk yöneticisi, kullanıcı tabanlı yetkilendirilen üyünü göstermek (esasir) için kullanılır. Ancak, yerel işletim sistemi tarafından bilinmeyen kullanıcı kimliklerine yetkiler verilebilir.

Bir kaynağa erişim yetkisi bir gruba ya da belirli bir kullanıcı kimliğine verilebilir.

Bir kullanıcı aşağıdaki yetkilerin toplamını alır:

- Belirli kullanıcıya verilen yetkiler.
- Kullanıcının ait olduğu her gruba verilen yetkiler.

Bir kullanıcı yerel işletim sistemi tarafından bilinmezse, yalnızca hiç kimse grubuna ait olduğu kabul edilir. Gruplarla ilgili daha fazla bilgi için bkz. [AIX, Linux, and Windows üzerindeki birincil kullanıcılar ve gruplar](#). Kullanıcı kimliği en çok 12 karakter uzunluğunda olmalı ve [IBM MQ nesnelere adlandırma kurallarına](#) uygun olmalıdır.



Var olan kuyruk yöneticilerini, yürürlükteki yapılanışı kaybetmeden bu ek seçeneği kullanacak şekilde değiştirebilirsiniz.

► **V 9.3.4** AuthToken kısmı belirtilirse, bu varsayılan değerdir.

#### **varsayılan**

Kuyruk yöneticisi, grup tabanlı yetki kullanır. Davranış, `group` seçeneğiyle aynıdır.

AuthToken kısmı belirtilmezse, bu varsayılan değerdir.

Öznitelik değerinde yapılan değişikliklerin yürürlüğe girmesi için kuyruk yöneticisini yeniden başlatın.

**Not:** ► **Linux** ► **AIX** ► **V 9.3.4** IBM MQ 9.3.4' den AuthToken kısmı belirtilirse, Hizmet kısmına ilişkin **SecurityPolicy** özneliğinin geçerli değeri `UserExternal` olarak ayarlanır. **SecurityPolicy** , Hizmet kısmı içinde belirtik olarak `Grup` olarak ayarlanırsa belirteç kimlik doğrulaması kullanılamaz. **SecurityPolicy** `Grup` olarak ayarlanırsa, **SecurityPolicy** özneliğini Hizmet dördünden kaldırın ve kuyruk yöneticisini yeniden başlatın. Daha fazla bilgi için, bkz. "[qm.ini](#) dosyasının AuthToken kısmı" sayfa 110.

#### **SharedBindingsUserId= kullanıcı-tipi**

**SharedBindingsUserId** özneliği yalnızca, belirtilen hizmet varsayılan yetkilendirme hizmeti (OAM) ise geçerlidir. **SharedBindingsUserId** özneliği yalnızca paylaşılan bağ tanımlarıyla ilişkili olarak kullanılır. Bu değer, `MQZ_AUTHENTICATE_USER` işlevindeki *IdentityContext* yapısındaki *UserIdentifier* alanının etkin kullanıcı kimliği mi, yoksa gerçek kullanıcı kimliği mi olduğunu belirtmenizi sağlar.

`MQZ_AUTHENTICATE_USER` işleviyle ilgili bilgi için [MQZ\\_AUTHENTICATE\\_USER-Kullanıcının Kimliğini Doğrulabaşlıklı konuya](#) bakın.

Olası değerler şunlardır:

#### **Varsayılan**

*UserIdentifier* alanının değeri, gerçek kullanıcı kimliği olarak ayarlanır.

#### **Real**

*UserIdentifier* alanının değeri, gerçek kullanıcı kimliği olarak ayarlanır.

#### **Yürürlük Tarihi**

*UserIdentifier* alanının değeri, geçerli kullanıcı kimliği olarak ayarlanır.

#### **FastpathBindingsUserId= kullanıcı-tipi**

**FastpathBindingsUserId** özneliği yalnızca, belirtilen hizmet varsayılan yetkilendirme hizmeti (OAM) ise geçerlidir. **FastpathBindingsUserId** özneliği, yalnızca fastpath bağ tanımlarıyla ilişkili olarak kullanılır. Bu değer, `MQZ_AUTHENTICATE_USER` işlevindeki *IdentityContext* yapısındaki *UserIdentifier* alanının etkin kullanıcı kimliği mi, yoksa gerçek kullanıcı kimliği mi olduğunu belirtmenizi sağlar.

`MQZ_AUTHENTICATE_USER` işleviyle ilgili bilgi için [MQZ\\_AUTHENTICATE\\_USER-Kullanıcının Kimliğini Doğrulabaşlıklı konuya](#) bakın.

Olası değerler şunlardır:

#### **Varsayılan**

*UserIdentifier* alanının değeri, gerçek kullanıcı kimliği olarak ayarlanır.

#### **Real**

*UserIdentifier* alanının değeri, gerçek kullanıcı kimliği olarak ayarlanır.

#### **Yürürlük Tarihi**

*UserIdentifier* alanının değeri, geçerli kullanıcı kimliği olarak ayarlanır.

#### **IsolatedBindingsUserId= kullanıcı-tipi**

**IsolatedBindingsUserId** özneliği yalnızca, belirtilen hizmet varsayılan yetkilendirme hizmeti (OAM) ise geçerlidir. **IsolatedBindingsUserId** özneliği yalnızca yalıtılmış bağ tanımlarıyla ilişkili olarak kullanılır. Bu değer, `MQZ_AUTHENTICATE_USER` işlevindeki *IdentityContext* yapısındaki *UserIdentifier* alanının etkin kullanıcı kimliği mi, yoksa gerçek kullanıcı kimliği mi olduğunu belirtmenizi sağlar.

MQZ\_AUTHENTICATE\_USER işleviyle ilgili bilgi için [MQZ\\_AUTHENTICATE\\_USER-Kullanıcının Kimliğini Doğrulabaşlıklı konuya](#) bakın.

Olası değerler şunlardır:

**Varsayılan**

*UserIdentifier* alanının değeri, geçerli kullanıcı kimliği olarak ayarlanır.

**Real**

*UserIdentifier* alanının değeri, gerçek kullanıcı kimliği olarak ayarlanır.

**Yürürlük Tarihi**

*UserIdentifier* alanının değeri, geçerli kullanıcı kimliği olarak ayarlanır.

Kurulabilir hizmetler ve bileşenlerle ilgili daha fazla bilgi için bkz. [AIX, Linux, and Windows için Kurulabilir hizmetler ve bileşenler](#).

Genel olarak güvenlik hizmetleriyle ilgili daha fazla bilgi için [AIX and Linux sistemlerinde güvenliği ayarlamabaşlıklı konuya](#) bakın.

## Örnek kısmı

```
Service:  
Name=AuthorizationService  
EntryPoints=14
```

### İlgili kavramlar

[AIX, Linux ve Windows için kurulabilir hizmetler ve bileşenler](#)

### İlgili başvurular

[IBM i üzerinde kurulabilir hizmetler ve bileşenler](#)

[Kurulabilir hizmetler için başvuru bilgileri](#)

Multi

## qm.ini dosyasının ServiceComponent kısmı

ServiceComponent kısmı, hizmet bileşenine ilişkin bilgileri belirtir. Yeni bir kurulabilir hizmet eklerken hizmet bileşeni bilgilerini belirtmeniz gerekir. Yetkilendirme hizmeti kısmı varsayılan olarak vardır ve ilişkili bileşen olan OAM etkindir.

**Service** ve **ServiceComponent** kısmı herhangi bir sırada olabilir ve bunların altındaki kıta anahtarları da herhangi bir sırada olabilir. Bu iki kıta için de tüm kıta anahtarları mevcut olmalı. Bir kıta anahtarı yinelenirse, sonuncusu kullanılır.

Başlatma sırasında kuyruk yöneticisi, yapılandırma dosyasındaki her hizmet bileşeni girişini sırayla işler. Daha sonra, belirtilen bileşen modülünü yükler ve bileşenin giriş noktasını çağırıp (bileşenin kullanıma hazırlanması için giriş noktası olması gerekir), bir yapılandırma tanıtıcısı geçirilir.

### Hizmet = *hizmet\_adi*

Gerekli hizmetin adı. Bu, Hizmet yapılandırma bilgilerinin Name özniteliğinde belirtilen değerle eşleşmelidir.

### Ad = *bileşen\_adi*

Hizmet bileşeninin açıklayıcı adı. Bu, benzersiz olmalı ve yalnızca IBM MQ nesnelerinin adları (örneğin, kuyruk adları) için geçerli olan karakterleri içermelidir. Bu ad, hizmet tarafından oluşturulan işletmen iletilerinde görülür. Bu adın bir şirket ticari markası veya benzer ayırt edici bir dizeyle başlamasını öneririz.

### Modül = *modül\_adi*

Bu bileşene ilişkin kodu içerecek modülün adı. Bu bir tam yol adı olmalıdır.

### ComponentDataBoyut = *boyut*

Her çağrıda bileşene geçirilen bileşen veri alanının bayt cinsinden boyutu. Bileşen verileri gerekmiyorsa sıfır belirtin.

## Örnek kısmı

```
ServiceComponent:  
  Service=AuthorizationService  
  Name=MQSeries.UNIX.auth.service  
  Module=amqzfu  
  ComponentDataSize=0
```

AuthorizationService kısmı ile ilişkili ServiceComponent kısmı ve NameService kısmı ile ilişkili ServiceComponent kısmı gösteren diğer örnekler için bkz. [“qm.ini dosyasının hizmet kısmı” sayfa 138.](#)

### İlgili kavramlar

[AIX, Linux ve Windows için kurulabilir hizmetler ve bileşenler](#)

### İlgili başvurular

[“qm.ini dosyasının hizmet kısmı” sayfa 138](#)

Hizmet kısmı, kurulabilir hizmetlerde değişiklik yapmak için kullanılır. Bu kıta, hizmetin adını ve hizmet için tanımlanan giriş noktalarının sayısını içerir.

[IBM i üzerinde kurulabilir hizmetler ve bileşenler](#)

[Kurulabilir hizmetler için başvuru bilgileri](#)

## **Windows** qm.ini dosyasının SPX kısmı (yalnızca Windows)

SPX kısmı, SPX protokolü yapılış deęiřtirgelerini belirtir. Bu parametreler, kanallar için varsayılan öznitelikleri geçersiz kılar.

SPX protokolü yapılış deęiřtirgelerini belirtmek için qm.ini kütüğündeki SPX kısmı kullanılır.

**Linux** **Windows** Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer SPX kuyruk yöneticisi özellikleri sayfasını kullanın.

### **Yuva = 5E86 (varsayılan) | socket\_number**

Onaltılı gösterimde SPX yuva numarası. Varsayılan deęer X'5E86' deęeridir.

### **BoardNum = 0 (varsayılan) | baędařtırıcı\_no**

LAN baędařtırıcısı numarası. Varsayılan deęer baędařtırıcı 0 'dır.

### **KeepAlive= NO | YES**

KeepAlive işlevini açın ya da kapatın.

KeepAlive=YES, SPX 'in baęlantısının dięer ucunun hala kullanılabilir olup olmadığını düzenli olarak denetlemesini saęlar. Deęilse, kanal kapalıdır.

### **Library1= DLLName1**

SPX DLL 'nin adı.

Varsayılan deęer WSOCK32.DLL.

### **Library2= DLLName2**

Kod iki ayrı kitaplıkta saklandıysa, LibraryName1 ile aynı.

Varsayılan deęer WSOCK32.DLL.

### **ListenerBacklog= sayı**

SPX dinleyicisi için bekleyen isteklerin varsayılan sayısını geçersiz kılın.

SPX 'te alma sırasında, bekleyen baęlantı isteęi sayısı üst sınırı ayarlanır. Bu, dinleyicinin isteęi kabul etmesi için SPX yuvasında bekleyen isteklerin birikim listesi olarak kabul edilebilir. Varsayılan dinleyici birikim listesi deęerleri [Çizelge 13 sayfa 144](#) içinde gösterilir.

Çizelge 13. Varsayılan bekleyen bağlantı istekleri (SPX)	
Hizmet olarak sunulan	Varsayılan ListenerBacklog değeri
Windows Sunucu	100
Windows İş İstasyonu	5

**Not:** Bazı işletim sistemleri, gösterilen varsayılan değerden daha büyük bir değeri destekler. Bağlantı sınırına ulaşmaktan kaçınmak için bunu kullanın.

Tersi durumda, bazı işletim sistemleri SPX birikim listesi boyutunu sınırlayabilir, bu nedenle etkin SPX birikim listesi burada istenenden daha küçük olabilir.

Birikim Çizelge 13 sayfa 144 içinde gösterilen değerlere ulaşırsa, SPX bağlantısı reddedilir ve kanal başlatılamaz. İleti kanalları için bu, kanalın RETRY durumuna girmesine ve bağlantıyı daha sonra yeniden denemesine neden olur. İstemci bağlantıları için, istemci MQCONN ' dan bir MQRC\_Q\_MGR\_NOT\_KULLANILABİLİR neden kodu alır ve bağlantıyı daha sonra yeniden denemelidir.

### Multi qm.ini dosyasının SSL kısmı

SSL kısmı, bir kuyruk yöneticisinde TLS kanallarını yapılandırmak için kullanılır.

### OCSP (Çevrimiçi Sertifika Durumu Protokolü)

Bir sertifika, AuthorityInfoErişim uzantısını içerebilir. Bu uzantı, OCSP (Online Certificate Status Protocol; Çevrimiçi Sertifika Durumu Protokolü) aracılığıyla iletişim kurulacak bir sunucuyu belirtir. Kuyruk yöneticinizdeki SSL ya da TLS kanallarının AuthorityInfoErişim uzantılarını kullanmasına izin vermek için, bu uzantılarda adı geçen OCSP sunucusunun kullanılabilir olduğundan, doğru yapılandırıldığından ve ağ üzerinden erişilebilir olduğundan emin olun. Daha fazla bilgi için [İptal edilen sertifikalarla çalışmabaşlıkları](#) konuya bakın.

### CrlDistributionNoktası (CDP)

Bir sertifika CrlDistributionPoint uzantısını içerebilir. Bu uzantı, hem bir sertifika iptal listesini (CRL) karşıdan yüklemek için kullanılan iletişim kuralını, hem de iletişim kurulacak sunucuyu tanımlayan bir URL içerir.

Kuyruk yöneticinizdeki SSL ya da TLS kanallarının CrlDistributionPoint uzantılarını kullanmasına izin vermek istiyorsanız, bu uzantılarda adı geçen CDP sunucusunun kullanılabilir, doğru şekilde yapılandırılmış ve ağ üzerinden erişilebilir olduğundan emin olun.

### SSL Stanza

qm.ini dosyasındaki SSL bölümünü kullanarak, kuyruk yöneticinizdeki TLS kanallarının aşağıdaki olanakları kullanmayı nasıl denediğini ve sorun oluştuğunda nasıl tepki verdiklerini yapılandırın.

Aşağıdaki durumların her birinde, sağlanan değer listelenen geçerli değerlerden biri değilse, varsayılan değer alınır. Geçersiz bir değer belirtildiğinden bahsedilirken hata iletisi yazılmaz.

#### V 9.3.0 OutboundSNI = KANAL | ANASİSTEM ADI

**OutboundSNI** CHANNEL olarak ayarlanırsa, SNI yetenekli istemciler, bir TLS bağlantısı başlatırken uzak sisteme SNI ' yi hedef IBM MQ kanal adına ayarlar.

Bu öznitelik HOSTNAME değerine ayarlanırsa, SNI yetenekli istemciler SNI üstbilgisini anasistem adı olarak ayarlayarak giden bağlantı isteklerinin TLS anlaşması sırasında uzak kuyruk yöneticisinin varsayılan sertifikasını almasına ve kanal başına sertifikaların kullanılmasına neden olur.

**Not:** **OutboundSNI=HOSTNAME** , yapılandırılmış bir sertifika etiketiyle uzak bir kanala bağlanmak için kullanılırsa, bağlantı MQRC\_SSL\_INITIALIZATION\_ERROR ile reddedilir ve uzak kuyruk yöneticisi hata günlüklerine bir [AMQ9673](#) iletisi yazdırılır.

### **AllowOutboundSNI = YES (varsayılan) | HAYIR**

Etkinleştirilirse, SNI özellikli istemciler, bir TLS bağlantısı başlatılırken uzak sisteme SNI ' yi hedef IBM MQ kanal adına ayarlar. Bu öznelik NOdeğerine ayarlanırsa, SNI özellikli istemciler, giden bağlantı isteklerinin TLS anlaşması sırasında uzak kuyruk yöneticisinin varsayılan sertifikasını almasına neden olan SNI üstbilgisini ayarlamaz ve bu nedenle kanal başına sertifikalar kullanılamaz.



**Uyarı:** **V 9.3.0** **Deprecated** IBM MQ 9.3.0 ' den **AllowOutboundSNI** özelliği kullanımdan kaldırılmıştır ve yalnızca geriye dönük uyumluluk amacıyla kullanılabilir.

**AllowOutboundSNI** ayarı YES , **OutboundSNI** ayarının CHANNELolarak ayarlandığı aynı işlevi sağlarken **AllowOutboundSNI** ayarı NO , **OutboundSNI** ayarının HOSTNAMEile aynı işlevi sağlar.

SSL kıtasında hem **AllowOutboundSNI** hem de **OutboundSNI** öznelikleri varsa, **OutboundSNI** ayarı önceliklidir.

### **AllowedCipherSpecs =name|name list| ALL**

Çoklu Platformlar üzerinde IBM MQ kanallarıyla kullanılmak üzere sipariş edilen ve etkinleştirilen özel bir CipherSpecs listesini belirtir.

- Tek bir CipherSpec adı.
- Yeniden etkinleştirilecek IBM MQ CipherSpec adlarının virgülle ayrılmış listesi.
- Tüm CipherSpecs ' i gösteren ALLözel değeri (önerilmez).

**Not: ALL** CipherSpecsöğesini seçmemelisiniz; bu, SSL 3.0 ve TLS 1.0 iletişim kurallarını ve çok sayıda zayıf şifreleme algoritmasını etkinleştirir.

Daha fazla bilgi için, TLS el sıkışmasında CipherSpec siparişindeki IBM MQ for Multiplatforms 'da Sipariş edilen ve etkinleştirilen CipherSpecs ' in özel bir listesini sağlamabaşlıklı konuya bakın.

### **IBM i** **ALW** **AllowTLSV13 =Y | YES | T | TRUE (varsayılan) | N | NO | F | FALSE**

Bir kuyruk yöneticisinin TLS 1.3 CipherSpecs' i kullanıp kullanamayacağını belirtir.

- Y (varsayılan), YES (varsayılan), T (varsayılan) ya da TRUE (varsayılan): Kuyruk yöneticisinin TLS 1.3 CipherSpecskullanmasını sağlayan TLS 1.3 ' ü etkinleştirir.
- N, NO, Fya da FALSE: Kuyruk yöneticisinin TLS 1.3 CipherSpecskullanmadığı anlamına gelen TLS 1.3' ü devre dışı bırakır.

Daha fazla bilgi için [CipherSpecs' in etkinleştirilmesibaşlıklı konuya bakın.](#)

### **CDPCheckExtensions= YES |NO (varsayılan)**

Bu kuyruk yöneticisindeki TLS kanallarının, CrIDistributionPoint sertifika uzantılarında adı belirtilen CDP sunucularını denetlemeye çalışıp çalışmadığını belirtir.

- YES(EVET): TLS kanalları, bir dijital sertifikanın iptal edilip edilmediğini belirlemek için CDP sunucularını denetlemeye çalışır.
- NO (varsayılan): TLS kanalları CDP sunucularını denetlemeye çalışmaz. Bu, varsayılan değerdir.

### **ALW** **MinimumRSAKeyBoyut=int**

TLS anlaşması sırasında kabul edilebilmesi için RSA sertifikalarının sahip olması gereken minimum anahtar boyutunu belirtir. 0 ya da üstüne eşit herhangi bir değere izin verir. Belirtilmezse, varsayılan değer olarak 1 kullanılır.

### **OCSPAAuthentication=REQUIRED (varsayılan) | WARN | OPTIONAL**

Bir OCSP sunucusundan iptal durumu saptanamadığında yapılacak işlemi belirtir.

OCSP denetimi etkinleştirildiyse, TLS kanal programı bir OCSP sunucusuyla iletişim kurmayı dener.

Kanal programı herhangi bir OCSP sunucusuyla iletişim kuramıyorsa ya da hiçbir sunucu sertifikanın iptal durumunu sağlayamıyorsa, OCSPAAuthentication parametresinin değeri kullanılır.

- **REQUIRED** (varsayılan): İptal durumunun saptanması, bağlantının bir hatayla kapatılmasına neden olur. Bu, varsayılan değerdir.
- **UYARI**: İptal durumunun saptanamaması, kuyruk yöneticisi hata günlüğüne bir uyarı iletisinin yazılmasına neden oluyor, ancak bağlantının devam etmesine izin veriliyor.
- **İSTEĞE BAĞLI**: İptal durumu belirlenemedi, bağlantının sessiz bir şekilde devam etmesine izin veriyor. Uyarı ya da hata verilmedi.

### **OCSPCheckExtensions = YES (varsayılan) | HAYIR**

Bu kuyruk yöneticisindeki TLS kanallarının AuthorityInfoErişim sertifikası uzantılarında adı belirtilen OCSP sunucularını denetlemeye çalışıp çalışmadığını belirtir.

- **YES** (Evet) (varsayılan): TLS kanalları, bir dijital sertifikanın iptal edilip edilmediğini belirlemek için OCSP sunucularını denetlemeye çalışır. Bu, varsayılan değerdir.
- **NO**: TLS kanalları OCSP sunucularını denetlemeye çalışmaz.

### **ALW OCSPTimeout= sayı**

İptal denetimi gerçekleştirilirken OCSP yanıtlayıcısının bekleneceği saniye sayısı.

IBM MQ 9.3.0 için 0 değeri ayarlanırsa, 30 saniyelik varsayılan zamanaşımı kullanılır.

Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 30 saniye kullanılır.

### **SSLHTTPProxyName= dizgi**

Dizgi, IBM Global Security Kit (GSKit) tarafından OCSP denetimleri için kullanılacak HTTP Yetkili sunucusunun anasistem adı ya da ağ adresidir. Bu adresi parantez içine alınmış isteğe bağlı bir kapı numarası izleyebilir. Kapı numarasını belirtmezseniz, varsayılan HTTP kapısı olan 80 kullanılır.

**AIX** AIX32 bit istemciler için ağ adresi yalnızca IPv4 adresi olabilir.

Diğer platformlarda, ağ adresi bir IPv4 ya da IPv6 adresi olabilir.

Örneğin, bir güvenlik duvarı OCSP yanıtlayıcısının URL adresine erişimi engelliyorsa bu öznelik gerekli olabilir.

### **ALW PeerCertChainValidation=dizgi**

Dizgi iki değerden biri olabilir:

- **Usepeerchain [ Varsayılan]**: Eşdüzey tarafından sağlanan sertifika zinciri, sertifikaları doğrularken güven zinciri boşlukları arasında köprü kurmak için kullanılabilir. Kök sertifika dışında.
- **Truststoreonly [ Önerilmez]**: Eşin sertifikasını doğrulamak için yalnızca güvenilirlik deposundaki sertifikalar kullanılır.

### **ALW SSLHTTPConnectTimeout= sayı|0**

İptal denetimi gerçekleştirilirken bir HTTP sunucusuyla ağ bağlantısının başarıyla kurulması için beklenecek saniye sayısı.

Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 0 (kapalı) kullanılır.

## **Örnek kısmı**

```
SSL:
  OutboundSNI=CHANNEL
  AllowedCipherSpecs=TLS13 CipherSpec list
  AllowTLSV13=Y
  CDPCheckExtensions=NO
  MinimumRSAKeySize=1
  OCSPAAuthentication=REQUIRED
  OCSPCheckExtensions=YES
  OCSPTimeout=30
  PeerCertChainValidation=Usepeerchain
  SSLHTTPConnectTimeout=0
```

## Notlar:

- **OutboundSNI** için varsayılan değer **Channel** değeridir.
- **TLS13 CipherSpec** listesi, diğer ad şifreleri değil, belirli CipherSpecs listesidir. Yalnızca TLS1.3 şifrelerine gereksinim duyarsanız, bunları listelemeniz gerekir. Örneğin:  
TLS\_CHACHA20\_POLY1305\_SHA256  
TLS\_AES\_256\_GCM\_SHA384  
TLS\_AES\_128\_GCM\_SHA256  
TLS\_AES\_128\_CCM\_SHA256  
TLS\_AES\_128\_CCM\_8\_SHA256
- **AllowTLSV13** için varsayılan değer, zayıf şifrelemeleri etkinleştirmedığınız sürece 'Y' dir, bu durumda (açıkça açmadığınız sürece) kapanır.
- **CDPCheckExtensions** değerleri yalnızca Evet ya da Hayır olabilir.
- **PeerCertChainValidation** değerleri yalnızca Usepeerchain ya da Truststoreonly olabilir.

### Multi

## qm.ini dosyasının alt havuz kısmı

Bu kıta IBM MQ tarafından oluşturulur. Bunu değiştirme.

Bir kuyruk yöneticisi yarattığınızda, bu bölüm içindeki Alt Havuz kısmı ve **ShortSubpoolName** özniteliği IBM MQ tarafından otomatik olarak yazılır. IBM MQ, **ShortSubpoolName** için bir değer seçer. Bu değeri değiştirmeyin.

Ad, IBM MQ ' un çalışan işlemleri arasında iç iletişim için kullandığı /var/mqm/sockets dizininde oluşturulan bir dizine ve simgesel bağlantıya karşılık gelir.

### Multi

## qm.ini dosyasının TCP kısmı

TCP kısmı, İletim Denetimi İletişim Kuralı/Internet Protocol (TCP/IP) konfigürasyon parametrelerini belirtir. Bu parametreler, kanallar için varsayılan öznitelikleri geçersiz kılar.

TCP/IP yapılanış değiştirgelerini belirtmek için qm . ini kütüğündeki TCP kütüğünü kullanın.

### Linux

### Windows

Alternatif olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer SPX TCP kuyruk yöneticisi özellikleri sayfasını kullanın.

### Kapı = 1414 (varsayılan) | kapı\_no

TCP/IP oturumları için ondalık gösterimde varsayılan kapı numarası. IBM MQ için *iyi bilinen* kapı numarası 1414 'tür.

### Windows

### Library1= DLLName1 (yalnızca Windows )

TCP/IP yuvaları DLL 'sinin adı.

Varsayılan değer WSOCK32' dir.

### V 9.3.0

### Multi

**SecureCommsYalnızca = NO | N | FALSE | F (varsayılan) | TRUE | T |**

**YES | Y**

Düz metin iletişimine izin verilir verilmeyeceğini belirtin.

### SecureCommsOnly=NO | N | FALSE | F

Düz metin iletişimine izin verilir ve kuyruk yöneticisi başlatıldığında bir uyarı iletisi yazılır.

### SecureCommsOnly=YES | Y | TRUE | T

Düz metin iletişimine izin verilmez ve kuyruk yöneticisi başlatıldığında bir bilgi iletisi yazılır.

### KeepAlive = NO (varsayılan) | XX\_ENCODE\_CASE\_ONE evet




KeepAlive işlevini açın ya da kapatın. KeepAlive=YES, TCP/IP ' nin bağlantının diğer ucunun hala kullanılabilir olup olmadığını düzenli aralıklarla denetlemesini sağlar. Değilse, kanal kapalıdır.



### ListenerBacklog= sayı

TCP/IP dinleyicisine ilişkin bekleyen isteklerin varsayılan sayısını geçersiz kılın.

TCP/IP ' de alma sırasında, işlem bekleyen bağlantı isteği sayısı üst sınırı belirlenir. Bu, dinleyicinin isteği kabul etmesi için TCP/IP kapısında bekleyen isteklerin birikim listesi olarak kabul edilebilir. Varsayılan dinleyici birikim listesi değerleri Çizelge 14 sayfa 148 içinde gösterilir.

Çizelge 14. Varsayılan işlem bekleyen bağlantı istekleri (TCP)	
Hizmet olarak sunulan	Varsayılan ListenerBacklog değeri
 Windows Sunucusu	100
 Linux	100
 AIX V5.3 ya da üstü	100

**Not:** Bazı işletim sistemleri, gösterilen varsayılan değerden daha büyük bir değeri destekler. Bağlantı sınırına ulaşmaktan kaçınmak için bunu kullanın.

Bunun tersine, bazı işletim sistemleri TCP birikim listesi boyutunu sınırlayabilir, bu nedenle etkin TCP birikim listesi burada istenenden daha küçük olabilir.

Birikim listesi Çizelge 14 sayfa 148 içinde gösterilen değerlere ulaşırsa, TCP/IP bağlantısı reddedilir ve kanal başlatılamaz. İleti kanalları için bu, kanalın RETRY durumuna girmesine ve bağlantıyı daha sonra yeniden denemesine neden olur. İstemci bağlantıları için, istemci MQCONN ' dan MQRC\_Q\_MGR\_NOT\_ALLOWED neden kodunu alır ve bağlantıyı daha sonra yeniden dener.

TCP/IP tarafından kullanılan arabelleklerin büyüklüğünü denetlemek için aşağıdaki özellikler grubu kullanılabilir. Değerler, işletim sisteminin TCP/IP katmanına doğrudan geçirilir. Bu özellikleri kullanırken çok dikkatli olunmalıdır. Değerler yanlış ayarlanırsa, TCP/IP başarımını olumsuz etkileyebilir. Bunun başarımı nasıl etkilediğine ilişkin ek bilgi için ortamınıza ilişkin TCP/IP belgelerine bakın. Sıfır değeri, işletim sisteminin IBM MQ tarafından düzeltilen arabellek boyutlarının tersine, arabellek boyutlarını yöneteceğini gösterir.

### Connect\_Timeout = 0 (varsayılan) |sayı


Yuvayı bağlama girişiminin zamanaşımına uğraması için geçen saniye sayısı. Varsayılan değer olan sıfır, bağlantı zamanaşımı olmadığını belirtir.

IBM MQ kanal işlemleri, engelleyici olmayan yuvalar üzerinden bağlanır. Bu nedenle, yuvanın diğer ucu hazır değilse, connect () hemen *EINPROGRESS* ya da *EWOULDBLOCK* ile geri döner. Bundan sonra, bir iletişim hatası bildirildiğinde toplam 20 girişimde yeniden bağlanma girişiminde bulunulacaktır.

Connect\_Timeout değeri sıfır olmayan bir değere ayarlanırsa IBM MQ , yuvanın hazır olması için select () üzerinden belirtilen süreyi bekler. Bu, sonraki bir connect () çağrısının başarı şansını artırır. Bu seçenek, ağ üzerindeki yüksek yük nedeniyle, bağlantıların bekleme süresi gerektirdiği durumlarda yararlı olabilir.

### SndBuffSize = sayı |0 (varsayılan)

Kanalların gönderilmesi tarafından kullanılan TCP/IP gönderme arabelleğinin bayt cinsinden büyüklüğü. Bu kıta değeri, kanal tipine daha özgü bir kıta tarafından geçersiz kılınabilir; örneğin, RcvSndBuffSize. Değer sıfır olarak ayarlanırsa, işletim sistemi varsayılanları kullanılır. Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 32768 kullanılır.

 IBM MQ 8.0' dan, yeni kuyruk yöneticileri otomatik olarak 0 varsayılan ayarıyla oluşturulur (bkz. "Örnek kısmı" sayfa 149).

### RcvBuffSize = sayı |0 (varsayılan)

Alıcı kanallar tarafından kullanılan TCP/IP alma arabelleğinin bayt cinsinden büyüklüğü. Bu kıta değeri, kanal tipine daha özgü bir kıta tarafından geçersiz kılınabilir; örneğin, RcvRcvBuffSize. Değer sıfır olarak ayarlanırsa, işletim sistemi varsayılanları kullanılır. Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 32768 kullanılır.



**Multi** IBM MQ 8.0' dan, yeni kuyruk yöneticileri otomatik olarak 0 varsayılan ayarıyla oluşturulur (bkz. “Örnek kısmı” sayfa 149).

#### **RcvSndBufferSize = sayı |0 (varsayılan)**

Alıcı kanalının gönderen ucu tarafından kullanılan TCP/IP gönderme arabelleğinin bayt cinsinden büyüklüğü. Değer sıfır olarak ayarlanırsa, işletim sistemi varsayılanları kullanılır. Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 32768 kullanılır.

**Multi** IBM MQ 8.0' dan, yeni kuyruk yöneticileri otomatik olarak 0 varsayılan ayarıyla oluşturulur (bkz. “Örnek kısmı” sayfa 149).

#### **RcvRcvBufferSize = number |0 (varsayılan)**

Bir alıcı kanalının alıcı sonu tarafından kullanılan TCP/IP alma arabelleğinin bayt cinsinden büyüklüğü. Değer sıfır olarak ayarlanırsa, işletim sistemi varsayılanları kullanılır. Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 32768 kullanılır.

**Multi** IBM MQ 8.0' dan, yeni kuyruk yöneticileri otomatik olarak 0 varsayılan ayarıyla oluşturulur (bkz. “Örnek kısmı” sayfa 149).

#### **SvrSndBufferSize = sayı |0 (varsayılan)**

Bir istemci bağlantısı sunucu bağlantısı kanalının sunucu sonu tarafından kullanılan TCP/IP gönderme arabelleğinin bayt cinsinden büyüklüğü. Değer sıfır olarak ayarlanırsa, işletim sistemi varsayılanları kullanılır. Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 32768 kullanılır.

**Multi** IBM MQ 8.0' dan, yeni kuyruk yöneticileri otomatik olarak 0 varsayılan ayarıyla oluşturulur (bkz. “Örnek kısmı” sayfa 149).

#### **SvrRcvBufferSize = sayı |0 (varsayılan)**

İstemci-bağlantı sunucusu bağlantı kanalının sunucu sonu tarafından kullanılan TCP/IP alma arabelleğinin bayt cinsinden büyüklüğü. Değer sıfır olarak ayarlanırsa, işletim sistemi varsayılanları kullanılır. Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 32768 kullanılır.

**Multi** IBM MQ 8.0' den yeni kuyruk yöneticileri otomatik olarak 0 varsayılan ayarıyla oluşturulur (bkz. “Örnek kısmı” sayfa 149).

## Örnek kısmı

```
TCP:
SndBufferSize=0
RcvBufferSize=0
RcvSndBufferSize=0
RcvRcvBufferSize=0
ClntSndBufferSize=0
ClntRcvBufferSize=0
SvrSndBufferSize=0
SvrRcvBufferSize=0
```

**Not:** **Multi** Multiplatforms üzerinde yeni kuyruk yöneticileri için, `qm.ini` file TCP ' nin TCP kısmı içindeki varsayılan TCP gönderme ve alma arabelleği büyüklüğü, işletim sistemi tarafından yönetilecek şekilde ayarlanır. Önceki örnekte gösterildiği gibi, yeni kuyruk yöneticileri, gönderme ve alma arabellekleri için 0 varsayılan ayarıyla otomatik olarak yaratılır. Bu yalnızca yeni kuyruk yöneticileri için geçerlidir. IBM MQ ' un önceki sürümlerinden geçirilen kuyruk yöneticilerine ilişkin TCP gönderme ve alma arabelleği ayarları korunur.

TCP arabellek büyüklüğü özellikleri `qm.ini` kütüğünden kaldırılırsa, varsayılan arabellek 32K olarak ayarlanır. Bu varsayılan değeri kullanırken dikkatli olmanız gerekir; 32K , tüm ileti sistemi senaryoları için uygun bir arabellek olmayabilir.

TCP gönderme ve alma arabelleği özellikleri sıfır olarak ayarlanırsa, işletim sistemi varsayılan değerleri kullanılır. Bu varsayılanları seçme yöntemi işletim sistemine göre değişir, ancak genellikle "tcp" ya da `get/setsockopt ()` OS el ile sayfalarında bulunabilir.

TuningParameters kısmı, kuyruk yöneticisinin ayarlanmasına ilişkin seçenekleri belirtir.

#### SuppressDspAuthFail= YES |NO (varsayılan)

YESdeğerine ayarlandığında, kuyruk yöneticisi, bir nesne için + dsp yetkisi yoksa, yetki olaylarının oluşturulmasını ve bir yetki denetimi başarısız olduğunda hata günlüğüne [AMQ8077](#) hata iletilerinin yazılması işlemini önerir.

#### ImplSyncOpenOutput=değer

**ImplSyncOpenOutput** , eşitleme noktasının dışında kalıcı bir koyma için örtük bir eşitleme noktası etkinleştirilmeden önce, kuyruğa konmak üzere açık olan uygulama sayısı alt sınırıdır.

**ImplSyncOpenOutput** varsayılan değeri 2' dir.

Bu, bir koyma işlemi için kuyruğun açık olduğu tek bir uygulama varsa, **ImplSyncOpenOutput** ' in kapatılması gibi bir etkiye sahiptir.

ImplSyncOpenOutput=1 belirtilmesi, örtük bir eşitleme noktasının her zaman dikkate alındığı anlamına gelir. Herhangi bir pozitif tamsayı değeri ayarlayabilirsiniz. Hiçbir zaman örtük bir eşitleme noktasının eklenmesini istemiyorsanız, ImplSyncOpenOutput=OFFdeğerini ayarlayın.

#### UniformClusterAd =kümenin adı

Tek tip küme olarak kullandığınız IBM MQ kümesinin adı.

#### OAMLdapConnectTimeout=time|0 (varsayılan)

LDAP istemcisinin sunucusuyla TCP bağlantısı kurmak için bekleyeceği saniye cinsinden süre üst sınırı. Bir bağlantı ad listesi aracılığıyla birden çok LDAP sunucusu sağladıysanız, zaman aşımı her bir bağlantı girişimi için geçerlidir ve bu nedenle, bu zaman aşımı değerine ulaşıldığında ad en üst sırasındaki bir sonraki girişle bağlantı kurma girişiminde bulunulacaktır.

time (süre) değerinin 3600 saniye üst sınırı vardır ve varsayılan değer yanısıra alt sınır değeri olan 0değeri, bekleme sınırının sınırsız olduğu anlamına gelir.

#### OAMLdapQueryTimeLimit=time|0 (varsayılan)

LDAP istemcisinin, bağlantı kurulduktan ve bir LDAP isteği gönderildikten sonra sunucudan LDAP isteğine yanıt almak için bekleyeceği süre üst sınırı (saniye olarak).

time (süre) değerinin 3600 saniye üst sınırı vardır ve varsayılan değer yanısıra alt sınır değeri olan 0değeri, bekleme sınırının sınırsız olduğu anlamına gelir.

V 9.3.2

V 9.3.0.5

#### OAMLdapResponseWarningTime=eşik

Bir LDAP sunucusuna yönelik bağlantı **OAMLdapResponseWarningTime** parametresi tarafından belirlenen eşik sayısından daha uzun sürerse, hata günlüğüne bir [AMQ5544W](#) iletisi yazılır. Varsayılan eşik değeri 10 saniyedir.

#### ExpiryInterval

Kuyruk yöneticisinin, diğer kuyruk etkinlikleri tarafından henüz temizlenmemiş olan süresi dolmuş iletileri arayan kuyrukları tarama sıklığını belirtir. Saniye cinsinden bir zaman aralığıdır.

Varsayılan olarak, süre bitimi tarayıcısı üretim IBM MQ oluşturmaları üzerinde yaklaşık beş dakikada bir çalışır.



**DİKKAT: ExpiryInterval** değerinin değiştirilmesi genellikle gerekli değildir ve bu değeri yalnızca IBM Destek rehberliğinde değiştirmeniz gerekir.

#### LivenessHeartBeatLen

Kuyruk yöneticisinin günlüğe yazma işlemlerini makul bir hızda denetleme sıklığını yapılandırır.

**LivenessHeartBeatLen** için üst sınır değeri 600 saniyedir (10 dakika) ve alt sınır değeri 0 'dır (bu değer, denetlemeyi tümüyle devre dışı bırakma etkisine sahiptir).



**DİKKAT:** Çoğu durumda, bu denetimlerin sıklığını değiştirmek gerekli değildir. IBM Destek tarafından yapmanız önerilmedikçe herhangi bir değişiklik yapmayın.

#### ECHeartBeatUzn

Kuyruk yöneticisi genel sağlıklı işletim denetimlerinin sıklığını yapılandırır. **ECHeartBeatLen** için alt sınır değeri 10000 milisaniye (10 saniye) ve üst sınır değeri 60000 milisaniyedir (60 saniye).



**DİKKAT:** Çoğu durumda, bu denetimlerin sıklığını değiştirmek gerekli değildir. IBM Destek tarafından yapmanız önerilmedikçe herhangi bir değişiklik yapmayın.

### FileLockHeartBeatLen

Dosya kilidine ilişkin varsayılan değeri, yürütme denetleyicisinin birincil çok eşgörümlü dosyada dışlayıcı kilidi tutmaya devam ettiğinden emin olmak için belirli aralıklarla gerçekleştirdiği çok eşgörümlü bir kuyruk yöneticisi olup olmadığını denetler. Varsayılan olarak, bu dosya kilidi denetimleri her 20 saniyede bir yapılır. **FileLockHeartBeatLen** için değer alt sınırı 10 saniyedir ve değer üst sınırı 600 saniyedir (10 dakika).



**DİKKAT:** Çoğu durumda, bu denetimlerin sıklığını değiştirmek gerekli değildir. IBM Destek tarafından yapmanız önerilmedikçe herhangi bir değişiklik yapmayın.

## Örnek kısmı

V 9.3.2 V 9.3.0.5

```
TuningParameters:  
  SuppressDspAuthFail=NO  
  ImplSyncOpenOutput=2  
  OAMLdapConnectTimeout=60  
  OAMLdapQueryTimeLimit=60  
  OAMLdapResponseWarningTime=10  
  ExpiryInterval=300
```

### İlgili kavramlar

[Örtük eşitleme noktası](#)

Multi

### qm.ini dosyasının değişkenler kısmı

Değişkenler kısmı, otomatik birörnek kümelerle kullanılacak yapılandırma değişkenlerini belirtir.

CONNNAME otomatik küme yapılandırması sırasında Değişkenler kısmında listelenen öznitelikleri ve bir küme alıcı kanalının kanal adı MQSC alanlarını kullanabilirsiniz. Yapılanış değişkenleri bir MQSC komut dosyasının başka bir ögesinde kullanılamaz.

#### öznitelik=değer

MQSC tanımlamaları sırasında araya ekleme olarak kullanılacak bir ad ve ilişkili değeri belirtir.

*attribute=value* çiftleri, bir kuyruk yöneticisi yarattığınızda **crtmqm** komutundaki **-iv** komut satırını seçeneği kullanılarak sağlanabilir.

## Örnek kısmı

```
Variables:  
  CONNNAME=127.0.0.1(1414)
```

### İlgili kavramlar

[“Otomatik uygulama dengeleme” sayfa 403](#)

Otomatik uygulama dengelemesi, IBM MQ tek tip kümesinin rasgele hale getirme ya da uygulamaların belirli kuyruk yöneticilerine el ile sabitlenmesi yerine küme genelinde uygulama dağıtımını yakından yönetmesini sağlayarak uygulama dağıtımını ve kullanılabilirliğini büyük ölçüde geliştirir.

### İlgili görevler

[“Yeni bir birörnek küme oluşturma” sayfa 415](#)

Yeni bir birörnek kümeyi nasıl oluşturduğunuz.

### İlgili başvurular

[“Otomatik küme yapılandırmasının kullanılması” sayfa 419](#)

IBM MQ yapılandırmasını, qm.ini yapılandırma bilgilerini değiştirerek otomatik yapılandırmayı etkinleştirecek şekilde yapılandırın.

## Multi qm.ini dosyasının XAResourceManager kısmı

XAResourceManager kısmı, kuyruk yöneticisi tarafından eşgüdümlü genel iş birimlerinde yer alan kaynak yöneticilerine ilişkin bilgileri belirtir.

qm.ini dosyasındaki XAResourceManager kısmı, kuyruk yöneticisi tarafından eşgüdümlü genel iş birimlerinde yer alan kaynak yöneticilerine ilişkin bilgileri belirtmek için kullanılır.

Linux Windows Diğer bir seçenek olarak, Linux (x86 ve x86-64) ve Windows sistemlerinde IBM MQ Explorer XA kaynak yöneticisi kuyruk yöneticisi özellikleri sayfasını kullanın.

Genel iş birimlerine katılan bir kaynak yöneticisinin her yönetim ortamı için XA kaynak yöneticisi yapıları bilgilerini el ile ekleyin; varsayılan değerler sağlanmaz.

Kaynak yöneticisi öznitelikleriyle ilgili ek bilgi için [Veritabanı eşgüdümü](#) konusuna bakın.

### Ad = ad (zorunlu)

Bu öznitelik, kaynak yöneticisi yönetim ortamını tanıtır.

Name değeri en çok 31 karakter uzunluğunda olabilir. Kaynak yöneticisinin adını, XA anahtarı yapısında tanımlandığı şekilde kullanabilirsiniz. Ancak, aynı kaynak yöneticisinin birden çok yönetim ortamını kullanıyorsanız, her yönetim ortamı için benzersiz bir ad oluşturmanız gerekir. Örneğin, Name dizgisine veritabanının adını ekleyerek benzersizliği güvenceye alabilirsiniz.

IBM MQ , iletilerde ve dspmqtrn komutunun çıkışında Name değerini kullanır.

İlişkili kuyruk yöneticisi başlatıldıktan ve kaynak yöneticisi adı yürürlüğe girdikten sonra, bir kaynak yöneticisi yönetim ortamının adını değiştirmeyin ya da bu yönetim ortamının girişini yapıları bilgilerinden silmeyin.

### SwitchFile= name (zorunlu)

Kaynak yöneticisinin XA anahtar yapısını içeren yükleme dosyasının tam olarak nitelenmiş adı.

32 bit uygulamalarla 64 bit kuyruk yöneticisi kullanıyorsanız, name değeri yalnızca, kaynak yöneticisinin XA anahtar yapısını içeren yükleme dosyasının temel adını içermelidir.

32 bit dosya, ExitsDefaultPathile belirtilen yoldan uygulamaya yüklenecek.

64 bitlik dosya, ExitsDefaultPath64ile belirtilen yoldan kuyruk yöneticisine yüklenecek.

### XAOpenString= dizgi (isteğe bağlı)

Kaynak yöneticisinin xa\_open giriş noktasına geçirilecek veri dizgisi. Dizginin içeriği kaynak yöneticisinin kendisine bağlıdır. Örneğin, dizgi kaynak yöneticisinin bu yönetim ortamının erişeceği veritabanını belirleyebilir. Bu özneliği tanımlama hakkında daha fazla bilgi için bkz:

- [Db2 için kaynak yöneticisi yapıları bilgilerinin eklenmesi](#)
- [Oracle](#)
- [Sybase](#)
- [Informix için kaynak yöneticisi yapıları bilgilerinin eklenmesi](#)

ve ilgili dizgi için kaynak yöneticisi belgelerimize bakın.

### XACloseString= dizgi (isteğe bağlı)

Kaynak yöneticisinin xa\_close giriş noktasına geçirilecek veri dizgisi. Dizginin içeriği kaynak yöneticisinin kendisine bağlıdır. Bu özneliği tanımlama hakkında daha fazla bilgi için bkz:

- [Db2 için kaynak yöneticisi yapıları bilgilerinin eklenmesi](#)
- [Oracle](#)
- [Sybase](#)
- [Informix için kaynak yöneticisi yapıları bilgilerinin eklenmesi](#)

ve uygun dizgi için veritabanı belgelerinize bakın.

## ThreadOfControl=THREAD | PROCESS

**Windows** Bu öznelik Windows için zorunludur. Kuyruk yöneticisi, kaynak yöneticisini kendi çok iş parçacıklı işlemlerinden birinden çağırması gerektiğinde diziselleştirme için bu değeri kullanır.

### Başlık

Kaynak yöneticisi tam olarak *iş parçacığını bilir*. Çok iş parçacıklı bir IBM MQ işleminde, dış kaynak yöneticisine aynı anda birden çok iş parçacığından XA işlev çağrıları yapılabilir.

### Süreç

Kaynak yöneticisi *iş parçacığı güvenli*dir. Çok iş parçacıklı IBM MQ işleminde, kaynak yöneticisine aynı anda yalnızca bir XA işlev çağrısı yapılabilir.

**ThreadOfControl** girişi, çok iş parçacıklı bir uygulama işleminde kuyruk yöneticisi tarafından verilen XA işlev çağrıları için geçerli değildir. Genel olarak, farklı iş parçacıklarında eşzamanlı iş birimleri bulunan bir uygulama, bu işlem kipinin kaynak yöneticilerinin her biri tarafından desteklenmesini gerektirir.

## Örnek kısmı

```
XAResourceManager:  
Name=DB2 Resource Manager Bank  
SwitchFile=/usr/bin/db2swit  
XAOpenString=MQBankDB  
XACloseString=  
ThreadOfControl=THREAD
```

**Not:** XAResourceManager stanzas üst sınırı 255 ile sınırlıdır. Ancak, işlem performansının düşmesini önlemek için yalnızca az sayıda bağımsız yazılım kullanmalısınız.

## IBM i için örnek qm.ini dosyası

IBM için bir kuyruk yöneticisi yapılandırma dosyasında öznelik gruplarının nasıl düzenlenebileceğini gösteren bir örnek.

```
#####  
#* Module Name: qm.ini *#  
#* Type : IBM MQ queue manager configuration file *#  
# Function : Define the configuration of a single queue manager *#  
#* *#  
#####  
#* Notes : *#  
#* 1) This file defines the configuration of the queue manager *#  
#* *#  
#####  
Log:  
LogPath=QMSATURN.Q  
LogReceiverSize=65536  
  
CHANNELS:  
MaxChannels = 20 ; Maximum number of channels allowed.  
; Default is 100.  
MaxActiveChannels = 10 ; Maximum number of channels allowed to be  
; active at any time. The default is the  
; value of MaxChannels.  
  
TCP:  
KeepAlive = Yes ; TCP/IP entries.  
; Switch KeepAlive on.  
SvrSndBuffSize=20000 ; Size in bytes of the TCP/IP send buffer for each  
; channel instance. Default is 32768.  
SvrRcvBuffSize=20000 ; Size in bytes of the TCP/IP receive buffer for each  
; channel instance. Default is 32768.  
Connect_Timeout=10000 ; Number of seconds before an attempt to connect the  
; channel instance times out. Default is zero (no timeout).  
  
QMErrorLog:  
ErrorLogSize = 262144  
ExcludeMessage = 7234
```

```
SuppressMessage = 9001,9002,9202  
SuppressInterval = 30
```

```
TuningParameters:  
  ImplSyncOpenOutput=2
```

## ALW Kuruluş yapılandırma dosyası, mqinst.ini

AIX and Linux sistemlerinde kuruluş yapılandırma dosyası (mqinst.ini), tüm IBM MQ kuruluşlarına ilişkin bilgileri içerir. Windows sistemlerinde kuruluş yapılandırma bilgileri kayıt defterinde bulunur.

### mqinst.ini dosyasının konumu

Linux AIX

mqinst.ini dosyası, AIX and Linux sistemlerinde /etc/opt/mqm dizininde bulunur. Bu belge, varsa, hangi kuruluşun birincil kuruluş olduğuna ve her kuruluşla ilişkin aşağıdaki bilgilere ilişkin bilgileri içerir:

- Kuruluş adı
- Kuruluş açıklaması
- Kuruluş tanıtıcısı
- Kuruluş yolu

**Önemli:** Biçimi düzeltilmediği ve değişebildiği için mqinst.ini dosyası doğrudan düzenlenmemeli ya da dosyaya başvurulmamalıdır.

Kuruluş tanıtıcısı, yalnızca iç kullanım için otomatik olarak ayarlanır ve değiştirilmemelidir.

mqinst.inidosyasını doğrudan düzenlemek yerine, dosyadaki değerleri oluşturmak, silmek, sorgulamak ve değiştirmek için aşağıdaki komutları kullanmanız gerekir:

- Girdi oluşturmak için [crtmqinst](#) .
- Girişleri silmek için [dlmqinst](#) .
- Girişleri görüntülemek için [dspmqinst](#) .
- Girişleri ayarlamak için [setmqinst](#) .

## Windows üzerinde kuruluş yapılandırma bilgileri

Windows

Windows üzerinde mqinst.ini dosyası yok. Kuruluş yapılandırma bilgileri kayıt defterinde bulunur ve aşağıdaki anahtarla tutulur:

```
HKLM\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName
```

**Önemli:** Biçimi düzeltilmediği ve değişebildiği için bu anahtar doğrudan düzenlenmemeli ya da bu anahtara başvurulmamalıdır.

Bunun yerine, kayıttaki değerleri sorgulamak ve değiştirmek için aşağıdaki komutları kullanmanız gerekir:

- Girişleri görüntülemek için [dspmqinst](#) .
- Girişleri ayarlamak için [setmqinst](#) .

Windows işletim sistemlerinde **crtmqinst** ve **dlmqinst** komutları kullanılamaz. Kuruluş ve kaldırma işlemleri, gerekli kayıt girişlerinin yaratılmasını ve silinmesini işler.

## Multi IBM MQ MQI client yapılandırma dosyası, mqclient.ini

Bir metin dosyasındaki öznitelikleri kullanarak istemcilerinizi yapılandırabilirsiniz. Bu öznitelikler, ortam değişkenleri ya da platforma özgü başka yollarla geçersiz kılınabilir.

IBM MQ MQI clients ' i, qm . inikuyruk yöneticisi yapılanış kütüğüne benzer bir metin kütüğü kullanarak yapılandırabilirsiniz. Dosya, her biri **attribute-name** = *değer* biçiminde bir dizi satır içeren bir dizi bağımsız parça içerir.

IBM MQ MQI client yapılanış kütüğü genellikle mqclient . ini olarak adlandırılır, ancak başka bir ad vermeyi seçebilirsiniz. Bu dosyadaki yapılandırma bilgileri aşağıdaki platformlar için geçerlidir:

- **ALW** AIX, Linux, and Windows
- **IBM i** IBM i

**Not:** IBM i' da varsayılan mqclient . ini dosyası yoktur. Ancak, dosyayı IBM i Integrated File System ' nde (IFS) oluşturabilirsiniz.

Daha fazla bilgi için bkz “İstemci yapılanış kütüğünün yeri” sayfa 156.

**Not:** **z/OS** z/OS platformu, IBM MQ istemcilerini çalıştırmak için kullanılamaz. Bu nedenle, mqclient . ini dosyası IBM MQ for z/OS üzerinde yoktur.

IBM MQ MQI client yapılandırma dosyasındaki öznitelikler, aşağıdakileri kullanan istemciler için geçerlidir:

- MQI
- IBM MQ classes for Java
- IBM MQ classes for JMS
- IBM MQ classes for .NET
- XMS

IBM MQ MQI client yapılandırma dosyasındaki öznitelikler çoğu IBM MQ istemcisi için geçerli olsa da, yönetilen .NET ve XMS .NET istemcileri tarafından ya da IBM MQ classes for Java ya da IBM MQ classes for JMS kullanan istemciler tarafından okunmayan bazı öznitelikler vardır. Daha fazla bilgi için bkz “Hangi IBM MQ istemcileri her özniteliği okuyabilir” sayfa 157.

Yapılanış özellikleri, bir istemci uygulamasının bir kuyruk yöneticisine yönelik tek bir bağlantıya özgü olmak yerine, herhangi bir kuyruk yöneticisine yaptığı tüm bağlantılar için geçerlidir. Tek bir kuyruk yöneticisine yönelik bağlantıyla ilgili öznitelikler, örneğin bir MQCD yapısı kullanılarak ya da bir İstemci Kanal Tanımlama Çizelgesi (CCDT) kullanılarak programlı olarak yapılandırılabilir.

Aşağıda bir istemci yapılandırma dosyası örneği verilmiştir:

```
## Module Name: mqclient.ini                ##
## Type       : IBM MQ MQI client configuration file    ##
## Function   : Define the configuration of a client    ##
##          ##
##*****##
## Notes     :                ##
## 1) This file defines the configuration of a client  ##
##          ##
##*****##

ClientExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

TCP:
  Library1=DLLName1
  KeepAlive = Yes
  ClntSndBuffSize=32768
  ClntRcvBuffSize=32768
  Connect_Timeout=0

MessageBuffer:
  MaximumSize=-1
  Updatepercentage=-1
  PurgeTime=0

LU62:
  TPName
  Library1=DLLName1
  Library2=DLLName2
```

```
PreConnect:
  Module=myMod
  Function=myFunc
  Data=ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
  Sequence=1

CHANNELS:
  DefRecon=YES
  ServerConnectionParms=SALES.SVRCONN/TCP/hostname.x.com(1414)

Connection:
  ApplName=ExampleApplName
```

İstemci yapılandırma dosyasını kullanarak birden çok kanal bağlantısı kuramazsınız.

IBM WebSphere MQ 7.0 ' den önceki yayınlarda desteklenen ortam değişkenleri sonraki yayınlarda desteklenmeye devam eder ve böyle bir ortam değişkeni istemci yapılandırma dosyasındaki eşdeğer bir değerle eşleştiginde, ortam değişkeni istemci yapılandırma dosyası değerini geçersiz kılar.

IBM MQ classes for JMSkullanan bir istemci uygulaması için, istemci yapılandırma dosyasını aşağıdaki yollarla geçersiz kılabilirsiniz:

- JMS yapılandırma dosyasındaki özellikleri ayarlayarak.
- JMS yapılandırma dosyasını geçersiz kılan Java sistem özelliklerini ayarlayarak.

.NET istemcisi için, .NET uygulama yapılandırma dosyasını kullanarak istemci yapılandırma dosyasını ve eşdeğer ortam değişkenlerini geçersiz kılabilirsiniz.

## Yapılandırma dosyasındaki yorumlar

Linux AIX

Yapılandırma dosyası içindeki bir açıklamanın başlangıcını işaretlemek için noktalı virgül ';' ve hash '#' karakterini kullanabilirsiniz. Bu, bir satırın tamamını açıklama olarak işaretleyebilir ya da bir ayarın değerine dahil edilmeyecek bir satırın sonundaki bir yorumu belirtebilir.

Bir değer bu karakterlerden birini gerektiriyorsa, '\' ters eğik çizgi karakterini kullanarak bu karaktere çıkış karakteri oluşturmanız gerekir.

Aşağıdaki örnek, yapılandırma dosyası içindeki açıklamaların kullanımını göstermektedir:

```
# Example of an SSL stanza with comments
SSL:
  ClientRevocationChecks=REQUIRED ; Example of an end of line comment
  SSLCryptoHardware=GSK_PKCS11=/driver\;label\;password\;SYMMETRIC_CIPHER_ON # Example of
  escaped comment characters.
```

## İlgili kavramlar

[Java yapılandırma dosyası için IBM MQ sınıfları](#)

## Multi İstemci yapılandırma dosyasının yeri

Bir IBM MQ MQI client yapılandırma dosyası çeşitli konumlarda tutulabilir.

Bir istemci uygulaması, IBM MQ MQI client yapılandırma dosyasını bulmak için aşağıdaki arama yolunu kullanır:

1. **MQCLNTCF** ortam değişkeniyle belirtilen konum.

Bu ortam değişkeninin biçimi tam bir URL' dir. Bu, dosya adının mqclient.ini olması gerekmeyeceği ve dosyanın ağa bağlı bir dosya sistemine yerleştirilmesini kolaylaştırdığı anlamına gelir.

### Notlar:

- C, .NET ve XMS istemcileri yalnızca file: iletişim kuralını destekler; URL dizgisi protocol: ile başlamazsa file: iletişim kuralı kabul edilir.



- Ortam değişkenlerinin okunmasını desteklemeyen Java 1.4.2 JRE 'lerine izin vermek için, **MQCLNTCF** ortam değişkeni bir **MQCLNTCF** Java Sistem Özelliği ile geçersiz kılınabilir.
2. Uygulamanın mevcut çalışma dizininde `mqclient.ini` adlı bir dosya.
  3. AIX, Linux, and Windows sistemleri için IBM MQ veri dizininde `mqclient.ini` adlı bir dosya.

#### Notlar:

- IBM MQ veri dizini, IBM i ve z/OS gibi belirli platformlarda ya da istemcinin başka bir ürünle birlikte sağlandığı durumlarda bulunmaz.

**IBM i** IBM i' de varsayılan `mqclient.ini` dosyası yoktur. Ancak dosya, /QIBM/UserData/mqm/dizindeki IBM i Integrated File System (IFS) ve **MQCLNTCF** ortam değişkeninde onu gösterecek şekilde tanımlanabilir. Örneğin:

```
ADDENVVAR ENVVAR(MQCLNTCF) VALUE('QIBM/UserData/mqm/mqclient.ini') REPLACE(*YES)
```

Ortam değişkenlerine ilişkin daha fazla örnek için bkz. [“Ortam değişkenleri açıklamaları”](#) sayfa 63.

**z/OS** z/OS platformu, IBM MQ istemcilerini çalıştırmak için kullanılamaz. Bu nedenle, `mqclient.ini` dosyası IBM MQ for z/OS üzerinde yoktur.

- **Linux** **AIX** AIX and Linux sistemlerinde izin şudur: `/var/mqm`.
  - **Windows** Windows platformlarında, kuruluş sırasında **MQ\_DATA\_PATH** ortam değişkenini veri dizinini gösterecek şekilde yapılandırırınız. Olağan durumda `C:\ProgramData\IBM\MQ`.
- Not:** Yalnızca bir istemci kuruyorsanız, ortam değişkeni **MQ\_FILE\_PATH** olabilir.
- Ortam değişkenlerinin okunmasını desteklemeyen Java 1.4.2 JRE 'lerine izin vermek için **MQ\_DATA\_PATH** ortam değişkenini bir **MQ\_DATA\_PATH** Java Sistem Özelliği ile el ile geçersiz kılabilirsiniz.

4. Platforma uygun ve kullanıcıların erişebileceği standart bir dizinde `mqclient.ini` adlı bir dosya:

- Tüm Java istemcileri için bu, `user.home` Java Sistem Özelliği değeridir.
- **Linux** **AIX** AIX and Linux platformlarındaki C istemcileri için bu, **HOME** ortam değişkeninin değeridir.
- **Windows** Windows üzerindeki C istemcileri için bu, **HOMEDRIVE** ve **HOMEPATH** ortam değişkenlerinin birleştirilmiş değerleridir.

### **Multi** Hangi IBM MQ istemcileri her özneliği okuyabilir

IBM MQ MQI client yapılandırma dosyasındaki özneliklerin çoğu C istemcisi ve yönetilmeyen .NET istemcileri tarafından kullanılabilir. Ancak, yönetilen .NET ve XMS .NET istemcileri ya da IBM MQ classes for Java ya da IBM MQ classes for JMS kullanan istemciler tarafından okunmayan bazı öznelikler vardır.

Çizelge 15. Her istemci tipi için geçerli olan öznelikler

<b>mqclient.ini kısmı adı ve öznelikleri</b>	<b>Açıklama</b>	<b>C ve yönetilmeyen .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Yönetilen .NET</b>	<b>Yönetilen XMS .NET</b>
<b>KANAL Kısıtı</b>						
<u>CCSID</u>	Kullanılacak kodlanmış karakter takımı numarası.	Evet	Hayır	Hayır	Evet	Evet

Çizelge 15. Her istemci tipi için geçerli olan öznitelikler (devamı var)

<b>mqclient.in</b> <b>ni kısmı adı</b> <b>ve</b> <b>öznitelikleri</b>	<b>Açıklama</b>	<b>C ve</b> <b>yönetilmeye</b> <b>n .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Yönetilen .N</b> <b>ET</b>	<b>Yönetilen</b> <b>XMS .NET</b>
<u>ChannelDefinitionDizini</u>	İstemci kanal tanımlama çizelgesini içeren dosyanın dizin yolu.	Evet	Hayır	Hayır	Evet	Evet
<u>ChannelDefinitionDosyası</u>	İstemci kanal tanımlama çizelgesini içeren dosyanın adı.	Evet	Hayır	Hayır	Evet	Evet
<u>ReconDelay</u>	Otomatik olarak yeniden bağlanabilen istemci programları için yeniden bağlanma gecikmesi yapılandırma üzere bir yönetim seçeneği.	Evet	Hayır	Evet	Evet	Evet
<u>DefRecon</u>	İstemci programlarının otomatik olarak yeniden bağlanmasını sağlamak ya da otomatik olarak yeniden bağlanmak üzere yazılan bir istemci programının otomatik yeniden bağlanmasını geçersiz kılmak için bir denetim seçeneği.	Evet	Hayır	Evet	Evet	Evet

Çizelge 15. Her istemci tipi için geçerli olan öznelikler (devamı var)

<b>mqclient.in</b> <b>ni kısmı adı</b> <b>ve</b> <b>öznelikleri</b>	<b>Açıklama</b>	<b>C ve</b> <b>yönetilmeye</b> <b>n .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Yönetilen .N</b> <b>ET</b>	<b>Yönetilen</b> <b>XMS .NET</b>
<a href="#">MQReconnectTimeout</a>	Bir istemciye yeniden bağlanmak için saniye cinsinden zaman aşımı.	Evet	Hayır	Hayır	Evet	Hayır
<a href="#">ServerConnectionParms</a>	Kullanılacak IBM MQ sunucusunun ve iletişim yönteminin konumu.	Evet	Hayır	Hayır	Evet	Evet
<a href="#">Put1DefaultAlwaysSync</a>	MQPMO_RESPONSE_AS_Q_DEF seçeneğiyle MQPUT1 işlev çağrısının davranışını denetler.	Evet	Evet	Evet	Evet	Evet
<a href="#">PasswordProtection</a>	SSL ya da TLS kullanmak yerine MQCSP yapısında korumalı parolalar ayarlamayı sağlar.	Evet	Evet	Evet	Evet	Evet
<b>ClientExitYol kısmı</b>						
<a href="#">ExitsDefaultYolu</a>	İstemciler için 32 bit kanal çıkışlarının konumunu belirtir.	Evet	Evet	Evet	Evet	Evet
<a href="#">ExitsDefaultPath64</a>	İstemciler için 64 bitlik kanal çıkışlarının konumunu belirtir.	Evet	Evet	Evet	Evet	Evet

Çizelge 15. Her istemci tipi için geçerli olan öznelikler (devamı var)

<b>mqclient.ini</b> kısmı adı ve öznelikleri	<b>Açıklama</b>	<b>C ve yönetilmeyen .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Yönetilen .NET</b>	<b>Yönetilen XMS .NET</b>
<a href="#">JavaExitsClassPath</a>	Bir Java çıkışı çalıştırıldığında sınıf yoluna eklenecek değerler.	Hayır	Evet	Evet	Hayır	Hayır
<b>Bağlantı kısmı</b>						
<a href="#">AppName</a>	İstemci yapılandırma dosyasında belirtilen uygulama adı.	Evet	Hayır	Hayır	Hayır	Hayır
<b>JMQI kısmı</b>						
<a href="#">useMQCSPAuthentication</a>	IBM MQ classes for Java ve IBM MQ classes for JMS uygulamalarının bir kuyruk yöneticisiyle kimlik doğrulaması yaparken Uyumluluk kipini mi, yoksa MQCSP kimlik doğrulama kipini mi kullanacağını denetler.	Hayır	Evet	Evet	Hayır	Hayır
<b>MessageBuffer kısmı</b>						
<a href="#">MaximumSize</a>	Önden okuma arabelleğinin büyüklüğü (kilobayt cinsinden), 1-999 999 aralığındadır.	Evet	Evet	Evet	Evet	Evet




Çizelge 15. Her istemci tipi için geçerli olan öznelikler (devamı var)

<b>mqclient.ini</b> kısmı adı ve öznelikleri	<b>Açıklama</b>	<b>C ve yönetilmeyen .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Yönetilen .NET</b>	<b>Yönetilen XMS .NET</b>
<u>PurgeTime</u>	Önden okuma arabelleğinde bırakılan iletilerin temizlenmesi için saniye cinsinden aralık.	Evet	Evet	Evet	Evet	Evet
<u>UpdatePercentage</u>	1-100 aralığındaki güncelleme yüzdesi değeri, bir istemci uygulamasının sunucuya ne zaman yeni bir istekte bulunacağını belirlemek için eşik değerinin hesaplanmasında kullanılır.	Evet	Evet	Evet	Evet	Evet
<b>PreConnect kısmı</b>						
<u>Veriler</u>	Bağlantı tanımlamalarının saklandığı havuzun URL'si.	Evet	Hayır	Hayır	Hayır	Hayır
<u>function</u>	PreConnect çıkış kodunu içeren kitaplıktaki işlevsel giriş noktasının adı.	Evet	Hayır	Hayır	Hayır	Hayır
<u>Modül</u>	API çıkış kodunu içeren modülün adı.	Evet	Hayır	Hayır	Hayır	Hayır

Çizelge 15. Her istemci tipi için geçerli olan öznelikler (devamı var)

mqclient.in i kısmı adı ve öznelikleri	Açıklama	C ve yönetilmeye n .NET	Java	JMS	Yönetilen .N ET	Yönetilen XMS .NET
<u>Sequence</u>	Bu çıkışın diğer çıkışlarla görelili olarak çağrıldığı sıra.	Evet	Hayır	Hayır	Hayır	Hayır
<b>Güvenlik kısmı</b>						
<u>DisableClientAMS</u>	Bir kuyruk yöneticisine istemci bağlantıları için AMS ' u devre dışı bırakır ya da etkinleştirir.	Evet	Evet	Evet	Hayır	Hayır
<b>SSL kısmı</b>						
<b>V9.3.0</b> <u>OutboundSNI</u>	SNI özellikli istemcilerin, bir TLS bağlantısı başlatılırken uzak sisteme SNI ' yi hedef IBM MQ kanal adına mı, yoksa anasistem adına mı ayarlayacağını belirler.	Evet	Evet	Evet	Evet	Hayır

Çizelge 15. Her istemci tipi için geçerli olan öznelikler (devamı var)


mqclient.i ni kısmı adı ve öznelikleri	Açıklama	C ve yönetilmeye n .NET	Java	JMS	Yönetilen .N ET	Yönetilen XMS .NET
<a href="#">AllowOutboundSNI</a>	<p>SNI özellikli istemcilerin, bir TLS bağlantısı başlatılırken uzak sisteme SNI 'yi hedef IBM MQ kanal adına ayarlayıp ayarlamayacağı belirler.</p> <p> <b>Uyarı:</b>                          IBM MQ 9.3.0 'dan bu özellik kullanımdan kaldırılmıştır. Bunun yerine <b>OutboundSNI</b> kullanın.</p>	Evet	Evet	Evet	Hayır	Hayır
<a href="#">AllowTLSV13</a>	Bir kuyruk yöneticisinin TLS 1.3 CipherSpecs' i kullanıp kullanamaya çağrısı.	Evet (C/C++ istemcileri)	Hayır	Hayır	Hayır	Hayır
<a href="#">CDPCheckExtensions</a>	Bu kuyruk yöneticisindeki SSL ya da TLS kanallarının, CrIDistributionPoint sertifikası uzantılarında adı belirtilen CDP sunucularını denetlemeye çalışıp çalışmadığını belirtir.	Evet	Hayır	Hayır	Hayır	Hayır

Çizelge 15. Her istemci tipi için geçerli olan öznelikler (devamı var)

<b>mqclient.inin kısmı adı ve öznelikleri</b>	<b>Açıklama</b>	<b>C ve yönetilmeyen .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Yönetilen .NET</b>	<b>Yönetilen XMS .NET</b>
<a href="#">CertificateLabel</a>	Kanal tanımlamasının sertifika etiketi.	Evet	Hayır	Hayır	Hayır	Hayır
<a href="#">CertificateValidation</a>	Kullanılan sertifika geçerlilik denetiminin tipini belirler.	Evet	Hayır	Hayır	Hayır	Hayır
<a href="#">ClientRevocationDenetimi</a>	İstemci bağlantı çağrısı bir SSL/TLS kanalı kullanıyorsa, sertifika iptal denetiminin nasıl yapılandırılacağını belirler.	Evet	Hayır	Hayır	Hayır	Hayır
<a href="#">EncryptionPolicySuiteB</a>	Bir kanalın Suite-B uyumlu şifreleme kullanıp kullanmadığını ve hangi güç düzeyinin kullanılacağını belirler.	Evet	Hayır	Hayır	Hayır	Hayır
<b>V9.3.0</b> <a href="#">Environment Scope</a>	IBM MQ 'in tüm süreç için tek bir IBM Global Security Kit (GSKit) ortamı mı, yoksa her bağlantı için bir GSKit ortamı mı kullanacağını denetler.	Evet (C istemcileri)	Hayır	Hayır	Hayır	Hayır





Çizelge 15. Her istemci tipi için geçerli olan öznelikler (devamı var)

<b>mqcClient.ini kısmı adı ve öznelikleri</b>	<b>Açıklama</b>	<b>C ve yönetilmeyen .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Yönetilen .NET</b>	<b>Yönetilen XMS .NET</b>
<a href="#">MinimumRSAKeyBoyutu</a>	RSA sertifikalarının kabul edilebilmesi için sahip olması gereken anahtar büyüklüğü alt sınırını belirler.	Evet (C/C++ istemcileri)	Hayır	Hayır	Hayır	Hayır
<a href="#">OCSPAuthentication</a>	OCSP etkinleştirildiğinde ve OCSP iptal denetiminin sertifika iptal durumunu saptayamadığında IBM MQ davranışını tanımlar.	Evet	Hayır	Hayır	Hayır	Hayır
<a href="#">OCSPCheckExtensions</a>	IBM MQ 'un AuthorityInfo Erişim sertifikası uzantılarında işlem yapıp yapılmayacağını denetler.	Evet	Hayır	Hayır	Hayır	Hayır
<a href="#">OCSPTimeout</a>	İptal denetimi gerçekleştirilirken OCSP yanıtlayıcısının bekleneceği saniye sayısı.	Evet	Hayır	Hayır	Hayır	Hayır
 <a href="#">PeerCertChainValidation</a>	GSKit sertifika geçerlilik denetimi ayarı.	Evet	Hayır	Hayır	Hayır	Hayır


Çizelge 15. Her istemci tipi için geçerli olan öznitelikler (devamı var)

<b>mqclient.in</b> <b>ni kısmı adı</b> <b>ve</b> <b>öznitelikleri</b>	<b>Açıklama</b>	<b>C ve</b> <b>yönetilmeye</b> <b>n .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Yönetilen .N</b> <b>ET</b>	<b>Yönetilen</b> <b>XMS .NET</b>
<a href="#">SSLCryptoHardware</a>	Sistemde bulunan PKCS #11 şifreleme donanımını yapılandırma k için gereken parametre dizgisini belirler.	Evet	Hayır	Hayır	Hayır	Hayır
<a href="#">SSLCryptoHardwareKeyFile</a>	<b>SSLCryptoHardware</b> özniteliğiyle belirtilen PKCS #11 şifreleme donanımı yapılandırma dizgisinde parolayı şifrelemek için kullanılan ilk anahtarı içeren dosyanın tam yolunu ve adını belirtir.	Evet	Hayır	Hayır	Hayır	Hayır
<a href="#">SSLFipsRequired</a>	IBM MQ içinde şifreleme gerçekleştirilirse yalnızca FIPS onaylı algoritmaların kullanılıp kullanılmaya çağını belirtir.	Evet	Hayır	Hayır	Hayır	Hayır

Çizelge 15. Her istemci tipi için geçerli olan öznelikler (devamı var)

<b>mqclient.ini kısmı adı ve öznelikleri</b>	<b>Açıklama</b>	<b>C ve yönetilmeyen .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Yönetilen .NET</b>	<b>Yönetilen XMS .NET</b>
<a href="#">SSLHTTPProxyName</a>	Dizgi, GSKit tarafından OCSP denetimleri için kullanılacak HTTP Yetkili sunucusunun adresi ya da ağ adresidir.	Evet	Hayır	Hayır	Hayır	Hayır
<a href="#">SSLHTTPConnectTimeout</a>	İptal denetimi gerçekleştirilirken bir HTTP sunucusuyla ağ bağlantısının başarıyla kurulması için beklenecek saniye sayısı.	Evet	Hayır	Hayır	Hayır	Hayır
<a href="#">SSLKeyRepository</a>	Kullanıcının sayısal sertifikasını kök biçiminde tutan anahtar havuzunun konumu.	Evet	Hayır	Hayır	Hayır	Hayır
  <a href="#">SSLKeyRepositoryParola</a>	Anahtar havuzuna erişmek için geçiş tımcısı.	Evet	Hayır	Hayır	Hayır	Hayır
<a href="#">SSLKeyReset Sayı</a>	Gizli anahtar yeniden anlaşılmadan önce bir SSL ya da TLS kanalında gönderilen ve alınan şifrelenmemiş bayt sayısı.	Evet	Hayır	Hayır	Hayır	Hayır
<b>TCP kısmı</b>						

Çizelge 15. Her istemci tipi için geçerli olan öznelikler (devamı var)

<b>mqclient.in</b> <b>ni kısmı adı</b> <b>ve</b> <b>öznelikleri</b>	<b>Açıklama</b>	<b>C ve</b> <b>yönetilmeye</b> <b>n .NET</b>	<b>Java</b>	<b>JMS</b>	<b>Yönetilen .N</b> <b>ET</b>	<b>Yönetilen</b> <b>XMS .NET</b>
<a href="#">ClntRcvBuffSize</a>	İstemci-bağlantı sunucusu bağlantı kanalının istemci sonu tarafından kullanılan TCP/IP alma arabelleğinin bayt cinsinden büyüklüğü.	Evet	Evet	Evet	Evet	Evet
<a href="#">ClntSndBuffSize</a>	İstemci-bağlantı sunucusu bağlantı kanalının istemci sonu tarafından kullanılan TCP/IP gönderme arabelleğinin bayt cinsinden büyüklüğü.	Evet	Evet	Evet	Evet	Evet
<a href="#">Connect_Timeout</a>	Yuvayı bağlama girişiminin zamanaşımına uğraması için geçen saniye sayısı.	Evet	Evet	Evet	Hayır	Hayır
<a href="#">IPAddressVersion</a>	Kanal bağlantısı için hangi IP protokolünün kullanılacağını belirler.	Evet	Hayır	Hayır	Evet	Evet
<a href="#">KeepAlive</a>	KeepAlive işlevini açar ya da kapatır.	Evet	Evet	Evet	Evet	Evet
 <a href="#">Library1</a>	Yalnızca Windows 'de, TCP/IP yuvaları DLL'inin adı.	Evet	Hayır	Hayır	Hayır	Hayır

Çizelge 15. Her istemci tipi için geçerli olan öznelikler (devamı var)

mqclient.ini kısmı adı ve öznelikleri	Açıklama	C ve yönetilmeyen .NET	Java	JMS	Yönetilen .NET	Yönetilen XMS .NET
<b>İzleme kısmı</b>						
<b>Not:</b> İzleme kısmı yalnızca IBM MQ .NET ve XMS .NET istemcileri için geçerlidir.						
<a href="#">V9.3.3</a> MQDotnetTraceDüzy	IBM MQ .NET izlemesini etkinleştirmek için kullanılır.	Hayır	Hayır	Hayır	Evet	Hayır
<a href="#">V9.3.3</a> MQDotnetTraceYolu	IBM MQ .NET izleme dosyalarının yaratılacağı bir klasörü gösterir.	Hayır	Hayır	Hayır	Evet	Hayır
<a href="#">V9.3.3</a> MQDotnetErrorYol	IBM MQ .NET izleme için hata günlüğü dosyalarının oluşturulacağı bir klasörü gösterir.	Hayır	Hayır	Hayır	Evet	Hayır
<a href="#">V9.3.3</a> XMSDotnetTraceDüzy	XMS .NET izlemesini etkinleştirmek için kullanılır.	Hayır	Hayır	Hayır	Hayır	Evet
<a href="#">V9.3.3</a> XMSDotnetTraceFilePath	XMS .NET izleme dosyalarının yaratılacağı bir klasörü gösterir.	Hayır	Hayır	Hayır	Hayır	Evet
<a href="#">V9.3.3</a> XMSDotnetTraceSpecification	XMS .NET için izlemek istediğiniz sınıfın adını belirtir.	Hayır	Hayır	Hayır	Hayır	Evet
<a href="#">V9.3.3</a> XMSDotnetTraceSpecification	XMS .NET için oluşturulması gereken izleme dosyası boyutu üst sınırını belirtir.	Hayır	Hayır	Hayır	Hayır	Evet

Çizelge 15. Her istemci tipi için geçerli olan öznitelikler (devamı var)

mqclient.inik kısmı adı ve öznitelikleri	Açıklama	C ve yönetilmeyen .NET	Java	JMS	Yönetilen .NET	Yönetilen XMS .NET
<a href="#">V9.3.3</a> <a href="#">XMSDotnetTraceFileSize</a>	XMS .NET için alıkonacak izleme dosyalarının sayısı.	Hayır	Hayır	Hayır	Hayır	Evet

### [V9.3.0](#) İstemci yapılış kütüğünün Application kısmı

Bu yapılandırmayı kullanarak bağlantı kuran belirli bir uygulama için tek tip küme dengeleme davranışını etkileyen öznitelikleri belirtmek için Application kısmı kullanın. Bu kütadaki değerler ApplicationDefaults kıstasından önceliklidir, ancak bir program aracılığıyla sağlanan bir MQBNO yapısı tarafından geçersiz kılınabilir.

**Not:** Bu kütta ögesinin her özneliğinin açıklaması, hangi IBM MQ istemcilerinin bu özneliği okuyabileceğini gösterir. Tüm IBM MQ MQI client yapılış kütüğü stanzas 'larına ilişkin özet çizelge için [Her istemci tarafından okunabilen IBM MQ öznitelikler](#) başlıklı konuya bakın.

Uygulama kısmı aşağıdaki öznitelikleri içerir:

#### **Ad = ApplicationName**

Seçeneklerin hangi uygulama adına uygulanacağını tanımlar.

#### **Tip = Basit,ReqRep**

IBM MQ Bu uygulamanın katıldığı IBM MQ etkinliğinin genel örüntüsünü gösterir.

#### **BalanceTimeout = Never,Immediate,0-999999999, Default**

Yeniden dengelemeye izin vermek için uygulama etkinliği kesilmeden önce IBM MQ ' e zamanaşımını belirtir; varsayılan değeri 10 saniye olan, hiçbir zaman ya da en çok 999.999.999 saniye olan bir değer.

#### **BalanceOptions = Yok,IgnTrans**

Dengeleme seçeneği belirlenmez ya da bir işlemde yer alan uygulamaların hemen kesilmesine izin verir.

### [V9.3.0](#) [Multi](#) İstemci yapılış kütüğünün ApplicationDefaults kısmı

Bu yapılandırmayı kullanarak bağlanan istemci uygulamaları için varsayılan tekdüze küme dengeleme davranışını etkileyen öznitelikleri belirtmek için ApplicationDefaults kısmı kullanın. Bu varsayılan değerler, uygulamaya özgü bir Application kısmı ya da bir program aracılığıyla sağlanan bir MQBNO yapısı tarafından geçersiz kılınabilir.

**Not:** Bu kütta ögesinin her özneliğinin açıklaması, hangi IBM MQ istemcilerinin bu özneliği okuyabileceğini gösterir. Tüm IBM MQ MQI client yapılış kütüğü stanzas 'larına ilişkin özet çizelge için [Her istemci tarafından okunabilen IBM MQ öznitelikler](#) başlıklı konuya bakın.

Aşağıdaki öznitelikler ApplicationDefaults kısmına eklenebilir:

#### **Tip = Basit,ReqRep**

IBM MQ Bu uygulamanın katıldığı IBM MQ etkinliğinin genel örüntüsünü gösterir.

#### **BalanceTimeout = Never,Immediate,0-999999999, Default**

Yeniden dengelemeye izin vermek için uygulama etkinliği kesilmeden önce IBM MQ ' e zamanaşımını belirtir; varsayılan değeri 10 saniye olan, hiçbir zaman ya da en çok 999.999.999 saniye olan bir değer.

## BalanceOptions = Yok,IgnTrans

Dengeleme seçeneği belirlenmez ya da bir işlemde yer alan uygulamaların hemen kesilmesine izin verir.

## Multi İstemci yapılanış kütüğünün kanal kısmı

İstemci kanallarına ilişkin bilgileri belirtmek için Kanallar kısmına bakın.

**Not:** Bu kıta ögesinin her özneliğinin açıklaması, hangi IBM MQ istemcilerinin bu özneliği okuyabileceğini gösterir. Tüm IBM MQ MQI client yapılanış kütüğü stanzas 'larına ilişkin özet çizelge için [Her istemci tarafından okunabilen IBM MQ öznelikleribaşlıklı konuya](#) bakın.

Kanal kısmı aşağıdaki öznelikleri içerir:

### CCSID = sayı

Kullanılacak kodlanmış karakter takımı numarası.

Bu öznelik C, yönetilmeyen .NET, yönetilen .NETve yönetilen XMS .NET istemcileri tarafından okunabilir.

CCSID numarası, [MQCCSID](#) ortam değişkenine eşdeğerdir.

### ChannelDefinitionDizin = yol

İstemci kanal tanımlama çizelgesini içeren dosyanın dizin yolu.

Bu öznelik C, yönetilmeyen .NET, yönetilen .NETve yönetilen XMS .NET istemcileri tarafından okunabilir.

**Windows** Windows sistemlerinde varsayılan değer IBM MQ veri ve günlük dosyaları dizinidir (genellikle C:\ProgramData\IBM\MQ).

**Linux** **AIX** AIX and Linux sistemlerinde varsayılan değer /var/mqm' dir.

ChannelDefinitionDizini, ChannelDefinitionDosya özneliğiyle birlikte çalışan bir URL içerebilir (bkz. "CCDT ' ye URL erişimi" sayfa 52).

ChannelDefinitionDizin yolu, [MQCHLLIB](#) ortam değişkenine eşdeğerdir.

### ChannelDefinitionFile = filename|AMQCLCHL . TAB

İstemci kanal tanımlama çizelgesini içeren dosyanın adı.

Bu öznelik C, yönetilmeyen .NET, yönetilen .NETve yönetilen XMS .NET istemcileri tarafından okunabilir.

İstemci kanal tanımlama çizelgesi, [MQCHLTAB](#) ortam değişkenine eşdeğerdir.

### ReconDelay = (delay [, rand]) (delay [, rand]) ...

ReconDelay özneliği, otomatik olarak yeniden bağlanabilen istemci programları için yeniden bağlanma gecikmesi yapılandırmak üzere bir yönetim seçeneği sağlar.

Bu öznelik C, yönetilmeyen .NET, IBM MQ classes for JMS, yönetilen .NETve yönetilen XMS .NET istemcileri tarafından okunabilir.

Örnek bir yapılandırma:

```
ReconDelay=(1000,200) (2000,200) (4000,1000)
```

Gösterilen örnek, bir saniyelik başlangıç gecikmesini ve 200 milisaniyeye kadar rasgele bir aralığı tanımlar. Bir sonraki gecikme iki saniye artı 200 milisaniyeye kadar rasgele bir aralıktır. Sonraki tüm gecikmeler dört saniye artı 1000 milisaniyeye kadar rasgele bir aralıktır.

### DefRecon = NO|YES|QMGR |DEVRE Dış1

DefRecon özneliği, istemci programlarının otomatik olarak yeniden bağlanmasını sağlamak ya da otomatik olarak yeniden bağlanmak üzere yazılmış bir istemci programının otomatik yeniden bağlanmasını geçersiz kılmak için bir yönetim seçeneği sağlar. Bir program yeniden bağlantıyla uyumlu olmayan MQPMO\_LOGICAL\_ORDER gibi bir seçenek kullanıyorsa, ikincisini ayarlayabilir.

Bu öznitelik C, yönetilmeyen .NET, IBM MQ classes for JMS, yönetilen .NET ve yönetilen XMS .NET istemcileri tarafından okunabilir.

IBM MQ classes for Java otomatik istemci yeniden bağlantısını desteklemez.

Otomatik istemci yeniden bağlantısı genellikle aşağıdaki iki değere bağlıdır:

- Uygulama MQCONNX 'te (ya da JMS Bağlantı Üreticisi) yeniden bağlanma seçeneği ayarlandı
- Kullanılmakta olan herhangi bir istemci bağlantısı tanımlamasında (örneğin, CCDT dosyası kullanılarak sağlanan MQCD yapısı) sağlanan varsayılan yeniden bağlanma seçeneği.

mqclient.ini dosya özneliği **yalnızca**, **DefReconnect** özneliğini ayarlayan bir kanal tanımlaması kullanılmıyorsa ve bu durumda bir kanal tanımlaması sağlanmış gibi davranıyorsa geçerlidir. Kanal **DefReconnect** özneliği (ve geçerliyse bu öznitelik):

- Herhangi biri DISABLED olarak ayarlandıysa uygulama kodunu geçersiz kılar
- Seçenekler MQCONNX 'te belirtildiyse, diğer tüm durumlarda uygulama kodu tarafından geçersiz kılınır

Uygulama ve kanal tanımlaması tarafından sağlanan değerlerin olası tüm birleşimlerini gösteren bir çizelge için [DEFRECON](#) tanımına bakın.

#### Notlar:

- Bir MQCD kullanımda, ancak MQCD\_VERSION\_10' dan önceyi kullanıyorsa, **DefReconnect** parametresi yapının bir parçası değildir. Bu durumda, bu eksik parametrenin değeri, belirtildiyse mqclient.ini file **DefReconnect** (dosya) değeriyle doldurulur. Örneğin, daha eski bir IBM MQ sürümünde oluşturulan ikili biçimli bir CCDT, bir istemci uygulaması tarafından kullanılmaya devam ediyorsa, bu durum oluşabilir.
- IBM MQ istemci kodu (JSON CCDT, bkz. "[JSON biçimi CCDT ' nin yapılandırılması](#)" sayfa 44) tarafından yorumlandığında, her zaman en son sürümde MQCD yapıları oluşturur ve dolayısıyla, belirtik olarak farklı bir değer olmadıkça, bu öznitelik için her zaman varsayılan değer (NO) sağlar.

#### MQReconnectTimeout

İstemcideki otomatik istemci yeniden bağlanma işlevinin bağlantıyı yeniden kurmaya çalıştığı süre üst sınırı (saniye olarak). Varsayılan değer 1800 saniyedir (30 dakika).

Bu öznitelik C ve yönetilmeyen .NET istemcileri ve yönetilen .NET istemcileri tarafından okunabilir.

IBM MQ classes for JMS istemcileri, CLIENTRECONNECTTIMEOUT bağlantı üreticisi özelliğini kullanarak yeniden bağlanmak için bir zaman aşımı belirtebilir. Bu özelliğin varsayılan değeri 1800 saniyedir (30 dakika).

IBM MQ classes for XMS .NET istemcileri, aşağıdaki özellikleri kullanarak yeniden bağlanmak için bir zaman aşımı belirtebilir:

- Bağlantı üreticisi özelliği CLIENTRECONNECTTIMEOUT. Bu özelliğin varsayılan değeri 1800 saniyedir (30 dakika). Bu özellik yalnızca Yönetilen kip için geçerlidir.
- XMSC.WMQ\_CLIENT\_RECONNECT\_TIMEOUT. Bu özelliğin varsayılan değeri 1800 saniyedir (30 dakika). Bu özellik yalnızca Yönetilen kip için geçerlidir.

#### ServerConnectionParms

ServerConnectionParms, [MQSERVER](#) ortam değişkenine eşdeğerdir ve IBM MQ sunucusunun yerini ve kullanılacak iletişim yöntemini belirtir.

Bu öznitelik C, yönetilmeyen .NET, yönetilen .NET ve yönetilen XMS .NET istemcileri tarafından okunabilir.

ServerConnectionParms özneliği yalnızca basit bir kanal tanımlar; bunu bir TLS kanalı ya da kanal çıkışları olan bir kanal tanımlamak için kullanamazsınız. *ChannelName/TransportType/ConnectionName*, *ConnectionName* tam olarak nitelenmiş bir ağ adı olmalıdır. *ChannelName* kanal adını, iletim tipini ve bağlantı adını ayırmak için bu karakter kullanıldığından, eğik çizgi (/) karakterini içeremez.



Bir istemci kanalı tanımlamak için `ServerConnectionParms` kullanıldığında, ileti uzunluğu üst sınırı olan 100 MB kullanılır. Bu nedenle, kanal için yürürlükte olan ileti büyüklüğü üst sınırı, sunucudaki `SVRCONN` kanalında belirtilen değerdir.

Yalnızca tek bir istemci kanal bağlantısı kurulabileceğini unutmayın. Örneğin, iki girdiniz varsa:

```
ServerConnectionParms=R1.SVRCONN/TCP/localhost(1963)
ServerConnectionParms=R2.SVRCONN/TCP/localhost(1863)
```

Sadece ikincisi kullanılır.

Belirtilen iletim tipi için virgülle ayrılmış ad listesi olarak *ConnectionName* belirtin. Genel olarak, yalnızca bir ad gereklidir. Aynı özelliklere sahip birden çok bağlantı yapılandırmak için birden çok *anasistem adı* sağlayabilirsiniz. Bağlantılar, bağlantı başarıyla kuruluncaya kadar bağlantı listesinde belirlendikleri sırayla denenecektir. Bağlantı başarılı olmazsa, istemci yeniden işlemeye başlar. Bağlantı listeleri, yeniden bağlanabilir istemciler için bağlantı yapılandırmak üzere kuyruk yöneticisi gruplarına bir alternatiftir.

### **Put1DefaultAlwaysSync = NO (varsayılan) | YES**

`MQPMO_RESPONSE_AS_Q_DEF` seçeneğiyle `MQPUT1` işlev çağrısının davranışını denetler.

Bu öznitelik C, yönetilmeyen .NET, IBM MQ classes for Java, IBM MQ classes for JMS, yönetilen .NET ve yönetilen XMS .NET istemcileri tarafından okunabilir.

#### **HAYIR**

`MQPUT1 MQPMO_SYNCPOINT` ile ayarlanırsa, `MQPMO_ASYNC_RESPONSE` olarak davranır. Benzer şekilde, `MQPUT1 MQPMO_NO_SYNCPOINT` ile ayarlanırsa, `MQPMO_SYNC_RESPONSE` gibi davranır. Bu varsayılan değerdir.

#### **EVET**

`MQPUT1`, `MQPMO_SYNCPOINT` ya da `MQPMO_NO_SYNCPOINT` ayarlı olsun ya da olmasın, `MQPMO_SYNC_RESPONSE` ayarlanmış gibi davranır.

### **PasswordProtection = Uyumlu (varsayılan) | her zaman | isteğe bağlı**

Bağlantı TLS şifrelemesini kullanmıyorsa, IBM MQ 8.0' den IBM MQ client uygulamalarının bir kuyruk yöneticisine bağlandığında belirttiği kimlik doğrulama kimlik bilgileri IBM MQ MQCSP parola koruma özelliği kullanılarak korunabilir.

Bu öznitelik C, yönetilmeyen .NET, IBM MQ classes for Java, IBM MQ classes for JMS, yönetilen .NET ve yönetilen XMS .NET istemcileri tarafından okunabilir.

MQCSP parola koruması, test ve geliştirme amaçlarıyla yararlıdır; MQCSP parola korumasının kullanılması TLS şifrelemesi ayarlanmasından daha basittir, ancak güvenli değildir.

MQCSP yapısındaki kimlik bilgilerinin ve bu öznitelik için ayarlanabilecek değerlerin korunması hakkında daha fazla bilgi için bkz. [MQCSP parola koruması](#).

### **İlgili görevler**

[IBM MQ MQI uygulamalarının kuyruk yöneticilerine bağlanması](#)

**Multi**

## **ClientExit İstemci yapısını kütüğünün yol kısmı**

İstemciye kanal çıkışlarının varsayılan yerlerini belirtmek için `ClientExitYol` bölümünü kullanın.

**Not:** Bu kıta ögesinin her özneliğinin açıklaması, hangi IBM MQ istemcilerinin bu özneliği okuyabileceğini gösterir. Tüm IBM MQ MQI client yapısını kütüğü stanzas 'larına ilişkin özet çizelge için [Her istemci tarafından okunabilen IBM MQ öznelikler](#) başlıklı konuya bakın.

`ClientExitYol` kısmına aşağıdaki öznelikler eklenebilir:

### **ExitsDefaultYol = dizgi**

İstemciler için 32 bit kanal çıkışlarının konumunu belirtir.

Bu öznitelik C, yönetilmeyen .NET, yönetilen .NET, yönetilen XMS .NET, IBM MQ classes for Javave IBM MQ classes for JMS istemcileri tarafından okunabilir. IBM MQ classes for Java ve IBM MQ classes for JMS istemcileri, Java içinde yazılmamış 32 bit kanal çıkışlarını bulmak için bu özniteliği kullanır.

#### **ExitsDefaultPath64 = dizgi**

İstemciler için 64 bitlik kanal çıkışlarının konumunu belirtir.

Bu öznitelik C, yönetilmeyen .NET, yönetilen .NET, yönetilen XMS .NET, IBM MQ classes for Javave IBM MQ classes for JMS istemcileri tarafından okunabilir. IBM MQ classes for Java ve IBM MQ classes for JMS istemcileri, Java içinde yazılmamış 64 bit kanal çıkışlarını bulmak için bu özniteliği kullanır.

#### **JavaExitsClassPath = dizgi**

Bir Java çıkışı çalıştırıldığında sınıf yoluna eklenecek değerler. Bu, başka bir dildeki çıkışlar tarafından yoksayılır.

Bu öznitelik IBM MQ classes for Java ve IBM MQ classes for JMS istemcileri tarafından okunabilir.

JMS yapılandırma dosyasında JavaExitsClassYol adına standart com.ibm.mq.cfg adı verilir. örnek ve bu tam ad IBM MQ sistem özelliğinde de kullanılır.

### **Multi İstemci yapılış kütüğünün bağlantı kısmı**

Bir uygulama adı belirtmek için bağlantı kısmı kullanın.

**Not:** Bu kıta ögesinin her özniteliğinin açıklaması, hangi IBM MQ istemcilerinin bu özniteliği okuyabileceğini gösterir. Tüm IBM MQ MQI client yapılış kütüğü stanzas 'larına ilişkin özet çizelge için [Her istemci tarafından okunabilen IBM MQ öznitelikleri](#) başlıklı konuya bakın.

Aşağıdaki öznitelik, Bağlantı kısmı içine eklenebilir:

#### **AppName = ExampleAppName**

İstemci yapılış dosyasında bir uygulama adı belirtebilirsiniz.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından kullanılabilir.

### **Multi İstemci yapılış dosyasının JMQI kısmı**

IBM MQ classes for Java ve IBM MQ classes for JMS tarafından kullanılan Java Message Queuing Interface (JMQUI) yapılış deęiřtirgelerini belirtmek için JMQUI kısmı kullanılır.

**Not:** Bu kıta ögesinin her özniteliğinin açıklaması, hangi IBM MQ istemcilerinin bu özniteliği okuyabileceğini gösterir. Tüm IBM MQ MQI client yapılış kütüğü stanzas 'larına ilişkin özet çizelge için [Her istemci tarafından okunabilen IBM MQ öznitelikleri](#) başlıklı konuya bakın.

JMQUI kısmı aşağıdaki özniteliği kapsayabilir:

#### **useMQCSPauthentication = NO|YES**

IBM MQ classes for Java ve IBM MQ classes for JMS uygulamalarının bir kuyruk yöneticisiyle kimlik doęrulaması yaparken Uyumluluk kipini mi, yoksa MQCSP kimlik doęrulama kipini mi kullanacağını denetler.

Bu öznitelik IBM MQ classes for Javave IBM MQ classes for JMS istemcileri tarafından okunabilir.

Bu özniteliğin aşağıdaki deęerleri olabilir:

#### **HAYIR**

Bir kuyruk yöneticisiyle kimlik doęrulaması yaparken uyumluluk kipini kullanın. Bu, IBM MQ 9.3.0' dan önceki sürümlerde varsayılan deęerdir.

#### **EVET**

Bir kuyruk yöneticisiyle kimlik doęrulaması yaparken MQCSP kimlik doęrulama kipini kullanın.

**V 9.3.0** Bu, IBM MQ 9.3.0' in varsayılan deęeridir.

**useMQCSPauthentication** özniteliğinin deęerinden önce gelen kimlik doęrulama kipini ayarlamanın başka yolları da vardır. Uyumluluk kipi ve MQCSP kimlik doęrulama kipi hakkında daha fazla bilgi için bkz. [Java istemcisiyle bağlantı kimlik doęrulaması](#).

## İstemci yapılış kütüğünün LU62, NETBIOS ve SPX kısmı

Yalnızca Windows sistemlerinde, belirtilen ağ protokollerine ilişkin yapılış değiştirgelerini belirtmek için bu bağımsız değişkenleri kullanın.

### LU62 kısmı

SNA LU 6.2 protokolü yapılış değiştirgelerini belirtmek için LU62 kısmına bakın. Bu kıtada aşağıdaki öznitelikler bulunabilir:

**Library1 = DLLName|WCPIC32**

APPC DLL ' nin adı.

**Library2 = DLLName|WCPIC32**

Kod iki ayrı kitaplıkta saklandıysa, Library1 ile aynı.

**TPAdı**

Uzak sitede başlatılacak TP adı.

### NETBIOS kısmı

NetBIOS protokolü yapılış değiştirgelerini belirtmek için NETBIOS kısmına bakın. Bu kıtada aşağıdaki öznitelikler bulunabilir:

**AdapterNum = sayı|0**

LAN bağdaştırıcısının numarası.

**Library1 = DLLName|NETAPI32**

NetBIOS DLL ' inin adı.

**LocalName = ad**

Bu bilgisayarın LAN üzerinde bilindiği ad.

Bu, MQNAME ortam değişkenine eşdeğerdir.

**NumCmds = sayı|1**

Ayrılacak komutlar.

**NumSess = number|1**

Ayrılacak oturum sayısı.

### SPX kısmı

SPX protokolü yapılış değiştirgelerini belirtmek için SPX kısmı kullanılır. Bu kıtada aşağıdaki öznitelikler bulunabilir:

**BoardNum = numara|0**

LAN bağdaştırıcısı numarası.

**KeepAlive = YES|NO**

KeepAlive işlevini açın ya da kapatın.

KeepAlive = YES , SPX 'in bağlantının diğer ucunun hala kullanılabilir olup olmadığını düzenli aralıklarla denetlemesini sağlar. Değilse, kanal kapalıdır.

**Library1 = DLLName|WSOCK32 . DLL**

SPX DLL ' nin adı.

**Library2 = DLLName|WSOCK32 . DLL**

Kod iki ayrı kitaplıkta saklandıysa, Library1 ile aynı.

**Yuva = sayı|5E86**

Onaltılı gösterimde SPX yuva numarası.

## İstemci yapılandırma dosyasının MessageBuffer kısmı

İleti arabellekleriyle ilgili bilgileri belirtmek için MessageBuffer kısmı kullanın.

**Not:** Bu kıta ögesinin her özneliğinin açıklaması, hangi IBM MQ istemcilerinin bu özneliği okuyabileceğini gösterir. Tüm IBM MQ MQI client yapıları kütüğü stanzas 'larına ilişkin özet çizelge için Her istemci tarafından okunabilen IBM MQ öznelikleribaşlıklı konuya bakın.

MessageBuffer kısmına aşağıdaki öznelikler eklenebilir:

**MaximumSize = tamsayı|1**

Önden okuma arabelleğinin büyüklüğü (kilobayt cinsinden), 1-999 999 aralığındadır.

Bu öznelik C, yönetilmeyen .NET, IBM MQ classes for Java, IBM MQ classes for JMS, yönetilen .NETve yönetilen XMS .NET istemcileri tarafından okunabilir.

Aşağıdaki özel değerler var:

**-1**

İstemci uygun değeri belirler.

**0**

İstemci için önden okuma devre dışı bırakıldı.

**PurgeTime = tamsayı|600**

Önden okuma arabelleğinde bırakılan iletilerin temizlenmesi için saniye cinsinden aralık.

Bu öznelik C, yönetilmeyen .NET, IBM MQ classes for Java, IBM MQ classes for JMS, yönetilen .NETve yönetilen XMS .NET istemcileri tarafından okunabilir.

İstemci uygulaması MsgId ya da CorrelId değerine dayalı olarak ileti seçiyorsa, önden okuma arabelleği daha önce istenen MsgId ya da CorrelIdile istemciye gönderilen iletileri içerebilir. Bu iletiler, uygun bir MsgId ya da CorrelIdile bir MQGET komutu verilinceye kadar, önden okuma arabelleğinde kalır. PurgeTimeayarını tanımlayarak, önden okuma arabelleğindeki iletileri temizleyebilirsiniz. Önden okuma arabelleğinde temizleme aralığından daha uzun süre kalan iletiler otomatik olarak temizlenir. Bu iletiler kuyruk yöneticisindeki kuyruktan zaten kaldırıldı; bu nedenle, bu iletilere göz atılmadıkça, iletiler kaybolur.

Geçerli aralık 1-999 999 999 saniye arasında ya da 0 özel değeri, temizleme işleminin gerçekleşmediği anlamına gelir.

**UpdatePercentage = tamsayı|-1**

1-100 aralığındaki güncelleme yüzdesi değeri, bir istemci uygulamasının sunucuya ne zaman yeni bir istekte bulunacağını belirlemek için eşik değerinin hesaplanmasında kullanılır. -1 özel değeri, istemcinin uygun değeri belirlediğini gösterir.

Bu öznelik C, yönetilmeyen .NET, IBM MQ classes for Java, IBM MQ classes for JMS, yönetilen .NETve yönetilen XMS .NET istemcileri tarafından okunabilir.

İstemci düzenli olarak, istemci uygulamasının ne kadar veri tükettiğini gösteren bir istek gönderir. MQGET çağrılarını yoluyla istemci tarafından alınan bayt sayısı  $n$  Teşğini aştığında bir istek gönderilir.  $n$  , sunucuya her yeni istek gönderildiğinde sıfırlanır.

T eşiği aşağıdaki gibi hesaplanır:

$$T = Upper - Lower$$

Üst, Kilobyte cinsinden *MaximumSize* özneliğiyle belirtilen önden okuma arabelleği büyüklüğüyle aynıdır. Varsayılan değeri 100 Kb 'dir.

Alt, Üst değerinden küçük ve *UpdatePercentage* özneliği tarafından belirtilir. Bu öznelik, 1-100 aralığındaki bir sayıdır ve varsayılan değeri 20 'dir. Alt değerler aşağıdaki gibi hesaplanır:

$$Lower = Upper \times UpdatePercentage / 100$$

**Örnek 1:**

MaximumSize ve UpdatePercentage öznelikleri varsayılan değerlerini 100 Kb ve 20 Kb olarak alır.

İstemci, bir iletiyi almak için MQGET ' i çağırır ve bunu art arda yapar. Bu, MQGET n byte tüketinceye kadar devam eder.

Hesaplamayı kullanma

$$T = Upper - Lower$$

$T (100-20) = 80$  Kb 'dir.

Bu nedenle, MQGET çağruları bir kuyruktan 80 Kb 'yi kaldırdığında, istemci otomatik olarak yeni bir istek yapar.

### Örnek 2:

MaximumSize öznitelikleri varsayılan değeri olan 100 Kb 'yi alır ve UpdatePercentage için 40 değeri seçilir.

İstemci, bir iletiyi almak için MQGET ' i çağırır ve bunu art arda yapar. Bu, MQGET n byte tüketinceye kadar devam eder.

Hesaplamayı kullanma

$$T = Upper - Lower$$

$T (100-40) = 60$  Kb

Bu nedenle, MQGET çağruları bir kuyruktan 60 Kb 'yi kaldırdığında, istemci otomatik olarak yeni bir istek yapar. Bu, varsayılan değerlerin kullanıldığı EXAMPLE 1 'den daha kısa bir sürede olur.

Bu nedenle daha büyük bir eşik  $T$  seçildiğinde, isteklerin istemciden sunucuya gönderilme sıklığı azalır. Ters durumda, daha küçük bir eşik  $T$  seçilmesi, istemciden sunucuya gönderilen isteklerin sıklığını artırma eğilimindedir.

Ancak, büyük bir eşik  $T$  seçilmesi, önden okuma arabelleğinin boş olma olasılığı arttıkça, önden okuma başarımının düşeceği anlamına gelebilir. Bu durumda bir MQGET çağrısı duraklatılabilir ve verilerin sunucudan gelmesini bekleyebilir.

## Multi

## İstemci yapılış kütüğünün PreConnect kısmı

mqclient.ini dosyasında PreConnect çıkışını yapılandırmak için PreConnect kısmı kullanın.

**Not:** Bu kıta ögesinin her özneliğinin açıklaması, hangi IBM MQ istemcilerinin bu özneliği okuyabileceğini gösterir. Tüm IBM MQ MQI client yapılış kütüğü stanzas 'larına ilişkin özet çizelge için [Her istemci tarafından okunabilen IBM MQ öznitelikler](#) başlıklı konuya bakın.

PreConnect kısmına aşağıdaki öznitelikler eklenebilir:

### Veri = kullanıcı\_verileri

Bu öznitelik, ön bağlantı çıkışına geçirilen kullanıcı verilerini belirtir. Ön bağlantı çıkışına geçirilen veriler, kullanmakta olduğunuz ön bağlantı çıkışının uygulanmasına ve iletilmesini beklediği verilere özgüdür.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Örneğin, bu öznitelik, LDAP sunucusu kullanılırken bağlantı tanımlamalarının saklandığı havuzun URL adresini belirtmek için kullanılabilir:

```
Data = ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
```

### İşlev = myFunc

PreConnect çıkış kodunu içeren kitaplıktaki işlevsel giriş noktasının adı.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

İşlev tanımlaması, PreConnect çıkış prototipine MQ\_PRECONNECT\_EXIT uymaktadır.

Bu alanın uzunluk üst sınırı: MQ\_EXIT\_NAME\_LENGTH.

### **Birim = myMod**

API çıkış kodunu içeren modülün adı.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Bu alan modülün tam yol adını içeriyorsa, olduğu gibi kullanılır.

### **Sıra = sequence\_number**

Bu çıkışın diğer çıkışlarla görelili olarak çağrıldığı sıra. Düşük sıra numarasına sahip bir çıkış, daha yüksek sıra numarasına sahip bir çıkıştan önce çağrılır. Çıkışların sıra numaralandırmasının sürekli olmasına gerek yoktur; 1, 2, 3 sırası 7, 42, 1096 ile aynı sonuca sahiptir. Bu öznitelik, işaretli bir sayısal değerdir.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

mqclient.ini dosyasında birden çok PreConnect kısmı tanımlanabilir. Her çıkışın işleme sırası, dörtlünün Sıra özniteliğine göre belirlenir.

### **İlgili görevler**

[Havuzdan ön bağlantı çıkışı kullanarak bağlantı tanımlamalarına gönderme yapma](#)

## **Multi İstemci yapılandırma kütüğünün güvenlik kısmı**

Bir kuyruk yöneticisine istemci bağlantıları için AMS 'i geçersiz kılmak ya da etkinleştirmek için Güvenlik kısmı özelliğini kullanın.

**Not:** Bu kısıt öğesinin her özneliliğinin açıklaması, hangi IBM MQ istemcilerinin bu özneliliği okuyabileceğini gösterir. Tüm IBM MQ MQI client yapılandırma kütüğü stanzas 'larına ilişkin özet çizelge için [Her istemci tarafından okunabilen IBM MQ öznelilikler başlıklı konuya](#) bakın.

Güvenlik kısmı aşağıdaki özneliliği kapsayabilir:

### **DisableClientAMS = NO|YES**

DisableClientAMS özneliliği, ürünün önceki bir sürümünden bir kuyruk yöneticisine bağlanmak için IBM MQ istemcisi kullanıyorsanız ve 2085 (MQRC\_UNKNOWN\_OBJECT\_NAME) hatası raporlandıysa, IBM MQ Advanced Message Security (AMS) öğesini devre dışı bırakmanızı sağlar.

IBM MQ Advanced Message Security (AMS), bir IBM MQ istemcisinde otomatik olarak etkinleştirilir ve varsayılan olarak istemci, kuyruk yöneticisindeki nesnelere ilişkin güvenlik ilkelerini denetlemeye çalışır.

Aşağıdaki örneklerde DisableClientAMS özneliliğinin nasıl kullanılacağı gösterilmektedir:

- AMS özelliğini devre dışı bırakmak için:

```
Security:  
DisableClientAMS=Yes
```

- AMS' i etkinleştirmek için:

```
Security:  
DisableClientAMS=No
```

Bu öznitelik C, IBM MQ classes for Javave IBM MQ classes for JMS istemcileri tarafından okunabilir.

### **V 9.3.0 V 9.3.0 MQIInitialKeyDosya = yol adı**

İstemci tarafından sağlanan kimlik bilgilerini şifrelemek için kullanılan ilk anahtar içerikli dosyanın tam yolu ve adı. Anahtar havuzu parolası **runmqicred** yardımcı programı kullanılarak şifrelendiğinde bir ilk anahtar dosyası belirtilirse, ilk anahtar belirtilmelidir.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

### **İlgili görevler**

[İstemcide Gelişmiş İletim Güvenliğinin Devre Dışı Bırakılması](#)

TLS kullanımına ilişkin bilgileri belirtmek için SSL kısmı kullanın.

**Not:** Bu kıta ögesinin her özneliğinin açıklaması, hangi IBM MQ istemcilerinin bu özneliği okuyabileceğini gösterir. Tüm IBM MQ MQI client yapıları kütüğü stanzas 'larına ilişkin özet çizelge için Her istemci tarafından okunabilen IBM MQ öznelikleribaşlıklı konuya bakın.

SSL kısmı aşağıdaki öznelikleri içerir:

V 9.3.0

**OutboundSNI = KANAL | ANASISTEM ADI**

**OutboundSNI** CHANNELolarak ayarlanırsa, SNI yetenekli istemciler, bir TLS bağlantısı başlatırken uzak sisteme SNI ' yi hedef IBM MQ kanal adına ayarlar.

Bu öznelik HOSTNAMEdeğerine ayarlanırsa, SNI yetenekli istemciler SNI üstbilgisini anasistem adı olarak ayarlayarak giden bağlantı isteklerinin TLS anlaşması sırasında uzak kuyruk yöneticisinin varsayılan sertifikasını almasına ve kanal başına sertifikaların kullanılmasına neden olur.

Bu öznelik C, yönetilmeyen .NET, IBM MQ classes for Javave IBM MQ classes for JMS istemcileri tarafından okunabilir.

Özellik değerleri Java/JMS istemcisi tarafından büyük/küçük harfe duyarlı olarak yorumlanır, bu nedenle YES/NO değerleri büyük harfle ayarlanmalıdır.

**OutboundSNI** özelliği HOSTNAMEolarak ayarlandıysa, IBM MQ 9.3.0' den IBM MQ yönetilen .NET istemcisi SERVERNAME değerini ilgili anasistem adına ayarlamak üzere güncellenmiştir. Bu ayar, IBM MQ yönetilen .NET istemcisinin [Red Hat OpenShift rotalarını](#) kullanarak bir kuyruk yöneticisine bağlanmasını sağlar.

**Not: OutboundSNI** ayarı HOSTNAME olan bir uygulama, sertifika etiketi yapılandırılmış bir kanala bağlanırsa, uygulama bir MQRC\_SSL\_INITIALIZATION\_ERROR ile reddedilir ve kuyruk yöneticisi hata günlüklerine bir AMQ9673 iletisi yazdırılır.

**AllowOutboundSNI = YES (varsayılan) | HAYIR**

Etkinleştirilirse, SNI özellikli istemciler, bir TLS bağlantısı başlatılırken uzak sisteme SNI ' yi hedef IBM MQ kanal adına ayarlar. Bu öznelik N0değerine ayarlanırsa, SNI özellikli istemciler, giden bağlantı isteklerinin TLS anlaşması sırasında uzak kuyruk yöneticisinin varsayılan sertifikasını almasına neden olan SNI üstbilgisini ayarlamaz ve bu nedenle kanal başına sertifikalar kullanılamaz.

Bu öznelik C, yönetilmeyen .NET, IBM MQ classes for Javave IBM MQ classes for JMS istemcileri tarafından okunabilir.

Özellik değerleri Java/JMS istemcisi tarafından büyük/küçük harfe duyarlı olarak yorumlanır, bu nedenle YES/NO değerleri büyük harfle ayarlanmalıdır.



**Uyarı:** **V 9.3.0** **Deprecated** IBM MQ 9.3.0 ' den **AllowOutboundSNI** özelliği kullanımdan kaldırılmıştır ve yalnızca geriye dönük uyumluluk amacıyla kullanılabilir.

**AllowOutboundSNI** ayarı YES , **OutboundSNI** ayarının CHANNELolarak ayarlandığı aynı işlevi sağlarken **AllowOutboundSNI** ayarı NO , **OutboundSNI** ayarının HOSTNAMEile aynı işlevi sağlar.

SSL kıtasında hem **AllowOutboundSNI** hem de **OutboundSNI** öznelikleri varsa, **OutboundSNI** ayarı önceliklidir.

IBM i

ALW

**AllowTLSV13 = Y | YES | T | TRUE (varsayılan) | N | NO | F | FALSE**

Bir kuyruk yöneticisinin TLS 1.3 CipherSpecs 'i kullanıp kullanamayacağını belirtir (bkz. [CipherSpecs](#) ' in etkinleştirilmesi).

Bu öznelik C/C++ istemcileri tarafından okunabilir.

Bu öznelik aşağıdaki olası değerleri içerir:

- Y (varsayılan), YES (varsayılan), T (varsayılan) ya da TRUE (varsayılan): Kuyruk yöneticisinin TLS 1.3 CipherSpecskullanmasını sağlayan TLS 1.3 ' ü etkinleştirir.



- N, NO, Fya da FALSE: Kuyruk yöneticisinin TLS 1.3 CipherSpecskullanamadığı anlamına gelen TLS 1.3' ü devre dışı bırakır.

**Not:** MQI istemcisi kullanılırken, uygulama tarafından kullanılmakta olan “İstemci yapılanış kütüğünün SSL kısmı” sayfa 179 dosyasının SSL kısmına belirtik olarak belirtilmedikçe **AllowTLSV13** değeri çıkarılır. Daha fazla bilgi için bkz. [IBM MQ MQI istemcisi ve TLS 1.3.](#)

### **CDPCheckExtensions = YES|NO (varsayılan)**

CDPCheckExtensions , bu kuyruk yöneticisindeki TLS kanallarının CrlDistributionPoint sertifika uzantılarında adı belirtilen CDP sunucularını denetlemeye çalışıp çalışmadığını belirtir.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Bu öznitelik aşağıdaki olası değerleri içerir:

- YES (Evet) (varsayılan): TLS kanalları, bir dijital sertifikanın iptal edilip edilmediğini belirlemek için CDP sunucularını denetlemeye çalışır.
- NO(HAYIR): TLS kanalları CDP sunucularını denetlemeye çalışmaz. Bu, varsayılan değerdir.

### **CertificateLabel = *dizgi***

Kanal tanımlamasının sertifika etiketi.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Ek bilgi için [Sertifika etiketi \(CERTLABL\)](#) konusuna bakın.

### **CertificateValPolicy = *dizgi***

Kullanılan sertifika geçerlilik denetiminin tipini belirler.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Bu öznitelik aşağıdaki olası değerleri içerir:

#### **Fark Etmez**

Temeldeki güvenli yuva kitaplığı tarafından desteklenen herhangi bir sertifika geçerlilik denetimi ilkesini kullanın. Bu ayar varsayılan ayardır.

#### **RFC5280**

Yalnızca RFC 5280 standardına uyan sertifika geçerlilik denetimini kullanın.

### **ClientRevocationDenetimleri = REQUIRED|İSTEĞE BAĞLI|DEVRE DİŞİ**

İstemci bağlantı çağrısı bir TLS kanalı kullanıyorsa, sertifika iptal denetiminin nasıl yapılandırılacağını belirler. Ayrıca bkz. [OCSPAuthentication](#).

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Bu öznitelik aşağıdaki olası değerleri içerir:

#### **REQUIRED (varsayılan)**

CCDT ' den sertifika iptal yapılandırmasını yüklemeyi ve yapılandırılan şekilde iptal denetimini gerçekleştirmeyi dener. CCDT kütüğü açılmıyorsa ya da sertifikanın geçerliliği denetlenemiyorsa (örneğin, bir OCSP ya da CRL sunucusu kullanılmadığı için), MQCONN çağrısı başarısız olur. CCDT ' de iptal yapılandırması yoksa iptal denetimi gerçekleştirilmez, ancak bu kanalın başarısız olmasına neden olmaz.

**Windows** Windows sistemlerinde, CRL iptal denetimi için Active Directory ' yi de kullanabilirsiniz. OCSP iptal denetimi için Active Directory ' yi kullanamazsınız.

MQSCO ya da CCDT kullanıyorsanız, bağlantı başarılı olur. CCDT dosyası yoksa ve MQSCO da sağlanmıyorsa, bağlantı 2059 neden koduyla başarısız olur ve hata günlüğü raporları AMQ9518E : Dosya ' /var/mqm/AMQCLCHL.TAB ' bulunamadı.

#### **İsteğe Bağlı**

GEREKLİ seçeneğine gelince, ancak sertifika iptal yapılandırması yüklenemezse, kanal başarısız olmaz.



## DEVRE DIŐI

CCDT ' den sertifika iptal yapılandırmasını yükleme girişiminde bulunulmadı ve sertifika iptal denetimi yapılmadı.

**Not:** MQCONN çağrılarını yerine MQCONNX kullanıyorsanız, MQSCO aracılığıyla kimlik doğrulama bilgileri kayıtlarını (MQAIR) belirtmeyi seçebilirsiniz. Bu nedenle, CCDT dosyası açılmazsa, ancak bir MQAIR sağladığınızı varsayarsanız (bunu yapmamayı seçerseniz bile), MQCONNX ile ilgili varsayılan davranış başarısız olmaz.

## EncryptionPolicySuiteB = *dizgi*

Bir kanalın Suite-B uyumlu şifreleme kullanıp kullanmadığını ve hangi güç düzeyinin kullanılacağını belirler.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Bu öznitelik aşağıdaki olası değerleri içerir:

### YOK

Suite-B uyumlu şifreleme kullanılmaz. Bu ayar varsayılan ayardır.

### 128\_BIT,192\_BIT

Güvenlik düzeyini hem 128 bit hem de 192 bit düzeylerine ayarlar.

### 128\_BIT

Güvenlik düzeyini 128 bit düzeyine ayarlar.

### 192\_BIT

Güvenlik düzeyini 192 bit düzeyine ayarlar.

V 9.3.0

ALW

## EnvironmentScope=PROCESS|CONNECTION

IBM MQ ' in tüm süreç için tek bir IBM Global Security Kit (GSKit) ortamı mı, yoksa her bağlantı için bir GSKit ortamı mı kullanacağını denetler.

Bu öznitelik C istemcileri tarafından okunabilir.

Bu öznitelik aşağıdaki olası değerleri içerir:

## Süreç

Süreç tarafından yaratılan birden çok bağlantı için tek bir GSKit ortamı kullanılır. Bu ayarın kullanılması, işlem içindeki tüm etkin TLS bağlantıları duruncaya kadar TLS anahtar deposu değişikliklerinin kullanılmayacağı anlamına gelir.

Bu değer varsayılan değerdir.

## Bağlantı

Aynı süreç içinde her bağlantı için bir GSKit ortamı yaratılır. Bunun etkinleştirilmesi, TLS anahtar deposu değişikliklerinin süreç tarafından başlatılan yeni TLS bağlantıları tarafından hemen alınacağı anlamına gelir.



**Uyarı:** Bu çalışma kipinin etkinleştirilmesi, uygulamaların her GSKit ortamını yaratmak için ek CPU ve Bellek kaynaklarını kullanmasına neden olur. Bu kaynak tüketimi, her ek eşzamanlı TLS bağlantısıyla birlikte artar.

ALW

## MinimumRSAKeyBoyut=int

RSA sertifikalarının kabul edilebilmesi için sahip olması gereken anahtar büyüklüğü alt sınırını belirler. 0 ya da üstüne eşit herhangi bir değere izin verir. Belirtilmezse, varsayılan değer olarak 1 kullanılır.

Bu öznitelik C/C++ istemcileri tarafından okunabilir.

## OCSPAuthentication = İSTEĞE BAĞLI|GEREKLI|UYARI

OCSP etkinleştirildiğinde ve OCSP iptal denetiminin sertifika iptal durumunu saptayamadığında IBM MQ davranışını tanımlar. Ayrıca bkz. [ClientRevocationChecks](#).

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Bu öznitelik aşağıdaki olası değerleri içerir:

## İsteğe Bağlı

OCSP denetimi tarafından belirlenemeyen iptal durumlu sertifikalar kabul edilir ve uyarı ya da hata iletisi oluşturulmaz. SSL ya da TLS bağlantısı, iptal denetimi yapılmamış gibi devam eder.

## ZORUNLU

OCSP denetimi, denetlenen her SSL ya da TLS sertifikası için kesin bir iptal sonucu vermelidir. Doğrulanamayan bir iptal durumuna sahip SSL ya da TLS sertifikası bir hata iletisiyle reddedilir. Kuyruk yöneticisi SSL olay iletileri etkinleştirildiyse, MQRQ\_SSL\_HANDSHAKE\_ERROR ReasonQualifier ile bir MQRQ\_CHANNEL\_SSL\_ERROR iletisi üretilir. Bağlantı kapatıldı.

Bu değer varsayılan değerdir.

## WARN

OCSP iptal denetimi herhangi bir SSL ya da TLS sertifikasının iptal durumunu belirleyemezse, kuyruk yöneticisi hata günlüklerinde bir uyarı bildirilir. Kuyruk yöneticisi SSL olay iletileri etkinleştirildiyse, MQRQ\_SSL\_UNKNOWN\_REVOCATION için ReasonQualifier olan bir MQRQ\_CHANNEL\_SSL\_WARNING iletisi üretilir. Bağlantının devam etmesine izin verilir.

## OCSPCheckExtensions = YES|NO (EVET)

IBM MQ ' un AuthorityInfoErişim sertifikası uzantılarında işlem yapıp yapılmayacağını denetler.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Değer NOolarak ayarlanırsa, IBM MQ AuthorityInfoErişim sertifikası uzantılarını yoksayar ve OCSP güvenlik denetimi girişiminde bulunmaz. Varsayılan değer YES(EVET) değeridir.

## ALW OCSPTimeout = sayı

İptal denetimi gerçekleştirilirken OCSP yanıtlayıcısının bekleneceği saniye sayısı.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

IBM MQ 9.3.0 için 0 değeri ayarlanırsa, 30 saniyelik varsayılan zamanaşımı kullanılır.

Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 30 saniye kullanılır.

## ALW PeerCertChainValidation=dizgi

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Dizgi iki değerden biri olabilir:

- Usepeerchain [ **Varsayılan**]: Eşdüzey tarafından sağlanan sertifika zinciri, sertifikaları doğrularken güven zinciri boşlukları arasında köprü kurmak için kullanılabilir. Kök sertifika dışında.
- Truststoreonly [ **Önerilmez**]: Eşin sertifikasını doğrulamak için yalnızca güvenilirlik deposundaki sertifikalar kullanılır.

## SSLCryptoHardware = dizgi

Sistemde bulunan PKCS #11 şifreleme donanımını yapılandırmak için gereken parametre dizgisini belirler.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Şu biçimde bir dizgi belirtin: GSK\_PKCS11 = *driver path and filename;token label;token password;symmetric cipher setting*;

Örneğin: GSK\_PKCS11=/usr/lib/pkcs11/  
PKCS11\_API.so;tokenlabel;passw0rd;SYMMETRIC\_CIPHER\_ON

Sürücü yolu, PKCS #11 kartı için destek sağlayan, paylaşılan kitaplığın mutlak yoludur. Sürücü dosyası adı, paylaşılan kitaplığın adıdır. PKCS #11 sürücü yolu ve dosya adı için gereken değerlerin bir örneği şudur: /usr/lib/pkcs11/PKCS11\_API.so. Simetrik şifreleme işlemlerine GSKitüzerinden erişmek için simetrik şifre ayarı parametresini belirtin. Bu parametrenin değeri aşağıdakilerden biri olabilir:

## SYMMETRIC\_CIPHER\_OFF

Simetrik şifreleme işlemlerine erişmeyin. Bu ayar varsayılan ayardır.

## SYMMETRIC\_CIPHER\_ON

Simetrik şifreleme işlemlerine erişin.

**Linux** **AIX** Dizginin farklı bileşenlerini sağladığınızda, noktalı virgül karakteri açıklama olarak değerlendirildiğinden, ters eğik çizgi karakterini kullanarak noktalı virgül karakterlerine çıkış karakteri eklemelisiniz. Örneğin: '\ ;'

**V 9.3.0** **SSLCryptoHardware** öznitelik dizgisinde bulunan simge parolasını korumanız gerekir. Daha fazla bilgi için bkz. [IBM MQ istemcileri şifreleme donanımı kullanıyor](#).

**V 9.3.0** Şifrelenmiş parolaları işlemek için artık dizginin uzunluğuna sınır yoktur.

Varsayılan değer boşluktur. Doğru biçimde olmayan bir dizgi belirtirseniz, bir hata oluşur.

#### **SSLCryptoHardwareKeyFile = yol adı**

**SSLCryptoHardware** özneliğiyle belirtilen PKCS #11 şifreleme donanımı yapılandırma dizgisinde parolayı şifrelemek için kullanılan ilk anahtar içeren dosyanın tam yolu ve adı. Şifreleme donanımı yapılandırma dizgisindeki parola **runp11cred** komutu kullanılarak şifrelendiğinde bir ilk anahtar dosyası belirtilirse ilk anahtar belirtilmelidir. Daha fazla bilgi için bkz. [IBM MQ Client using cryptographic hardware](#).

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

#### **SSLFipsRequired = YES|NO (SSLFipsGerekli)**

IBM MQ içinde şifreleme gerçekleştirilirse yalnızca FIPS onaylı algoritmaların kullanılıp kullanılmayacağını belirtir.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Şifreleme donanımı yapılandırıldıysa, kullanılan şifreleme modülleri donanım ürünü tarafından sağlanan modüllerdir. Bunlar, kullanılmakta olan donanım ürününe bağlı olarak belirli bir düzeyde FIPS onaylı olabilir ya da olmayabilir.

#### **SSLHTTPProxyName = dizgi**

Dizgi, GSKit tarafından OCSP denetimleri için kullanılacak HTTP Yetkili sunucusunun anasistem adı ya da ağ adresidir. Bu adresi parantez içine alınmış isteğe bağlı bir kapı numarası izleyebilir. Kapı numarasını belirtmezseniz, varsayılan HTTP kapısı olan 80 kullanılır.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

**AIX** AIX üzerindeki 32 bit istemcileri için ağ adresi yalnızca bir IPv4 adresi olabilir.

Diğer platformlarda ağ adresi bir IPv4 ya da IPv6 adresi olabilir.

Örneğin, bir güvenlik duvarı OCSP yanıtlayıcısının URL adresine erişimi engelliyorsa bu öznitelik gerekli olabilir.

#### **ALW SSLHTTPConnectTimeout = sayı|0**

İptal denetimi gerçekleştirilirken bir HTTP sunucusuyla ağ bağlantısının başarıyla kurulması için beklenecek saniye sayısı.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 0 (kapalı) kullanılır.

#### **SSLKeyRepository = yol adı**

**V 9.3.0** **V 9.3.0** Kullanıcının sayısal sertifikasını tutan anahtar havuzunun tam yolu ve dosya adı. Dosya uzantısı belirtilmezse, dosya uzantısının .kdb olduğu varsayılır.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

#### **V 9.3.0** **V 9.3.0** **SSLKeyRepositoryParola = parola**

Anahtar havuzuna erişmek için parola. Değer bir düz metin dizgisi ya da **runmqicred** yardımcı programı kullanılarak şifrelenmiş bir parola olabilir.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

### **SSLKeyResetSayı = tamsayı|0**

Gizli anahtar yeniden anlaşılmadan önce TLS kanalında gönderilen ve alınan şifrelenmemiş bayt sayısı.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

Değer 0-999999999 aralığında olmalıdır.

Varsayılan değer 0 'dır; bu, gizli anahtarların hiçbir zaman yeniden anlaşılmadığı anlamına gelir.

1-32768 arasında bir değer belirtirseniz, TLS kanalları 32768 (32Kb) gizli anahtar sıfırlama sayısını kullanır. Bu, küçük gizli anahtar sıfırlama değerleri için ortaya çıkabilecek aşırı anahtar sıfırlamalarını önlemek içindir.

## **Multi İstemci yapılanış kütüğünün TCP kısmı**

TCP ağ protokolü yapılanış değiştirgelerini belirtmek için TCP kısmı kullanılır.

**Not:** Bu kıta ögesinin her özneliğinin açıklaması, hangi IBM MQ istemcilerinin bu özneliği okuyabileceğini gösterir. Tüm IBM MQ MQI client yapılanış kütüğü stanzas 'larına ilişkin özet çizelge için Her istemci tarafından okunabilen IBM MQ öznelikleri başlıklı konuya bakın.

TCP kısmı aşağıdaki öznelikleri içerir:

### **ClntRcvBuffSize = sayı|0**

İstemci-bağlantı sunucusu bağlantı kanalının istemci sonu tarafından kullanılan TCP/IP alma arabelleğinin bayt cinsinden büyüklüğü.

Bu öznitelik C, yönetilmeyen .NET, IBM MQ classes for Java, IBM MQ classes for JMS, yönetilen .NET ve yönetilen XMS .NET istemcileri tarafından okunabilir.

Sıfır değeri, işletim sisteminin IBM MQ tarafından düzeltilen arabellek boyutlarının tersine, arabellek boyutlarını yöneteceğini gösterir. Değer sıfır olarak ayarlanırsa, işletim sistemi varsayılanları kullanılır. Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 32768 kullanılır.

### **ClntSndBuffSize = sayı|0**

İstemci-bağlantı sunucusu bağlantı kanalının istemci sonu tarafından kullanılan TCP/IP gönderme arabelleğinin bayt cinsinden büyüklüğü.

Bu öznitelik C, yönetilmeyen .NET, IBM MQ classes for Java, IBM MQ classes for JMS, yönetilen .NET ve yönetilen XMS .NET istemcileri tarafından okunabilir.

Sıfır değeri, işletim sisteminin IBM MQ tarafından düzeltilen arabellek boyutlarının tersine, arabellek boyutlarını yöneteceğini gösterir. Değer sıfır olarak ayarlanırsa, işletim sistemi varsayılanları kullanılır. Herhangi bir değer ayarlanmazsa, IBM MQ varsayılan değeri olan 32768 kullanılır.

### **Connect\_Timeout = sayı**

Yuvayı bağlama girişiminin zamanaşımına uğraması için geçen saniye sayısı.

Zamanuyumsuz bir connect () çağrısından önce **ConnectTimeout** = 0 ve SOCK\_NONBLOCK komutu verilirse, çağrı engellenemez. 20 saniyelik varsayılan zamanaşımı değeri (CONNECT\_WAIT\_MAX), bağlantı durumunu denetlemek için geçerlidir.

Bu öznitelik C, yönetilmeyen .NET, IBM MQ classes for Java ve IBM MQ classes for JMS istemcileri tarafından okunabilir.

IBM MQ kanal işlemleri, engelleyici olmayan yuvalar üzerinden bağlanır. Bu nedenle, yuvanın diğer ucu hazır değilse, connect () hemen *EINPROGRESS* ya da *EWOULDBLOCK* ile geri döner. Bundan sonra, yeniden bağlanma girişiminde bulunulmaz.

Connect\_Timeout değeri sıfır dışında bir değere ayarlanırsa, IBM MQ yuvanın hazır olması için select () çağrısında belirtilen süreyi bekler. Bu, sonraki bir connect () çağrısının başarı şansını artırır. Bu seçenek, ağ üzerindeki yüksek yük nedeniyle, bağlantıların bekleme süresi gerektirdiği durumlarda yararlı olabilir.

Connect\_Timeout, ClntSndBuffSize ve ClntRcvBuffSize parametreleri arasında ilişki yoktur.

### **IPAddressVersion = MQIPADDR\_IPV4|MQIPADDR\_IPV6**

Kanal bağlantısı için hangi IP protokolünün kullanılacağını belirler.

Bu öznitelik C, yönetilmeyen .NET, yönetilen .NET ve yönetilen XMS .NET istemcileri tarafından okunabilir.

Olası MQIPADDR\_IPV4 ya da MQIPADDR\_IPV6 dizgi değerleri vardır. Bu değerler, **ALTER QMGR IPADDRV** ve **MQIPADDRV** ortam değişkenindeki IPV4 ve IPV6 ile aynı anlamlara sahiptir.

### **KeepAlive = YES|NO**

KeepAlive işlevini açın ya da kapatın. KeepAlive=YES, TCP/IP 'nin bağlantısının diğer ucunun hala kullanılabilir olup olmadığını düzenli aralıklarla denetlemesini sağlar. Değilse, kanal kapalıdır.

Bu öznitelik C, yönetilmeyen .NET, IBM MQ classes for Java, IBM MQ classes for JMS, yönetilen .NET ve yönetilen XMS .NET istemcileri tarafından okunabilir.

### **Windows Library1 = DLLName|WSOCK32**

(Yalnızca Windows ) TCP/IP yuvaları DLL 'sinin adı.

Bu öznitelik C ve yönetilmeyen .NET istemcileri tarafından okunabilir.

## **Linux Windows V 9.3.3 İstemci yapılandırma kütüğünün izleme kısmı**

IBM MQ .NET ve XMS .NET istemci kitaplıkları için izlemeyi etkinleştirmek üzere İzleme kısmına bakın.

TRACE kısmı aşağıdaki öznitelikleri içerir:

### **MQDotnetTraceLevel=0 (varsayılan) |1|2**

**MQDotnetTraceLevel** özneliği, IBM MQ .NET izlemesini başlatmak ya da durdurmak için kullanılır:

- 0: İzlemeyi durdurur-bu varsayılan değerdir.
- 1: İzlemeyi daha az ayrıntıyla başlatır.
- 2: İzlemeyi tam ayrıntılarla başlatır-önerilir.

Bu öznitelik, yönetilen IBM MQ .NET istemcisi tarafından okunabilir.

### **MQDotnetTracePath =yol adı**

**MQDotnetTracePath** özneliği, IBM MQ .NET izleme dosyalarının yaratılacağı bir klasörü gösterir. Yol boşsa ya da **MQDotnetTracePath** özelliği tanımlanmamışsa, uygulamanın yürürlükteki dizini kullanılır.

Bu öznitelik, yönetilen IBM MQ .NET istemcisi tarafından okunabilir.

### **MQDotnetErrorYol =yol adı**

**MQDotnetErrorPath** özneliği, IBM MQ .NET izlemesi için hata günlüğü dosyalarının yaratılacağı bir klasörü gösterir. Yol boşsa ya da **MQDotnetErrorPath** özneliği tanımlanmamışsa, uygulamanın yürürlükteki dizini kullanılır.

Bu öznitelik, yönetilen IBM MQ .NET istemcisi tarafından okunabilir.

### **XMSDotnetTraceLevel=0 (varsayılan) |1|2**

**XMSDotnetTraceLevel** özneliği, XMS .NET izlemesini başlatmak ya da durdurmak için kullanılır:

- 0: İzlemeyi durdurur-bu varsayılan değerdir.
- 1: İzlemeyi temel biçimle başlatır.
- 2: İzlemeyi Gelişmiş biçimiyle başlatır.

Bu öznitelik, yönetilen XMS .NET istemcisi tarafından okunabilir.

### **XMSDotnetTraceFilePath=dosyaadı**

**XMSDotnetTraceFilePath** özneliği için bir değer ayarlanmamışsa ya da bu öznitelik varsa, ancak boş bir dizgi içeriyorsa, XMS .NET ile ilgili izleme dosyası yürürlükteki dizine yerleştirilir. İzleme dosyasını adlandırılmış bir dizinde saklamak için, **XMSDotnetTraceFilePath** dizininde dizin adını belirtin; örneğin, XMSDotnetTraceFilePath="c : \somepath".

Bu öznitelik, yönetilen XMS .NET istemcisi tarafından okunabilir.

#### **XMSDotnetTraceSpecification =ComponentName=type=state**

**XMSDotnetTraceSpecification** özniteliği, izlemek istediğiniz sınıfın adını ve XMS .NETiçin gereken izleme tipini belirtir:

- *ComponentName* , izlemek istediğiniz sınıfın adıdır. Bu adda bir \* genel arama karakteri kullanabilirsiniz. Örneğin, \*=all=enabled tüm sınıfları izlemek istediğinizi belirtir ve IBM.XMS.impl.\*=all=enabled yalnızca API izlemesine gerek duyduğınızı belirtir.
- *tip* şu izleme tiplerinden herhangi biri olabilir: all, debug, event, EntryExit.
- *state* (durum) etkinleştirilebilir ya da devre dışı bırakılabilir.

Bir ':' (iki nokta üst üste) sınırlayıcı kullanarak birden çok izleme öğesini bir araya getirebilirsiniz.

Bu öznitelik, yönetilen XMS .NET istemcisi tarafından okunabilir.

#### **XMSDotnetTraceFileSize=size**

**XMSDotnetTraceFileSize** özniteliği, XMS .NETiçin oluşturulması gereken izleme dosyasının büyüklük üst sınırını belirtir. Varsayılan üst sınır, XMSDotnetTraceFileSize=20olarak belirtilen 20 MB 'dir.

Bu öznitelik, yönetilen XMS .NET istemcisi tarafından okunabilir.

#### **XMSDotnetTraceFileNumber=sayı**

**XMSDotnetTraceFileNumber** özniteliği, XMS .NETiçin alınacak izleme dosyalarının sayısını belirtir. Varsayılan değer 4 ' tür (bir etkin dosya ve üç arşiv dosyası). İzin verilen alt sınır 2' dir.

Bu öznitelik, yönetilen XMS .NET istemcisi tarafından okunabilir.

#### **İlgili görevler**

[IBM MQ .NET uygulamalarının mqclient.ini ile izlenmesi](#)

[XMS .NET uygulamalarının mqclient.ini ile izlenmesi](#)

#### **Multi**

### **Etkinlik izleme yapılandırma dosyası, mqat.ini**

Etkinlik izleme yapılandırma dosyası mqat . ini, etkinlik izleme davranışını yapılandırmak için kullanılır. Bu dosya, raporlama etkinliği izleme verilerinin düzeyini ve sıklığını tanımlamak için kullanılır. Dosya, bir uygulamanın adına dayalı olarak etkinlik izlemesini etkinleştirme ve devre dışı bırakma kurallarını tanımlamak için de bir yol sağlar.

mqat . ini dosyası, mqs . ini ve qm . ini dosyalarıyla aynı kıta anahtarı ve parametre değer çifti biçimini izler. Dosya, tüm etkinlik izlemesi için varsayılan olarak raporlama etkinliği izleme verilerinin düzeyini ve sıklığını yapılandırmak için kullanılan AllActivityTrace adlı tek bir bölümden oluşur. Dosya birden çok ApplicationTrace (Uygulama İzlemesi) kısmı da içerebilir. Bu bağımsız birimlerin her biri, bir ya da daha çok bağlantıya ilişkin izleme davranışı için, bağlantıların uygulama adını kuralla eşleştirmeye dayalı olarak bir kural tanımlar. Daha fazla bilgi için bkz. [Uygulama etkinliği izlemesi ve mqat . ini kullanarak etkinlik izleme davranışını yapılandırma](#).

Kuyruk yöneticisi, bir bağlantı için hangi stanzas ayarlarının kullanılacağını belirlemek için bir dizi kural uygular. İsteğe bağlı olarak, bir ApplicationTrace (Uygulama İzlemesi) kistasıyla eşleşen bağlantılar için AllActivityİzleme kısmı altında genel izleme düzeyi ve sıklık ayarlarını geçersiz kılabilirsiniz. Daha fazla bilgi için bkz. [mqat . inikullanarak etkinlik izleme davranışını yapılandırma](#).

#### **Dizin konumları**

**Linux** **IBM i** **AIX** AIX and Linux ve IBM i sistemlerinde mqat . ini , qm . ini dosyasıyla aynı konum olan kuyruk yöneticisi veri dizininde bulunur.

**Windows** Windows sistemlerinde mqat . ini , C:\Program Files\IBM\WebSphere MQ\qmgrs\queue\_manager\_namekuyruk yöneticisi veri dizininde bulunur. İzlenecek uygulamaları çalıştıran kullanıcıların bu dosyayı okuma izni olması gerekir.

## AllActivity mqat.ini dosyasının izleme kısmı

mqat.ini yapılandırma dosyasının AllActivityİzleme kısmı, bir kuyruk yöneticisine ilişkin izleme düzeylerini yapılandırmak için kullanılan parametreleri belirtir.

Tek bir AllActivityİzleme kısmı, geçersiz kılınmadıkça, tüm IBM MQ bağlantılarına uygulanan etkinlik izlemesine ilişkin ayarları tanımlar.

AllActivityİzleme bölümündeki tek tek değerler, bir ApplicationTrace (Uygulama İzlemesi) bölümündekidaha özel bilgilerle geçersiz kılınabilir.

Birden çok AllActivityİzleme kısmı belirtilirse, son kıtadaki değerler kullanılır. Seçilen AllActivityİzlemesinden eksik parametreler varsayılan değerleri alır. Önceki AllActivityİzleme bağımsız değişkenlerindeki parametreler ve değerler yoksayılır.

### ActivityInterval

İzleme iletileri arasındaki saniye cinsinden zaman aralığı. Etkinlik izleme bir süreölçer iş parçacığı kullanmaz; bu nedenle, izleme iletileri zaman geçtiğinde tam olarak yazılmaz; zaman aralığı geçtikten sonra ilk MQI işlemi yürütüldüğünde yazılır. Bu değer 0 ise, bağlantı kesildiğinde (ya da etkinlik sayısına ulaşıldığında) izleme iletileri yazılır. Varsayılan değer 1 'dir.

### ActivityCount

İzleme iletileri arasındaki MQI işlemlerinin sayısı. Bu değer 0 ise, bağlantı kesildiğinde (ya da etkinlik aralığı geçtiğinde) izleme iletileri yazılır. Varsayılan değer 100 'dür.

### TraceLevel

Her işlem için izlenecek parametre ayrıntısı miktarı. Her izleme düzeyi için hangi parametrelerin dahil edileceği tek tek işlem ayrıntılarının açıklaması. LOW, MEDIUMya da HIGHolarak ayarlayın. Varsayılan değer MEDIUM' dur.

### TraceMessageVerileri

MQGET, MQPUT, MQPUT1ve Callback işlemleri için byte cinsinden izlenecek ileti verileri miktarı. Varsayılan değeri 0'dır.

### StopOnGetTraceMsg

ON ya da OFFolarak ayarlanabilir. Varsayılan değer olarak ON(AÇIK) değeri kullanılır.

### SubscriptionDelivery

BATCHED ya da IMMEDIATEolarak ayarlanabilir. Bir ya da daha çok etkinlik izleme aboneliği olduğunda **ActivityInterval** ve **ActivityCount** parametrelerinin kullanılıp kullanılmayacağını belirler. Bu parametrenin IMMEDIATE olarak ayarlanması, izleme verileri eşleşen bir aboneliğe sahip olduğunda **ActivityInterval** ve **ActivityCount** değerlerinin 1 efektif değerleriyle geçersiz kılınmasına neden olur. Her etkinlik izleme kaydı, aynı bağlantıdaki diğer kayıtlarla toplu olarak gönderilmez ve bunun yerine gecikme olmadan hemen aboneliğe teslim edilir. IMMEDIATE ayarı, etkinlik izleme verilerinin toplanmasına ilişkin başarıyı artırır. Varsayılan ayar BATCHEDdeğeridir.

### İlgili görevler

[mqat.ini ile etkinlik izleme davranışını yapılandırma](#)

## mqat.ini dosyasının ApplicationTrace kısmı

mqat.ini yapılandırma dosyası birden çok ApplicationTrace stanzas içerebilir. Bu bağımsız birimlerin her biri, bir ya da daha çok bağlantıya ilişkin izleme davranışı için, bağlantıların uygulama adını kuralla eşleştirmeye dayalı olarak bir kural tanımlar.

ApplicationTrace kısmı için aşağıdaki değerleri ayarlayabilirsiniz:

### Takip edin

AÇIK ya da KAPALIolarak ayarlanabilen etkinlik izleme anahtarı. **Trace** parametresi, varsayılan değeri olmayan zorunlu bir parametredir. Geçerli uygulama kısmı kapsamı için etkinlik izlemesinin etkin olup olmadığını belirlemek üzere uygulamaya özgü kıtada kullanılabilir. Bu değer, kuyruk yöneticisine ilişkin **ACTVTRC** ve **ACTVCONO** ayarlarını geçersiz kıldığını unutmayın.



## AppName

**AppName** değıştirgesi bir karakter dizisi olarak belirtildi ve varsayılan değeri olmayan zorunlu bir değıştirgedir. Bu değeri, ApplicationTrace (Uygulama İzleme) kısmı için geçerli olan uygulamaları belirlemek için kullanılır. API çıkış bağlamı yapısındaki ( MQMD.PutAppName ile eşdeğerdır) **AppName** değeriyle eşleştirilir. **AppName** değeri için uygulama ortamına göre değışir.

Multiplatforms üzerinde, yalnızca MQAXC.AppName , kıtadaki değeri eşleştirildi. Karşılaştırma yapılırken, en sağdaki yol ayırıcının solundaki karakterler yoksayılır.

**AppName** değeri için, o noktadan sonra herhangi bir sayıda karakterle eşleşmesi için tek bir genel arama karakteri (\*) kullanılabilir. **AppName** değeri tek bir genel arama karakteri (\*) olarak ayarlanırsa, **AppName** değeri tüm uygulamalarla eşleşir.

## IBM i ApplFunction

**ApplFunction** parametresi bir karakter dizisi olarak belirtilir. Varsayılan değeri \* değeri. Bu parametrenin değeri, ApplicationTrace kısmı ve **AppName** değeri için geçerli olan uygulama programlarını nitelemek için kullanılır.

Bölüm isteğe bağlıdır ve yalnızca IBM i kuyruk yöneticileri için geçerlidir. Herhangi bir sayıda karakterle eşleşmesi için **AppName** değeri için tek bir genel arama karakteri (\*) kullanılabilir. Örneğin, **AppName** = \* ve **ApplFunction** = AMQSPUTO belirten bir ApplicationTrace kısmı, herhangi bir işten AMQSPUTO programının tüm çağrılarını için geçerlidir.

## ApplClass

**ApplClass** değıştirgesi bir uygulamanın sınıfını tanımlar ve aşağıdaki değeri ayarlanabilir:

- KULLANICI
- MCA
- ALL (Varsayılan değeri budur)

**AppType** değeri için IBM MQ bağlantılarına nasıl karşılık geldiğine ilişkin bir açıklama için [mqat.inikullanılarak etkinlik izleme davranışının yapılandırılması başlıklı konudaki Tablo 3](#) konusuna bakın.

İsteğe bağlı olarak, AllActivityİzleme kısmı altındaki genel izleme düzeyi ve sıklık ayarları, bir ApplicationTrace kısmı ile eşleşen bağlantılar için geçersiz kılınabilir.

Aşağıdaki parametreler bir ApplicationTrace (Uygulama İzleme) kısmı altında ayarlanabilir. Bu değeri ayarlanmazsa, değeri [AllActivity izleme kısmı](#) ayarlarından devralınır:

## ActivityInterval

İzleme iletileri arasındaki saniye cinsinden zaman aralığı. Etkinlik izleme bir süreölçer iş parçacığı kullanmaz; bu nedenle, izleme iletileri zaman geçtiğinde tam olarak yazılmaz; zaman aralığı geçtikten sonra ilk MQI işlemi yürütüldüğünde yazılır. Bu değeri 0 ise, bağlantı kesildiğinde (ya da etkinlik sayısına ulaşıldığında) izleme iletileri yazılır. Varsayılan değeri 1 'dir.

## ActivityCount

İzleme iletileri arasındaki MQI işlemlerinin sayısı. Bu değeri 0 ise, bağlantı kesildiğinde (ya da etkinlik aralığı geçtiğinde) izleme iletileri yazılır. Varsayılan değeri 100 'dür.

## TraceLevel

Her işlem için izlenecek parametre ayrıntısı miktarı. Her izleme düzeyi için hangi parametrelerin dahil edileceği tek tek işlem ayrıntılarının açıklaması. LOW, MEDIUM ya da HIGH olarak ayarlayın. Varsayılan değeri MEDIUM' dur.

## TraceMessageVerileri

MQGET, MQPUT, MQPUT1 ve Callback işlemleri için byte cinsinden izlenecek ileti verileri miktarı. Varsayılan değeri 0'dır.

## StopOnGetTraceMsg

ON ya da OFF olarak ayarlanabilir. Varsayılan değeri olarak ON(AÇIK) değeri kullanılır.

## İlgili görevler

[mqat.ini ile etkinlik izleme davranışını yapılandırma](#)



## Dağıtılmış kuyruğa alma yapılandırılıyor



Bu bölümde, kuyruk tanımlaması, kanal tanımlaması, tetikleme ve eşitleme noktası yordamları da içinde olmak üzere IBM MQ kuruluşları arasındaki iletişime ilgili daha ayrıntılı bilgi sağlanır.

### Başlamadan önce

Bu bölümü okumadan önce, kanalları, kuyrukları ve [Dağıtılmış kuyruğa alma ve kümeler](#)' de tanıtılan diğer kavramları anlamanız yararlı olur.

Farklı fiziksel ağlarda bulunan iki kuyruk yöneticisini bağlamanız gerekiyorsa ya da bir güvenlik duvarı üzerinden iletişim kurmanız gerekiyorsa, IBM MQ Internet Pass-Thru komutunu kullanarak yapılandırmayı basitleştirebilirsiniz. Daha fazla bilgi için bkz. [IBM MQ Internet Pass-Thru](#).

### Yordam

- Dağıtılmış kuyruğa alma özelliğini kullanarak uygulamalarınızı bağlamak için aşağıdaki alt konulardaki bilgileri kullanın:
  - [“IBM MQ dağıtılmış kuyruğa alma teknikleri” sayfa 189](#)
  - [“Dağıtılmış kuyruk yönetimine giriş” sayfa 209](#)
  - [“Başka bir kuyruk yöneticisine ileti gönderilmesi” sayfa 211](#)
  - [“Tetikleme kanalları” sayfa 232](#)
  - [“İletilerin güvenliği” sayfa 229](#)
  -  [“AIX, Linux, and Windows üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 239](#)
  -  [“IBM i üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 262](#)

### İlgili kavramlar

[“ayarlarIBM MQ for z/OS” sayfa 881](#)

IBM MQ for z/OS sisteminizi uyarlamak için adım adım bu konuyu kullanın.

### İlgili görevler

[“İstemci ile sunucu arasındaki bağlantıların yapılandırılması” sayfa 14](#)

IBM MQ MQI clients ile sunucular arasındaki iletişim bağlantılarını yapılandırmak için iletişim protokolünüze karar verin, bağlantının her iki ucundaki bağlantıları tanımlayın, bir dinleyici başlatın ve kanalları tanımlayın.

[“Kuyruk yöneticisi kümesinin yapılandırılması” sayfa 283](#)

Kümeler, hem ilk yapılandırmayı hem de devam eden yönetimi basitleştiren bir şekilde birbirine bağlanan kuyruk yöneticileri için bir mekanizma sağlar. Küme bileşenleri tanımlayabilir ve kümeleri oluşturabilir ve yönetebilirsiniz.

[“Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerini değiştirme” sayfa 83](#)

Yapılandırma (.ini) dosyalarındaki bilgileri düzenleyerek, IBM MQ ya da tek bir kuyruk yöneticisinin davranışını kuruluşunuzun gereksinimlerine uyacak şekilde değiştirebilirsiniz. IBM MQ MQI clients için yapılandırma seçeneklerini de değiştirebilirsiniz.

[“z/OS üzerinde kuyruk yöneticilerini yapılandırma” sayfa 876](#)

IBM MQ for z/OS üzerinde kuyruk yöneticilerini yapılandırmak için bu yönergeleri kullanın.





[“z/OS üzerinde diğer kuyruk yöneticileriyle iletişim kurulması” sayfa 954](#)

Bu bölümde, dağıtılmış kuyruğa alma işlevini kullanmaya başlamadan önce yapmanız gereken IBM MQ for z/OS hazırlıkları açıklanmaktadır.

## IBM MQ dağıtılmış kuyruğa alma teknikleri

Bu bölümdeki alt konularda, kanalları planlarken kullanılan teknikler açıklanmaktadır. Bu alt konularda, kuyruk yöneticilerinizi birbirine nasıl bağlayacağınızı planlamanıza ve uygulamalarınız arasındaki ileti akışını yönetmenize yardımcı olacak teknikler açıklanmaktadır.

İleti kanalı planlama örnekleri için bkz:

-  [AIX, Linux, and Windows için ileti kanalı planlama örneği](#)
-  [IBM için ileti kanalı planlama örneği](#)
-  [z/OS için ileti kanalı planlama örneği](#)
-  [z/OS kuyruk paylaşım gruplarını kullanma için ileti kanalı planlama örneği](#)

## İlgili kavramlar

[Kanallar](#)

[İleti kuyruklama girişine giriş](#)

[Dağıtılmış kuyruğa alma ve kümeler](#)

## İlgili görevler

[“Dağıtılmış kuyruğa alma yapılandırılıyor” sayfa 189](#)


Bu bölümde, kuyruk tanımlaması, kanal tanımlaması, tetikleme ve eşitleme noktası yordamları da içinde olmak üzere IBM MQ kuruluşları arasındaki iletişikle ilgili daha ayrıntılı bilgi sağlanır.

## İlgili başvurular

[Örnek yapılandırma bilgileri](#)

## İleti akışı denetimi

İleti akışı denetimi, kuyruk yöneticileri arasındaki ileti rotalarının ayarlanmasını ve bakımını içeren bir görevdir. Birçok kuyruk yöneticisi üzerinden birden çok sıçrayan rotalar için önemlidir. Bu bölümde, ileti akışı denetimine ulaşmak için sisteminizdeki kuyrukları, diğer ad kuyruğu tanımlamalarını ve ileti kanallarını nasıl kullandığınız açıklanmaktadır.

İleti akışını, [“Dağıtılmış kuyruğa alma yapılandırılıyor” sayfa 189](#) içinde tanımlanan birkaç tekniği kullanarak denetleyebilirsiniz. Kuyruk yöneticiniz bir kümedeyse, ileti akışı [“İleti akışı denetimi” sayfa 190](#) içinde açıkladığı gibi farklı teknikler kullanılarak denetlenir.  Kuyruk yöneticileriniz bir kuyruk paylaşım grubundaydı ve grup içi kuyruğa alma (IGQ) etkinleştirildiyse, ileti akışı IGQ araçları tarafından denetlenebilir. Bu araçlar [Grup içi kuyruğa alma](#) konusunda açıklanmıştır.

İleti akışı denetimini gerçekleştirmek için aşağıdaki nesnelere kullanabilirsiniz:

- İletim kuyrukları
- İleti kanalları
- Uzak kuyruk tanımlaması
- Kuyruk yöneticisi diğer adı tanımlaması
- Yanıtın gönderileceği kuyruk diğer adı tanımlaması

Kuyruk yöneticisi ve kuyruk nesnelere Nesne tipleri içinde açıklanır. İleti kanalları [Dağıtılmış kuyruğa alma bileşenleri](#) içinde açıklanır. Aşağıdaki teknikler, sisteminizde ileti akışları yaratmak için bu nesnelere kullanır:

- Uzak kuyruklara ileti konması
- Belirli iletim kuyrukları yoluyla yönlendirme
- İleti alınması
- İletilerin sisteminizden geçirilmesi
- İleti akışlarını ayırma
- İleti akışının başka bir hedefe değiştirilmesi
- Yanıt kuyruğu adının bir diğer ada çözülmesi

## Not

Bu bölümde açıklanan tüm kavramlar, bir ağdaki tüm düğümler için geçerlidir ve ileti kanallarının gönderme ve alma uçlarını içerir. Bu nedenle, çoğu örnekte yalnızca bir düğüm gösterilir. Kural dışı durum, örneğin bir ileti kanalının diğer ucunda yönetici tarafından açık bir işbirliği gerektirdiğidir.

Bireysel tekniklere geçmeden önce, ad çözünürlüğü kavramlarını ve uzak kuyruk tanımlarını kullanmanın üç yolunu yeniden ele almak yararlıdır. Bkz. [Dağıtılmış kuyruğa alma ve kümeler](#).

### İlgili kavramlar

[“İletim üstbilgisindeki kuyruk adları” sayfa 191](#)

Hedef kuyruk adları, hedef kuyruğa ulaşıncaya kadar iletim üstbilgisindeki iletiyle birlikte hareket eder.

[“Kuyruk yöneticisi ve yanıtlama diğer adları yaratılması” sayfa 191](#)

Bu konuda, uzak kuyruk tanımı yaratmanın üç yolu açıklanmaktadır.

### **İletim üstbilgisindeki kuyruk adları**

Hedef kuyruk adları, hedef kuyruğa ulaşıncaya kadar iletim üstbilgisindeki iletiyle birlikte hareket eder.

Uygulama tarafından kullanılan kuyruk adı (mantıksal kuyruk adı), kuyruk yöneticisi tarafından hedef kuyruk adına çözülür. Başka bir deyişle, fiziksel kuyruk adı. Bu hedef kuyruk adı, hedef kuyruğa ulaşıncaya kadar iletiyle birlikte ayrı bir veri alanında, iletim üstbilgisinde seyahat eder. Daha sonra iletim başlığı soyularak çıkarılmış.

Koşut hizmet sınıfları yarattığınızda, bu kuyruk adının kuyruk yöneticisi kısmını değiştirmeniz gerekir. Hizmet sınıfı sapması sonuna ulaşıldığında kuyruk yöneticisi adını özgün ada döndürmeyi unutmayın.

### **Kuyruk yöneticisi ve yanıtlama diğer adları yaratılması**

Bu konuda, uzak kuyruk tanımı yaratmanın üç yolu açıklanmaktadır.

Uzak kuyruk tanımlaması nesnesi üç farklı şekilde kullanılır. [Çizelge 16 sayfa 192](#) , aşağıdaki üç yöntemden her birinin nasıl tanımlanacağını açıklar:

- Yerel kuyruk adını yeniden tanımlamak için uzak kuyruk tanımı kullanılıyor.

Uygulama yalnızca bir kuyruk açarken kuyruk adını sağlar ve bu kuyruk adı uzak kuyruk tanımlamasının adıdır.

Uzak kuyruk tanımlaması, hedef kuyruk ve kuyruk yöneticisinin adlarını içerir. İsteğe bağlı olarak, tanım kullanılacak iletim kuyruğunun adını içerebilir. İletim kuyruğu adı sağlanmazsa, kuyruk yöneticisi, iletim kuyruğu adı için uzak kuyruk tanımından alınan kuyruk yöneticisi adını kullanır. Bu adda bir iletim kuyruğu tanımlanmamışsa, ancak varsayılan bir iletim kuyruğu tanımlanmamışsa, varsayılan iletim kuyruğu kullanılır.

- Bir kuyruk yöneticisi adını yeniden tanımlamak için uzak kuyruk tanımlaması kullanılıyor.

Uygulama ya da kanal programı, kuyruğu açarken uzak kuyruk yöneticisi adıyla birlikte bir kuyruk adı sağlar.

Kuyruk yöneticisi adıyla aynı adı taşıyan bir uzak kuyruk tanımlaması belirtirdiyseniz ve tanımlamada kuyruk adını boş bıraktıysanız, kuyruk yöneticisi, açık çağrıdaki kuyruk yöneticisi adını tanımdaki kuyruk yöneticisi adıyla değiştirir.

Ayrıca, tanım kullanılacak iletim kuyruğunun adını da içerebilir. İletim kuyruğu adı sağlanmazsa, kuyruk yöneticisi, iletim kuyruğu adı için uzak kuyruk tanımından alınan kuyruk yöneticisi adını alır. Bu adda bir iletim kuyruğu tanımlanmamışsa, ancak varsayılan bir iletim kuyruğu tanımlanmamışsa, varsayılan iletim kuyruğu kullanılır.

- Yanıtlanacak kuyruk adını yeniden tanımlamak için uzak kuyruk tanımı kullanılıyor.

Bir uygulama bir iletiyi kuyruğa her yerleştirdiğinde, yanıt iletileri için yanıt kuyruğunun adını belirleyebilir, ancak kuyruk yöneticisi adı boş olabilir.

Yanıt kuyruğuyla aynı adı taşıyan bir uzak kuyruk tanımlaması sağlarsanız, yerel kuyruk yöneticisi, yanıt kuyruğu adını tanımlamanızdaki kuyruk adıyla değiştirir.

Tanımda bir kuyruk yöneticisi adı sağlayabilirsiniz, ancak bir iletim kuyruğu adı vermezsiniz.

Çizelge 16. Uzak kuyruk tanımlaması nesnesini kullanmanın üç yolu			
Kullanım	Kuyruk yöneticisi adı	Kuyruk adı	İletim kuyruğu adı
1. Uzak kuyruk tanımı (OPEN çağrısında)			
Aramada sağlandı	boş ya da yerel QM	(*) gerekli	geçerli değil
Tanımlamada sağlandı	gerekli	gerekli	isteğe bağlı
2. Kuyruk yöneticisi diğer adı (OPEN çağrısında)			
Aramada sağlandı	(*) gerekli ve yerel QM değil	gerekli	geçerli değil
Tanımlamada sağlandı	gerekli	blank (boşluk)	isteğe bağlı
3. Yanıt kuyruğu diğer adı (PUT çağrısında)			
Aramada sağlandı	blank (boşluk)	(*) gerekli	geçerli değil
Tanımlamada sağlandı	isteğe bağlı	isteğe bağlı	blank (boşluk)

**Not:** (\*), bu adın tanımlama nesnesinin adı olduğu anlamına gelir

Resmi bir açıklama için bkz. [Kuyruk adı çözünürlüğü](#).

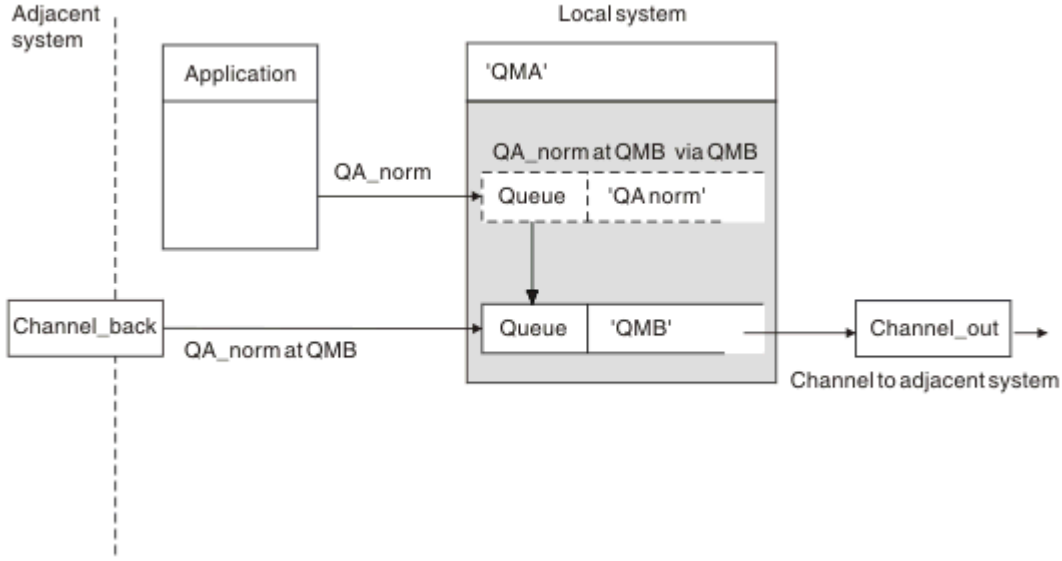
## Uzak kuyruklara ileti konulması

Uzak kuyruk tanımı nesnelere, bir kuyruk adını bitişik kuyruk yöneticisine bir iletim kuyruğuna çözümlenmek için kullanabilirsiniz.

Dağıtılmış kuyruğa alma ortamında, iletim kuyruğu ve kanal, iletilerin yerel sisteminizdeki uygulamalardan kaynaklanmasından ya da bitişik bir sistemden kanallar üzerinden gelmesinden bağımsız olarak, bir konuma gönderilen tüm iletilerin odak noktasıdır. Şekil 6 sayfa 193 içinde, 'QA\_norm' adlı mantıksal kuyruğa ileti yerleştiren bir uygulama gösterilmektedir. Ad çözünürlüğü, QMB iletim kuyruğunu seçmek için 'QA\_norm' uzak kuyruk tanımını kullanır. Daha sonra 'QMB' de QA\_norm ' belirten iletilere bir iletim üstbilgisi ekler.

'Channel\_back' üzerindeki bitişik sistemden gelen iletiler, örneğin, QMB 'deki' QA\_norm ' fiziksel kuyruk adına sahip bir iletim üstbilgisine sahiptir. Bu iletiler, QMB iletim kuyruğuna değiştirilmeden yerleştirilir.

Kanal, iletileri bitişik bir kuyruk yöneticisine taşır.



Şekil 6. Uzak kuyruk tanımı, bir kuyruk adını bitişik kuyruk yöneticisine bir iletim kuyruğuna çözmek için kullanılır.

IBM MQ sistem yöneticisiyseniz, aşağıdakileri gerçekleştirmeniz gerekir:

- Bitişik sistemden ileti kanalının tanımlanması
- Bitişik sisteme ilişkin ileti kanalının tanımlanması
- QMB iletim kuyruğunun yaratılması
- Uygulamalar tarafından kullanılan kuyruk adını hedef kuyruk adına, hedef kuyruk yöneticisi adına ve iletim kuyruğu adına çözmek için 'QA\_norm' uzak kuyruk nesnesini tanımlayın

Kümeleme ortamında, yalnızca yerel kuyruk yöneticisinde bir küme alıcı kanalı tanımlamanız gerekir. Bir iletim kuyruğu ya da uzak kuyruk nesnesi tanımlamanıza gerek yoktur. Bkz. [Kümeler](#).

## Ad çözümlemesi hakkında daha fazla bilgi

Uzak kuyruk tanımlamasının etkisi, fiziksel bir hedef kuyruk adı ve kuyruk yöneticisi adı tanımlanmasıdır. Bu adlar, iletilerin iletim üstbilgilerine konmuştur.

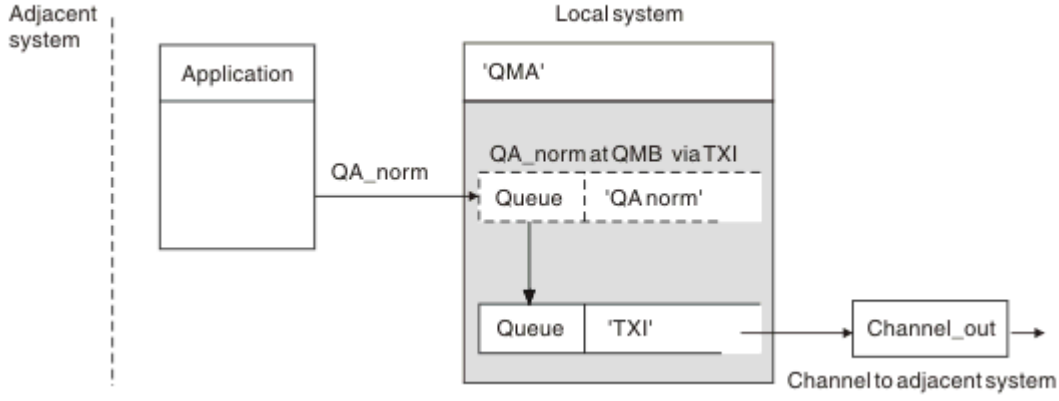
Bitişik bir sistemden gelen iletiler, özgün kuyruk yöneticisi tarafından bu tip bir ad çözme işlemi gerçekleştirilmiş. Bu nedenle, fiziksel hedef kuyruk adını ve kuyruk yöneticisi adını gösteren iletim üstbilgisine sahiptir. Bu iletiler uzak kuyruk tanımlamalarından etkilenmez.

### İlgili başvurular

[Kuyruk adı çözümlemesi](#)

## İletim kuyruğunun seçilmesi

Uzak kuyruk tanımını, farklı bir iletim kuyruğunun aynı bitişik kuyruk yöneticisine ileti göndermesine izin vermek için kullanabilirsiniz.



Şekil 7. Uzak kuyruk tanımı, farklı bir iletim kuyruğunun kullanılmasına izin verir

Dağıtılmış kuyruğa alma ortamında, bir kanaldan diğerine ileti akışını değiştirmeniz gerekirse, “Uzak kuyruklara ileti konulması” sayfa 192’de Şekil 6 sayfa 193’te gösterilen aynı sistem yapılandırmasını kullanın. Bu konudaki Şekil 7 sayfa 194’te, uzak kuyruk tanımının farklı bir iletim kuyruğu üzerinden ve dolayısıyla farklı bir kanal üzerinden aynı bitişik kuyruk yöneticisine ileti göndermek için nasıl kullanılacağı gösterilmektedir.

Şekil 7 sayfa 194’te gösterilen yapılandırma için 'QA\_norm' uzak kuyruk nesnesini ve 'TX1' iletim kuyruğunu sağlamanız gerekir. Uzak kuyruk yöneticisinde 'QA\_norm' kuyruğunu, 'TX1' iletim kuyruğunu ve 'QMB\_priority' kuyruk yöneticisini seçmek için 'QA\_norm' sağlamanız gerekir. Sistemin yanındaki kanal tanımında 'TX1' değerini belirtin.

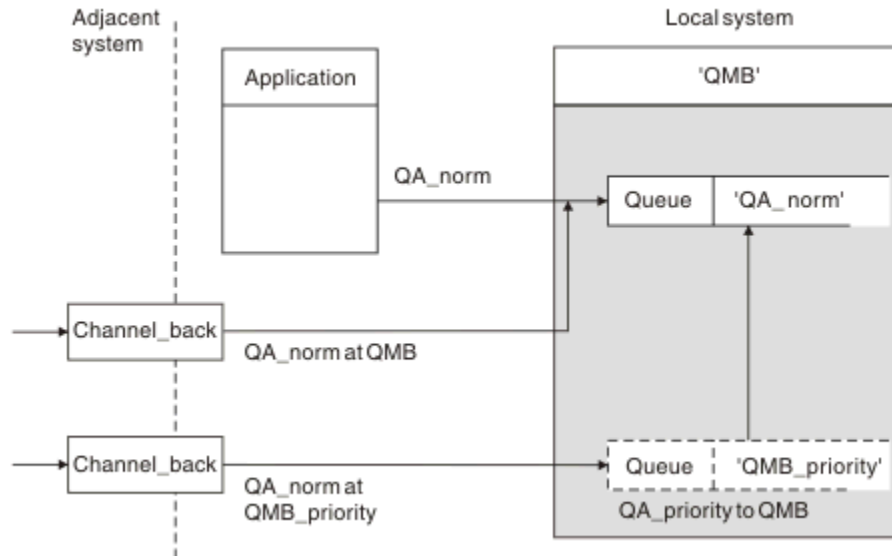
İletiler, 'TX1' iletim kuyruğuna 'QMB\_priority' ile 'QA\_norm' iletim üstbilgisiyle yerleştirilir ve kanal üzerinden bitişik sisteme gönderilir.

channel\_back, bir kuyruk yöneticisi diğer adı gerekeceği için bu şekilde bırakılmıştır.

Kümeleme ortamında, bir iletim kuyruğu ya da uzak kuyruk tanımı tanımlamanıza gerek yoktur. Daha fazla bilgi için “Küme kuyruklarının tanımlanması” sayfa 284 başlıklı konuya bakın.

## İleti alınması

Kuyruk yöneticisini, diğer kuyruk yöneticilerinden ileti alacak şekilde yapılandırabilirsiniz. Kasıtsız ad çözümlemesinin gerçekleşmediğinden emin olmalısınız.



Şekil 8. İletilerin doğrudan alınması ve diğer ad kuyruk yöneticisi adı çözümleniyor

İletilerin gönderilmesini ayarlamanın yanı sıra, sistem denetimcisinin bitişik kuyruk yöneticilerinden ileti alınması için de düzenleme gerçekleştirilmesi gerekir. Alınan iletiler, iletim üstbilgisindeki hedef kuyruk yöneticisinin ve kuyruğun fiziksel adını içerir. Bunlar, hem kuyruk yöneticisi adını hem de kuyruk adını belirten yerel bir uygulamadaki iletilerle aynı şekilde işlenir. Bu işlem nedeniyle, sisteminize giren iletilerin kasıtlı olmayan bir ad çözümüne sahip olmadığından emin olmanız gerekir. Bu senaryo için bkz. [Şekil 8 sayfa 194](#) .

Bu yapılandırma için aşağıdakileri hazırlamanız gerekir:

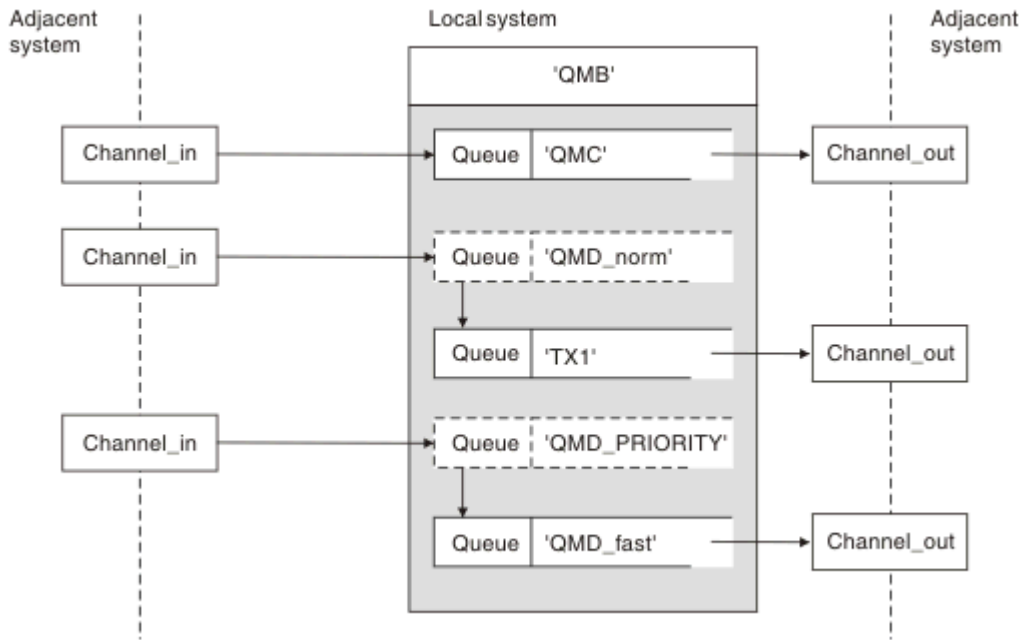
- Bitişik kuyruk yöneticilerinden ileti alacak ileti kanalları
- Yerel kuyruk yöneticisi adına ('QMB') gelen ileti akışını ('QMB\_priority') çözmek için bir kuyruk yöneticisi diğer adı tanımlaması
- Yerel kuyruk yoksa, 'QA\_norm'

## Diğer ad kuyruk yöneticisi adları alınıyor

Bu resimde kuyruk yöneticisi diğer adı tanımlamasının kullanılması farklı bir hedef kuyruk yöneticisi seçmedi. Bu yerel kuyruk yöneticisinden geçen ve 'QMB\_priority' değerine adreslenen iletiler, 'QMB' kuyruk yöneticisine yöneliktir. Diğer ad kuyruk yöneticisi adı, ayrı ileti akışını yaratmak için kullanılır.

## İletilerin sisteminizden geçirilmesi

İletileri, konum adını kullanarak, kuyruk yöneticisi için bir diğer ad kullanarak ya da bir iletim kuyruğu seçerek sisteminizden üç şekilde geçirebilirsiniz.



Şekil 9. İletileri sisteminizden geçirmenin üç yöntemi

“İleti alınması” sayfa 194’ünde Şekil 8 sayfa 194 içinde gösterilen teknik, bir diğer ad akışının nasıl yakalandığını gösterdi. Şekil 9 sayfa 195 içinde daha önce açıklanan teknikleri bir araya getirerek ağların nasıl oluşturulduğu gösterilmektedir.

Yapılandırma, farklı hedeflere sahip üç ileti teslim eden bir kanalı gösterir:

1. QB saat: QMC
2. QB saat: QMD\_norm
3. QB saat: QMD\_PRIORITY

İlk ileti akışını sisteminizden değiştirmeden geçirmeniz gerekir. İkinci ileti akışını farklı bir iletim kuyruğu ve kanalından geçirmeniz gerekir. İkinci ileti akışı için, QMD\_noım diğer ad kuyruk yöneticisi adına ilişkin iletileri QMDkuyruk yöneticisine çözeniz gerekir. Üçüncü ileti akışı, başka bir değişiklik olmadan farklı bir iletim kuyruğu seçer.

Kümeleme ortamında, iletiler küme iletim kuyruğundan geçirilir. Normalde tek bir iletim kuyruğu, SYSTEM. CLUSTER. TRANSMIT. QUEUE, kuyruk yöneticisinin üyesi olduğu tüm kümelerdeki tüm kuyruk yöneticilerine tüm iletileri aktarır; bkz. *Kuyruk yöneticileri kümesi*. Kuyruk yöneticisinin üyesi olduğu kümelerdeki kuyruk yöneticilerinin tümü ya da bazıları için ayrı iletim kuyrukları tanımlayabilirsiniz.

Aşağıdaki yöntemler, dağıtılmış kuyruğa alma ortamı için geçerli olan teknikleri açıklar.

## Bu yöntemleri kullan

Bu yapılandırmalar için aşağıdakileri hazırlamanız gerekir:

- Giriş kanalı tanımları
- Çıkış kanalı tanımları
- İletim kuyrukları:
  - QMC
  - TX1
  - QMD\_fast
- Kuyruk yöneticisi diğer ad tanımlamaları:
  - QMD\_noım with QMD\_noım to QMD to TX1 (ile üzerinden)
  - QMD\_PRIORITY with QMD\_PRIORITY to QMD\_PRIORITY to QMD\_fast (ile üzerinden)

**Not:** Örnekte gösterilen ileti akışlarının hiçbiri hedef kuyruğu değiştirmez. Kuyruk yöneticisi adı diğer adları, ileti akışlarının ayrılmasını sağlar.

## Yöntem 1: Gelen konum adını kullan

QMCgibi başka bir konum adı içeren bir iletim üstbilgisine sahip iletiler alacaksınız. En basit yapılandırma, QMCadıyla bir iletim kuyruğu oluşturmaktır. İletim kuyruğuna hizmet veren kanal, iletiyi bir sonraki hedefe değiştirmeden teslim eder.

## Yöntem 2: Kuyruk yöneticisi için diğer ad kullan

İkinci yöntem, kuyruk yöneticisi diğer ad nesnesi tanımını kullanmak, ancak yeni bir yer adı ( QMD) ve belirli bir iletim kuyruğu ( TX1) belirlemektir. Bu eylem:

- Kuyruk yöneticisi diğer adı QMD\_noım(yani, adı belirtilen hizmet sınıfı QMD\_noım) tarafından ayarlanan diğer ad ileti akışını sonlandırır.
- Bu iletilerdeki iletim üstbilgilerini QMD\_noım 'den QMD' a değiştirir.

## Yöntem 3: Bir iletim kuyruğu seçin

Üçüncü yöntem, hedef konumla ( QMD\_PRIORITY) aynı adla tanımlanmış bir kuyruk yöneticisi diğer adı nesnesine sahip olmaktır. Belirli bir iletim kuyruğunu ( QMD\_fast) ve dolayısıyla başka bir kanalı seçmek için kuyruk yöneticisi diğer ad tanımlamasını kullanın. Bu iletilerdeki iletim üstbilgileri değişmeden kalır.

## İleti akışlarını ayırma

Aynı kuyruk yöneticisine ileti göndermek üzere ayrı ileti akışları yaratmak için bir kuyruk yöneticisi diğer adını kullanabilirsiniz.

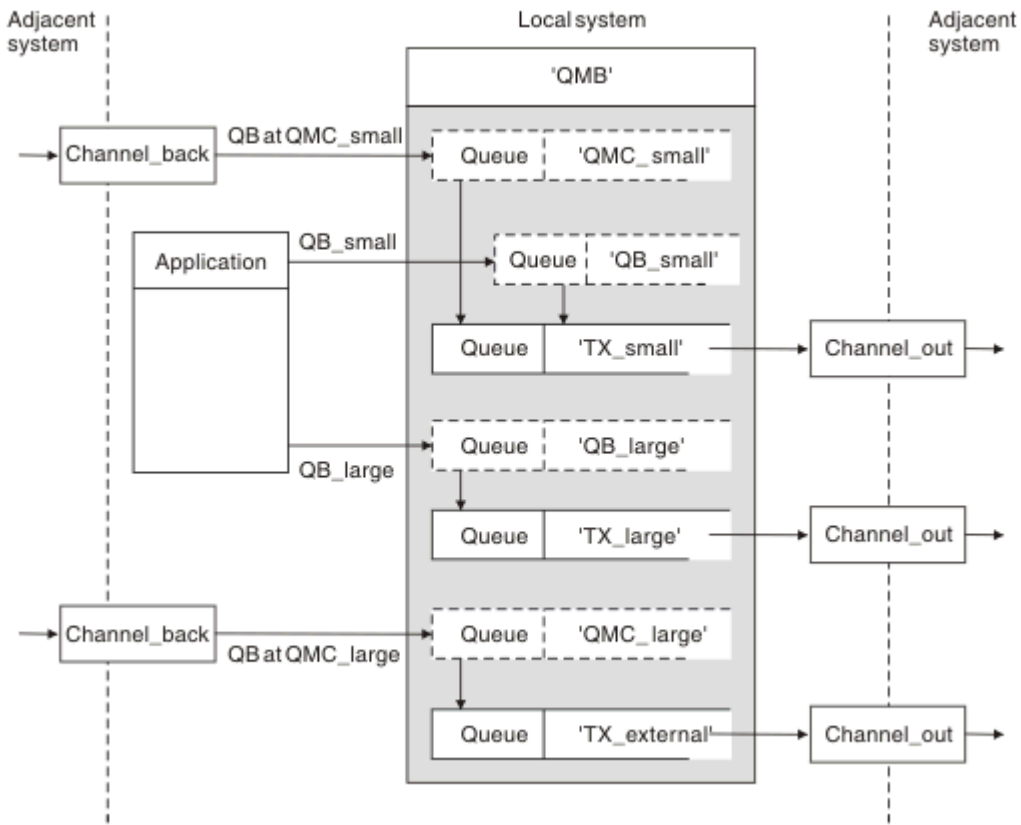


## İletileri farklı ileti akışlarına ayırma nedenleri

Dağıtılmış kuyruğa alma ortamında, iletileri aynı kuyruk yöneticisine farklı ileti akışlarına ayırma gereksinimi çeşitli nedenlerden kaynaklanabilir. Örneğin:

- Büyük, orta ve küçük iletiler için ayrı bir akış sağlamanız gerekebilir. Bu gereksinim bir kümeleme ortamında da geçerlidir ve bu durumda çakışan kümeler oluşturabilirsiniz. Bunu yapabilmemiz için çeşitli nedenler vardır; örneğin:
  - Farklı kuruluşların kendi yönetimine sahip olmasına izin vermek için.
  - Bağımsız uygulamaların ayrı olarak yönetilmesine izin vermek için.
  - Hizmet sınıfı yaratmak için. Örneğin, STAFF adlı bir kümeniz olabilir; bu küme, "ÖĞRENCİLER" adlı kümenin bir altkümesidir. STAFF kümesinde duyurulan bir kuyruğa bir ileti koyduğunuzda, sınırlı bir kanal kullanılır. Öğrenci kümesinde ilan edilen bir kuyruğa bir mesaj koyduğunuzda, genel bir kanal ya da sınırlı bir kanal kullanılabilir.
  - Test ve üretim ortamlarının yaratılması.
- Gelen iletileri, yerel olarak oluşturulan iletilerin yolundan farklı yollarla yönlendirmek gerekebilir.
- Kuruluşunuz, iletilerin belirli zamanlarda (örneğin, bir gecede) taşınmasını zamanlamasını gerektirebilir ve daha sonra, iletilerin zamanlanıncaya kadar ayrılmış kuyruklarda saklanması gerekebilir.

## Örnek ileti akışı



Şekil 10. İleti akışlarını ayırma

Şekil 10 sayfa 197’inde gösterilen örnekte, gelen iki akış 'QMC\_small' ve 'QMC\_large' diğer ad kuyruk yöneticisi adlarını içerir. Bu akışları, yerel kuyruk yöneticisi için bu akışları yakalamak üzere bir kuyruk yöneticisi diğer adı tanımlamasıyla sağlarsınız. İki uzak kuyruğu ele alan bir uygulamanız var ve bu ileti akışlarının ayrı tutulması gerekiyor. Aynı yeri ('QMC') belirleyen, ancak farklı iletim kuyrukları belirleyen iki uzak kuyruk tanımları belirlersiniz. Bu tanımlama, akışları ayrı tutar ve iletim üstbilgilerinde aynı hedef kuyruk yöneticisi adına sahip oldukları için uzak uçta ek bir işlem gerekmez. Aşağıdakileri sağlarlar:

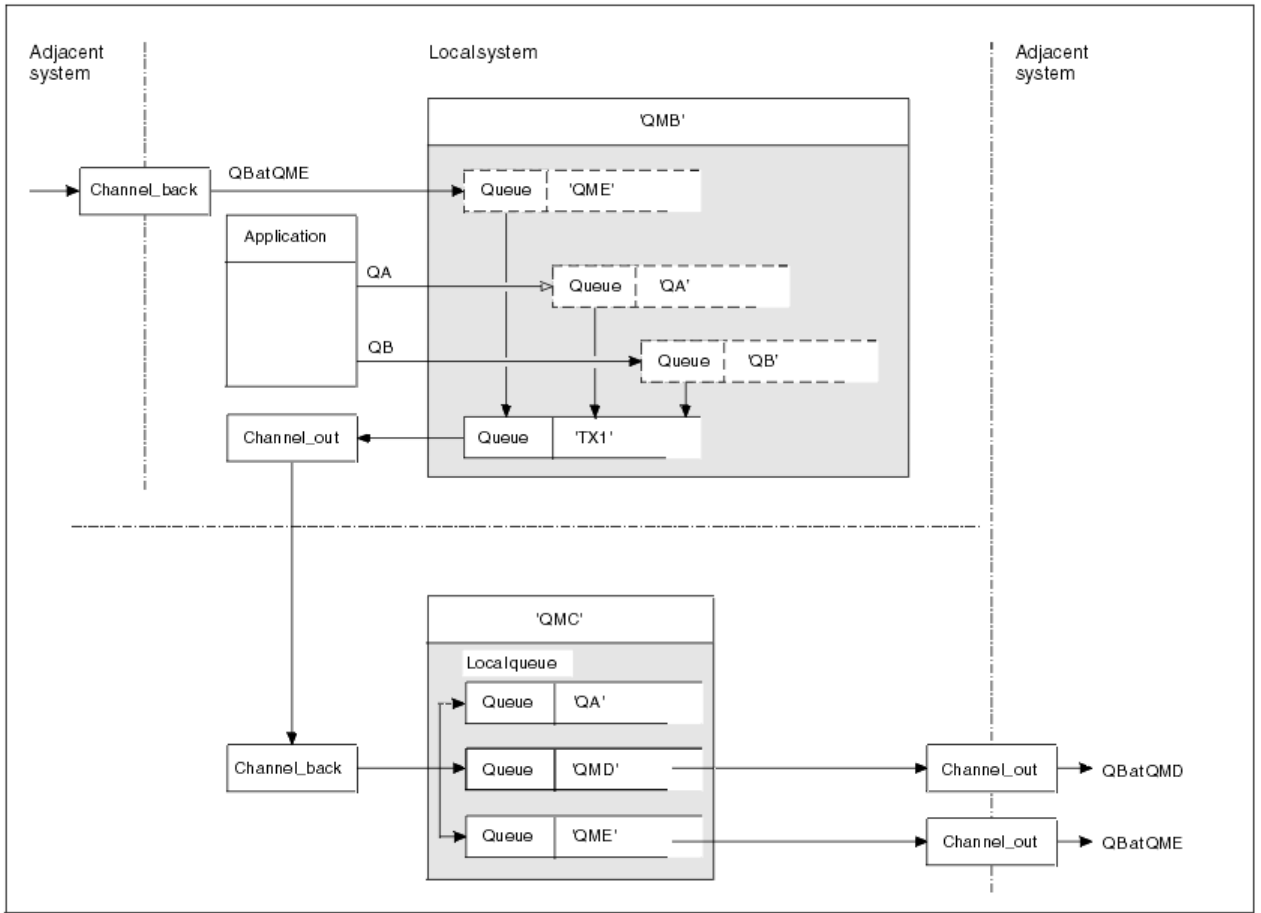
- Gelen kanal tanımları
- İki uzak kuyruk tanımları QB\_small ve QB\_large
- İki kuyruk yöneticisi diğer adı tanımları QMC\_small ve QMC\_large
- Gönderen üç kanal tanımları
- Üç iletim kuyruğu: TX\_small, TX\_large ve TX\_external

## Bitişik sistemlerle koordinasyon

Ayrı bir ileti akışı oluşturmak için bir kuyruk yöneticisi diğer adı kullandığınızda, ilgili kuyruk yöneticisi diğer adının kullanılabilir olduğundan emin olmak için bu etkinliği ileti kanalının uzak ucundaki sistem yöneticisiyle eşgüdüm sağlamanız gerekir.

## İletilerin çeşitli konumlarda yoğunlaştırılması

Tek bir kanalda çeşitli konumlara yönlendirilmiş iletileri odaklayabilirsiniz.



Şekil 11. İleti akışlarının bir kanalda birleştirilmesi

Şekil 11 sayfa 198 içinde, tek bir kanalda çeşitli konumlara yönlendirilmiş iletilerin yoğunlaştırılması için dağıtılmış kuyruğa alma tekniği gösterilmektedir. İki olası kullanım şöyle olabilir:

- Ağ geçidi üzerinden ileti trafiğinin yoğunlaştırılması
- Döğümler arasında geniş bant genişliği otoyollarını kullanma

Bu örnekte, farklı kaynaklardan, yerel ve bitişik ve farklı hedef kuyruklarına ve kuyruk yöneticilerine sahip iletiler, 'TX1' iletim kuyruğundan QMC kuyruk yöneticisine aktarılır. Kuyruk yöneticisi QMC, iletileri hedeflere göre teslim eder. Kuyruk yöneticisi QMD 'ye iletimi için bir iletim kuyruğu' QMD 'ye ayarlandı.

QME kuyruk yöneticisine iletimi için başka bir 'QME' iletim kuyruğu kümesi. Diğer iletiler 'QA' yerel kuyruğuna kondu.

Aşağıdakileri sağlamanız gerekir:

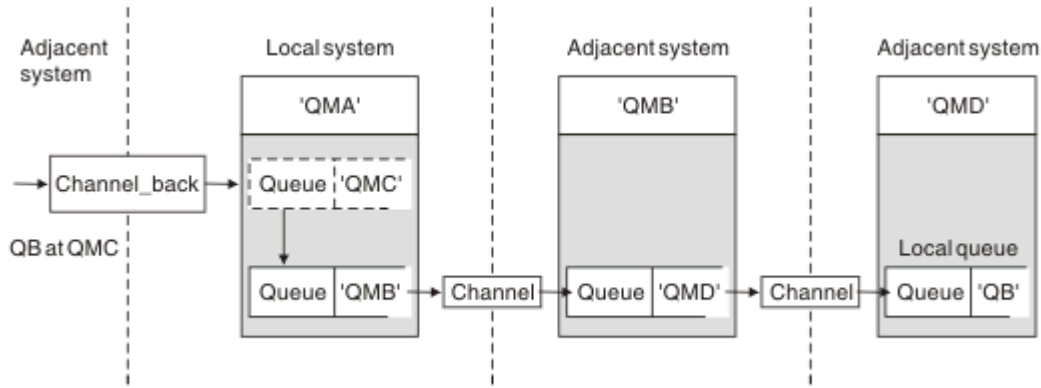
- Kanal tanımları
- İletim kuyruğu TX1
- Uzak kuyruk tanımlamaları:
  - TX1'aracılığıyla QMC' de 'QA' ile QA
  - TX1'aracılığıyla QMD' de 'QB' ile QB
- Kuyruk yöneticisi diğer adı tanımlaması:
  - 'QME ile QME- TX1'

QMC ' yi yapılandırabilen tamamlayıcı yönetici aşağıdakileri sağlamalıdır:

- Aynı kanal adıyla kanal tanımı alınıyor
- İlişkili gönderen kanal tanımına sahip iletim kuyruğu QMD
- İlişkili gönderen kanal tanımına sahip iletim kuyruğu QME
- Yerel kuyruk nesnesi QA.

## İleti akışlarının başka bir hedefe yönlendirilmesi

Kuyruk yöneticisi diğer adlarını ve iletim kuyruklarını kullanarak belirli iletilerin hedefini yeniden tanımlayabilirsiniz.



Şekil 12. İleti akışlarını başka bir hedefe yönlendirme

Şekil 12 sayfa 199 içinde belirli iletilerin hedefini nasıl yeniden tanımlayabileceğiniz gösterilir. QMA 'ya gelen iletiler' QMC 'de QB' ye yönlendirildi. Genellikle QMA 'ya ulaşırlar ve QMC' ye bir kanalın parçası olan QMC adlı bir iletim kuyruğuna yerleştirilirler. QMA, mesajları QMD 'ye yönlendirmelidir, ancak QMB üzerinden QMD' ye ulaşabilir. Bu yöntem, bir hizmeti bir konumdan diğerine taşımanız ve abonelerin yeni adrese ayarlanıncaya kadar geçici olarak ileti göndermeye devam etmelerine izin vermeniz gerektiğinde yararlıdır.

Belirli bir kuyruk yöneticisine yönelik gelen iletileri farklı bir kuyruk yöneticisine yeniden yönlendirme yöntemi şunları kullanır:

- Hedef kuyruk yöneticisini başka bir kuyruk yöneticisine değiştirmek ve bitişik sisteme bir iletim kuyruğu seçmek için bir kuyruk yöneticisi diğer adı
- Bitişik kuyruk yöneticisine hizmet vermek için bir iletim kuyruğu
- Hedef kuyruk yöneticisine yönlendirmek için bitişik kuyruk yöneticisinde bir iletim kuyruğu

Aşağıdakileri sağlamanız gerekir:

- Channel\_back tanımlaması

- QMD-QMB arasında QB ile QMC kuyruk yöneticisi diğer adı nesne tanımı
- Kanal\_çıkış tanımlaması
- İlişkili iletim kuyruğu QMB

QMB 'yi yapılandırabilen tamamlayıcı yönetici aşağıdakileri sağlamalıdır:

- İlgili channel\_back tanımlaması
- İletim kuyruğu, QMD
- QMD ile ilişkili kanal tanımı

Kümeleme ortamında diğer adları kullanabilirsiniz. Bilgi için bkz. [“Kuyruk yöneticisi diğer adları ve kümeleri” sayfa 377.](#)

## Dağıtım listesine ileti gönderilmesi

Bir uygulamanın çeşitli hedeflere ileti göndermesini sağlamak için tek bir MQPUT çağrısı kullanabilirsiniz.

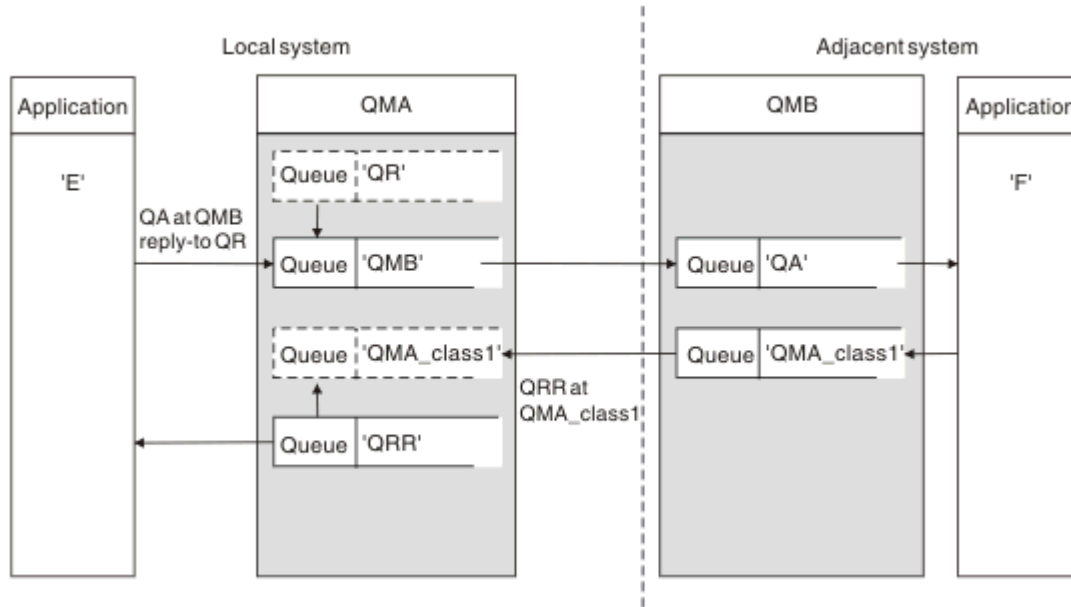
IBM MQ içinde z/OS dışındaki tüm platformlarda, bir uygulama tek bir MQPUT çağrısıyla birden çok hedefe ileti gönderebilir. Bunu hem dağıtılmış kuyruğa alma ortamında hem de kümeleme ortamında yapabilirsiniz. [Dağıtım listeleri](#)' de açıklandığı gibi, bir dağıtım listesindeki hedefleri tanımlamanız gerekir.

Tüm kuyruk yöneticileri dağıtım listelerini desteklemez. Bir MCA bir iş ortağıyla bağlantı kurduğunda, ortağın dağıtım listelerini destekleyip desteklemediğini belirler ve iletim kuyruğunda buna göre bir işaret belirler. Bir uygulama dağıtım listesine gönderilecek bir ileti göndermeye çalışırsa, ancak ortak dağıtım listelerini desteklemiyorsa, gönderen MCA iletiyi yakalar ve istenen her hedef için iletim kuyruğuna bir kez koyar.

Alıcı MCA, dağıtım listesine gönderilen iletilerin tüm hedeflerde güvenli bir şekilde alınmasını sağlar. Herhangi bir hedef başarısız olursa, MCA hangilerinin başarısız olduğu belirler. Daha sonra bunlar için kural dışı durum raporları oluşturabilir ve iletileri onlara yeniden göndermeyi deneyebilir.

## Yanıtın gönderileceği kuyruk

Bir yanıtla kuyruğunu kullanarak tam bir uzak kuyruk işleme döngüsü yaratabilirsiniz.



Şekil 13. PUT çağrısı sırasında yanıt kuyruğu adı yerine koyma değeri

Şekil 13 sayfa 200 içinde, yanıt kuyruğu kullanan tam bir uzak kuyruk işleme döngüsü gösterilir. Bu döngü, hem dağıtılmış kuyruğa alma ortamında hem de kümeleme ortamında geçerlidir. Ayrıntılar [Çizelge 20 sayfa 207](#) içinde gösterildiği gibi.

Uygulama QA 'yı QMB' de açar ve iletileri bu kuyruğa koyar. İletilere, kuyruk yöneticisi adı belirtilmeden QR yanıt kuyruğu adı verilir. Kuyruk yöneticisi QMA, QR yanıtlanacak kuyruk nesnesini bulur ve QRR diğer adını ve QMA\_class1kuyruk yöneticisi adını alır. Bu adlar, iletilerin yanıtlanma alanlarına konmuştur.

QMB 'deki uygulamalardan gelen yanıt iletileri, QMA\_class1adresindeki QRR' ye adreslenir. Kuyruk yöneticisi diğer ad tanımlaması QMA\_class1 , kuyruk yöneticisi tarafından iletileri kendisine ve QRR kuyruğuna akışı için kullanılır.

Bu senaryoda, uygulamalara yanıt iletileri için bir hizmet sınıfı seçme olanağı verme biçiminiz gösterilir. Sınıf, QMA 'daki kuyruk yöneticisi diğer adı tanımlaması QMA\_class1 ile birlikte QMB' deki QMA\_class1 iletiler kuyruğu tarafından uygulanır. Bu şekilde, bir uygulamanın yanıt kuyruğunu, akışların uygulamayı içermeden ayrılması için değiştirebilirsiniz. Uygulama her zaman bu hizmet sınıfı için QR ' yi seçer. Yanıt kuyruğu tanımlaması QR ile hizmet sınıfını değiştirme fırsatınız vardır.

Şunları oluşturmalısınız:

- Yanıt kuyruğu tanımlaması QR
- İletim kuyruğu nesnesi QMB
- Kanal\_çıkış tanımlaması
- Channel\_back tanımlaması
- Kuyruk yöneticisi diğer ad tanımlaması QMA\_class1
- Yoksa, yerel kuyruk nesnesi QRR

Bitişik sistemdeki tamamlayıcı yönetici aşağıdakileri oluşturmalıdır:

- Kanal tanımlaması alınıyor
- İletim kuyruğu nesnesi QMA\_class1
- İlişkili gönderme kanalı
- Yerel kuyruk nesnesi QA.

Uygulama programlarınız aşağıdakileri kullanır:

- Yanıtlanacak kuyruğu adı, koyma çağrılarında QR
- Alma çağrılarında kuyruk adı QRR

Bu şekilde, uygulamayı dahil etmeden hizmet sınıfını gerektiği gibi değiştirebilirsiniz. 'QR' yanıt diğer adını, 'QMA\_class1' iletiler kuyruğu ve 'QMA\_class1' kuyruk yöneticisi diğer adıyla birlikte değiştirin.

İleti kuyruğa konduğunda yanıtlanacak diğer ad nesnesi bulunamazsa, yerel kuyruk yöneticisi adı boş yanıtlanacak kuyruk yöneticisi adı alanına eklenir. Yanıt kuyruğu adı değişmeden kalır.

## Ad çözme kısıtlaması

Özgün ileti yerleştirildiğinde 'QMA' kuyruğunda yanıt kuyruğu için ad çözünürlüğü gerçekleştirildiğinden, 'QMB' de başka ad çözümlemesine izin verilmez. İleti, yanıtlanan uygulama tarafından yanıt kuyruğunun fiziksel adıyla yerleştirilir.

Uygulamalar, yanıt kuyruğu için kullandıkları adın, dönüş iletilerinin bulunacağı gerçek kuyruğun adından farklı olduğunu bilmelidir.

Örneğin, yanıtlanacak kuyruk diğer adı 'C1\_alias' ve 'C2\_alias' olan uygulamaların kullanımı için iki hizmet sınıfı sağlandığında, uygulamalar bu adları ileti koyma çağrılarında yanıtlanacak kuyruk adları olarak kullanır. Ancak uygulamalar, 'C1\_alias' için 'C1' kuyruklarında ve 'C2\_alias' için 'C2' kuyruklarında ileti görüntülenmesini bekliyor.

Ancak bir uygulama, yanıt iletilerini almak için kullanması gereken gerçek kuyruğun adını denetlemek için diğer ad kuyruğunda sorgu çağrısı yapabilir.

## İlgili kavramlar

“Kuyruk yöneticisi ve yanıtlanacak diğer adları yaratılması” sayfa 191  
Bu konuda, uzak kuyruk tanımlaması yaratmanın üç yolu açıklanmaktadır.

### “Yanıt kuyruğu diğer adı örneği” sayfa 202

Bu örnek, döndürülen iletilere ilişkin farklı bir rota (iletim kuyruğu) seçmek için bir yanıt diğer adının kullanılmasını gösterir. Bu olanağın kullanılması, uygulamalarla işbirliği içinde yanıtlanacak kuyruk adının değiştirilmesini gerektirir.

### “Örnek nasıl çalışır?” sayfa 203

Örneğin ve kuyruk yöneticisinin yanıtlanacak kuyruk diğer adını nasıl kullandığına ilişkin açıklama.

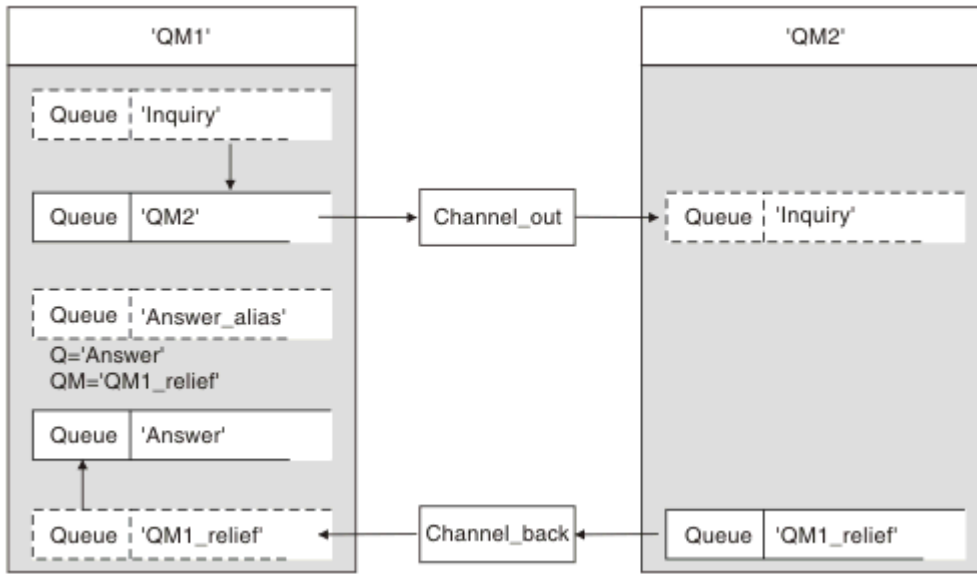
### “Yanıtın gönderileceği kuyruk diğer adı” sayfa 204

Uzak kuyruğa bir ileti koyan bir uygulamadan, yanıt iletilisini diğer ad yanıt kuyruğundan kaldıran aynı uygulamaya geçen bir süreç.

## **Yanıt kuyruğu diğer adı örneği**

Bu örnek, döndürülen iletilere ilişkin farklı bir rota (iletim kuyruğu) seçmek için bir yanıt diğer adının kullanılmasını gösterir. Bu olanağın kullanılması, uygulamalarla işbirliği içinde yanıtlanacak kuyruk adının değiştirilmesini gerektirir.

Şekil 14 sayfa 202’inde gösterildiği gibi, iletim kuyruğu, kanal ve kuyruk yöneticisi diğer adı da içinde olmak üzere, yanıt iletileri için dönüş rotası kullanılabilir olmalıdır.



Şekil 14. Yanıt kuyruğu diğer adı örneği

Bu örnek, 'QM2' adresindeki sunucu uygulamalarına ileti gönderen 'QM1' içindeki istekte bulunan uygulamalar içindir. Sunucudaki iletiler, 'QM1\_relief' iletim kuyruğu kullanılarak alternatif bir kanal aracılığıyla döndürülecek (varsayılan dönüş kanalı 'QM1' iletim kuyruğuyla sunulacaktır).

Yanıt kuyruğu diğer adı, 'Answer\_alias' adlı uzak kuyruk tanımlamasının belirli bir kullanımınıdır. QM1 'deki uygulamalar,' Sorgu 'kuyruğuna koydukları tüm iletilerin yanıtlanma alanında bu adı (' Answer\_alias ') içerir.

Yanıt kuyruğu tanımı 'Answer\_alias', 'Yanıt: QM1\_relief' olarak tanımlandı. QM1 'deki uygulamalar, yanıtlanmanın' Yanıt ' adlı yerel kuyrukta görünmesini bekler.

QM2 'deki sunucu uygulamaları, QM1adresindeki istekte bulunana yanıt iletilerine ilişkin kuyruk ve kuyruk yöneticisi adlarını almak için alınan iletilerin yanıt alanı alanını kullanır.

## **Bu örnekte kullanılan tanımlar: QM1**

QM1 adresindeki IBM MQ sistem yöneticisi, diğer nesnelere birlikte yanıt kuyruğunun 'Yanıt' oluşturulduğundan emin olmalıdır. '\*' ile işaretli kuyruk yöneticisi diğer adının adı, yanıt kuyruğu diğer adı tanımlamasında '\*' ile de işaretlenmiş olan kuyruk yöneticisi adıyla aynı olmalıdır.

<b>Nesne</b>	<b>Tanım</b>	
Yerel iletim kuyruğu	QM2	
Uzak kuyruk tanımlaması	Nesne adı	Sorgu
	Uzak kuyruk yöneticisi adı	QM2
	Uzak kuyruk adı	Sorgu
	İletim kuyruğu adı	QM2 (VARSAYILAN)
Kuyruk yöneticisi diğer adı	Nesne adı	QM1_relief *
	Kuyruk yöneticisi adı	QM1
	Kuyruk adı	(boş)
Yanıtın gönderileceği kuyruk diğer adı	Nesne adı	Yanıt_diğer adı
	Uzak kuyruk yöneticisi adı	QM1_relief *
	Uzak kuyruk adı	Yanıt

### **Tanımı QM1 ' e koy**

Uygulamalar, yanıtlanacak kişi alanlarını yanıtlanacak kişi kuyruğu diğer adıyla doldurur ve kuyruk yöneticisi adı alanını boş bırakır.

<b>Alan</b>	<b>İçindekiler</b>
Kuyruk adı	Sorgu
Kuyruk yöneticisi adı	(boş)
Yanıtın gönderileceği kuyruk adı	Yanıt_diğer adı
Yanıtın gönderileceği kuyruk yöneticisi	(boş)

### **Bu örnekte kullanılan tanımlar: QM2**

QM2 ' deki IBM MQ sistem yöneticisi, gelen iletiler için yerel kuyruğun var olduğundan ve doğru adlandırılmış iletim kuyruğunun yanıt iletileri için kullanılabilir olduğundan emin olmalıdır.

<b>Nesne</b>	<b>Tanım</b>
Yerel kuyruk	Sorgu
İletim kuyruğu	QM1_relief

### **Tanımı QM2 ' ye koy**

QM2 ' deki uygulamalar, özgün iletiden yanıt kuyruğu adını ve kuyruk yöneticisi adını alır ve yanıt iletisini yanıt kuyruğuna koyarken bunları kullanır.

<b>Alan</b>	<b>İçindekiler</b>
Kuyruk adı	Yanıt
Kuyruk yöneticisi adı	QM1_relief

### **Örnek nasıl çalışır?**

Örneğin ve kuyruk yöneticisinin yanıtlanacak kuyruk diğer adını nasıl kullandığına ilişkin açıklama.

Bu örnekte, QM1 'deki istekte bulunan uygulamalar, her zaman' Answer\_alias ' değerini, koyma çağrısının ilgili alanında yanıt kuyruğu olarak kullanır. İletilerini her zaman 'Yanıt' adlı kuyruktan alır.

Yanıt kuyruğu diğer ad tanımları, QM1 sistem yöneticisi tarafından, yanıt kuyruğunun adını 'Yanıt' ve dönüş rotasını 'QM1\_relief' olarak değiştirmek için kullanılabilir.

QM1 uygulamaları bu kuyrukte yanıtlarını beklediğinden, 'Yanıt' kuyruk adının değiştirilmesi kullanışlı değildir. Ancak, QM1 sistem denetimcisi dönüş rotasını (hizmet sınıfı) gerektiği şekilde değiştirebilir.

## Kuyruk yöneticisinin yanıtlanacak kuyruk diğer adını kullanma şekli

Kuyruk yöneticisi QM1 , uygulama tarafından yapılan yanıtlama çağrısında yer alan yanıt kuyruğu adı, yanıtlama kuyruğu diğer adıyla aynı olduğunda ve kuyruk yöneticisi kısmı boş olduğunda, yanıtın gönderileceği kuyruk diğer adından tanımlamaları alır.

Kuyruk yöneticisi, koyma çağrısındaki yanıt kuyruğu adını, tanımdaki kuyruk adıyla değiştirir. Koyma çağrısındaki boş kuyruk yöneticisi adını, tanımdaki kuyruk yöneticisi adıyla değiştirir.

Bu adlar, ileti tanımlayıcıdaki iletiyle birlikte taşınır.

Çizelge 17. Yanıtın gönderileceği kuyruk diğer adı		
Alan adı	Arama koy	İletim üstbilgisi
Yanıtın gönderileceği kuyruk adı	Yanıt_diğer adı	Yanıt
Yanıtın gönderileceği kuyruk yöneticisi adı	(boş)	QM1_relief

## Yanıtın gönderileceği kuyruk diğer adı

Uzak kuyruğa bir ileti koyan bir uygulamadan, yanıt iletisini diğer ad yanıt kuyruğundan kaldıran aynı uygulamaya geçen bir süreç.

Bu örneği tamamlamak için, sürece bir göz atalım.

1. Uygulama 'Sorgu' adlı bir kuyruğu açar ve iletileri bu kuyruğa koyar. Uygulama, ileti tanımlayıcısının yanıt alanlarını şu şekilde ayarlar:

Yanıtın gönderileceği kuyruk adı	Yanıt_diğer adı
Yanıtın gönderileceği kuyruk yöneticisi adı	(boş)

2. Kuyruk yöneticisi 'QM1', 'Answer\_alias' adlı uzak kuyruk tanımlamasını denetleyerek boş kuyruk yöneticisi adına yanıt verir. Hiçbir şey bulunamazsa, kuyruk yöneticisi ileti tanımlayıcısının yanıt kuyruğu yöneticisi alanına kendi adını ('QM1') yerleştirir.
3. Kuyruk yöneticisi 'Answer\_alias' adlı bir uzak kuyruk tanımlaması bulursa, tanımlamadan kuyruk adını ve kuyruk yöneticisi adlarını alır (kuyruk adı= 'Yanıt' ve kuyruk yöneticisi adı= 'QM1\_relief'). Daha sonra, bunları ileti tanımlayıcısının yanıtlama alanlarına yerleştirir.
4. 'QM1' kuyruk yöneticisi, istenen hedef kuyruğun 'QM2' kuyruk yöneticisinde olduğunu ve iletinin 'QM2' iletim kuyruğuna yerleştirildiğini belirlemek için 'Sorgu' uzak kuyruk tanımını kullanır. 'QM2', 'QM2' kuyruk yöneticisindeki kuyruklara yönlendirilmiş iletilere ilişkin varsayılan iletim kuyruğu adıdır.
5. 'QM1' kuyruk yöneticisi iletiyi iletim kuyruğuna koyduğunda, iletiye bir iletim üstbilgisi ekler. Bu üstbilgi, hedef kuyruğun adını ('Sorgu') ve hedef kuyruk yöneticisini ('QM2') içerir.
6. İleti, 'QM2' kuyruk yöneticisine gelir ve 'Sorgu' yerel kuyruğuna yerleştirilir.
7. Bir uygulama iletiyi bu kuyruktan alır ve iletiyi işler. Uygulama bir yanıt iletisi hazırlar ve bu yanıt iletisini özgün iletinin ileti tanımlayıcısından yanıt kuyruğu adına yerleştirir:

Yanıtın gönderileceği kuyruk adı	Yanıt
Yanıtın gönderileceği kuyruk yöneticisi adı	QM1_relief

8. 'QM2' kuyruk yöneticisi, koyma komutunu gerçekleştirir. Kuyruk yöneticisi adının ('QM1\_relief') uzak bir kuyruk yöneticisi olduğunu bularak, iletiyi iletim kuyruğuna aynı adla ('QM1\_relief') yerleştirir.



İletiyi, hedef kuyruğun adını ('Yanıt') ve hedef kuyruk yöneticisini ('QM1\_relief') içeren bir iletim üstbilgisi verilir.


9. İleti, 'QM1' kuyruk yöneticisine aktarıldı. Kuyruk yöneticisi, 'QM1\_relief' kuyruk yöneticisi adının bir diğer ad olduğunu anlar ve 'QM1\_relief' diğer ad tanımından 'QM1' fiziksel kuyruk yöneticisi adını çıkarır.
10. Kuyruk yöneticisi 'QM1', iletiyi iletim üstbilgisinde bulunan 'Yanıt' kuyruk adına koyar.
11. Uygulama yanıt iletisini 'Yanıt' kuyruğundan çıkarır.

## Ağ ile ilgili önemli noktalar

Dağıtılmış kuyruğa alma ortamında, ileti hedefleri yalnızca bir kuyruk adı ve kuyruk yöneticisi adıyla adreslendiğinden, belirli kurallar geçerlidir.

1. Kuyruk yöneticisi adının verildiği ve adın yerel kuyruk yöneticisi adından farklı olduğu yer:
  - Aynı adı taşıyan bir iletim kuyruğu kullanılabilir olmalıdır. Bu iletim kuyruğu, iletileri başka bir kuyruk yöneticisine aktaran bir ileti kanalının parçası olmalıdır ya da
  - Kuyruk yöneticisi adını aynı ya da başka bir kuyruk yöneticisi adına ve isteğe bağlı iletim kuyruğuna çözümlenmek için bir kuyruk yöneticisi diğer adı tanımlaması var olmalıdır ya da
  - İletim kuyruğu adı çözümlenmezse ve varsayılan iletim kuyruğu tanımlandıysa, varsayılan iletim kuyruğu kullanılır.
2. Yalnızca kuyruk adı sağlanırsa, yerel kuyruk yöneticisinde herhangi bir tipte ancak aynı ada sahip bir kuyruk bulunmalıdır. Bu kuyruk, bitişik bir kuyruk yöneticisine ilişkin bir iletim kuyruğu, bir kuyruk yöneticisi adı ve isteğe bağlı bir iletim kuyruğuna çözümlen bir uzak kuyruk tanımı olabilir.

Bunun kümeleme ortamında nasıl çalıştığını görmek için bkz. [Kümelere](#).

 Kuyruk yöneticileri bir kuyruk paylaşım grubunda (QSG) çalışıyorsa ve grup içi kuyruğa alma (IGQ) etkinleştirilmişse, SYSTEM.QSG.TRANSMIT.QUEUE. Daha fazla bilgi için bkz. [Grup içi kuyruğa alma](#).

Bir ileti kanalının, iletileri dağıtılmış kuyruğa alma ortamında bir kuyruk yöneticisinden diğerine taşıma senaryosunu göz önünde bulundurun.

Taşınmakta olan iletiler, ağdaki başka bir kuyruk yöneticisinden kaynaklanmış ve hedef olarak bilinmeyen bir kuyruk yöneticisi adı olan bazı iletiler gelmiş olabilir. Bu sorun, örneğin, bir kuyruk yöneticisi adı değiştiğinde ya da sistemden kaldırıldığında ortaya çıkabilir.

Kanal programı, bu iletilere ilişkin bir iletim kuyruğu bulamadığında bu durumu tanır ve iletileri teslim edilmeyen ileti kuyruğuna yerleştirir. Bu mesajların doğru hedefe iletilmesi için gerekli düzenlemeleri yapmak sizin sorumluluğunuzdadır. Diğer bir seçenek olarak, bunları oluşturanın tespit edilebildiği yaratıcıya geri döndürün.

Özgün iletide rapor iletileri istendiyse, bu koşullarda kural dışı durum raporları oluşturulur.

## Ad çözme kuralı

Hedef kuyruğun kimliğini değiştiren (yani, mantıksal olarak fiziksel ad değişikliği), yalnızca bir kez ve yalnızca kaynak kuyruk yöneticisinde gerçekleşen ad çözümlenmesi.

Çeşitli diğer ad olanaklarının sonraki kullanımı yalnızca ileti akışları ayrılırken ve birleştirilirken kullanılmalıdır.

## İade yönlendirmesi

İletiler, bir kuyruk ve kuyruk yöneticisi adı biçiminde bir dönüş adresi içerebilir. Bu dönüş adresi formu hem dağıtılmış kuyruğa alma ortamında hem de kümeleme ortamında kullanılabilir.

Bu adres, genellikle iletiyi oluşturan uygulama tarafından belirlenir. Daha sonra, kullanıcı çıkışı uygulamaları da içinde olmak üzere, iletiyi işleyen herhangi bir uygulama tarafından değiştirilebilir.

Bu adresin kaynağından bağımsız olarak, iletiyi işleyen herhangi bir uygulama, yanıt, durum ya da rapor iletilerini kaynak uygulamaya döndürmek için bu adresi kullanmayı seçebilir.

Bu yanıt iletilerinin yöneltişi şekli, özgün iletinin yöneltmesinden farklı değildir. Diğer kuyruk yöneticilerine yarattığınız ileti akışlarının karşılık gelen dönüş akışlarına ihtiyacı olduğunu bilmeniz gerekir.

## Fiziksel ad çakışmaları

Hedef yanıt kuyruğu adı, özgün kuyruk yöneticisinde bir fiziksel kuyruk adına çözüldü. Yanıt veren kuyruk yöneticisinde yeniden çözümlenmemelidir.

Bu, yalnızca fiziksel ve mantıksal kuyruk adlarıyla ilgili ağ çapında bir sözleşme tarafından önlenemeyen ad çakışması sorunları için olası bir olasılıktır.

## Kuyruk adı çevirilerini yönetme

Bir kuyruk yöneticisi diğer adı tanımlaması ya da uzak kuyruk tanımlaması yarattığınızda, bu adı taşıyan her ileti için ad çözme işlemi gerçekleştirilir. Bu durum idare edilmeli.

Bu açıklama, bitişik sistemlere ileti kanalları olan tek bir sistemle ilgili uygulama tasarımcıları ve kanal planlamacıları için sağlanır. Kanal planlaması ve denetiminin yerel bir görünümünü alır.

Bir kuyruk yöneticisi diğer adı tanımlaması ya da uzak kuyruk tanımlaması yarattığınızda, iletinin kaynağına bakılmaksızın, bu adı taşıyan her ileti için ad çözme işlemi gerçekleştirilir. Bir kuyruk yöneticisi ağında çok sayıda kuyruk içerebilecek bu durumu denetlemek için aşağıdaki çizelgeleri alıkoymanız gerekir:

- Çözülmüş kuyruk adlarına, çözülmüş kuyruk yöneticisi adlarına ve çözülmüş iletim kuyruğu adlarına göre kaynak kuyrukların ve kaynak kuyruk yöneticilerinin adları
- Aşağıdakilere göre kaynak kuyruklarının adları:
  - Çözümlenen hedef kuyruk adları
  - Çözümlenen hedef kuyruk yöneticisi adları
  - İletim kuyrukları
  - İleti kanalı adları
  - Bitişik sistem adları
  - Yanıtın gönderileceği kuyruk adları

**Not:** Bu bağlamda *kaynak* teriminin kullanılması, uygulama tarafından sağlanan kuyruk adını ya da kuyruk yöneticisi adını ya da ileti koymak için bir kuyruk açarken kanal programını belirtir.

Çizelge 18 sayfa 206, Çizelge 19 sayfa 207 ve Çizelge 20 sayfa 207 içinde bu tabloların her birine bir örnek gösterilmektedir.

Bu tablolardaki adlar, bu bölümdeki örneklerden türetilir ve bu tablo, bir düğümdeki kuyruk adı çözümlenmesi için pratik bir örnek olarak tasarlanmamıştır.

Kuyruk açıldığında belirlenen kaynak kuyruk	Kuyruk açıldığında belirtilen kaynak kuyruk yöneticisi	Çözülmüş kuyruk adı	Çözülmüş kuyruk yöneticisi adı	Çözülmüş iletim kuyruğu adı	Çözüm tipi
QA_norm	-	QA_norm	QMB	QMB	Uzak kuyruk
(herhangi biri)	QMB	-	-	QMB	(yok)
QA_norm	-	QA_norm	QMB	TX1	Uzak kuyruk

Çizelge 18. Kuyruk yöneticisi QMA ' da kuyruk adı çözümlemesi (devamı var)

Kuyruk açıldığında belirlenen kaynak kuyruk	Kuyruk açıldığında belirtilen kaynak kuyruk yöneticisi	Çözülmüş kuyruk adı	Çözülmüş kuyruk yöneticisi adı	Çözülmüş iletim kuyruğu adı	Çözüm tipi
QB	QMC	QB	QMD	QMB	Kuyruk yöneticisi diğer adı

Çizelge 19. Kuyruk yöneticisi QMB ' de kuyruk adı çözümlemesi

Kuyruk açıldığında belirlenen kaynak kuyruk	Kuyruk açıldığında belirtilen kaynak kuyruk yöneticisi	Çözülmüş kuyruk adı	Çözülmüş kuyruk yöneticisi adı	Çözülmüş iletim kuyruğu adı	Çözüm tipi
QA_norm	-	QA_norm	QMB	-	(yok)
QA_norm	QMB	QA_norm	QMB	-	(yok)
QA_norm	QMB_PRIORITY	QA_norm	QMB	-	Kuyruk yöneticisi diğer adı
(herhangi biri)	QMC	(herhangi biri)	QMC	QMC	(yok)
(herhangi biri)	QMD_norm	(herhangi biri)	QMD_norm	TX1	Kuyruk yöneticisi diğer adı
(herhangi biri)	QMD_PRIORITY	(herhangi biri)	QMD_PRIORITY	QMD_fast	Kuyruk yöneticisi diğer adı
(herhangi biri)	QMC_küçük	(herhangi biri)	QMC_küçük	TX_küçük	Kuyruk yöneticisi diğer adı
(herhangi biri)	QMC_large	(herhangi biri)	QMC_large	TX_external	Kuyruk yöneticisi diğer adı
QB_small	QMC	QB_small	QMC	TX_küçük	Uzak kuyruk
QB_large (Büyük)	QMC	QB_large (Büyük)	QMC	TX_büyük	Uzak kuyruk
(herhangi biri)	QME	(herhangi biri)	QME	TX1	Kuyruk yöneticisi diğer adı
QA	QMC	QA	QMC	TX1	Uzak kuyruk
QB	QMD	QB	QMD	TX1	Uzak kuyruk

Çizelge 20. QMA kuyruk yöneticisinde yanıt kuyruğu adı çevirisi

Uygulama tasarımı		Yanıtın gönderileceği diğer ad tanımlaması	
<b>Yerel QMGR</b>	<b>İletiler için kuyruk adı</b>	<b>Yanıt kuyruğu diğer adı</b>	<b>Yeniden Tanımlanan:</b>
QMA	QRR	QR	QRR: QMA_class1

## Kanal ileti sırası numaralandırması

Kanal, iletilerin iletim kuyruğundan alındıkları sırada teslim edildiklerini denetlemek için sıra numaralarını kullanır.

Kanal sıra numaraları, bir kanal başlatıldığında denetlenir ve bir uyumsuzluk oluştuğunda, kanalın her iki tarafında da kalıcı eşitleme verilerinin kaybolduğu anlamına gelir; örneğin, bir olağanüstü durumdan kurtarma (DR) yapılandırması ya da kanal şüpheli olduğunda toplu işleme sonu kesintiye uğradı.

Sıra numarası yanlış eşleşmelerinin sıfırlanması ya da yoksayılması, bkz. *qm.ini dosyasının kanal kısmi* içindeki **IgnoreSeqNumberMismatch**, bir ileti kümesinin kaybedilmesi ya da çoğaltılması riskini taşımaz ve bir kanalın belirsiz durumunu sıfırlamaz.

Bu bilgiler `DISPLAY CHSTATUS` kullanılarak görüntülenebilir. Sıra numarası ve LUWID adı verilen bir tanıttıcı, bir toplu işte aktarılan son ileti için kalıcı saklama alanında saklanır. Bu değerler, bağlantının her iki ucunun hangi iletilerin başarıyla aktarıldığı konusunda anlaşığından emin olmak için kanal başlatma sırasında kullanılır.

## İletilerin sıralı olarak alınması

Bir uygulama aynı hedef kuyruğa bir ileti dizisi koyarsa, bu iletiler, aşağıdaki koşullar karşılandıysa, bir dizi MQGET işlemiyle birlikte **tek** uygulaması tarafından sırayla alınabilir:

- Tüm koyma istekleri aynı uygulamadan yapıldı.
- Tüm koyma istekleri aynı iş biriminden ya da tüm koyma istekleri bir iş biriminin dışında yapıldı.
- Tüm iletilerin önceliği aynı.
- Tüm mesajlar aynı kalıcılığı taşıyor.
- Uzak kuyruklama için, yapılanış, uygulamanın kuyruk yöneticisi aracılığıyla hedef kuyruk yöneticisine ve hedef kuyruğa tek bir yol göndermesini sağlar.
- İletiler bir gitmeyen iletiler kuyruğuna konmaz (örneğin, bir kuyruk geçici olarak doluysa).
- İletiyi alan uygulama, örneğin belirli bir *MsgId* ya da *CorrelId* belirtilerek ya da ileti önceliklerini kullanarak alma sırasını kasıtlı olarak değiştirmez.
- Hedef kuyruktan iletileri almak için alma işlemleri yalnızca bir uygulama tarafından gerçekleştiriliyor. Birden fazla uygulama varsa, bu uygulamalar, gönderen bir uygulama tarafından konan her bir sıradaki tüm iletileri alacak şekilde tasarlanmalıdır.

**Not:** Diğer görevlerden ve iş birimlerinden gelen iletiler, sıranın tek bir iş biriminden konduğu yerde bile diziyle birlikte serpiştirilebilir.

Bu koşullar karşılanamıyorsa ve hedef kuyruktaki iletilerin sırası önemliyse, uygulama, iletilerin sırasını sağlamak için iletinin bir parçası olarak kendi ileti sıra numarasını kullanacak şekilde kodlanabilir.

## Hızlı, kalıcı olmayan iletilerin alınma sırası

Hızlı bir kanaldaki kalıcı olmayan iletiler, aynı kanaldaki kalıcı iletileri aşabilir ve bu nedenle sıra dışı olarak gelebilir. Alan MCA, kalıcı olmayan iletileri hedef kuyruğa hemen koyar ve görünür hale getirir. Kalıcı iletiler, sonraki eşitleme noktasına kadar görünür kılınmaz.

## Geri döngü sınaması

*Geriçevrim sınaması*, z/OS dışı platformlarda, bir iletişim bağlantısını başka bir makineyle bağlantı kurmadan sınamanızı sağlayan bir tekniktir.

İki kuyruk yöneticisi arasında ayrı makinelerdeki gibi bir bağlantı kurarsınız, ancak aynı makinedeki başka bir işleme geri dönerek bağlantıyı sınavsınız. Bu teknik, etkin bir ağa gerek kalmadan iletişim kodunuzu sınavabileceğiniz anlamına gelir.

Bunu yapma şekliniz, hangi ürün ve protokolleri kullandığınıza bağlıdır.

Windows sistemlerinde "loopback" bağdaştırıcısını kullanabilirsiniz.

Daha fazla bilgi için kullandığınız ürünlere ilişkin belgelere bakın.

## Rota izleme ve etkinlik kaydı

Bir iletinin bir dizi kuyruk yöneticisinden izlediği rotayı iki şekilde doğrulayabilirsiniz.

IBM MQ Denetim komutuyla kullanılabilen rota görüntüleme uygulamasını **dspmqrte** kullanabilir ya da etkinlik kaydını kullanabilirsiniz. Bu konuların her ikisi de [İzleme başvurusu](#) içinde açıklanmıştır.

## Dağıtılmış kuyruk yönetimine giriş

Dağıtılmış kuyruk yönetimi (DQM), kuyruk yöneticileri arasındaki iletişimi tanımlamak ve denetlemek için kullanılır.

Dağıtılmış kuyruk yönetimi:

- Kuyruk yöneticileri arasındaki iletişim kanallarını tanımlamanızı ve denetlemenizi sağlar
- İletileri, iletim kuyruğu olarak bilinen *yerel kuyruk* tipinden yerel sistemdeki iletişim bağlantılarına ve hedef kuyruk yöneticisindeki yerel kuyruklara iletişim bağlantılarından taşımanız için bir ileti kanalı hizmeti sağlar.
- Kanalların çalışmasını izlemek ve sorunları tanılamak için, panoları, komutları ve programları kullanarak size olanaklar sağlar

Kanal tanımları, kanal adlarını iletim kuyrukları, iletişim bağı tanıtıcıları ve kanal öznitelikleriyle ilişkilendirir. Kanal tanımları farklı platformlarda farklı şekillerde uygulanır. İleti gönderme ve alma, iletişimi başlatmak ve denetlemek için kanal tanımlarını kullanan *ileti kanalı araçları* (MCA) olarak bilinen programlar tarafından denetlenir.

MCA ' lar da DQM tarafından denetlenir. Yapı platforma bağlıdır, ancak genellikle dinleyicileri ve tetikleyici monitörlerini işletmen komutları ve panoları ile birlikte içerir.

*İleti kanalı* , iletileri bir kuyruk yöneticisinden diğerine taşımak için kullanılan tek yönlü bir veri bağlantısıdır. Bu nedenle, bir ileti kanalının bir çift MCA ile temsil edilen iki uç noktası vardır. Her uç nokta, ileti kanalının sonuna ilişkin bir tanımlamayı içerir. Örneğin, bir uç bir göndereni, diğer uç bir alıcıyı tanımlar.

Kanalların nasıl tanımlanacağına ilişkin ayrıntılar için aşağıdaki konuya bakın:

- **ALW** [“AIX, Linux, and Windows üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 239](#)
- **z/OS** [“z/OS üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 958](#)
- **IBM i** [“IBM i üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 262](#)

İleti kanalı planlama örnekleri için bkz:

- **ALW** [AIX, Linux, and Windows için ileti kanalı planlama örneği](#)
- **IBM i** [IBM için ileti kanalı planlama örneği](#)
- **z/OS** [z/OS için ileti kanalı planlama örneği](#)
- **z/OS** [z/OS kuyruk paylaşım gruplarını kullanma için ileti kanalı planlama örneği](#)

Kanal çıkışlarıyla ilgili bilgi için [İleti kanalları için kanal çıkış programları](#) başlıklı konuya bakın.

### İlgili kavramlar

[“İleti gönderme ve alma” sayfa 210](#)

Aşağıdaki şekilde, iletiler iletilildiğinde varlıklar arasındaki ilişkileri ayrıntılandıran dağıtılmış kuyruk yönetimi modeli gösterilmektedir. Ayrıca kontrol için akışı da gösterir.

[“Kanal denetimi işlevi” sayfa 217](#)

Kanal denetimi işlevi, kanalları tanımlamanız, izlemeniz ve denetlemeniz için olanaklar sağlar.

[“Bir ileti teslim edilemediğinde ne olur?” sayfa 230](#)

Bir ileti teslim edilemediğinde, MCA bunu birkaç şekilde işleyebilir. Yeniden deneyebilir, gönderene geri döndürebilir ya da göndermeyen iletiler kuyruğuna koyabilir.

“Başlatma ve yapılandırma dosyaları” sayfa 235

Kanal kullanıma hazırlama verilerinin işlenmesi IBM MQ platformunuza bağlıdır.

“İletiler için veri dönüştürme” sayfa 236

IBM MQ iletileri, farklı kuyruk yöneticilerindeki kuyruklar arasında gönderildiğinde veri dönüştürme gerektirebilir.

“Kendi ileti kanalı araçlarınızı yazma” sayfa 237

IBM MQ , kendi ileti kanalı aracı (MCA) programlarınızı yazmanızı ya da bağımsız bir yazılım satıcısından bir tane kurmanızı sağlar.

“Dağıtılmış kuyruk yönetimi için göz önünde bulundurulması gereken diğer konular” sayfa 237

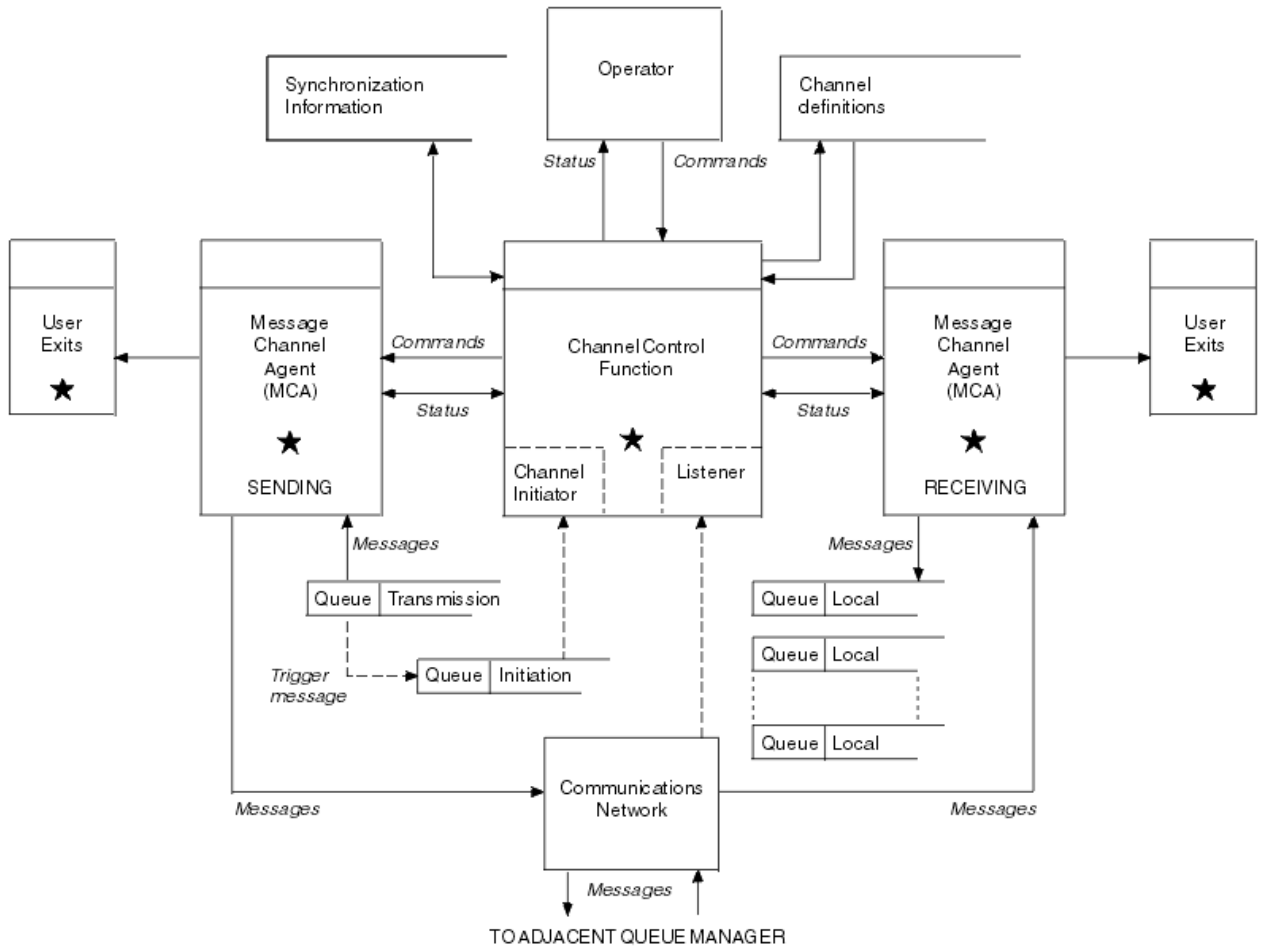
IBM MQ ' u dağıtılmış kuyruk yönetimi için hazırlarken dikkate alınacak diğer konular. Bu konu, Teslim Edilmeyen ileti kuyruğu, Kullanılmakta olan Kuyruklar, Sistem uzantıları ve kullanıcı çıkışı programları ve Güvenilen uygulamalar olarak Çalışan kanalları ve dinleyicileri kapsar.

## İlgili başvurular

Örnek yapılandırma bilgileri

## İleti gönderme ve alma

Aşağıdaki şekilde, iletiler iletilindiğinde varlıklar arasındaki ilişkileri ayrıntılandıran dağıtılmış kuyruk yönetimi modeli gösterilmektedir. Ayrıca kontrol için akışı da gösterir.



Şekil 15. Dağıtılmış kuyruk yönetimi modeli

**Not:**

1. Platforma bağılı olarak kanal başına bir MCA vardır. Belirli bir kuyruk yöneticisi için bir ya da daha çok kanal denetim işlevi olabilir.
2. MCA ' ların ve kanal kontrol fonksiyonlarının uygulanması yüksek düzeyde platforma bağılıdır. Bunlar, programlar, süreçler ya da iş parçacıkları olabilir ve bunlar tek bir varlık ya da birkaç bağımsız ya da bağılı parçadan oluşan birçok varlık olabilir.
3. Bir yıldızla işaretlenmiş tüm bileşenler MQI kullanabilir.

## Kanal parametreleri

MCA, deęiřtirgelerini ařaęıdaki yollardan biriyle alır:

- Bir komut tarafından başlatılırsa, kanal adı bir veri alanına iletilir. MCA daha sonra, özniteliklerini almak için kanal tanımını doğrudan okur.
- Gönderen ve bazı durumlarda sunucu kanalları için MCA, kuyruk yöneticisi tetikleyicisi tarafından otomatik olarak başlatılabilir. Kanal adı, uygulanabilir olduęu şekilde, tetikleyici işlemi tanımından alınır ve MCA ' ya geçirilir. Geri kalan işlem, daha önce açıklanan işlemle aynı. Sunucu kanalları yalnızca tam olarak nitelenmişse (yani, bağlantı kurmak için bir CONNAME belirtilirse) tetiklenecek şekilde ayarlanmalıdır.
- Bir gönderen, sunucu, istekte bulunan ya da istemci-baęlantı tarafından uzaktan başlatılırsa, kanal adı ortak ileti kanalı aracısından gelen ilk verilere aktarılır. MCA, özniteliklerini almak için kanal tanımını doğrudan okur.

Kanal tanımında tanımlı olmayan bazı öznitelikler de kararlařtırılabilir:

### İletileri böl

Bir uç bölünmüş iletileri desteklemiyorsa, bölünmüş iletiler gönderilmez.

### Dönüřtürme yeteneęi

Bir uç, gerektięinde gerekli kod sayfası dönüřtürme ya da sayısal kodlama dönüřtürmesini gerçekleřtiremezse, dięer uç bunu işlemelidir. İki ucu da desteklemiyorsa, gerektięinde kanal başlatılamaz.

### Daęıtım listesi desteęi

Bir uç daęıtım listelerini desteklemiyorsa, ortak MCA, iletim kuyruęunda bir işaret belirler; böylece, birden çok hedefe yönelik iletileri kesmeyi bilir.

## Kanal durumu ve sıra numaraları

İleti kanalı aracısı programları, her kanal için geçerli sıra numarası ve mantıksal iş birimi numarasının ve kanalın genel durumunun kayıtlarını tutar. Bazı platformlar, kanalları denetlemenize yardımcı olmak için bu durum bilgilerini görüntülemenizi saęlar.

## Bařka bir kuyruk yöneticisine ileti gönderilmesi

Bu bölümde, gerekli önkořullar ve yetkiler de içinde olmak üzere kuyruk yöneticileri arasında ileti göndermenin en basit yolu açıklanmaktadır. Uzak kuyruk yöneticisine ileti göndermek için dięer yöntemler de kullanılabilir.


Bir kuyruk yöneticisinden dięerine ileti göndermeden önce ařaęıdaki adımları gerçekleřtirmeniz gerekir:

1. Seętięiniz iletiřim protokolünün kullanılabilir olup olmadıęını denetleyin.
2. Kuyruk yöneticilerini başlatın.
3. Kanal başlatıcılarını başlatın.
4. Dinleyicileri başlatın.



Ayrıca, gerekli nesnelere yaratmak için doęru IBM MQ güvenlik yetkinizin olması gerekir.

Bir kuyruk yöneticisinden dięerine ileti göndermek için:

- Kaynak kuyruk yöneticisinde ařaęıdaki nesnelere tanımlayın:

- Gönderen kanalı
- Uzak kuyruk tanımlaması
- Başlatma kuyruğu (  z/OS için gerekli, aksi takdirde isteğe bağlı)
- İletim kuyruğu
- Gönderilmeyen iletiler kuyruğu
- Hedef kuyruk yöneticisinde aşağıdaki nesnelere tanımlayın:
  - Alıcı kanalı
  - Hedef kuyruk
  - Gönderilmeyen iletiler kuyruğu

IBM MQ platformunuza bağlı olarak bu nesnelere tanımlamak için birkaç farklı yöntem kullanabilirsiniz:

- Tüm altyapılarda, MQSC komutları içinde açıklanan IBM MQ komut dosyası komutlarını (MQSC) kullanabilirsiniz. Bu komutlar Yönetim görevlerini otomatikleştirme da IBM MQ Gezgini 'nde açıklanan programlanabilir komut biçimi (PCF) komutlarını içerir.
-  z/OS işletim sistemlerinde, Yönetme IBM MQ for z/OS başlıklı konuda açıklanan İşlem ve Denetim panolarını da kullanabilirsiniz.
-  IBM i üzerinde, pano arabirimini de kullanabilirsiniz.

Başka bir kuyruk yöneticisine ileti göndermeye ilişkin bileşenlerin yaratılmasına ilişkin ek bilgi için aşağıdaki alt başlıklara bakın:

### İlgili kavramlar

“IBM MQ dağıtılmış kuyruğa alma teknikleri” sayfa 189

Bu bölümdeki alt konularda, kanalları planlarken kullanılan teknikler açıklanmaktadır. Bu alt konularda, kuyruk yöneticilerinizi birbirine nasıl bağlayacağınızı planlamanıza ve uygulamalarınız arasındaki ileti akışını yönetmenize yardımcı olacak teknikler açıklanmaktadır.

“Dağıtılmış kuyruk yönetimine giriş” sayfa 209

Dağıtılmış kuyruk yönetimi (DQM), kuyruk yöneticileri arasındaki iletişimi tanımlamak ve denetlemek için kullanılır.

“Tetikleme kanalları” sayfa 232

IBM MQ , bir kuyruktaki belirli koşullar karşılandığında uygulamayı otomatik olarak başlatmak için bir olanak sağlar. Bu tesise tetikleme denir.

“İletilerin güvenliği” sayfa 229

Dağıtılmış kuyruk yönetimi, IBM MQ ürününün tipik kurtarma özelliklerine ek olarak, ileti kanalının iki ucu arasında eşgüdümlü bir eşitleme noktası yordamı kullanılarak iletilerin düzgün şekilde teslim edilmesini sağlar. Bu yordam bir hata saptarsa, sorunu araştırmanız için kanalı kapatır ve kanal yeniden başlatılıncaya kadar iletileri güvenli bir şekilde iletim kuyruğunda tutar.

### İlgili görevler

“Çoklu platformlarda kuyruk yöneticileri oluşturma” sayfa 7

İletileri ve kuyrukları kullanabilmek için önce en az bir kuyruk yöneticisi ve ilişkili nesnelere yaratmanız ve başlatmanız gerekir. Bir kuyruk yöneticisi, kendisiyle ilişkili kaynakları, özellikle de sahip olduğu kuyrukları yönetir. IBM MQ nesnelere yaratmak, değiştirmek, görüntülemek ve silmek için İletim Kuyruklama Arabirimi (MQI) çağrılarını ve komutlarına ilişkin uygulamalara kuyruğa alma hizmetleri sağlar.

“AIX, Linux, and Windows üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 239

DQM için, uzak kuyruk yöneticilerine yönelik kanalları yaratmanız, izlemeniz ve denetlemeniz gerekir. Komutları, programları, IBM MQ Explorer, kanal tanımlarına ilişkin dosyaları ve eşitleme bilgileri için bir depolama alanını kullanarak kanalları denetleyebilirsiniz.

“IBM i üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 262

Uzak kuyruk yöneticilerine kanallar yaratmak, izlemek ve denetlemek için DQM komutlarını ve panolarını kullanın. Her kuyruk yöneticisinin, uyumlu uzak kuyruk yöneticileriyle olan bağlantıları denetlemek için bir DQM programı vardır.



### “İstemci ile sunucu arasındaki bağlantıların yapılandırılması” sayfa 14

IBM MQ MQI clients ile sunucular arasındaki iletişim bağlantılarını yapılandırmak için iletişim protokolünüze karar verin, bağlantının her iki ucundaki bağlantıları tanımlayın, bir dinleyici başlatın ve kanalları tanımlayın.

### “Kuyruk yöneticisi kümesinin yapılandırılması” sayfa 283

Kümeler, hem ilk yapılandırmayı hem de devam eden yönetimi basitleştiren bir şekilde birbirine bağlanan kuyruk yöneticileri için bir mekanizma sağlar. Küme bileşenleri tanımlayabilir ve kümeleri oluşturabilir ve yönetebilirsiniz.

### “z/OS üzerinde diğer kuyruk yöneticileriyle iletişim kurulması” sayfa 954

Bu bölümde, dağıtılmış kuyruğa alma işlevini kullanmaya başlamadan önce yapmanız gereken IBM MQ for z/OS hazırlıkları açıklanmaktadır.

## ***Kanalların tanımlanması***

Bir kuyruk yöneticisinden diğerine ileti göndermek için iki kanal tanımlamanız gerekir. Kaynak kuyruk yöneticisinde bir kanal ve hedef kuyruk yöneticisinde bir kanal tanımlamanız gerekir.

### **Kaynak kuyruk yöneticisinde**

Kanal tipi SENDER olan bir kanal tanımlayın. Aşağıdakileri belirtmeniz gerekir:

- Kullanılacak iletim kuyruğunun adı (XMITQ özniteliği).
- Ortak sistemin bağlantı adı (CONNNAME özniteliği).
- Kullanmakta olduğunuz iletişim protokolünün adı (TRPTYPE özniteliği). IBM MQ for z/OS sistemlerinde protokol TCP ya da LU6.2 olmalıdır. Diğer platformlarda bunu belirtmeniz gerekmez. Değeri varsayılan kanal tanımınızdan almak için bu değeri bırakabilirsiniz.

Tüm kanal özniteliklerinin ayrıntıları [Kanal öznitelikleri](#) içinde verilmiştir.

### **Hedef kuyruk yöneticisinde**

Kanal tipi ALICI olan ve gönderen kanalla aynı adı taşıyan bir kanal tanımlayın.

Kullanmakta olduğunuz iletişim protokolünün adını belirtin (TRPTYPE özniteliği). IBM MQ for z/OS sistemlerinde protokol TCP ya da LU6.2 olmalıdır. Diğer platformlarda bunu belirtmeniz gerekmez. Değeri varsayılan kanal tanımınızdan almak için bu değeri bırakabilirsiniz.

Alıcı kanal tanımlamaları soysal olabilir. Bu, aynı alıcıyla iletişim kuran birden çok kuyruk yöneticiniz varsa, gönderen kanalların tümü günlük nesnesi için aynı adı belirleyebilir ve bir alıcı tanımlaması tümü için geçerli olur.

Kanalı tanımladığınızda, PING CHANNEL komutunu kullanarak kanalı sınavabilirsiniz. Bu komut, gönderen kanaldan alıcı kanalına özel bir ileti gönderir ve döndürülüp döndürülmediğini denetler.

**Not:** TRPTYPE parametresinin değeri, yanıt veren ileti kanalı aracısı tarafından yoksayılr. Örneğin, gönderen kanal tanımındaki TCP ' nin TRPTYPE değeri, alıcı kanal tanımında LU62 TRPTYPE değeri ile başarıyla başlar.

## ***Kuyrukların tanımlanması***

Bir kuyruk yöneticisinden diğerine ileti göndermek için en çok altı kuyruk tanımlamanız gerekir. Kaynak kuyruk yöneticisinde en çok dört kuyruk ve hedef kuyruk yöneticisinde en çok iki kuyruk tanımlamanız gerekir.

### **Kaynak kuyruk yöneticisinde**

- Uzak kuyruk tanımlaması

Bu tanımda aşağıdakileri belirtin:

#### **Uzak kuyruk yöneticisi adı**

Hedef kuyruk yöneticisinin adı.

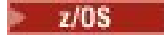
#### **Uzak kuyruk adı**


Hedef kuyruk yöneticisindeki hedef kuyruğun adı.

### İletim kuyruğu adı


İletim kuyruğunun adı. Bu iletim kuyruğu adını belirlemenize gerek yoktur. Bunu yapmazsanız, hedef kuyruk yöneticisiyle aynı adı taşıyan bir iletim kuyruğu kullanılır. Bu yoksa, varsayılan iletim kuyruğu kullanılır. Varsayılan olarak kuyruğun bulunması için, iletim kuyruğuna hedef kuyruk yöneticisiyle aynı adı vermeniz önerilir.

- Başlatma kuyruğu tanımlaması

 Bu gereklidir. SYSTEM.CHANNEL.INITQ.

 Bu isteğe bağlıdır. SYSTEM.CHANNEL.INITQ.

- İletim kuyruğu tanımı

USAGE özniteliği XMITQ olarak ayarlanmış bir yerel kuyruk.  IBM MQ for IBM i yerel arabirimini kullanıyorsanız, USAGE özniteliği \*TMQ olur.

- Gönderilmeyen ileti kuyruğu tanımı

Teslim edilmeyen iletilerin yazılabileceği bir teslim edilmeyen ileti kuyruğu tanımlayın.

### Hedef kuyruk yöneticisinde

- Yerel kuyruk tanımlaması

Hedef kuyruk. Bu kuyruğun adı, kaynak kuyruk yöneticisindeki uzak kuyruk tanımlamasının uzak kuyruk adı alanında belirtilenle aynı olmalıdır.

- Gönderilmeyen ileti kuyruğu tanımı

Teslim edilmeyen iletilerin yazılabileceği bir teslim edilmeyen ileti kuyruğu tanımlayın.

### İlgili kavramlar

[“İletim kuyruğu yaratılması” sayfa 214](#)

Bir kanal (istekte bulunan kanal dışında) başlatılmadan önce, iletim kuyruğunun bu bölümde açıklandığı gibi tanımlanması gerekir. İletim kuyruğunun kanal tanımında adlandırılması gerekir.

[“IBM i üzerinde iletim kuyruğu yaratılması” sayfa 214](#)

MQM Kuyruğu Yarat panosunu kullanarak IBM i altyapısında bir iletim kuyruğu yaratabilirsiniz.

#### İletim kuyruğu yaratılması

Bir kanal (istekte bulunan kanal dışında) başlatılmadan önce, iletim kuyruğunun bu bölümde açıklandığı gibi tanımlanması gerekir. İletim kuyruğunun kanal tanımında adlandırılması gerekir.

Her gönderen ileti kanalı için, USAGE özniteliği XMITQ olarak ayarlanmış bir yerel kuyruk tanımlayın. Uzak kuyruk tanımlarınızda belirli bir iletim kuyruğunu kullanmak istiyorsanız, gösterildiği gibi uzak bir kuyruk yaratın.

Bir iletim kuyruğu yaratmak için, aşağıdaki örneklerde gösterildiği gibi IBM MQ Komutlarını (MQSC) kullanın:

#### İletim kuyruğu yaratılması örneği

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') USAGE(XMITQ)
```

#### Uzak kuyruk yaratma örneği

```
DEFINE QREMOTE(PAYROLL) DESCR('Remote queue for QM2') +  
XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

İletim kuyruğuna, örneklerde gösterildiği gibi, uzak sistemdeki kuyruk yöneticisi adını adlandırmayı düşünebilirsiniz.

#### IBM i üzerinde iletim kuyruğu yaratılması

MQM Kuyruğu Yarat panosunu kullanarak IBM i altyapısında bir iletim kuyruğu yaratabilirsiniz.

Her gönderen ileti kanalı için, Kullanım alanı özneliği \*TMQ olarak ayarlanmış bir yerel kuyruk tanımlamanız gerekir.

Uzak kuyruk tanımlarını kullanmak istiyorsanız, \*RMT tipinde bir kuyruk yaratmak ve \*NORMAL değerini kullanmak için aynı komutu kullanın.

Bir iletim kuyruğu yaratmak için, komut satırındaki CRTMQMQ komutunu kullanarak ilk kuyruk yaratma panosunu sunun; bkz. [Şekil 16 sayfa 215](#).

```
Create MQM Queue (CRTMQMQ)
Type choices, press Enter.
Queue name . . . . .
Queue type . . . . . ____ *ALS, *LCL, *MDL, *RMT
Message Queue Manager name . . . *DFT_____
-----

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
+
```

*Şekil 16. Kuyruk yarat (1)*

Kuyruğun adını yazın ve yaratmak istediğiniz kuyruk tipini belirleyin: Yerel, Uzak ya da Diğer Ad. Bir iletim kuyruğu için, bu panoda Local (\*LCL) seçeneğini belirleyin ve Enter tuşuna basın.

MQM Kuyruğu Yarat panosunun ikinci sayfası görüntülenir; bkz. [Şekil 17 sayfa 215](#).

```
Create MQM Queue (CRTMQMQ)
Type choices, press Enter.
Queue name . . . . . > HURS.2.HURS.PRIORIT
Queue type . . . . . > *LCL *ALS, *LCL, *MDL, *RMT
Message Queue Manager name . . . *DFT
Replace . . . . . *NO *NO, *YES
Text 'description' . . . . .
Put enabled . . . . . *YES *SYSDFTQ, *NO, *YES
Default message priority . . . . 0 0-9, *SYSDFTQ
Default message persistence . . . *NO *SYSDFTQ, *NO, *YES
Process name . . . . .
Triggering enabled . . . . . *NO *SYSDFTQ, *NO, *YES
Get enabled . . . . . *YES *SYSDFTQ, *NO, *YES
Sharing enabled . . . . . *YES *SYSDFTQ, *NO, *YES

More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```

*Şekil 17. Kuyruk yarat (2)*

Gösterilen varsayılan değerlerden herhangi birini değiştirin. Sonraki ekrana gitmek için sayfayı aşağı doğru kaydırın; bkz. [Şekil 18 sayfa 216](#).

Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

```
Default share option . . . . . *YES      *SYSDFTQ, *NO, *YES
Message delivery sequence . . . *PTY    *SYSDFTQ, *PTY, *FIFO
Harden backout count . . . . . *NO     *SYSDFTQ, *NO, *YES
Trigger type . . . . . *FIRST  *SYSDFTQ, *FIRST, *ALL...
Trigger depth . . . . . 1          1-99999999, *SYSDFTQ
Trigger message priority . . . . 0       0-9, *SYSDFTQ
Trigger data . . . . . '          '
Retention interval . . . . . 999999999 0-999999999, *SYSDFTQ
Maximum queue depth . . . . . 5000    1-24000, *SYSDFTQ
Maximum message length . . . . . 4194304 0-4194304, *SYSDFTQ
Backout threshold . . . . . 0         0-999999999, *SYSDFTQ
Backout requeue queue . . . . . '          '
Initiation queue . . . . . '          '

```

More...

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys

### Şekil 18. Kuyruk yarat (3)

İletim kuyruğu için, bu panonun Kullanım alanına \*TMQ yazın ve diğer alanlarda gösterilen varsayılan değerlerden herhangi birini değiştirin.

Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

```
Usage . . . . . *TMQ      *SYSDFTQ, *NORMAL, *TMQ
Queue depth high threshold . . . 80      0-100, *SYSDFTQ
Queue depth low threshold . . . 20     0-100, *SYSDFTQ
Queue full events enabled . . . *YES   *SYSDFTQ, *NO, *YES
Queue high events enabled . . . *YES   *SYSDFTQ, *NO, *YES
Queue low events enabled . . . *YES   *SYSDFTQ, *NO, *YES
Service interval . . . . . 999999999 0-999999999, *SYSDFTQ
Service interval events . . . . *NONE  *SYSDFTQ, *HIGH, *OK, *NONE
Distribution list support . . . *NO    *SYSDFTQ, *NO, *YES
Cluster Name . . . . . *SYSDFTQ
Cluster Name List . . . . . *SYSDFTQ
Default Binding . . . . . *SYSDFTQ *SYSDFTQ, *OPEN, *NOTFIXED

```

Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys

### Şekil 19. Kuyruk yarat (4)

Alanların doğru verileri içerdiğinden emin olduğunuzda, kuyruğu yaratmak için Enter tuşuna basın.

### **Kanal başlatılıyor**

İletileri kaynak kuyruk yöneticisinde tanımlanan uzak kuyruğa yerleştirdiğinizde, bu iletiler kanal başlatılıncaya kadar iletim kuyruğunda saklanır. Kanal başlatıldığında, iletiler uzak kuyruk yöneticisindeki hedef kuyruğa teslim edilir.

START CHANNEL komutunu kullanarak, gönderen kuyruk yöneticisinde kanalı başlatın. Gönderme kanalını başlattığınızda, alıcı kanal otomatik olarak (dinleyici tarafından) başlatılır ve iletiler hedef kuyruğa gönderilir. İletilerin aktarılabilmesi için ileti kanalının her iki ucunun da çalışıyor olması gerekir.

Kanalın iki ucu farklı kuyruk yöneticilerine ait olduğundan, bunlar farklı özniteliklerle tanımlanmış olabilir. Farklılıkları çözmek için, kanal başlatıldığında iki uç arasında bir ilk veri anlaşması vardır. Genel olarak,

kanalın iki ucu, daha az kaynak gerektiren özniteliklerle çalışır. Bu, daha büyük sistemlerin, ileti kanalının diğer ucunda daha küçük sistemlerin kaynaklarını barındırmasına olanak sağlar.

Gönderen MCA, büyük iletileri kanal boyunca göndermeden önce böler. Bunlar uzak kuyruk yöneticisinde yeniden birleştirilir. Bu, kullanıcı tarafından anlaşılmıyor.

Bir MCA, birden çok iş parçacığı kullanarak ileti aktarabilir. *Pipelining* adı verilen bu işlem, MCA 'nın iletileri daha verimli ve daha az bekleme durumuyla aktarmasını sağlar. Boru hattı kullanımı kanal performansını artırır.

## Kanal denetimi işlevi

Kanal denetimi işlevi, kanalları tanımlamanız, izlemeniz ve denetlemeniz için olanaklar sağlar.

Komutlar, panolar, programlar ya da bir komut satırından kanal denetimi işlevine gönderilir. Pano arabirimi, kanal durumu ve kanal tanımı verilerini de görüntüler. “AIX, Linux, and Windows üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 239 içinde ayrıntılı olarak açıklanan Programlanabilir Komut Biçimlerini ya da bu IBM MQ komutlarını (MQSC) ve denetim komutlarını kullanabilirsiniz.

Komutlar aşağıdaki gruplara ayrılır:

- Kanal yönetimi
- Kanal denetimi
- Kanal durumu izleme

Kanal yönetimi komutları, kanalların tanımlarıyla ilgilenir. Aşağıda belirtilenleri gerçekleştirmenizi sağlar:

- Kanal tanımı yarat
- Kanal tanımını kopyala
- Kanal tanımının değiştirilmesi
- Kanal tanımını sil

Kanal denetim komutları, kanalların çalışmasını yönetir. Aşağıda belirtilenleri gerçekleştirmenizi sağlar:

- Kanal başlatılması
- Kanalı durdur
- İş ortağıyla yeniden eşitle (bazı uygulamalarda)
- İleti sıra numaralarını sıfırla
- Belirsiz bir ileti kümesini çözün
- Ping: kanal üzerinden test iletişimi gönder

Kanal izleme, kanalların durumunu görüntüler; örneğin:

- Geçerli kanal ayarları
- Kanalın etkin ya da devre dışı olup olmadığı
- Kanalın eşitlenmiş durumda sonlandırılıp sonlandırılmadığını belirler

## İlgili kavramlar

Sorun belirlemeye yardımcı olacak bilgilerin bulunduğu yer

## Kanallar hazırlanıyor

Bir ileti kanalını ya da MQI kanalını başlatmayı denemeden önce, kanalı hazırlamanız gerekir. Yerel ve uzak kanal tanımlarının tüm özniteliklerinin doğru ve uyumlu olduğundan emin olmanız gerekir.

Kanal öznitelikleri , kanal tanımlarını ve özniteliklerini açıklar.

Açık kanal tanımları oluştursanız da, bir kanal başlatıldığında gerçekleştirilen kanal görüşmeleri, tanımlanan değerlerden birini ya da diğerini geçersiz kılabilir. Bu davranış normaldir ve kullanıcı tarafından anlaşılır değildir ve bu şekilde düzenlenmiştir, aksi takdirde uyumsuz tanımlar birlikte çalışabilir.

## **Alıcı ve sunucu bağlantısı kanallarının otomatik tanımı**

IBM MQ içinde z/OS dışındaki tüm platformlarda, uygun bir kanal tanımlaması yoksa, otomatik tanımlama etkinleştirilmiş bir alıcı ya da sunucu bağlantısı kanalı için otomatik olarak bir tanımlama yaratılır.

Tanımlama aşağıdaki kullanılarak yaratılır:

1. Uygun model kanal tanımı, SYSTEM.AUTO.RECEIVER ya da SYSTEM.AUTO.SVRCONN. Otomatik tanımlamaya ilişkin model kanal tanımlamaları, sistem varsayılan değerleri olan SYSTEM.DEF.RECEIVER ve SYSTEM.DEF.SVRCONN, açıklama alanı dışında, "Otomatik olarak tanımlanan" ve ardından 49 boşluk gelir. Sistem yöneticisi, sağlanan model kanalı tanımlarının herhangi bir bölümünü değiştirmeyi seçebilir.
2. Ortak sistemden gelen bilgiler. Ortağın değerleri kanal adı ve sıra numarası kaydırma değeri için kullanılır.
3. Otomatik tanımlama tarafından yaratılan değerleri değiştirmek için kullanabileceğiniz bir kanal çıkış programı. Bkz. [Kanal otomatik tanımlama çıkış programı](#).

Daha sonra, tanımın otomatik tanımlama çıkışı tarafından değiştirilip değiştirilmediğini ya da model tanımlamasının değiştirilip değiştirilmediğini saptamak için tanımlama denetlenir. İlk 44 karakterin ardından 29 boşluk eklenmeye devam ederse, kuyruk yöneticisi adı eklenir. Son 20 karakter hala boşluktaysa, yerel saat ve tarih eklenir.

Tanımlama yaratılıp saklandığında, kanal başlangıcı tanımlama her zaman varmış gibi devam eder. Toplu iş boyutu, iletim boyutu ve ileti boyutu iş ortağıyla karşılaştırılır.

## **Diğer nesnelere tanımlama**

Bir ileti kanalının başlatılabilmesi için, kuyruk yöneticisinde her iki ucu da tanımlanmalı (ya da otomatik tanımlama için etkinleştirilmelidir). Hizmet vereceği iletim kuyruğu, gönderen uçta kuyruk yöneticisine tanımlanmalıdır. İletişim bağlantısı tanımlanmalı ve kullanılabilir olmalıdır. "[Dağıtılmış kuyruğa alma yapılandırılıyor](#)" sayfa 189 içinde açıklanan senaryoları gerçekleştirmek için, uzak kuyruk tanımlamaları, kuyruk yöneticisi diğer adı tanımlamaları ve yanıt kuyruğu diğer adı tanımlamaları gibi diğer IBM MQ nesnelere hazırlamanız gerekebilir.

MQI kanallarını tanımlama hakkında bilgi için bkz. "[MQI kanallarının tanımlanması](#)" sayfa 28.

## **İletim kuyruğu başına birden çok ileti kanalı**

İletim kuyruğu başına birden çok kanal tanımlanabilir, ancak bu kanallardan yalnızca biri aynı anda etkin olabilir. Trafik dengeleme ve bağlantı hatası düzeltme işlemleri için kuyruk yöneticileri arasında alternatif rotalar sağlanması için bu seçeneği göz önünde bulundurun. İletim kuyruğu, önceki kanal tarafından sona erdirildiğinde, gönderme sonunda bir ileti kümesi şüpheli olarak bırakılırsa, başka bir kanal tarafından kullanılamaz. Daha fazla bilgi için "[Belirsiz kanalların işlenmesi](#)" sayfa 228 başlıklı konuya bakın.

## **Kanal başlatılması**

Bir kanal, iletileri dört yoldan biriyle iletmeye başlayabilir. Şöyle olabilir:

- Bir işleç tarafından başlatıldı (alıcı, küme alıcısı ya da sunucu bağlantısı kanalları değil).
- İletim kuyruğundan tetiklendi. Bu yöntem, yalnızca gönderen kanallar ve tam nitelikli sunucu kanalları (CONNNAME belirten kanallar) için geçerlidir. Tetikleyici kanallar için gerekli nesnelere hazırlamalısınız.
- Bir uygulama programından başlatıldı (alıcı, küme alıcısı ya da sunucu bağlantısı kanallarından değil).
- Bir gönderen, küme gönderen, istekte bulunan, sunucu ya da istemci-bağlantı kanalı tarafından ağdan uzaktan başlatıldı. Alıcı, küme alıcısı ve muhtemelen sunucu ve istekte bulunan kanal iletimleri bu şekilde başlatılır; sunucu bağlantısı kanalları da başlatılır. Kanalların kendileri zaten başlatılmalıdır (yani, etkinleştirilmelidir).

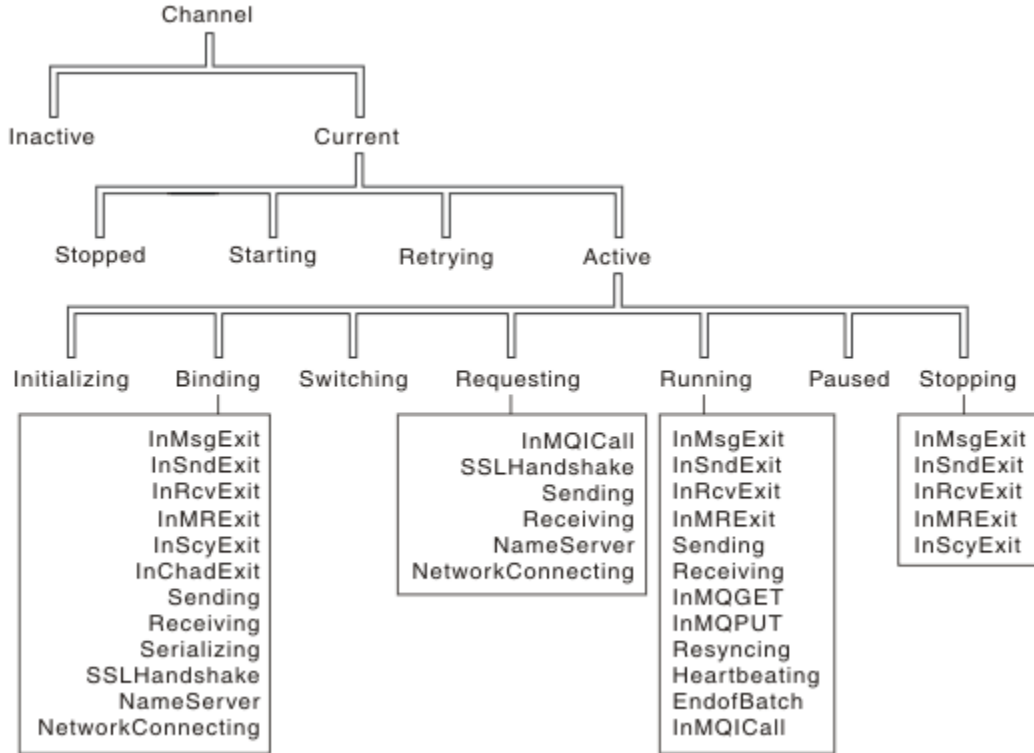
**Not:** Bir kanal 'başlatıldığından' mutlaka ileti göndermez. Bunun yerine, daha önce açıklanan dört olaydan biri gerçekleştiğinde iletimi başlatmak için 'etkin' olabilir. Bir kanalın etkinleştirilmesi ve devre dışı bırakılması, START ve STOP işletmen komutları kullanılarak gerçekleştirilir.

### **Kanal durumları**

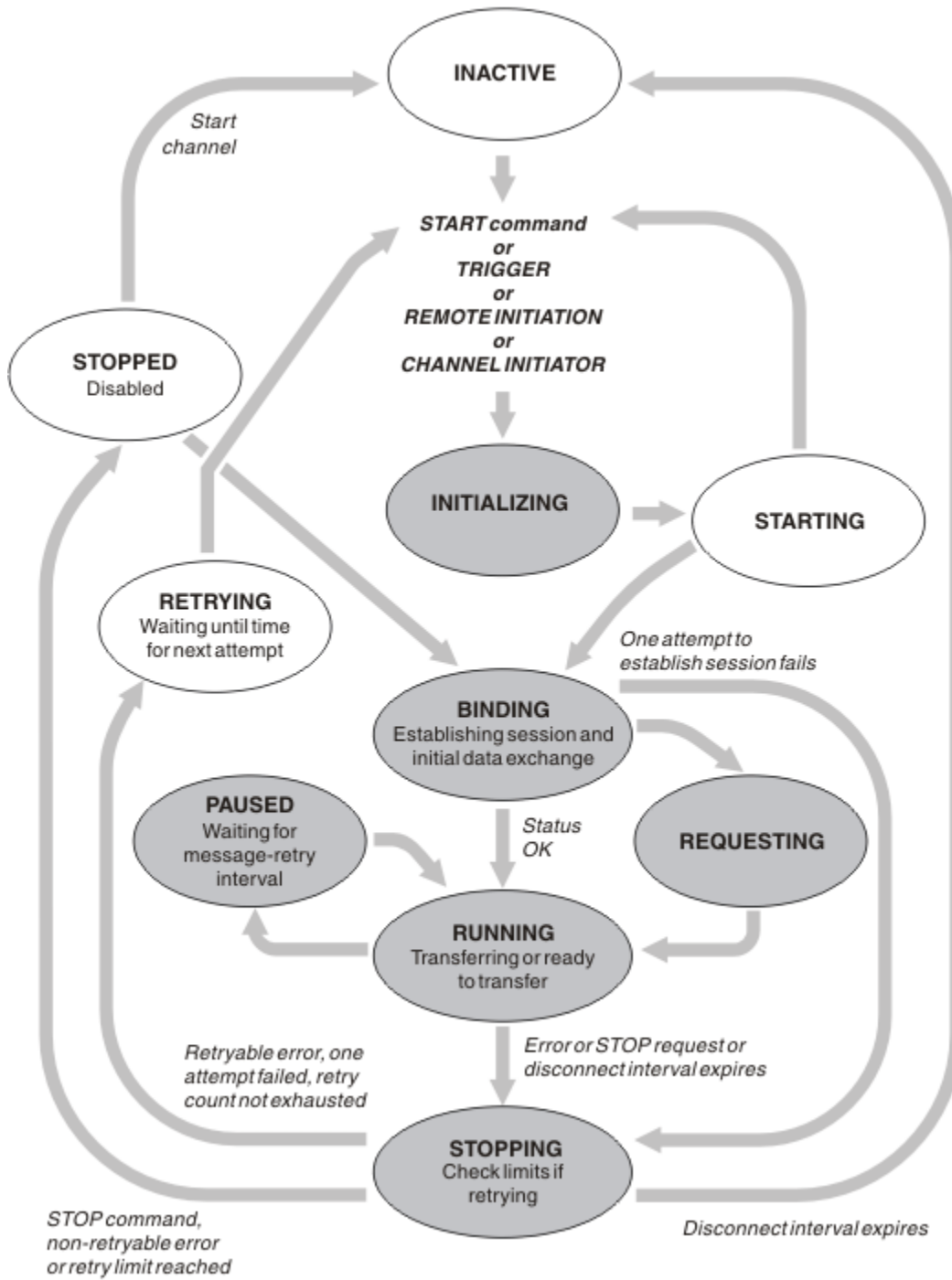
Bir kanal herhangi bir zamanda birçok eyaletten birinde olabilir. Bazı eyaletlerin de alt devletleri vardır. Belirli bir durumdan bir kanal diğer durumlara geçebilir.

Şekil 20 sayfa 219 , kanal durumlarının her biri için geçerli olan tüm olası kanal durumlarının ve alt durumların sıradüzenini gösterir.

Şekil 21 sayfa 220 , kanal durumları arasındaki bağlantıları gösterir. Bu bağlantılar, tüm ileti kanalı ve sunucu bağlantısı kanalları için geçerlidir.



Şekil 20. Kanal durumları ve alt durumları



Şekil 21. Kanal durumları arasındaki akışlar

## Geçerli ve etkin

Bir kanal, etkin değil dışında bir durumdaysa, *yürürlükteki* kanaldır. REDENYOR, DURDURULDU ya da STARTING durumunda değilse, yürürlükteki kanal *etkin* olur. Bir kanal etkin olduğunda, kaynağı tüketir ve bir işlem ya da iş parçacığı çalışır. Etkin bir kanalın yedi olası durumu (INITIALIZING, BINDING, SWITCHING, İSTEKTE BULUNMA, RUNNING, PAUSED ya da DURDURMA) Şekil 21 sayfa 220’inde vurgulanır.

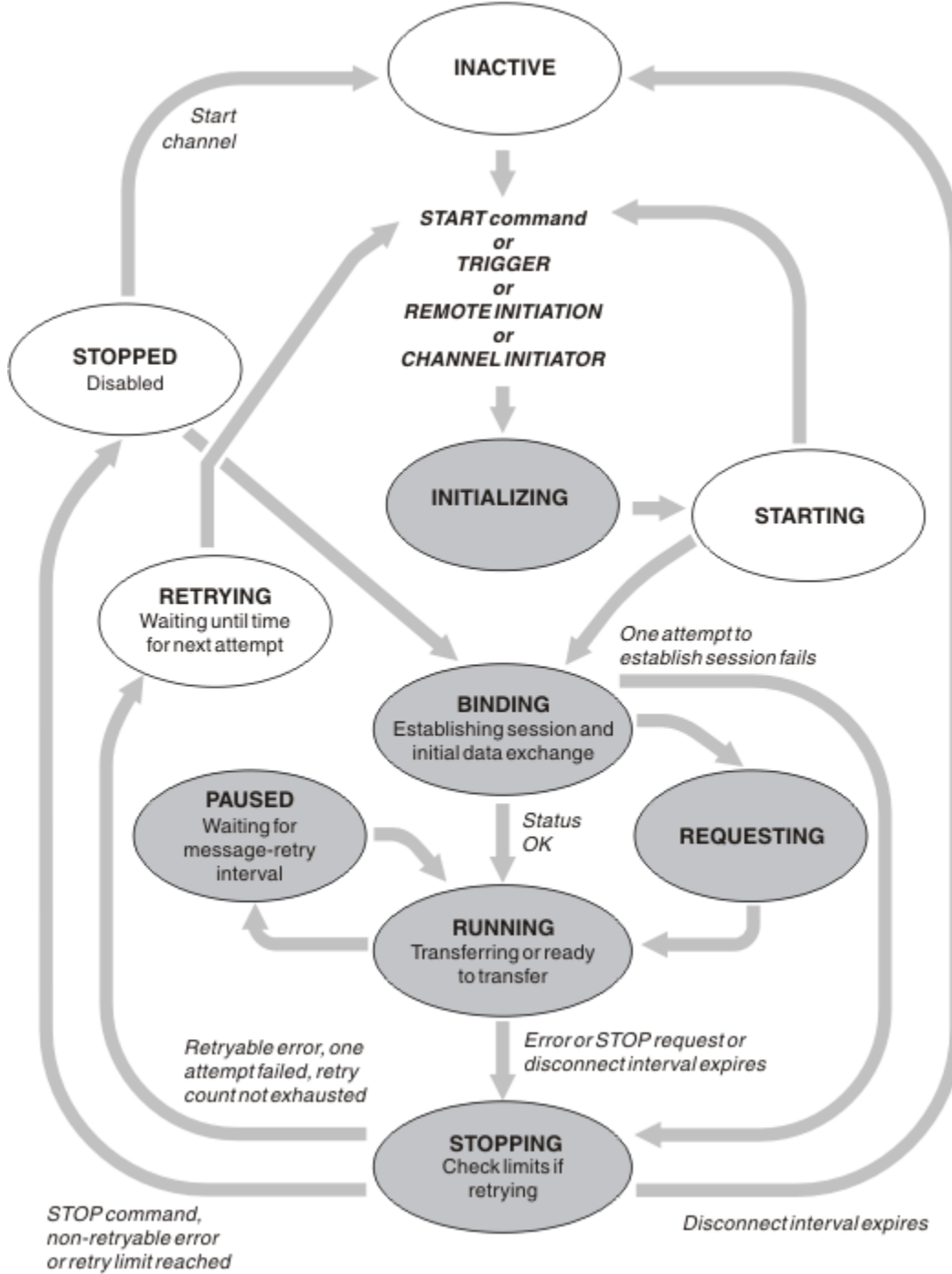
Aktif bir kanal, kanalın tam olarak ne yaptığına dair daha fazla ayrıntı veren bir alt durumu da gösterebilir. Her duruma ilişkin alt durumlar Şekil 20 sayfa 219’inde gösterilir.



*Geçerli ve etkin*

Kanal, etkin değil dışında bir durumdaysa "yürürlükteki" kanaldır. REDENYOR, DURDURULDU ya da STARTING durumunda değilse, yürürlükteki kanal "etkin" olur.

Bir kanal "etkin" ise, kanalın tam olarak ne yaptığına ilişkin daha fazla ayrıntı veren bir alt durum da gösterilebilir.



Şekil 22. Kanal durumları arasındaki akışlar





**Not:**

1. Bir kanal Şekil 22 sayfa 221 (INITIALIZING, BINDING, İSTEKTE BULUNMA, RUNNING, PAUSED ya da DURDURMA) içinde vurgulanan altı durumdan birinde olduğunda, kaynağı tüketir ve bir işlem ya da iş parçacığı çalışır; kanal *etkin*.




2. Bir kanal DURDURULDU durumundayken, sonraki durum henüz bilinmediği için oturma etkin olabilir.

## Yürürlükteki kanal sayısı üst sınırını belirtme

Aynı anda geçerli olabilecek kanal sayısı üst sınırını belirtebilirsiniz. Bu sayı, yeniden denenmekte olan kanallar ve durdurulan kanallar da içinde olmak üzere, kanal durumu çizelgesinde girişleri olan kanalların sayısıdır. Altyapınız için bunu belirtin:

-  ALTER QMGR MAXCHL komutunu kullanın.
-  Kuyruk yöneticisi kullanıma hazırlama dosyasını düzenleyin.
-   Kuyruk yöneticisi yapılandırma dosyasını düzenleyin.
- IBM MQ Explorer kullanın.

Kullanıma hazırlama ya da yapılandırma kütüğü kullanılarak ayarlanan değerlerle ilgili ek bilgi için [Dağıtılmış kuyruğa alma için yapılandırma kütüğü](#) stanzasına bakın. Kanal sayısı üst sınırını belirtme hakkında daha fazla bilgi için aşağıdaki konulara bakın:

-  [IBM MQ Yönetimi](#).
-  [IBM MQ for IBM i Yönetimi](#).
-  [IBM MQ for z/OS Yönetimi](#).

### Not:

1. Sunucu bağlantısı kanalları bu sayıda yer alır.
2. Bir kanalın etkinleşebilmesi için önce geçerli olması gerekir. Bir kanal başlatılırsa, ancak güncel olamazsa, başlangıç başarısız olur.

## Etkin kanal sayısı üst sınırının belirlenmesi

Sisteminizin birçok başlangıç kanalı tarafından aşırı yüklenmesini önlemek için etkin kanal sayısı üst sınırını da belirleyebilirsiniz. Bu yöntemi kullanırsanız, diğer kanallar sona erer ermez bekleme kanallarının başlamasına izin vermek için bağlantı kesme aralığı öznelikliğini düşük bir değere ayarlayın.


Ortağıyla bağlantı kurmaya çalışan bir kanal her zaman etkin bir kanal olmalıdır. Girişim başarısız olursa, sonraki deneme zamanı gelene kadar etkin olmayan geçerli bir kanal olarak kalır. Bir kanalın yeniden deneme sayısı ve ne sıklıkta yeniden deneme aralığı kanal özneliklerine göre belirlenir. Bu özneliklerin her ikisi için de kısa ve uzun değerler vardır. Ek bilgi için [Kanal öznelikleri](#) konusuna bakın.

Bir kanal etkin bir kanal olmak zorunda olduğunda (bir START komutu verildiği için ya da tetiklendiği için ya da başka bir yeniden deneme girişiminin zamanı geldiği için), ancak etkin kanal sayısı üst sınır değerinde olduğu için kanal, etkin yuvalardan biri etkin olmayı durduran başka bir kanal örneği tarafından serbest bırakılana kadar bekler. Ancak, bir kanal uzaktan başlatıldığı için başlatılırsa ve o sırada kanal için kullanılabilir etkin yuva yoksa, uzaktan başlatma reddedilir.

İstekçi kanalı dışında bir kanal etkin olmaya çalıştığında, STARTING durumuna geçer. Bu durum, yalnızca kısa bir süre STARTING durumunda olsa da hemen kullanılabilir etkin bir yuva olsa da ortaya çıkar. Ancak, kanalın etkin bir yuvayı beklemesi gerekirse, beklerken STARTING durumunda olur.

İstekte bulunan kanallar STARTING durumuna geçmez. Etkin kanal sayısı zaten sınırdaki olduğu için bir istek kanalı başlatılamıyorsa, kanal olağandışı sona erer.

İstekte bulunan kanal dışında bir kanal etkin bir yuva alamadığında ve bu nedenle bir yuva için

beklediğinde, günlükte  ya da z/OS konsola bir ileti yazılır ve bir olay oluşturulur. Daha sonra bir yuva serbest bırakıldığında ve kanal bunu elde edebiliyorsa, başka bir ileti ve olay oluşturulur. Kanal hemen bir yuva edinebiliyorsa, bu olayların ve iletilerin hiçbiri oluşturulmaz.

Kanal etkin olmayı beklerken bir STOP CHANNEL komutu verilirse, kanal DURDURULDU durumuna geçer. Kanal durdurulmuş bir olay oluştu.

Sunucu bağlantısı kanalları, etkin kanal sayısı üst sınırına dahil edilmiştir.

Etkin kanal sayısı üst sınırını belirtme hakkında daha fazla bilgi için aşağıdaki konulara bakın:

- **ALW** [IBM MQ Yönetimi](#).
- **IBM i** [IBM MQ for IBM i Yönetimi](#).
- **z/OS** [IBM MQ for z/OS Yönetimi](#).

#### *Kanal hataları*

Kanallardaki hatalar, kanalın diğer iletimleri durdurmasına neden olur. Kanal bir gönderen ya da sunucuya, sorunun kendisini temizlemesi olabileceğinden, kanal REPLY durumuna geçer. Yeniden DENEYİN durumuna geçemezse, kanal DURDURULDU durumuna geçer.

Kanalları göndermek için, ilişkili iletim kuyruğu GET (DISABLED) olarak ayarlanır ve tetikleme kapanır. (STATUS (DURDURULDU) durumlu bir STOP komutu, DURDURULDU durumuna verilen tarafı alır; yalnızca bağlantı kesme aralığının ya da STATUS (INACTIVE) durumuna sahip bir STOP komutunun süresi, olağan şekilde sona ermesini ve etkin olmamasını sağlar.) DURDURULDU durumundaki kanallar yeniden başlatılmadan önce işletmen müdahalesine gerek duyar (bkz. [“Durdurulan kanallar yeniden başlatılıyor”](#) sayfa 228).

**Not:** **IBM i** IBM i, AIX, Linux, and Windows sistemlerinde, yeniden deneme girişiminde bulunabilmesi için bir kanal başlatıcının çalışıyor olması gerekir. Kanal başlatıcı kullanılmıyorsa, kanal devre dışı kalır ve el ile yeniden başlatılması gerekir. Kanalı başlatmak için bir komut dosyası kullanıyorsanız, komut dosyasını çalıştırmayı denemeden önce kanal başlatıcının çalıştığından emin olun.

Uzun yeniden deneme sayısı (LONGRTY) , yeniden denemenin nasıl çalıştığını açıklar. Hata temizlenirse, kanal otomatik olarak yeniden başlatılır ve iletim kuyruğu yeniden etkinleştirilir. Hata temizlenmeden yeniden deneme sınırına ulaşırsa, kanal DURDURULDU durumuna geçer. Durdurulan bir kanal, işletmen tarafından el ile yeniden başlatılmalıdır. Hata devam ediyorsa, yeniden denenmez. Başarılı bir şekilde başladığında, iletim kuyruğu yeniden etkinleştirilir.

**z/OS** Kanal başlatıcısı, kanal REDENYOR ya da DURDURULDU durumundayken durursa, kanal başlatıcısı yeniden başlatıldığında kanal durumu hatırlanır. Ancak kanal DURDURULDU durumundayken kanal başlatıcısı durursa, SVRCONN kanal tipine ilişkin kanal durumu ilk durumuna getirilir.

**Multi** Bir kanal REDENYOR ya da DURDURULDU durumundayken kuyruk yöneticisi durursa, kuyruk yöneticisi yeniden başlatıldığında kanal durumu hatırlanır. IBM MQ 8.0 ' den itibaren bu, SVRCONN kanalları için de geçerlidir. Daha önce, kanal DURDURULDU durumundayken kanal başlatıcısı durdurulduysa, SVRCONN kanal tipine ilişkin kanal durumu sıfırlanmıştı.

Bir kanal, kuyruk dolu olduğu ya da engellendiği için bir iletiyi hedef kuyruğa koyamazsa, kanal işlemi bir zaman aralığında (ileti-yeniden deneme aralığı öznitelğinde belirtilir) bir kez yeniden deneyebilir. Diğer bir seçenek olarak, hangi koşulların yeniden denemeye neden olduğunu ve yapılan deneme sayısını belirleyen kendi ileti-yeniden deneme çıkışını yazabilirsiniz. Kanal, ileti yeniden deneme aralığının bitmesini beklerken PAUSED durumuna geçer.

Kanal öznitelikleriyle ilgili bilgi için [Kanal öznitelikleri](#) ve ileti-yeniden deneme çıkışıyla ilgili bilgi için [İleti alışverişi kanalları için kanal çıkış programları](#) başlıklı konuya bakın.

#### ***Sunucu bağlantısı kanal sınırları***

**MAXINST** parametresiyle istemci uygulamalarının kuyruk yöneticisi kanal kaynaklarını tüketmesini önlemek ve tek bir istemci uygulamasının **MAXINSTC** parametresiyle sunucu bağlantısı kanal kapasitesini tüketmesini önlemek için sunucu bağlantısı kanal sınırları ayarlayabilirsiniz.

**MAXINST** ve **MAXINSTC** için **DEFINE CHANNEL** komutunu kullanın.

Tek bir kuyruk yöneticisinde herhangi bir zamanda toplam kanal sayısı üst sınırı etkin olabilir. Etkin kanal sayısı üst sınırına sunucu bağlantısı kanal örneklerinin toplam sayısı dahil edilir.

Başlatılabilecek bir sunucu bağlantısı kanalının eşzamanlı eşgörünüm sayısı üst sınırını belirlemezseniz, tek bir sunucu bağlantısı kanalına bağlanan tek bir istemci uygulaması kullanılabilir etkin kanal sayısı üst sınırını tüketebilir. Etkin kanal sayısı üst sınırına ulaşıldığında, kuyruk yöneticisinde başka kanalların başlatılmasını önler. Bu durumu önlemek için, hangi istemcinin başlatmış olduğuna bakılmaksızın, başlatılabilen tek bir sunucu bağlantısı kanalının eşzamanlı eşgörünümlerinin sayısını sınırlamanız gerekir.

Sınırın değeri, sunucu bağlantı kanalının şu anda çalışmakta olan eşgörünüm sayısının (sıfıra bile) altına düşürülürse, çalışan kanallar bundan etkilenmez. Çalışmakta olan eşgörünümlerin sayısının sınırın değerinden az olması için, var olan eşgörünümlerin yeterli sayıda olması duruncaya kadar yeni eşgörünümler başlatılamaz.

Ayrıca, birçok farklı istemci-bağlantı kanalı tek bir sunucu bağlantısı kanalına bağlanabilir. Başlatılabilecek tek bir sunucu bağlantısı kanalının, hangi istemci tarafından başlatıldığına bakılmaksızın, eşzamanlı olarak başlatılabilecek eşgörünümlerinin sayısı sınırı, herhangi bir istemcinin kuyruk yöneticisinin etkin kanal kapasitesi üst sınırını tüketmesini önler. Tek bir istemciden başlatılabilecek tek bir sunucu bağlantısı kanalının eşzamanlı eşgörünümlerinin sayısını da sınırlamazsanız, tek bir hatalı istemci uygulamasının tek bir sunucu bağlantısı kanalına ayrılan kanal kapasitesini tüketen ve bu nedenle kanalı kullanması gereken diğer istemcilerin bağlanmasını önleyen çok sayıda bağlantı açması mümkündür. Bu durumu önlemek için, tek bir istemciden başlatılabilen tek bir sunucu bağlantısı kanalının eşzamanlı eşgörünümlerinin sayısını sınırlamanız gerekir.

Tek tek istemci sınırının değeri, şu anda tek tek istemcilerden çalıştırılan sunucu bağlantısı kanalının eşgörünümlerinin sayısının altına düşerse, hatta sıfıra indirgenirse, çalışan kanallar bundan etkilenmez. Ancak, çalışmakta olan eşgörünümlerin sayısının bu değıştirgenin değerinden az olması için, o istemcideki yeterli sayıda var olan yönetim ortamının çalışması duruncaya kadar, sunucu bağlantısı kanalının yeni yönetim ortamları yeni sınırı aşan bir istemciden başlatılamaz.

### İlgili başvurular

[Kanal öznitelikleri ve kanal tipleri](#)

[KANAL TANIMLAYIN](#)

### ***Kanalın diğer ucunun hala kullanılabilir olup olmadığı denetleniyor***

Kanalın diğer ucunun kullanılabilir olup olmadığını denetlemek için sağlıklı işletim bildirim aralığını, canlı tutma aralığını ve alma zamanlaşımını kullanabilirsiniz.

### **Sağlıklı işletim bildirimleri**

[Heartbeat interval \(HBINT\)](#) içinde açıklandığı gibi, iletim kuyruğunda ileti olmadığına akışların gönderen MCA ' dan geçireceğini belirtmek için sağlıklı işletim bildirim aralığı kanal özniteliğini kullanabilirsiniz.

### **Canlı tut**

**z/OS** z/OS işletim sistemlerinde, iletim protokolü olarak TCP/IP kullanıyorsanız, **Keepalive** aralık kanalı özniteliği (**KAINT**) için bir değer de belirtebilirsiniz. **Keepalive** aralığına sağlıklı işletim bildirim aralığından daha yüksek bir değer ve bağlantı kesme değerinden daha küçük bir değer vermeniz önerilir. Bu özniteliği, [Canlı Tutma Aralığı \(KAINT\)](#) konusunda açıklandığı gibi, her kanal için bir zaman aşımı değeri belirtmek için kullanabilirsiniz.

**Multi** IBM i, AIX, Linux, and Windows sistemlerinde, TCP ' yi iletim protokolünüz olarak kullanıyorsanız, **keepalive=yes** değerini ayarlayabilirsiniz. Bu seçeneği belirlerseniz, TCP düzenli aralıklarla bağlantının diğer ucunun kullanılabilir olup olmadığını denetler. Değil, kanal sonlandırıldı. Bu seçenek [Keepalive Interval \(KAINT\)](#) (KAINT) içinde açıklanmaktadır.

TCP hatalarını bildiren güvenilir olmayan kanallarınız varsa, **Keepalive** seçeneğinin kullanılması kanallarınızın kurtarılması olasılığının daha yüksek olduğu anlamına gelir.

**Keepalive** seçeneğinin davranışını denetlemek için zaman aralıkları belirtebilirsiniz. Zaman aralığını değıştirdiğinizde, değışiklik etkilendikten sonra yalnızca TCP/IP kanalları başlatılır. Zaman aralığı için seçtiğiniz değerin, kanala ilişkin bağlantı kesme aralığı değerinden küçük olduğundan emin olun.

**Keepalive** seçeneğinin kullanılmasıyla ilgili daha fazla bilgi için **DEFINE CHANNEL** komutundaki **KAINT** parametresine bakın.

## Alma zamanasını

İletim protokolünüz olarak TCP kullanıyorsanız, bir dönem için veri alınmazsa, boшта duran MQI olmayan kanal bağlantısının alma sonu da kapatılır. Bu dönem, *alma zamanasını* değeri, **HBINT** (sağlıklı işletim bildirim aralığı) değerine göre belirlenir.

IBM MQ for IBM i, AIX, Linux, and Windows sistemlerinde *receive time-out* değeri aşağıdaki gibi ayarlanır:

1. İlk akış sayısı için, herhangi bir anlaşma gerçekleşmeden önce, *alma zaman sınırı* değeri, kanal tanımından **HBINT** değerinin iki katıdır.
2. Kanallar bir **HBINT** değeri kararlaştırdıktan sonra, **HBINT** 60 saniyeden kısa bir değere ayarlanırsa, *alma zaman aşımına* değeri bu değer iki katına ayarlanır. **HBINT** 60 saniye ya da daha fazla olarak ayarlanırsa, *alma zaman aşımına* değeri, **HBINT** değerinden 60 saniye daha büyük olarak ayarlanır.

**z/OS** z/OS işletim sistemlerinde *alma zamanını* değeri aşağıdaki gibi ayarlanır:

1. İlk akış sayısı için, herhangi bir anlaşma gerçekleşmeden önce, *alma zaman sınırı* değeri, kanal tanımından **HBINT** değerinin iki katıdır.
2. **RCVTIME** ayarlanırsa, zamanasını **RCVTYPE** parametresine bağlı olarak aşağıdaki değerlerden birine ayarlanır ve geçerliyse, **RCVMTN** tarafından zorunlu kılınan herhangi bir sınıra tabidir:
  - Anlaşmalı **HBINT**, bir sabitle çarpılır
  - Üzerinde anlaşmaya varılan **HBINT** artı sabit saniye sayısı
  - Sabit saniye sayısı

**RCVMTN**, **RCVTYPE (EQUAL)** yapılandırıldığında geçerli değildir. **RCVTIME** sabit değerini kullanırsanız ve bir sağlıklı işletim bildirim aralığı kullanırsanız, sağlıklı işletim bildirim aralığından daha küçük bir **RCVTIME** belirtmeyin. **RCVTIME**, **RCVMTN** ve **RCVTYPE** özneliklerine ilişkin ayrıntılar için **ALTER QMGR** komutuna bakın.

### Not:

1. Değerlerden biri sıfırda, zamanasını yoktur.
2. Sağlıklı işletim sinyallerini desteklemeyen bağlantılar için, **HBINT** değeri adım 2 'de sıfır olarak kararlaştırılır ve bu nedenle zamanasını yoktur, bu nedenle TCP/IP **KEEPALIVE** değerini kullanmanız gerekir.
3. Paylaşım etkileşimlerini kullanan istemci bağlantıları için, sağlıklı işletim bildirimleri yalnızca bir MQGET sıradayken değil, kanalda (her iki uçtan) her zaman akabilir.
4. Paylaşım etkileşimleri kullanılmayan istemci bağlantıları için, sağlıklı işletim bildirimleri yalnızca istemci bir MQGET çağrısına bekleme verdiğinde sunucudan taşar. Bu nedenle, sağlıklı işletim bildirim aralığını istemci kanalları için çok küçük olarak ayarlamamız önerilmez. Örneğin, sağlıklı işletim bildirim 10 saniye olarak ayarlanırsa, MQCMIT çağrısının kesinleştirilmesi 20 saniyeden uzun sürerse (**MQRC\_CONNECTION\_BROKEN** ile), bu süre içinde veri akışı olmadığı için bir MQCMIT çağrısı başarısız olur. Bu, büyük iş birimlerinde gerçekleşebilir. Ancak, sağlıklı işletim bildirim aralığı için uygun değerler seçilirse bu gerçekleşmez; yalnızca bekleme içeren MQGET önemli dönemler alır.

**SHARECNV** sıfır değilse, istemci tam çift yönlü bağlantı kullanır; bu da, istemcinin tüm MQI çağrıları sırasında sağlıklı işletim bildirim alabileceği (ve yapabileceği) anlamına gelir.

5. Sağlıklı işletim bildirim aralığının iki katından sonra bağlantının iptal edilmesi, en az her sağlıklı işletim bildirim aralığında bir veri ya da sağlıklı işletim bildirim akışı beklendiği için geçerlidir. Sağlıklı işletim bildirim aralığının çok küçük bir değere ayarlanması, özellikle kanal çıkışlarını kullanıyorsanız sorunlara neden olabilir. Örneğin, **HBINT** değeri bir saniyeye ve bir gönderme ya da alma çıkışı kullanılırsa, alma ucu kanalı iptal etmeden önce yalnızca 2 saniye bekler. MCA iletiyi şifreleme gibi bir görevi gerçekleştiriyorsa, bu değer çok kısa olabilir.

## Önerilen ayarlar

### z/OS IBM MQ for z/OS

Başlangıç başlangıç noktası olarak şunları kullanabilirsiniz:

```
/cpe ALTER QMGR TCPKEEP(YES) RCVTTYTYPE(ADD) RCVTIME(60) ADOPTMCA(ALL) ADOPTCHK(ALL)
```

Burada cpe , kuyruk yöneticisi altsistemine ilişkin komut önekidir.

Çeşitli parametrelerle ilgili daha fazla bilgi için bkz. [ALTER QMGR](#) ve [IBM MQ ağ kullanılabilirliği](#) .

Gönderenin IP adresi birden çok adrese çevrilebilirse, **ADOPTCHK** parametresini ALLyerine QMNAME değerine ayarlamamız gerekebilir.

### Multi IBM MQ for Multiplatforms

qm.ini içine aşağıdaki bilgileri ekleyin:

```
TCP:  
KeepAlive=Yes  
CHANNELS:  
AdoptNewMCA=ALL  
AdoptNewMCACheck=ALL
```

Daha fazla bilgi için bkz. [ALTER QMGR](#), [Dağıtılmış kuyruğa alma için yapılandırma dosyası stanzasve "qm.ini dosyasının kanal kısmı" sayfa 115](#) .

Gönderenin IP adresi birden çok adrese çevrilebilirse, **AdoptNewMCACheck** değerini ALLyerine QMNAME olarak ayarlamamız gerekebilir.

### **MCA benimsenmesi**

MCA 'yı Benimse işlevi, IBM MQ ' in bir alıcı kanalı iptal etmesini ve yerine yeni bir kanal başlatmasını sağlar.

Bir kanal temas kaybederse, alıcı kanal 'iletişim alma' durumunda bırakılabilir. İletişim yeniden kurulduğunda gönderen kanal yeniden bağlanmayı dener. Uzak kuyruk yöneticisi alıcı kanalının zaten çalıştığını öğrenirse, aynı alıcı kanalının başka bir sürümünün başlatılmasına izin vermez. Bu sorun, sorunu düzeltmek ya da sistemin canlı olarak kullanılmasını sağlamak için kullanıcı müdahalesi gerektirir.

MCA 'yı Benimse işlevi sorunu otomatik olarak çözer. IBM MQ ' in bir alıcı kanalını iptal etmesini ve yerine yeni bir alıcı kanalı başlatmasını sağlar.

### **İlgili görevler**

[YönetmeIBM MQ](#)

[YönetmeIBM MQ for z/OS](#)

[YönetmeIBM MQ for IBM i](#)

### **Kanalları durdurma ve susturma**

Bağlantı kesme zaman aralığı sona ermeden önce bir kanalı durdurabilir ve durdurabilirsiniz.



İleti kanalları, yalnızca bağlantı kesme aralığı kanal özniteliği tarafından denetlenen düzenli sonlandırma ile kuyruk yöneticileri arasında uzun süreli bağlantılar olacak şekilde tasarlanmıştır. Bu mekanizma, işletmenin bağlantı kesme zaman aralığı sona ermeden önce kanalı sonlandırması gerekmedikçe iyi çalışır. Bu gereksinim aşağıdaki durumlarda ortaya çıkabilir:

- Sistemi susturma
- Kaynak koruma
- Bir kanalın bir ucunda tek taraflı eylem

Bu durumda kanalı durdurabilirsiniz. Bunu aşağıdaki işlemleri kullanarak yapabilirsiniz:

- STOP CHANNEL MQSC komutu

- Stop Channel PCF komutu
- IBM MQ Gezginini

-   Aşağıdaki gibi platforma özgü diğer mekanizmalar:

 **z/OS için:**

Kanal durdurulması panosu

 **IBM için:**

WRKMQMCHL panosundaki ENDMQMCHL CL komutu ya da END seçeneği


Bu komutları kullanarak kanalları durdurmak için üç seçenek vardır:

### QUIESCE

QUIESCE seçeneği, kanalı durdurmadan önce yürürlükteki ileti kümesini sona erdirmeyi dener.


### ZORLA

FORCE seçeneği kanalı hemen durdurmayı dener ve kanal yeniden başlatıldığında kanalın yeniden eşzamanlanması gerekebilir.

 IBM MQ for z/OS üzerinde, FORCE devam eden herhangi bir ileti yeniden ayrılmasını kesintiye uğratır; bu, BIND\_NOT\_FIXED iletilerini kısmen yeniden tahsis ya da sıradışı bırakabilir.

### TERMINATE

TERMINATE seçeneği kanalı hemen durdurmayı dener ve kanalın iş parçacığını ya da işlemini sona erdirir.

 IBM MQ for z/OS üzerinde, TERMINATE, BIND\_NOT\_FIXED iletilerini kısmen yeniden tahsis ya da sıradışı bırakabilecek, devam eden ileti yeniden ayrılmasını kesintiye uğratır.

Tüm bu seçenekler kanalı DURDURULDU durumunda bırakır ve yeniden başlatmak için işletmen müdahalesi gerektirir.

Kanalın gönderme ucunda durdurulması etkili olur, ancak işletmen müdahalesinin yeniden başlatılmasını gerektirir. Kanalın alıcı ucunda, MCA gönderen taraftan veri beklediği ve alıcı taraftan kanalın *sıralı* sonlandırmasını başlatmanın bir yolu olmadığı için işler çok daha zordur; MCA, veri bekleme durumundan dönünceye kadar durdurma komutu beklemede kalır.

Sonuç olarak, gerekli işletim özelliklerine bağlı olarak kanalları kullanmanın üç önerilen yolu vardır:

- Kanallarınızın uzun süre çalışmasını istiyorsanız, yalnızca gönderen uçtan düzgün bir şekilde sonlandırma olabileceğini unutmayın. Kanallar kesildiğinde, yani durdurulduğunda, kanalları yeniden başlatmak için işletmen müdahalesi (START CHANNEL komutu) gerekir.
- Kanallarınızın yalnızca iletilecek iletiler olduğunda etkin olmasını istiyorsanız, bağlantı kesme aralığını oldukça düşük bir değere ayarlayın. Varsayılan ayar yüksektir ve bu denetim düzeyinin gerekli olduğu kanallar için önerilmez. Alıcı kanalı kesmek zor olduğu için, en ekonomik seçenek, iş yükü talepleri doğrultusunda kanalın otomatik olarak bağlantısını kesmesi ve yeniden bağlanmasıdır. Çoğu kanal için, bağlantı kesme aralığının uygun ayarı buluşsal olarak oluşturulabilir.
- Sağlıklı işletim bildirim aralığı özneliğini, gönderen MCA 'nın gönderilecek ileti olmadığı dönemlerde alan MCA 'ya sağlıklı işletim bildirim akışı göndermesine neden olmak için kullanabilirsiniz. Bu işlem, alan MCA 'yı bekleme durumundan çıkarır ve bağlantı kesme aralığının süresinin dolmasını beklemeden kanalı susturma fırsatı verir. Sağlıklı işletim bildirim aralığına, bağlantı kesme aralığının değerinden düşük bir değer verin.

### Not:

1. Bağlantı kesme aralığını düşük bir değere ayarlamanız ya da sunucu kanalları için sağlıklı işletim bildirim kullanmanız önerilir. Bu düşük değer, sunucu kanalı için gönderilecek ileti olmadığı istekte bulunan kanalın olağandışı bitmesine (örneğin, kanal iptal edildiği için) izin verilmesini sağlar. Bağlantı kesme aralığı yüksek olarak ayarlanırsa ve sağlıklı işletim bildirimleri kullanımda değilse, sunucu istekte bulunanın sona erdiğini algılamaz (yalnızca istekte bulunana bir ileti göndermeyi bir sonraki denemesinde bunu yapar). Sunucu çalışmaya devam ederken, kuyruğa gelen diğer iletileri almak için iletim kuyruğunu dışlayıcı giriş için açık tutar. Kanalı istekte bulunandan yeniden başlatma



girişiminde bulunulursa, sunucu dışlayıcı giriş için iletim kuyruğunu hala açık olduğu için başlatma isteği bir hata alır. Sunucu kanalının durdurulması ve daha sonra, isteği sunan kanaldan yeniden başlatılması gerekir.

### **Durdurulan kanallar yeniden başlatılıyor**

Bir kanal DURDURULDU durumuna geçtiğinde, kanalı el ile yeniden başlatmanız gerekir.

### **Bu görev hakkında**

Gönderen ya da sunucu kanalları için, kanal DURDURULDU durumuna girdiğinde, ilişkili iletim kuyruğu GET (DISABLED) olarak ayarlandı ve tetikleme geçersiz kılındı. Başlatma isteği alındığında, bu öznelikler otomatik olarak ilk durumuna getirilir.

**z/OS** Kanal başlatıcısı, kanal REDENYOR ya da DURDURULDU durumundayken durursa, kanal başlatıcısı yeniden başlatıldığında kanal durumu hatırlanır. Ancak kanal başlatıcı DURDURULDU durumundayken durursa, SVRCONN kanal tipine ilişkin kanal durumu ilk durumuna getirilir.

**Multi** Bir kanal REDENYOR ya da DURDURULDU durumundayken kuyruk yöneticisi durursa, kuyruk yöneticisi yeniden başlatıldığında kanal durumu hatırlanır. IBM MQ 8.0 ' den itibaren bu, SVRCONN kanalları için de geçerlidir. Daha önce, kanal DURDURULDU durumundayken kanal başlatıcısı durdurulduysa, SVRCONN kanal tipine ilişkin kanal durumu ilk durumuna getirilmişti.

### **Yordam**

- Kanalı aşağıdaki yollardan biriyle yeniden başlatın:
  - [START CHANNEL MQSC komutunu](#) kullanarak.
  - [Start Channel PCF \(Kanal PCF ' yi Başlat\) komutunu](#) kullanarak.
  - [IBM MQ Explorer](#) kullanarak
  - **z/OS** z/OS sistemlerinde gerekir olarak bakın şunun için [Start a channel panel](#) (Kanal panosu başlat).
  - **IBM i** IBM sistemlerinde, [WRKMQMCHL panosunda STRMQMCHL CL komutunu](#) ya da [START](#) seçeneğini kullanarak.

### **Belirsiz kanalların işlenmesi**

Belirsiz bir kanal, iletilerin gönderildiği ve alındığı uzak bir kanalla ilgili şüphe içinde olan bir kanaldır.

### **Bu görev hakkında**

Bu ve bir kuyruk yöneticisi arasındaki farkın, hangi iletilerin bir kuyruksa kesinleştirileceği konusunda şüphe içinde olduğunu göz önünde bulundurun.

Batch Heartbeat kanal parametresini (**BATCHHB**) kullanarak bir kanalın şüpheye düşme fırsatını azaltabilirsiniz. Bu parametre için bir değer belirlendiğinde, gönderen kanal, daha fazla işlem yapmadan önce uzak kanalın etkin olup olmadığını denetler. Herhangi bir yanıt alınmazsa, alıcı kanal artık etkin olarak kabul edilir. İletiler geriye işlenebilir ve yeniden yönlendirilebilir ve gönderen kanalı şüphe içine alınmaz. Bu, kanalın, alıcı kanalının hala etkin olduğunu doğrulayan ve alıcı kanalının gönderilen iletileri aldığını doğrulayan gönderen kanal arasındaki döneme yerleştirilme süresini kısaltır. Toplu sağlıklı işletim bildirim parametresiyle ilgili daha fazla bilgi için bkz. [Kanal öznelikleri](#) .

Belirsiz kanal sorunları genellikle otomatik olarak çözülür. İletişim kesildiğinde ve bir kanal, giriş durumu bilinmeyen bir ileti grubu ile kuşkulandığında bile, iletişim yeniden kurulduğunda durum çözülür. Sıra numarası ve LUWID kayıtları bu amaçla saklanır. Kanal, LUWID bilgileri değiş tokuş edilinceye kadar şüphe içinde kalır ve kanal için yalnızca bir ileti kümesi şüpheli olabilir.

Gerektiğinde, kanalı el ile yeniden eşitleyebilirsiniz. Terim kılavuzu, IBM MQ sistem yönetimi komutlarını içeren işletmenlerin ya da programların kullanımını içerir. El ile yeniden eşzamanlama işlemi aşağıdaki gibi çalışır. Bu tanım MQSC komutlarını kullanır, ancak PCF eşdeğerlerini de kullanabilirsiniz.



## Yordam

1. Kanalın her bir tarafına ilişkin son kesinleştirilen mantıksal iş birimi tanıtıcısını (LUWID) bulmak için **DISPLAY CHSTATUS** komutunu kullanın.

Bunu aşağıdaki komutları kullanarak yapın:

- Kanalın belirsiz tarafı için:

```
DISPLAY CHSTATUS(name) SAVED CURLUWID
```

Kanalı daha ayrıntılı tanımlamak için **CONNAME** ve **XMITQ** parametrelerini kullanabilirsiniz.

- Kanalın alıcı tarafı için:

```
DISPLAY CHSTATUS( name ) SAVED LSTLUWID
```

Kanalı daha ayrıntılı tanımlamak için **CONNAME** parametresini kullanabilirsiniz.

**Not:** Komutlar farklıdır, çünkü kanalın yalnızca gönderen tarafı şüphe içinde olabilir. Alan taraf asla şüphe içinde değildir.

**IBM i** IBM işletim sistemlerinde, **DISPLAY CHSTATUS** komutu **STRMQMMQSC** komutu ya da **WRKMQMCHSTMQM** Kanal Durumu CL komutu kullanılarak bir dosyadan yürütülebilir.

2. İki LUWID aynıysa, belirsiz iletileri kesinleştirmek için **RESOLVE CHANNEL** komutunu kullanın.

İki LUWID aynıysa, alıcı taraf, gönderenin şüphe içinde olduğunu düşündüğü iş birimini kesinleştirmiştir. Gönderen taraf artık belirsiz iletileri iletim kuyruğundan kaldırabilir ve yeniden etkinleştirebilir. Bu, aşağıdaki **RESOLVE CHANNEL** komutuyla yapılır:

```
RESOLVE CHANNEL(name) ACTION(COMMIT)
```

3. İki LUWID farklıysa, belirsiz iletileri geri almak için **RESOLVE CHANNEL** komutunu kullanın.

İki LUWID farklıysa, alıcı taraf, gönderenin şüphe içinde olduğunu düşündüğü iş birimini kesinleştirmez. Gönderen tarafın, ileti kuyruğundaki belirsiz iletileri saklaması ve yeniden göndermesi gerekir. Bu, aşağıdaki **RESOLVE CHANNEL** komutuyla yapılır:

```
RESOLVE CHANNEL( name ) ACTION(BACKOUT)
```

**IBM i** IBM i' de, **RSVMQMCHLMQM** Kanalını Çözümle komutunu kullanabilirsiniz.

## Sonuçlar

Bu işlem tamamlandığında kanal artık kuşku duymaz. İletim kuyruğu, gerekiyorsa, artık başka bir kanal tarafından kullanılabilir.

### İlgili başvurular

[DISPLAY CHSTATUS \(KANAL DURUMUNU GÖRÜNTÜLE\)](#)

[RESOLVE CHANNEL \(Bir kanaldan belirsiz iletileri çözmesini isteyin\)](#)

## İletilerin güvenliği

Dağıtılmış kuyruk yönetimi, IBM MQürününün tipik kurtarma özelliklerine ek olarak, ileti kanalının iki ucu arasında eşgüdümlü bir eşitleme noktası yordamı kullanılarak iletilerin düzgün şekilde teslim edilmesini sağlar. Bu yordam bir hata saptarsa, sorunu araştırmanız için kanalı kapatır ve kanal yeniden başlatılıncaya kadar iletileri güvenli bir şekilde iletim kuyruğunda tutar.

Eşitleme noktası yordamı, kanal başlatıldığında *belirsiz* durumu kurtarmaya çalışacağından ek bir avantaja sahiptir. (*belirsiz*, bir eşitleme noktasının istendiği, ancak isteğin sonucunun henüz bilinmediği bir kurtarma biriminin durumudur.) Bu olanakla ilişkili iki işlev de vardır:

1. Kesinleřtirme ya da geriletme ile çözümler
2. Sıra numarasını sıfırla

Bu işlevlerin kullanımı, kanal çoęu durumda otomatik olarak iyileřtięi için yalnızca istisnai durumlarda gerçekteşir.

### **Hızlı, kalıcı olmayan iletiler**

Kanaldaki kalıcı olmayan iletilerin daha hızlı teslim edileceęini belirtmek için kalıcı olmayan ileti hızı (NPMSPEED) kanal öznitelięi kullanılabilir. Bu öznitelikle ilgili daha fazla bilgi için bkz. [Kalıcı olmayan ileti hızı \(NPMSPEED\)](#).

Bir kanal hızlı, kalıcı olmayan iletiler taşınırken sona ererse, iletiler kaybolabilir ve gerektiğinde bunların kurtarılmasını düzenlemek uygulamaya kalmıř olabilir.

Alıcı kanal iletiyi hedef kuyruęuna yerleřtiremezse, ileti tanımlanmıřsa, gitmeyen iletiler kuyruęuna yerleřtirilir. Deęilse, ileti atılır.

**Not:** Kanalın dięer ucu seçeneęi desteklemiyorsa, kanal normal hızda çalıřır.

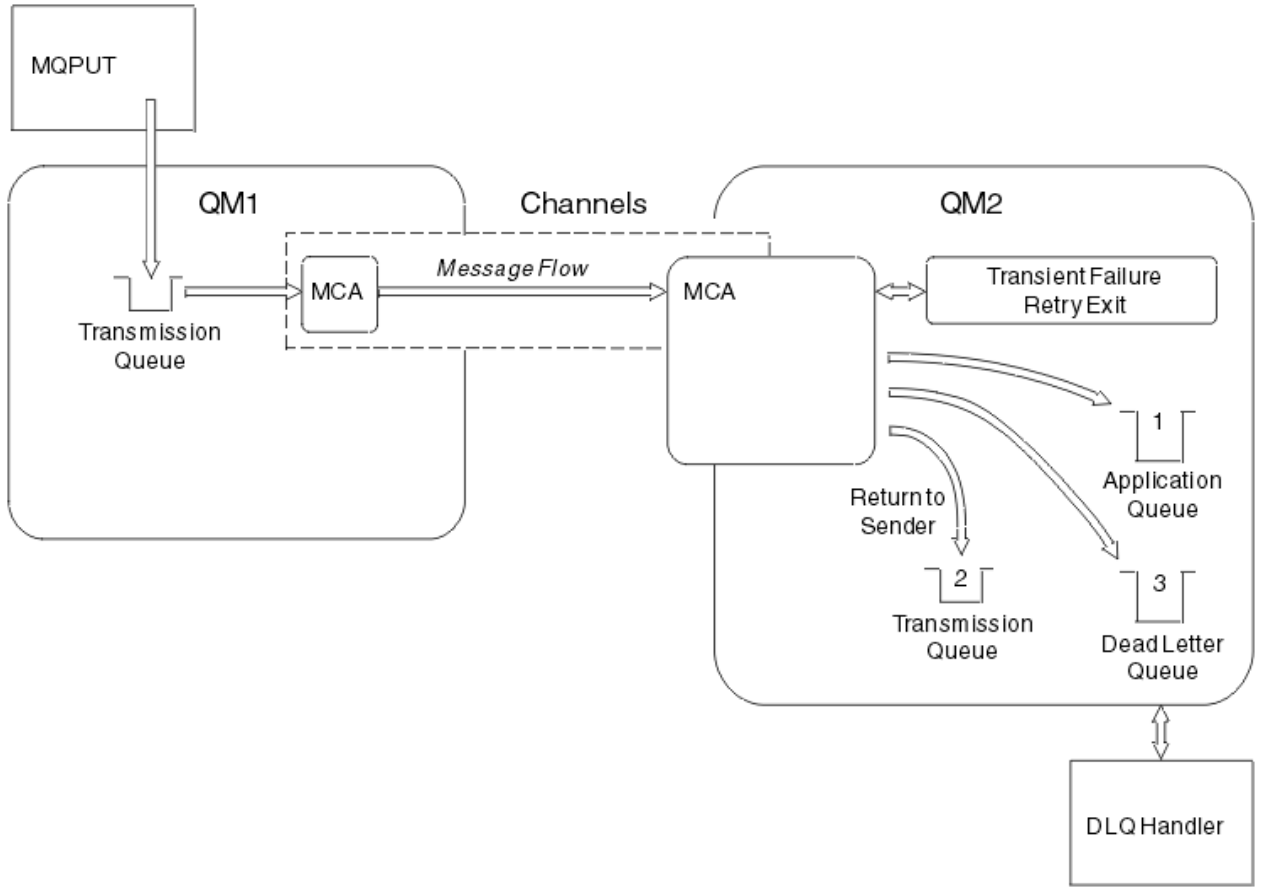
### **Teslim edilmeyen İletiler**

Bir ileti teslim edilemedięinde ne olacaęı hakkında bilgi için bkz. [“Bir ileti teslim edilemedięinde ne olur?” sayfa 230](#).

### **Bir ileti teslim edilemedięinde ne olur?**

Bir ileti teslim edilemedięinde, MCA bunu birkaç řekilde işleyebilir. Yeniden deneyebilir, gönderene geri döndürebilir ya da göndermeyen iletiler kuyruęuna koyabilir.

[Şekil 23 sayfa 231](#) , bir MCA bir iletiyi hedef kuyruęa koyamadıęında oluřan işlemleri gösterir. (Gösterilen seçenekler tüm platformlarda geçerli deęildir.)



Şekil 23. Bir ileti teslim edilemediğinde ne olur?

Şekilde gösterildiği gibi MCA, teslim edemediği bir mesajla birkaç şey yapabilir. Yapılan işlem, kanal tanımlandığında ve iletiye ilişkin MQPUT rapor seçeneklerinde belirtilen seçeneklere göre belirlenir.

#### 1. İleti-yeniden dene

MCA, geçici olabilecek bir nedenle (örneğin, kuyruk dolu olduğu için) hedef kuyruğa bir ileti yerleştiremezse, MCA bekleyebilir ve işlemi daha sonra yeniden deneyebilir. MCA 'nın bekleyip beklemeyeceğini, ne kadar süre ve kaç kez deneyeceğini belirleyebilirsiniz.

- Kanalınızı tanımlarken MQPUT hataları için bir ileti yeniden deneme süresi ve aralığı belirtebilirsiniz. Kuyruk dolu olduğu için ya da girişler için engellendiği için ileti hedef kuyruğa konamazsa, MCA, belirtilen zaman aralığında işlemi belirtilen sayıda dener.
- Kendi iletinizi yazabilirsiniz-çıkışı yeniden deneyin. Çıkış, MCA 'nın MQPUT ya da MQOPEN işlemini yeniden denemesini istediğiniz koşulları belirtmenizi sağlar. Kanalı tanımlarken çıkışın adını belirtin.

#### 2. Gönderene geri dön

İleti-yeniden deneme başarısız olduysa ya da farklı bir hata tipiyle karşılaşıldıysa, MCA iletiyi kaynak kişiye geri gönderebilir. Gönderene geri dönmeyi etkinleştirmek için, iletiyi özgün kuyruğa yerleştirirken ileti tanımlayıcıda aşağıdaki seçenekleri belirlemeniz gerekir:

- MQRO\_EXCEPTION\_WITH\_FULL\_DATA rapor seçeneği
- MQRO\_DISCARD\_MSG rapor seçeneği
- Yanıtın gönderileceği kuyruk ve yanıtın gönderileceği kuyruk yöneticisinin adı

MCA iletiyi hedef kuyruğa koyamazsa, özgün iletiyi içeren bir kural dışı durum raporu oluşturur ve özgün iletide belirtilen yanıt kuyruğuna gönderilecek bir iletim kuyruğuna koyar. (Yanıt kuyruğu MCA ile aynı kuyruk yöneticisiyse, ileti bir iletim kuyruğuna değil, doğrudan o kuyruğa yerleştirilir.)

#### 3. Gönderilmeyen iletiler kuyruğu

Bir ileti teslim edilemezse ya da geri verilemezse, teslim edilmeyen ileti kuyruğuna (DLQ) yerleştirilir. İletiyi işlemek için DLQ işleyicisini kullanabilirsiniz. Bu işlem, IBM MQ for UNIX, Linux ve Windows sistemleri için Kuyruktaki İletilerin İşlenmesi başlıklı konuda ve z/OS sistemleri için The dead-letter queue handler utility (CSQUDLQH) adlı belgede açıklanmaktadır. Gitmeyen iletiler kuyruğu yoksa, gönderen MCA iletiyi iletim kuyruğuna bırakır ve kanal durur. Hızlı bir kanalda, ileti kuyruğuna yazılamayan kalıcı olmayan iletiler kaybolur.

IBM WebSphere MQ 7.0 işletim sistemi üzerinde, yerel bir gitmeyen iletiler kuyruğu tanımlanmamışsa, uzak kuyruk kullanılabilir ya da tanımlanmamışsa ve uzak gitmeyen iletiler kuyruğu yoksa, gönderen kanal RETRY ' a gider ve iletiler otomatik olarak iletim kuyruğuna geri alınır.

## İlgili başvurular

[Teslim Edilemeyen Mektup Kuyruğunu Kullan \(USEDLQ\)](#)

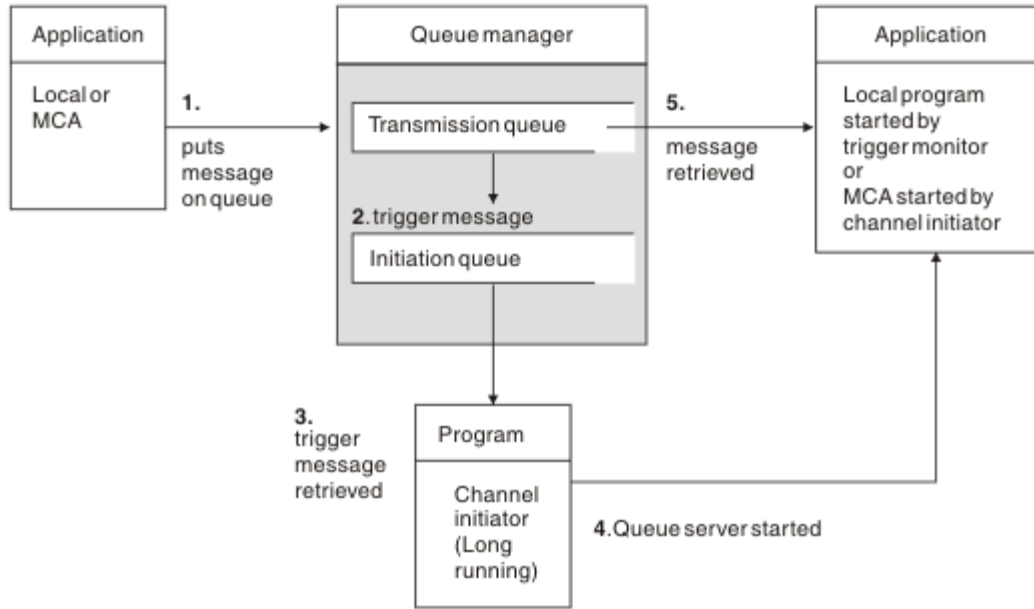
## Tetikleme kanalları

IBM MQ , bir kuyruktaki belirli koşullar karşılandığında uygulamayı otomatik olarak başlatmak için bir olanak sağlar. Bu tesise tetikleme denir.

Bu açıklama, tetikleyici kavramlara genel bir bakış olarak tasarlanmıştır. Tam açıklama için [Tetikleyiciler](#) kullanılarak IBM MQ uygulamalarının başlatılması başlıklı konuya bakın.

Platforma özgü bilgiler için aşağıdaki konulara bakın:

- AIX, Linux, and Windows için bkz. [“AIX, Linux, and Windows üzerinde tetikleyici kanallar.”](#) sayfa 233
- **IBM i** IBM için bkz. [“IBM MQ for IBM i içinde tetikleyici kanallar”](#) sayfa 234
- **z/OS** z/OS için bkz. [“İletim kuyrukları ve tetikleme kanalları”](#) sayfa 957





Şekil 24. Tetikleme kavramları

Tetikleme için gerekli nesnelere Şekil 24 sayfa 232’de gösterilir. Aşağıdaki olay sırasını gösterir:

1. Yerel kuyruk yöneticisi, iletim kuyruğuna bir uygulamadan ya da bir ileti kanalı aracısından (MCA) ileti yerleştirir.
2. Tetikleme koşulları yerine getirildiğinde, yerel kuyruk yöneticisi başlatma kuyruğuna bir tetikleyici ileti yerleştirir.
3. Uzun süre çalışan kanal başlatıcı programı başlatma kuyruğunu izler ve geldiklerinde iletileri alır.

4. Kanal başlatıcı, tetikleyici iletileri, içindeki bilgilere göre işler. Bu bilgiler kanal adını içerebilir; bu durumda ilgili MCA başlatılır.
5. Tetiklenmiş olan yerel uygulama ya da MCA, iletileri iletim kuyruğundan alır.

Bu senaryoyu ayarlamak için aşağıdakileri yapmanız gerekir:

- İletim kuyruğunu başlatma kuyruğunun adıyla ( SYSTEM.CHANNEL.INITQ).
- Başlatma kuyruğunun (SYSTEM.CHANNEL.INITQ) var.
- Kanal başlatıcı programının kullanılabilir ve çalışır durumda olduğundan emin olun. Kanal başlatıcı programı, başlatma komutunda başlatma kuyruğunun adıyla birlikte sağlanmalıdır.  z/OS' ta, başlatma kuyruğunun adı düzeltildiği için başlatma komutunda kullanılmaz.
- İsteğe bağlı olarak, tetikleme için süreç tanımlamasını yaratın (yoksa) ve *UserData* alanının, hizmet ettiği kanalın adını içerdiğinden emin olun. Bir işlem tanımı yaratmak yerine, iletim kuyruğunun **TriggerData** özneliğinde kanal adını belirtebilirsiniz. IBM MQ  IBM i, AIX, Linux, and Windows sistemleri için, kanal adının boş olarak belirlenmesine izin verin; bu durumda, bu iletim kuyruğuna sahip ilk kullanılabilir kanal tanımı kullanılır.
- İletim kuyruğu tanımlamasının, hizmet verecek süreç tanımlamasının adını (varsa), başlatma kuyruğu adını ve en uygun olduğuna hissettiğin tetikleme özelliklerini içerdiğinden emin olun. Tetikleyici denetim özneliği, tetikleme özelliğinin gerektiği şekilde etkinleştirilmesini ya da etkinleştirilmemesini sağlar.

#### Not:

1. Kanal başlatıcı programı, kanalları başlatmak için kullanılan başlatma kuyruğunu izleyen bir 'tetikleyici izleyici' görevi görür.
2. Başlatma kuyruğu ve tetikleme işlemi, herhangi bir sayıda kanalı tetiklemek için kullanılabilir.
3. Herhangi bir sayıda başlatma kuyruğu ve tetikleme işlemi tanımlanabilir.
4. Sistemin kanal ile dolmasını önlemek için FIRST tetikleyici tipi önerilir.

## AIX, Linux, and Windowsüzerinde tetikleyici kanallar.



Tetiklenecek süreçleri tanımlayarak IBM MQçinde bir süreç tanımlaması yaratabilirsiniz. İletiler bir iletim kuyruğuna geldiğinde tetiklenecek işlemi adlandırmak üzere bir süreç tanımlaması yaratmak için MQSC DEFINE PROCESS komutunu kullanın. İşlem tanımının USERDATA özneliği, iletim kuyruğunun hizmet verdiği kanalın adını içerir.

Kanalı başlatan uygulamayı tetiklemek için tetikleyici iletilerin başlatma kuyruğuna (IQ) yazılacağını belirterek yerel kuyruğu tanımlayın (QM4) (QM3.TO.QM4):

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(P1) USAGE(XMITQ)
```

Başlatılacak uygulamayı (işlem P1) tanımlayın:

```
DEFINE PROCESS(P1) USERDATA(QM3.TO.QM4)
```

Diğer bir seçenek olarak, IBM MQ for UNIX, Linux ve Windows sistemleri için, iletim kuyruğunun TRIGDATA özneliğinde kanal adını belirterek bir süreç tanımlaması gereksinimini ortadan kaldırabilirsiniz.

Yerel kuyruğu tanımlayın (QM4). Kanalı başlatan uygulamayı (işlem P1) tetiklemek için tetikleyici iletilerin varsayılan başlatma kuyruğuna SYSTEM.CHANNEL.INITQyazılacağını belirtin (QM3.TO.QM4):

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ)
USAGE(XMITQ) TRIGDATA(QM3.TO.QM4)
```

Bir kanal adı belirlemezseniz, kanal başlatıcı, adı belirtilen iletim kuyruğuyla ilişkilendirilmiş bir kanal buluncaya kadar kanal tanımlama dosyalarını arar.

## IBM MQ for IBM i içinde tetikleyici kanallar

IBM i

IBM MQ for IBM i içinde kanalların tetiklenmesi, kanal başlatıcı işlemiyle gerçekleştirilir. SYSTEM.CHANNEL.INITQ , kuyruk yöneticisi SCHINIT özneliği değiştirilerek geçersiz kılınmadıkça, kuyruk yöneticisiyle otomatik olarak başlatılır.

SYSTEM.CHANNEL.INITQ ve kuyruk için tetikleme etkinleştiriliyor. Kanal başlatıcı, bu iletim kuyruğunu belirleyen ilk kullanılabilir kanalı başlatır.

```
CRTMQMQ QNAME(MYXMITQ1) QTYPE(*LCL) MQMNAME(MYQMGR)
TRGENBL(*YES) INITQNAME(SYSTEM.CHANNEL.INITQ)
USAGE(*TMQ)
```

**Deprecated** STRMQMCHLI komutuyla en çok üç kanal başlatıcı işlemi el ile başlatabilir ve farklı başlatma kuyrukları belirtebilirsiniz. Ayrıca, iletim kuyruğunu işlemek için birden çok kanal belirleyebilir ve başlatılacak kanalı seçebilirsiniz. Bu yetenek, önceki yayınlarla uyumlu olmaya devam etmektedir. Kullanımı kullanımdan kaldırılmıştır.

**Not:** Bir iletim kuyruğunu aynı anda yalnızca bir kanal işleyebilir.

```
STRMQMCHLI QNAME(MYINITQ)
```

TRGENBL(\*YES) belirleyerek kanalın iletim kuyruğunu ayarlayın ve hangi kanalın başlatılacağını seçmek için TRIGDATA alanında kanal adını belirleyin. Örneğin:

```
CRTMQMQ QNAME(MYXMITQ2) QTYPE(*LCL) MQMNAME(MYQMGR)
TRGENBL(*YES) INITQNAME(MYINITQ)
USAGE(*TMQ) TRIGDATA(MYCHANNEL)
```

### İlgili kavramlar

[“Kanal başlatıcının başlatılması ve durdurulması” sayfa 234](#)

Tetikleme, kanal başlatıcı işlemi kullanılarak gerçekleştirilir.

### İlgili görevler

[“Dağıtılmış kuyruğa alma yapılandırılıyor” sayfa 189](#)

Bu bölümde, kuyruk tanımlaması, kanal tanımlaması, tetikleme ve eşitleme noktası yordamları da içinde olmak üzere IBM MQ kuruluşları arasındaki iletişikle ilgili daha ayrıntılı bilgi sağlanır.

### İlgili başvurular

[AIX, Linux, and Windows üzerinde kanal programları](#)

IBM i

[IBM i üzerinde iletişim işleri](#)

IBM i

[IBM i üzerindeki kanal durumları](#)

### Kanal başlatıcının başlatılması ve durdurulması

Tetikleme, kanal başlatıcı işlemi kullanılarak gerçekleştirilir.

Bu kanal başlatıcı işlemi MQSC komutu START CHINIT ile başlatılır. Varsayılan başlatma kuyruğunu kullanmıyorsan, komuttaki başlatma kuyruğunun adını belirleyin. Örneğin, varsayılan kuyruk yöneticisi için kuyruk IQ 'sunu başlatmak üzere START CHINIT komutunu kullanmak için şunu girin:

START CHINIT INITQ(IQ)

Varsayılan olarak, SYSTEM.CHANNEL.INITQ. Tüm kanal başlatıcılarınızı el ile başlatmak istiyorsanız aşağıdaki adımları izleyin:

1. Kuyruk yöneticisini yaratın ve başlatın.
2. Kuyruk yöneticisinin SCHINIT özelliğini MANUAL olarak değiştir
3. Kuyruk yöneticisini sona erdirin ve yeniden başlatın

IBM MQ for Multiplatforms sistemlerinde, bir kanal başlatıcı otomatik olarak başlatılır. Başlatabileceğiniz kanal başlatıcı sayısı sınırlıdır. Varsayılan değer ve üst sınır değeri 3 'tür. Bunu, AIX and Linux sistemleri için qm.ini dosyasında ve Windows sistemleri için kayıt defterinde MAXINITIATORS kullanarak değiştirebilirsiniz.

Kanal başlatıcı komutunu çalıştır **runmqchi** ve diğer denetim komutlarına ilişkin ayrıntılar için bkz. [IBM MQ Denetim komutları](#) .

## Kanal başlatıcıyı durdurma

Varsayılan kanal başlatıcısı, bir kuyruk yöneticisini başlattığınızda otomatik olarak başlatılır. Bir kuyruk yöneticisi durdurulduğunda tüm kanal başlatıcıları otomatik olarak durdurulur.

## Başlatma ve yapılandırma dosyaları

Kanal kullanıma hazırlama verilerinin işlenmesi IBM MQ platformunuza bağlıdır.

### IBM MQ for z/OS



IBM MQ for z/OS içinde, kullanıma hazırlama ve yapılandırma bilgileri **ALTER QMGR MQSC** komutu kullanılarak belirtilir. **ALTER QMGR** komutlarını CSQINP2 kullanıma hazırlama giriş veri kümesine koyarsanız, bunlar kuyruk yöneticisi her başlatıldığında işlenir.

Kanal başlatıcısını her başlattığınızda **START LISTENER** gibi MQSC komutlarını çalıştırmak için, bunları CSQINPX kullanıma hazırlama giriş verileri kümesine koyun ve kanal başlatıcı tarafından başlatılan görev yordamındaki isteğe bağlı DD deyimini CSQINPX 'i belirtin.

CSQINP2 ve CSQINPX ile ilgili ek bilgi için [Kullanıma hazırlama giriş veri kümelerinin uyarlanması ve ALTER QMGR](#) başlıklı konuya bakın.

### IBM MQ for Multiplatforms



IBM MQ for Multiplatforms içinde, IBM MQ kuruluşuna ilişkin temel yapılandırma bilgilerini bulunduracak yapılandırma dosyaları vardır.

İki yapılanış dosyası vardır: Biri makine için, diğeri tek bir kuyruk yöneticisi için geçerlidir.

#### IBM MQ yapılandırma dosyası

Bu dosya, IBM MQ sistemindeki tüm kuyruk yöneticileriyle ilgili bilgileri içerir. Dosyanın adı `mqc.ini`. Bu, "[IBM MQ yapılandırma dosyası, mqc.ini](#)" sayfa 85 içinde açıklanır.

#### Kuyruk yöneticisi yapılanış dosyası

Bu dosya, belirli bir kuyruk yöneticisiyle ilgili yapılanış bilgilerini içerir. Dosyanın adı `qm.ini`.

Kuyruk yöneticisi yaratılırken yaratılır ve kuyruk yöneticisinin herhangi bir yönüyle ilgili yapılanış bilgilerini tutabilir. Dosyada tutulan bilgiler, günlük yapılandırmasının IBM MQ yapılandırma dosyasındaki varsayılan yapılandırmadan nasıl farklı olduğuna ilişkin ayrıntıları içerir.


Kuyruk yöneticisi yapılanış dosyası, kuyruk yöneticisinin kapladığı dizin ağacının kökünde tutulur. Örneğin, **DefaultPath** öznitelikleri için, QMNAME adlı kuyruk yöneticisine ilişkin kuyruk yöneticisi yapılanış dosyaları şöyle olabilir:

AIX and Linux sistemleri için:

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

Windows sistemleri için:

```
C:\ProgramData\IBM\MQ\qmgrs\QMNAME\qm.ini
```

 IBM için:

```
/QIBM/UserData/mqm/qmgrs/QMNAME/qm.ini
```

Burada bir `qm.ini`'den alınan bir özet yer almaktadır. TCP/IP dinleyicisinin 2500 numaralı kapıda dinleyeceğini, geçerli kanal sayısı üst sınırının 200 olduğunu ve etkin kanal sayısı üst sınırının 100 olduğunu belirtir.

```
TCP:
Port=2500
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

Giden kanal tarafından kullanılacak bir TCP/IP kapısı aralığı belirtebilirsiniz. Bir yöntem, kapı değerleri aralığının başlangıcını ve sonunu belirtmek için `qm.ini` dosyasını kullanmaktır. Aşağıdaki örnekte, kanal aralığı belirten bir `qm.ini` dosyası gösterilmektedir:

```
TCP:
StrPort=2500
EndPort=3000
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

**StrPort** ya da **EndPort** için bir değer belirtirseniz, her ikisi için de bir değer belirtmeniz gerekir. **EndPort** değeri her zaman **StrPort** değerinden büyük olmalıdır.

Kanal, belirlenen aralıktaki kapı değerlerinin her birini kullanmayı dener. Bağlantı başarılı olduğunda, kapı değeri, kanalın daha sonra kullandığı kapıdır.

`qm.ini` dosyalarıyla ilgili daha fazla bilgi için bkz. [“Kuyruk yöneticisi yapılanış dosyaları, qm.ini” sayfa 97.](#)

## İletiler için veri dönüştürme

IBM MQ iletileri, farklı kuyruk yöneticilerindeki kuyruklar arasında gönderildiğinde veri dönüştürme gerektirebilir.

IBM MQ iletisi iki bölümden oluşur:

- İleti tanımlayıcıdaki denetim bilgileri
- Uygulama Verileri

İki bölümden biri, farklı kuyruk yöneticilerindeki kuyruklar arasında gönderildiğinde veri dönüştürme gerektirebilir. Uygulama verilerini dönüştürme hakkında bilgi için bkz. [Uygulama verilerini dönüştürme.](#)



## Kendi ileti kanalı aracılarınızı yazma

IBM MQ , kendi ileti kanalı aracı (MCA) programlarınızı yazmanızı ya da bağımsız bir yazılım satıcısından bir tane kurmanızı sağlar.

IBM MQ 'in kendi özel iletişim protokolünüz üzerinden birlikte çalışmasını sağlamak ya da IBM MQ ' in desteklemediği bir protokol üzerinden ileti göndermesini sağlamak için kendi MCA programlarınızı yazmak isteyebilirsiniz. (Diğer uçta IBM MQ tarafından sağlanan bir MCA ile birlikte çalışmak için kendi MCA ' ınızı yazamazsınız.)

IBM MQ tarafından sağlanmayan bir MCA kullanmaya karar verirseniz, aşağıdaki noktaları göz önünde bulundurmanız gerekir.

### İleti gönderme ve alma

İletileri uygulamanızın koyduğu yerden (örneğin, bir iletim kuyruğundan) alan ve iletişim kurmak istediğiniz bir iletişim kuralıyla gönderen bir gönderen uygulama yazmanız gerekir. Ayrıca, bu protokolden ileti alan ve bunları hedef kuyruklara koyan bir alıcı uygulama da yazmanız gerekir. Gönderen ve alan uygulamalar, özel arabirimler değil, ileti kuyruğu arabirimi (MQI) çağrılarını kullanır.

İletilerin yalnızca bir kez teslim edildiğinden emin olmanız gerekir. Bu teslimde yardımcı olmak için eşitleme noktası koordinasyonu kullanılabilir.

### Kanal denetimi işlevi

Kanalları denetlemek için kendi yönetim işlevlerinizi sağlamanız gerekir. Kanallarınızı yapılandırmak (örneğin, DEFINE CHANNEL komutu) ya da izlemek (örneğin, DISPLAY CHSTATUS) için IBM MQ kanal yönetimi işlevlerini kullanamazsınız.

### kullanıma hazırlama dosyası

Gerekliyse, kendi kullanıma hazırlama kütüğünüzü belirtmelisiniz.

### Uygulama verilerini dönüştürme

Büyük olasılıkla, farklı bir sisteme gönderdiğiniz iletiler için veri dönüştürmeye izin vermek istiyorsunuz. Bu durumda, uygulamanızın iletileri koyduğu yerden (örneğin, iletim kuyruğu) alırken MQGET çağrısında MQGMO\_CONVERT seçeneğini kullanın.

### Kullanıcı çıkışları

Kullanıcı çıkışlarına gerek duyup duymadığınızı göz önünde bulundurun. Bu durumda, IBM MQ ' in kullandığı aynı arabirim tanımlarını kullanabilirsiniz.

### Tetikleme

Uygulamanız iletileri bir iletim kuyruğuna koyarsa, ileti kuyruğuna geldiğinde MCA ' yı göndermeniz tetiklenecek şekilde iletim kuyruğu özniteliklerini ayarlayabilirsiniz.

### Kanal başlatıcı

Kendi kanal başlatıcınızı sağlamanız gerekebilir.

## Dağıtılmış kuyruk yönetimi için göz önünde bulundurulması gereken diğer konular

IBM MQ ' u dağıtılmış kuyruk yönetimi için hazırlarken dikkate alınacak diğer konular. Bu konu, Teslim Edilmeyen ileti kuyruğu, Kullanılmakta olan Kuyruklar, Sistem uzantıları ve kullanıcı çıkışı programları ve Güvenilen uygulamalar olarak Çalışan kanalları ve dinleyicileri kapsar.

## Teslim edilmedi-ileti kuyruğu

Teslim edilmeyen ileti kuyruğuna gelen iletilerin (kullanılmayan ileti kuyruğu ya da DLQ olarak da bilinir) işlenmesini sağlamak için, bu iletileri işlemek üzere düzenli aralıklarla tetiklenebilen ya da çalıştırılabilen bir program oluşturun.

**Linux** **AIX** AIX and Linux sistemlerinde bir DLQ işleyicisi IBM MQ ile birlikte verilir; daha fazla bilgi için bkz. [Örnek DLQ işleyicisi, amqsdlq](#).

**IBM i** IBM MQ for IBM i ile ilgili daha fazla bilgi için bkz. [IBM MQ for IBM i gönderilmeyen ileti kuyruğu işleyicisi](#).

## Kullanılan kuyruklar

Alıcı kanallarına ilişkin MCA ' lar, iletiler iletilmediğinde bile hedef kuyrukları açık tutabilir. Bu, kuyrukların "kullanımda" gibi görünmesine neden olur.

## Kanal sayısı üst sınırı

**IBM i** Açık IBM MQ for IBM i : Sisteminizde izin verilen kanal sayısı üst sınırını ve aynı anda etkin olabilecek kanal sayısı üst sınırını belirtebilirsiniz. Bu numaraları `qm.ini` dosyasında `QIBM/UserData/mqm/qmgrs/queue_manager_namedizininde` belirtirsiniz. Bkz. [Dağıtılmış kuyruğa alma için yapılandırma kütüğü kısmı](#).

## Sistem uzantıları ve kullanıcı çıkış programları

Kanal tanımında, iletilerin işlenmesi sırasında ek programların tanımlı zamanlarda çalıştırılmasını sağlamak için bir olanak sağlanır. Bu programlar IBM MQ ile birlikte sağlanmaz, ancak yerel gereksinimlere göre her kuruluş tarafından sağlanabilir.

Bu kullanıcı çıkış programlarının çalıştırılabilmesi için önceden tanımlanmış adların olması ve kanal programları çağrılınca kullanılabilir olması gerekir. İleti kanalı tanımlarında, kullanıcı çıkış programlarının adları yer alır.

Bu programlara denetim devir ve bu programlardan denetim iadesi için tanımlanmış bir denetim bloğu arabirimi vardır.

Bu programların çağrıldığı kesin yerler ve denetim bloklarının ve adlarının ayrıntıları, [İleti sistemi kanalları için kanal çıkış programlarındabulunur](#).

## Kanalları ve dinleyicileri güvenilir uygulamalar olarak çalıştırma

Ortamınızda performans önemliyse ve ortamınız istikrarlıysa, FASTPATH bağ tanımını kullanarak kanallarınızı ve dinleyicilerinizi güvenilir olarak çalıştırabilirsiniz. Kanalların ve dinleyicilerin güvenilir olarak çalışıp çalışmadığını etkileyen iki faktör vardır:

- `MQ_CONNECT_TYPE=FASTPATH` ya da `MQ_CONNECT_TYPE = STANDARD` ortam değişkeni. Bu büyük ve küçük harfe duyarlıdır. Geçerli olmayan bir değer belirtirseniz yoksayılr.
- `qm.ini` ya da kayıt dosyasının Kanal (Channel) kısmına ilişkin `MQIBindType` . Bunu FASTPATH ya da STANDARD olarak ayarlayabilirsiniz ve büyük ve küçük harfe duyarlı değildir. Varsayılan değer STANDARD değeridir.

Gerekli etkiyi aşağıdaki gibi elde etmek için ortam değişkeniyle ilişkili olarak `MQIBindType` ' ı kullanabilirsiniz:

MQIBindType	Ortam değişkeni	Sonuç
Standart	Tanımlı değil	Standart
FastPath	Tanımlı değil	FastPath
Standart	Standart	Standart
FastPath	Standart	Standart
Standart	FastPath	Standart
FastPath	FastPath	FastPath
Standart	CLIENT	CLIENT
FastPath	CLIENT	Standart
Standart	LOCAL	Standart
FastPath	LOCAL	Standart

Özet olarak, kanalları ve dinleyicileri güvenilir olarak çalıştırmanın yalnızca iki yolu vardır:

1. `qm.ini` ya da kayıttaki `MQIBindType= FASTPATH` belirtilerek ve ortam değişkenini belirtmeyerek.
2. `qm.ini` içinde `MQIBindType= FASTPATH` belirterek ya da ortam değişkenini `FASTPATH` olarak ayarlayarak.

Dinleyiciler durağan işlemler olduğundan, dinleyicileri güvenilir olarak çalıştırmayı düşünün. Dengesiz kanal çıkışları ya da `STOP CHANNEL MODE (TERMINATE)` komutunu kullanmıyorsan, kanalları güvenilir olarak çalıştırmayı düşünün.

## **ALW** AIX, Linux, and Windows üzerinde kanalların izlenmesi ve denetlenmesi

DQM için, uzak kuyruk yöneticilerine yönelik kanalları yaratmanız, izlemeniz ve denetlemeniz gerekir. Komutları, programları, IBM MQ Explorer, kanal tanımlarına ilişkin dosyaları ve eşitleme bilgileri için bir depolama alanını kullanarak kanalları denetleyebilirsiniz.

### **Bu görev hakkında**

Kanalları denetlemek için aşağıdaki komut tiplerini kullanabilirsiniz:

#### **IBM MQ komutları (MQSC)**

MQSC 'i AIX, Linux, and Windows sistemlerinde bir MQSC oturumunda tek komut olarak kullanabilirsiniz. Daha karmaşık ya da birden çok komut vermek için MQSC, komut satırından çalıştırdığınız bir dosyaya oluşturulabilir. Ayrıntılar için bkz. MQSC komutları. Bu kısımda, dağıtım kuyruğa alma için MQSC kullanılmasına ilişkin bazı yalın örnekler verilmiştir.

Kanal komutları, IBM MQ komutlarının (MQSC) bir altkümesidir. MQSC ve denetim komutları aşağıdaki işlemleri gerçekleştirmek için kullanılır:

- Kanal tanımlarının yaratılması, kopyalanması, görüntülenmesi, değiştirilmesi ve silinmesi
- Bağlantılar yeniden oluşturulamadığı zaman kanalları başlat ve durdur, ping, kanal sıra numaralarını sıfırla ve belirsiz iletileri çözümle
- Kanallara ilişkin durum bilgilerini görüntüle

#### **Denetim komutları**

Bu işlevlerden bazıları için komut satırında *denetim komutları* da yayınlayabilirsiniz. Ayrıntılar için bkz. [Denetim komutlarını kullanarak IBM MQ for Multiplatforms yönetme](#).

#### **Programlanabilir komut biçimi komutları**

Ayrıntılar için bkz. [PCF komutları](#).

#### **Linux Windows IBM MQ Explorer**

Linux ve Windows sistemlerinde IBM MQ Explorer kullanabilirsiniz. Denetim komutlarının ya da MQSC komutlarının kullanılmasına alternatif olarak denetim görevlerini gerçekleştirmek için bir grafik denetim arabirimi sağlar. Kanal tanımlamaları kuyruk yöneticisi nesnelere tutulur.

Her kuyruk yöneticisinin, uyumlu uzak kuyruk yöneticileriyle olan bağlantıları denetlemek için bir DQM bileşeni vardır. Depolama alanı, sıra numaralarını ve *mantıksal iş birimi (LUW)* tanıtıcılarını tutar. Bunlar kanal eşitleme amacıyla kullanılır.

Farklı komut tiplerini kullanarak ileti kanallarını ayarlarken ve denetlerken kullanabileceğiniz işlevlerin bir listesi için bkz. [Çizelge 21 sayfa 240](#).

### **Yordam**

- [“Kanalları kurmak ve denetlemek için gerekli işlevler” sayfa 240](#)
- [“Nesnelerle çalışmaya başlama” sayfa 242](#)
- [“Windows üzerinde iletişimi tuzak Ayarları ve ve” sayfa 249](#)
- [“AIX and Linux üzerinde iletişimi ayarlama” sayfa 256](#)

## İlgili görevler

“IBM i üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 262

Uzak kuyruk yöneticilerine kanallar yaratmak, izlemek ve denetlemek için DQM komutlarını ve panolarını kullanın. Her kuyruk yöneticisinin, uyumlu uzak kuyruk yöneticileriyle olan bağlantıları denetlemek için bir DQM programı vardır.

## İlgili başvurular

**ALW** AIX, Linux, and Windows üzerinde kanal programları

**ALW** AIX, Linux, and Windows için ileti kanalı planlama örneği

[Örnek yapılandırma bilgileri](#)

[Kanal öznitelikleri](#)

## **ALW** Kanalları kurmak ve denetlemek için gerekli işlevler

Kanalları kurmak ve denetlemek için bir dizi IBM MQ işlevi gerekebilir. Kanal işlevleri bu konuda açıklanmıştır.

Kanal tanımını, kanal adını, yaratmakta olduğunuz kanal tipini, kullanılacak iletişim yöntemini, iletim kuyruğu adını ve bağlantı adını belirterek IBM MQ tarafından sağlanan varsayılan değerleri kullanarak yaratabilirsiniz.

Kanal adı, kanalın her iki ucunda da aynı olmalı ve ağ içinde benzersiz olmalıdır. Ancak, kullanılan karakterleri IBM MQ nesne adları için geçerli olan karakterlerle sınırlamanız gerekir.

Kanalla ilgili diğer işlevler için aşağıdaki konulara bakın:



- [“Nesnelerle çalışmaya başlama” sayfa 242](#)
- [“İlişkili nesneler yaratılıyor” sayfa 243](#)
- [“Varsayılan nesneler yaratılıyor” sayfa 243](#)
- [“Kanal yaratılması” sayfa 243](#)
- [“Kanal görüntülenmesi” sayfa 244](#)
- [“Kanal durumunun görüntülenmesi” sayfa 244](#)
- [“Ping komutunu kullanarak bağlantıların denetlenmesi” sayfa 245](#)
- [“Kanal başlatılması” sayfa 245](#)
- [“Kanalı durdurma” sayfa 247](#)
- [“Kanalı yeniden adlandırma” sayfa 247](#)
- [“Kanalı sıfırlama” sayfa 248](#)
- [“Kanaldaki belirsiz iletilerin çözülmesi” sayfa 248](#)

Çizelge 21 sayfa 240 içinde, gereksinim duyabileceğiniz IBM MQ işlevlerinin tam listesi gösterilmektedir.

Çizelge 21. AIX, Linux, and Windows sistemlerinde gerekli işlevler			
İşlev	Denetim komutları	MQSC	IBM MQ Explorer eşdeğeri?
Kuyruk yöneticisi işlevleri			
Kuyruk yöneticisini değiştir		<a href="#">ALTER QMGR</a>	Evet
Kuyruk yöneticisi yarat	<a href="#">crtmqm</a>		Evet
Kuyruk yöneticisini sil	<a href="#">dlmqm</a>		Evet
Kuyruk yöneticisini görüntüle		<a href="#">QMGR ' YI GÖRÜNTÜLE</a>	Evet
Kuyruk yöneticisini sona erdir	<a href="#">endmqm</a>		Evet

Çizelge 21. AIX, Linux, and Windows sistemlerinde gerekli işlevler (devamı var)

İşlev	Denetim komutları	MQSC	IBM MQ Explorer eşdeğeri?
Ping kuyruk yöneticisi		<u>PING QMGR</u>	Hayır
Kuyruk yöneticisini başlat	<u>strmqm</u>		Evet
Komut sunucusu işlevleri			
Komut sunucusunu görüntüle	<u>dspmqcsv</u>		Hayır
Komut sunucusunu sona erdir	<u>endmqcsv</u>		Hayır
Komut sunucusunu başlat	<u>strmqcsv</u>		Hayır
Kuyruk işlevleri			
Kuyruğu değiştir		ALTER QALIAS ALTER QLOCAL ALTER QMODEL ALTER QREMOTE  Bkz. <u>ALTER kuyrukları</u> .	Evet
Kuyruğu temizle		<u>QLOCAL OLANAĞINI TEMİZLE</u>	Evet
Kuyruk yarat		QALIAS DEFINE QLOCAL DEFINE QMODEL DEFINE QREMOTE TANIMLAYIN  Bkz. <u>DEFINE kuyrukları</u> .	Evet
Kuyruğu sil		QALIAS DELETE QLOCAL DELETE QMODEL DELETE QREMOTE  Bkz. <u>DELETE kuyrukları</u> .	Evet
Kuyruğu görüntüle		<u>KUYRUĞU GÖRÜNTÜLE</u>	Evet
işlem işlevleri			
Süreci değiştir		<u>İŞLEMI DEĞİŞT</u>	Evet
Süreç yarat		<u>SÜRECI TANIMLAYIN</u>	Evet
Süreci Sil		<u>Süreci Sil</u>	Evet
Süreci görüntüle		<u>SÜRECI GÖRÜNTÜLE</u>	Evet
Kanal işlevleri			
Kanalı değiştir		<u>KANAL DEĞİŞTİN</u>	Evet
Kanal oluşturun		<u>KANALI TANIMLAYIN</u>	Evet
Kanalı sil		<u>KANALI SIL</u>	Evet

Çizelge 21. AIX, Linux, and Windows sistemlerinde gerekli işlevler (devamı var)			
İşlev	Denetim komutları	MQSC	IBM MQ Explorer eşdeğeri?
Kanalı görüntüle		KANALı GÖRÜNTÜLE	Evet
Kanal durumunu görüntüle		CHSTATUS ' U GÖRÜNTÜLE	Evet
Bitiş kanalı		KANALı DURDUR	Evet
Ping kanalı		PING KANALı	Evet
Kanalı sıfırla		KANALı SIFIRLAYIN	Evet
Kanalı çözümler		KANALı ÇÖZÜMLE	Evet
Kanalı çalıştır	runmqchl	KANALı BAŞLAYIN	Evet
Kanal başlatıcıyı çalıştır	runmqchi	CHINIT ' I BAŞLAYIN	Hayır
Dinleyiciyi çalıştır <sup>1</sup>	runmqslr	DINLEYICIYI BAŞLAYIN	Hayır
Dinleyiciyi sona erdir	endmqslr, yalnızca aşağıdaki altyapılarda: <ul style="list-style-type: none"> <li>•  AIX</li> <li>•  Windows</li> </ul> Windows sistemleri		Hayır

**Not:**

1. Kuyruk yöneticisi başlatıldığında bir dinleyici otomatik olarak başlatılabilir.

 **Nesnelerle çalışmaya başlama**

Bir kanalın başlatılabilmesi için kanallar tanımlanmalı ve ilişkili nesnelere var olmalı ve kullanılabilir olmalıdır. Bu bölüm size nasıl olduğunu gösterir.

IBM MQ komutlarını (MQSC) ya da IBM MQ Explorer kullanarak aşağıdakileri yapın:

1. İleti kanallarını ve ilişkili nesnelere tanımla
2. İleti kanallarını izleyin ve denetleyin

Tanımlamanız gereken ilişkili nesnelere şunlardır:

- İletim kuyrukları
- Uzak kuyruk tanımlamaları
- Kuyruk yöneticisi diğer ad tanımlamaları
- Yanıtın gönderileceği kuyruk diğer adı tanımlamaları
- Yanıtlanacak yerel kuyruklar
- Tetikleme işlemleri (MCA)
- İleti kanalı tanımlamaları

Bir kanal çalıştırılmadan önce, her bir kanal için belirli bir iletişim bağlantısı tanımlanmalı ve kullanılabilir olmalıdır. LU 6.2, TCP/IP, NetBIOS, SPX ve DECnet bağlantılarının nasıl tanımlandığına ilişkin açıklamalar için kuruluşunuza ilişkin iletişim kılavuzuna bakın. Ayrıca bkz. [Örnek yapılandırma bilgileri](#).

Nesne yaratma ve nesnelere çalışma hakkında daha fazla bilgi için aşağıdaki alt konulara bakın:

### **ALW İlişkili nesnelere yaratılıyor**

MQSC, ilişkili nesnelere yaratmak için kullanılır.

Kuyruk ve diğer ad nesnelere yaratmak için MQSC ' yi kullanın: iletim kuyrukları, uzak kuyruk tanımlamaları, kuyruk yöneticisi diğer adı tanımlamaları, yanıtlanacak kuyruk diğer adı tanımlamaları ve yanıtlanacak yerel kuyruklar.

Benzer bir şekilde tetikleme (MCA ' lar) için süreç tanımlarını da oluşturun.

Gerekli tüm nesnelere nasıl oluşturulacağını gösteren bir örnek için bkz. [AIX, Linux, and Windows için ileti kanalı planlama örneği](#).

### **ALW Varsayılan nesnelere yaratılıyor**

Bir kuyruk yöneticisi yaratıldığında varsayılan nesnelere otomatik olarak yaratılır. Bu nesnelere, kuyruklar, kanallar, süreç tanımlaması ve denetim kuyruklarıdır. Varsayılan nesnelere yaratıldıktan sonra, strmqm komutunu -c seçeneğiyle çalıştırarak bunları istediğiniz zaman değiştirebilirsiniz.

Bir kuyruk yöneticisi yaratmak için crtmqm komutunu kullandığınızda, komut bir dizi varsayılan nesne yaratmak için de bir program başlatır.

1. Her bir varsayılan nesne sırayla oluşturulur. Program, kaç nesnenin başarıyla tanımlandığını, kaç nesnenin var olduğunu ve kaç nesnenin değiştirildiğini ve kaç başarısız girişimde bulunduğunu belirtir.
2. Program sonuçları size görüntüler ve herhangi bir hata oluşursa, ayrıntılar için sizi uygun hata günlüğüne yönlendirir.

Program çalışmayı tamamlayınca, kuyruk yöneticisini başlatmak için strmqm komutunu kullanabilirsiniz.

crtmqm ve strmqm komutlarıyla ilgili ek bilgi için [Denetim komutlarını kullanarak IBM MQ for Multiplatforms yönetme başlıklı konuya](#) bakın.

## **Varsayılan nesnelere değiştirilmesi**

-c seçeneğini belirlediğinizde, nesnelere yaratılırken kuyruk yöneticisi geçici olarak başlatılır ve yeniden kapanır. -c seçeneğiyle strmqm komutu verilmesi, var olan sistem nesnelere varsayılan değerlerle yeniler (örneğin, bir kanal tanımlamasının MCAUSER özniteliği boş olarak ayarlanır). Kuyruk yöneticisini başlatmak istiyorsanız, -c seçeneği olmadan strmqm komutunu yeniden kullanmalısınız.

Varsayılan nesnelere değiştirmek istiyorsanız, eski amqscoma.tst dosyasının kendi sürümünü oluşturabilir ve düzenleyebilirsiniz.

### **ALW Kanal yaratılması**

Bağlantının her bir ucunda bir tane olacak şekilde iki kanal tanımlaması oluşturun. İlk kanal tanımlamasını ilk kuyruk yöneticisinde yaratırsınız. Daha sonra, ikinci kanal tanımlamasını bağlantının diğer ucunda ikinci kuyruk yöneticisinde yaratırsınız.

Her iki uç da aynı kanal adı kullanılarak tanımlanmalıdır. İki uç uyumlu kanal tiplerine sahip olmalıdır; örneğin: Gönderen ve Alıcı.

Bağlantının bir ucuna ilişkin bir kanal tanımlaması yaratmak için MQSC DEFINE CHANNEL komutunu kullanın. Kanalın adını, bağlantının bu ucuna ilişkin kanal tipini, bir bağlantı adını, bir tanımlamayı (gerekliyse), iletim kuyruğunun adını (gerekliyse) ve iletim protokolünü ekleyin. Ayrıca, daha önce topladığınız bilgileri kullanarak, gerekli kanal tipine ilişkin sistem varsayılan değerlerinden farklı olmasını istediğiniz diğer öznitelikleri de ekleyin.

[Kanal öznitelikleri](#) ' ne ilişkin kanal özniteliklerinin değerlerini belirlemenize yardımcı olur.

**Not:** Ağınızda tüm kanalları benzersiz bir şekilde adlandırmanız önerilir. Kaynak ve hedef kuyruk yöneticisi adlarının kanal adına eklenmesi, bunu yapmanın iyi bir yoludur.

## Kanal örneği oluştur

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) +  
DESCR('Sender channel to QM2') +  
CONNAME(QM2) TRPTYPE(TCP) XMITQ(QM2) CONVERT(YES)
```

Tüm MQSC örneklerinde, komut bir komut dosyasında görüldüğü gibi ve AIX, Linux, and Windows'te yazıldığı gibi gösterilir. İki yöntem aynı görünür; ancak, bir komutu etkileşimli olarak yayınlamak için önce bir MQSC oturumu başlatmanız gerekir. Varsayılan kuyruk yöneticisi için `runmqsc` yazın ya da `runmqsc qmname` yazın; burada `qmname`, gerekli kuyruk yöneticisinin adıdır. Daha sonra, örneklerde gösterildiği gibi, istediğiniz sayıda komut yazın.

Taşınabilirlik için, komutlarınızın satır uzunluğunu 72 karakterle sınırlayın. Birden çok satır üzerinde devam etmek için +birleştirme karakterini kullanın:

- **Windows** Windows üzerinde, komut satırındaki girdiyi sonlandırmak için Ctrl-z tuşlarını kullanın.
- **Linux** **AIX** AIX and Linux üzerinde Ctrl-d tuşlarını kullanın.
- Alternatif olarak, AIX, Linux, and Windows üzerinde **end** komutunu kullanın.

### **ALW** Kanal görüntülenmesi

Bir kanalın özniteliklerini görüntülemek için MQSC DISPLAY CHANNEL komutunu kullanın.

DISPLAY CHANNEL komutunun ALL parametresi, belirli bir öznitelik istenmezse ve belirtilen kanal adı sosyal değilse varsayılan olarak kabul edilir.

Öznitelikler, [Kanal öznitelikleri](#) içinde açıklanır.

## Kanal örneklerini görüntüle

```
DISPLAY CHANNEL(QM1.TO.QM2) TRPTYPE, CONVERT  
DISPLAY CHANNEL(QM1.TO.*) TRPTYPE, CONVERT  
DISPLAY CHANNEL(*) TRPTYPE, CONVERT  
DISPLAY CHANNEL(QM1.TO.QMR34) ALL
```

### **ALW** Kanal durumunun görüntülenmesi

Kanal adını ve kanalların yürürlükteki durumunu ya da kaydedilen bilgilerin durumunu belirtmek için MQSC DISPLAY CHSTATUS komutunu kullanın.

DISPLAY CHSTATUS tüm ileti kanallarına uygulanır. Sunucu bağlantısı kanalları dışındaki MQI kanalları için geçerli değildir.

Görüntülenen bilgiler şunlardır:

- Kanal adı
- İletişim bağlantısı adı
- Kanalın belirsiz durumu (uygun olduğunda)
- Son sıra numarası
- İletim kuyruğu adı (uygun olduğunda)
- Belirsiz tanıtıcı (uygun olduğunda)
- Kesinleştirilen son sıra numarası
- Mantıksal iş birimi tanıtıcısı
- Süreç Tanıtıcısı
- **Windows** İş parçacığı tanıtıcısı (yalnızca Windows)



## Kanal durumu örneklerini görüntüle

```
DISPLAY CHSTATUS(*) CURRENT
DISPLAY CHSTATUS(QM1.TO.*) SAVED
```

Kaydedilen durum, kanalda en az bir ileti kümesi iletilinceye kadar geçerli olmaz. Durum, bir kanal durdurulduğunda (STOP CHL komutu kullanılarak) ve kuyruk yöneticisi sona erdirildiğinde de kaydedilir.

### **ALW** Ping komutunu kullanarak bağlantıların denetlenmesi

Sabit bir veri iletilmesini uzak uçla değiştirmek için **PING CHANNEL MQSC** komutunu kullanın.

Ping, sistem gözetmenine bağlantının kullanılabilir ve çalışır durumda olduğuna dair biraz güven verir.

Ping komutu, iletim kuyruklarının ve hedef kuyrukların kullanılmasını içermez. Kanal tanımlarını, ilgili iletişim bağlantısını ve ağ ayarlarını kullanır. Yalnızca kanal şu anda etkin değilse kullanılabilir.

Yalnızca gönderen, sunucu ve küme gönderen kanallarından kullanılabilir. İlgili kanal, bağlantının uzak tarafında başlatılır ve başlatma parametresi kararlaştırmasını gerçekleştirir. Hatalar olağan şekilde bildirilir.

İleti değiş tokuşunun sonucu Ping complete ya da bir hata iletilisi olarak gösterilir.

## 6.2 LU ile ping komutu gönder

Ping çağrıldığında, varsayılan olarak hiçbir kullanıcı kimliği ya da parola alma sonuna akmaz. Kullanıcı kimliği ve parola gerekliyse, bunlar kanal tanımının başlangıç ucunda yaratılabilir. Kanal tanımına bir parola girilirse, parola kaydedilmeden önce IBM MQ tarafından şifrelenir. Daha sonra konuşma boyunca akmadan önce şifresi çözülür.

### İlgili görevler

[İletişimi Sınamak İçin Ping Komutu Kullanılması](#)

[Bağlantıyı doğrulamak için bir kanala ping komutunu çınlama](#)

### İlgili başvurular

[PING CHANNEL \(sınama kanalı yanıtı\)](#)

### **ALW** Kanal başlatılması

Gönderen, sunucu ve istekte bulunan kanallar için MQSC START CHANNEL komutunu kullanın. Uygulamaların ileti alışverişi yapılabilmesi için, gelen bağlantılar için bir dinleyici programı başlatmanız gerekir.

START CHANNEL, kuyruk yöneticisi tetikleyicisiyle bir kanal ayarlandığında gerekli değildir.

Başlatıldığında, gönderen MCA kanal tanımlarını okur ve iletim kuyruğunu açar. Alıcı ya da sunucu kanalının karşılık gelen MCA 'yı uzaktan başlatan bir kanal başlatma sırası yayınlandı. Bunlar başlatıldığında, gönderen ve sunucu, iletim kuyruğuna gelen iletileri bekler ve vardıklarında iletir.

İş parçacığı olarak tetikleme ya da çalıştırma kanallarını kullanırken, başlatma kuyruğunu izlemek için kanal başlatıcının kullanılabilir olduğundan emin olun. Kanal başlatıcı, kuyruk yöneticisinin bir parçası olarak varsayılan olarak başlatılır.

Ancak, TCP ve LU 6.2 diğer yetenekleri sağlar:

- ▶ **Linux** ▶ **AIX** AIX and Linux üzerinde TCP için, inetd bir kanal başlatmak üzere yapılandırılabilir. inetd ayrı bir işlem olarak başlatıldı.
- ▶ **Linux** ▶ **AIX** AIX and Linux içindeki LU 6.2 için, SNA ürününüzü LU 6.2 yanıtlayıcı işlemini başlatabileceğiniz şekilde yapılandırın.

- **Windows** Windowsiçindeki LU 6.2 için, SNA Server 'i kullanarak bir kanal başlatmak üzere TpStart (SNA Server ile birlikte sağlanan bir yardımcı program) kullanabilirsiniz. TpStart ayrı bir işlem olarak başlatılır.

Başlat seçeneğinin kullanılması, gerektiğinde kanalın yeniden eşzamanlanmasına neden olur.

Başlangıçların başarılı olması için:

- Yerel ve uzak kanal tanımları var olmalıdır. Bir alıcı ya da sunucu bağlantısı kanalı için uygun bir kanal tanımı yoksa, kanal otomatik olarak tanımlanır, varsayılan bir kanal otomatik olarak yaratılır. Bkz. [Kanal otomatik tanımlama çıkış programı](#).
- İletim kuyruğunun var olması ve onu kullanan başka bir kanal olmaması gerekir.
- Yerel ve uzak MCA ' lar var olmalıdır.
- İletişim bağlantısı kullanılabilir olmalıdır.
- Kuyruk yöneticileri çalışıyor, yerel ve uzak olmalıdır.
- İleti kanalı zaten çalışmıyor olmalıdır.

Bir kanalı başlatma isteğinin kabul edildiğini doğrulayan bir ileti ekrana gönderilir. Başlatma komutunun başarılı olduğunu doğrulamak için hata günlüğünü denetleyin ya da DISPLAY CHSTATUS komutunu kullanın. Hata günlükleri şunlardır:

#### **Windows Windows**

*MQ\_DATA\_PATH\mqgrs\qmname\errors\AMQERR01.LOG* ( qmnameadlı her kuyruk yöneticisi için)

*MQ\_DATA\_PATH\mqgrs\@SYSTEM\errors\AMQERR01.LOG* (genel hatalar için)

*MQ\_DATA\_PATH* , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

**Not:** Windowsişletim sisteminde, yine de Windows sistemleri uygulama olay günlüğünde bir ileti alırsınız.

#### **Linux AIX AIX and Linux**

*/var/mqm/qmgrs/qmname/errors/AMQERR01.LOG* ( qmnameadlı her kuyruk yöneticisi için)

*/var/mqm/qmgrs/@SYSTEM/errors/AMQERR01.LOG* (genel hatalar için)

AIX, Linux, and Windowsistemlerinde, IBM MQ dinleyici işlemini başlatmak için **runmqcls** komutunu kullanın. Varsayılan olarak, kanal bağlantısı için gelen istekler, dinleyici işleminin MCA ' ları amqrmppa işleminin iş parçacıkları olarak başlatmasına neden olur.

```
runmqcls -t tcp -m QM2
```

Giden bağlantılar için kanalı aşağıdaki üç yoldan biriyle başlatmanız gerekir:

1. MCATYPE parametresine bağlı olarak kanalı bir işlem ya da iş parçacığı olarak başlatmak için kanal adını belirterek MQSC START CHANNEL komutunu kullanın. (Kanallar iş parçacığı olarak başlatılırsa, bunlar bir kanal başlatıcısının iş parçacıklarıdır.)

```
START CHANNEL(QM1.TO.QM2)
```

2. Kanalı bir işlem olarak başlatmak için runmqchl denetim komutunu kullanın.

```
runmqchl -c QM1.TO.QM2 -m QM1
```

3. Kanalı tetiklemek için kanal başlatıcıyı kullanın.

Kanaldan etkinliği durdurmasını istemek için MQSC STOP CHANNEL komutunu kullanın. Kanal, işletmen kanalı yeniden başlatıncaya kadar yeni bir ileti kümesi başlatmaz.

Durdurulan kanalları yeniden başlatma hakkında bilgi için bkz. [“Durdurulan kanallar yeniden başlatılıyor” sayfa 228.](#)

Bu komut, MQCHT\_CLNTCONN dışında herhangi bir kanalda yayınlanabilir.

İstediğiniz durdurma tipini seçebilirsiniz:

### Stop quiesce (susturmayı durdur) örneği

```
STOP CHANNEL(QM1.TO.QM2) MODE(QUIESCE)
```

Bu komut, kanalın düzgün bir şekilde kapanmasını ister. Geçerli ileti kümesi tamamlanır ve eşitleme noktası yordamı kanalın diğer ucuyla gerçekleştirilir. Kanal boşsa, bu komut bir alıcı kanalı sonlandırmaz.

### Durdurma zorlama örneği

```
STOP CHANNEL(QM1.TO.QM2) MODE(FORCE)
```

Bu seçenek kanalı hemen durdurur, ancak kanalın iş parçacığı ya da işlemi sonlandırmaz. Kanal, geçerli ileti kümesini işlemeyi tamamlamaz ve bu nedenle kanalı şüpheli olarak bırakabilir. Genel olarak, susturma durdurma seçeneğini kullanmayı düşünün.

### Sonlandırma örneğini durdur

```
STOP CHANNEL(QM1.TO.QM2) MODE(TERMINATE)
```

Bu seçenek kanalı hemen durdurur ve kanalın iş parçacığı ya da işlemi sona erdirir.

### Stop (quiesce) durduruldu örneği

```
STOP CHANNEL(QM1.TO.QM2) STATUS(STOPPED)
```

Bu komut bir MODE belirlemez; bu nedenle, varsayılan değer olarak MODE (QUIESCE) kullanılır. Kanalın otomatik olarak yeniden başlatılamayacağı, ancak el ile başlatılması gerektiği için kanalın durdurulmasını ister.

### Stop (quiesce) inactive (Durdurma) örneği

```
STOP CHANNEL(QM1.TO.QM2) STATUS(INACTIVE)
```

Bu komut bir MODE belirlemez; bu nedenle, varsayılan değer olarak MODE (QUIESCE) kullanılır. Gerekliğinde otomatik olarak yeniden başlatılması için kanalın devre dışı bırakılmasını ister.

Bir ileti kanalını yeniden adlandırmak için MQSC 'yi kullanın.

Aşağıdaki adımları gerçekleştirmek için MQSC 'yi kullanın:

1. Kanalı durdurmak için STOP CHANNEL 'ı kullanın.
2. Yeni adla yinelenen bir kanal tanımı yaratmak için DEFINE CHANNEL seçeneğini kullanın.

3. Doğru oluşturulup oluşturulmadığını denetlemek için DISPLAY CHANNEL 'ı kullanın.

4. Özgün kanal tanımını silmek için DELETE CHANNEL 'ı kullanın.

Bir ileti kanalını yeniden adlandırmaya karar verirsiniz, bir kanalın her bir ucunda birer tane olmak üzere iki kanal tanımı olduğunu unutmayın. Kanalı aynı anda her iki uçta da yeniden adlandırdığınızdan emin olun.

### **ALW Kanalı sıfırlama**

İleti sıra numarasını değiştirmek için MQSC RESET CHANNEL komutunu kullanın.

RESET CHANNEL komutu herhangi bir ileti kanalı için kullanılabilir, ancak MQI kanalları (istemci bağlantısı ya da sunucu bağlantısı) için kullanılamaz. İlk ileti, kanal yeniden başlatıldığında yeni sırayı başlatır.

Komut bir gönderen ya da sunucu kanalında verilirse, kanal yeniden başlatıldığında değişikliğin diğer tarafını bilgilendirir.

#### **İlgili kavramlar**

[“Nesnelerle çalışmaya başlama” sayfa 242](#)

Bir kanalın başlatılabilmesi için kanallar tanımlanmalı ve ilişkili nesnelere var olmalı ve kullanılabilir olmalıdır. Bu bölüm size nasıl olduğunu gösterir.

[“Kanal denetimi işlevi” sayfa 217](#)

Kanal denetimi işlevi, kanalları tanımlamanız, izlemeniz ve denetlemeniz için olanaklar sağlar.

#### **İlgili görevler**

[“Dağıtılmış kuyruğa alma yapılandırılıyor” sayfa 189](#)

Bu bölümde, kuyruk tanımlaması, kanal tanımlaması, tetikleme ve eşitleme noktaları yordamları da içinde olmak üzere IBM MQ kuruluşları arasındaki iletişimle ilgili daha ayrıntılı bilgi sağlar.

#### **İlgili başvurular**

[KANALI İLK DURUMUNA GETİR](#)

### **ALW Kanaldaki belirsiz iletilerin çözülmesi**

İletiler bir gönderen ya da sunucu tarafından kuşkuyla tutulduğunda MQSC RESOLVE CHANNEL komutunu kullanın. Örneğin, bağlantının bir ucu sonlandırıldığından ve kurtarılması olasılığı olmadığından.

RESOLVE CHANNEL komutu şu iki parametreden birini kabul eder: BACKOUT ya da COMMIT. Geriletme, iletileri iletim kuyruğuna geri yüklerken, Kesinleştirme bunları atar.

Kanal programı, bir iş ortağıyla oturum kurmaya çalışmaz. Bunun yerine, belirsiz iletileri gösteren mantıksal iş birimi tanıtcısını (LUWID) belirler. Daha sonra, istendiği gibi, aşağıdaki sorunlardan biri ortaya konmaktadır:

- İletiler iletim kuyruğuna geri yüklemek için BACKOUT; ya da
- İletiler iletim kuyruğundan silmek için COMMIT.

Çözümün başarılı olması için:

- Kanal etkin değil olmalıdır
- Kanal şüphe içinde olmalı.
- Kanal tipi gönderen, sunucu ya da küme gönderen olmalıdır
- Yerel bir kanal tanımı var olmalıdır
- Yerel kuyruk yöneticisi çalışıyor olmalıdır

#### **İlgili kavramlar**

[“Nesnelerle çalışmaya başlama” sayfa 242](#)

Bir kanalın başlatılabilmesi için kanallar tanımlanmalı ve ilişkili nesnelere var olmalı ve kullanılabilir olmalıdır. Bu bölüm size nasıl olduğunu gösterir.

[“Kanal denetimi işlevi” sayfa 217](#)

Kanal denetimi işlevi, kanalları tanımlamanız, izlemeniz ve denetlemeniz için olanaklar sağlar.

## İlgili görevler

[“Dağıtılmış kuyruğa alma yapılandırılıyor” sayfa 189](#)

Bu bölümde, kuyruk tanımlaması, kanal tanımlaması, tetikleme ve eşitleme noktası yordamları da içinde olmak üzere IBM MQ kuruluşları arasındaki iletişimi ilgili daha ayrıntılı bilgi sağlanır.

## İlgili başvurular


[KANAL ÇÖZÜMLE](#)

## **Windows** Windows üzerinde iletişimi tuzak Ayarları ve ve

Dağıtılmış kuyruğa alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmaya çalışır. Bunun başarılı olması için bağlantının tanımlanması ve kullanılabilir olması gerekir. Bu bölümde, IBM MQ for Windows sistemleri için kullanılabilir iletişim formlarını kullanarak bunun nasıl yapılacağını açıklanmaktadır.

## Başlamadan önce

Örnek yapılandırma- IBM MQ for Windows konusuna başvurmanız yararlı olabilir.

 TCP/IP kullanan bir ileti kanalı, ağ verimini önemli ölçüde artıracak hızlı bir TCP/IP tüneli sağlayan bir IBM Aspera faspio Gateway' e işaret edilebilir. Herhangi bir yetkili platformda çalışan bir kuyruk yöneticisi, Aspera gateway aracılığıyla bağlanabilir. Ağ geçidinin kendisi Red Hat ya da Ubuntu Linux ya da Windows üzerinde devreye alınır. Bkz. [Linux ya da Windows üzerinde Aspera gateway bağlantısı tanımlanması](#).

## Bu görev hakkında

Windows üzerinde IBM MQ iletişimi ayarlarken aşağıdaki iletişim tipleri arasından seçim yapabilirsiniz:

- TCP/IP
- LU 6.2
- NetBIOS

## Yordam

- Windows sisteminize ilişkin iletişimi ayarlama hakkında bilgi için, seçtiğiniz iletişim tipine ilişkin alt konuya bakın:
  - [“Windows üzerinde TCP bağlantısı tanımlanması” sayfa 250](#)
  - [“Windows üzerinde LU 6.2 bağlantısı tanımlanması” sayfa 251](#)
  - [“Windows üzerinde NetBIOS bağlantısı tanımlanması” sayfa 253](#)

TCP/IP dışında iletişim protokolleri kullanan ortamlarda IBM MQ for Windows işlevlerinin ve olanaklarının tümü kullanılamaz. Kullanılabilir olmayan öge IBM MQ Explorer.

## İlgili görevler

[“AIX, Linux, and Windows üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 239](#)

DQM için, uzak kuyruk yöneticilerine yönelik kanalları yaratmanız, izlemeniz ve denetlemeniz gerekir. Komutları, programları, IBM MQ Explorer, kanal tanımlarına ilişkin dosyaları ve eşitleme bilgileri için bir depolama alanını kullanarak kanalları denetleyebilirsiniz.

[“İstemci ile sunucu arasındaki bağlantıların yapılandırılması” sayfa 14](#)

IBM MQ MQI clients ile sunucular arasındaki iletişim bağlantılarını yapılandırmak için iletişim protokolünüze karar verin, bağlantının her iki ucundaki bağlantıları tanımlayın, bir dinleyici başlatın ve kanalları tanımlayın.

[“AIX and Linux üzerinde iletişimi ayarlama” sayfa 256](#)

Bir dağıtılmış kuyruğa alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmayı dener. Bunun başarılı olması için bağlantının tanımlanması ve kullanılabilir olması gerekir. Bu bölümde,

IBM MQ for UNIX or Linux sistemleri için kullanılabilir iletişim formlarını kullanarak bunun nasıl yapılacağını açıklanmaktadır.

### İlgili başvurular


“Kullanılacak iletişim tipi” sayfa 15

Farklı platformlar farklı iletişim protokollerini destekler. İletim iletişim kuralı seçiminiz, IBM MQ MQI client ve sunucu platformlarının birleşimine bağlıdır.

### Windows **Windows üzerinde TCP bağlantısı tanımlanması**

Hedefin adresini belirlemek için gönderme ucunda bir kanal konfigürasyonu tanımlayarak ve alıcı ucunda bir dinleyici programı çalıştırarak bir TCP bağlantısı tanımlayın.

### Başlamadan önce

 TCP/IP kullanan bir ileti kanalı, ağ verimini önemli ölçüde artıracak hızlı bir TCP/IP tüneli sağlayan bir IBM Aspera faspio Gateway' e işaret edilebilir. Herhangi bir yetkili platformda çalışan bir kuyruk yöneticisi, Aspera gateway aracılığıyla bağlanabilir. Ağ geçidinin kendisi Red Hat ya da Ubuntu Linux ya da Windows üzerinde devreye alınır. Bkz. [Linux ya da Windows üzerinde Aspera gateway bağlantısı tanımlanması](#).

### Gönderme sonu

Kanal tanımının Bağlantı adı alanında, hedef makinenin anasistem adını ya da TCP adresini belirleyin.

Bağlanacak kapı varsayılan olarak 1414 değerine ayarlanır. Kapı numarası 1414, Internet Assigned Numbers Authority tarafından IBM MQ adresine atanır.

Varsayılan dışında bir kapı numarası kullanmak için, kanal nesnesi tanımlamasının bağlantı adı alanında bu değeri belirleyin:

```
DEFINE CHANNEL('channel name') CHLTYPE(SDR) +
  TRPTYPE(TCP) +
  CONNAME('OS2ROG3(1822)') +
  XMITQ('XMITQ name') +
  REPLACE
```

Burada OS2ROG3 uzak kuyruk yöneticisinin DNS adıdır ve 1822 gerekli kapıdır. (Bu, alıcı uçtaki dinleyicinin dinlediği kapı olmalıdır.)

Kanal nesne tanımında herhangi bir değişiklik yapılabilmesi için çalışan bir kanal durdurulmalı ve yeniden başlatılmalıdır.

Varsayılan kapı numarasını, IBM MQ for Windows için .ini dosyasında belirterek değiştirebilirsiniz:

```
TCP:
Port=1822
```

**Not:** Hangi TCP/IP kapı numarasının kullanılacağını seçmek için IBM MQ , aşağıdaki sırada bulunduğu ilk kapı numarasını kullanır:

1. Kanal tanımında ya da komut satırında açık olarak belirtilen kapı numarası. Bu numara, bir kanal için varsayılan kapı numarasının geçersiz kılınmasına izin verir.
2. .ini dosyasının TCP tarafında belirtilen kapı özniteliği. Bu sayı, bir kuyruk yöneticisi için varsayılan kapı numarasının geçersiz kılınmasına izin verir.
3. Varsayılan değer olan 1414. Bu, hem gelen hem de giden bağlantılar için Internet Assigned Numbers Authority tarafından IBM MQ ' e atanan numaradır.

qm.inikullanarak ayarladığınız değerlerle ilgili daha fazla bilgi için bkz. [Dağıtılmış kuyruğa alma için yapılandırma dosyası stanzası](#).

## TCP ' de alma

Bir alıcı kanal programını başlatmak için, gelen ağ isteklerini saptamak ve ilişkili kanalı başlatmak üzere bir dinleyici programı başlatılmalıdır. IBM MQ dinleyicisini kullanabilirsiniz.

Kanal programlarını alma işlemi, gönderen kanaldan gelen başlatma isteğine yanıt olarak başlatılır.

Bir alıcı kanal programını başlatmak için, gelen ağ isteklerini saptamak ve ilişkili kanalı başlatmak üzere bir dinleyici programı başlatılmalıdır. IBM MQ dinleyicisini kullanabilirsiniz.

IBM MQ ile sağlanan ve yeni kanalları iş parçacığı olarak başlatan Dinleyiciyi çalıştırmak için [runmqclsr](#) komutunu kullanın.

**runmqclsr** komutunun kullanılmasına ilişkin temel bir örnek:

```
runmqclsr -t tcp [-m QMNAME] [-p 1822]
```

Köşeli ayraç isteğe bağlı parametreleri gösterir; varsayılan kuyruk yöneticisi için QMNAME gerekli değildir ve varsayılan değeri (1414) kullanıyorsanız kapı numarası gerekli değildir. Kapı numarası 65535 'i aşmamalıdır.

**Not:** Hangi TCP/IP kapı numarasının kullanılacağını seçmek için IBM MQ , aşağıdaki sırada bulunduğu ilk kapı numarasını kullanır:

1. Kanal tanımında ya da komut satırında açık olarak belirtilen kapı numarası. Bu numara, bir kanal için varsayılan kapı numarasının geçersiz kılınmasına izin verir.
2. .ini dosyasının TCP tarafında belirtilen kapı özniteliği. Bu sayı, bir kuyruk yöneticisi için varsayılan kapı numarasının geçersiz kılınmasına izin verir.
3. Varsayılan değer olan 1414. Bu, hem gelen hem de giden bağlantılar için Internet Assigned Numbers Authority tarafından IBM MQ ' e atanan numaradır.

En iyi performans için, IBM MQ dinleyicisini “Kanalları ve dinleyicileri güvenilir uygulamalar olarak çalıştırma” sayfa 238’inde açıklandığı gibi güvenilir bir uygulama olarak çalıştırın. Güvenilir uygulamalarla ilgili bilgi için [Güvenilen uygulamalara ilişkin kısıtlamalar](#) başlıklı konuya bakın.

## TCP/IP SO\_KEEPALIVE seçeneğinin kullanılması

Windows SO\_KEEPALIVE seçeneğini kullanmak istiyorsanız, kaydınıza aşağıdaki girişi eklemelisiniz:

```
TCP:  
KeepAlive=yes
```

SO\_KEEPALIVE seçeneğiyle ilgili daha fazla bilgi için bkz. “Kanalın diğer ucunun hala kullanılabilir olup olmadığı denetleniyor” sayfa 224.

Windows işletim sistemlerinde, Windows **KeepAliveTime** seçeneği için HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters kayıt değeri, bağlantı denetlenmeden önce geçen aralığı denetler. Varsayılan değer iki saattir.

## TCP dinleyici birikim listesi seçeneğinin kullanılması

TCP ' de, sunucu ile istemci arasında üç yönlü tokalaşma gerçekleşmediği sürece bağlantılar eksik olarak işlenir. Bu bağlantılara, bekleyen bağlantı istekleri denir. Bu bekleyen bağlantı istekleri için bir değer üst sınırı belirlenir ve dinleyicinin isteği kabul etmesi için TCP kapısında bekleyen isteklerin birikim listesi olarak kabul edilebilir.

Daha fazla bilgi ve Windows için özel değer için bkz. “IBM MQ for Multiplatforms üzerinde TCP dinleyici birikim listesi seçeneğinin kullanılması” sayfa 259 .

## Windows üzerinde LU 6.2 bağlantısı tanımlanması

SNA, iki makine arasında bir LU 6.2 etkileşimi kurulabilecek şekilde yapılandırılmalıdır.

SNA yapılandırıldıktan sonra aşağıdaki gibi devam edin.

Bilgi için aşağıdaki çizelgeye bakın.

<i>Çizelge 22. Uzak kuyruk yöneticisi platformuna ilişkin yerel Windows sistemindeki ayarlar</i>		
<b>Uzak altyapı</b>	<b>TPNAME</b>	<b>TPPATH</b>
z/OS ya da MVS/ESA olmadan CICS	Uzak kuyruk yöneticisine ilişkin yan bilgilerle aynı.	-
z/OS ya da MVS/ESA kullanma CICS	CKRC (gönderen) CKSV (istekte bulunan) CKRC (sunucu)	-
IBM i	IBM i sistemindeki yöneltme girişindeki karşılaştırma değeriyle aynı.	-
AIX and Linux sistemleri	Uzak kuyruk yöneticisine ilişkin yan bilgilerle aynı.	<i>MQ_INSTALLATION_PATH/bin/amqcrs6a</i>
Windows	Windows Run Listener komutunda ya da Windows üzerinde TpSetup kullanılarak tanımlanan çağrılabilir Hareket Programı 'nda belirtildiği şekilde.	<i>MQ_INSTALLATION_PATH\bin\amqcrs6a</i>

*MQ\_INSTALLATION\_PATH* , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

Aynı makinede birden çok kuyruk yöneticiniz varsa, kanal tanımlamalarındaki TPadlarının benzersiz olmasına dikkat edin.

AnyNet TCP/IP Üzerinde SNA ' yı yapılandırmaya ilişkin en son bilgiler için şu çevrimiçi IBM belgelerine bakın: [AnyNet TCP/IP Üzerinde SNA](#) ve [SNA Düğümü İşlemleri](#).

### **İlgili kavramlar**

[“Windows üzerinde LU 6.2 üzerinde gönderme sonu” sayfa 252](#)

Kullanmakta olduğunuz LU 6.2 ürününün denetim uygulamasından bir CPI-C tarafı nesnesi (simgesel hedef) yaratın. Kanal tanımındaki Bağlantı adı alanına bu adı girin. Ortak için bir LU 6.2 bağlantısı da yaratın.

[“Windows üzerinde LU 6.2 ' de alma” sayfa 252](#)

Kanal programlarını alma işlemi, gönderen kanaldan gelen başlatma isteğine yanıt olarak başlatılır.

**Windows** *Windows üzerinde LU 6.2 üzerinde gönderme sonu*

Kullanmakta olduğunuz LU 6.2 ürününün denetim uygulamasından bir CPI-C tarafı nesnesi (simgesel hedef) yaratın. Kanal tanımındaki Bağlantı adı alanına bu adı girin. Ortak için bir LU 6.2 bağlantısı da yaratın.

CPI-C tarafı nesnesinde, alan makinedeki ortak LU adını, TP adını ve kip adını girin. Örneğin:

```
Partner LU Name      OS2R0G2
Partner TP Name      recv
Mode Name            #INTER
```

**Windows** *Windows üzerinde LU 6.2 ' de alma*

Kanal programlarını alma işlemi, gönderen kanaldan gelen başlatma isteğine yanıt olarak başlatılır.

Bir alıcı kanal programını başlatmak için, bir dinleyici programının gelen ağ isteklerini saptaması ve ilişkili kanalı başlatması gerekir. Bu dinleyici programını RUNMQLSR komutuyla başlatarak, TpName ' e dinleme olanağını verirsiniz. Diğer bir seçenek olarak, Windows için SNA Server altında TpStart komutunu kullanabilirsiniz.



## RUNMQLSR komutunun kullanılması

Dinleyiciyi başlatmak için komut örneği:

```
RUNMQLSR -t LU62 -n RECV -m QMNAME
```

Burada RECV , diğer (gönderen) uçta "TpName uzak taraftan başlayacak" olarak belirtilen TpName ' dir. Bu komutun son kısmında kullanılan **-m** değiştirgesi isteğe bağlıdır ve varsayılan kuyruk yöneticisi için gerekli değildir.

Bir makinede birden çok kuyruk yöneticisi çalışıyor olabilir. Her kuyruk yöneticisine farklı bir TpName atmanız ve her biri için bir dinleyici programı başlatmanız gerekir. Örneğin:

```
RUNMQLSR -t LU62 -m QM1 -n TpName1  
RUNMQLSR -t LU62 -m QM2 -n TpName2
```

En iyi performans için, IBM MQ dinleyicisini [Kanalların ve dinleyicilerin güvenilir uygulamalar olarak çalıştırılması](#) konusunda açıklandığı gibi güvenilir bir uygulama olarak çalıştırın. Güvenilir uygulamalarla ilgili bilgi için [Güvenilen uygulamalara ilişkin kısıtlamalar](#) başlıklı konuya bakın.

Aşağıdaki komutu kullanarak, etkin olmayan bir kuyruk yöneticisinde çalışan tüm IBM MQ dinleyicilerini durdurabilirsiniz:

```
ENDMQLSR -m QMNAME
```

## Windows üzerinde Microsoft SNA Server 'i kullanma

Daha sonra amqcrs6a.exe' yi yönlendiren bir çağrılabilir TP tanımlamak için TpSetup komutunu kullanabilirsiniz ya da çeşitli kayıt değerlerini el ile ayarlayabilirsiniz. amqcrs6a.exe ' ye geçirilecek parametreler şunlardır:

```
-m QM -n TpName
```

Burada QM , kuyruk yöneticisi adıdır, TpName ise TP adıdır. Ek bilgi için *Microsoft SNA Server APPC Programmers Guide* ya da *Microsoft SNA Server CPI-C Programmers Guide* adlı belgeye bakın.

Bir kuyruk yöneticisi adı belirtmezseniz, varsayılan kuyruk yöneticisi varsayılır.

### **Windows** **Windows üzerinde NetBIOS bağlantısı tanımlanması**

NetBIOS bağlantısı yalnızca Windowsçalıştıran bir istemci ve sunucu için geçerlidir. IBM MQ , başka bir IBM MQ ürününe NetBIOS bağlantısı kurarken üç tip NetBIOS kaynağı kullanır: oturumlar, komutlar ve adlar. Bu kaynakların her birinin bir sınırı vardır; bu sınır varsayılan olarak ya da NetBIOSkuruluşu sırasında belirlenmiştir.

Her çalışan kanal, tipinden bağımsız olarak, bir NetBIOS oturumu ve bir NetBIOS komutu kullanır. IBM NetBIOS uygulaması, birden çok sürecin aynı yerel NetBIOS adını kullanmasını sağlar. Bu nedenle, IBM MQtarafından kullanılmak üzere yalnızca bir NetBIOS adının kullanılabilir olması gerekir. Novell 'in NetBIOS öykünmesi gibi diğer satıcı firmaların uygulamaları, işlem başına farklı bir yerel ad gerektirir. Kullandığınız NetBIOS ürününe ilişkin belgelerden gereksinimlerinizi doğrulayın.

Her durumda, her tipte yeterli kaynağın kullanılabilir olduğundan emin olun ya da yapılandırmada belirtilen üst sınırı artırın. Değerlerde yapılan değişiklikler sistemin yeniden başlatılmasını gerektirir.

Sistem başlatılırken NetBIOS aygıt sürücüsü, uygulamalar tarafından kullanılacak oturum, komut ve ad sayısını görüntüler. Bu kaynaklar, aynı sistemde çalışan herhangi bir NetBIOStabanlı uygulama tarafından kullanılabilir. Bu nedenle, IBM MQ ' in bu kaynakları edinmesi gerekmeden önce diğer uygulamalar bu kaynakları kullanabilir. LAN ağ yöneticiniz bunu sizin için netleştirebilmelidir.

## İlgili kavramlar

“IBM MQ Yerel NetBIOS adının tanımlanması” sayfa 254

IBM MQ kanal işlemleri tarafından kullanılan yerel NetBIOS adı üç şekilde belirtilebilir.

“Kuyruk yöneticisi NetBIOS oturumu, komutu ve ad sınırlamasının oluşturulması” sayfa 254  
NetBIOS oturumları, komutları ve adları için kuyruk yöneticisi sınırları iki şekilde belirlenebilir.

“LAN bağdaştırıcısı numarasının oluşturulması” sayfa 255

Kanalların NetBIOS üzerinde başarılı bir şekilde çalışması için her uçta bağdaştırıcı desteği uyumlu olmalıdır. IBM MQ , qm.ini dosyanızın NETBIOS kısmına ilişkin AdapterNum değerini kullanarak ve runmqclsr komutunda **-a** parametresini belirterek LAN bağdaştırıcısı (LANA) sayısını denetlemenizi sağlar.

“NetBIOS bağlantısının başlatılması” sayfa 255

Bağlantı başlatmak için gereken adımları tanımlama.

“NetBIOS bağlantısı için hedef dinleyicinin tanımlanması” sayfa 255

NetBIOS bağlantısının alıcı ucunda gerçekleştirilecek adımları tanımlama.

### Windows IBM MQ Yerel NetBIOS adının tanımlanması

IBM MQ kanal işlemleri tarafından kullanılan yerel NetBIOS adı üç şekilde belirtilebilir.

Öncelik sırasına göre üç yol vardır:

1. **runmqclsr** komutunun **-l** parametresinde belirtilen değer; örneğin:

```
runmqclsr -t netbios -l my_station
```

2. Komut tarafından oluşturulan bir değere sahip **MQNAME** ortam değişkeni:

```
SET MQNAME= my_station
```

Örneğin:

```
SET MQNAME=CLIENT1
```

Her işlem için **MQNAME** değerini ayarlayabilirsiniz. Alternatif olarak, bunu Windows kaydında bir sistem düzeyinde ayarlayabilirsiniz.

Benzersiz adlar gerektiren bir NetBIOS uygulaması kullanıyorsanız, IBM MQ işleminin başlatıldığı her pencerede bir **SET MQNAME** komutu girmeniz gerekir. **MQNAME** değeri isteğe bağlı, ancak her işlem için benzersiz olmalıdır.

3. qm. inikuyruk yöneticisi yapılanış kütüğündeki **NETBIOS** kısmı . Örneğin:

```
NETBIOS:  
LocalName= my_station
```

### Not:

1. Desteklenen NetBIOS ürünlerinin uygulanmasındaki farklılıklar nedeniyle, her bir NetBIOS adını ağda benzersiz kılmanız önerilir. Bunu yapmazsanız, beklenmedik sonuçlar ortaya çıkabilir. Bir NetBIOS kanalı oluştururken sorun yaşıyorsanız ve kuyruk yöneticisi hata günlüğünde NetBIOS X'15 ' dönüş kodunu gösteren hata iletileri varsa, NetBIOS adlarını gözden geçirin.
2. Windows işletim sistemlerinde, makine adınızı Windows zaten kullandığı için NetBIOS adı olarak kullanamazsınız.
3. Gönderen kanalı başlatma, NetBIOS adının qm.ini dosyasında MQNAME ortam değişkeni ya da LocalName kullanılarak belirtilmesini gerektirir.

### Windows Kuyruk yöneticisi NetBIOS oturumu, komutu ve ad sınırlamasının oluşturulması

NetBIOS oturumları, komutları ve adları için kuyruk yöneticisi sınırları iki şekilde belirlenebilir.

Öncelik sırasına göre bu yollar şunlardır:

1. RUNMQLSR komutunda belirtilen değerler:

```
-s Sessions  
-e Names  
-o Commands
```

Komutta -m işleneni belirtilmezse, değerler yalnızca varsayılan kuyruk yöneticisi için geçerlidir.

2. qm.inikuyruk yöneticisi yapılandırma kütüğündeki NETBIOS kısmı. Örneğin:

```
NETBIOS:  
NumSess= Qmgr_max_sess  
NumCmds= Qmgr_max_cmds  
NumNames= Qmgr_max_names
```

**Windows** LAN bağdaştırıcısı numarasının oluşturulması

Kanalların NetBIOS üzerinde başarılı bir şekilde çalışması için her uçta bağdaştırıcı desteği uyumlu olmalıdır. IBM MQ, qm.ini dosyanızın NETBIOS kısmına ilişkin AdapterNum değerini kullanarak ve runmqlsr komutunda **-a** parametresini belirterek LAN bağdaştırıcısı (LANA) sayısını denetlemenizi sağlar.

IBM MQ tarafından NetBIOS bağlantıları için kullanılan varsayılan LAN bağdaştırıcısı numarası 0'dır. Sisteminizde kullanılan numarayı aşağıdaki gibi doğrulayın:

Windows işletim sisteminde, LAN bağdaştırıcısı numarasını doğrudan işletim sistemi üzerinden sorgulamak mümkün değildir. Bunun yerine, LANACFG.EXE komut satırı yardımcı programı, Microsoft adresinden edinilebilir. Aracın çıkışı, sanal LAN bağdaştırıcısı numaralarını ve bunların etkin bağ tanımlarını gösterir. LAN bağdaştırıcısı numaralarına ilişkin ek bilgi için Microsoft Knowledge Base makale 138037 *HOWTO: Use LANA Numbers in a 32-bit Environment* başlıklı konuya bakın.

qm.ini: kuyruk yöneticisi yapılandırma kütüğünün NETBIOS kısmına doğru değeri girin.

```
NETBIOS:  
AdapterNum= n
```

burada n, bu sistem için doğru LAN bağdaştırıcısı numarasıdır.

**Windows** NetBIOS bağlantısının başlatılması

Bağlantı başlatmak için gereken adımları tanımlama.

Bağlantıyı başlatmak için, gönderme ucunda aşağıdaki adımları izleyin:

1. MQNAME ya da LocalName değerini kullanarak NetBIOS istasyonu adını tanımlayın.
2. Sisteminizde kullanılmakta olan LAN bağdaştırıcısı numarasını doğrulayın ve AdapterNum komutunu kullanarak doğru dosyayı belirleyin.
3. Kanal tanımının ConnectionName alanında, hedef dinleyici programı tarafından kullanılan NetBIOS adını belirtin. Windows işletim sistemlerinde NetBIOS kanalları iş parçacığı olarak çalıştırılmalıdır. Kanal tanımında MCATYPE (THREAD) belirterek bunu yapın.

```
DEFINE CHANNEL (chname) CHLTYPE(SDR) +  
TRPTYPE(NETBIOS) +  
CONNNAME(your_station) +  
XMITQ(xmitq) +  
MCATYPE(THREAD) +  
REPLACE
```

**Windows** NetBIOS bağlantısı için hedef dinleyicinin tanımlanması

NetBIOS bağlantısının alıcı ucunda gerçekleştirilecek adımları tanımlama.

Alma ucunda, aşağıdaki adımları izleyin:

1. MQNAME ya da LocalName değerini kullanarak NetBIOS istasyonu adını tanımlayın.
2. Sisteminizde kullanılmakta olan LAN bağdaştırıcısı numarasını doğrulayın ve AdapterNumkomutunu kullanarak doğru dosyayı belirleyin.
3. Alıcı kanalını tanımlayın:

```
DEFINE CHANNEL (chname) CHLTYPE(RCVR) +  
TRPTYPE(NETBIOS) +  
REPLACE
```

4. İstasyon oluşturmak ve iletişim kurmak için IBM MQ dinleyici programını başlatın. Örneğin:

```
RUNMQLSR -t NETBIOS -l your_station [-m qmgr]
```

Bu komut, `your_station` 'yi iletişim kurulmayı bekleyen bir NetBIOS istasyonu olarak oluşturur. NetBIOS istasyonu adı, NetBIOS ağınızda benzersiz olmalıdır.

En iyi performans için, IBM MQ dinleyicisini “[Kanalları ve dinleyicileri güvenilir uygulamalar olarak çalıştırma](#)” sayfa 238’inde açıklandığı gibi güvenilir bir uygulama olarak çalıştırın. Güvenilir uygulamalarla ilgili bilgi için [Güvenilen uygulamalara ilişkin kısıtlamalar](#) başlıklı konuya bakın.

Aşağıdaki komutu kullanarak, etkin olmayan bir kuyruk yöneticisinde çalışan tüm IBM MQ dinleyicilerini durdurabilirsiniz:

```
ENDMQLSR [-m QMNAME]
```



Bir kuyruk yöneticisi adı belirtmezseniz, varsayılan kuyruk yöneticisi varsayılır.

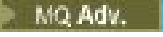

## **AIX and Linux üzerinde iletişimi ayarlama**

Bir dağıtım kuyruğa alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmayı dener. Bunun başarılı olması için bağlantının tanımlanması ve kullanılabilir olması gerekir. Bu bölümde, IBM MQ for UNIX or Linux sistemleri için kullanılabilir iletişim formlarını kullanarak bunun nasıl yapılacağını açıklanmaktadır.

### **Başlamadan önce**

Aşağıdaki bölümlere başvurmanız yararlı olabilir:

-  [Örnek yapılandırma- IBM MQ for AIX](#)
-  [Örnek yapılandırma- Linux için IBM MQ](#)

  TCP/IP kullanan bir ileti kanalı, ağ verimini önemli ölçüde artıracak hızlı bir TCP/IP tüneli sağlayan bir IBM Aspera faspio Gateway' e işaret edilebilir. Herhangi bir yetkili platformda çalışan bir kuyruk yöneticisi, Aspera gateway aracılığıyla bağlanabilir. Ağ geçidinin kendisi Red Hat ya da Ubuntu Linux ya da Windows üzerinde devreye alınır. Bkz. [Linux ya da Windows üzerinde Aspera gateway bağlantısı tanımlanması](#).

### **Bu görev hakkında**

Bir dağıtım kuyruğa alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmayı dener. Başarılı olmak için, bağlantının tanımlanması ve kullanılabilir olması gerekir. Bu bölümde bunun nasıl yapılacağını açıklanır.

AIX and Linux üzerinde IBM MQ için iletişim ayarlarken, aşağıdaki iletişim tipleri arasından seçim yapabilirsiniz:

- TCP/IP
- LU 6.2

Her kanal tanımı, iletim protokolü (İletim Tipi) özniteliği olarak bir tane belirlemelidir. Bir ya da daha çok protokol bir kuyruk yöneticisi tarafından kullanılabilir.

IBM MQ MQI clients için, farklı iletim protokolleri kullanan alternatif kanalların olması yararlı olabilir. Bkz. [IBM MQ MQI clients](#).

## Yordam

AIX ya da Linux sisteminize ilişkin iletişimi ayarlamaya ilişkin bilgi için, seçtiğiniz iletişim tipine ilişkin alt konuya bakın:

- [“AIX and Linux üzerinde TCP bağlantısı tanımlanması” sayfa 257](#)
- [“AIX and Linux üzerinde LU 6.2 bağlantısı tanımlanması” sayfa 261](#)
- [MQ Adv. > MQ Adv. VUE](#) [“Linux ya da Windows platformlarında Aspera gateway bağlantısı tanımlanması” sayfa 827](#)

## İlgili görevler

[“AIX, Linux, and Windows üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 239](#)

DQM için, uzak kuyruk yöneticilerine yönelik kanalları yaratmanız, izlemeniz ve denetlemeniz gerekir. Komutları, programları, IBM MQ Explorer, kanal tanımlarına ilişkin dosyaları ve eşitleme bilgileri için bir depolama alanını kullanarak kanalları denetleyebilirsiniz.

[“İstemci ile sunucu arasındaki bağlantıların yapılandırılması” sayfa 14](#)

IBM MQ MQI clients ile sunucular arasındaki iletişim bağlantılarını yapılandırmak için iletişim protokolünüze karar verin, bağlantının her iki ucundaki bağlantıları tanımlayın, bir dinleyici başlatın ve kanalları tanımlayın.

[“Windows üzerinde iletişimi tuzak Ayarları ve ve” sayfa 249](#)

Dağıtılmış kuyruğa alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmaya çalışır. Bunun başarılı olması için bağlantının tanımlanması ve kullanılabilir olması gerekir. Bu bölümde, IBM MQ for Windows sistemleri için kullanılabilir iletişim formlarını kullanarak bunun nasıl yapılacağını açıklanmaktadır.

## İlgili başvurular

[“Kullanılacak iletişim tipi” sayfa 15](#)

Farklı platformlar farklı iletişim protokollerini destekler. İletim iletişim kuralı seçiminiz, IBM MQ MQI client ve sunucu platformlarının birleşimine bağlıdır.

## [Linux > AIX](#) **AIX and Linux üzerinde TCP bağlantısı tanımlanması**

Gönderme ucundaki kanal tanımı, hedefin adresini belirler. Dinleyici ya da inet cini, alma sonunda bağlantı için yapılandırıldı.

## Başlamadan önce

[MQ Adv. > CD](#) TCP/IP kullanan bir ileti kanalı, ağ verimini önemli ölçüde artıracak hızlı bir TCP/IP tüneli sağlayan bir IBM Aspera faspio Gateway' e işaret edilebilir. Herhangi bir yetkili platformda çalışan bir kuyruk yöneticisi, Aspera gateway aracılığıyla bağlanabilir. Ağ geçidinin kendisi Red Hat ya da Ubuntu Linux ya da Windows üzerinde devreye alınır. Bkz. [Linux ya da Windows üzerinde Aspera gateway bağlantısı tanımlanması](#).

## Gönderme sonu

Kanal tanımının Bağlantı Adı alanında, hedef makinenin anasistem adını ya da TCP adresini belirleyin. Bağlanacak kapı varsayılan olarak 1414 değerine ayarlanır. Kapı numarası 1414, Internet Assigned Numbers Authority tarafından IBM MQ adresine atanır.

Varsayılan dışında bir kapı numarası kullanmak için, bağlantı adı alanını aşağıdaki gibi değiştirin:

```
Connection Name REMHOST(1822)
```

Burada REMHOST uzak makinenin anasistem adı ve 1822 gerekli kapı numarasıdır. (Bu, alıcı uçtaki dinleyicinin dinlediği kapı olmalıdır.)

Diğer bir seçenek olarak, kapı numarasını kuyruk yöneticisi yapılandırma dosyasında (qm.ini) belirterek değiştirebilirsiniz:

```
TCP:
Port=1822
```

qm.inikullanarak ayarladığınız değerlerle ilgili daha fazla bilgi için bkz. [Dağıtılmış kuyruğa alma için yapılandırma dosyası stanzas.](#)

## TCP ' de alma

TCP/IP dinleyicisini (inetd) ya da IBM MQ dinleyicisini kullanabilirsiniz.

Bazı Linux dağıtımları artık inet cini yerine genişletilmiş inet yardımcı programını (xinetd) kullanıyor. Genişletilmiş inet cininin Linux sisteminde nasıl kullanılacağına ilişkin ek bilgi için [Example: setting up IBM MQ cross-platform communication on Linux](#) başlıklı konuya ilişkin [Step 2](#) başlıklı konuya bakın.

### İlgili kavramlar

[“AIX and Linux üzerinde TCP/IP dinleyicisinin kullanılması” sayfa 258](#)

Kanalları AIX and Linux üzerinde başlatmak için /etc/services dosyası ve inetd.conf dosyası düzenlenmelidir

[“IBM MQ for Multiplatforms üzerinde TCP dinleyici birikim listesi seçeneğinin kullanılması” sayfa 259](#)  
TCP ' de, sunucu ile istemci arasında üç yönlü tokalaşma gerçekleşmediği sürece bağlantılar eksik olarak işlenir. Bu bağlantılara, bekleyen bağlantı istekleri denir. Bu bekleyen bağlantı istekleri için bir değer üst sınırı belirlenir ve dinleyicinin isteği kabul etmesi için TCP kapısında bekleyen isteklerin birikim listesi olarak kabul edilebilir.

[“IBM MQ dinleyicisinin kullanılması” sayfa 260](#)

Yeni kanalları iş parçacığı olarak başlatan IBM MQ ile sağlanan dinleyiciyi çalıştırmak için runmq1sr komutunu kullanın.

[“TCP/IP SO\\_KEEPALIVE seçeneğinin kullanılması” sayfa 260](#)

Bazı AIX and Linux sistemlerinde, TCP ' nin bağlantının kullanılabilir olup olmadığını denetlemeden önce ne kadar bekleyeceğini ve ilk denetim başarısız olursa bağlantıyı yeniden ne sıklıkta deneyeceğini tanımlayabilirsiniz. Bu, bir çekirdek ayarlanabilir parametresidir ya da komut satırına girilebilir.

**Linux** **AIX** [AIX and Linux üzerinde TCP/IP dinleyicisinin kullanılması](#)

Kanalları AIX and Linux üzerinde başlatmak için /etc/services dosyası ve inetd.conf dosyası düzenlenmelidir

Aşağıdaki yönergeleri izleyin:

1. /etc/services dosyasını düzenleyin:

**Not:** /etc/services dosyasını düzenlemek için ayrıcalıklı kullanıcı ya da kök olarak oturum açmanız gerekir. Bunu değiştirebilirsiniz, ancak bu, gönderme sonunda belirtilen kapı numarasıyla eşleşmelidir.

Şu satırı dosyaya ekleyin:

```
MQSeries 1414/tcp
```

Burada 1414, IBM MQ için gerekli kapı numarasıdır. Kapı numarası 65535 'i aşmamalıdır.

2. amqcrsta programını çağırmak için inetd.conf dosyasına bir satır ekleyin; burada `MQ_INSTALLATION_PATH`, IBM MQ ' in kurulu olduğu üst düzey dizini temsil eder:

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
[-m Queue_Man_Name]
```

Güncellemeler, inetd yapılandırma dosyalarını yeniden okuduktan sonra etkindir. Bunu yapmak için kök kullanıcı kimliğinden aşağıdaki komutları verin:

- **AIX** AIX'ta:

```
refresh -s inetd
```

- **Linux** Linux sistemlerinde:

```
kill -1 process_number
```

Inetd tarafından başlatılan dinleyici programı ülke değerini inetd 'den edindiğinde, MQMDE yerine getirilmemiş (birleştirilmiş) ve kuyruğa ileti verileri olarak yerleştirilmiş olabilir. MQMDE ' nin yerine getirildiğinden emin olmak için ülke değerini doğru şekilde ayarlamanız gerekir. inetd tarafından ayarlanan yerel ayar, IBM MQ işlemleri tarafından kullanılan diğer yerel ayarlar için seçilen yerel ayarlarla eşleşmeyebilir. Ülke değerini ayarlamak için:

1. LANG, LC\_COLLATE, LC\_CTYPE, LC\_MONETARY, LC\_NUMERIC, LC\_TIME ve LC\_MESSAGES ülke değeri ortam değişkenlerini diğer IBM MQ işlemi için kullanılan ülke değerine ayarlayan bir kabuk komut dosyası yaratın.
2. Aynı kabuk komut dosyasında dinleyici programını çağırın.
3. inetd.conf dosyasını değiştirerek, dinleyici programı yerine kabuk komut dosyanızı çağırın.

Sunucuda birden çok kuyruk yöneticisi olabilir. Kuyruk yöneticilerinin her biri için iki dosyanın her birine bir satır eklemelisiniz. Örneğin:

```
MQSeries1 1414/tcp
MQSeries2 1822/tcp
```

```
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

Burada `MQ_INSTALLATION_PATH` , IBM MQ ' in kurulu olduğu üst düzey dizini temsil eder.

Bu, tek bir TCP kapısında kuyruğa alınan bekleyen bağlantı isteklerinin sayısında bir sınırlama varsa, hata iletilerinin oluşturulmasını önler. Bekleyen bağlantı isteklerinin sayısı hakkında bilgi için bkz. [“IBM MQ for Multiplatforms üzerinde TCP dinleyici birikim listesi seçeneğinin kullanılması” sayfa 259.](#)

► **Multi** *IBM MQ for Multiplatforms üzerinde TCP dinleyici birikim listesi seçeneğinin kullanılması*  
TCP ' de, sunucu ile istemci arasında üç yönlü tokalaşma gerçekleşmediği sürece bağlantılar eksik olarak işlenir. Bu bağlantılara, bekleyen bağlantı istekleri denir. Bu bekleyen bağlantı istekleri için bir değer üst sınırı belirlenir ve dinleyicinin isteği kabul etmesi için TCP kapısında bekleyen isteklerin birikim listesi olarak kabul edilebilir.

Varsayılan dinleyici birikim listesi değerleri [Çizelge 23 sayfa 259](#) içinde gösterilir.

<i>Çizelge 23. TCP/IP kapısında kuyruğa alınan bağlantı isteği sayısı üst sınırı</i>	
<b>Sunucu altyapısı</b>	<b>Bağlantı isteği sayısı üst sınırı</b>
► <b>AIX</b> AIX	100
► <b>Linux</b> Linux	100
► <b>IBM i</b> IBM i	255
► <b>Windows</b> Windows Sunucusu	100

Birikim listesi [Çizelge 23 sayfa 259](#) içinde gösterilen değerlere ulaşırsa, TCP/IP bağlantısı reddedilir ve kanal başlatılamaz.

MCA kanalları için bu, kanalın RETRY durumuna girmesine ve daha sonra bağlantıyı yeniden denemesine neden olur.

Ancak, bu hatayı önlemek için `qm.ini` dosyasına bir girdi ekleyebilirsiniz:

```
TCP:
ListenerBacklog = n
```

Bu, varsayılan ödenmemiş istek sayısı üst sınırını geçersiz kılar (bkz. [Çizelge 23 sayfa 259](#)) (TCP/IP dinleyicisi için).

**Not:** Bazı işletim sistemleri varsayılan değerden daha büyük bir değeri destekler. Gerekirse, bağlantı sınırına ulaşılmasını önlemek için bu değer kullanılabilir.

Dinleyiciyi birikim listesi seçeneği etkinleştirilmiş olarak çalıştırmak için aşağıdakilerden birini yapın:

- `runmqclsr -b` komutunu kullanın ya da
- BACKLOG özneteliği gerekli değere ayarlanmış olarak **DEFINE LISTENER MQSC** komutunu kullanın.

**runmqclsr** komutuyla ilgili bilgi için bkz. [runmqclsr](#). **DEFINE LISTENER** komutuyla ilgili bilgi için [DEFINE LISTENER](#) başlıklı konuya bakın.

### İlgili kavramlar

[“z/OS üzerinde TCP dinleyici birikim listesi seçeneğinin kullanılması” sayfa 977](#)

TCP/IP 'de alma sırasında, işlem bekleyen bağlantı isteği sayısı üst sınırı belirlenir. Bu bekleyen istekler, dinleyicinin isteği kabul etmesi için TCP/IP kapısında bekleyen isteklerin *birikim listesi* olarak kabul edilebilir.

 **Linux** → **AIX** *IBM MQ dinleyicisinin kullanılması*

Yeni kanalları iş parçacığı olarak başlatan IBM MQ ile sağlanan dinleyiciyi çalıştırmak için `runmqclsr` komutunu kullanın.

Örneğin:

```
runmqclsr -t tcp [-m QMNAME] [-p 1822]
```


Köşeli ayraç isteğe bağlı parametreleri gösterir; varsayılan kuyruk yöneticisi için QMNAME gerekli değildir ve varsayılan değeri (1414) kullanıyorsanız kapı numarası gerekli değildir. Kapı numarası 65535 'i aşmamalıdır.

En iyi performans için, IBM MQ dinleyicisini [“Kanalları ve dinleyicileri güvenilir uygulamalar olarak çalıştırma” sayfa 238](#) içinde açıklandığı gibi güvenilir bir uygulama olarak çalıştırın. Güvenilir uygulamalarla ilgili bilgi için [Güvenilen uygulamalara ilişkin kısıtlamalar](#) başlıklı konuya bakın.

Aşağıdaki komutu kullanarak, etkin olmayan bir kuyruk yöneticisinde çalışan tüm IBM MQ dinleyicilerini durdurabilirsiniz:

```
endmqclsr [-m QMNAME]
```

Bir kuyruk yöneticisi adı belirtmezseniz, varsayılan kuyruk yöneticisi varsayılır.

 **Linux** → **AIX** *TCP/IP SO\_KEEPALIVE seçeneğinin kullanılması*

Bazı AIX and Linux sistemlerinde, TCP 'nin bağlantının kullanılabilir olup olmadığını denetlemeden önce ne kadar bekleyeceğini ve ilk denetim başarısız olursa bağlantıyı yeniden ne sıklıkta deneyeceğini tanımlayabilirsiniz. Bu, bir çekirdek ayarlanabilir parametresidir ya da komut satırına girilebilir.



SO\_KEEPALIVE seçeneğini kullanmak istiyorsanız (ek bilgi için bkz. “Kanalın diğer ucunun hala kullanılabilir olup olmadığı denetleniyor” sayfa 224 ) Kuyruk yöneticisi yapılandırma kütüğüne (qm.ini) aşağıdaki girişi eklemeniz gerekir:

```
TCP:
KeepAlive=yes
```

Ek bilgi için AIX ya da Linux sisteminize ilişkin belgelere bakın.

### Linux → AIX **AIX and Linux üzerinde LU 6.2 bağlantısı tanımlanması**

SNA, iki makine arasında bir LU 6.2 etkileşimi kurulabilecek şekilde yapılandırılmalıdır.

TCP/IP üzerinde SNA ' yı yapılandırmaya ilişkin en son bilgiler için şu çevrimiçi IBM belgelerine bakın: [Communications Server](#).

SNA, iki sistem arasında bir LU 6.2 etkileşimi kurulabilecek şekilde yapılandırılmalıdır.

Bilgi için *Multiplatform APPC Configuration Guide* belgesine ve aşağıdaki çizelgeye bakın.

Uzak altyapı	TPNAME	TPPATH
z/OS olmadan CICS	Uzak kuyruk yöneticisine ilişkin yan bilgilerdeki karşılık gelen TPName ile aynı.	-
z/OS kullanma CICS	CKRC (gönderen) CKSV (istekte bulunan) CKRC (sunucu)	-
IBM i	IBM i sistemindeki yöneltme girişindeki karşılaştırma değeriyle aynı.	-
AIX and Linux sistemleri	Uzak kuyruk yöneticisine ilişkin yan bilgilerdeki karşılık gelen TPName ile aynı.	<i>MQ_INSTALLATION_PATH</i> /bin/amqcrs6a
Windows	Windows Run Listener komutunda ya da Windows üzerinde TpSetup kullanılarak tanımlanan çağrılabilir Hareket Programı 'nda belirtildiği şekilde.	<i>MQ_INSTALLATION_PATH</i> \bin\amqcrs6a

*MQ\_INSTALLATION\_PATH* , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

Aynı makinede birden çok kuyruk yöneticiniz varsa, kanal tanımlamalarındaki TPAdlarının benzersiz olmasına dikkat edin.

#### İlgili kavramlar

“AIX and Linux üzerinde LU 6.2 üzerinde gönderme sonu” sayfa 261

AIX and Linux sistemlerinde bir CPI-C tarafı nesnesi (simgesel hedef) yaratın ve kanal tanımındaki Bağlantı adı alanına bu adı girin. Ortak için bir LU 6.2 bağlantısı da yaratın.

“AIX and Linux üzerinde LU 6.2 ' de alma” sayfa 262

AIX and Linux sistemlerinde, alıcı ucunda bir dinleme bağlantısı, bir LU 6.2 mantıksal bağlantı tanıtımı ve bir TPN tanıtımı yaratın.

### Linux → AIX **AIX and Linux üzerinde LU 6.2 üzerinde gönderme sonu**

AIX and Linux sistemlerinde bir CPI-C tarafı nesnesi (simgesel hedef) yaratın ve kanal tanımındaki Bağlantı adı alanına bu adı girin. Ortak için bir LU 6.2 bağlantısı da yaratın.

CPI-C yan nesnesinde, alan makinedeki ortak LU adını, hareket programı adını ve kip adını girin. Örneğin:

```
Partner LU Name          REMHOST
```

```
Remote TP Name      recv
Service Transaction Program no
Mode Name           #INTER
```

IBM MQ bir SNA oturumu oluşturmayı denediğinde, CPI-C tarafından desteklendiği yerde, SECURITY PROGRAM kullanılır.

**Linux** **AIX** *AIX and Linux üzerinde LU 6.2 ' de alma*

AIX and Linux sistemlerinde, alıcı ucunda bir dinleme bağlantısı, bir LU 6.2 mantıksal bağlantı tanıtımı ve bir TPN tanıtımı yaratın.

TPN tanıtımında, yürütülebilir dosyanın tam yolunu ve İşlem Programı adını girin:

```
Full path to TPN executable  MQ_INSTALLATION_PATH/bin/amqcrs6a
Transaction Program name     recv
User ID                       0
```

`MQ_INSTALLATION_PATH` , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

Kullanıcı kimliğini ayarlayabileceğiniz sistemlerde, mqm grubunun üyesi olan bir kullanıcı belirtin.

**AIX** AIX sistemlerinde APPCTPN (hareket adı) ve APPCLLU (yerel LU adı) ortam değişkenlerini ayarlayın (çağrılan hareket programına ilişkin yapılanış panolarını kullanabilirsiniz).

Varsayılan kuyruk yöneticisinden başka bir kuyruk yöneticisi kullanmanız gerekebilir. Bu durumda, aşağıdakileri çağırın bir komut dosyası tanımlayın:

```
amqcrs6a -m Queue_Man_Name
```

daha sonra komut dosyasını çağırın.

## **IBM i** **IBM i üzerinde kanalların izlenmesi ve denetlenmesi**

Uzak kuyruk yöneticilerine kanallar yaratmak, izlemek ve denetlemek için DQM komutlarını ve panolarını kullanın. Her kuyruk yöneticisinin, uyumlu uzak kuyruk yöneticileriyle olan bağlantıları denetlemek için bir DQM programı vardır.

### **Bu görev hakkında**

Aşağıdaki liste, kanal denetimi işlevinin bileşenlerinin kısa bir açıklamasıdır:

- Kanal tanımlamaları kuyruk yöneticisi nesnelere olarak tutulur.
- Kanal komutları, IBM MQ for IBM i komut kümesinin bir alt kümesidir.
  - IBM MQ for IBM i komutlarının tam kümesini görüntülemek için GO CMDMQM komutunu kullanın.
- Kanal tanımlama panolarını ya da komutları aşağıdaki işlemleri gerçekleştirmek için kullanabilirsiniz:
  - Kanal tanımlarının yaratılması, kopyalanması, görüntülenmesi, değiştirilmesi ve silinmesi
  - Bağlantılar yeniden oluşturulmadığı zaman kanalları başlat ve durdur, ping, kanal sıra numaralarını sıfırla ve belirsiz iletileri çözümü
  - Kanallara ilişkin durum bilgilerini görüntüle
- Kanallar MQSC kullanılarak da yönetilebilir
- Kanallar IBM MQ Explorer kullanılarak da yönetilebilir
- Sıra numaraları ve *mantıksal iş birimi (LUW)* tanıtıcıları eşzamanlama dosyasında saklanır ve kanal eşzamanlama amacıyla kullanılır.

İleti kanallarını ve ilişkili nesnelere tanımlamak, ileti kanallarını izlemek ve denetlemek için komutları ve panoları kullanabilirsiniz. F4=Prompt anahtarını kullanarak ilgili kuyruk yöneticisini belirtebilirsiniz. Bilgi

istemini kullanmazsanız, varsayılan kuyruk yöneticisi varsayılır. F4=Promptile, ilgili kuyruk yöneticisi adını ve bazen de diğer verileri girebileceğiniz ek bir pano görüntülenir.

Panolarla tanımlamanız gereken nesnelere şunlardır:

- İletim kuyrukları
- Uzak kuyruk tanımlamaları
- Kuyruk yöneticisi diğer ad tanımlamaları
- Yanıtın gönderileceği kuyruk diğer adı tanımlamaları
- Yanıtlanacak yerel kuyruklar
- İletim kanalı tanımlamaları

Bu nesnelere kullanımıyla ilgili kavramlara ilişkin ek bilgi için bkz. [“Dağıtılmış kuyruğa alma yapılandırılıyor” sayfa 189.](#)

Bir kanalın başlatılabilmesi için kanallar tam olarak tanımlanmalı ve ilişkili nesnelere var olması ve kullanılabilir olması gerekir.

Ayrıca, bir kanalın çalıştırılabilmesi için her kanal için belirli bir iletişim bağlantısının tanımlanması ve kullanılabilir olması gerekir. LU 6.2 ve TCP/IP bağlantılarının nasıl tanımlandığına ilişkin açıklamalar için, kuruluşunuza ilişkin iletişim kılavuzuna bakın.

## Yordam

- Nesne yaratma ve nesnelere çalışma hakkında daha fazla bilgi için bkz:
  - [“IBM i üzerinde nesne yaratılması” sayfa 263](#)
  - [“IBM i üzerinde bir kanal oluşturma” sayfa 264](#)
  - [“IBM i üzerinde bir kanal başlatma” sayfa 266](#)
  - [“IBM i üzerinde bir kanal seçilmesi” sayfa 266](#)
  - [“IBM i üzerinde bir kanala göz atma” sayfa 267](#)
  - [“IBM i üzerinde bir kanalı yeniden adlandırma” sayfa 269](#)
  - [“IBM i üzerinde kanal durumuyla çalışma” sayfa 269](#)
  - [“IBM i üzerinde kanalla çalışma seçenekleri” sayfa 270](#)

## İlgili kavramlar

[“IBM i için iletişimi ayarlama” sayfa 276](#)

Bir dağıtılmış kuyruğa alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmayı dener. Başarılı olması için, bağlantının tanımlanması ve kullanılabilir olması gerekir.

## İlgili görevler

[“İstemci ile sunucu arasındaki bağlantıların yapılandırılması” sayfa 14](#)

IBM MQ MQI clients ile sunucular arasındaki iletişim bağlantılarını yapılandırmak için iletişim protokolünüze karar verin, bağlantının her iki ucundaki bağlantıları tanımlayın, bir dinleyici başlatın ve kanalları tanımlayın.

## İlgili başvurular

[Örnek yapılandırma- IBM MQ for IBM i](#)

[IBM MQ for IBM i için ileti kanalı planlama örneği](#)

[IBM MQ for IBM i CL komutları](#)

## IBM i IBM i üzerinde nesne yaratılması

Kuyruk ve diğer ad nesnelere yaratmak için CRTMQMQ komutunu kullanabilirsiniz.

Kuyruk ve diğer ad nesnelere yaratabilirsiniz: İletim kuyrukları, uzak kuyruk tanımlamaları, kuyruk yöneticisi diğer adı tanımlamaları, yanıtlanacak kuyruk diğer adı tanımlamaları ve yanıtlanacak yerel kuyruklar.

Varsayılan nesnelerin listesi için [Sistem ve varsayılan nesnelere](#) başlıklı konuya bakın.

## IBM i **IBM i üzerinde bir kanal oluşturma**

Kanal Yarat panosundan ya da komut satırındaki CRTMQMCHL komutunu kullanarak bir kanal yaratabilirsiniz.

Bir kanal oluşturmak için:

1. MQM Kanallarıyla Çalış panosunda (WRKMQMCHL) F6 komutunu kullanın.  
Diğer bir seçenek olarak, komut satırından CRTMQMCHL komutunu kullanın.  
Her iki şekilde de, Create Channel (Kanal Oluştur) panosu görüntülenir. Tür:
  - Sağlanan alandaki kanalın adı
  - Bağlantının bu ucuna ilişkin kanal tipi
2. Enter tuşuna basın.

**Not:** Ağınızdaki tüm kanalları benzersiz bir şekilde adlandırmalısınız. Kanal adındaki kaynak ve hedef kuyruk yöneticisi adları da içinde olmak üzere [Tüm kanalları gösteren ağ çizgesi](#) içinde gösterildiği gibi, bunu yapmak için iyi bir yoldur.

Girdileriniz doğrulanır ve hatalar hemen bildirilir. Hataları düzeltip devam edin.

Seçtiğiniz kanal tipi için uygun kanal ayarları panosu görüntülenir. Alanları daha önce topladığımız bilgilerle doldurun. Kanalı oluşturmak için Enter tuşuna basın.

Yardım panolarındaki kanal tanımlama panolarının açıklamalarında ve [Kanal öznelikler](#) alanında çeşitli alanların içeriğine karar vermenize yardımcı olur.

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Channel name . . . . . > CHANNAME_____
Channel type . . . . . > *SDR__ *RCVR, *SDR, *SVR, *RQSTR...
Message Queue Manager name *DFT_____

Replacē . . . . . *NO_ *NO, *YES
Transport type . . . . . *TCP___ *LU62, *TCP, *SYSDFTCHL
Text 'description' . . . . . > 'Example Channel Definition'_____

Connection name . . . . . *SYSDFTCHL_____

-----
More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```

Şekil 25. Kanal oluştur (1)

Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

```
Transmission queue . . . . . 'TRANSMISSION_QUEUE_NAME' _____
-----
Message channel agent . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
Message channel agent user ID . *SYSDFTCHL__ Character value...
Coded Character Set Identifier *SYSDFTCHL__ 0-9999, *SYSDFTCHL
Batch size . . . . . 50_____ 1-9999, *SYSDFTCHL
Disconnect interval . . . . . 6000_____ 1-999999, *SYSDFTCHL
Short retry interval . . . . . 60_____ 0-999999999, *SYSDFTCHL
Short retry count . . . . . 10_____ 0-999999999, *SYSDFTCHL
Long retry interval . . . . . 1200_____ 0-999999999, *SYSDFTCHL
Long retry count . . . . . 999999999__ 0-999999999, *SYSDFTCHL
Security exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
Security exit user data . . . . *SYSDFTCHL_____
-----
More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```

Şekil 26. Kanal oluştur (2)

Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

```
Send exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
+ for more values _____
Send exit user data . . . . . _____
+ for more values _____
Receive exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
+ for more values _____
-----
Receive exit user data . . . . . _____
+ for more values _____
Message exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
+ for more values _____
-----
More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```

Şekil 27. Kanal oluştur (3)

Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

```
Message exit user data . . . . . -----
+ for more values -----
Convert message . . . . . *SYSDFTCHL_ *YES, *NO, *SYSDFTCHL
Sequence number wrap . . . . . 99999999__ 100-99999999, *SYSDFTCHL
Maximum message length . . . . . 4194304____ 0-4194304, *SYSDFTCHL
Heartbeat interval . . . . . 300_____ 0-99999999, *SYSDFTCHL
Non Persistent Message Speed . . *FAST_____ *FAST, *NORMAL, *SYSDFTCHL
Password . . . . . *SYSDFTCHL_ Character value, *BLANK...
Task User Profile . . . . . *SYSDFTCHL_ Character value, *BLANK...
Transaction Program Name . . . . *SYSDFTCHL
```

Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys

Şekil 28. Kanal oluşturma (4)

## IBM i IBM i üzerinde bir kanal başlatma

Kanalları Kanallarla Çalış panosundan ya da komut satırındaki STRMQMCHL komutunu kullanarak başlatabilirsiniz.

Dinleyiciler yalnızca TCP için geçerlidir. SNA dinleyicileri için, iletişim altsisteminizi yapılandırmanız gerekir.

Uygulamaların ileti değış tokuşu yapabilmesi için, STRMQMLSR komutunu kullanarak gelen bağlantılar için bir dinleyici programı başlatmanız gerekir.

Giden bağlantılar için, kanalı aşağıdaki yollardan biriyle başlatmanız gerekir:

1. MCATYPE parametresine bağlı olarak kanalı bir işlem ya da iş parçacığı olarak başlatmak için kanal adını belirterek STRMQMCHL CL komutunu kullanın. (Kanallar iş parçacığı olarak başlatılırsa, bunlar bir kanal başlatıcısının iş parçacıklarıdır.)

```
STRMQMCHL CHLNAME(QM1.TO.QM2) MQNAME(MYQMGR)
```

2. Kanalı tetikleme için bir kanal başlatıcı kullanın. Bir kanal başlatıcı, kuyruk yöneticisi başlatıldığında otomatik olarak başlatılır. Bu otomatik başlatma, o kuyruk yöneticisine ilişkin qm.ini dosyasındaki chinit kısmı değıştirilerek ortadan kaldırılabilir.
3. Kanallarla Çalış panosuna başlamak için WRMQMCHL komutunu kullanın ve kanal başlatmak için seçenek 14 'ü belirleyin.

## IBM i IBM i üzerinde bir kanal seçilmesi

Kanallarla Çalış panosundan bir kanal seçebilirsiniz.

Bir kanal seçmek için, WRMQMCHL komutunu kullanarak Kanallarla Çalış panosundan başlayın:

1. İmleci, gerekli kanal adıyla ilişkili seçenek alanına taşıyın.
2. Bir seçenek numarası yazın.
3. Seçiminizi etkinleştirmek için Enter tuşuna basın.

Birden çok kanal seçerseniz, seçenekler sırayla etkinleştirilir.

Work with MQM Channels

Queue Manager Name . . : CNX

Type options, press Enter.

2=Change 3=Copy 4=Delete 5=Display 8=Work with Status 13=Ping  
14=Start 15=End 16=Reset 17=Resolve

Opt	Name	Type	Transport	Status
	CHLNIC	*RCVR	*TCP	INACTIVE
	CORSAIR.TO.MUSTANG	*SDR	*LU62	INACTIVE
	FV.CHANNEL.MC.DJE1	*RCVR	*TCP	INACTIVE
	FV.CHANNEL.MC.DJE2	*SDR	*TCP	INACTIVE
	FV.CHANNEL.MC.DJE3	*RQSTR	*TCP	INACTIVE
	FV.CHANNEL.MC.DJE4	*SVR	*TCP	INACTIVE
	FV.CHANNEL.PETER	*RCVR	*TCP	INACTIVE
	FV.CHANNEL.PETER.LU	*RCVR	*LU62	INACTIVE
	FV.CHANNEL.PETER.LU1	*RCVR	*LU62	INACTIVE

More...  
Parameters or command  
===>  
F3=Exit F4=Prompt F5=Refresh F6=Create F9=Retrieve F12=Cancel  
F21=Print

Şekil 29. Kanallarla çalışma

## IBM i **IBM i üzerinde bir kanala göz atma**

Kanal Görüntüle panosundan ya da komut satırındaki DSPMQMCHL komutunu kullanarak bir kanala göz atabilirsiniz.

Bir kanalın ayarlarına göz atmak için, Kanal Görüntüle panosundan başlamak üzere WRKMQMCHL komutunu kullanın:

1. Gerekli kanal adına ilişkin 5 (Görüntüle) seçeneğini yazın.
2. Seçiminizi etkinleştirmek için Enter tuşuna basın.

Birden çok kanal seçerseniz, bu kanallar sırayla sunulur.

Diğer bir seçenek olarak, komut satırından DSPMQMCHL komutunu kullanabilirsiniz.

Bu, kanala ilişkin geçerli ayarların ayrıntılarıyla birlikte uygun Görüntü Kanalı panosunun görüntülenmesiyle sonuçlanır. Alanlar, [Kanal öznitelikleri](#) içinde açıklanır.

```
Display MQM Channel

Channel name . . . . . : ST.JST.2T01
Queue Manager Name . . . . . : QMREL
Channel type . . . . . : *SDR
Transport type . . . . . : *TCP
Text 'description' . . . . . : John's sender to WINSDOA1

Connection name . . . . . : MUSTANG

Transmission queue . . . . . : WINSDOA1

Message channel agent . . . . . :
Library . . . . . :
Message channel agent user ID : *NONE
Batch interval . . . . . : 0
Batch size . . . . . : 50
Disconnect interval . . . . . : 6000

F3=Exit F12=Cancel F21=Print
```

Şekil 30. TCP/IP kanalının görüntülenmesi (1)

```
Display MQM Channel

Short retry interval . . . . . : 60
Short retry count . . . . . : 10
Long retry interval . . . . . : 6000
Long retry count . . . . . : 10
Security exit . . . . . :
Library . . . . . :
Security exit user data . . . . . :
Send exit . . . . . :
Library . . . . . :
Send exit user data . . . . . :
Receive exit . . . . . :
Library . . . . . :
Receive exit user data . . . . . :
Message exit . . . . . :
Library . . . . . :
Message exit user data . . . . . :
More...

F3=Exit F12=Cancel F21=Print
```

Şekil 31. TCP/IP kanalının görüntülenmesi (2)



```
Display MQM Channel
Sequence number wrap . . . . . : 999999999
Maximum message length . . . . : 10000
Convert message . . . . . : *NO
Heartbeat interval . . . . . : 300
Nonpersistent message speed . . *FAST
```

Bottom

F3=Exit F12=Cancel F21=Print

Şekil 32. TCP/IP kanalının görüntülenmesi (3)

### **IBM i IBM i üzerinde bir kanalı yeniden adlandırma**

Kanallarla Çalış panosundan bir kanalı yeniden adlandırabilirsiniz.

Bir ileti kanalını yeniden adlandırmak için Kanallarla Çalış panosundan başlayın:

1. Kanalı sonlandır.
2. Yeni adla bir yineleme yaratmak için seçenek 3 'ü (Kopyala) kullanın.
3. Doğru yaratılıp yaratılmadığını denetlemek için seçenek 5 'i (Görüntü) kullanın.
4. Özgün kanalı silmek için seçenek 4 'ü (Sil) kullanın.

Bir ileti kanalını yeniden adlandırmaya karar verirsiniz, her iki kanalın da aynı anda yeniden adlandırıldığından emin olun.

### **IBM i IBM i üzerinde kanal durumuyla çalışma**

Kanal Durumuyla Çalış panosundan kanal durumuyla çalışabilirsiniz.

Kanallarınızın durumunu gösteren ilk pano kümesini görüntülemek için WRKMQMCHST komutunu kullanın. Değişiklik Görünümü (F11) seçeneğini belirlediğinizde durum panolarını sırayla görüntüleyebilirsiniz.

Diğer bir seçenek olarak, Work with MQM Channel (MQM Kanallarıyla Çalış) panosunda seçenek 8 'in (Work with Status) belirlenmesi ilk durum panosunu da görüntüler.

## MQSeries Work with Channel Status

Type options, press Enter.

5=Display 13=Ping 14=Start 15=End 16=Reset 17=Resolve

Opt Name	Connection	Indoubt	Last Seq
CARTS_CORSAIR_CHAN	GBIBMIYA.WINSDOA1	NO	1
CHLNIC	9.20.2.213	NO	3
FV.CHANNEL.PETER2	9.20.2.213	NO	6225
JST.1.2	9.20.2.201	NO	28
MP_MUST_TO_CORS	9.20.2.213	NO	100
MUSTANG.TO.CORSAIR	GBIBMIYA.WINSDOA1	NO	10
MP_CORS_TO_MUST	9.20.2.213	NO	101
JST.2.3	9.5.7.126	NO	32
PF_WINSDOA1_LU62	GBIBMIYA.IYA80020	NO	54
PF_WINSDOA1_LU62	GBIBMIYA.WINSDOA1	NO	500
ST.JCW.EXIT.2T01.CHL	9.20.2.213	NO	216

Bottom

Parameters or command

===>

F3=Exit F4=Prompt F5=Refresh F6=Create F9=Retrieve F11=Change view  
F12=Cancel F21=Print

### Şekil 33. Kanal durumu panoları kümesinin ilki

Kanal Durumuyla Çalış panosunda kullanılabilir seçenekler şunlardır:

Menü seçeneği	Açıklama
5=Display	Kanal ayarlarını görüntüler.
13=Ping	Uygun olduğunda bir Ping işlemi başlatır.
14=Start	Kanalı başlatır.
15=End	Kanalı durdurur.
16=Reset	Kanal sıra numarasını ilk durumuna getirir.
17=Resolve	Belirsiz bir kanal durumunu el ile çözer.

## IBM i üzerinde kanalla çalışma seçenekleri

WRKMQMCHL komutuyla Kanallarla Çalışma panosuna ulaşılır ve listelenen tüm kanalların durumunu izlemenizi ve seçilen kanallar için komut vermenizi sağlar.

Kanalla Çalış panosunda kullanabileceğiniz seçenekler şunlardır:

Menü seçeneği	Açıklama
<u>"2=Change" sayfa 271</u>	Bir kanalın özniteliklerini değiştirir.
<u>"3=Copy" sayfa 271</u>	Bir kanalın özniteliklerini yeni bir kanala kopyalar.
<u>"4=Delete" sayfa 271</u>	Bir kanalı siler.
<u>"5=Display" sayfa 271</u>	Kanala ilişkin geçerli ayarları görüntüler.
<u>"6=Create" sayfa 271</u>	Create channel (Kanal oluştur) panosunu görüntüler
<u>"8=Work" sayfa 272</u>	Kanal durumu panolarını görüntüler.
<u>"13=Ping" sayfa 272</u>	Uzak uçla sabit bir veri iletisi değişik tokuş ederek bitişik sistemle bağlantıyı sınamak için Ping olanağını çalıştırır.
<u>"14=Start" sayfa 273</u>	Seçilen kanalı başlatır ya da devre dışı bırakılmış bir alıcı kanalı ilk durumuna getirir.
<u>"15=End" sayfa 274</u>	Kanalın kapanmasını istiyor.

## Menü seçeneği

## Açıklama

<a href="#">“16=Reset” sayfa 274</a>	Kanalın, bağlantısının bu ucundaki sıra numaralarını sıfırlamasını ister. Kanalın başlaması için sayılar her iki uçta da eşit olmalıdır.
<a href="#">“17=Resolve” sayfa 275</a>	Kanalın diğer uçla bağlantı kurmadan belirsiz iletileri çözmesini ister.
<a href="#">“18=Display yetkisi” sayfa 275</a>	IBM MQ nesne yetkisini görüntüler
<a href="#">“19=Grant” sayfa 275</a>	IBM MQ nesne yetkisi verir
<a href="#">“20=Revoke” sayfa 275</a>	IBM MQ nesne yetkisini geri alır
<a href="#">“21=Recover nesnesi” sayfa 275</a>	IBM MQ nesnesini kurtarır
<a href="#">“22=Record görüntüsü” sayfa 276</a>	Kayıtlar IBM MQ nesne resmi

## IBM i **2=Change**

Var olan bir kanal tanımını değiştirmek için Değiştir seçeneğini kullanın.

Değiştir seçeneği ya da CHGMQMCHL komutu, kanal adı dışında, varolan bir kanal tanımını değiştirir. Kanal tanımlama panosunda değiştirilecek alanların üzerine yazın ve Enter tuşuna basarak güncellenen tanımlamayı kaydedin.

## IBM i **3=Copy**

Var olan bir kanalı kopyalamak için Kopyala seçeneğini kullanın.

Kopyala seçeneği, var olan bir kanalı kopyalamak için CPYMQMCHL komutunu kullanır. Copy (Kopyala) panosu, yeni kanal adını tanımlamanızı sağlar. Ancak, kullanılan karakterleri IBM i nesne adları için geçerli olan karakterlerle sınırlamanız gerekir; bkz. [Yönetme IBM MQ for IBM i](#).

Geçerli ayarların ayrıntılarını görüntülemek için Copy (Kopyala) panosunda Enter tuşuna basın. Yeni kanal ayarlarından herhangi birini değiştirebilirsiniz. Enter tuşuna basarak yeni kanal tanımını kaydedin.

## IBM i **4>Delete**

Seçilen kanalı silmek için Sil seçeneğini kullanın.

İsteğinizi onaylamak ya da iptal etmek için bir pano görüntülenir.

## IBM i **5=Display**

Kanala ilişkin yürürlükteki tanımlamaları görüntülemek için Görüntüle seçeneğini kullanın.

Bu seçenek, panoyu parametrelerin yürürlükteki değerlerini gösteren ve kullanıcı girişine karşı korunan alanlarla birlikte görüntüler.

## IBM i **6=Create**

Create channel (Kanal yarat) panosunu görüntülemek için Create (Oluştur) seçeneğini kullanın.

Kanal Oluştur panosunu almak için Oluştur seçeneğini kullanın ya da komut satırından CRTMQMCHL komutunu girin. [Şekil 25 sayfa 264](#) adresinden başlayan Create Channel panoları örnekleri vardır.

Bu panoda, IBM MQ for IBM tarafından sağlanan varsayılan değerlerle doldurulan alanların ekranından bir kanal tanımı yaratırsınız. Kanalın adını yazın, yaratmakta olduğunuz kanal tipini ve kullanılacak iletişim yöntemini seçin.

Enter tuşuna bastığınızda pano görüntülenir. Bu panodaki tüm zorunlu alanlara ve geri kalan panolara bilgi yazın ve Enter tuşuna basarak tanımlamayı kaydedin.

Kanal adı, kanalın her iki ucunda da aynı olmalı ve ağ içinde benzersiz olmalıdır. Ancak, kullanılan karakterleri IBM MQ for IBM i nesne adları için geçerli olan karakterlerle sınırlamanız gerekir.

Tüm panoların bazı alanlar için IBM MQ for IBM i tarafından sağlanan varsayılan değerleri vardır. Bu değerleri özelleştirebilir ya da kanalları oluştururken ya da kopyalarken değiştirebilirsiniz. Değerleri özelleştirmek için bkz. *IBM MQ for IBM i Sistem Yönetimi*.

Her kanal tipi için gerekli varsayılan değerlerle kukla kanallar kurarak ve bunları her yeni kanal tanımlamaları yaratmak istediğinizde kopyalayarak kendi kanal varsayılan değerleri kümenizi oluşturabilirsiniz.

### **İlgili başvurular**

Kanal öznitelikleri

#### **IBM i 8=Work**

Ayrıntılı kanal durumu bilgilerini görmek için Durumla Çalış seçeneğini kullanın.

Durum kolonu, kanalın etkin olup olmadığını ve MQM Kanallarıyla Çalış panosunda sürekli olarak görüntülenip görüntülenmediğini gösterir. Görüntülenen diğer durum bilgilerini görmek için seçenek 8 'i (Durumla Çalışma) kullanın. Diğer bir seçenek olarak, bu bilgiler komut satırından WRKMQMCHST komutuyla görüntülenebilir. Bkz. "IBM i üzerinde kanal durumuyla çalışma" sayfa 269.

- Kanal adı
- Kanal tipi
- Kanal durumu
- Kanal örneği
- Uzak kuyruk yöneticisi
- İletim kuyruğu adı
- İletişim bağlantısı adı
- Kanalın belirsiz durumu
- Son sıra numarası
- Belirsiz ileti sayısı
- Belirsiz sıra numarası
- İletim kuyruğundaki ileti sayısı
- Mantıksal iş birimi tanıtıcısı
- Belirsiz mantıksal iş birimi tanıtıcısı
- Kanal alt durumu
- Kanal izleme
- Üstbilgi sıkıştırması
- İleti sıkıştırması
- Sıkıştırma süresi göstergesi
- Sıkıştırma hızı göstergesi
- İletim kuyruğu zaman göstergesi
- Ağ saati göstergesi
- Çıkış zamanı göstergesi
- Toplu iş boyutu göstergesi
- Geçerli paylaşılan sohbetler
- Paylaşılan etkileşim sayısı üst sınırı

#### **IBM i 13=Ping**

Sabit bir veri iletisini uzak uçla değiş tokuş etmek için Ping seçeneğini kullanın.

Başarılı bir IBM MQ Ping, sistem gözetmenine kanalın kullanılabilir ve çalışır durumda olduğuna dair biraz güven verir.

Ping komutu, iletim kuyruklarının ve hedef kuyrukların kullanılmasını içermez. Kanal tanımlarını, ilgili iletişim bağlantısını ve ağ ayarlarını kullanır.

Yalnızca gönderen ve sunucu kanallarından kullanılabilir. İlgili kanal, bağlantının uzak tarafında başlatılır ve başlatma parametresi kararlaştırmasını gerçekleştirir. Hatalar olağan şekilde bildirilir.

İleti değiş tokuşunun sonucu sizin için Ping panosunda sunulur ve iletinin gönderildiği saat ve yanıtın alındığı saatle birlikte döndürülen ileti metnidir.

## 6.2 LU ile ping komutu gönder

Ping komutu IBM MQ for IBM i içinde çağrıldığında, işlevi isteyen kullanıcının kullanıcı kimliği ile çalıştırılır; ancak kanal programının olağan çalışma şekli, kanal programları için alınacak QMQM kullanıcı kimliği içindir. Kullanıcı kimliği alıcı tarafa akar ve LU 6.2 etkileşimin ayrılması için alma ucunda geçerli olmalıdır.

### IBM i 14=Start

Bir kanalı el ile başlatmak için Başlat seçeneğini kullanın.

Başlat seçeneği gönderen, sunucu ve istekte bulunan kanallar için kullanılabilir. Kuyruk yöneticisi tetikleyicisiyle bir kanal ayarlandığında bu gerekli değildir.

Başlat seçeneği, alıcı, sunucu bağlantısı, küme gönderen ve küme alıcı kanalları için de kullanılır. DURDURULDU durumundaki bir alıcı kanalının başlatılması, uzak kanaldan başlatılabileceği anlamına gelir.

Başlatıldığında, gönderen MCA kanal tanımlama dosyasını okur ve iletim kuyruğunu açar. Alıcı ya da sunucu kanalının karşılık gelen MCA ' yı uzaktan başlatan bir kanal başlatma sırası yayınlandı. Bunlar başlatıldığında, gönderen ve sunucu, iletim kuyruğuna gelen iletileri bekler ve vardıklarında iletir.

Tetikleme özelliğini kullandığınızda, başlatma kuyruğunu izlemek için sürekli olarak çalışan tetikleme işlemi başlatmanız gerekir. İşlemi başlatmak için STRMQMCHLI komutu kullanılabilir.

Bir kanalın uzak ucunda, gönderme ucundan kanal başlatmaya yanıt olarak alma işlemi başlatılabilir. Bunu yapma yöntemi LU 6.2 ve TCP/IP bağlantılı kanallar için farklıdır:

- LU 6.2 bağlantılı kanallar, bir kanalın alıcı ucunda belirttik bir işlem yapılmasını gerektirmez.
- TCP bağlantılı kanallar, bir dinleyici işleminin sürekli olarak çalışmasını gerektirir. Bu işlem, bağlantının uzak ucundan kanal başlatma isteklerini bekler ve o bağlantıya ilişkin kanal tanımlamalarında tanımlanan işlemi başlatır.

Uzak sistem IBM i olduğunda, STRMQMLSR komutunu kullanabilirsiniz.

Başlat seçeneğinin kullanılması, gerektiğinde kanalın yeniden eşzamanlanmasına neden olur.

Başlangıçların başarılı olması için:

- Kanal tanımları, yerel ve uzak var olmalıdır. Bir alıcı ya da sunucu bağlantısı kanalı için uygun bir kanal tanımı yoksa, kanal otomatik olarak tanımlanır, varsayılan bir kanal otomatik olarak yaratılır. Bkz. [Kanal otomatik tanımlama çıkış programı](#).
- İletim kuyruğu var olmalı, GETler için etkinleştirilmeli ve bunu kullanan başka kanal olmamalıdır.
- Yerel ve uzak MCA ' lar var olmalıdır.
- İletişim bağlantısı kullanılabilir olmalıdır.
- Kuyruk yöneticileri çalışıyor, yerel ve uzak olmalıdır.
- İleti kanalı etkin değil olmalıdır.

İletileri aktarmak için uzak kuyruklar ve uzak kuyruk tanımlamaları varolmalıdır.

Panoya, kanal başlatma isteğinin kabul edildiğini doğrulayan bir ileti gönderilir. Başlatma işleminin başarılı olduğunu doğrulamak için sistem günlüğünü denetleyin ya da F5 tuşuna basın (ekranı yenileyin).

## IBM i 15=End

Kanal etkinliğini durdurmak için Sona Erdir özelliğini kullanın

Kanalı etkinliği durdurması için istekte bulunmak üzere Son seçeneğini kullanın. Kanal başka ileti göndermiyor.

Kanalın DURDURULDU mu, yoksa INACTIVE mu olacağını ve kanalı CHECKED ya da IMMEDIATE durarak mı durdurulacağını belirlemek için Enter tuşuna basmadan önce F4 seçeneğini belirleyin. Durdurulan bir kanal, yeniden etkin olabilmesi için işletmen tarafından yeniden başlatılmalıdır. Etkin olmayan bir kanal tetiklenebilir.

## Hemen durdur

Herhangi bir iş birimini tamamlamadan bir kanalı durdurmak için Hemen Durdur 'u kullanın.

Bu seçenek kanal işlemini sonlandırır. Sonuç olarak kanal, geçerli ileti kümesinin işlenmesini tamamlamaz ve bu nedenle kanalı şüpheli olarak bırakamaz. Genel olarak, operatörlerin denetimli durdurma seçeneğini kullanması daha iyidir.

## Denetimli durdur

Yürürlükteki iş biriminin sonunda bir kanalı durdurmak için Stop (Durdur) denetimli işlevini kullanın.

Bu seçenek, kanalın düzgün bir şekilde kapatılmasını ister; geçerli ileti kümesi tamamlanır ve eşitleme noktası yordamı kanalın diğer ucuyla gerçekleştirilir.

## Durdurulan kanallar yeniden başlatılıyor

Bir kanal DURDURULDU durumuna geçtiğinde, kanalı el ile yeniden başlatmanız gerekir. Kanalı aşağıdaki şekillerde yeniden başlatabilirsiniz:

- **START CHANNEL** MQSC komutunu kullanarak.
- **Start Channel** PCF komutunu kullanarak.
- IBM MQ Explorer komutunu kullanarak.
- **z/OS** z/OS üzerinde, Start a channel (Kanal başlat) panosunu kullanarak.
- **IBM i** IBM sistemlerinde, WRKMQMCHL panosunda **STRMQMCHL CL** komutunu ya da **START** seçeneğini kullanarak.

Gönderen ya da sunucu kanalları için, kanal DURDURULDU durumuna girdiğinde, ilişkili iletim kuyruğu GET (DISABLED) olarak ayarlandı ve tetikleme geçersiz kılındı. Başlatma isteği alındığında, bu öznitelikler otomatik olarak sıfırlanır.

**z/OS** Kanal başlatıcısı, kanal REDENYOR ya da DURDURULDU durumundayken durursa, kanal başlatıcısı yeniden başlatıldığında kanal durumu hatırlanır. Ancak kanal DURDURULDU durumundayken kanal başlatıcısı durursa, SVRCONN kanal tipine ilişkin kanal durumu ilk durumuna getirilir.

**Multi** Bir kanal REDENYOR ya da DURDURULDU durumundayken kuyruk yöneticisi durursa, kuyruk yöneticisi yeniden başlatıldığında kanal durumu hatırlanır. IBM MQ 8.0 ' den itibaren bu, SVRCONN kanalları için de geçerlidir. Daha önce, kanal DURDURULDU durumundayken kanal başlatıcısı durdurulduysa, SVRCONN kanal tipine ilişkin kanal durumu sıfırlanmıştı.

## IBM i 16=Reset

Yeni bir ileti sırasını zorlamak için Sıfırla seçeneğini kullanın.

Sıfırla seçeneği, ileti sıra numarasını değiştirir. Bunu dikkatli bir şekilde ve yalnızca belirsiz durumları çözmek için Çözümle seçeneğini kullandıktan sonra kullanın. Bu seçenek yalnızca gönderen ya da sunucu kanalında kullanılabilir. İlk ileti, kanal yeniden başlatıldığında yeni sırayı başlatır.

## IBM i **17=Resolve**

Yerel kesinleştirme ya da iletim kuyruğunda tutulan belirsiz iletilerin geriletmesini zorlamak için Çözümle seçeneğini kullanın.

İletiler bir gönderen ya da sunucu tarafından kuşkuyla tutulduğunda Çözümle seçeneğini kullanın; örneğin, bağlantının bir ucu sonlandırıldı ve kurtarma olasılığı yok. Çözme seçeneği şu iki parametreden birini kabul eder: BACKOUT ya da COMMIT. Geriletme, iletileri iletim kuyruğuna geri yüklerken, Kesinleştirme bunları atar.

Kanal programı, bir iş ortağıyla oturum kurmaya çalışmaz. Bunun yerine, belirsiz iletileri gösteren mantıksal iş birimi tanıtıcısını (LUWID) belirler. Daha sonra, istendiği gibi, aşağıdaki sorunlardan biri ortaya konmaktadır:

- İletileri iletim kuyruğuna geri yüklemek için BACKOUT; ya da
- İletileri iletim kuyruğundan silmek için COMMIT.

Çözümün başarılı olması için:

- Kanal etkin değil olmalıdır
- Kanal şüphe içinde olmalı.
- Kanal tipi gönderen ya da sunucu olmalıdır
- Yerel kanal tanımı var olmalıdır
- Kuyruk yöneticisi çalışıyor olmalı, yerel

## IBM i **18=Display yetkisi**

Kullanıcının belirli bir IBM MQ nesnesi üzerinde hangi işlemleri gerçekleştirme yetkisine sahip olduğunu görüntülemek için yetkiyi görüntüle seçeneğini kullanın.

Seçilen bir nesne ve kullanıcı için, DSPMQAUT komutu kullanıcının IBM MQ nesnesi üzerinde işlem gerçekleştirmesi gereken yetkileri gösterir. Kullanıcı birden çok grubun üyesiye, komut, nesne için tüm grupların birleşik yetkilendirmesini gösterir.

## IBM i **19=Grant**

IBM MQ nesneleri üzerinde başka bir kullanıcıya ya da kullanıcı grubuna işlem gerçekleştirme yetkisi vermek için Yetki Ver seçeneğini kullanın.

GRTMQMAUT komutu yalnızca QMQMADM grubundaki kullanıcılar tarafından kullanılabilir. QMQMADM 'deki bir kullanıcı, diğer kullanıcılara, komutta adı belirtilen IBM MQ nesneleri üzerinde, kullanıcıları ada göre tanıyarak ya da \*PUBLIC içindeki tüm kullanıcılara yetki vererek işlem gerçekleştirme yetkisi verir.

## IBM i **20=Revoke**

Kullanıcılardan nesnelere üzerinde işlem gerçekleştirme yetkisini kaldırmak için yetkiyi iptal edin.

RVKMQMAUT komutu yalnızca QMQMADM grubundaki kullanıcılar tarafından kullanılabilir. QMQMADM grubundaki bir kullanıcı, kullanıcıları adlarına göre tanıyarak ya da \*PUBLIC içindeki tüm kullanıcıların yetkilerini iptal ederek, komutta adı belirtilen IBM MQ nesneleri üzerinde işlem gerçekleştirme yetkisini diğer kullanıcılardan kaldırır.

## IBM i **21=Recover nesnesi**

IBM MQ günlüklerinde saklanan bilgilerden zarar görmüş nesnelere geri yüklemek için Kurtarın nesnesini kullanın.

Kurtarma nesnesi, komutta adı belirtilen zarar görmüş tüm nesnelere kurtarmak için MQ Nesnesi Yeniden Yarat komutunu (RCRMQMOBJ) kullanır. Bir nesne zarar görmediyse, o nesne üzerinde herhangi bir işlem yapılmaz.

## IBM i 22=Record görüntüsü

Bir nesne kümesinin kurtarılması için gereken günlük nesnesi sayısını azaltmak ve kurtarma süresini en aza indirmek için Kayıt görüntüsünü kullanın.

RCDMQMIMG komutu, komutta seçilen tüm nesnelere için bir denetim noktası alır. Tümleşik dosya sistemindeki (IFS) nesnelere yürürlükteki değerlerini, günlük nesnelere kaydedilen MQPUTs ve MQGETs gibi nesnelere ilgili daha sonraki bilgilerle uyumlulaştırır.

Komut tamamlandığında, IFS ' deki nesnelere güncel olur ve nesnelere kurtarmak için bu günlük nesnelere mevcut olması gerekmez. Bağlantısı kesilen günlük nesnelere ayrılabilir (diğer nesnelere kurtarmak için var olması gerekmediği sürece).

## IBM i için iletişimi ayarlama

Bir dağıtım kuyruğuna alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmayı dener. Başarılı olması için, bağlantının tanımlanması ve kullanılabilir olması gerekir.

DQM, IBM MQ for IBM için uzak bir kuyruğuna alma olanağıdır. Sistem işletmeni tarafından denetlenebilir iletişim bağlantılarına arabirim oluşturan IBM MQ for IBM i kuyruk yöneticisi için kanal denetim programları sağlar.

Bir dağıtım kuyruğuna alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmayı dener. Başarılı olması için, bağlantının tanımlanması ve kullanılabilir olması gerekir. Bu bölümde, bağlantının tanımlandığından ve kullanılabilir olduğundan nasıl emin olabileceğiniz açıklanmaktadır.

Bir kanal başlatılmadan önce, iletim kuyruğunun bu bölümde açıklandığı gibi tanımlanması ve ileti kanalı tanımına eklenmesi gerekir.

IBM MQ for IBM i sistemleri arasında aşağıdaki iki iletişim biçiminden birini seçebilirsiniz:

- [“IBM i üzerinde TCP bağlantısı tanımlanması” sayfa 277](#)

TCP için bir anasistem adresi kullanılabilir ve bu bağlantılar *IBM i Communication Configuration Reference* adlı yayında açıklandığı gibi ayarlanır.

TCP ortamında, her dağıtılmış hizmete, hizmete erişmek için uzak makineler tarafından kullanılacak benzersiz bir TCP adresi ayrılır. TCP adresi, bir anasistem adı/numarası ve bir kapı numarasından oluşur. Tüm kuyruk yöneticileri TCP aracılığıyla birbirleriyle iletişim kurmak için böyle bir sayı kullanır.

- [“TCP ' de alma” sayfa 277](#)

Bu iletişim biçimi, yerel kuyruk yöneticisine hizmet veren IBM i sistemi ile uzak kuyruk yöneticisine hizmet veren sistem arasında fiziksel bağlantı sağlayan IBM i SNA mantıksal birim tipi 6.2 (LU 6.2) tanımlamasını gerektirir. IBM içinde iletişim yapılandırılmasına ilişkin ayrıntılar için *IBM i Communication Configuration Reference* adlı yayına bakın.

Gerektiğinde, tetikleme düzenlemesi de gerekli süreç ve kuyrukların tanımlanmasıyla birlikte hazırlanmalıdır.

MQ Adv.

CD

TCP/IP kullanan bir ileti kanalı, ağ verimini önemli ölçüde artıracak hızlı bir TCP/IP tüneli sağlayan bir IBM Aspera faspio Gateway' e işaret edilebilir. Herhangi bir yetkili platformda çalışan bir kuyruk yöneticisi, Aspera gateway aracılığıyla bağlanabilir. Ağ geçidinin kendisi Red Hat ya da Ubuntu Linux ya da Windows üzerinde devreye alınır. Bkz. [Linux ya da Windows üzerinde Aspera gateway bağlantısı tanımlanması](#).

### İlgili görevler

[“IBM i üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 262](#)

Uzak kuyruk yöneticilerine kanallar yaratmak, izlemek ve denetlemek için DQM komutlarını ve panolarını kullanın. Her kuyruk yöneticisinin, uyumlu uzak kuyruk yöneticileriyle olan bağlantıları denetlemek için bir DQM programı vardır.

### İlgili başvurular

[Örnek yapılandırma- IBM MQ for IBM i](#)

[IBM MQ for IBM i için ileti kanalı planlama örneği](#)



## IBM i **IBM i üzerinde TCP bağlantısı tanımlanması**

Bağlantı Adı alanını kullanarak kanal tanımında bir TCP bağlantısı tanımlayabilirsiniz.

Kanal tanımı, hedefin TCP ağ adresini ya da anasistem adını (örneğin, ABCHOST) içeren bir alan (CONNECTION NAME) içeriyor. TCP ağ adresi IPv4 noktalı onlu biçimde (örneğin, 127.0.0.1) ya da IPv6 onaltılı biçimde (örneğin, 2001:DB8:0:0:0:0:0:0) olabilir. CONNECTION NAME bir anasistem adı ya da ad sunucusuysa, anasistem adını TCP anasistem adresine dönüştürmek için IBM i anasistem çizelgesi kullanılır.

Tam TCP adresi için bir kapı numarası gerekir; bu numara sağlanmazsa, varsayılan kapı numarası 1414 kullanılır. Bir bağlantının başlatılırken (gönderen, istekte bulunan ve sunucu kanal tipleri), bağlantı için isteğe bağlı bir kapı numarası sağlanmalıdır; örneğin:

```
Connection name 127.0.0.1 (1555)
```

Bu durumda, başlatıcı sona erdirme işlemi 1555 numaralı kapıdaki bir alıcı programa bağlanmayı dener.

**MQ Adv.** **CD** TCP/IP kullanan bir ileti kanalı, ağ verimini önemli ölçüde artıracak hızlı bir TCP/IP tüneli sağlayan bir IBM Aspera faspio Gateway' e işaret edilebilir. Herhangi bir yetkili platformda çalışan bir kuyruk yöneticisi, Aspera gateway aracılığıyla bağlanabilir. Ağ geçidinin kendisi Red Hat ya da Ubuntu Linux ya da Windows üzerinde devreye alınır. Bkz. [Linux ya da Windows üzerinde Aspera gateway bağlantısı tanımlanması](#).

## TCP dinleyici birikim listesi seçeneğinin kullanılması

TCP ' de, sunucu ile istemci arasında üç yönlü tokalaşma gerçekleşmediği sürece bağlantılar eksik olarak işlenir. Bu bağlantılara, bekleyen bağlantı istekleri denir. Bu bekleyen bağlantı istekleri için bir değer üst sınırı belirlenir ve dinleyicinin isteği kabul etmesi için TCP kapısında bekleyen isteklerin birikim listesi olarak kabul edilebilir.

Daha fazla bilgi ve IBM için özel değer için bkz. [“IBM MQ for Multiplatforms üzerinde TCP dinleyici birikim listesi seçeneğinin kullanılması” sayfa 259](#).

### İlgili kavramlar

“TCP ' de alma” sayfa 277

Kanal programlarını alma işlemi, gönderen kanaldan gelen başlatma isteğine yanıt olarak başlatılır. Başlatma isteğine yanıt vermek için, gelen ağ isteklerini saptamak ve ilişkili kanalı başlatmak için bir dinleyici programının başlatılması gerekir. Bu dinleyici programını STRMQMLSR komutuyla başlatabilirsiniz.

### IBM i **TCP ' de alma**

Kanal programlarını alma işlemi, gönderen kanaldan gelen başlatma isteğine yanıt olarak başlatılır. Başlatma isteğine yanıt vermek için, gelen ağ isteklerini saptamak ve ilişkili kanalı başlatmak için bir dinleyici programının başlatılması gerekir. Bu dinleyici programını STRMQMLSR komutuyla başlatabilirsiniz.

Her kuyruk yöneticisi için birden çok dinleyici başlatabilirsiniz. Varsayılan olarak, STRMQMLSR komutu 1414 kapısını kullanır, ancak bu değeri geçersiz kılabilirsiniz. Varsayılan ayarı geçersiz kılmak için, seçilen kuyruk yöneticisinin qm.ini dosyasına aşağıdaki deyimleri ekleyin. Bu örnekte, dinleyicinin 2500 numaralı kapıyı kullanması gerekir:

```
TCP:  
Port=2500
```

qm.ini dosyası şu IFS dizininde bulunur: /QIBM/UserData/mqm/qmgrs/ *kuyruk yöneticisi adi*.

Bu yeni deęer, TCP dinleyicisi bařlatıldıęında salt okunur olur. alıřmakta olan bir dinleyiciniz varsa, bu deęiřiklik o program tarafından grlmez. Yeni deęeri kullanmak iin dinleyiciyi durdurun ve STRMQMLSR komutunu yeniden verin. Őimdi, STRMQMLSR komutunu kullandıęınızda, dinleyici varsayılan olarak yeni kapıyı kullanır.

Dięer bir seenek olarak, STRMQMLSR komutunda farklı bir kapı numarası belirtebilirsiniz. rneęin:

```
STRMQMLSR MQMNAME( queue manager name ) PORT(2500)
```

Bu deęiřiklik, dinleyici iřinin sresi boyunca dinleyicinin varsayılan olarak yeni kapıya ayarlanmasını saęlar.

## TCP SO\_KEEPALIVE seeneęinin kullanılması

SO\_KEEPALIVE seeneęini kullanmak istiyorsanız (ek bilgi iin bkz. “Kanalın dięer ucunun hala kullanılabilir olup olmadıęı denetleniyor” sayfa 224 ) Ařaęıdaki giriři kuyruk yneticisi yapılıř ktęnze (IFS dizinindekiqm.ini , /QIBM/UserData/mqm/qmgrs/ *kuyruk yneticisi adı* ) eklemelisiniz:

```
TCP:
KeepAlive=yes
```

Daha sonra ařaęıdaki komutu alıřtırmanız gerekir:

```
CFGTCP
```

Seenek 3 ' (TCP zneliklerini deęiřtir) belirleyin. Őimdi dakika cinsinden bir zaman aralıęı belirtebilirsiniz. 1-40320 dakika aralıęında bir deęer belirtebilirsiniz; varsayılan deęer 120 'dir.

## TCP dinleyici birikim listesi seeneęinin kullanılması

TCP ' de alma sırasında, iřlem bekleyen baęlantı isteęi sayısı st sınırı belirlenir. Bu sayı, dinleyicinin isteęi kabul etmesi iin TCP kapısında bekleyen isteklerin *birikim listesi* olarak kabul edilebilir.

IBM i zerindeki varsayılan dinleyici birikim listesi deęeri 255 'tir. Birikim listesi bu deęere ulařırsa, TCP baęlantısı reddedilir ve kanal bařlatılamaz.

MCA kanalları iin bu, kanalın RETRY durumuna girmesine ve daha sonra baęlantıyı yeniden denemesine neden olur.

İstemci baęlantıları iin, istemci MQCONN ' dan bir MQRC\_Q\_MGR\_NOT\_VAR olan neden kodunu alır ve baęlantıyı daha sonra yeniden deneyebilir.

Ancak, bu hatayı nlemek iin qm.ini dosyasına bir girdi ekleyebilirsiniz:

```
ListenerBacklog = n
```

Bu, TCP dinleyicisi iin varsayılan iřlem bekleyen istek sayısı st sınırını (255) geersiz kılar.

**Not:** Bazı iřletim sistemleri varsayılan deęerden daha byk bir deęeri destekler. Gerekirse, baęlantı sınırına ulařılmasını nlemek iin bu deęer kullanılabilir.

## IBM i **IBM i zerinde LU 6.2 baęlantısı tanımlanması**

Tam olarak nitelenmiř LU 6.2 baęlantısının kip adını, TP adını ve baęlantı adını kullanarak LU 6.2 iletiřim ayrıntılarını tanımlayın.

Bu CSI nesnesini tamamlamak iin baęlantının bařlatılan ucunda bir yneltme giriři tanımlaması olmalıdır. Uzak LU 6.2 sistemlerinden iř isteklerinin ynetilmesiyle ilgili ek bilgi iin *IBM i Programming: Work Management Guide* belgesine bakın.

Bilgi iin *Multiplatform APPC Configuration Guide* belgesine ve ařaęıdaki izelgeye bakın.

Çizelge 25. Uzak kuyruk yöneticisi platformuna ilişkin yerel IBM i sistemindeki ayarlar

Uzak altyapı	TPNAME
z/OS ya da MVS	Uzak kuyruk yöneticisine ilişkin yan bilgilerle aynı.
IBM i	IBM i sistemindeki yöneltme girişindeki karşılaştırma değeriyle aynı.
AIX and Linux sistemleri	Uzak LU 6.2 yapılışında tanımlanan çağrılabilir hareket programı.
Windows	Windows Run Listener komutunda ya da Windowsüzerinde TpSetup kullanılarak tanımlanan çağrılabilir Hareket Programı 'nda belirtildiği şekilde.

Aynı bilgisayarda birden çok kuyruk yöneticiniz varsa, kanal tanımlamalarındaki TPadlarının benzersiz olmasına dikkat edin.

### İlgili kavramlar

[“Bitiş başlatılıyor \(Gönderen\)” sayfa 279](#)

\*LU62aktarım tipinde bir kanal tanımlamak için CRTMQMCHL komutunu kullanın.

[“Başlatılan sona erdirme \(Alıcı\)” sayfa 281](#)

\*LU62iletim tipiyle ileti kanalı bağlantısının alma sonunu tanımlamak için CRTMQMCHL komutunu kullanın.

**IBM i** *Bitiş başlatılıyor (Gönderen)*

\*LU62aktarım tipinde bir kanal tanımlamak için CRTMQMCHL komutunu kullanın.

CSI nesnesinin kullanımı, IBM MQ for IBM i V5.3 ya da sonraki bir yayın düzeyinde isteğe bağlıdır.

Başlangıç uç panosu, Şekil LU 6.2 iletişim ayarı paneli başlatma ucundagösterilir. Tam panoyu gösterildiği gibi almak için ilk panelden F10 tuşuna basın.

```
Create Comm Side Information (CRTCSI)
```

```
Type choices, press Enter.
```

```
Side information . . . . . > WINSDOA1 Name
Library . . . . . > QSYS Name, *CURLIB
Remote location . . . . . > WINSDOA1 Name
Transaction program . . . . . > MQSERIES
```

```
Text 'description' . . . . . *BLANK
```

```
Additional Parameters
```

```
Device . . . . . *LOC Name, *LOC
Local location . . . . . *LOC Name, *LOC, *NETATR
Mode . . . . . JSTMOD92 Name, *NETATR
Remote network identifier . . . *LOC Name, *LOC, *NETATR, *NONE
Authority . . . . . *LIBCRTAUT Name, *LIBCRTAUT, *CHANGE...
```

```
Bottom
```

```
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```

Şekil 34. LU 6.2 iletişim ayarı panosu-başlatma ucu

Başlangıç bitiş alanlarını aşağıdaki gibi tamamlayın:

### Yan bilgiler

Bu tanımlamaya, yaratılacak yan bilgi nesnesini saklamak için kullanılan bir ad verin; örneğin, WINSDOA1.

**Not:** LU 6.2 için, ileti kanalı tanımlaması ile iletişim bağlantısı arasındaki bağlantı, gönderen uçta ileti kanalı tanımlamasının **Bağlantı adı** alanıdır. Bu alan CSI nesnesinin adını içerir.

## Kitaplık

Bu tanımlamanın saklandığı kitaplığın adı.

CSI nesnesi, ileti kanalını sunan programın erişebileceği bir kitaplıkta (örneğin, QSYS, QMQM ve QGPL) bulunmalıdır.

Ad yanlışsa, eksikse ya da bulunamazsa, kanal başlatılırken bir hata ortaya çıkar.

## Uzak konum

Programınızın iletişim kurduğu uzak konum adını belirler.

Kısacası, bu zorunlu parametre, iki sistem arasındaki iletişim bağlantısı için kullanılan aygıt tanımlamasında tanımlandığı şekilde, uzak sistemdeki ortağın mantıksal birim adını içerir.

**Uzak yer** adı, uzak sistemde DSPNETA komutu çalıştırılarak ve varsayılan yerel yer adını görerek bulunabilir.

## Hareket işleme programı

Başlatılacak uzak sistemdeki hareket programının adını (en çok 64 karakter) belirler. Bir işlem işlemi adı, program adı, kanal adı ya da yöneltme girişindeki **Değeri karşılaştır** ile eşleşen bir karakter dizisi olabilir.

Bu parametre gereklidir.

**Not:** SNA hizmeti hareket programı adlarını belirtmek için, hizmet hareketi programı adının onaltılı gösterimini girin. Örneğin, 21F0F0F1onaltılı gösterimiyle bir hizmet hareketi programı adı belirtmek için X'21F0F0F1' girmeniz gerekir.

SNA hizmeti hareket programı adlarıyla ilgili ek bilgi için *SNA Transaction Programmer's Reference* adlı elkitabındaki LU Type 6.2adlı elkitabına bakın.

Alıcı uç başka bir IBM i sistemiyse, **Hareket programı** adı, gönderme ucundaki CSI nesnesini, giriş sonundaki yöneltme girişiyle eşleştirmek için kullanılır. Bu ad, hedef IBM i sistemindeki her kuyruk yöneticisi için benzersiz olmalıdır. **Program to call** parametresi için *Initiated end (Receiver)* başlıklı konuya bakın. Ayrıca, Add Routing Entry (Yönlendirme Girişi Ekle) panosunda **Comparison data: comparison value** (Karşılaştırma verileri: değer karşılaştırma) değiştirgesine bakın.

## Tanımlama metni

Bu bağlantının amaçlanan kullanımını anımsatacak bir tanımlama (en çok 50 karakter).

## Aygıt

Uzak sistem için kullanılan aygıt tanımlamasının adını belirler. Olası değerler şunlardır:

### \*LOC

Aygıt sistem tarafından belirlenir.

### Aygıt-adi

Uzak konumla ilişkili aygıtın adını belirtin.

## Yerel konum

Yerel konum adını belirtir. Olası değerler şunlardır:

### \*LOC

Yerel konum adı sistem tarafından belirlenir.

### \*NETATR

Sistem ağ özniteliklerinde belirlenen LCLLOCNAME değeri kullanılır.

### Yerel-konum-adi

Konumunuzun adını belirtin. Uzak konum için belirli bir konum adı belirtmek istiyorsanız yerel konumu belirtin. Konum adı, DSPNETA komutu kullanılarak bulunabilir.

## Mod

Oturumu denetlemek için kullanılan kipi belirtir. Bu ad, CPI (Common Programming Interface; Ortak Programlama Arabirimi)-İletişim Mode\_Adi ile aynıdır. Olası değerler şunlardır:

### \*NETATR

Ağ özniteliklerindeki kip kullanılır.

**blank (boşluk)**

Sekiz boş karakter kullanılır.

**Kip-adı**

Uzak konum için bir kip adı belirtin.

**Not:** Kip iletişim oturumunun iletim önceliğini belirlediğinden, gönderilmekte olan iletilerin önceliğine bağlı olarak farklı kipler tanımlamak yararlı olabilir; örneğin, MQMODE\_HI, MQMODE\_MED ve MQMODE\_LOW. (Aynı yeri gösteren birden fazla CSI ' ınız olabilir.)

**Uzak ağ tanıtıcısı**

Uzak konumla birlikte kullanılan uzak ağ tanıtıcısını belirler. Olası değerler şunlardır:

**\*LOC**

Uzak konum için uzak ağ tanıtıcısı kullanılır.

**\*NETATR**

Ağ özniteliklerinde belirlenen uzak ağ tanıtıcısı kullanılır.

**\*YOK**

Uzak ağın adı yok.

**Uzak ağ tanıtıcısı**

Bir uzak ağ tanıtıcısı belirleyin. Bu ağ tanıtıcısının adını bulmak için uzak yerdeki DSPNETA komutunu kullanın. Uzak konumdaki 'yerel ağ tanıtıcısı'.

**Authority**

Nesne üzerinde belirli bir yetkisi olmayan, yetki listesinde yer almayan ve nesne üzerinde belirli bir yetkisi olmayan kullanıcılar için verdiğiniz yetkiyi belirler. Olası değerler şunlardır:

**\*LIBCRTAUT**

Nesne için genel yetki, belirtilen kitaplığın CRTAUT parametresinden alınır. Bu değer, yaratma sırasında belirlenir. Nesne yaratıldıktan sonra kitaplığa ilişkin CRTAUT değeri değişirse, yeni değer var olan nesnelere etkilemez.

**\*DEĞİŞTİR**

Değiştirme yetkisi, kullanıcının nesne üzerinde temel işlevleri gerçekleştirmesini sağlar, ancak kullanıcı nesneyi değiştiremez. Değiştirme yetkisi, nesne işletim yetkisi ve tüm veri yetkisi sağlar.

**\*Tümü**

Kullanıcı, sahip ile sınırlı ya da yetki listesi yönetim yetkisi tarafından denetlenen işlemler dışında tüm işlemleri gerçekleştirebilir. Kullanıcı, nesnenin varlığını denetleyebilir ve nesnenin güvenliğini belirleyebilir, nesneyi değiştirebilir ve nesne üzerinde temel işlevleri gerçekleştirebilir. Kullanıcı nesnenin iyeliğini değiştirebilir.

**\*KULLANIM**

Kullanma yetkisi, nesne işletim yetkisi ve okuma yetkisi sağlar.

**\*EXCLUDE**

Dışlama yetkisi, kullanıcının nesneye erişmesini engeller.

**Yetki Listesi**

Yan bilgiler için kullanılan yetkiye sahip yetki listesinin adını belirleyin.

**IBM i**

*Başlatılan sona erdirme (Alıcı)*

\*LU62iletim tipiyle ileti kanalı bağlantısının alma sonunu tanımlamak için CRTMQMCHL komutunu kullanın.

CONNECTION NAME alanını boş bırakın ve ilgili ayrıntıların, kanalın gönderen ucuyla eşleştiğinden emin olun. Ayrıntılar için bkz. [Kanal oluşturma](#).

Başlangıç ucunun alıcı kanalı başlatmasını sağlamak için, başlatılan uçta bir altsisteme yöneltme girişi ekleyin. Altsistem, LU 6.2 oturumlarında kullanılan APPC aygıtını ayıran altsistem olmalıdır. Bu nedenle, bu aygıt için geçerli bir iletişim girişi olmalıdır. Yöneltme girişi, ileti kanalının alıcı ucunu başlatan programı çağırır.

Bir iletişim oturumu tarafından başlatılan bağlantının sonunu tanımlamak için IBM i komutlarını (örneğin, ADDRTGE) kullanın.

Başlatılan bitiş panosu LU 6.2 iletişim ayarı panosu-yönelme girişi ekleğinde gösterilir.

```
Add Routing Entry (ADDRTGE)

Type choices, press Enter.

Subsystem description . . . . . QCMN      Name
Library . . . . . *LIBL      Name, *LIBL, *CURLIB
Routing entry sequence number . 1      1-9999
Comparison data:
Compare value . . . . . MQSERIES

Starting position . . . . . 37      1-80
Program to call . . . . . AMQCRC6B    Name, *RTGDTA
Library . . . . . QMAS400      Name, *LIBL, *CURLIB
Class . . . . . *SBSD      Name, *SBSD
Library . . . . . *LIBL      Name, *LIBL, *CURLIB
Maximum active routing steps . . *NOMAX 0-1000, *NOMAX
Storage pool identifier . . . . . 1      1-10

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```

Şekil 35. LU 6.2 iletişim ayarı panosu tarafından başlatılan sona erdirmeye

#### Altsistem tanımlaması

Bu tanımlamanın bulunduğu altsisteminin adı. Yönelme girişine ilişkin uygun altsistem tanımlamasını görüntülemek ve güncellemek için IBM i WRKSBSD komutunu kullanın.

#### Yönelme girişi sıra numarası

Altsistemizde bu iletişim tanımını tanımlayacak benzersiz bir numara. 1-9999 aralığındaki değerleri kullanabilirsiniz.

#### Karşılaştırma verileri: Değeri karşılaştır

Oturum bir **Hareket programı** parametresi tarafından başlatıldığında alınan dizgiyle karşılaştırılacak metin dizgisi; bkz. [Şekil 1](#). Karakter dizgisi, gönderen CSI ' in İşlem programı alanından türetilir.

#### Karşılaştırma verileri: Başlangıç konumu

Karşılaştırmanın başlayacağı dizedeki karakter konumu.

**Not:** Başlangıç konumu alanı, karşılaştırma için dizedeki karakter konumudur ve bu konum her zaman 37 'dir.

#### Çağrılacak program

Oturumu başlatmak için çağrılacak gelen ileti programını çalıştıran programın adı.

Varsayılan kuyruk yöneticisi için AMQCRC6A programı çağrılır. Bu program IBM MQ for IBM i ile birlikte verilir ve ortamı ayarlar ve AMQCRS6A' yı çağırır.

Ek kuyruk yöneticileri için:

- Her kuyruk yöneticisinin, kitaplığında belirli bir LU 6.2 çağrılabilir programı bulunur. Bu program AMQCRC6B olarak adlandırılır ve kuyruk yöneticisi yaratıldığında otomatik olarak oluşturulur.
- Her kuyruk yöneticisi, benzersiz yönelme verileri eklenecek belirli bir yönelme girişi gerektirir. Bu yönelme verileri, istekte bulunan sistem tarafından sağlanan **Hareket programı** adıyla eşleşmelidir (bkz. [Başlatma sonu \(Gönderen\)](#)).

LU 6.2 iletişim ayarı panosu-görüntü yönelme girişleri için bir örnek gösterilmiştir:

```

Display Routing Entries
System: MY400
Subsystem description: QCMN      Status: ACTIVE

Type options, press Enter.
5=Display details

Start
Opt  Seq Nbr  Program      Library      Compare Value  Pos
10   *RTGDTA           'QZSCSRVR'    37
20   *RTGDTA           'QZRCSRVR'    37
30   *RTGDTA           'QZHQTRG'    37
50   *RTGDTA           'QVPPRINT'    37
60   *RTGDTA           'QNPSRVR'     37
70   *RTGDTA           'QNMAPPINGD'  37
80   QNMAREXECD  QSYS      'AREXECD'     37
90   AMQCR6A    QMQMBW    'MQSERIES'    37
100  *RTGDTA           'QTFDWNLD'   37
150  *RTGDTA           'QMFRVCVR'   37

F3=Exit  F9=Display all detailed descriptions  F12=Cancel

```

Şekil 36. LU 6.2 iletişim ayarı panosu tarafından başlatılan sona erdirme

LU 6.2 iletişim ayarı panosu-yönelme girişlerini görüntülerinde 90 sıra numarası, varsayılan kuyruk yöneticisini temsil eder ve önceki IBM MQ for IBM i yayın düzeylerindeki ( V3R2, V3R6, V3R7ve V4R2) yapılandırmalarla uyumluluk sağlar. Bu yayınlar yalnızca bir kuyruk yöneticisine izin verir. 92 ve 94 sıra numaraları, QMALPHA ve QMBETA kitaplıkları ile oluşturulan ALPHA ve BETA adlı iki ek kuyruk yöneticisini temsil eder.

**Not:** Farklı yönelme verileri kullanarak her kuyruk yöneticisi için birden çok yönelme girişiniz olabilir. Bu girişler, kullanılan sınıflara bağlı olarak farklı iş öncelikleri seçeneği sağlar.

### Sınıf

Bu yönelme girişinden başlatılan adımlar için kullanılan sınıfın adı ve kitaplığı. Sınıf, yönelme adımının çalışma ortamının özneliklerini tanımlar ve iş önceliğini belirtir. Uygun bir sınıf girişi belirtilmelidir. Örneğin, varolan sınıfları görüntülemek ya da bir sınıf yaratmak için WRKCLS komutunu kullanın. Uzak LU 6.2 sistemlerinden iş isteklerinin yönetilmesiyle ilgili ek bilgi için *IBM i Programming: Work Management Guide* belgesine bakın.

### İş Yönetimine İlişkin Not

AMQCR6A işi, diğer IBM MQ işleriyle aynı şekilde başlatılmadığından İş yönetimi içinde belgelenen olağan IBM i iş yönetimi özelliklerinden yararlanamaz. LU62 günlük nesnesi işlerinin yürütme zamanı özelliklerini değiştirmek için aşağıdaki değişikliklerden birini yapabilirsiniz:

- AMQCR6A işi için yönelme girişinde belirtilen sınıf tanımlamasını değiştirin
- İletişim girişindeki iş tanımlamasını değiştir

İletişim İşlerinin yapılandırılmasına ilişkin ek bilgi için *IBM i Programming: Work Management Guide* belgesine bakın.

## Kuyruk yöneticisi kümesinin yapılandırılması

Kümeler, hem ilk yapılandırmayı hem de devam eden yönetimi basitleştiren bir şekilde birbirine bağlanan kuyruk yöneticileri için bir mekanizma sağlar. Küme bileşenleri tanımlayabilir ve kümeleri oluşturabilir ve yönetebilirsiniz.

### Başlamadan önce

Kümeleme kavramlarına giriş için bkz. [Kümeler](#).

Kuyruk yöneticisi kümenizi tasarlarken bazı kararlar almanız gerekir. Bkz. [Örnek kümeler ve Kümeleri tasarlama](#).

### İlgili görevler

[“Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması” sayfa 429](#)

Yöneltilen ya da doğrudan yöneltilen konu kümelerinde, bir kuyruk yöneticisinin devreden çıkarılması sırasında ya da bir küme kuyruk yöneticisinin başarısız olması ya da önemli bir süre için uygun olmaması nedeniyle bir küme konusu tanımlamasını taşımanız gerekebilir.

### İlgili başvurular

[KONUUYU SIL](#)

## Kümenin bileşenlerini tanımlama

Kümeler, kuyruk yöneticileri, küme kanalları ve küme kuyruklarından oluşur. Küme kuyrukları tanımlayabilir ve varsayılan küme nesnelерinin bazı yönlerini değiştirebilirsiniz. Otomatik olarak tanımlanan kanallar ve tek tek küme gönderen kanalları ile iletim kuyrukları arasındaki ilişki hakkında yapılandırma ve durum bilgilerini alabilirsiniz.

Küme bileşenlerinin her birinin tanımlanmasına ilişkin bilgi için aşağıdaki alt başlıklara bakın:

### İlgili kavramlar

[Bir kümenin bileşenleri](#)

[Küme kanalları](#)

### İlgili görevler

[Küme konularının tanımlanması](#)

[“Yeni bir küme ayarlama” sayfa 297](#)

Örnek kümeyi ayarlamak için bu yönergeleri izleyin. Ayrı yönergeler, kümenin TCP/IP, LU 6.2 üzerinde ve tek bir iletim kuyruğuyla ya da birden çok iletim kuyruğuyla ayarlanmasını açıklar. Bir kuyruk yöneticisinden diğerine ileti göndererek kümeyi sınavın.

[“Küme kuyruk yöneticisi eklenmesi” sayfa 308](#)

Yarattığınız küme kuyruk yöneticisi eklemek için bu yönergeleri izleyin. Küme kuyruklarına ve konularına ilişkin iletiler, tek küme iletim kuyruğu SYSTEM . CLUSTER . TRANSMIT . QUEUE kullanılarak aktarılır.

## Küme kuyruklarının tanımlanması


Küme kuyruğu, bir küme kuyruk yöneticisi tarafından barındırılan ve kümedeki diğer kuyruk yöneticilerinin kullanımına sunulan bir kuyruktur. Bir küme kuyruğunu, kuyruğun barındırıldığı küme kuyruğu yöneticisinde yerel kuyruk olarak tanımlayın. Kuyruğun ait olduğu kümenin adını belirtin.

Aşağıdaki örnekte, CLUSTER seçeneğiyle bir küme kuyruğu tanımlamak için bir **runmqsc** komutu gösterilmektedir:

```
DEFINE QLOCAL(Q1) CLUSTER(SALES)
```

Kümedeki diğer kuyruk yöneticilerine bir küme kuyruğu tanımlaması tanıtılır. Kümedeki diğer kuyruk yöneticileri, iletileri karşılık gelen uzak kuyruk tanımlamasına gerek kalmadan bir küme kuyruğuna yerleştirebilirler. Küme kuyruğu, küme adı kullanılarak birden çok kümede duyurulabilir.

Bir kuyruk duyurulduğunda, kümedeki herhangi bir kuyruk yöneticisi bu kuyruğa ileti yerleştirebilir. Bir ileti koymak için kuyruk yöneticisinin, kuyruğun bulunduğu tam havuzlardan bunu öğrenmesi gerekir. Daha sonra iletiye bazı yöneltme bilgileri ekler ve iletiyi bir küme iletim kuyruğuna yerleştirir.

 Küme kuyruğu, IBM MQ for z/OS' daki bir kuyruk paylaşım grubunun üyeleri tarafından paylaşılan bir kuyruk olabilir.



## Bağ Tanımı

Birden çok kuyruk yöneticisinin aynı küme kuyruğunun eşgörünümünü bulundurduğu bir küme yaratabilirsiniz. Bir sıradaki tüm iletilerin kuyruğun aynı eşgörünümüne gönderildiğinden emin olun. MQOPEN çağrısında MQOO\_BIND\_ON\_OPEN seçeneğini kullanarak bir dizi iletiyi belirli bir kuyruğa bağlayabilirsiniz.


## Küme iletim kuyrukları

Kuyruk yöneticisi, birden çok iletim kuyruğundaki bir kümedeki diğer kuyruk yöneticilerine ilişkin iletileri saklayabilir. Bir kuyruk yöneticisini, iletileri birden çok küme iletim kuyruğunda iki farklı şekilde saklaması için yapılandırabilirsiniz. **DEFCLXQ** kuyruk yöneticisi özniteliğini CHANNEL olarak ayarlarsanız, her küme gönderen kanalı için SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE ' den otomatik olarak farklı bir küme iletim kuyruğu yaratılır. CLCHNAME iletim kuyruğu seçeneğini bir ya da daha çok küme gönderen kanalla eşleşecek şekilde ayarlarsanız, kuyruk yöneticisi, o iletim kuyruğundaki eşleşen kanallara ilişkin iletileri saklayabilir.



**Uyarı:** IBM WebSphere MQ 7.5'un önceki bir sürümünden yükseltilmiş bir kuyruk yöneticisiyle özel olarak ayrılmış SYSTEM . CLUSTER . TRANSMIT . QUEUES kullanıyorsanız, SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE ' de SHARE/NOSHARE seçeneğinin **SHARE** olarak ayarlandığından emin olun.

Gönderilmeden önce, farklı bir kuyruk yöneticisindeki küme kuyruğuna ilişkin bir ileti küme iletim kuyruğuna yerleştirilir. Küme gönderen kanal, iletileri bir küme iletim kuyruğundan diğer kuyruk yöneticilerindeki küme alıcı kanallarına aktarır. Varsayılan olarak, bir sistem tanımlı küme iletim kuyruğu, diğer küme kuyruğu yöneticilerine aktarılacak tüm iletileri tutar. Kuyruğa SYSTEM . CLUSTER . TRANSMIT . QUEUE adı verilir. Bir kümenin parçası olan bir kuyruk yöneticisi, bu küme iletim kuyruğundaki iletileri aynı kümedeki başka bir kuyruk yöneticisine gönderebilir.

z/OS'dışında her kuyruk yöneticisinde varsayılan olarak tek bir SYSTEM . CLUSTER . TRANSMIT . QUEUE kuyruğu için bir tanımlama yaratılır.  z/OS üzerinde, tanımlama sağlanan örnekle tanımlanabilir **CSQ4INSX**.

Bir kuyruk yöneticisini, birden çok iletim kuyruğunu kullanarak iletileri diğer kümelenmiş kuyruk yöneticilerine aktarabilecek şekilde yapılandırabilirsiniz. Ek küme iletim kuyruklarını el ile tanımlayabilir ya da kuyruk yöneticisinin kuyrukları otomatik olarak yaratmasını sağlayabilirsiniz.

Kuyruk yöneticisi tarafından otomatik olarak yaratılan kuyrukların olması için, DEFCLXQ kuyruk yöneticisi özniteliğini SCTQ 'dan CHANNEL ' a çevirin. Sonuç olarak, kuyruk yöneticisi yaratılan her küme gönderen kanalı için tek bir küme iletim kuyruğu yaratır. İletim kuyrukları, model kuyruğundan ( SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE ) kalıcı dinamik kuyruklar olarak yaratılır. Her bir kalıcı dinamik kuyruğun adı şudur: SYSTEM . CLUSTER . TRANSMIT . *ChannelName*. Her bir kalıcı dinamik küme iletim kuyruğunun ilişkilendirildiği küme gönderen kanalının adı, CLCHNAME yerel iletim kuyruğu özniteliğinde belirlenir. Uzak kümelenmiş kuyruk yöneticilerine ilişkin iletiler, SYSTEM . CLUSTER . TRANSMIT . QUEUE yerine, ilişkili küme gönderen kanalı için kalıcı dinamik küme iletim kuyruğuna yerleştirilir.

Küme iletim kuyruklarını el ile oluşturmak için, USAGE özniteliği XMITQ olarak ayarlanmış bir yerel kuyruk ve CLCHNAME özniteliği bir ya da daha çok küme gönderen kanalına çözülen sosyal bir kanal adına ayarlanmış bir yerel kuyruk oluşturun; bkz. *ClusterChannelName*. Küme iletim kuyruklarını el ile yaratırsanız, iletim kuyruğunu tek bir küme gönderen kanalla ya da birden çok küme gönderen kanalla ilişkilendirmeyi seçebilirsiniz. CLCHNAME özniteliği sosyal bir addır; bu, ada birden çok genel arama karakteri ( "\*" ) yerleştirebileceğiniz anlamına gelir.

Bir kuyruk yöneticisini tam bir havuza bağlamak için el ile oluşturduğunuz ilk küme gönderen kanalları dışında, küme gönderen kanallar otomatik olarak oluşturulur. Bunlar, bir küme kuyruk yöneticisine aktarılacak bir ileti olduğunda otomatik olarak yaratılır. Bunlar, hedef kuyruk yöneticisindeki belirli bir kümeye ilişkin küme iletilerini alan küme alıcı kanalının adıyla aynı adla yaratılır.

Küme alıcı kanalları için bir adlandırma kuralını izlerseniz, farklı tipte küme iletilerini farklı iletim kuyruklarına süzen CLCHNAME için sosyal bir değer tanımlayabilirsiniz. Örneğin, *ClusterName* .

*QmgrName*' un küme alıcı kanallarına ilişkin adlandırma kuralını izlerseniz, soysal ad *ClusterName* . \* farklı kümelere ilişkin iletileri farklı iletim kuyruklarına süzer. İletim kuyruklarını el ile tanımlamalı ve her iletim kuyruğunda CLCHNAME değerini *ClusterName* . \*olarak belirlemelisiniz.

Küme iletim kuyruklarının küme gönderen kanallarla ilişkilendirilmesinde yapılan değişiklikler hemen yürürlüğe girmez. Küme gönderen kanalının hizmet vermekte olduğu ilişkili iletim kuyruğu, küme gönderen kanalı tarafından aktarılmakta olan iletileri içerebilir. Yalnızca, ilişkili iletim kuyruğundaki hiçbir ileti bir küme-gönderen kanalı tarafından işlenmediği zaman, kuyruk yöneticisi kümeyi gönderen kanalın ilişkilendirmesini farklı bir iletim kuyruğu olarak değiştirebilir. Bu durum, küme gönderen kanalı tarafından işlenecek iletim kuyruğunda hiçbir ileti kalmadığında ya da iletilerin işlenmesi askıya alındığında ve küme gönderen kanalda "hareket halinde" ileti olmadığında ortaya çıkabilir. Bu durumda, küme gönderen kanalına ilişkin işlenmemiş iletiler yeni ilişkili iletim kuyruğuna aktarılır ve küme gönderen kanalının ilişkilendirmesi değişir.

Küme iletim kuyruğuna çözülen bir uzak kuyruk tanımı yaratabilirsiniz. Tanımda, QMX kuyruk yöneticisi yerel kuyruk yöneticisiyle aynı kümededir ve QMXiletim kuyruğu yoktur.

```
DEFINE QREMOTE(A) RNAME(B) RQMNAME(QMX)
```

Kuyruk adı çözülürken, küme iletim kuyruğu varsayılan iletim kuyruğundan önceliklidir. A ' e yerleştirilen bir ileti, küme iletim kuyruğunda saklanır ve daha sonra, QMXüzerindeki B uzak kuyruğuna gönderilir.

Kuyruk yöneticileri, bir kümenin parçası olmayan diğer kuyruk yöneticileriyle de iletişim kurabilir. Kanalları ve iletim kuyruğunu diğer kuyruk yöneticisine, dağıtılmış kuyruğa alma ortamındaki gibi tanımlamanız gerekir.

**Not:** Uygulamalar, küme iletim kuyruğuna çözülecek kuyruklara yazmalı ve doğrudan küme iletim kuyruğuna yazmamalıdır.

## Uzak kuyrukların otomatik tanımlaması

Kümedeki bir kuyruk yöneticisi, kümedeki uzak kuyruklar için uzak kuyruk tanımlamasına gerek duymaz. Küme kuyruğu yöneticisi, uzak bir kuyruğun yerini tam havuzdan bulur. İletiyeye yöneltme bilgileri ekler ve iletiyi küme iletim kuyruğuna yerleştirir. IBM MQ , iletinin gönderilebilmesi için otomatik olarak uzak kuyruk tanımlamasına eşdeğer bir tanımlama yaratır.

Otomatik olarak yaratılan bir uzak kuyruk tanımlamasını değiştiremez ya da silemezsiniz. Ancak, DISPLAY QUEUE **runmqsc** komutunu CLUSINFO özniteliğiyle kullanarak, uzak kuyruk yöneticilerindeki küme kuyrukları da içinde olmak üzere, bir kuyruk yöneticisindeki tüm yerel kuyrukları ve tüm küme kuyruklarını görüntüleyebilirsiniz. Örneğin:

```
DISPLAY QUEUE(*) CLUSINFO
```

### İlgili kavramlar

[Küme kuyrukları](#)

[Hangi küme iletim kuyruğunun kullanılacağını seçme](#)

### İlgili başvurular

[ClusterChannelAdı \(MQCHAR20\)](#)

## Otomatik tanımlı küme gönderen kanallarıyla çalışma

Bir kuyruk yöneticisini ilk CLUSSDR ve CLUSRCVR tanımlamalarını yaparak bir kümeye tanıttıktan sonra, IBM MQ iletileri kümedeki başka bir kuyruk yöneticisine taşımak için gerektiğinde otomatik olarak diğer küme gönderen kanal tanımlarını oluşturur. Otomatik olarak tanımlanan küme gönderen kanalları hakkındaki bilgileri görüntüleyebilirsiniz, ancak bunları değiştiremezsiniz. Davranışlarını değiştirmek için bir kanal otomatik tanımlama çıkışı kullanabilirsiniz.

## Başlamadan önce

Otomatik tanımlı kanallara giriş için bkz. [Otomatik tanımlı küme gönderen kanalları](#).

## Bu görev hakkında

Otomatik olarak tanımlanan küme gönderen kanalları, gerektiğinde ve gerektiği şekilde küme tarafından oluşturulur ve normal bağlantı kesme aralığı kuralları kullanılarak kapatılıncaya kadar etkin kalır.

Küme gönderen kanalları (CLUSSDR) hem uygulama iletilerini hem de iç küme yönetimi iletilerini taşımak için otomatik olarak tanımlanabilir. Örneğin, bir Yayınla/abone ol kümesinde (kümeli bir konunun tanımlandığı), 'yetkili sunucu aboneliği' durumuna izin vermek için kısmi havuzlar arasında kanallar tanımlanabilir. Uzun bir süre için gerekli olmadığına (etkin olmadığına) otomatik olarak tanımlanan CLUSSDR 'ler kısmi bir havuzun küme bilgileri önbelleğinden kaldırılır ve artık o kuyruk yöneticisinde görünmez.

**Multi** Çoklu platformlar üzerinde, OAM (nesne yetkisi yöneticisi) otomatik olarak tanımlanan küme gönderen kanallarının varlığından haberdar değildir. Otomatik olarak tanımlanan bir küme gönderen kanalda **start**, **stop**, **ping**, **reset** ya da **resolve** komutları yayınlarsanız, OAM, eşleşen küme alıcı kanalında aynı işlemi gerçekleştirme yetkiniz olup olmadığını denetler.

**z/OS** z/OS işletim sistemi üzerinde, otomatik olarak tanımlanan bir küme gönderen kanalının güvenliğini diğer kanallarla aynı şekilde sağlayabilirsiniz.

## Yordam

- Belirli bir küme kuyruk yöneticisine ilişkin otomatik tanımlı kanallara ilişkin bilgileri görüntüler.

DISPLAY CHANNEL **runmqsc** komutunu kullanarak otomatik olarak tanımlanan kanalları göremezsiniz. Otomatik tanımlı kanalları görmek için aşağıdaki komutu kullanın:

```
DISPLAY CLUSQMGR(qMgrName)
```

- Belirli bir CLUSRCVR için otomatik tanımlı kanalın durumunu görüntüler.

Yarattığınız CLUSRCVR kanal tanımlamasına karşılık gelen otomatik tanımlı CLUSSDR kanalının durumunu görüntülemek için aşağıdaki komutu kullanın:

```
DISPLAY CHSTATUS(channelName)
```

- Otomatik tanımlı bir kanalın davranışını değiştirmek için kanal otomatik tanımlama çıkışı kullanın.

Bir küme gönderen kanalı ya da küme alıcı kanalını uyarlamak için bir kullanıcı çıkış programı yazmak istiyorsanız, IBM MQ kanal otomatik tanımlama çıkışı kullanabilirsiniz. Örneğin, aşağıdaki değişikliklerden herhangi birini yapmak için bir küme ortamında kanal otomatik tanımlama çıkışı kullanabilirsiniz:

- SNA LU6.2 adlarını uyarlamak için iletişim tanımlamalarını kullanın.
- Güvenlik çıkışları gibi diğer çıkışları ekleyin ya da kaldırın.
- Kanal çıkışlarının adlarını değiştirin.

CLUSSDR kanal çıkışının adı, CLUSRCVR kanal tanımından otomatik olarak oluşturulur ve bu nedenle, özellikle de kanalın iki ucu farklı platformlardaysa, gereksinimleriniz için uygun olmayabilir.

Çıkış adlarının biçimi farklı platformlarda farklı. Örneğin:

- **z/OS** z/OS platformunda, SCYEXIT (*güvenlik çıkışı adı*) parametresinin biçimi şöyledir: SCYEXIT('SECEXIT')
- **Windows** Windows altyapılarında, SCYEXIT (*güvenlik çıkışı adı*) değiştirgesinin biçimi şöyledir: SCYEXIT('drive:\path\library (secexit)')

**Not:** **z/OS** Kanal otomatik tanımlama çıkışı yoksa, z/OS kuyruk yöneticisi, kanalın diğer ucundaki CLUSRCVR kanal tanımından CLUSSDR kanal çıkış adını türetir. z/OS çıkış adınız/OS olmayan bir addan türetmek için aşağıdaki algoritma kullanılır:

- Çoklu platformlar üzerindeki çıkış adları *path/library (işlev)* genel biçimindedir.
- *İşlev* varsa, bunlardan en çok sekiz karakter kullanılır.
- Ters durumda, en çok sekiz karakter *kitaplık* kullanılır.

Örneğin:

- /var/mqm/exits/myExit.so(MsgExit) şuna dönüştürür MSGEXIT
- /var/mqm/exits/myExit şuna dönüştürür MYEXIT
- /var/mqm/exits/myExit.so(ExitLongName) şuna dönüştürür EXITLONG
- Kümenin, IBM MQ kuyruk yöneticisinden ürünün önceki bir sürümünde kuyruk yöneticisine giden iletilerden RFH2 gibi uygulama üstbilgilerini kaldırmak için **PROPCTL** komutunu kullanması gerekiyorsa, **PROPCTL** için NONEdeğerini ayarlayan bir kanal otomatik tanımlama çıkışı yazmanız gerekir.
- Adreslemenin özelliklerini denetlemek için LOCLADDR kanal özniteliğini kullanın.
  - Giden (TCP) kanalının belirli bir IP adresi, kapı ya da kapı aralığını kullanmasını sağlamak için LOCLADDRkanal özniteliğini kullanın. Bu, birden fazla ağ kartınız varsa ve bir kanalın giden iletişim için belirli bir kanal kullanmasını istiyorsanız kullanışlıdır.
  - CLUSSDR kanallarında bir sanal IP adresi belirtmek için el ile tanımlanan CLUSSDRüzerindeki LOCLADDR konumundan IP adresini kullanın. Kapı aralığını belirtmek için CLUSRCVR' deki kapı aralığını kullanın.
  - Bir kümenin belirli bir IP adresine bağlanacak giden iletişim kanallarını almak için LOCLADDR kullanması gerekiyorsa, LOCLADDR değerini otomatik olarak tanımlanan CLUSSDR kanallarından herhangi birine zorlamak için bir kanal otomatik tanımlama çıkışı yazabilirsiniz. El ile tanımlanan CLUSSDR kanalında da belirtmeniz gerekir.
  - Bir kümedeki tüm kuyruk yöneticilerinin tüm giden iletişimleri için belirli bir kapı ya da kapı aralığı kullanmasını istiyorsanız, CLUSRCVR kanalının LOCLADDR alanına bir kapı numarası ya da kapı aralığı girin.

**Not:** Tüm kuyruk yöneticileri aynı sunucuda değilse, CLUSRCVR kanalının LOCLADDR alanına bir IP adresi koymayın. LOCLADDR IP adresi, CLUSRCVR kanalı kullanılarak bağlanan tüm kuyruk yöneticilerinin otomatik olarak tanımlanan CLUSSDR kanallarına yayılır.

**Multi** Çoklu platformlar üzerinde, yerel adresi tanımlanmamış tüm gönderen kanalları için kullanılan varsayılan bir yerel adres değeri ayarlayabilirsiniz. Varsayılan değer, kuyruk yöneticisini başlatmadan önce MQ\_LCLADDR ortam değişkeni ayarlanarak tanımlanır. Değerin biçimi, LOCLADDRMQSC özniteliğiyle eşleşiyor.

## İlgili başvurular

[Yerel Adres \(LOCLADDR\)](#)

## Varsayılan küme nesneleriyle çalışma

MQSC ya da PCF komutlarını çalıştırarak, varsayılan kanal tanımlamalarını diğer kanal tanımlamalarıyla aynı şekilde değiştirebilirsiniz. SYSTEM . CLUSTER . HISTORY . QUEUE dışında, varsayılan kuyruk tanımlamalarını değiştirmeyin.

Bu nesnelere tam listesi için bkz. [Varsayılan küme nesneleri](#). Aşağıdaki liste yalnızca değiştirebileceğiniz nesnelere içerir.


### SYSTEM . CLUSTER . HISTORY . QUEUE

Bir kümedeki her kuyruk yöneticisinin SYSTEM . CLUSTER . HISTORY . QUEUE adlı yerel bir kuyruğu vardır. SYSTEM . CLUSTER . HISTORY . QUEUE , hizmet amacıyla küme durumu bilgilerinin geçmişi saklamak için kullanılır.

Varsayılan nesne ayarlarında SYSTEM . CLUSTER . HISTORY . QUEUE , PUT ( ENABLED) olarak ayarlanır. Geçmiş derlemeyi engellemek için ayarı PUT ( DISABLED) olarak değiştirin.

## SYSTEM . CLUSTER . TRANSMIT . QUEUE

Her kuyruk yöneticisinin SYSTEM . CLUSTER . TRANSMIT . QUEUE adlı yerel bir kuyruk için bir tanımlaması vardır. SYSTEM . CLUSTER . TRANSMIT . QUEUE , kümeler içindeki tüm kuyruklara ve kuyruk yöneticilerine gönderilen tüm iletilere ilişkin varsayılan iletim kuyruğudur. DEFXMLTQ

 z/OS dışında kuyruk yöneticisi özniteliğini değiştirerek, her küme gönderen kanalı için varsayılan iletim kuyruğunu SYSTEM . CLUSTER . TRANSMIT . ChannelName olarak değiştirebilirsiniz. SYSTEM . CLUSTER . TRANSMIT . QUEUE ögesini silemezsiniz. Kullanılan varsayılan iletim kuyruğunun SYSTEM . CLUSTER . TRANSMIT . QUEUE ya da SYSTEM . CLUSTER . TRANSMIT . ChannelName olup olmadığını belirlemek için de kullanılır.

### İlgili kavramlar

[Varsayılan küme nesnelere](#)

### **Küme iletim kuyrukları ve küme gönderen kanallarıyla çalışma**

Kümelenmiş kuyruk yöneticileri arasındaki iletiler küme iletim kuyruklarında saklanır ve küme gönderen kanallar tarafından iletilir. Herhangi bir zamanda, küme gönderen kanal bir iletim kuyruğuyla ilişkilendirilir. Kanalın konfigürasyonunu değiştirirseniz, kanal bir sonraki başlatılışında farklı bir iletim kuyruğuna geçebilir. Bu anahtarın işlenmesi otomatiktir ve işlemseldir.

Küme gönderen kanallarının ilişkilendirildiği iletim kuyruklarını görüntülemek için aşağıdaki MQSC komutunu çalıştırın:

```
DISPLAY CHSTATUS(*) WHERE(CHLTYPE EQ CLUSSDR)
```

```
AMQ8417: Display Channel Status details.  
CHANNEL (TO.QM2)          CHLTYPE (CLUSSDR)  
CONNNAME (9.146.163.190(1416))  CURRENT  
RQMNAME (QM2)            STATUS (STOPPED)  
SUBSTATE ( )             XMITQ (SYSTEM . CLUSTER . TRANSMIT . QUEUE)
```

Durdurulan bir küme gönderen kanalının kayıtlı kanal durumunda gösterilen iletim kuyruğu, kanal yeniden başlatıldığında değişebilir. [“Kümeğe göre varsayılan iletim kuyruklarının seçilmesi-gönderen kanalları” sayfa 290](#) , varsayılan bir iletim kuyruğunun seçilme işlemini açıklar; [“Küme-gönderen kanallarına göre el ile tanımlanan iletim kuyruklarının seçilmesi” sayfa 290](#) el ile tanımlanan bir iletim kuyruğunun seçilme işlemini açıklar.

Herhangi bir küme gönderen kanal başlatıldığında, iletim kuyruklarıyla ilişkisini yeniden denetler. İletim kuyruklarının konfigürasyonu ya da kuyruk yöneticisi varsayılan değerleri değişirse, kanalı farklı bir iletim kuyruğuyla yeniden ilişkilendirebilir. Bir konfigürasyon değişikliğinin sonucu olarak kanal farklı bir iletim kuyruğuyla yeniden başlarsa, iletileri yeni ilişkili iletim kuyruğuna aktarma işlemi gerçekleşir. [“Kümeği gönderen kanalı farklı bir iletim kuyruğuna aktarma işlemi nasıl çalışır?” sayfa 291](#) içinde, bir küme gönderen kanalının bir iletim kuyruğundan diğerine aktarılması açıklanmaktadır.

Küme-gönderen kanallarının davranışı, gönderen ve sunucu kanallarından farklıdır. **XMITQ** kanal özniteliği değiştirilinceye kadar aynı iletim kuyruğuyla ilişkili kalırlar. Bir gönderen ya da sunucu kanalında iletim kuyruğu özniteliğini değiştirir ve yeniden başlatırsanız, iletiler eski iletim kuyruğundan yeni iletime aktarılmaz.

Küme gönderen kanalları ile gönderen ya da sunucu kanalları arasındaki diğer bir fark, birden çok küme gönderen kanalının bir küme iletim kuyruğunu açabilmesidir, ancak yalnızca bir gönderen ya da sunucu kanalı normal bir iletim kuyruğunu açabilir. İletim kuyruklarını paylaşmayan küme-gönderen kanalları seçeneğiniz vardır. Ayrıcalık uygulanmaz; bu, yapılandırmanın bir sonucudur. Bir iletinin bir kümede izlediği yolu, diğer uygulamalar arasında akan iletilerle herhangi bir iletim kuyruklarını ya da kanallarını paylaşmayacak şekilde yapılandırabilirsiniz. Bkz. Kümeleme: Küme iletim kuyruklarının nasıl yapılandırılacağını planlama ve [“Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme ve küme iletim kuyruğu eklenmesi” sayfa 340](#).

## Kümeye göre varsayılan iletim kuyruklarının seçilmesi-gönderen kanalları

Küme iletim kuyruğu, adı SYSTEM . CLUSTER . TRANSMIT ile başlayan bir sistem varsayılan kuyruğu ya da el ile tanımlanan bir kuyruktur. Küme gönderen kanal, küme iletim kuyruğuyla şu iki yoldan biriyle ilişkilendirilir: Varsayılan küme iletim kuyruğu mekanizması ya da el ile konfigürasyon.

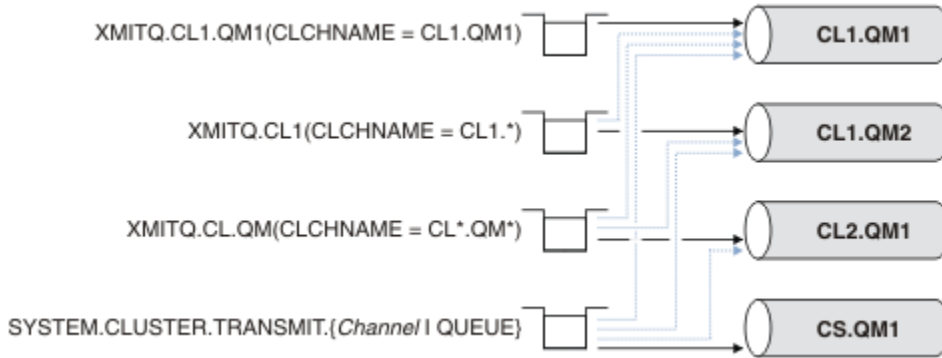
Varsayılan küme iletim kuyruğu, **DEFCLXQ** kuyruk yöneticisi özniteliği olarak belirlenir. Değeri SCTQ ya da CHANNEL' dır. Yeni ve geçirilen kuyruk yöneticileri SCTQ olarak ayarlanır. Değeri KANAL olarak değiştirebilirsiniz.

SCTQ değeri belirlenirse, varsayılan küme iletim kuyruğu SYSTEM . CLUSTER . TRANSMIT . QUEUE olur. Her küme gönderen kanal bu kuyruğu açabilir. Kuyruğu açan küme gönderen kanallar, el ile tanımlanan küme iletim kuyruklarıyla ilişkilendirilmemiş kanallardır.

CHANNEL ayarlanırsa, kuyruk yöneticisi her küme gönderen kanal için ayrı bir kalıcı dinamik iletim kuyruğu yaratabilir. Her kuyruk SYSTEM . CLUSTER . TRANSMIT . ChannelName olarak adlandırılır ve model kuyruğundan ( SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE) yaratılır. El ile tanımlanan küme iletim kuyruğuyla ilişkilendirilmemiş her küme gönderen kanal, kalıcı dinamik bir küme iletim kuyruğuyla ilişkilendirilir. Kuyruk, bu küme gönderen kanalının hizmet verdiği küme hedefi için ayrı bir küme iletim kuyruğu gerektirdiğinde kuyruk yöneticisi tarafından yaratılır ve kuyruk yoktur.

Bazı küme hedeflerine, el ile tanımlanan iletim kuyruklarıyla ilişkili küme-gönderen kanalları, diğerlerine ise varsayılan kuyruk ya da kuyruklar tarafından hizmet verilebilir. Küme gönderen kanalların iletim kuyruklarıyla ilişkilendirilmesinde, el ile tanımlanan iletim kuyrukları her zaman varsayılan iletim kuyruklarından önceliklidir.

Küme iletim kuyruklarının önceliği Şekil 37 sayfa 290 içinde gösterilmektedir. El ile tanımlanan küme iletim kuyruğuyla ilişkilendirilmemiş tek küme gönderen kanal CS . QM1. İletim kuyruklarının **CLCHNAME** özniteliğindeki kanal adlarının hiçbiri CS . QM1 ile eşleşmediğinden, el ile tanımlanan iletim kuyruğuyla ilişkilendirilmez.



Şekil 37. İletim kuyruğu/küme-gönderen kanalı önceliği

## Küme-gönderen kanallarına göre el ile tanımlanan iletim kuyruklarının seçilmesi

El ile tanımlanan bir kuyrukte, iletim kuyruğu özniteliği **USAGE** özniteliği XMITQ olarak ayarlanmış ve küme kanalı adı özniteliği **CLCHNAME** belirli ya da sosyal bir kanal adına ayarlanmış.

**CLCHNAME** kuyruk özniteliğindeki ad bir küme gönderen kanal adıyla eşleşirse, kanal kuyrukla ilişkilendirilir. Ad, genel arama karakteri içermiyorsa, tam eşleşme ya da genel arama karakteri içeriyorsa en iyi eşleşme olur.

Birden çok iletim kuyruğundaki **CLCHNAME** tanımları aynı küme gönderen kanalla eşleşiyorsa, tanımların örtüştüğü söyilir. Belirsizliği çözmek için eşleşmeler arasında bir öncelik sırası vardır. Kesin eşleşmeler her zaman önceliklidir. Şekil 37 sayfa 290 , iletim kuyrukları ile küme gönderen kanalları arasındaki ilişkileri gösterir. Siyah oklar gerçek ilişkileri ve gri okları, olası ilişkileri gösterir. Şekil 37 sayfa 290 içindeki iletim kuyruklarının öncelik sırası,

### **XMITQ.CL1.QM1**

XMITQ.CL1.QM1 iletim kuyruğunun **CLCHNAME** özniteliği CL1.QM1olarak ayarlı. **CLCHNAME** özniteliğinin tanımlamasında (CL1.QM1) joker karakter yoktur ve diğer iletim kuyruklarında tanımlanan ve joker karakterlerle eşleşen diğer CLCHNAME özniteliklerinden önceliklidir. Kuyruk yöneticisi, XMITQ.CL1.QM1 iletim kuyruğundaki CL1.QM1 küme gönderen kanalı tarafından aktarılan küme iletilerini saklar. Tek kural dışı durum, birden çok iletim kuyruğunun **CLCHNAME** özniteliği CL1.QM1olarak ayarlandığında ortaya çıktı. Bu durumda, kuyruk yöneticisi CL1.QM1 küme gönderen kanalına ilişkin iletileri bu kuyruklardan herhangi birinde saklar. Kanal başladığında rastgele bir kuyruk seçer. Kanal yeniden başlatıldığında farklı bir kuyruk seçebilir.

### **XMITQ.CL1**

XMITQ.CL1 iletim kuyruğunun **CLCHNAME** özniteliği CL1.\*olarak ayarlı. **CLCHNAME** özniteliğinin tanımlaması (CL1.\*), CL1. ile başlayan herhangi bir küme gönderen kanalının adıyla eşleşen bir sondaki genel arama karakterine sahiptir. The queue manager stores any cluster message that is to be transferred by any cluster-sender channel whose name begins with CL1. on the transmission queue XMITQ.CL1, unless there is a transmission queue with a more specific match, such as the queue XMITQ.CL1.QM1. Bir sondaki genel arama karakteri, tanımı, genel arama karakteri içermeyen bir tanımdan daha az, birden çok genel arama karakteri içeren bir tanımdan daha özel ya da onu izleyen daha çok genel arama karakteri içeren bir tanımdan daha özel yapar.

### **XMITQ.CL.QM**

XMITQ.CL.QM, **CLCHNAME** özniteliği CL\*.QM\*olarak ayarlanmış iletim kuyruğunun adıdır. CL\*.QM\* tanımlamasında, CL. ile başlayan ve QMile biten herhangi bir küme gönderen kanalının adıyla eşleşen iki genel arama karakteri vardır. Eşleşme, bir genel arama karakteriyle eşleşmekten daha az özeldir.

### **SYSTEM.CLUSTER.TRANSMIT. channelName | QUEUE**

Kuyruk yöneticisinin kullanacağı kümeyi gönderen kanalın adıyla eşleşen bir iletim kuyruğunun **CLCHNAME** özniteliği yoksa, kuyruk yöneticisi varsayılan küme iletim kuyruğunu kullanır. Varsayılan küme iletim kuyruğu, tek sistem kümesi iletim kuyruğu SYSTEM.CLUSTER.TRANSMIT.QUEUEya da kuyruk yöneticisinin belirli bir küme gönderen kanal (SYSTEM.CLUSTER.TRANSMIT.channelName) için yarattığı bir sistem kümesi iletim kuyruğudur. Varsayılan kuyruk olan kuyruk, kuyruk yöneticisi **DEFXMITQ** özniteliğinin ayarına bağlıdır.

**İpucu:** Çakışan tanımlamalara açık bir gereksiniminiz yoksa, bunların anlaşılması zor karmaşık yapılandırmalara yol açabildiği için bunlardan kaçınınız.

## **Kümeyi gönderen kanalını farklı bir iletim kuyruğuna aktarma işlemi nasıl çalışır?**

Küme gönderen kanallarının ilişkilendirmesini küme iletim kuyruklarıyla değiştirmek için, istediğiniz zaman **CLCHNAME** herhangi bir iletim kuyruğunun ya da kuyruk yöneticisi parametresinin değerini **DEFCLXQ** değiştirin. Hemen bir şey olmaz. Değişiklikler yalnızca bir kanal başlatıldığında oluşur. Başladığında, iletilerin aynı iletim kuyruğundan iletmeye devam edilip edilmeyeceğini denetler. Üç tür değişiklik, bir küme-gönderen kanalının bir iletim kuyruğuyla ilişkisini değiştirir.

1. Küme gönderen kanalının şu anda ilişkilendirildiği iletim kuyruğunun **CLCHNAME** parametresinin yeniden tanımlanması daha az belirli ya da boş olacak şekilde ya da kanal durdurulduğunda küme iletim kuyruğunun silinmesi.

Başka bir küme iletim kuyruğu, kanal adıyla daha iyi eşleşebilir. Ya da, küme-gönderen kanalının adıyla eşleşen başka iletim kuyruğu yoksa, ilişkilendirmenin varsayılan iletim kuyruğuna geri dönmesi gerekir.

2. Başka bir küme iletim kuyruğunun **CLCHNAME** parametresi yeniden tanımlıyor ya da bir küme iletim kuyruğu ekleniyor.

Başka bir iletim kuyruğunun **CLCHNAME** parametresi, küme-gönderen kanalı için, küme gönderen kanalının ilişkili olduğu iletim kuyruğundan daha iyi eşleşme olabilir. Kümeyi gönderen kanal varsayılan bir küme iletim kuyruğuyla ilişkilendirilmişse, el ile tanımlanan bir küme iletim kuyruğuyla ilişkilendirilebilir.

3. Kümeyi gönderen kanal varsayılan bir küme iletim kuyruğuyla ilişkilendirilmişse, **DEFCLXQ** kuyruk yöneticisi değiştirilmesini değiştirin.



Bir küme-gönderen kanalının ilişkilendirmesi değişirse, kanal başlatıldığında ilişkilendirmesini yeni iletim kuyruğuna değiştirir. Anahtar sırasında, iletilerin kaybolmamasını sağlar. İletiler, yeni iletim kuyruğuna, kanalın iletileri uzak kuyruk yöneticisine aktaracağı sırayla aktarılır.

**Unutmayın:** Bir kümedeki iletilerin iletilmesiyle ortak olarak, sırayla teslim edilmesi gereken iletilerin teslim edildiğinden emin olmak için iletileri gruplara yerleştirmeniz gerekir. Nadir durumlarda, mesajlar bir kümede sıradışı olabilir.

Geçiş işlemi aşağıdaki işlemsel adımlardan geçer. Geçiş işlemi kesilirse, kanal yeniden başlatıldığında geçerli işlem adımı sürdürülür.

#### **Adım 1-Özgün iletim kuyruğundaki iletileri işle**

Küme gönderen kanal, diğer küme gönderen kanallarıyla paylaşabileceği yeni iletim kuyruğuyla ilişkilendirilir. Kümeyi gönderen kanala ilişkin iletiler özgün iletim kuyruğuna yerleştirilmeye devam eder. Bir geçiş anahtarı işlemi, iletileri özgün iletim kuyruğundan yeni iletim kuyruğuna aktarır. Küme gönderen kanal, iletileri yeni iletim kuyruğundan küme alıcı kanalına iletir. Kanal durumu, eski iletim kuyruğuyla ilişkilendirilmiş küme gönderen kanalını gösterir.

Geçiş işlemi, yeni gelen iletileri de aktarmaya devam eder. Bu adım, anahtar işlemi tarafından iletilecek kalan ileti sayısı sifıra ulaşıncaya kadar devam eder. İleti sayısı sifıra ulaştığında, yordam 2. adıma geçer.

Adım 1 sırasında, kanala ilişkin disk etkinliği artar. Kalıcı iletiler, ilk iletim kuyruğunda ve ikinci iletim kuyruğunda kesinleştirilir. Bu disk etkinliği, iletilerin normal olarak aktarılmasının bir parçası olarak iletilerin üzerine yerleştirilip iletim kuyruğundan kaldırılmasına ek olarak kesinleştirilmiştir. İdeal olarak, geçiş işlemi sırasında ileti gelmediği için geçiş mümkün olduğunca hızlı gerçekleşebilir. İletiler gelirse, geçiş işlemi tarafından işlenir.

#### **Adım 2-Yeni iletim kuyruğundaki iletilerin işlenmesi**

Kümeyi gönderen kanal için özgün iletim kuyruğunda hiçbir ileti kalmadığı anda, yeni iletiler doğrudan yeni iletim kuyruğuna yerleştirilir. Kanal durumu, küme gönderen kanalının yeni iletim kuyruğuyla ilişkilendirildiğini gösterir. Kuyruk yöneticisi hata günlüğüne şu ileti yazılır: " AMQ7341 Kanala ilişkin iletim kuyruğu *ChannelName* : *QueueName* ."

### **Birden çok küme iletim kuyruğu ve küme iletim kuyruğu özneliği**

Küme iletilerini, iletileri tek bir küme iletim kuyruğunda ya da birden çok kuyrukta saklayarak farklı kuyruk yöneticilerine iletme seçeneğiniz vardır. Bir kuyrukta, ayarlanacak ve sorgulanacak bir küme iletim kuyruğu öznelikleri kümesi vardır; birden çok kuyrukta birden çok kümeniz vardır. Bazı öznelikler için, birden çok kümeye sahip olmak bir avantajdır: örneğin, kuyruk derinliğinin sorgulanması, tüm kanallar tarafından değil, bir ya da bir kanal kümesi tarafından iletmeyi bekleyen ileti sayısını gösterir. Diğer öznelikler için, birden çok kümenin olması bir dezavantajdır: Örneğin, her küme iletim kuyruğu için aynı erişim izinlerinin konfigürasyonunu tanımlamak istemeyebilirsiniz. Bu nedenle, erişim izinleri her zaman SYSTEM. CLUSTER. TRANSMIT. QUEUEtanıtımıyla karşılaştırılarak denetlenir ve belirli bir küme iletim kuyruğuna ilişkin tanımlarla karşılaştırılmaz. Daha ayrıntılı güvenlik denetimleri uygulamak istiyorsanız bkz. [Erişim denetimi ve birden çok küme iletim kuyruğu](#).

### **Birden çok küme gönderen kanal ve birden çok iletim kuyruğu**

Kuyruk yöneticisi, bir iletiyi küme gönderen kanalına iletmeden önce küme iletim kuyruğuna saklar. İletinin hedefine bağlı bir küme gönderen kanalı seçer. Tümünü aynı hedefe bağlanan küme gönderici kanalları seçeneği olabilir. Hedef, birden çok küme gönderen kanalla tek bir kuyruk yöneticisine bağlı aynı fiziksel kuyruk olabilir. Hedef, aynı kümede farklı kuyruk yöneticilerine barındırılan, aynı kuyruk adına sahip birçok fiziksel kuyruk da olabilir. Bir hedefe bağlı küme-gönderen kanalları seçeneği varsa, iş yükü dengeleme algoritması birini seçer. Seçim bir dizi etkene bağlıdır; bkz. [Küme iş yükü yönetimi algoritması](#).

Şekil 38 sayfa 293, CL1. QM1, CL1. QM2 ve CS. QM1 içinde aynı hedefe yol açabilecek tüm kanallar vardır. Örneğin, CL1 içinde QM1 ve QM2 üzerinde Q1 tanımlarsanız, CL1. QM1 ve CL1. QM2 iki farklı kuyruk yöneticisinde aynı hedefe ( Q1) rotalar sağlar. CS. QM1 kanalı da CL1içindeyse, bu kanal Q1 iletilerinin alabileceği bir kanaldır. CS. QM1 küme üyeliği, bir küme ad listesi tarafından tanımlanabilir; bu nedenle kanal adı, oluşturulmasında bir küme adını içermez. İş yükü dengeleme parametrelerine ve gönderen



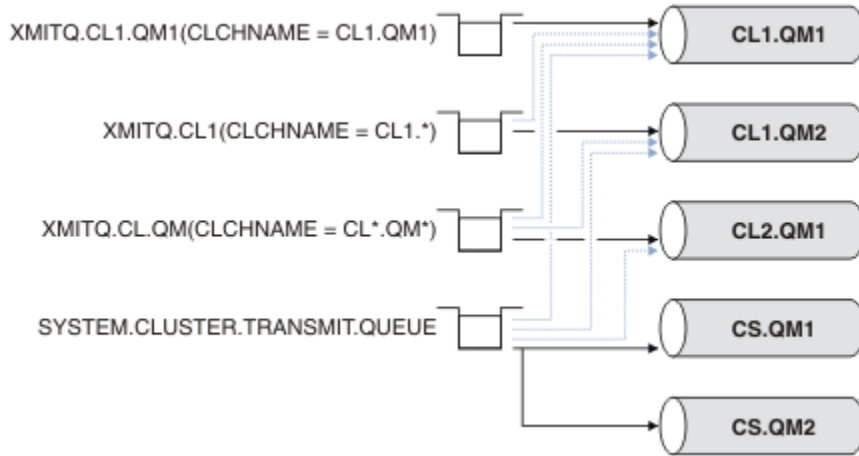
uygulamaya bağılı olarak, Q1 ile ilgili bazı iletiler her iletim kuyruğuna ( XMITQ . CL1 . QM1, XMITQ . CL1 ve SYSTEM . CLUSTER . TRANSMIT . CS . QM1) yerleştirilebilir.

Aynı hedefe ilişkin iletilerin farklı hedeflere ilişkin iletilerle kuyrukları ya da kanalları paylaşmaması için ileti trafiğini ayırmayı planlıyorsanız, önce trafiği farklı küme gönderen kanallarına nasıl böleceğinizi ve daha sonra belirli bir kanal için iletileri farklı bir iletim kuyruğuna nasıl ayıracağınızı göz önünde bulundurmanız gerekir. Aynı kümedeki, aynı kuyruk yöneticisindeki küme kuyrukları genellikle aynı küme kanallarını paylaşır. Tek başına birden çok küme iletim kuyruğunun tanımlanması, küme iletileri trafiğini farklı kuyruklara ayırmak için yeterli değildir. Farklı hedef kuyruklarına ilişkin iletileri farklı kanallarda ayırmadığınız sürece, iletiler aynı küme iletim kuyruğunu paylaşır.

İletilerin götürdüğü kanalları ayırmanın basit bir yolu, birden çok küme oluşturmaktır. Her bir kümedeki herhangi bir kuyruk yöneticisinde yalnızca bir küme kuyruğu tanımlayın. Daha sonra, her küme/kuyruk yöneticisi birleşimi için farklı bir küme alıcı kanalı tanımlarsanız, her küme kuyruğuna ilişkin iletiler diğer küme kuyruklarına ilişkin iletilerle bir küme kanalını paylaşmaz. Küme kanalları için ayrı iletim kuyrukları tanımlarsanız, gönderen kuyruk yöneticisi her iletim kuyruğunda yalnızca bir küme kuyruğuna ilişkin iletileri saklar. Örneğin, iki küme kuyruğunun kaynakları paylaşmamasını istiyorsanız, bunları aynı kuyruk yöneticisindeki farklı kümelere ya da aynı kümedeki farklı kuyruk yöneticilerine yerleştirebilirsiniz.

Küme iletim kuyruğunun seçimi, iş yükü dengeleme algoritmasını etkilemez. İş yükü dengeleme algoritması, bir iletiyi iletmek için hangi küme-gönderen kanalını seçer. İletiyi, o kanal tarafından hizmet verilen iletim kuyruğuna yerleştirir. İş yükü dengeleme algoritması yeniden seçim yapmak için çağrılırsa, örneğin kanal durursa, iletiyi iletmek için farklı bir kanal seçebilir. Farklı bir kanal seçerse ve yeni kanal iletileri farklı bir küme iletim kuyruğundan iletiyorsa, iş yükü dengeleme algoritması iletiyi diğer iletim kuyruğuna aktarır.

Şekil 38 sayfa 293'ünde, iki küme gönderen kanal ( CS . QM1 ve CS . QM2) varsayılan sistem iletim kuyruğuyla ilişkilendirilir. İş yükü dengeleme algoritması SYSTEM . CLUSTER . TRANSMIT . QUEUE' da ya da başka bir küme iletim kuyruğunda bir ileti sakladığında, iletiyi ileten küme gönderen kanalının adı, iletinin ilinti tanıtıcısında saklanır. Her kanal, yalnızca kanal adıyla ilinti tanıtıcısıyla eşleşen iletileri iletir.



Şekil 38. Birden çok küme gönderen kanalı

CS . QM1 durursa, bu küme gönderen kanal için iletim kuyruğundaki iletiler incelenir. Başka bir kanal tarafından iletilebilen iletiler, iş yükü dengeleme algoritması tarafından yeniden işlenir. İlinti tanıtıcıları, diğer bir küme-gönderen kanal adına sıfırlanır. Diğer küme gönderen kanalı CS . QM2 ise, ileti SYSTEM . CLUSTER . TRANSMIT . QUEUE üzerinde kalır. Diğer kanal CL1 . QM1 ise, iş yükü dengeleme algoritması iletiyi XMITQ . CL1 . QM1'e aktarır. Küme gönderen kanal yeniden başlatıldığında, farklı bir küme gönderen kanal için işaretli olmayan yeni iletiler ve iletiler kanal tarafından yeniden aktarılır.

Çalışmakta olan bir sistemdeki iletim kuyrukları ile küme gönderen kanalları arasındaki ilişkilendirmeyi değiştirebilirsiniz. İletim kuyruğundaki bir **CLCHNAME** parametresini ya da **DEFCLXQ** kuyruk yöneticisi parametresini değiştirebilirsiniz. Değişiklikten etkilenen bir kanal yeniden başlatıldığında, iletim kuyruğu geçiş işlemini başlatır; bkz. "Kümüye gönderen kanalını farklı bir iletim kuyruğuna aktarma işlemi nasıl çalışır?" sayfa 291.

İletim kuyruğunu deęiřtirme iřlemi, kanal yeniden bařlatıldıęında bařlar. İř yk yeniden dengeleme iřlemi kanal durdurulduęunda bařlar. İki iřlem paralel olarak alıřabilir.

Kme gnderen kanalının durdurulması, yeniden dengeleme iřleminin kuyruktaki iletileri iletmek iin kme gnderen kanalını deęiřtirmesine neden olmaz. Bu durumda, bařka bir kme gnderen kanalı iletileri doęru hedefe iletemez. İletileri hedeflerine iletmek iin alternatif bir kme gnderen kanalı olmadıęından, kme gnderen kanal durduktan sonra iletiler aynı kme gnderen kanalı iin iřaretlenmiř olarak kalır. Kanal bařlatıldıęında, bir anahtar beklemede olduęunda, geiř iřlemleri iletileri aynı kme gnderen kanalı tarafından iřlendikleri farklı bir iletim kuyruęuna tařır.

Daha karmařık bir durum, birden fazla kme gnderen kanalının bazı iletileri aynı hedefe iřleyebileceęi durumdur. İletim kuyruęu anahtarını tetiklelemek iin kme gnderen kanalını durdurup yeniden bařlatın. oęu durumda, kanalı yeniden bařlattıęınızda, iř yk dengeleme algoritması iletileri zgn iletim kuyruęundan farklı kme gnderen kanalları tarafından sunulan farklı iletim kuyruklarına tařımiřtır. Yalnızca farklı bir kme gnderen kanal tarafından iletilemeyen iletiler yeni iletim kuyruęuna aktarılacak. Bazı durumlarda, kanal hızlı bir Őekilde yeniden bařlatılırsa, iř yk dengeleme algoritması tarafından aktarılabilecek bazı iletiler kalır. Bu durumda, geri kalan bazı iletiler iř yk dengeleme iřlemi tarafından ve bazıları iletim kuyruęunu deęiřtirme iřlemi tarafından deęiřtirilir.

### **İlgili kavramlar**

Kme kanalları

Kmeleme: Birden ok kme iletim kuyruęu kullanan uygulama yalıtımı

“Gnlęn boyutunun hesaplanması” sayfa 633

Bir kuyruk yneticisinin gereksinim duyduęu gnlk boyutu tahmin ediyor.

### **İlgili grevler**

Kmeleme: Kme iletim kuyruklarının nasıl yapılandırılacaęını planlama

“Aę geidi kuyruk yneticisiyle akiřan iki kme oluřturma” sayfa 330

Bir aę geidi kuyruk yneticisiyle akiřan kmeler oluřturmak iin grevdeki ynergeleri izleyin. Kmeleri, bir uygulamadaki iletileri bir kmedeki dięer uygulamalara iletilerden yalıtma rnekleri iin bařlangı noktası olarak kullanın.

“Kmeye kuyruk yneticisi eklenmesi: ayrı iletim kuyrukları” sayfa 310

Yarattıęınız kmeye bir kuyruk yneticisi eklemek iin bu ynergeleri izleyin. Kme kuyruklarına ve konularına iliřkin iletiler, birden ok kme iletim kuyruęu kullanılarak aktarılır.

“Aę geidi kuyruk yneticisinden gnderilen kme iletileri yalıtma iin kme iletim kuyruęu eklenmesi” sayfa 337

Aę geidi kuyruk yneticisini kullanan akiřan kmelerin yapılandırmasını deęiřtirin. Deęiřiklik iletileri, dięer kme iletileriyle aynı iletim kuyruęu ya da kanalları kullanılmadan aę geidi kuyruk yneticisinden bir uygulamaya aktarıldıktan sonra, ' den bir uygulamaya aktarıldıktan sonra zm, ileti trafięini kmedeki tek bir kuyruk yneticisine ayırmak iin ek bir kme iletim kuyruęu kullanır.

“Aę geidi kuyruk yneticisinden gnderilen kme iletileri yalıtma iin kme ve kme iletim kuyruęu eklenmesi” sayfa 340

Aę geidi kuyruk yneticisini kullanan akiřan kmelerin yapılandırmasını deęiřtirin. Deęiřiklik iletileri, dięer kme iletileriyle aynı iletim kuyruęu ya da kanalları kullanılmadan aę geidi kuyruk yneticisinden bir uygulamaya aktarıldıktan sonra, ' den bir uygulamaya aktarıldıktan sonra zm, iletileri belirli bir kme kuyruęunda yalıtma iin ek bir kme kullanır.

## **Kmede iletiřim kurulması**

Teslim edilecek bir ileti olduęunda bir iletiřim kanalı bařlatmak iin kanal bařlatıcıyı gerekir. Kanal dinleyicisi, iletiyi almak iin kanalın dięer ucunu bařlatmayı bekler.

### **Bařlamadan nce**

Bir kmedeki kuyruk yneticileri arasında iletiřim kurmak iin, desteklenen iletiřim protokollerinden birini kullanarak bir baęlantı yapılandırın. Desteklenen protokoller řunlardır:

- Herhangi bir altyapıda TCP ya da LU 6.2

- **Windows** Windows sistemlerinde NetBIOS ya da SPX

Bu yapılandırmanın bir parçası olarak, dağıtılmış kuyruğa alma sırasında olduğu gibi kanal başlatıcılarına ve kanal dinleyicilerine de gereksinim duyarsınız.

## Bu görev hakkında

Tüm küme kuyruğu yöneticileri, sistem tarafından tanımlanan başlatma kuyruğunu SYSTEM.CHANNEL.INITQ izlemek için bir kanal başlatıcısına gereksinim duyar. SYSTEM.CHANNEL.INITQ, küme iletim kuyruğu da içinde olmak üzere tüm iletim kuyruklarına ilişkin başlatma kuyruğudur.

Her kuyruk yöneticisinin bir kanal dinleyicisi olmalıdır. Kanal dinleyici programı gelen ağ isteklerini bekler ve gerektiğinde uygun alıcı kanalını başlatır. Kanal dinleyicilerinin uygulanması platforma özgüdür, ancak bazı ortak özellikler vardır.

Tüm IBM MQ platformlarında, dinleyici **START LISTENER** komutu kullanılarak başlatılabilir.

**Multi** Multiplatforms üzerinde, dinleyiciyi kuyruk yöneticisiyle aynı anda otomatik olarak başlatabilirsiniz. Dinleyiciyi otomatik olarak başlatmak için, LISTENER nesnesinin CONTROL özniteliğini QMGR ya da STARTONLY olarak ayarlayın.

**z/OS** z/OS üzerinde CLUSRCVR kanalları için ve z/OS için CLUSSDR kanalları için paylaşılmayan bir dinleyici kapısı (INDISP (QMGR)) kullanılmalıdır.

## Yordam

### 1. Kanal başlatıcıyı başlatın.

- **z/OS** z/OS işletim sistemi üzerinde, her kuyruk yöneticisi için bir kanal başlatıcı vardır ve ayrı bir adres alanı olarak çalışır. Bunu, kuyruk yöneticisi başlatmanın bir parçası olarak çalıştırdığınız **MQSC START CHINIT** komutunu kullanarak başlatabilirsiniz.
- **ALW** AIX, Linux, and Windows işletim sistemlerinde, bir kuyruk yöneticisini başlattığınızda SCHINIT kuyruk yöneticisi özniteliği QMGR olarak ayarlanırsa, bir kanal başlatıcı otomatik olarak başlatılır. Ters durumda, **runmqsc START CHINIT** komutu ya da **runmqchi** denetim komutu kullanılarak başlatılabilir.
- **IBM i** IBM işletim sistemlerinde, bir kuyruk yöneticisini başlattığınızda SCHINIT kuyruk yöneticisi özniteliği QMGR olarak ayarlanırsa, bir kanal başlatıcı otomatik olarak başlatılır. Ters durumda, **runmqsc START CHINIT** komutu ya da **runmqchi** denetim komutu kullanılarak başlatılabilir.

### 2. Kanal dinleyicisini başlatın.

- **z/OS** z/OS üzerinde, IBM MQ tarafından sağlanan kanal dinleyici programını kullanın. Bir IBM MQ kanal dinleyicisini başlatmak için, kanal başlatıcı başlatmanın bir parçası olarak yayınladığınız **MQSC** komutunu **START LISTENER** kullanın. Örneğin:

```
START LISTENER PORT(1414) TRPTYPE(TCP)
```

ya da:

```
START LISTENER LUNAME(LONDON.LUNAME) TRPTYPE(LU62)
```

Bir kuyruk paylaşım grubunun üyeleri, her kuyruk yöneticisi için bir dinleyici yerine paylaşılan bir dinleyici kullanabilir. Kümelerle paylaşılan dinleyicileri kullanmayın. Özellikle, CLUSRCVR kanalının CONNAME değerini, kuyruk paylaşım grubunun paylaşılan dinleyicisinin adresi yapmayın. Bunu yaparsanız, kuyruk yöneticileri, tanımlaması olmayan kuyruklara ilişkin iletileri alabilir.

- **IBM i** IBM üzerinde, IBM MQ tarafından sağlanan kanal dinleyici programını kullanın. Bir IBM MQ kanal dinleyicisini başlatmak için **CL** komutunu MQMLSR kullanın. Örneğin:

```
STRMQMLSR MQMNAME(QM1) PORT(1414)
```

- **Windows** Windows işletim sisteminde, IBM MQ tarafından sağlanan kanal dinleyici programını ya da işletim sistemi tarafından sağlanan olanakları kullanın.

IBM MQ kanal dinleyicisini başlatmak için RUNMQLSR komutunu kullanın. Örneğin:

```
RUNMQLSR -t tcp -p 1414 -m QM1
```

- **Linux** **AIX** AIX and Linux işletim sisteminde, IBM MQ tarafından sağlanan kanal dinleyici programını ya da işletim sistemi tarafından sağlanan olanakları kullanın; örneğin, TCP iletişimi için **inetd**.

IBM MQ kanal dinleyicisini başlatmak için **runmqlsr** komutunu kullanın. Örneğin:

```
runmqlsr -t tcp -p 1414 -m QM1
```

Kanalları başlatmak üzere **inetd** 'yi kullanmak için iki dosya yapılandırın:

- /etc/services dosyasını düzenleyin. Ayrıcalıklı kullanıcı ya da kök olarak oturum açmanız gerekir. Dosyada aşağıdaki satır yoksa, gösterildiği gibi ekleyin:

```
MQSeries 1414/tcp # WebSphere MQ channel listener
```

Burada 1414, IBM MQ için gerekli olan kapı numarasıdır. Kapı numarasını değiştirebilirsiniz, ancak kapı numarasının gönderme sonunda belirtilen kapı numarasıyla eşleşmesi gerekir.

- /etc/inetd.conf dosyasını düzenleyin. Bu dosyada aşağıdaki satır yoksa, gösterildiği gibi ekleyin:

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
-m queue.manager.name
```

Burada `MQ_INSTALLATION_PATH`, IBM MQ 'un kurulu olduğu üst düzey dizinle değiştirilir.

**inetd** yapılandırma dosyalarını yeniden okuduktan sonra güncellemeler etkin duruma gelir. Kök kullanıcı kimliğinden aşağıdaki komutları verin:

**AIX** AIX'ta:

```
refresh -s inetd
```

**Linux** Linux üzerinde:

- inetd** işlem tanıtıcısını şu komutla bulun:

```
ps -ef | grep inetd
```

- Uygun komutu çalıştırın.

Linux için:

```
kill -1 inetd processid
```

## Yeni bir küme ayarlama

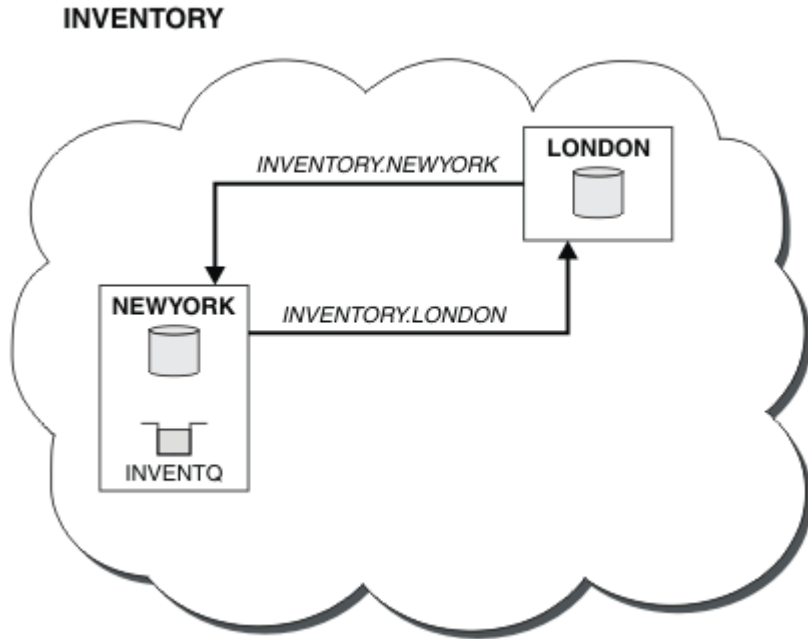
Örnek kümeyi ayarlamak için bu yönergeleri izleyin. Ayrı yönergeler, kümenin TCP/IP, LU 6.2 üzerinde ve tek bir iletim kuyruğuyla ya da birden çok iletim kuyruğuyla ayarlanmasını açıklar. Bir kuyruk yöneticisinden diğerine ileti göndererek kümeyi sınavın.

### Başlamadan önce

- Bu yönergeleri izlemek yerine, IBM MQ Explorer ile verilen sihirbazlardan birini kullanarak bu görev tarafından yaratılan kümeye benzer bir küme yaratabilirsiniz. Kuyruk Yöneticisi Kümeleri klasörünü sağ tıklayın ve **Yeni > Kuyruk Yöneticisi Kümesi** seçeneğini tıklayın ve sihirbazda verilen yönergeleri izleyin.
- Bir küme oluşturmak için atılan adımları anlamanızda size yardımcı olacak arka plan bilgileri için bkz. [“Küme kuyruklarının tanımlanması”](#) sayfa 284, [Küme kanalları](#) ve [Dinleyiciler](#).

### Bu görev hakkında

Bir zincir mağaza için yeni bir IBM MQ ağı kurarsınız. Mağazanın biri Londra 'da, diğeri New York 'ta olmak üzere iki şubesi vardır. Her bir depoya ilişkin veriler ve uygulamalar, ayrı kuyruk yöneticileri çalıştıran sistemler tarafından barındırılır. İki kuyruk yöneticisine LONDON ve NEWYORK adı verilir. Döküm uygulaması, NEWYORK kuyruk yöneticisine bağlı olarak New York 'taki sistemde çalışır. Uygulama, NEWYORK tarafından barındırılan INVENTQ kuyruğundaki iletilerin gelişiyle yönlendirilir. İki kuyruk yöneticisi ( LONDON ve NEWYORK), her ikisi de INVENTQ' e ileti yerleştirebilmek için INVENTORY adlı bir kümede bağlanabilir.



Bu kümenin görünüşü şöyledir:

Kümedeki her kuyruk yöneticisini, farklı küme iletim kuyrukları kullanarak kümedeki diğer kuyruk yöneticilerine ileti gönderecek şekilde yapılandırabilirsiniz.

Kümenin kurulmasına ilişkin yönergeler, iletim protokolüne, iletim kuyruklarının sayısına ya da platforma göre değişiklik gösterir. Üç kombinasyon seçeneğiniz var. Doğrulama yordamı tüm birleşimler için aynı kalır.

INVENTORY küçük bir kümedir. Bununla birlikte, kavramın bir kanıtı olarak kullanışlıdır. Bu küme hakkında anlamanız gereken önemli şey, gelecekteki geliştirme için sunduğu kapsamdır.

### Yordam

- [“Kuyruk yöneticisi başına tek bir iletim kuyruğuyla TCP/IP kullanılarak küme oluşturulması”](#) sayfa 298

- “Her kuyruk yöneticisi için birden çok iletim kuyruğu kullanılarak TCP/IP ' de küme ayarlanması” sayfa 301
- “z/OS üzerinde LU 6.2 kullanılarak küme ayarlanması” sayfa 304
- “Kümenin doğrulanması” sayfa 306

### İlgili kavramlar

Kümeler

Kümeleme ve dağıtılmış kuyruğa alma karşılaştırması

Bir kümenin bileşenleri

### İlgili görevler

“Kuyruk yöneticisi kümesinin yapılandırılması” sayfa 283

Kümeler, hem ilk yapılandırmayı hem de devam eden yönetimi basitleştiren bir şekilde birbirine bağlanan kuyruk yöneticileri için bir mekanizma sağlar. Küme bileşenleri tanımlayabilir ve kümeleri oluşturabilir ve yönetebilirsiniz.

## ***Kuyruk yöneticisi başına tek bir iletim kuyruğuyla TCP/IP kullanılarak küme oluşturulması***

Bu, basit bir küme için farklı yapılandırmaları açıklayan üç başlıktan biridir.


### Başlamadan önce

Yaratılmakta olan kümeye genel bakış için bkz. “Yeni bir küme ayarlama” sayfa 297.

Kuyruk yöneticisi özniteliği ( **DEFCLXQ**), varsayılan değeri ( SCTQ) olarak bırakılmalıdır.

### Bu görev hakkında

İletim protokolü TCP/IP ' yi kullanarak Çoklu platformlar üzerinde bir küme oluşturmak için aşağıdaki

adımları izleyin.  z/OS üzerinde, “4” sayfa 299. adımda dinleyicileri tanımlamak yerine TCP/IP bağlantısını kurmak için “z/OS üzerinde TCP bağlantısı tanımlanması” sayfa 975 içindeki yönergeleri izlemeniz gerekir. Tersi durumda, z/OS için adımlar aynıdır, ancak hata iletileri kuyruk yöneticisi hata günlüğü yerine konsola yazılır.

### Yordam

1. Kümenin kuruluşuna ve adına karar verin.

İki kuyruk yöneticisini ( LONDON ve NEWYORK) bir kümeye bağlamaya karar verdiniz. Yalnızca iki kuyruk yöneticisine sahip bir küme, dağıtılmış kuyruğa alma özelliğini kullanan bir ağ üzerinde yalnızca marjinal yarar sağlar. Bu, başlamak için iyi bir yoldur ve gelecekteki genişleme için kapsam sağlar. Mağazanızın yeni dallarını açtığınızda, yeni kuyruk yöneticilerini kümeye kolayca ekleyebilirsiniz. Yeni kuyruk yöneticilerinin eklenmesi var olan ağı kesintiye uğratmaz; bkz. “Kümeye kuyruk yöneticisi eklenmesi” sayfa 308.

Şu anda çalıştırdığınız tek uygulama stok uygulamasıdır. Küme adı: INVENTORY.

2. Hangi kuyruk yöneticilerinin tam havuzları tutacağına karar verin.

Herhangi bir kümede, tam havuzları tutmak için en az bir ya da tercihen iki kuyruk yöneticisini aday göstermeniz gerekir. Bu örnekte, her ikisi de tam havuzları tutan yalnızca iki kuyruk yöneticisi vardır: LONDON ve NEWYORK.

- a. Geri kalan adımları herhangi bir sırada gerçekleştirebilirsiniz.
- b. Adımlarda ilerledikçe, uyarı iletileri kuyruk yöneticisi günlüğüne yazılabilir. İletiler, henüz eklediğiniz tanımların eksik olmasından kaynaklanır.

Examples of the responses to the commands are shown in a box like this after each step in this task. These examples show the responses returned by IBM MQ for AIX. The responses vary on other platforms.

- c. Bu adımlara devam etmeden önce kuyruk yöneticilerinin başlatıldığından emin olun.
3. Havuz tanımlamaları eklemek için kuyruk yöneticisi tanımlamalarını değiştirin.

Tam havuzu tutacak her kuyruk yöneticisinde, ALTER QMGR REPOS komutunu kullanarak ve REPOS özniteliğini belirterek yerel kuyruk yöneticisi tanımlamasını değiştirin:

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

Örneğin, şunu girerseniz:

- a. runmqsc LONDON  
b. ALTER QMGR REPOS(INVENTORY)

LONDON , tam bir havuza çevrilir.

4. Dinleyicileri tanımlayın.

Kümedeki her kuyruk yöneticisi için diğer kuyruk yöneticilerinden gelen ağ isteklerini kabul eden bir dinleyici tanımlayın. LONDON kuyruk yöneticilerine aşağıdaki komutu verin:

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

CONTROL özniteliği, dinleyicinin kuyruk yöneticisi başlattığında başlamasını ve durmasını sağlar.

Dinleyici tanımlandığında başlatılmadığı için, aşağıdaki MQSC komutuyla ilk kez el ile başlatılmalıdır:

```
START LISTENER(LONDON_LS)
```

Her biri için dinleyici adını değiştirerek, kümedeki diğer tüm kuyruk yöneticileri için benzer komutlar verin.

[Dinleyiciler'](#) de gösterildiği gibi, bu dinleyicileri tanımlamanın birkaç yolu vardır.

5. LONDON kuyruk yöneticisi için CLUSRCVR kanalını tanımlayın.

Bir kümedeki her kuyruk yöneticisinde, kuyruk yöneticisinin iletileri alabileceği bir küme-alıcı kanalı tanımlarsınız. Bkz. Küme alıcı kanalı: CLUSRCVR . CLUSRCVR kanalı, kuyruk yöneticisinin bağlantı adını tanımlar. Bağlantı adı, diğer kuyruk yöneticilerinin başvuruda bulunabileceği havuzlarda saklanır. CLUSTER anahtar sözcüğü, kümedeki diğer kuyruk yöneticilerinden ileti almak için kuyruk yöneticisinin kullanılabilirliğini gösterir.

Bu örnekte kanal adı INVENTORY . LONDON ve bağlantı adı ( CONNAME ) , kuyruk yöneticisinin bulunduğu makinenin ( LONDON . CHSTORE . COM ) ağ adresidir. Ağ adresi, alfasayısal bir DNS anasistem adı ya da IPv4 noktalı onlu biçiminde bir IP adresi olarak girilebilir. Örneğin, 192 . 0 . 2 . 0 ya da IPv6 onaltılı biçimi; örneğin, 2001 : DB8 : 0204 : acff : fe97 : 2c34 : fde0 : 3485 . Kapı numarası belirtilmediğinden varsayılan kapı (1414) kullanılır.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

#### 6. NEWYORK kuyruk yöneticisi için CLUSRCVR kanalını tanımlayın.

Kanal dinleyicisi varsayılan kapıyı (genellikle 1414) kullanıyorsa ve küme z/OSüzerinde bir kuyruk yöneticisi içermiyorsa, CONNAME değerini atlayabilirsiniz.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

#### 7. LONDON kuyruk yöneticisinde CLUSSDR kanalını tanımlayın.

Her tam havuz kuyruğu yöneticisinden kümedeki diğer tüm havuz kuyruğu yöneticisine bir CLUSSDR kanalını el ile tanımlayabilirsiniz. Bkz. [Cluster-sender channel: CLUSSDR](#). Bu durumda, her ikisi de tam havuzları tutan yalnızca iki kuyruk yöneticisi vardır. Her biri, diğer kuyruk yöneticisinde tanımlanan CLUSRCVR kanalını gösteren el ile tanımlanmış bir CLUSSDR kanalına gereksinim duyar. CLUSSDR tanımlamalarında belirtilen kanal adları, ilgili CLUSRCVR tanımlamalarındaki kanal adlarıyla eşleşmelidir. Bir kuyruk yöneticisinin aynı kümedeki hem küme alıcı kanalı hem de küme gönderen kanalı için tanımlamaları varsa, küme gönderen kanalı başlatılır.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

#### 8. NEWYORK kuyruk yöneticisinde CLUSSDR kanalını tanımlayın.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

#### 9. Küme kuyruğunun tanımlanması INVENTQ

NEWYORK kuyruk yöneticisinde INVENTQ kuyruğunu, CLUSTER anahtar sözcüğünü belirterek tanımlayın.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

CLUSTER anahtar sözcüğü, kuyruğun kümeye duyurulmasına neden olur. Kuyruk tanımları tanımlanmaz, kümedeki diğer kuyruk yöneticilerinin kullanımına sunulur. Uzak kuyruk tanımlaması yapmak zorunda kalmadan ileti gönderebilirler.

Tüm tanımlar tamamlandı. Tüm altyapılarda, her kuyruk yöneticisinde bir dinleyici programı başlatın. Dinleyici programı gelen ağ isteklerini bekler ve gerektiğinde küme alıcı kanalını başlatır.

## Sonraki adım

Artık [kümeyi doğrulamaya](#) hazırsınız.



## İlgili görevler

[“Her kuyruk yöneticisi için birden çok iletim kuyruğu kullanılarak TCP/IP ' de küme ayarlanması” sayfa 301](#)

Bu, basit bir küme için farklı yapılandırmaları açıklayan üç başlıktan biridir.

[“z/OS üzerinde LU 6.2 kullanılarak küme ayarlanması” sayfa 304](#)

Bu, basit bir küme için farklı yapılandırmaları tanımlayan ağaç konularından biridir.

## ***Her kuyruk yöneticisi için birden çok iletim kuyruğu kullanılarak TCP/IP ' de küme ayarlanması***

Bu, basit bir küme için farklı yapılandırmaları açıklayan üç başlıktan biridir.

## Başlamadan önce

Yaratılmakta olan kümeye genel bakış için bkz. [“Yeni bir küme ayarlama” sayfa 297](#).

## Bu görev hakkında

İletim protokolü TCP/IP ' yi kullanarak Çoklu platformlar üzerinde bir küme oluşturmak için aşağıdaki adımları izleyin. Havuz kuyruğu yöneticileri, birbirine ve kümedeki diğer kuyruk yöneticilerine ileti göndermek için farklı bir küme iletim kuyruğu kullanacak şekilde yapılandırılır. Farklı iletim kuyrukları kullanacak olan kümeye kuyruk yöneticileri eklerseniz, [“Kümeye kuyruk yöneticisi eklenmesi: ayrı iletim kuyrukları” sayfa 310](#) görevini izleyin.

## Yordam

1. Kümenin kuruluşuna ve adına karar verin.

İki kuyruk yöneticisini ( LONDON ve NEWYORK) bir kümeye bağlamaya karar verdiğinizde. Yalnızca iki kuyruk yöneticisine sahip bir küme, dağıtılmış kuyruğa alma özelliğini kullanan bir ağ üzerinde yalnızca marjinal yarar sağlar. Bu, başlamak için iyi bir yoldur ve gelecekteki genişleme için kapsam sağlar. Mağazanızın yeni dallarını açtığınızda, yeni kuyruk yöneticilerini kümeye kolayca ekleyebilirsiniz. Yeni kuyruk yöneticilerinin eklenmesi var olan ağı kesintiye uğratmaz; bkz. [“Kümeye kuyruk yöneticisi eklenmesi” sayfa 308](#).

Şu anda çalıştırdığınız tek uygulama stok uygulamasıdır. Küme adı: INVENTORY.

2. Hangi kuyruk yöneticilerinin tam havuzları tutacağına karar verin.

Herhangi bir kümede, tam havuzları tutmak için en az bir ya da tercihen iki kuyruk yöneticisini aday göstermeniz gerekir. Bu örnekte, her ikisi de tam havuzları tutan yalnızca iki kuyruk yöneticisi vardır: LONDON ve NEWYORK.

- a. Geri kalan adımları herhangi bir sırada gerçekleştirebilirsiniz.
- b. Adımlarda ilerledikçe, uyarı iletileri kuyruk yöneticisi günlüğüne yazılabilir. İletiler, henüz eklenmediğiniz tanımların eksik olmasından kaynaklanır.

Examples of the responses to the commands are shown in a box like this after each step in this task. These examples show the responses returned by IBM MQ for AIX. The responses vary on other platforms.

- c. Bu adımlara devam etmeden önce kuyruk yöneticilerinin başlatıldığından emin olun.

3. Havuz tanımlamaları eklemek için kuyruk yöneticisi tanımlamalarını değiştirin.

Tam havuzu tutacak her kuyruk yöneticisinde, ALTER QMGR komutunu kullanarak ve REPOS özniteliğini belirterek yerel kuyruk yöneticisi tanımlamasını değiştirin:

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

Örneğin, şunu girerseniz:

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON , tam bir havuza çevrilir.

4. Kuyruk yöneticisi tanımlarını, her hedef için ayrı küme iletim kuyrukları yaratacak şekilde değiştirin.

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

Kümeye eklediğiniz her kuyruk yöneticisinde, ayrı iletim kuyruklarının kullanılıp kullanılmayacağına karar verin. “Kümeye kuyruk yöneticisi eklenmesi” sayfa 308 ve “Kümeye kuyruk yöneticisi eklenmesi: ayrı iletim kuyrukları” sayfa 310 konularına bakın.

5. Dinleyicileri tanımlayın.

Kümedeki her kuyruk yöneticisi için diğer kuyruk yöneticilerinden gelen ağ isteklerini kabul eden bir dinleyici tanımlayın. LONDON kuyruk yöneticilerine aşağıdaki komutu verin:

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

CONTROL özniteliği, dinleyicinin kuyruk yöneticisi başlattığında başlamasını ve durmasını sağlar.

Dinleyici tanımlandığında başlatılmadığı için, aşağıdaki MQSC komutuyla ilk kez el ile başlatılmalıdır:

```
START LISTENER(LONDON_LS)
```

Her biri için dinleyici adını değiştirerek, kümedeki diğer tüm kuyruk yöneticileri için benzer komutlar verin.

Dinleyiciler' de gösterildiği gibi, bu dinleyicileri tanımlamanın birkaç yolu vardır.

6. LONDON kuyruk yöneticisi için CLUSRCVR kanalını tanımlayın.

Bir kümedeki her kuyruk yöneticisinde, kuyruk yöneticisinin iletileri alabileceği bir küme-alıcı kanalı tanımlarsınız. Bkz. Küme alıcı kanalı: CLUSRCVR . CLUSRCVR kanalı, kuyruk yöneticisinin bağlantı adını tanımlar. Bağlantı adı, diğer kuyruk yöneticilerinin başvuruda bulunabileceği havuzlarda saklanır. CLUSTER anahtar sözcüğü, kümedeki diğer kuyruk yöneticilerinden ileti almak için kuyruk yöneticisinin kullanılabilirliğini gösterir.

Bu örnekte kanal adı INVENTORY . LONDON ve bağlantı adı ( CONNAME ) , kuyruk yöneticisinin bulunduğu makinenin ( LONDON . CHSTORE . COM ) ağ adresidir. Ağ adresi, alfasayısal bir DNS anasistem adı ya da IPv4 noktalı onlu biçiminde bir IP adresi olarak girilebilir. Örneğin, 192 . 0 . 2 . 0 ya da IPv6 onaltılı biçimi; örneğin, 2001 : DB8 : 0204 : acff : fe97 : 2c34 : fde0 : 3485 . Kapı numarası belirtilmediğinden varsayılan kapı (1414) kullanılır.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

7. NEWYORK kuyruk yöneticisi için CLUSRCVR kanalını tanımlayın.

Kanal dinleyicisi varsayılan kapıyı (genellikle 1414) kullanıyorsa ve küme z/OS üzerinde bir kuyruk yöneticisi içermiyorsa, CONNAME değerini atlayabilirsiniz.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

#### 8. LONDON kuyruk yöneticisinde CLUSSDR kanalını tanımlayın.

Her tam havuz kuyruğu yöneticisinden kümedeki diğer tüm havuz kuyruğu yöneticisine bir CLUSSDR kanalını el ile tanımlayabilirsiniz. Bkz. [Cluster-sender channel: CLUSSDR](#) . Bu durumda, her ikisi de tam havuzları tutan yalnızca iki kuyruk yöneticisi vardır. Her biri, diğer kuyruk yöneticisinde tanımlanan CLUSRCVR kanalını gösteren el ile tanımlanmış bir CLUSSDR kanalına gereksinim duyar. CLUSSDR tanımlamalarında belirtilen kanal adları, ilgili CLUSRCVR tanımlamalarındaki kanal adlarıyla eşleşmelidir. Bir kuyruk yöneticisinin aynı kümedeki hem küme alıcı kanalı hem de küme gönderen kanalı için tanımlamaları varsa, küme gönderen kanalı başlatılır.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

#### 9. NEWYORK kuyruk yöneticisinde CLUSSDR kanalını tanımlayın.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

#### 10. Küme kuyruğunun tanımlanması INVENTQ

NEWYORK kuyruk yöneticisinde INVENTQ kuyruğunu, CLUSTER anahtar sözcüğünü belirterek tanımlayın.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

CLUSTER anahtar sözcüğü, kuyruğun kümeye duyurulmasına neden olur. Kuyruk tanımları tanımlanmaz, kümedeki diğer kuyruk yöneticilerinin kullanımına sunulur. Uzak kuyruk tanımlaması yapmak zorunda kalmadan ileti gönderebilirler.

Tüm tanımlar tamamlandı. Tüm altyapılarda, her kuyruk yöneticisinde bir dinleyici programı başlatın. Dinleyici programı gelen ağ isteklerini bekler ve gerektiğinde küme alıcı kanalını başlatır.

### Sonraki adım

Artık kümeyi doğrulamaya hazırsınız.

### İlgili görevler

“Kuyruk yöneticisi başına tek bir iletim kuyruğuyla TCP/IP kullanılarak küme oluşturulması” sayfa 298  
Bu, basit bir küme için farklı yapılandırmaları açıklayan üç başlıktan biridir.

“z/OS üzerinde LU 6.2 kullanılarak küme ayarlanması” sayfa 304

Bu, basit bir küme için farklı yapılandırmaları tanımlayan ağaç konularından biridir.

## ***z/OS üzerinde LU 6.2 kullanılarak küme ayarlanması***

Bu, basit bir küme için farklı yapılandırmaları tanımlayan ağaç konularından biridir.

### **Başlamadan önce**

Yaratılmakta olan kümeye genel bakış için bkz. [“Yeni bir küme ayarlama” sayfa 297.](#)

### **Yordam**

1. Kümenin kuruluşuna ve adına karar verin.

İki kuyruk yöneticisini ( LONDON ve NEWYORK) bir kümeye bağlamaya karar verdiniz. Yalnızca iki kuyruk yöneticisine sahip bir küme, dağıtılmış kuyruğa alma özelliğini kullanan bir ağ üzerinde yalnızca marjinal yarar sağlar. Bu, başlamak için iyi bir yoldur ve gelecekteki genişleme için kapsam sağlar. Mağazanızın yeni dallarını açtığınızda, yeni kuyruk yöneticilerini kümeye kolayca ekleyebilirsiniz. Yeni kuyruk yöneticilerinin eklenmesi var olan ağı kesintiye uğratmaz; bkz. [“Kümeye kuyruk yöneticisi eklenmesi” sayfa 308.](#)

Şu anda çalıştırdığınız tek uygulama stok uygulamasıdır. Küme adı: INVENTORY.

2. Hangi kuyruk yöneticilerinin tam havuzları tutacağına karar verin.

Herhangi bir kümede, tam havuzları tutmak için en az bir ya da tercihen iki kuyruk yöneticisini aday göstermeniz gerekir. Bu örnekte, her ikisi de tam havuzları tutan yalnızca iki kuyruk yöneticisi vardır: LONDON ve NEWYORK.

- a. Geri kalan adımları herhangi bir sırada gerçekleştirebilirsiniz.
- b. Adımlarda ilerledikçe, uyarı iletileri z/OS sistem konsoluna yazılabilir. İletiler, henüz ekmediğiniz tanımların eksik olmasından kaynaklanır.
- c. Bu adımlara devam etmeden önce kuyruk yöneticilerinin başlatıldığından emin olun.

3. Havuz tanımlamaları eklemek için kuyruk yöneticisi tanımlamalarını değiştirin.

Tam havuzu tutacak her kuyruk yöneticisinde, ALTER QMGR REPOS kullanarak ve REPOS özniteliğini belirterek yerel kuyruk yöneticisi tanımlamasını değiştirin:

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

Örneğin, şunu girerseniz:

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON , tam bir havuza çevrilir.

4. Dinleyicileri tanımlayın.

 Bkz. [z/OS üzerindeki kanal başlatıcısı ve “LU 6.2 üzerinde alma” sayfa 979.](#)

Dinleyici tanımlandığında başlatılmadığı için, aşağıdaki MQSC komutuyla ilk kez el ile başlatılmalıdır:

```
START LISTENER(LONDON_LS)
```

Her biri için dinleyici adını değiştirerek, kümedeki diğer tüm kuyruk yöneticileri için benzer komutlar verin.

5. LONDON kuyruk yöneticisi için CLUSRCVR kanalını tanımlayın.

Bir kümedeki her kuyruk yöneticisinde, kuyruk yöneticisinin iletileri alabileceği bir küme-alıcı kanalı tanımlarsınız. Bkz. [Küme alıcı kanalı: CLUSRCVR](#) . CLUSRCVR kanalı, kuyruk yöneticisinin bağlantı adını tanımlar. Bağlantı adı, diğer kuyruk yöneticilerinin başvuruda bulunabileceği havuzlarda saklanır. CLUSTER anahtar sözcüğü, kümedeki diğer kuyruk yöneticilerinden ileti almak için kuyruk yöneticisinin kullanılabilirliğini gösterir.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

#### 6. NEWYORK kuyruk yöneticisi için CLUSRCVR kanalını tanımlayın.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager NEWYORK')
```

#### 7. LONDON kuyruk yöneticisinde CLUSSDR kanalını tanımlayın.

Her tam havuz kuyruğu yöneticisinden kümedeki diğer tüm havuz kuyruğu yöneticisine bir CLUSSDR kanalını el ile tanımlayabilirsiniz. Bkz. [Cluster-sender channel: CLUSSDR](#) . Bu durumda, her ikisi de tam havuzları tutan yalnızca iki kuyruk yöneticisi vardır. Her biri, diğer kuyruk yöneticisinde tanımlanan CLUSRCVR kanalını gösteren el ile tanımlanmış bir CLUSSDR kanalına gereksinim duyar. CLUSSDR tanımlamalarında belirtilen kanal adları, ilgili CLUSRCVR tanımlamalarındaki kanal adlarıyla eşleşmelidir. Bir kuyruk yöneticisinin aynı kümedeki hem küme alıcı kanalı hem de küme gönderen kanalı için tanımlamaları varsa, küme gönderen kanalı başlatılır.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(CPIC) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

#### 8. NEWYORK kuyruk yöneticisinde CLUSSDR kanalını tanımlayın.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from NEWYORK to repository at LONDON')
```

#### 9. Küme kuyruğunun tanımlanması INVENTQ

NEWYORK kuyruk yöneticisinde INVENTQ kuyruğunu, CLUSTER anahtar sözcüğünü belirterek tanımlayın.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

CLUSTER anahtar sözcüğü, kuyruğun kümeye duyurulmasına neden olur. Kuyruk tanımları tanımlanmaz, kümedeki diğer kuyruk yöneticilerinin kullanımına sunulur. Uzak kuyruk tanımlaması yapmak zorunda kalmadan ileti gönderebilirler.

Tüm tanımlar tamamlandı. Tüm altyapılarda, her kuyruk yöneticisinde bir dinleyici programı başlatın. Dinleyici programı gelen ağ isteklerini bekler ve gerektiğinde küme alıcı kanalını başlatır.

## Sonraki adım

Artık kümeyi [doğrulamaya](#) hazırsınız.

### İlgili görevler

[“Kuyruk yöneticisi başına tek bir iletim kuyruğuyla TCP/IP kullanılarak küme oluşturulması” sayfa 298](#)  
Bu, basit bir küme için farklı yapılandırmaları açıklayan üç başlıktan biridir.

[“Her kuyruk yöneticisi için birden çok iletim kuyruğu kullanılarak TCP/IP ' de küme ayarlanması” sayfa 301](#)

Bu, basit bir küme için farklı yapılandırmaları açıklayan üç başlıktan biridir.

## Kümenin doğrulanması

Eşdüzey konular, basit bir küme için üç farklı yapılandırmayı açıklar. Bu konu, kümenin nasıl doğrulanacağını açıklar.

## Başlamadan önce

Bu konuda, aşağıdaki görevlerden biriyle yarattığınız bir kümeyi doğruladığınız varsayılır:

- [“Kuyruk yöneticisi başına tek bir iletim kuyruğuyla TCP/IP kullanılarak küme oluşturulması” sayfa 298.](#)
- [“Her kuyruk yöneticisi için birden çok iletim kuyruğu kullanılarak TCP/IP ' de küme ayarlanması” sayfa 301.](#)
- [“z/OS üzerinde LU 6.2 kullanılarak küme ayarlanması” sayfa 304.](#)

Yaratılan kümeye ilişkin genel bakış için bkz. [“Yeni bir küme ayarlama” sayfa 297.](#)

## Bu görev hakkında

Kümeyi aşağıdaki yollardan birini ya da birkaçını kullanarak doğrulayabilirsiniz:

1. Küme ve kanal özniteliklerini görüntülemek için yönetim komutlarını çalıştırma.
2. Bir küme kuyruğundaki iletileri göndermek ve almak için örnek programları çalıştırın.
3. Küme kuyruğuna bir istek iletileri göndermek ve kümelenmemiş yanıt kuyruğuna yanıt iletileri göndermek için kendi programlarınızı yazın.

## Yordam

Kümeyi doğrulamak için DISPLAY **runmqsc** komutlarını yayınlayın.

Gördüğünüz yanıtlar, takip eden adımlardaki yanıtlara benzer olmalı.

1. NEWYORK kuyruk yöneticisinden **DISPLAY CLUSQMgr** komutunu çalıştırın:

```
dis clusqmgr(*)
```

```
1 : dis clusqmgr(*)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(NEWYORK) CLUSTER(INVENTORY)
CHANNEL(INVENTORY.NEWYORK)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(LONDON) CLUSTER(INVENTORY)
CHANNEL(INVENTORY.LONDON)
```

2. NEWYORK kuyruk yöneticisinden **DISPLAY CHANNEL STATUS** komutunu çalıştırın:

```
dis chstatus(*)
```

```
1 : dis chstatus(*)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.NEWYORK) XMITQ( )
CONNNAME(192.0.2.0) CURRENT
CHLTYPE(CLUSRCVR) STATUS(RUNNING)
RQMNAME(LONDON)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.LONDON) XMITQ(SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON)
CONNNAME(192.0.2.1) CURRENT
CHLTYPE(CLUSSDR) STATUS(RUNNING)
RQMNAME(LONDON)
```

**amqspu**t komutunu kullanarak iki kuyruk yöneticisi arasında ileti gönderin.

3. LONDON üzerinde **amqspu**t **INVENTQ LONDON** komutunu çalıştırın.

Ardından boş bir satır gelen bazı iletileri yazın.

4. NEWYORK üzerinde **amqsge**t **INVENTQ NEWYORK** komutunu çalıştırın.

Şimdi LONDON' da girdiğiniz iletileri görürsünüz. 15 saniye sonra program sona erer.

Kendi programlarınızı kullanarak iki kuyruk yöneticisi arasında ileti gönderin.

Aşağıdaki adımlarda LONDON , NEWYORK adresindeki INVENTQ ' e bir ileti gönderir ve kuyruğunda bir yanıt alır LONDON\_reply.

5. LONDON ' ta küme kuyruğuna bir ileti koyun.

- LONDON\_reply adlı bir yerel kuyruk tanımlayın.
- MQOPEN seçeneklerini MQOO\_OUTPUT olarak ayarlayın.
- Kuyruğu açmak için MQOPEN çağrısı INVENTQ yayınlayın.
- İleti tanımlayıcıdaki ReplyToQ adını LONDON\_reply olarak ayarlayın.
- İletiyi koymak için MQPUT aramasını yayınlayın.
- İletiyi kesinleştirin.

6. NEWYORK üzerinde, küme kuyruğundaki iletiyi alın ve yanıt kuyruğuna bir yanıt koyun.

- MQOPEN seçeneklerini MQOO\_BROWSE olarak ayarlayın.
- Kuyruğu açmak için MQOPEN çağrısı INVENTQ yayınlayın.
- INVENTQ' den iletiyi almak için MQGET çağrısına yanıt verin.
- İleti tanımlayıcıdan ReplyToQ adını alın.
- Nesne tanımlayıcının ObjectName alanına ReplyToQ adını yazın.
- MQOPEN seçeneklerini MQOO\_OUTPUT olarak ayarlayın.
- Kuyruk yöneticisinde LONDON\_reply dosyasını açmak için MQOPEN çağrısı LONDON yayınlayın.
- İletiyi LONDON\_reply' e koymak için MQPUT aramasını yayınlayın.

7. LONDON tarihinde yanıtı alın.

- MQOPEN seçeneklerini MQOO\_BROWSE olarak ayarlayın.
- Kuyruğu açmak için MQOPEN çağrısı LONDON\_reply yayınlayın.
- LONDON\_reply' den iletiyi almak için MQGET aramasını yayınlayın.

## Kümeye kuyruk yöneticisi eklenmesi

Yarattığınız kümeye bir kuyruk yöneticisi eklemek için bu yönergeleri izleyin. Küme kuyruklarına ve konularına ilişkin iletiler, tek küme iletim kuyruğu SYSTEM . CLUSTER . TRANSMIT . QUEUE kullanılarak aktarılır.

### Başlamadan önce

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- INVENTORY kümesi, “Yeni bir küme ayarlama” sayfa 297’inde açıklandığı gibi ayarlanır. Her ikisi de tam havuzları tutan iki kuyruk yöneticisi ( LONDON ve NEWYORK) içerir.
- Kuyruk yöneticisi PARIS birincil kuruluşa aittir. Değilse, PARIS ' in ait olduğu kuruluş için komut ortamını ayarlamak üzere **setmqenv** komutunu çalıştırmanız gerekir.
- Üç sistem arasında TCP bağlantırlığı vardır ve kuyruk yöneticisi, kuyruk yöneticisinin denetimi altında başlayan bir TCP dinleyicisiyle yapılandırılır.

### Bu görev hakkında

1. Paris 'te zincir mağazanın yeni bir dalı kuruluyor ve kümeye PARIS adlı bir kuyruk yöneticisi eklemek istiyorsunuz.
2. Kuyruk yöneticisi PARIS , INVENTORY kuyruğuna iletiler koyarak New York 'ta sistemde çalışan uygulamaya döküm güncellemeleri gönderir.

Bir kümeye kuyruk yöneticisi eklemek için aşağıdaki adımları izleyin.

### Yordam

1. PARIS ' nin önce hangi tam havuza başvuruda bulunacağına karar verin.

Bir kümedeki her kuyruk yöneticisi, tam havuzlardan birine ya da diğerine başvurmalıdır. Kümeyle ilgili bilgileri tam bir havuzdan toplar ve böylece kendi kısmi havuzunu oluşturur. Havuzlardan birini tam havuz olarak seçin. Kümeye yeni bir kuyruk yöneticisi eklenir eklenmez, diğer havuz hakkında da hemen bilgi edinir. Bir kuyruk yöneticisinde yapılan değişikliklerle ilgili bilgiler doğrudan iki havuza gönderilir. Bu örnekte, PARIS ögesini yalnızca coğrafi nedenlerden ötürü LONDON kuyruk yöneticisine bağlayabilirsiniz.


**Not:** PARIS kuyruk yöneticisi başlatıldıktan sonra, geri kalan adımları herhangi bir sırayla gerçekleştirin.

2. PARIS kuyruk yöneticisinde bir CLUSRCVR kanalı tanımlayın.

Bir kümedeki her kuyruk yöneticisi, ileti alabileceği bir küme-alıcı kanalı tanımlamalıdır. PARIS üzerinde aşağıdakileri tanımlayın:

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

Küme alıcı kanalı, INVENTORY kümesindeki diğer kuyruk yöneticilerinden ileti almak için kuyruk yöneticisinin kullanılabilirliğini tanıtır. INVENTORY . PARIS küme alıcı kanalına gönderme için diğer kuyruk yöneticilerine tanımlama yapmayın. Diğer tanımlar gerektiğinde otomatik olarak yapılır. Bkz. Küme kanalları.

3.  z/OS  
IBM MQ for z/OS üzerinde kanal başlatıcıyı başlatın.
4. PARIS kuyruk yöneticisinde bir CLUSSDR kanalı tanımlayın.



Bir kümeye tam havuz olmayan bir kuyruk yöneticisi eklediğinizde, tam havuzla ilk bağlantı kurmak için yalnızca bir küme gönderen kanalı tanımlarsınız. Bkz. [Cluster-sender channel: CLUSSDR](#) .

PARİS işletim sistemlerinde, LONDON . CHSTORE . COMağ adresiyle kuyruk yöneticisine INVENTORY . LONDON adlı bir CLUSSDR kanalı için aşağıdaki tanımlamayı yapın.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

5. İsteğe bağlı: Bir kümeye daha önce aynı kümeden kaldırılmış bir kuyruk yöneticisi ekliyorsanız, kümenin şimdi küme üyesi olarak gösterilip göstermediğini denetleyin. Yoksa, aşağıdaki ek adımları tamamlayın:

a) Eklemede olduğunuz kuyruk yöneticisinde **REFRESH CLUSTER** komutunu verin.

Bu adım, küme kanallarını durdurur ve yerel küme önbelleğinize, kümenin geri kalanında güncel olacağı garantisini olan yeni bir dizi sıra numarası sağlar.

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

**Not:** Büyük kümeler için, **REFRESH CLUSTER** komutunun kullanılması, devam ederken kümede kesintiye neden olabilir ve bundan sonra küme nesneleri otomatik olarak ilgili tüm kuyruk yöneticilerine durum güncellemelerini gönderdiğinde, 27 günlük aralıklarla kümede kesintiye neden olabilir. Bkz. [Büyük bir kümede yenilenmesi kümenin performansını ve kullanılabilirliğini etkileyebilir.](#)

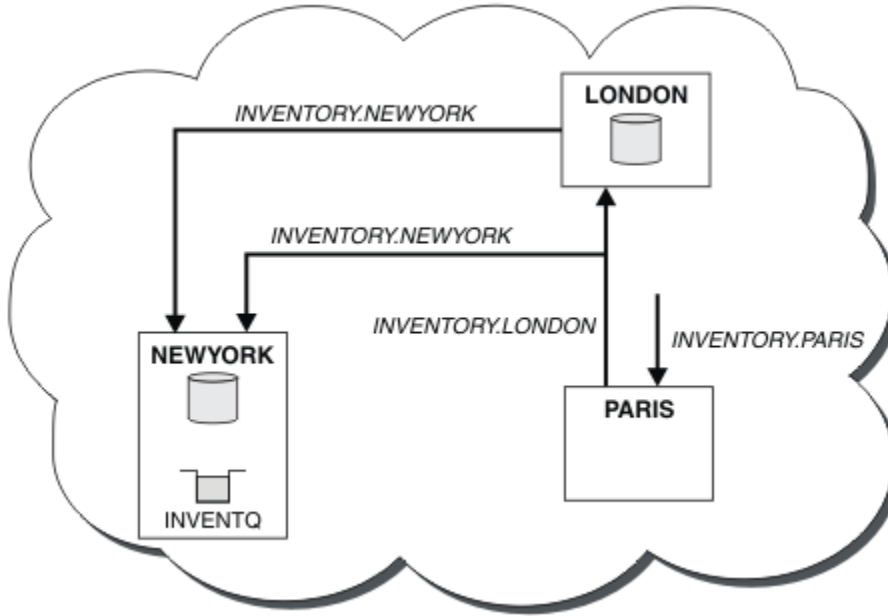
b) CLUSSDR kanalını yeniden başlat

(örneğin, [START CHANNEL](#) komutunu kullanarak).

c) CLUSRCVR kanalını yeniden başlatın.

## Sonuçlar

Aşağıdaki şekil, bu görev tarafından ayarlanan kümeyi göstermektedir.



Şekil 39. Üç kuyruk yöneticisiyle INVENTORY kümesi

CLUSRCVR tanımlaması ve CLUSSDR tanımlaması olmak üzere yalnızca iki tanımlama yaparak, kümeye kuyruk yöneticisini PARIS ekledik.

Şimdi PARIS kuyruk yöneticisi, LONDONadresindeki tam havuzdan INVENTQ kuyruğunun kuyruk yöneticisi tarafından barındırıldığını NEWYORKöğrenir. Paris 'te sistem tarafından barındırılan bir uygulama INVENTQ' e ileti yerleştirmeye çalıştığında, PARIS küme alıcı kanalına bağlanmak için otomatik olarak bir küme gönderen kanalı tanımlar INVENTORY . NEWYORK. Uygulama, kuyruk yöneticisi adı hedef kuyruk yöneticisi olarak belirtildiğinde ve bir yanıt kuyruğu sağlandığında yanıt alabilir.

### **Kümeye kuyruk yöneticisi eklenmesi: ayrı iletim kuyrukları**

Yarattığınız kümeye bir kuyruk yöneticisi eklemek için bu yönergeleri izleyin. Küme kuyruklarına ve konularına ilişkin iletiler, birden çok küme iletim kuyruğu kullanılarak aktarılır.

### **Başlamadan önce**

- Kuyruk yöneticisi herhangi bir kümenin üyesi değil.
- Küme var; bu kuyruk yöneticisinin doğrudan bağlanabileceği ve havuzun kullanılabilirliği tam bir havuz var. Kümeyi yaratmaya ilişkin adımlar için bkz. [“Yeni bir küme ayarlama” sayfa 297.](#)

### **Bu görev hakkında**

Bu görev, küme iletilerini tek bir iletim kuyruğuna yerleşen bir kümeye kuyruk yöneticisi eklediğiniz [“Kümeye kuyruk yöneticisi eklenmesi” sayfa 308'](#) e bir alternatiftir.

Bu görevde, her küme gönderen kanalı için otomatik olarak ayrı küme iletim kuyrukları yaratan bir kümeye kuyruk yöneticisi eklersiniz.

Kuyruk tanımlarının sayısını küçük tutmak için varsayılan değer, tek bir iletim kuyruğunun kullanılmasıdır. Farklı kuyruk yöneticilerine ve farklı kümelere yönlendirilmiş trafiği izlemek istiyorsanız, ayrı iletim kuyruklarının kullanılması avantajlıdır. Yalıtım ya da performans hedeflerine ulaşmak için trafiği farklı hedeflere ayırmak da isteyebilirsiniz.

### **Yordam**

1. Varsayılan küme kanalı iletim kuyruğu tipini değiştirin.

PARISKuyruk yöneticisini değiştirin:

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

Kuyruk yöneticisi, kuyruk yöneticisine ileti göndermek için bir küme gönderen kanalı oluşturduğunda, bir küme iletim kuyruğu yaratır. İletim kuyruğu yalnızca bu küme-gönderen kanalı tarafından kullanılır. İletim kuyruğu kalıcı dinamik. Model kuyruğundan ( SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE ), SYSTEM . CLUSTER . TRANSMIT . *ChannelName* adıyla oluşturulur.



**Uyarı:** IBM WebSphere MQ 7.5 sürümünden önceki bir sürümden yükseltilmiş bir kuyruk yöneticisiyle özel olarak ayrılmış SYSTEM . CLUSTER . TRANSMIT . QUEUES kullanıyorsanız, SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE ' de SHARE/NOSHARE seçeneğinin **SHARE** olarak ayarlandığından emin olun.

2. PARIS ' nin önce hangi tam havuza başvuruda bulunacağına karar verin.

Bir kümedeki her kuyruk yöneticisi, tam havuzlardan birine ya da diğerine başvurmalıdır. Kümeyle ilgili bilgileri tam bir havuzdan toplar ve böylece kendi kısmi havuzunu oluşturur. Havuzlardan birini tam havuz olarak seçin. Kümeye yeni bir kuyruk yöneticisi eklenir eklenmez, diğer havuz hakkında da hemen bilgi edinir. Bir kuyruk yöneticisinde yapılan değişikliklerle ilgili bilgiler doğrudan iki havuza gönderilir. Bu örnekte, PARIS ögesini yalnızca coğrafi nedenlerden ötürü LONDONkuyruk yöneticisine bağlayabilirsiniz.

**Not:** PARIS kuyruk yöneticisi başlatıldıktan sonra, geri kalan adımları herhangi bir sırayla gerçekleştirin.

3. PARISKuyruk yöneticisinde bir CLUSRCVR kanalı tanımlayın.

Bir kümedeki her kuyruk yöneticisi, ileti alabileceği bir küme-alıcı kanalı tanımlamalıdır. PARIS üzerinde aşağıdakileri tanımlayın:

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

Küme alıcı kanalı, INVENTORY kümesindeki diğer kuyruk yöneticilerinden ileti almak için kuyruk yöneticisinin kullanılabilirliğini tanıtır. INVENTORY.PARIS küme alıcı kanalına gönderme için diğer kuyruk yöneticilerine tanımlama yapmayın. Diğer tanımlar gerektiğinde otomatik olarak yapılır. Bkz. [Küme kanalları](#).

#### 4. PARIS kuyruk yöneticisinde bir CLUSSDR kanalı tanımlayın.

Bir kümeye tam havuz olmayan bir kuyruk yöneticisi eklediğinizde, tam havuzla ilk bağlantı kurmak için yalnızca bir küme gönderen kanalı tanımlarsınız. Bkz. [Cluster-sender channel: CLUSSDR](#).

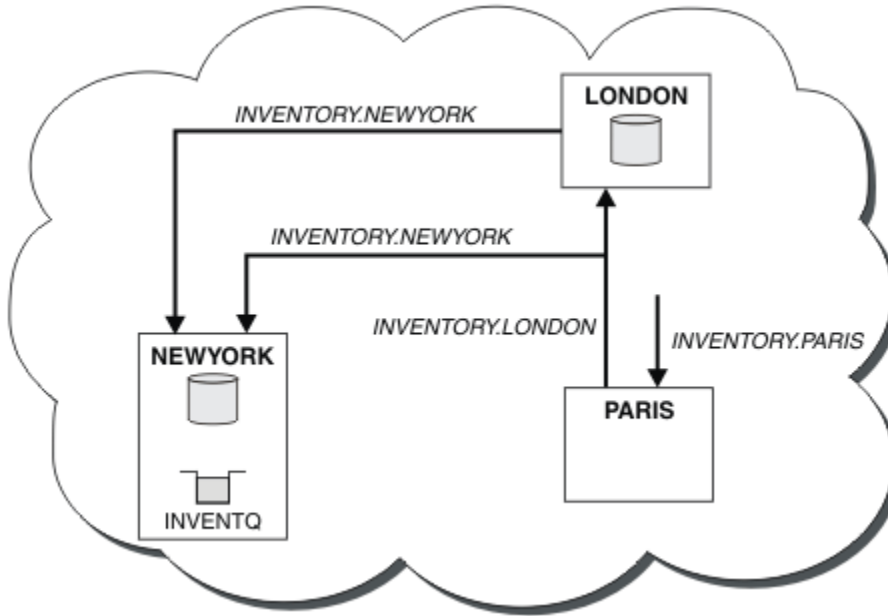
PARIS işletim sistemlerinde, LONDON.CHSTORE.COM ağ adresiyle kuyruk yöneticisine INVENTORY.LONDON adlı bir CLUSSDR kanalı için aşağıdaki tanımlamayı yapın.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

Kuyruk yöneticisi, kalıcı dinamik küme iletim kuyruğunu SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON model kuyruğundan SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE otomatik olarak yaratır. İletim kuyruğunun CLCHNAME özneliğini INVENTORY.LONDON olarak ayarlar.

## Sonuçlar

Aşağıdaki şekil, bu görev tarafından ayarlanan kümeyi göstermektedir.



Şekil 40. Üç kuyruk yöneticisiyle INVENTORY kümesi

CLUSRCVR tanımlaması ve CLUSSDR tanımlaması olmak üzere yalnızca iki tanımlama yaparak, kümeye kuyruk yöneticisini PARIS ekledik.

Şimdi PARIS kuyruk yöneticisi, LONDON adresindeki tam havuzdan INVENTQ kuyruğunun kuyruk yöneticisi tarafından barındırıldığını NEWYORK öğrenir. Paris 'te sistem tarafından barındırılan bir uygulama INVENTQ' e ileti yerleştirmeye çalışıldığında, PARIS küme alıcı kanalına bağlanmak için otomatik olarak bir küme

gönderen kanalı tanımlar INVENTORY . NEWYORK. Uygulama, kuyruk yöneticisi adı hedef kuyruk yöneticisi olarak belirtildiğinde ve bir yanıt kuyruğu sağlandığında yanıt alabilir.

### **İlgili kavramlar**

Hangi küme iletim kuyruğunun kullanılacağını seçme

### **İlgili görevler**

DHCP kullanarak bir kümeye kuyruk yöneticisi ekleme

DHCP kullanarak bir kümeye kuyruk yöneticisi ekleyin. Görev, CLUSRCVR tanımlamasında CONNAME değerinin atılmasını gösterir.

### **DHCP kullanarak bir kümeye kuyruk yöneticisi ekleme**

DHCP kullanarak bir kümeye kuyruk yöneticisi ekleyin. Görev, CLUSRCVR tanımlamasında CONNAME değerinin atılmasını gösterir.

### **Başlamadan önce**

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Görev iki özel özelliği gösterir:

- CLUSRCVR tanımlamasında CONNAME değerini atlama yeteneği.
- Bir CLUSSDR tanımında +QMNAME+ kullanma yeteneği.

z/OS üzerinde hiçbir özellik sağlanmaz.

Senaryo:

- INVENTORY kümesi, "Yeni bir küme ayarlama" sayfa 297 içinde açıklandığı gibi ayarlandı. Her ikisi de tam havuzları tutan iki kuyruk yöneticisi ( LONDON ve NEWYORK) içerir.
- Paris 'te zincir mağazanın yeni bir dalı kuruluyor ve kümeye PARIS adlı bir kuyruk yöneticisi eklemek istiyorsunuz.
- Kuyruk yöneticisi PARIS , iletileri INVENTQ kuyruğuna koyarak New York 'ta sistemde çalışan uygulamaya döküm güncellemeleri gönderir.
- Üç sistem arasında ağ bağlantısı vardır.
- Ağ protokolü TCP 'dir.
- PARIS kuyruk yöneticisi sistemi DHCP kullanır; bu, sistem yeniden başlatıldığında IP adreslerinin değişebileceğini gösterir.
- PARIS ve LONDON sistemleri arasındaki kanallar, tanımlı bir adlandırma kuralına göre adlandırılır. Bu kural, LONDON üzerindeki tam havuz kuyruk yöneticisinin kuyruk yöneticisi adını kullanır.
- PARIS kuyruk yöneticisinin denetimcilerinin LONDON havuzundaki kuyruk yöneticisinin adıyla ilgili bilgileri yoktur. LONDON havuzundaki kuyruk yöneticisinin adı değiştirilebilir.

### **Bu görev hakkında**

DHCP kullanarak bir kümeye kuyruk yöneticisi eklemek için aşağıdaki adımları izleyin.

### **Yordam**

1. PARIS ' nin önce hangi tam havuza başvuruda bulunacağına karar verin.

Bir kümedeki her kuyruk yöneticisi, tam havuzlardan birine ya da diğerine başvurmalıdır. Kümeyle ilgili bilgileri tam bir havuzdan toplar ve böylece kendi kısmi havuzunu oluşturur. Havuzlardan birini tam havuz olarak seçin. Kümeye yeni bir kuyruk yöneticisi eklenir eklenmez, diğer havuz hakkında da hemen bilgi edinir. Bir kuyruk yöneticisinde yapılan değişikliklerle ilgili bilgiler doğrudan iki havuza gönderilir. Bu örnekte, PARIS ' i yalnızca coğrafi nedenlerden ötürü LONDON kuyruk yöneticisine bağlamayı seçiyoruz.

**Not:** PARIS kuyruk yöneticisi başlatıldıktan sonra, geri kalan adımları herhangi bir sırayla gerçekleştirin.

2. PARIS kuyruk yöneticisinde bir CLUSRCVR kanalı tanımlayın.

Bir kümedeki her kuyruk yöneticisinin, iletileri alabileceği bir küme alıcı kanalı tanımlaması gerekir. PARIS üzerinde aşağıdakileri tanımlayın:

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR)
TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

Küme alıcı kanalı, INVENTORY kümesindeki diğer kuyruk yöneticilerinden ileti almak için kuyruk yöneticisinin kullanılabilirliğini tanıtır. Küme alıcı kanalında CONNAME değerini belirtmenize gerek yoktur. IBM MQ 'den CONNAME değerini çıkararak ya da CONNAME(' ') değerini belirterek bağlantı adını sistemden bulmasını isteyebilirsiniz. IBM MQ , sistemin geçerli IP adresini kullanarak CONNAME değerini oluşturur; bkz. [CONNAME . INVENTORY . PARIS](#) küme alıcı kanalına gönderme için diğer kuyruk yöneticilerine tanımlama yapmaya gerek yoktur. Diğer tanımlar gerektiğinde otomatik olarak yapılır.

3. PARIS kuyruk yöneticisinde bir CLUSSDR kanalı tanımlayın.

Bir kümedeki her kuyruk yöneticisinin, ilk tam havuzuna ileti gönderebildiği bir küme-gönderen kanalı tanımlaması gerekir. PARIS sistemlerinde, INVENTORY . +QMNAME+ adlı bir kanal için, LONDON . CHSTORE . COMağ adresiyle kuyruk yöneticisine aşağıdaki tanımlamayı yapın.

```
DEFINE CHANNEL(INVENTORY.+QMNAME+) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. İsteğe bağlı: Bir kümeye daha önce aynı kümeden kaldırılmış bir kuyruk yöneticisi ekliyorsanız, kümenin şimdi küme üyesi olarak gösterilip göstermediğini denetleyin. Yoksa, aşağıdaki ek adımları tamamlayın:

a) Eklemede olduğunuz kuyruk yöneticisinde **REFRESH CLUSTER** komutunu verin.

Bu adım, küme kanallarını durdurur ve yerel küme önbelleğinize, kümenin geri kalanında güncel olacağı garantisini olan yeni bir dizi sıra numarası sağlar.

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

**Not:** Büyük kümeler için, **REFRESH CLUSTER** komutunun kullanılması, devam ederken kümede kesintiye neden olabilir ve bundan sonra küme nesnelere otomatik olarak ilgili tüm kuyruk yöneticilerine durum güncellemelerini gönderdiğinde, 27 günlük aralıklarla kümede kesintiye neden olabilir. Bkz. [Büyük bir kümede yenilenmesi kümenin performansını ve kullanılabilirliğini etkileyebilir.](#)

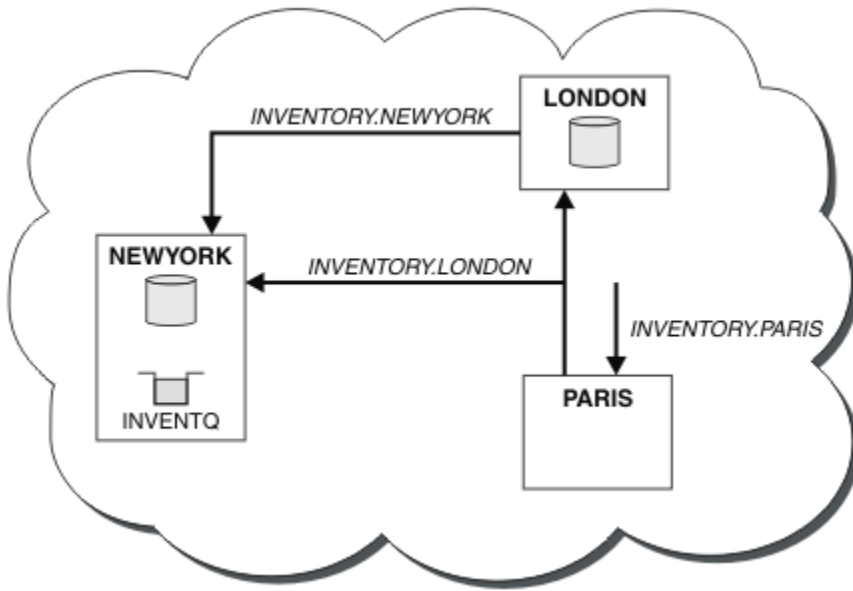
b) CLUSSDR kanalını yeniden başlat

(örneğin, [START CHANNEL](#) komutunu kullanarak).

c) CLUSRCVR kanalını yeniden başlatın.

## Sonuçlar

Bu görev tarafından ayarlanan küme, "[Küme kuyruk yöneticisi eklenmesi](#)" sayfa 308 ile aynıdır:



Şekil 41. Üç kuyruk yöneticisiyle INVENTORY kümesi

CLUSRCVR tanımlaması ve CLUSSDR tanımlaması olmak üzere yalnızca iki tanımlama yaparak, kümeye kuyruk yöneticisini PARIS ekledik.

PARIS kuyruk yöneticisinde, +QMNAME+ dizgisini içeren CLUSSDR başlar. LONDON Sistemde IBM MQ , +QMNAME+ ögesini kuyruk yöneticisi adına ( LONDON) çözer. IBM MQ , INVENTORY . LONDON adlı bir kanala ilişkin tanımlamayı ilgili CLUSRCVR tanımlamasıyla eşleştirir.

IBM MQ , çözülen kanal adını PARIS kuyruk yöneticisine geri gönderir. PARIS'da, INVENTORY . +QMNAME+ adlı kanala ilişkin CLUSSDR kanal tanımı, INVENTORY . LONDON için dahili olarak oluşturulan CLUSSDR tanımıyla değiştirilir. Bu tanımlama, çözülen kanal adını içerir; tersi durumda, yaptığınız +QMNAME+ tanımlamasıyla aynıdır. Küme havuzları, yeni çözülen kanal adıyla birlikte kanal tanımlamasıyla da güncel olarak getirilir.

#### Not:

1. +QMNAME+ adıyla oluşturulan kanal hemen devre dışı kalır. Hiçbir zaman veri iletmek için kullanılmaz.
2. Kanal çıkışları, kanal adının bir başlatma ile sonraki başlatma arasında değiştiğini görebilir.

Şimdi PARIS kuyruk yöneticisi, LONDON adresindeki havuzdan INVENTQ kuyruğunun kuyruk yöneticisi tarafından barındırıldığını NEWYORK öğrenir. Paris 'te sistem tarafından barındırılan bir uygulama, INVENTQ, PARIS ' e otomatik olarak küme alıcı kanalına bağlanmak için bir küme gönderen kanalı tanımlar INVENTORY . NEWYORK. Uygulama, kuyruk yöneticisi adı hedef kuyruk yöneticisi olarak belirtildiğinde ve bir yanıt kuyruğu sağlandığında yanıt alabilir.

#### İlgili görevler

Kümeye kuyruk yöneticisi eklenmesi: ayrı iletim kuyrukları

Yarattığınız kümeye bir kuyruk yöneticisi eklemek için bu yönergeleri izleyin. Küme kuyruklarına ve konularına ilişkin iletiler, birden çok küme iletim kuyruğu kullanılarak aktarılır.

#### İlgili başvurular

KANAL TANIMLAYIN

### Kuyruğu barındıran bir kuyruk yöneticisi eklenmesi

Başka bir INVENTQ kuyruğu barındırmak için kümeye başka bir kuyruk yöneticisi ekleyin. İstekler, her kuyruk yöneticisindeki kuyruklara sırayla gönderilir. Var olan INVENTQ anasisteminde değişiklik yapılmasına gerek yoktur.

## Başlamadan önce

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- INVENTORY kümesi, “Kümeye kuyruk yöneticisi eklenmesi” sayfa 308’inde açıklandığı gibi ayarlandı. Üç kuyruk yöneticisi içerir; LONDON ve NEWYORK her ikisi de tam havuzları tutar, PARIS kısmi bir havuzu tutar. Döküm uygulaması, NEWYORK kuyruk yöneticisine bağlı olarak New York 'taki sistemde çalışır. Uygulama, INVENTORY kuyruğundaki iletilerin gelişyle yönlendirilir.
- Toronto 'da yeni bir mağaza kuruluyor. Ek kapasite sağlamak için New York 'un yanı sıra Toronto 'daki sistemde stok uygulamasını çalıştırmak istiyorsunuz.
- Dört sistem arasında ağ bağlantısı vardır.
- Ağ protokolü TCP 'dir.

**Not:** TORONTO kuyruk yöneticisi yalnızca kısmi bir havuz içeriyor. Bir kümeye tam havuz kuyruk yöneticisi eklemek istiyorsanız bkz. “Tam havuzun başka bir kuyruk yöneticisine taşınması” sayfa 319.

## Bu görev hakkında

Bir kuyruğu barındıran bir kuyruk yöneticisi eklemek için aşağıdaki adımları izleyin.

### Yordam

1. TORONTO ' nin önce hangi tam havuza başvuruda bulunacağına karar verin.

Bir kümedeki her kuyruk yöneticisi, tam havuzlardan birine ya da diğerine başvurmalıdır. Kümeyle ilgili bilgileri tam bir havuzdan toplar ve böylece kendi kısmi havuzunu oluşturur. Seçtiğiniz havuzun özel bir önemi yoktur. Bu örnekte, NEWYORK'ögesini seçiyoruz. Yeni kuyruk yöneticisi kümeye katıldıktan sonra, her iki havuzla da iletişim kurar.

2. CLUSRCVR kanalını tanımlayın.

Bir kümedeki her kuyruk yöneticisinin, iletileri alabileceği bir küme alıcı kanalı tanımlaması gerekir. TORONTO'üzerinde bir CLUSRCVR kanalı tanımlayın:

```
DEFINE CHANNEL(INVENTORY.TORONTO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(TORONTO.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for TORONTO')
```

TORONTO kuyruk yöneticisi, küme alıcı kanalını kullanarak INVENTORY kümesindeki diğer kuyruk yöneticilerinden ileti almak için kullanılabilirliği tanıtır.

3. TORONTOkuyruk yöneticisinde bir CLUSSDR kanalı tanımlayın.

Bir kümedeki her kuyruk yöneticisinin, ilk tam havuzuna ileti gönderebildiği bir küme gönderen kanalı tanımlaması gerekir. Bu durumda NEWYORKseçeneğini belirleyin. TORONTO için aşağıdaki tanımlama gereklidir:

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from TORONTO to repository at NEWYORK')
```

4. İsteğe bağlı: Bir kümeye daha önce aynı kümeden kaldırılmış bir kuyruk yöneticisi ekliyorsanız, kümenin şimdi küme üyesi olarak gösterilip göstermediğini denetleyin. Yoksa, aşağıdaki ek adımları tamamlayın:

- a) Eklemede olduğunuz kuyruk yöneticisinde **REFRESH CLUSTER** komutunu verin.

Bu adım, küme kanallarını durdurur ve yerel küme önbelleğinize, kümenin geri kalanında güncel olacağı garantisini olan yeni bir dizi sıra numarası sağlar.

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

**Not:** Büyük kümeler için, **REFRESH CLUSTER** komutunun kullanılması, devam ederken kümede kesintiye neden olabilir ve bundan sonra küme nesneleri otomatik olarak ilgili tüm kuyruk yöneticilerine durum güncellemelerini gönderdiğinde, 27 günlük aralıklarla kümede kesintiye neden olabilir. Bkz. Büyük bir kümede yenilenmesi kümenin performansını ve kullanılabilirliğini etkileyebilir.

- b) CLUSSDR kanalını yeniden başlat  
(örneğin, `START CHANNEL` komutunu kullanarak).
  - c) CLUSRCVR kanalını yeniden başlatın.
5. İleti yakınlıkları için stok uygulamasını gözden geçirin.

Devam etmeden önce, döküm uygulamasının iletilerin işleme sırasına bağımlılığı olmadığından emin olun ve uygulamayı Toronto 'da sisteme kurun.

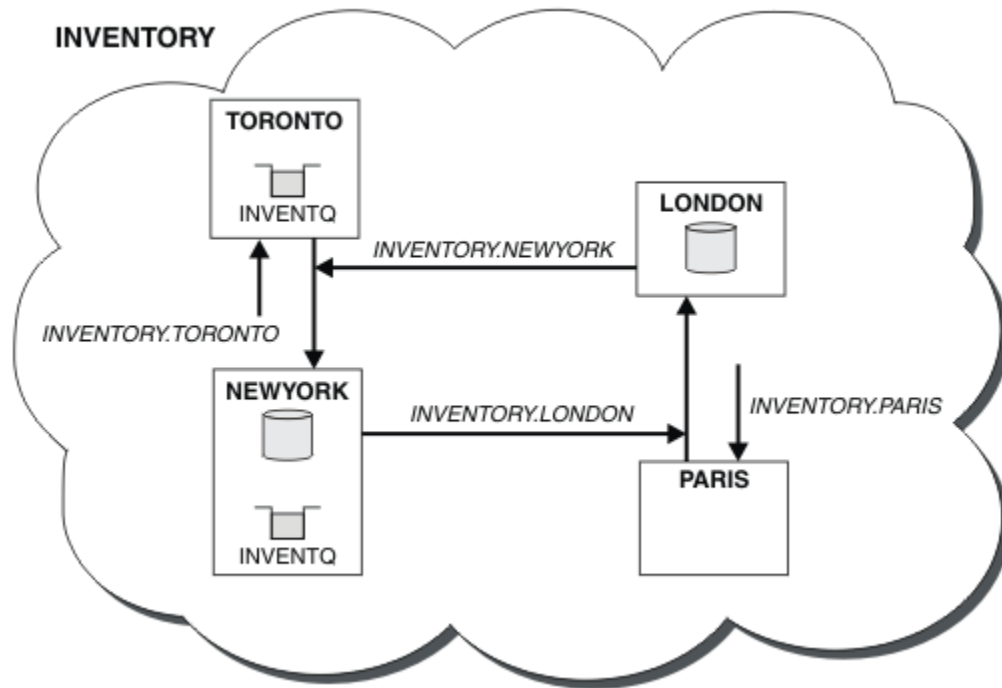
6. INVENTQ küme kuyruğunu tanımlayın.

NEWYORK kuyruk yöneticisi tarafından barındırılan INVENTQ kuyruğu da TORONTO tarafından barındırılır. TORONTO kuyruk yöneticisinde aşağıdaki gibi tanımlayın:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

## Sonuçlar

Şekil 42 sayfa 316 içinde bu görev tarafından ayarlanan INVENTORY kümesi gösterilmektedir.



Şekil 42. Dört kuyruk yöneticisiyle INVENTORY kümesi

INVENTQ kuyruğu ve döküm uygulaması artık kümedeki iki kuyruk yöneticisinde barındırılır. Bu, kullanılabilirliğini artırır, iletilerin verimini hızlandırır ve iş yükünün iki kuyruk yöneticisi arasında dağıtılmasını sağlar. INVENTQ 'e TORONTO ya da NEWYORK tarafından konan iletiler, mümkün olduğunda yerel kuyruk yöneticisindeki eşgörünüm tarafından işlenir. LONDON ya da PARIS tarafından konan iletiler, iş yükünün dengelenmesi için dönüşümlü olarak TORONTO ya da NEWYORK kuyruğuna yönlendirilir.



Kümede yapılan bu deęişiklik, kuyruk yöneticilerindeki NEWYORK, LONDONve PARIS tanımlamalarını deęiřtirmek zorunda kalmadan gerekleřtirildi. Bu kuyruk yöneticilerindeki tüm havuzlar, TORONTOadresindeki INVENTQ ' e ileti gönderebilmeleri için gereksinim duydukları bilgilerle otomatik olarak güncellenir. NEWYORK ya da TORONTO kuyruk yöneticisinden biri kullanılamaz duruma gelirse ve yeterli kapasiteye sahipse, döküm uygulaması alıřmaya devam eder. Stok uygulaması, her iki konumda da barındırıldıysa doęru bir řekilde alıřabilmelidir.

Bu görevin sonucunda görebileceęiniz gibi, aynı uygulama birden ok kuyruk yöneticisinde alıřıyor olabilir. Daęıtım iř yüküne eřit olarak kümeleme yapabilirsiniz.

Bir uygulama her iki konumdaki kayıtları iřleyemeyebilir. Örneęin, LONDON ve NEWYORKiinde alıřan bir müşteri hesabı sorgusu eklemeye ve uygulamayı güncellemeye karar verdięinizi varsayalım. Bir hesap kaydı yalnızca bir yerde tutulabilir. Bir veri bölümlene teknięi kullanarak isteklerin daęıtımını denetlemeye karar verebilirsiniz. Kayıtların daęıtımını bölebilirsiniz. Kayıtların yarısının (örneęin, 00000-49999 hesap numaralarının) LONDONiinde tutulmasını ayarlayabilirsiniz. Dięer yarısı (50000-99999 aralıęında) NEWYORKiinde tutulur. Daha sonra, tüm iletilerdeki hesap alanını incelemek için bir küme iř yükü ıkıř programı yazabilir ve iletileri uygun kuyruk yöneticisine yöneltebilirsiniz.

## Sonraki adım

Tüm tanımları tamamladıktan sonra, henüz tamamlamadıysanız, kanal bařlatıcısını IBM MQ for z/OSüzerinde bařlatın. Tüm altyapılarda, TORONTOkuyruk yöneticisinde bir dinleyici programı bařlatın. Dinleyici programı gelen aę isteklerini bekler ve gerektięinde küme alıcı kanalını bařlatır.

## Var olan kümelere bir kuyruk paylařım grubu ekleme

Var olan kümelere z/OS üzerinde bir kuyruk paylařım grubu ekleyin.

## Bařlamadan önce

### Not:

1. Bir kümede yapılan deęiřikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi bařlatmadan önce havuzlarınızın kullanılabilir olduęundan emin olun.
2. Kuyruk paylařım grupları yalnızca IBM MQ for z/OSüzerinde desteklenir. Bu görev dięer platformlar için geerli deęildir.

Senaryo:

- INVENTORY kümesi, “Yeni bir küme ayarlama” sayfa 297iinde aıklandıęı gibi ayarlandı. İki kuyruk yöneticisi ierir: LONDON ve NEWYORK.
- Bu kümeye bir kuyruk paylařım grubu eklemek istiyorsunuz. Grup, QSGP, üç kuyruk yöneticilerinden oluşur: P1, P2ve P3. Bunlar, P1tarafından tanımlanacak INVENTQ kuyruęunun bir örneęini paylařır.

## Bu görev hakkında

Paylařılan bir kuyruęu barındıran yeni kuyruk yöneticileri eklemek için ařaęıdaki adımları izleyin.

## Yordam

1. Kuyruk yöneticilerinin önce hangi tam havuza bařvuracaęına karar verin.

Bir kümedeki her kuyruk yöneticisi, tam havuzlardan birine ya da dięerine bařvurmalıdır. Kümeyle ilgili bilgileri tam bir havuzdan toplar ve böylece kendi kısmi havuzunu oluşturur. Bu, seçtięiniz tam havuz için özel bir önem tařımaz. Bu örnekte NEWYORKseeneęini belirleyin. Kuyruk paylařım grubu kümeye katıldıktan sonra, her iki havuzla da iletiřim kurar.

2. CLUSRCVR kanallarını tanımlayın.

Bir kümedeki her kuyruk yöneticisinin, iletileri alabileceği bir küme alıcı kanalı tanımlaması gerekir. P1, P2 ve P3 sistemlerinde aşağıdakileri tanımlayın:

```
DEFINE CHANNEL(INVENTORY.Pn) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(Pn.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for sharing queue manager')
```

Küme alıcı kanalı, INVENTORY kümesindeki diğer kuyruk yöneticilerinden ileti almak için her kuyruk yöneticisinin kullanılabilirliğini tanıtır.

3. Kuyruk paylaşım grubu için bir CLUSSDR kanalı tanımlayın.

Bir kümenin her üyesinin, ilk tam havuzuna ileti gönderebildiği bir küme-gönderen kanalı tanımlaması gerekir. Bu durumda NEWYORK'ı seçtik. Kuyruk paylaşım grubundaki kuyruk yöneticilerinden biri için aşağıdaki grup tanımlaması gerekir. Tanım, her kuyruk yöneticisinin bir küme gönderen kanal tanımına sahip olmasını sağlar.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) QSGDISP(GROUP)
DESCR('Cluster-sender channel to repository at NEWYORK')
```

4. Paylaşılan kuyruğu tanımlayın.

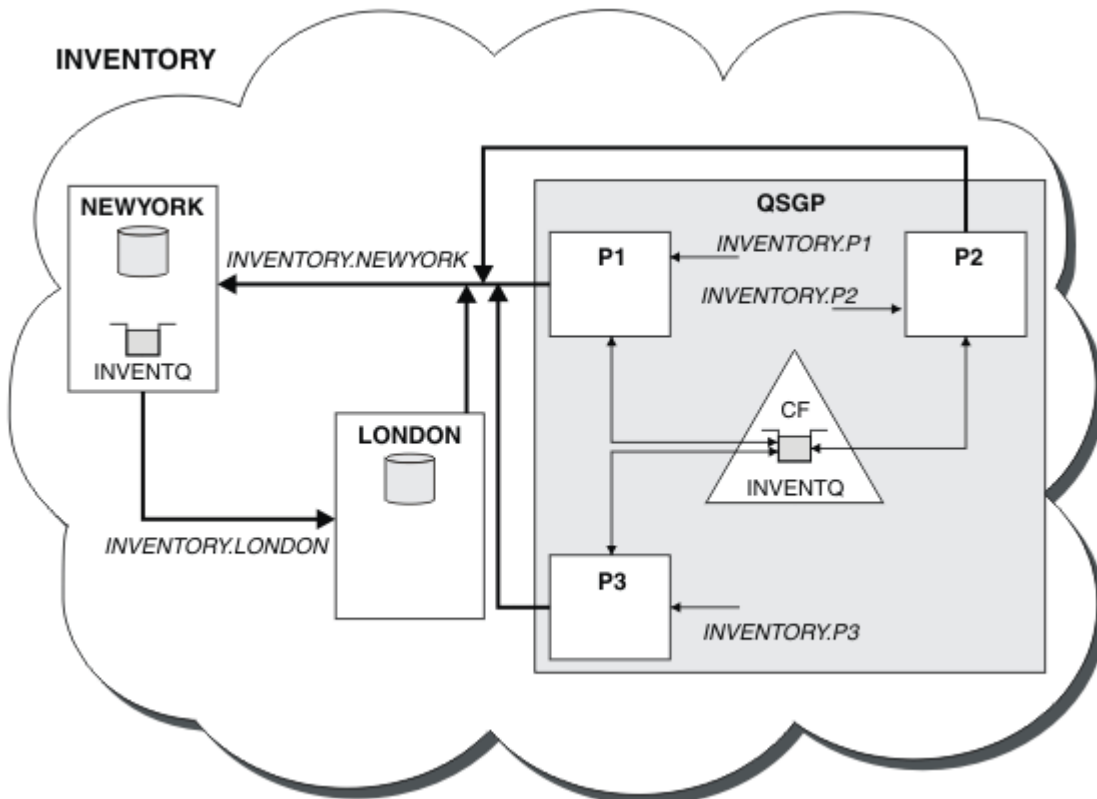
P1 üzerinde INVENTQ kuyruğunu aşağıdaki gibi tanımlayın:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
```

Yeni kuyruk yöneticisinde kanal başlatıcıyı ve bir dinleyici programını başlatın. Dinleyici programı gelen ağ isteklerini dinler ve gerektiğinde küme-alıcı kanalını başlatır.

## Sonuçlar

Şekil 43 sayfa 318, bu görev tarafından ayarlanan kümeyi gösterir.



Şekil 43. Küme ve kuyruk paylaşım grubu

Şimdi INVENTQ tarafından LONDON kuyruğuna konan iletiler, kuyruğu barındıran dört kuyruk yöneticisine dönüşümlü olarak yönlendirilir.

## Sonraki adım

Bir kuyruk paylaşım grubu anasisteminin üyelerinin bir küme kuyruğuna sahip olması, grubun herhangi bir üyesinin bir isteği yanıtlayabilmesinden yarar sağlar. Bu durumda, paylaşılan kuyrukta bir ileti alındıktan sonra P1 kullanılmaz duruma gelir. Kuyruk paylaşım grubunun başka bir üyesi bunun yerine yanıt verebilir.

## Tam havuzun başka bir kuyruk yöneticisine taşınması

Yeni havuzu ikinci havuzda tutulan bilgilerden oluşturarak bir tam havuzu bir kuyruk yöneticisinden diğerine taşıyın.

## Başlamadan önce

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- INVENTORY kümesi, [“Kümeye kuyruk yöneticisi eklenmesi” sayfa 308](#) içinde açıklandığı gibi ayarlandı.
- İş nedenleriyle, LONDON kuyruk yöneticisinden tüm havuzu kaldırmak ve PARIS kuyruk yöneticisinde tam bir havuzla değiştirmek istiyorsunuz. NEWYORK kuyruk yöneticisi, tam bir havuzu tutmaya devam edecektir.

## Bu görev hakkında

Bir havuzu başka bir kuyruk yöneticisine taşımak için aşağıdaki adımları izleyin.

## Yordam

1. PARIS ' u tam havuz kuyruğu yöneticisi yapacak şekilde değiştirin.

PARIS' da şu komutu verin:

```
ALTER QMGR REPOS(INVENTORY)
```

2. PARIS üzerinde CLUSSDR kanalı ekleme

PARIS 'in şu anda LONDON' u gösteren bir küme gönderen kanalı var. LONDON artık küme için tam havuz tutmaz. PARIS 'in, diğer tam havuzun tutulduğu NEWYORK' u gösteren yeni bir küme gönderen kanalı olmalıdır.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at NEWYORK')
```

3. NEWYORK üzerinde PARIS adresini gösteren bir CLUSSDR kanalı tanımlayın

Şu anda NEWYORK , LONDON ögesini gösteren bir küme gönderen kanalına sahiptir. Diğer tam havuz PARIS konumuna taşındıktan sonra, NEWYORK adresindeki PARIS adresini gösteren yeni bir küme gönderen kanal eklemeniz gerekir.

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from NEWYORK to repository at PARIS')
```

Kümüye gönderen kanalı PARIS'e eklediğinizde, PARIS kümeyi NEWYORK' den öğrenir. NEWYORK içindeki bilgileri kullanarak kendi tam havuzunu oluşturur.

#### 4. PARIS kuyruk yöneticisinin artık tam bir havuz olup olmadığını denetleyin

PARIS kuyruk yöneticisinin, NEWYORK kuyruk yöneticisindeki tam havuzdan kendi tam havuzunu oluşturduğunu doğrulayın. Aşağıdaki komutları verin:

```
DIS QCLUSTER(*) CLUSTER (INVENTORY)
DIS CLUSQMGR(*) CLUSTER (INVENTORY)
```

Bu komutların, bu kümedeki NEWYORK ile aynı kaynakların ayrıntılarını gösterdiğini denetleyin.

**Not:** NEWYORK kuyruk yöneticisi yoksa, bu bilgi oluşturma işlemi tamamlanamaz. Görev tamamlanıncaya kadar sonraki adıma geçmeyin.

#### 5. LONDON üzerinde kuyruk yöneticisi tanımlamasını değiştir

Son olarak, LONDON konumundaki kuyruk yöneticisini, küme için artık tam havuz bulundurmayacak şekilde değiştirin. LONDON' da şu komutu verin:

```
ALTER QMGR REPOS(' ')
```

Kuyruk yöneticisi artık herhangi bir küme bilgisi almıyor. 30 gün sonra, tam havuzda saklanan bilgilerin süresi dolar. LONDON kuyruk yöneticisi şimdi kendi kısmi havuzunu oluşturur.

#### 6. Bekleyen tanımlamaları kaldırın ya da değiştirin.

Kümenin yeni düzeninin beklendiği gibi çalıştığından emin olduğunuzda, artık doğru olmayan el ile tanımlanan CLUSSDR tanımlarını kaldırın ya da değiştirin.

- PARIS kuyruk yöneticisinde, kümenin otomatik kanalları yeniden kullanabilmesi için kümeyi gönderen kanalı LONDON' e durdurup silmeniz ve kanal başlatma komutunu çalıştırmanız gerekir:

```
STOP CHANNEL (INVENTORY.LONDON)
DELETE CHANNEL (INVENTORY.LONDON)
START CHANNEL (INVENTORY.LONDON)
```

- NEWYORK kuyruk yöneticisinde, kümenin otomatik kanalları yeniden kullanabilmesi için kümeyi gönderen kanalı LONDON' e durdurup silmeniz ve kanal başlatma komutunu çalıştırmanız gerekir:

```
STOP CHANNEL (INVENTORY.LONDON)
DELETE CHANNEL (INVENTORY.LONDON)
START CHANNEL (INVENTORY.LONDON)
```

- Kümedeki tüm kuyruk yöneticilerindeki LONDON 'e işaret eden diğer el ile tanımlanan küme gönderen kanallarını NEWYORK ya da PARIS' yi işaret eden kanallarla değiştirin. Bir kanalı sildikten sonra, kümenin otomatik kanalları yeniden kullanabilmesini sağlamak için her zaman **start channel** komutunu verin. Bu küçük örnekte, başka kimse yok. Unutup unutmadığınızı denetlemek için, her kuyruk yöneticisinden TYPE (CLUSSDR) belirterek DISPLAY CHANNEL komutunu verin. Örneğin:

```
DISPLAY CHANNEL(*) TYPE (CLUSSDR)
```

Bu görevi, tam havuzu LONDON 'den PARIS' e taşıdıktan sonra mümkün olan en kısa sürede gerçekleştirmeniz önemlidir. Bu görevi gerçekleştirmeden önce, INVENTORY . LONDON adlı CLUSSDR kanallarını el ile tanımlayan kuyruk yöneticileri, bu kanalı kullanarak bilgi istekleri gönderebilir.

LONDON tam havuz olmaktan çıktıktan sonra, bu tür istekleri alırsa, hata iletilerini kuyruk yöneticisi hata günlüğüne yazar. Aşağıdaki örneklerde, LONDON' da hangi hata iletilerinin görülebileceği gösterilmektedir:

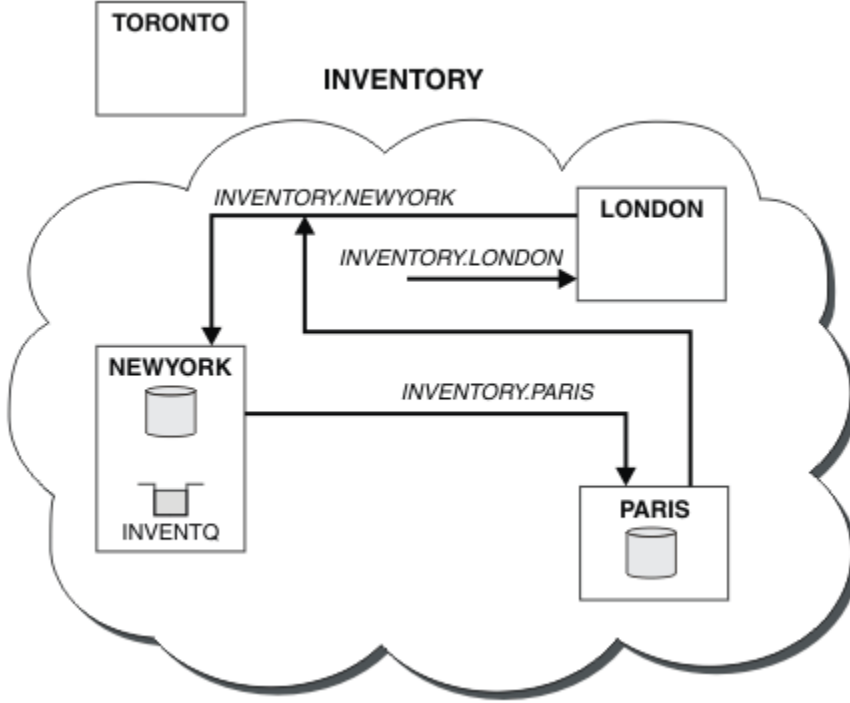
- AMQ9428: Unexpected publication of a cluster queue object received
- AMQ9432: Query received by a non-repository queue manager

LONDON kuyruk yöneticisi, artık tam havuz olmadığı için bilgi isteklerine yanıt vermiyor. LONDON 'den bilgi isteyen kuyruk yöneticileri, el ile tanımlanan CLUSSDR tanımlamaları PARIS ögesini gösterecek

şekilde düzeltilinceye kadar küme bilgileri için NEWYORK ' e güvenmelidir. Bu durum, uzun vadede geçerli bir yapılandırma olarak kabul edilmemelidir.

## Sonuçlar

Şekil 44 sayfa 321 , bu görev tarafından ayarlanan kümeyi gösterir.



Şekil 44. Tam havuzun PARIS konumuna taşındığı INVENTORY kümesi

## Var olan bir ağı kümeye dönüştürme

Var olan bir dağıtılmış kuyruğa alma ağını bir kümeye dönüştürün ve kapasiteyi artırmak için ek bir kuyruk yöneticisi ekleyin.

## Başlamadan önce

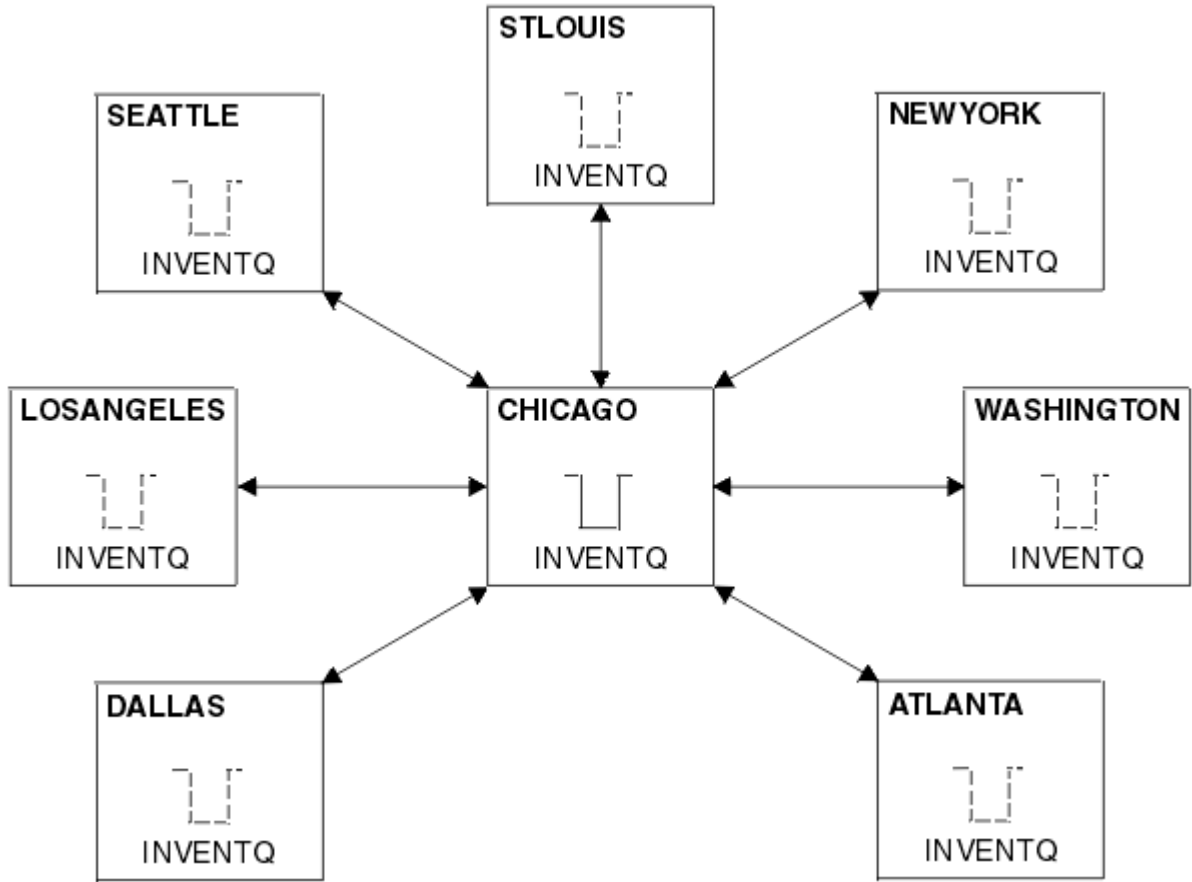
“Yeni bir küme ayarlama” sayfa 297 ile “Tam havuzun başka bir kuyruk yöneticisine taşınması” sayfa 319 arasında yeni bir küme oluşturup genişletmiş bulundunuz. Sonraki iki görev farklı bir yaklaşımı keşfedecek: var olan bir kuyruk yöneticisi ağını kümeye dönüştürme.

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- Bir zincir mağazanın ülke çapındaki şubelerini birbirine bağlayan bir IBM MQ ağı zaten var. Bir göbek ve konuşma yapısına sahiptir: tüm kuyruk yöneticileri bir merkezi kuyruk yöneticisine bağlıdır. Merkezi kuyruk yöneticisi, döküm uygulamasının çalıştığı sistemde bulunur. Uygulama, her kuyruk yöneticisinin bir uzak kuyruk tanımlamasına sahip olduğu INVENTQ kuyruğundaki iletilerin gelişiyile yönlendirilir.

Bu ağı Şekil 45 sayfa 322’de gösterilmektedir.



Şekil 45. Bir merkez ve konuşma ağı

- Yönetimi kolaylaştırmak için bu ağı bir kümeye dönüştürecek ve iş yükünü paylaşmak üzere merkezi yerde başka bir kuyruk yöneticisi yaratacaksınız.

Küme adı: CHNSTORE.

**Not:** Küme adı CHNSTORE , küme alıcı kanal adlarının *cluster\_name . queue\_manager\_name* biçiminde 20 karakter üst sınırını aşmayan adlar kullanılarak yaratılmasına izin vermek üzere seçildi; örneğin, CHNSTORE . WASHINGTON.

- Her iki merkezi kuyruk yöneticisi de tam havuzları barındıracak ve döküm uygulaması tarafından erişilebilir olacaktır.
- Döküm uygulaması, merkezi kuyruk yöneticilerinden birinin barındırdığı INVENTQ kuyruğundaki iletilerin gelişle yönlendirilir.
- Döküm uygulaması, koştur olarak çalışan ve birden çok kuyruk yöneticisi tarafından erişilebilen tek uygulama olacak. Diğer tüm uygulamalar önceki gibi çalışmaya devam eder.
- Tüm dalların iki merkezi kuyruk yöneticisine ağ bağlantısı vardır.
- Ağ protokolü TCP 'dir.

### Bu görev hakkında

Var olan bir ağı kümeye dönüştürmek için aşağıdaki adımları izleyin.

#### Yordam

1. İleti yakınlıkları için stok uygulamasını gözden geçirin.

Devam etmeden önce, uygulamanın ileti yakınlıkları işleyebildiğinden emin olun. İleti yakınlıkları, iletilerin belirli bir kuyruk yöneticisi ya da belirli bir sıra tarafından işlenmesi gereken iki uygulama

arasında deęiş tokuř edilen etkileřimli iletiler arasındaki iliřkidir. İleti yakınlıkları hakkında daha fazla bilgi için bkz: “İleti yakınlıkları iřlenmektedir” sayfa 398

2. İki merkezi kuyruk yöneticisini, bunları tam havuz kuyruęu yöneticileri yapacak řekilde deęiřtirin.

İki kuyruk yöneticisi CHICAGO ve CHICAGO2 bu aęın göbeęindedirler. Zincir maęaza kümesiyle iliřkili tüm etkinlikleri bu iki kuyruk yöneticisine odaklamaya karar verdiniz. Döküm uygulamasının ve INVENTQ kuyruęuna iliřkin tanımlamaların yanı sıra, bu kuyruk yöneticilerinin kümeye iliřkin iki tam havuzu bulundurmalarını da istiyorsunuz. İki kuyruk yöneticisinin her birinde řu komutu verin:

```
ALTER QMGR REPOS(CHNSTORE)
```

3. Her kuyruk yöneticisinde bir CLUSRCVR kanalı tanımlayın.

Kümedeki her kuyruk yöneticisinde, bir küme alıcı kanalı ve bir küme gönderen kanalı tanımlayın. Önce hangi kanalı tanımladıęınız önemli deęil.

Her kuyruk yöneticisini, aę adresini ve dięer bilgileri kümeye tanıtmak için bir CLUSRCVR tanımlaması yapın. Örneęin, ATLANTA kuyruk yöneticisinde:

```
DEFINE CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(ATLANTA.CHSTORE.COM) CLUSTER(CHNSTORE)
DESCR('Cluster-receiver channel')
```

4. Her kuyruk yöneticisinde bir CLUSSDR kanalı tanımlayın

Her kuyruk yöneticisinde, o kuyruk yöneticisini tam havuz kuyruęu yöneticilerinden birine ya da dięerine baęlamak için bir CLUSSDR tanımlaması yapın. Örneęin, ATLANTA ile CHICAGO2 arasında baęlantı oluřturabilirsiniz:

```
DEFINE CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO2.CHSTORE.COM) CLUSTER(CHNSTORE)
DESCR('Cluster-sender channel to repository queue manager')
```

5. Stok uygulamasını CHICAGO2 üzerine kurun.

CHICAGO kuyruk yöneticisinde döküm uygulaması zaten var. řimdi CHICAGO2 kuyruk yöneticisinde bu uygulamanın bir kopyasını oluřturmanız gerekir.

6. Merkezi kuyruk yöneticilerindeki INVENTQ kuyruęunu tanımlayın.

CHICAGO' ta, INVENTQ kuyruęunun yerel kuyruk tanımlamasını deęiřtirerek, kuyruęun kümede kullanılabilir olmasını saęlayın. řu komutu verin:

```
ALTER QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
```

CHICAGO2' da, aynı kuyruk için bir tanımlama yapın:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
```

z/OS iřletim sistemlerinde, CHICAGO üzerindeki INVENTQ 'un CHICAGO2 üzerinde tam bir kopyasını oluřturmak için **CSQUTIL** ' in COMMAND iřlevinin MAKEDEF seęeneęini kullanabilirsiniz.

Bu tanımlamaları yaptıęınızda, CHICAGO ve CHICAGO2 adresindeki tam havuzlara bir ileti gönderilir ve bu havuzlardaki bilgiler güncellenir. Kuyruk yöneticisi, INVENTQ' e bir ileti gönderdięinde, tüm havuzlardan iletilere iliřkin bir hedef seęimi olduęunu öęrenir.

7. Küme deęiřikliklerinin yayılıp yayılmadıęını denetleyin.

Önceki adımda yarattıęınız tanımlamaların kümede yayılıp yayılmadıęını denetleyin. Tam havuz kuyruęu yöneticisinde řu komutu verin:

```
DIS QCLUSTER(INVENTQ)
```

## Yeni, birbirine bağı bir küme eklenmesi

Bazı kuyruk yöneticilerini var olan bir kümeyle paylaşan yeni bir küme ekleyin.

### Başlamadan önce

#### Not:

1. Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.
2. Bu görevi başlatmadan önce, kuyruk adı çakışmalarını denetleyin ve sonuçları anlayın. Devam etmeden önce bir kuyruğu yeniden adlandırmanız ya da kuyruk diğer adlarını ayarlamanız gerekebilir.

Senaryo:

- Bir IBM MQ kümesi, “Var olan bir ağı kümeye dönüştürme” sayfa 321’inde açıklandığı gibi ayarlandı.
- MAILORDER adlı yeni bir küme uygulanacak. Bu küme, CHNSTORE kümesinde bulunan dört kuyruk yöneticisinden oluşur: CHICAGO, CHICAGO2, SEATTLE ve ATLANTA ve iki ek kuyruk yöneticisi; HARTFORD ve OMAHA. MAILORDER uygulaması, OMAHA kuyruk yöneticisine bağı Omaha 'da sistemde çalışır. Kümedeki diğer kuyruk yöneticileri tarafından yönlendirilir ve iletileri MORDERQ kuyruğuna yerleştirir.
- MAILORDER kümesine ilişkin tüm havuzlar iki kuyruk yöneticisinde CHICAGO ve CHICAGO2 tutulur.
- Ağ protokolü TCP 'dir.

### Bu görev hakkında

Yeni, birbirine bağı bir küme eklemek için aşağıdaki adımları izleyin.

#### Yordam

1. Küme adlarından oluşan bir ad listesi yaratın.

CHICAGO ve CHICAGO2 konumundaki tam havuz kuyruğu yöneticileri artık her iki küme için de CHNSTORE ve MAILORDER tam havuzlarını tutacak. Önce, kümelerin adlarını içeren bir ad listesi oluşturun. CHICAGO ve CHICAGO2 üzerinde ad melist değerini aşağıdaki gibi tanımlayın:

```
DEFINE NAMELIST(CHAINMAIL)
DESCR('List of cluster names')
NAMES(CHNSTORE, MAILORDER)
```

2. İki kuyruk yöneticisi tanımlamasını değiştirin.

Şimdi CHICAGO ve CHICAGO2 adresindeki iki kuyruk yöneticisi tanımlamasını değiştirin. Şu anda bu tanımlamalar, kuyruk yöneticilerinin CHNSTORE kümesi için tam havuzları tuttukları gösteriyor. Bu tanımlamayı değiştirerek, kuyruk yöneticilerinin CHAINMAIL ad listesinde listelenen tüm kümeler için tam havuzları tutmasını sağlayın. CHICAGO ve CHICAGO2 kuyruk yöneticisi tanımlamalarını değiştirin:

```
ALTER QMGR REPOS(' ') REPOSNL(CHAINMAIL)
```

3. CHICAGO ve CHICAGO2 üzerinde CLUSRCVR kanallarını değiştirin.

CHICAGO ve CHICAGO2 adresindeki CLUSRCVR kanal tanımları, kanalların kümede kullanılabilir olduğunu gösterir CHNSTORE. Kanalların CHAINMAIL ad listesinde yer alan tüm kümeler için kullanılabilir olduğunu göstermek üzere küme alıcı tanımlamasını değiştirmeniz gerekir. CHICAGO adresindeki küme günlük nesnesi tanımlamasını değiştirin:

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```



CHICAGO2' da şu komutu girin:

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

4. CHICAGO ve CHICAGO2 üzerinde CLUSSDR kanallarını değiştirin.

Ad listesi eklemek için iki CLUSSDR kanal tanımını değiştirin. CHICAGO' da şu komutu girin:

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

CHICAGO2' da şu komutu girin:

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

5. SEATTLE ve ATLANTA üzerinde bir ad listesi oluşturun.

SEATTLE ve ATLANTA birden çok kümenin üyesi olacağından, kümelerin adlarını içeren bir ad listesi oluşturmanız gerekir. SEATTLE ve ATLANTA üzerinde ad melist değerini aşağıdaki gibi tanımlayın:

```
DEFINE NAMELIST(CHAINMAIL)
DESCR('List of cluster names')
NAMES(CHNSTORE, MAILORDER)
```

6. SEATTLE ve ATLANTA üzerinde CLUSRCVR kanallarını değiştirin.

SEATTLE ve ATLANTA adresindeki CLUSRCVR kanal tanımları, kanalların kümede kullanılabilir olduğunu gösterir CHNSTORE. CHAINMAIL ad listesinde yer alan tüm kümelerde kanalların kullanılabilir olduğunu göstermek için küme alma kanalı tanımlarını değiştirin. SEATTLE' da şu komutu girin:

```
ALTER CHANNEL(CHNSTORE.SEATTLE) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

ATLANTA' da şu komutu girin:

```
ALTER CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

7. SEATTLE ve ATLANTA üzerinde CLUSSDR kanallarını değiştirin.

Ad listesi eklemek için iki CLUSSDR kanal tanımını değiştirin. SEATTLE' da şu komutu girin:

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

ATLANTA' da şu komutu girin:

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

8. HARTFORD ve OMAHA üzerinde CLUSRCVR ve CLUSSDR kanallarını tanımlayın.

İki yeni kuyruk yöneticisinde HARTFORD ve OMAHA, küme alıcısı ve küme gönderen kanallarını tanımlayın. Tanımları hangi sırada oluşturacağınız önemli değildir. HARTFORD' da şunu girin:

```
DEFINE CHANNEL(MAILORDER.HARTFORD) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(HARTFORD.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for HARTFORD')
```

```
DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from HARTFORD to repository at CHICAGO')
```

OMAHA' da şunu girin:

```
DEFINE CHANNEL(MAILORDER.OMAHA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(OMAHA.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for OMAHA')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from OMAHA to repository at CHICAGO')
```

9. OMAHA üzerinde MORDERQ kuyruğunu tanımlayın.

Bu görevi tamamlamak için son adım, kuyruk yöneticisinde MORDERQ kuyruğunu tanımlamaktır OMAHA. OMAHA' da şunu girin:

```
DEFINE QLOCAL(MORDERQ) CLUSTER(MAILORDER)
```

10. Küme değişikliklerinin yayılıp yayılmadığını denetleyin.

Önceki adımlarla yarattığınız tanımlamaların kümede yayılıp yayılmadığını denetleyin. Tam havuz kuyruğu yöneticisinde aşağıdaki komutları verin:

```
DIS QCLUSTER (MORDERQ)
DIS CLUSQMGR
```

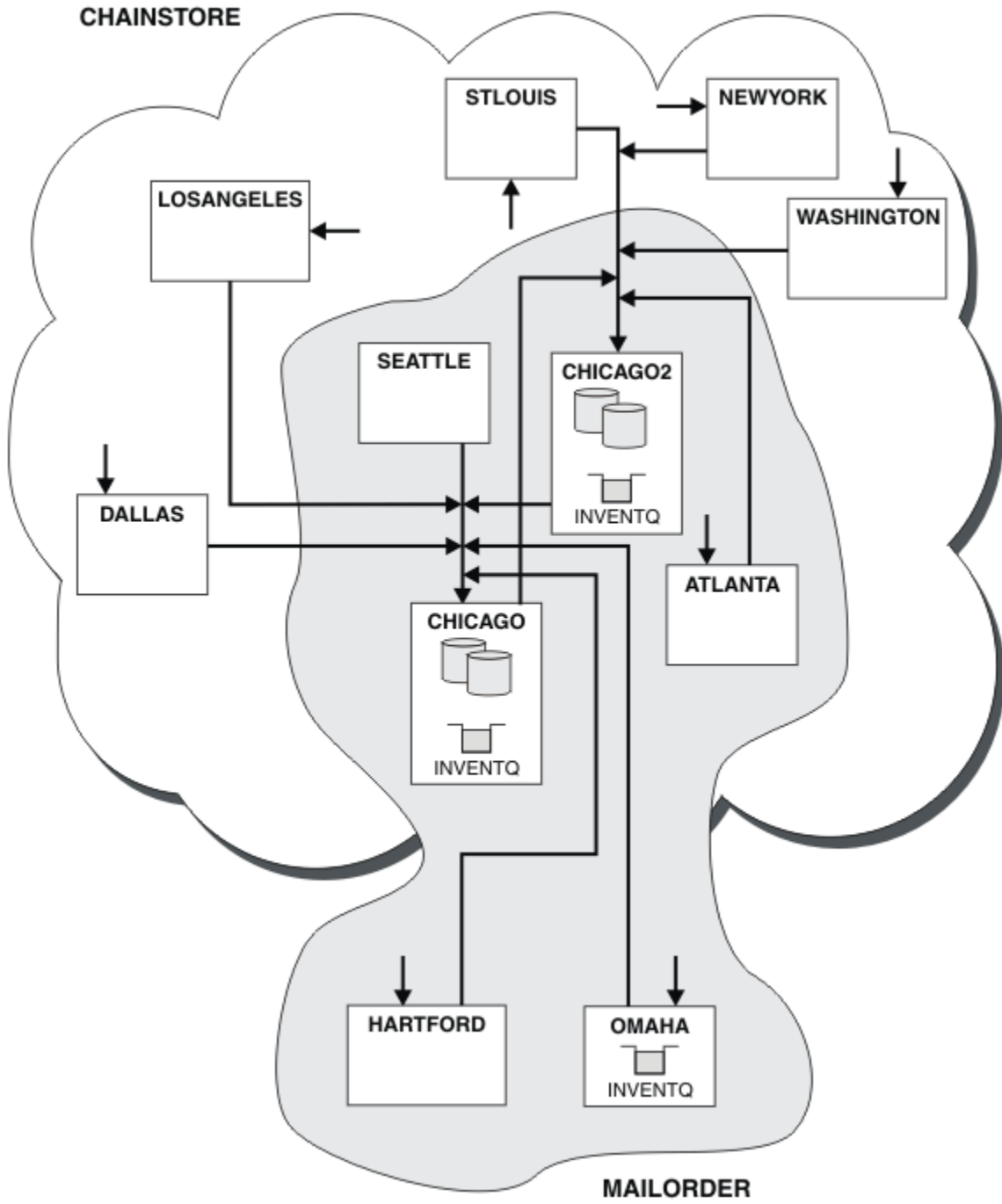
11.

## Sonuçlar

Bu görev tarafından ayarlanan küme [Şekil 46 sayfa 327](#) içinde gösterilir.

Şimdi örtüşen iki kümemiz var. Her iki kümeye ilişkin tam havuzlar CHICAGO ve CHICAGO2 adresinde tutulur. OMAHA üzerinde çalışan posta siparişi uygulaması, CHICAGO adresinde çalışan stok uygulamasından bağımsızdır. Ancak, CHNSTORE kümesindeki bazı kuyruk yöneticileri de MAILORDER kümesinde yer alır ve bu nedenle her iki uygulamaya da ileti gönderebilirler. Bu görevi iki kümeyi çakışacak şekilde gerçekleştirmeden önce, kuyruk adı çakışmaları olasılığını göz önünde bulundurun.

CHNSTORE kümesinde NEWYORK ve MAILORDER kümesinde OMAHA üzerinde ACCOUNTQ adlı bir kuyruk olduğunu varsayın. Kümelerle örtüşür ve SEATTLE üzerindeki bir uygulama kuyruğa bir ileti yerleştirirse ACCOUNTQ, ileti ACCOUNTQ eşgörünümünden herhangi birine gidebilir.



Şekil 46. Birbirine bağlı kümeler

### Sonraki adım

MAILORDER kümesini, CHNSTORE adlı büyük bir küme oluşturmak için CHNSTORE kümesiyle birleştirmeye karar verdiğiniz varsayın.

MAILORDER kümesini, CHICAGO ve CHICAGO2 tüm havuzları tutacak şekilde CHNSTORE kümesiyle birleştirmek için:

- CHICAGO ve CHICAGO2 için kuyruk yöneticisi tanımlamalarını değiştirin, ad değerini ( CHAINMAIL) belirten REPOSNL özniteliğini kaldırın ve küme adını ( CHNSTORE) belirten bir REPOS özniteliğiyle değiştirin. Örneğin:

```
ALTER QMGR(CHICAGO) REPOSNL(' ') REPOS(CHNSTORE)
```

- MAILORDER kümesindeki her kuyruk yöneticisinde, CLUSTER özniteliğinin değerini MAILORDER değerinden CHNSTORE değerine değiştirmek için tüm kanal tanımlamalarını ve kuyruk tanımlamalarını değiştirin. Örneğin, HARTFORD alanına şunu girin:

```
ALTER CHANNEL(MAILORDER.HARTFORD) CLUSTER(CHNSTORE)
```

OMAHA ' da şunu girin:

```
ALTER QLOCAL(MORDERQ) CLUSTER(CHNSTORE)
```

- Küme ad listesi CHAINMAIL (yani, CHICAGO, CHICAGO2, SEATTLE ve ATLANTA adresindeki CLUSRCVR ve CLUSSDR kanal tanımlarını, bunun yerine kümeyi belirtmek için değiştirin CHNSTORE.

Bu örnekte, ad listesi kullanma avantajını görebilirsiniz. CHICAGO ve CHICAGO2 için kuyruk yöneticisi tanımlamalarını değiştirmek yerine, ad listesi değerini değiştirebilirsiniz CHAINMAIL. Benzer şekilde, CHICAGO, CHICAGO2, SEATTLE ve ATLANTA adresindeki CLUSRCVR ve CLUSSDR kanal tanımlarını değiştirmek yerine, ad bilgilerini değiştirerek gerekli sonucu elde edebilirsiniz.

### İlgili görevler

Küme ağının kaldırılması

Bir kümeyi ağdan kaldırın ve dağıtılmış kuyruğa alma yapılandırmasını geri yükleyin.

### ***Küme ağının kaldırılması***

Bir kümeyi ağdan kaldırın ve dağıtılmış kuyruğa alma yapılandırmasını geri yükleyin.

## Başlamadan önce

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- Bir IBM MQ kümesi, [“Var olan bir ağı kümeye dönüştürme” sayfa 321](#) içinde açıklandığı gibi ayarlandı.
- Bu küme şimdi sistemden kaldırılacak. Kuyruk yöneticileri ağı, küme uygulanmadan önce olduğu gibi çalışmaya devam edecektir.

## Bu görev hakkında

Bir küme ağını kaldırmak için aşağıdaki adımları izleyin.

### Yordam

1. CHNSTORE kümesinden küme kuyruklarını kaldırın.

CHICAGO ve CHICAGO2 sistemlerinde, kuyruğu kümeden kaldırmak için INVENTQ kuyruğuna ilişkin yerel kuyruk tanımlamasını değiştirin. Şu komutu verin:

```
ALTER QLOCAL(INVENTQ) CLUSTER(' ')
```

Kuyruğu değiştirdiğinizde, tüm havuzlardaki bilgiler güncellenir ve kümede yayılır. MQ00\_BIND\_NOT\_FIXED kullanan etkin uygulamalar ve kuyruğun DEFBIND (NOTFIXED) ile tanımlandığı MQ00\_BIND\_AS\_Q\_DEF kullanan uygulamalar, bir sonraki denenen MQPUT ya da MQPUT1 çağrısında başarısız olur. Neden kodu MQRC\_UNKOWNN\_OBJECT\_NAME döndürülür.

Önce 1. adımı gerçekleştirmeniz gerekmez, ancak gerçekleştirmezseniz 4. Adımdan sonra gerçekleştirin.

2. Küme kuyruğuna erişimi olan tüm uygulamaları durdurun.

Küme kuyruklarına erişimi olan tüm uygulamaları durdurun. Bunu yapmazsanız, 5. Adımda kümeyi yenilediğinizde bazı küme bilgileri yerel kuyruk yöneticisinde kalabilir. Tüm uygulamalar durduğunda ve küme kanallarının bağlantısı kesildiğinde bu bilgiler kaldırılır.

3. Havuz özniteliğini tam havuz kuyruğu yöneticilerinden kaldırın.

CHICAGO ve CHICAGO2 üzerinde, havuz özniteliğini kaldırmak için kuyruk yöneticisi tanımlamalarını değiştirin. Bunu yapmak için şu komutu verin:

```
ALTER QMGR REPOS(' ')
```

Kuyruk yöneticileri, kümedeki diğer kuyruk yöneticilerine artık tam havuzları tutmadıklarını bildirir. Diğer kuyruk yöneticileri bu bilgileri aldığında, havuzun tamamının sona erdiğini belirten bir ileti görürsünüz. CHNSTORE kümesi için artık kullanılabilir havuz olmadığını belirten bir ya da daha çok ileti de görürsünüz.

4. Küme kanallarını kaldırın.

CHICAGO ' de küme kanallarını kaldırın:

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) CLUSTER(' ')
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR) CLUSTER(' ')
```

**Not:** Önce CLUSSDR komutunu, sonra CLUSRCVR komutunu yayınlamak önemlidir. Önce CLUSRCVR komutunu, daha sonra CLUSSDR komutunu vermeyin. Bunu yaparak, DURDURULDU durumuna sahip belirsiz kanallar yaratılır. Daha sonra, durdurulan kanalları kurtarmak için bir START CHANNEL komutu vermeniz gerekir; örneğin, START CHANNEL (CHNSTORE . CHICAGO).

CHNSTORE kümesi için havuz olmadığını belirten iletiler görüntülenir.

1. adımda açıklandığı gibi küme kuyruklarını kaldırmadıysanız, şimdi kaldırın.

5. Küme kanallarını durdurun.

CHICAGO ' da aşağıdaki komutlarla küme kanallarını durdurun:

```
STOP CHANNEL(CHNSTORE.CHICAGO2)
STOP CHANNEL(CHNSTORE.CHICAGO)
```

6. Kümedeki her kuyruk yöneticisi için 4. ve 5. adımları yineleyin.

7. Küme kanallarını durdurun ve küme kanallarına ve küme kuyruklarına ilişkin tüm tanımlamaları her kuyruk yöneticisinden kaldırın.

8. İsteğe bağlı: Kuyruk yöneticisi tarafından tutulan önbelleğe alınan küme bilgilerini temizleyin.

Kuyruk yöneticileri artık kümenin üyesi olmamalarına rağmen, her biri kümeyle ilgili bilgilerin önbelleğe alınmış bir kopyasını saklar. Bu verileri kaldırmak istiyorsanız bkz. [“Bir kuyruk yöneticisinin küme öncesi durumuna geri yüklenmesi” sayfa 356.](#)

9. INVENTQ uzak kuyruk tanımlamalarını başkasıyla değiştir

Ağın çalışmaya devam edebilmesi için, her kuyruk yöneticisinde INVENTQ uzak kuyruk tanımını değiştirin.

10. Kümeyi toparla.

Artık gerekli olmayan kuyruk ya da kanal tanımlarını silin.

### İlgili görevler

Yeni, birbirine bağlı bir küme eklenmesi

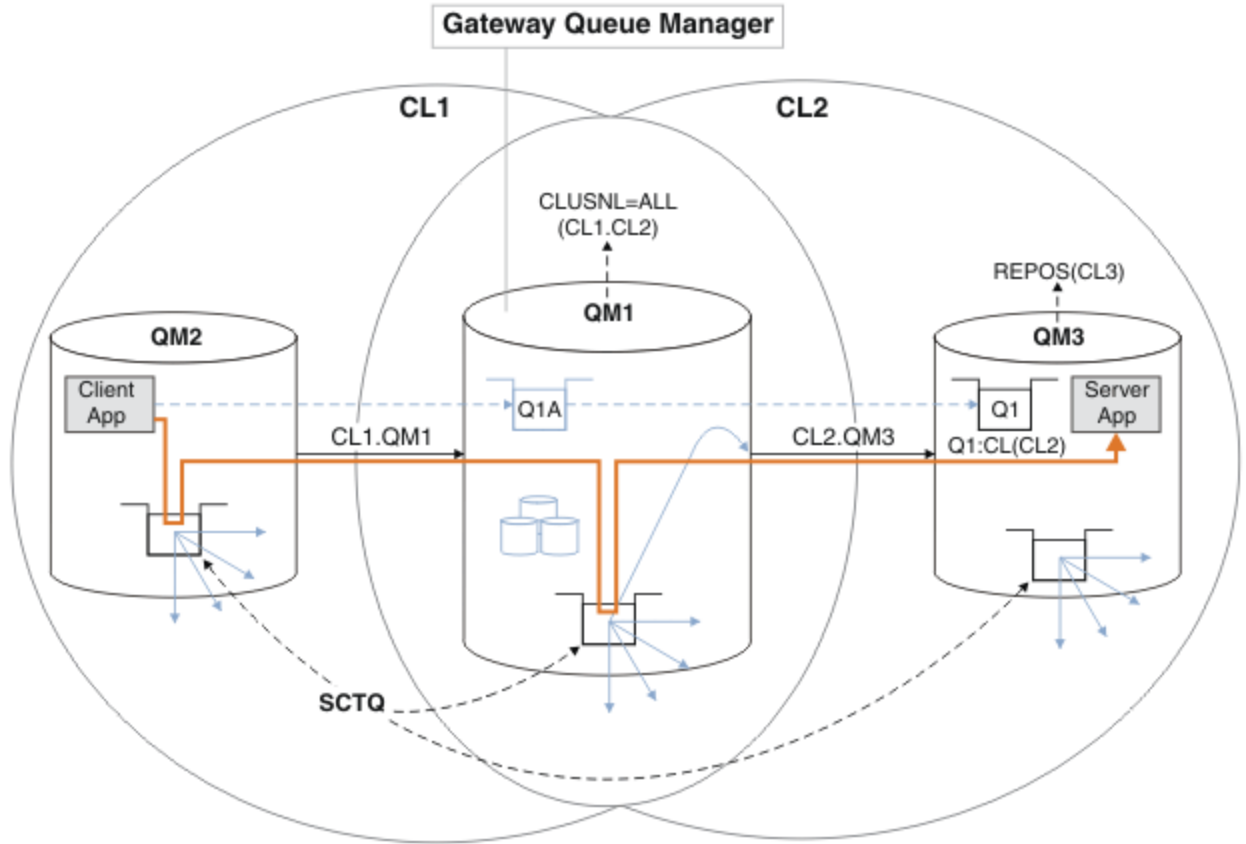
Bazı kuyruk yöneticilerini var olan bir kümeyle paylaşan yeni bir küme ekleyin.

## Ağ geçidi kuyruk yöneticisiyle çakışan iki küme oluşturma

Bir ağ geçidi kuyruk yöneticisiyle çakışan kümeler oluşturmak için görevdeki yönergeleri izleyin. Kümeleri, bir uygulamadaki iletileri bir kümedeki diğer uygulamalara iletilerden yalıtma örnekleri için başlangıç noktası olarak kullanın.

### Bu görev hakkında

Küme ileti trafiğinin yalıtılmasını göstermek için kullanılan örnek küme yapılandırması Şekil 47 sayfa 330 içinde gösterilir. Örnek, [Clustering: Application isolation using multiple cluster](#) iletim kuyrukları başlıklı konuda açıklanmıştır.



Şekil 47. IBM MQ kümelerini kullanarak merkeze konuşlandırılan ve mimariye konuşlandırılmış istemci-sunucu uygulaması

Örneği mümkün olduğunca az sayıda oluşturmak için adımların sayısını yapmak için yapılandırma gerçekçi değil, basit tutulur. Bu örnek, iki ayrı kuruluş tarafından oluşturulan iki kümenin bütünleştirilmesini temsil edebilir. Daha gerçekçi bir senaryo için bkz. [Clustering: Planning nasıl küme iletim kuyrukları yapılandırılacağını planlama](#).

Kümeleri oluşturmak için adımları izleyin. Kümeler, istemci uygulamasından sunucu uygulamasına ileti trafiğinin yalıtılmasına ilişkin aşağıdaki örneklerde kullanılır.

Bu yönergelerde, her kümenin iki havuzu olması için fazladan birkaç kuyruk yöneticisi eklenir. Ağ geçidi kuyruk yöneticisi, performans nedenleriyle havuz olarak kullanılmaz.

### Yordam

1. Kuyruk yöneticilerini QM1, QM2, QM3, QM4, QM5 oluşturun ve başlatın.

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QM n
strmqm QmgrName
```

**Not:** QM4 ve QM5 , kümeler için tam havuzları yedekler.

2. Her kuyruk yöneticisi için dinleyici tanımlayın ve başlatın.

```
*... On QM n
DEFINE LISTENER(TCP141 n) TRPTYPE(TCP) IPADDR(hostname) PORT(141 n) CONTROL(QMGR) REPLACE
START LISTENER(TCP141 n)
```

3. Tüm kümeler için bir küme adı listesi oluşturun.

```
*... On QM1
DEFINE NAMELIST(ALL) NAMES(CL1, CL2) REPLACE
```

4. CL2için QM2 ve QM4 tam havuzlarını CL1, QM3 ve QM5 tam havuzlarını yapın.

a) CL1için:

```
*... On QM2 and QM4
ALTER QMGR REPOS(CL1) DEFCLXQ(SCTQ)
```

b) CL2için:

```
*... On QM3 and QM5
ALTER QMGR REPOS(CL2) DEFCLXQ(SCTQ)
```

5. Her kuyruk yöneticisi ve küme için küme gönderen ve küme alıcı kanallarını ekleyin.

QM2, QM3, QM4 ve QM5sistemlerinde aşağıdaki komutları çalıştırın; burada *c*, *nve m* , her kuyruk yöneticisi için [Çizelge 26 sayfa 331](#) içinde gösterilen değerleri alır:

<i>Çizelge 26. Küme 1 ve 2 oluşturmak için parametre değerleri</i>			
<b>Kuyruk yöneticisi</b>	<b>Küme <i>c</i></b>	<b>Diğer havuz <i>n</i></b>	<b>Bu havuz <i>m</i></b>
QM2	1	4	2
QM4	1	2	4
QM3	2	5	3
QM5	2	3	5

```
*... On QM m
DEFINE CHANNEL(CL c.QM n) CHLTYPE(CLUSSDR) CONNAME('localhost(141 n)') CLUSTER(CL c) REPLACE
DEFINE CHANNEL(CL c.QM m) CHLTYPE(CLUSRCVR) CONNAME('localhost(141 m)') CLUSTER(CL c) REPLACE
```

6. Ağ geçidi kuyruk yöneticisini (QM1) kümelerin her birine ekleyin.

```
*... On QM1
DEFINE CHANNEL(CL1.QM2) CHLTYPE(CLUSSDR) CONNAME('localhost(1412)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL1.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL2.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL2) REPLACE
DEFINE CHANNEL(CL2.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL2) REPLACE
```

7. Q1 yerel kuyruğunu CL2kümesindeki QM3 kuyruk yöneticisine ekleyin.

```
*... On QM3
DEFINE QLOCAL(Q1) CLUSTER(CL2) REPLACE
```

8. Q1A kümelenmiş kuyruk yöneticisi diğer adını ağ geçidi kuyruk yöneticisine ekleyin.

```
*... On QM1
DEFINE QALIAS(Q1A) CLUSNL(ALL) TARGET(Q1) TARGTYPE(Queue) DEFBIND(NOTFIXED) REPLACE
```

**Not:** QM1 dışında herhangi bir kuyruk yöneticisinde kuyruk yöneticisi diğer adını kullanan uygulamalar, diğer ad kuyruğunu açtığı anda DEFBIND (NOTFIXED) belirtmelidir. **DEFBIND** , kuyruk uygulama tarafından açıldığında ileti üstbilgisindeki yöneltme bilgilerinin düzeltilip düzeltilmeyeceğini belirtir. Varsayılan değere ( OPEN) ayarlanırsa, iletiler Q1@QM1' e yöneltilir. Q1@QM1 yoktur, bu nedenle diğer kuyruk yöneticilerinden gelen iletiler gitmeyen iletiler kuyruğuna gönderilir. Kuyruk özneliğini DEFBIND (NOTFIXED) olarak ayarlayarak, varsayılan olarak **DEFBIND** kuyruk ayarına ayarlanan **amqspu**t gibi uygulamalar doğru şekilde davranır.

9. Tüm kümelenmiş kuyruk yöneticilerine ilişkin küme kuyruğu yöneticisi diğer ad tanımlamalarını ağ geçidi kuyruk yöneticisine ( QM1) ekleyin.

```
*... On QM1
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSNL(ALL) REPLACE
```

**İpucu:** Başka bir kümedeki bir kuyruk yöneticisine gönderme yapan ağ geçidi kuyruk yöneticisi aktarma iletilerindeki kuyruk yöneticisi diğer adı tanımlamaları; bkz. [Kümelenmiş kuyruk yöneticisi diğer adları](#).

## Sonraki adım

1. Kuyruk diğer adı tanımlamasını sınamak için, QM3 kuyruk diğer adı tanımlamasını Q1A kullanarak QM2 'dan Q1 ' a bir ileti gönderebilirsiniz.
- a. Bir ileti koymak için QM2 üzerinde **amqspu**t örnek programını çalıştırın.

```
C:\IBM\MQ>amqspu Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

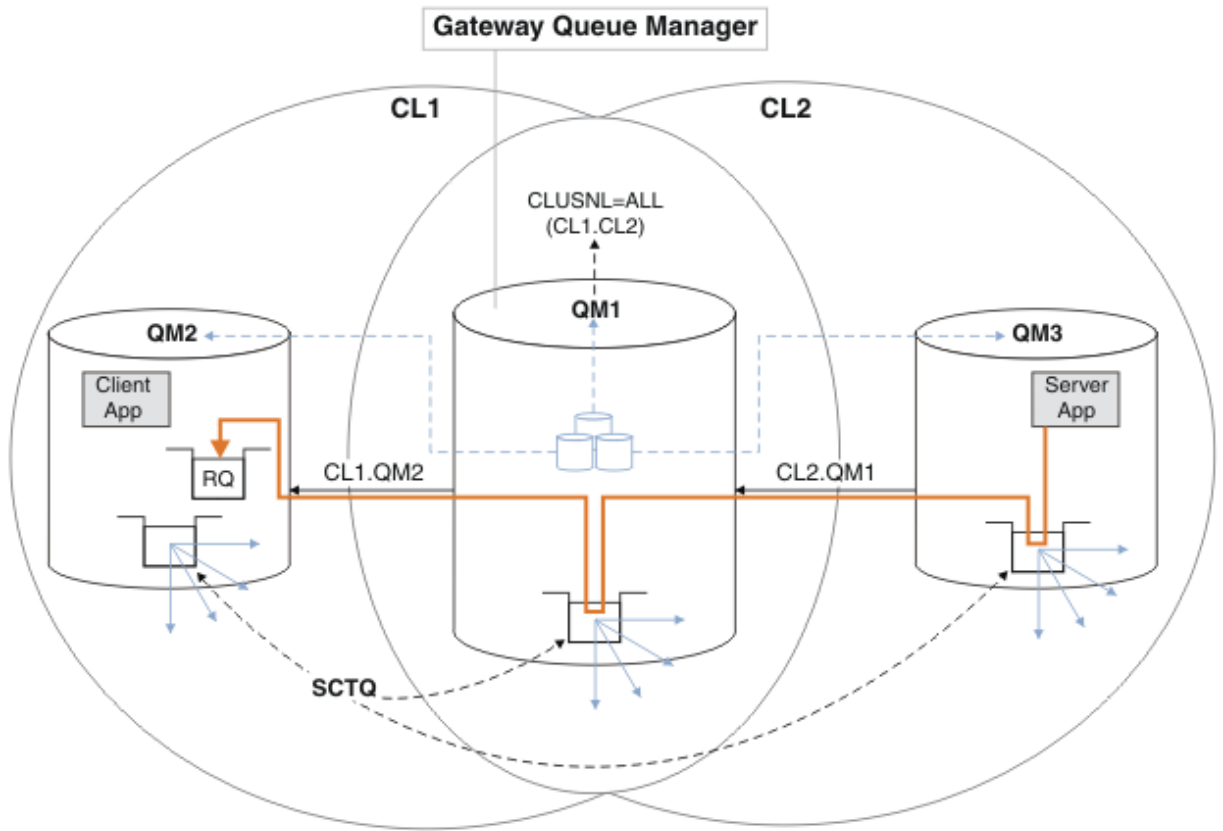
- b. İletiyi şu adresten almak için **amqsge**t örnek programını çalıştırın: Q1 on QM3

```
C:\IBM\MQ>amqsge Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

2. Bir istek iletileri göndererek ve geçici dinamik yanıt kuyruğundan bir yanıt iletileri alarak kuyruk yöneticisi diğer ad tanımlarını sınavın.

Çizge, yanıt iletileri tarafından alınan yolu, RQolarak adlandırılan geçici bir dinamik kuyruğa geri gösterir. QM3' e bağlı sunucu uygulaması, QM2 kuyruk yöneticisi adını kullanarak yanıt kuyruğunu açar. Kuyruk yöneticisi adı QM2 , QM1 üzerinde kümelenmiş kuyruk yöneticisi diğer adı olarak tanımlanır. QM3 , yanıt iletilerini QM1' e yönlendirir. QM1 , iletileri QM2' e yönlendirir.





Şekil 48. Yanıt iletisini farklı bir kümeye döndürmek için kuyruk yöneticisi diğer adının kullanılması

Yönlendirmenin çalışma şekli şu şekildedir. Her kümedeki her kuyruk yöneticisinin QM1 üzerinde bir kuyruk yöneticisi diğer adı tanımlaması vardır. Diğer adlar tüm kümelerde kümelenir. Her bir diğer addan bir kuyruk yöneticisine gelen gri çizgili oklar, her kuyruk yöneticisi diğer adının kümelerin en az birinde gerçek bir kuyruk yöneticisine çözümlendiğini gösterir. Bu durumda, QM2 diğer adı hem kümede CL1 hem de CL2 kümelenir ve CL1 içindeki QM2 gerçek kuyruk yöneticisine çözümlenir. Sunucu uygulaması, yanıt iletisini RQ kuyruk adını kullanarak ve QM2 kuyruk yöneticisine yanıt vererek yaratır. QM2 kuyruk yöneticisi diğer ad tanımlaması QM1 kümesinde CL2 ve QM2 kuyruk yöneticisinde CL2 tanımlı olmadığı için ileti QM1 olarak yöneltilir. İleti hedef kuyruk yöneticisine gönderilemediğinden, diğer ad tanımlamasına sahip kuyruk yöneticisine gönderilir.

QM1 places the message on the cluster transmission queue on QM1 for transferal to QM2. QM1, QM2 için QM1 üzerindeki kuyruk yöneticisi diğer adı tanımlaması QM2 'i gerçek hedef kuyruk yöneticisi olarak tanımladığından, iletiyi QM2 'e yöneltir. Diğer ad tanımlamaları yalnızca gerçek tanımlamalara gönderme yapabildiği için tanımlama çevrimsel değildir; diğer ad kendisini gösteremez. Hem QM1 hem de QM2 aynı kümede ( CL1) olduğundan, gerçek tanımlama QM1 tarafından çözümlenir. QM1, CL1 havuzundan QM2 ile ilgili bağlantı bilgilerini bulur ve iletiyi QM2 'e yönlendirir. İletinin QM1 tarafından yeniden yönlendirilmesi için, sunucu uygulaması yanıt kuyruğunu DEFBIND seçeneği MQBND\_BIND\_NOT\_FIXED olarak ayarlanmış şekilde açmış olmalıdır. Sunucu uygulaması yanıt kuyruğunu MQBND\_BIND\_ON\_OPEN seçeneğiyle açtıysa, ileti yeniden yönlendirilmez ve gitmeyen iletiler kuyruğuna gönderilir.

a. QM3 üzerinde bir tetikleyici ile kümelenmiş bir istek kuyruğu yaratın.

```
*... On QM3
DEFINE QLOCAL(QR) CLUSTER(CL2) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
```

b. Ağ geçidi kuyruk yöneticisinde ( QM1) QR için kümelenmiş bir kuyruk diğer adı tanımlaması yaratın.

```
*... On QM1
DEFINE QALIAS(QRA) CLUSNL(ALL) TARGET(QR) TARGTYPE(Queue) DEFBIND(NOTFIXED) REPLACE
```

c. QM3üzerinde **amqsech** örnek yankı programını başlatmak için bir süreç tanımlaması yaratın.

```
*... On QM3
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

d. Geçici dinamik yanıt kuyruğunu yaratmak için QM2 örnek program **amqsreq** için bir model kuyruğu yaratın.

```
*... On QM2
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

e. Kuyruk diğer adı tanımlamasını sınamak için, QRAkuyruk diğer adı tanımlamasını kullanarak QM2 'den QR ' e QM3 üzerinde bir istek gönderin.

i) Tetikleyici izleme programını QM3üzerinde çalıştırın.

```
runmqtrm -m QM3
```

Çıkış:

```
C:\IBM\MQ>runmqtrm -m QM3
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
01/02/2012 16:17:15: IBM MQ trigger monitor started.
```

```
-----
01/02/2012 16:17:15: Waiting for a trigger message
```

ii) İstek koymak ve yanıt beklemek için örnek programı **amqsreq** on QM2 çalıştırın.

```
C:\IBM\MQ>amqsreq QRA QM2
Sample AMQSREQ0 start
server queue is QRA
replies to 4F2961C802290020
A request message from QM2 to QR on QM3

response <A request message from QM2 to QR on QM3>
no more replies
Sample AMQSREQ0 end
```

### İlgili kavramlar

Erişim denetimi ve birden çok küme iletim kuyruğu

Kümeleme: Birden çok küme iletim kuyruğu kullanan uygulama yalıtımı

### İlgili görevler

Kümeleme: Küme iletim kuyruklarının nasıl yapılandırılacağını planlama

“Kümeyle kuyruk yöneticisi eklenmesi: ayrı iletim kuyrukları” sayfa 310

Yarattığınız kümeyle bir kuyruk yöneticisi eklemek için bu yönergeleri izleyin. Küme kuyruklarına ve konularına ilişkin iletiler, birden çok küme iletim kuyruğu kullanılarak aktarılır.

### **Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtım için uzak kuyruk tanımlaması eklenmesi**

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir

uygulamaya aktarıldıktan sonra. Çözüm, kümelenmiş bir kuyruk uzak tanımı ve ayrı bir gönderen kanalı ve iletim kuyruğunu kullanır.

## Başlamadan önce

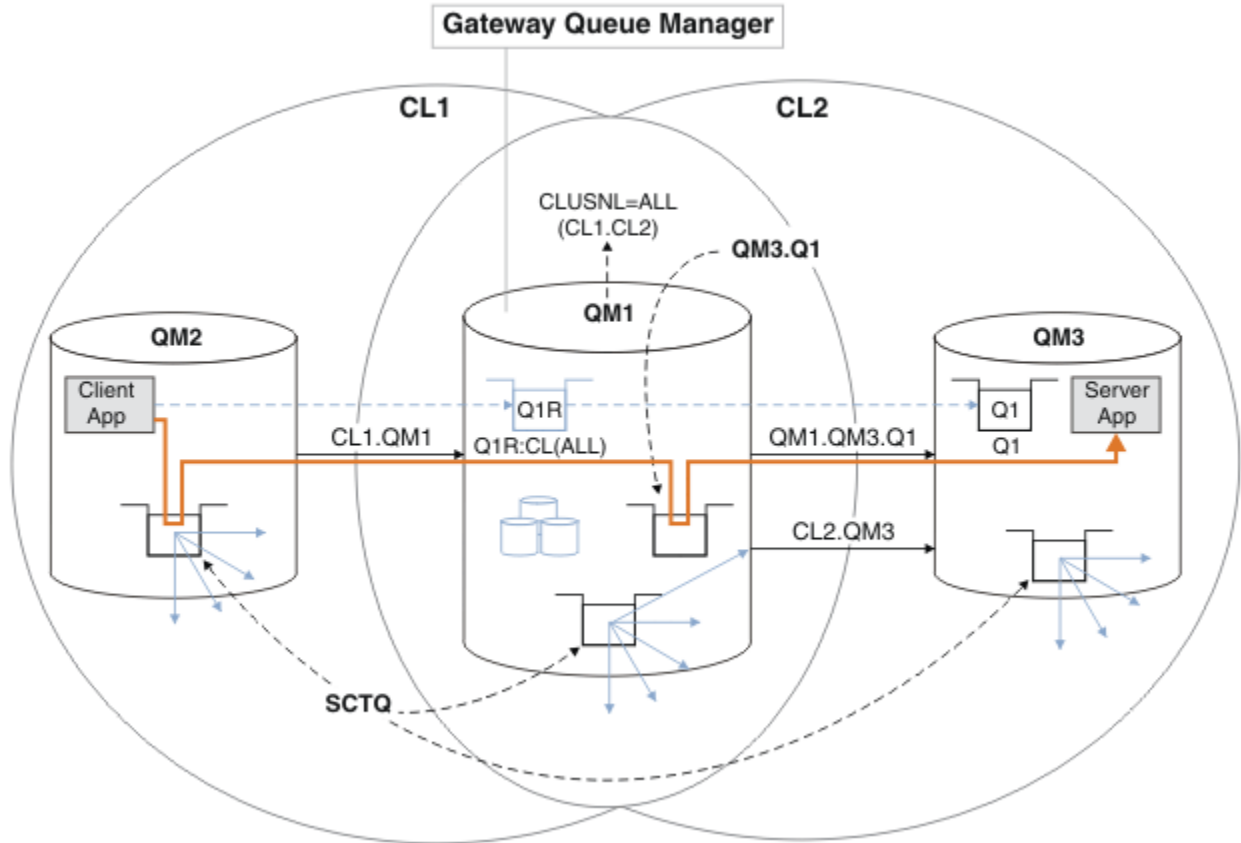
Construct the overlapping clusters shown in IBM MQ kümelerini kullanarak merkeze konuşlandırılan ve mimariye konuşlandırılmış istemci-sunucu uygulaması in “Ağ geçidi kuyruk yöneticisiyle çıkan iki küme oluşturma” sayfa 330 by following the steps in that task.

## Bu görev hakkında

Çözüm, Server App uygulamasına ilişkin iletileri ağ geçidi kuyruk yöneticisindeki diğer ileti trafiğinden ayırmak için dağıtılmış kuyruğa alma özelliğini kullanır. İletileri farklı bir iletim kuyruğuna ve farklı bir kanala yönlendirmek için QM1 üzerinde kümelenmiş bir uzak kuyruk tanımı tanımlamanız gerekir. Uzak kuyruk tanımı, iletileri yalnızca QM3 üzerinde Q1 için saklayan belirli bir iletim kuyruğuna başvuru içermelidir. Şekil 49 sayfa 335 içinde, küme kuyruğu diğer adı Q1A bir uzak kuyruk tanımı Q1R ile tamamlanmıştır ve bir iletim kuyruğu ve gönderen kanalı eklenmiştir.

Bu çözümde, tüm yanıt iletileri ortak SYSTEM.CLUSTER.TRANSMIT.QUEUE kullanılarak döndürülür.

Bu çözümün avantajı, aynı kümede, aynı kuyruk yöneticisinde birden çok hedef kuyruk için trafiği ayırmanın kolay olması. Çözümün dezavantajı, farklı kuyruk yöneticilerindeki birden çok Q1 kopyası arasında küme iş yükü dengelemeyi kullanamamamızdır. Bu dezavantajı aşmak için bkz. “Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtım için küme iletim kuyruğu eklenmesi” sayfa 337. Ayrıca anahtarı bir iletim kuyruğundan diğerine yönetmeniz gerekir.



Şekil 49. Uzak kuyruk tanımlamaları kullanılarak göbeğe ve konuştu küme mimarisine konuşlandırılan istemci-sunucu uygulaması

## Yordam

1. Q1 ileti trafiğini ağ geçidi kuyruk yöneticisinden ayırmak için bir kanal oluşturun
  - a) QM1ağ geçidi kuyruk yöneticisinde hedef kuyruk yöneticisine ( QM3) bir gönderen kanalı oluşturun.

```
DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(SDR) CONNAME(QM3HostName(1413)) XMITQ(QM3.Q1) REPLACE
```

- b) Hedef kuyruk yöneticisinde bir alıcı kanal oluşturun, QM3.

```
DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(RCVR) REPLACE
```

2. Q1 ' e ileti trafiği için ağ geçidi kuyruk yöneticisinde bir iletim kuyruğu oluşturun

```
DEFINE QLOCAL(QM3.Q1) USAGE(XMITQ) REPLACE  
START CHANNEL(QM1.QM3.Q1)
```

İletim kuyruğuyla ilişkilendirilen kanal başlatılırken, iletim kuyruğunu kanalla ilişkilendirir. Kanal, iletim kuyruğu kanalla ilişkilendirildikten sonra otomatik olarak başlar.

3. Ağ geçidi kuyruk yöneticisinde Q1 için kümelenmiş kuyruk diğer adı tanımlamasını, kümelenmiş bir uzak kuyruk tanımlamasıyla tamamlar.

```
DEFINE QREMOTE CLUSNL(ALL) RNAME(Q1) RQNAME(QM3) XMITQ(QM3.Q1) REPLACE
```

## Sonraki adım

Test the configuration by sending a message to Q1 on QM3 from QM2 using the clustered queue remote definition Q1R on the gateway queue manager QM1.

1. Bir ileti koymak için QM2 üzerinde **amqspout** örnek programını çalıştırın.

```
C:\IBM\MQ>amqspout Q1R QM2  
Sample AMQSPUT0 start  
target queue is Q1R  
Sample request message from QM2 to Q1 using Q1R
```

```
Sample AMQSPUT0 end
```

2. İletiyi şu adresten almak için **amqsget** örnek programını çalıştırın: Q1 on QM3

```
C:\IBM\MQ>amqsget Q1 QM3  
Sample AMQSGET0 start  
message <Sample request message from QM2 to Q1 using Q1R>  
no more messages  
Sample AMQSGET0 end
```

## İlgili kavramlar

Kümeleme: [Birden çok küme iletim kuyruğu kullanan uygulama yalıtımı](#)

[Erişim denetimi ve birden çok küme iletim kuyruğu](#)

## İlgili görevler

[Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtım için küme iletim kuyruğu eklenmesi](#)

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir uygulamaya aktarıldıktan sonra. ' den bir uygulamaya aktarıldıktan sonra çözüm, ileti trafiğini kümedeki tek bir kuyruk yöneticisine ayırmak için ek bir küme iletim kuyruğu kullanır.

Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme ve küme iletim kuyruğu eklenmesi

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir uygulamaya aktarıldıktan sonra. ' den bir uygulamaya aktarıldıktan sonra çözüm, iletileri belirli bir küme kuyruğunda yalıtma için ek bir küme kullanır.

İleti trafiğini yalıtma üzere küme iletim kuyruklarını ayırmak için varsayılan değerin değiştirilmesi

Varsayılan olarak, bir kuyruk yöneticisinin iletileri bir iletim kuyruğunda kümelenmiş bir kuyruğa ya da konuya ilişkin olarak saklama şeklini değiştirebilirsiniz. Varsayılan değerin değiştirilmesi, ağ geçidi kuyruk yöneticisindeki küme iletilerini yalıtmanızı sağlar.

Kümeleme: Küme iletim kuyruklarının nasıl yapılandırılacağını planlama

“Kümeyle kuyruk yöneticisi eklenmesi: ayrı iletim kuyrukları” sayfa 310

Yarattığınız kümeyle bir kuyruk yöneticisi eklemek için bu yönergeleri izleyin. Küme kuyruklarına ve konularına ilişkin iletiler, birden çok küme iletim kuyruğu kullanılarak aktarılır.

## ***Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme iletim kuyruğu eklenmesi***

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir uygulamaya aktarıldıktan sonra. ' den bir uygulamaya aktarıldıktan sonra çözüm, ileti trafiğini kümedeki tek bir kuyruk yöneticisine ayırmak için ek bir küme iletim kuyruğu kullanır.

## **Başlamadan önce**

1. Ağ geçidi kuyruk yöneticisi IBM MQ üzerinde olmalıdır.
2. Construct the overlapping clusters shown in IBM MQ kümelerini kullanarak merkeze konuşlandırılan ve mimariye konuşlandırılmış istemci-sunucu uygulaması in “Ağ geçidi kuyruk yöneticisiyle çakışan iki küme oluşturma” sayfa 330 by following the steps in that task.

## **Bu görev hakkında**

Ağ geçidi kuyruk yöneticisinde QM1, bir iletim kuyruğu ekleyin ve kuyruk özniteliğini ayarlayın CLCHNAME. CLCHNAME değerini, QM3 üzerindeki küme alıcı kanalının adına ayarlayın; bkz. Şekil 50 sayfa 338.

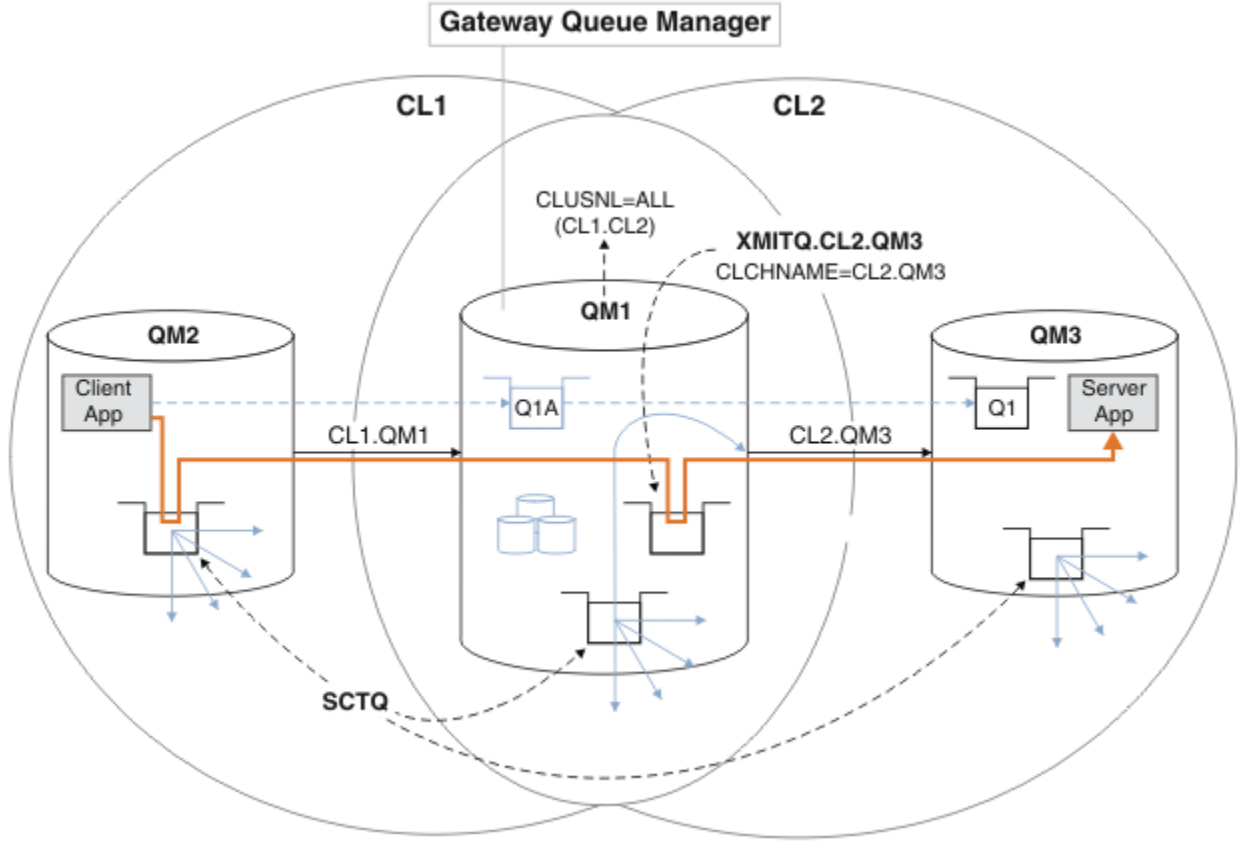
Bu çözüm, “Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtma için uzak kuyruk tanımlaması eklenmesi” sayfa 334 içinde açıklanan çözüme göre birçok avantaja sahiptir:

- Daha az ek tanımlama gerektirir.
- It supports workload balancing between multiple copies of the target queue, Q1, on different queue managers in the same cluster, CL2.
- Ağ geçidi kuyruk yöneticisi, kanal herhangi bir ileti kaybetmeden yeniden başlatıldığında otomatik olarak yeni yapılandırmaya geçer.
- Ağ geçidi kuyruk yöneticisi, iletileri aldığı sırayla iletmeye devam eder. Anahtar, QM3 hala açık SYSTEM. CLUSTER. TRANSMIT. QUEUE olan Q1 kuyruğuna ilişkin iletilerle birlikte gerçekleşse bile, bunu yapar.

Şekil 50 sayfa 338 içindeki küme iletisi trafiğini yalıtma yapılandırması, “Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtma için uzak kuyruk tanımlaması eklenmesi” sayfa 334 içindeki uzak kuyrukları kullanan yapılandırma kadar büyük bir trafiğin yalıtılmasına neden olmaz. CL2 içindeki QM3 kuyruk yöneticisi bir dizi farklı küme kuyruğu ve sunucu uygulaması barındırıyorsa, tüm bu kuyruklar QM1 ile QM3 arasında bağlantı kuran CL2. QM3 küme kanalını paylaşır. Ek akışlar Şekil 50 sayfa 338 içinde SYSTEM. CLUSTER. TRANSMIT. QUEUE ' den küme-gönderen kanalına CL2. QM3 olası küme iletisi trafiğini gösteren gri ok gösterilir.

Çözüm, kuyruk yöneticisini belirli bir kümede bir küme kuyruğu bulunduracak şekilde sınırlamaktır. Kuyruk yöneticisi zaten bir dizi küme kuyruğu barındırıyorsa, bu kısıtlamayı karşılamak için başka bir kuyruk

yöneticisi oluşturmanız ya da başka bir küme oluşturmanız gerekir; bkz. “Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtım için küme ve küme iletim kuyruğu eklenmesi” sayfa 340.



Şekil 50. Merkeze yerleştirilen istemci-sunucu uygulaması ve ek bir küme iletim kuyruğu kullanılarak konuşlandırılmış mimari.

## Yordam

1. QM1ağ geçidi kuyruk yöneticisindeki CL2 .QM3 kümeli gönderen kanalı için ek bir küme iletim kuyruğu oluşturun.

```
*... on QM1  
DEFINE QLOCAL(XMITQ.CL2.QM3) USAGE(XMITQ) CLCHNAME(CL2.QM3)
```

2. İletim kuyruğunu kullanmaya geçin, XMITQ .CL2 .QM3.
  - a) CL2 .QM3küme gönderen kanalını durdurun.

```
*... On QM1  
STOP CHANNEL(CL2.QM3)
```

Yanıt, komutun kabul edildiği:

AMQ8019: Stop IBM MQ channel accepted.

- b) CL2 .QM3 kanalının durdurulup durdurulmadığını denetleyin

Kanal durmazsa, **STOP CHANNEL** komutunu FORCE seçeneğiyle yeniden çalıştırabilirsiniz. FORCE seçeneğinin ayarlanmasına örnek olarak, kanal durmazsa ve kanalı eşzamanlamak için diğer kuyruk yöneticisini yeniden başlatamamanız verilebilir.

```
*... On QM1
start
```

Yanıt, kanal durumunun bir özetidir

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)           CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)              STATUS(STOPPED)
SUBSTATE(MQGET)           XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

c) Kanalı başlatın, CL2.QM3.

```
*... On QM1
START CHANNEL(CL2.QM3)
```

Yanıt, komutun kabul edildiği:

```
AMQ8018: Start IBM MQ channel accepted.
```

d) Kanalin başlatıldığını kontrol edin.

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

Yanıt, kanal durumunun bir özetidir:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)           CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)              STATUS(RUNNING)
SUBSTATE(MQGET)           XMITQ(XMITQ.CL2.QM3)
```

e) İletim kuyruğunun değiştirildiğini denetleyin.

```
" AMQ7341 CL2.QM3 : XMITQ.CL2.QM3 "
```

## Sonraki adım

Ayrı iletim kuyruğunu sınamak için, kuyruk diğer adı tanımını kullanarak QM2 to Q1 on QM3 (açık) seçeneğinden bir ileti gönderebilirsiniz. Q1A

1. Bir ileti koymak için QM2 üzerinde **amqspuT** örnek programını çalıştırın.

```
C:\IBM\MQ>amqspuT Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

2. İletiyi şu adresten almak için **amqsgeT** örnek programını çalıştırın: Q1 on QM3

```
C:\IBM\MQ>amqsgeT Q1 QM3
Sample AMQSGET0 start
```

```
message <Sample request message from QM2 to Q1 using Q1A>  
no more messages  
Sample AMQSGETO end
```

### **İlgili kavramlar**

Erişim denetimi ve birden çok küme iletim kuyruğu

Kümeleme: Birden çok küme iletim kuyruğu kullanan uygulama yalıtımı

“Küme iletim kuyrukları ve küme gönderen kanallarıyla çalışma” sayfa 289

Kümelenmiş kuyruk yöneticileri arasındaki iletiler küme iletim kuyruklarında saklanır ve küme gönderen kanallar tarafından iletilir. Herhangi bir zamanda, küme gönderen kanal bir iletim kuyruğuyla ilişkilendirilir. Kanalin konfigürasyonunu değiştirirseniz, kanal bir sonraki başlatılışında farklı bir iletim kuyruğuna geçebilir. Bu anahtarın işlenmesi otomatiktir ve işlemseldir.

### **İlgili görevler**

Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtma için uzak kuyruk tanımlaması eklenmesi

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir uygulamaya aktarıldıktan sonra. Çözüm, kümelenmiş bir kuyruk uzak tanımı ve ayrı bir gönderen kanalı ve iletim kuyruğunu kullanır.

Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtma için küme ve küme iletim kuyruğu eklenmesi

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir uygulamaya aktarıldıktan sonra. ' den bir uygulamaya aktarıldıktan sonra çözüm, iletileri belirli bir küme kuyruğunda yalıtma için ek bir küme kullanır.

İleti trafiğini yalıtma üzere küme iletim kuyruklarını ayırmak için varsayılan değerin değiştirilmesi

Varsayılan olarak, bir kuyruk yöneticisinin iletileri bir iletim kuyruğunda kümelenmiş bir kuyruğa ya da konuya ilişkin olarak saklama şeklini değiştirebilirsiniz. Varsayılan değerin değiştirilmesi, ağ geçidi kuyruk yöneticisindeki küme iletilerini yalıtmanızı sağlar.

Kümeleme: Küme iletim kuyruklarının nasıl yapılandırılacağını planlama

“Kümeyle kuyruk yöneticisi eklenmesi: ayrı iletim kuyrukları” sayfa 310

Yarattığınız kümeyle bir kuyruk yöneticisi eklemek için bu yönergeleri izleyin. Küme kuyruklarına ve konularına ilişkin iletiler, birden çok küme iletim kuyruğu kullanılarak aktarılır.

### ***Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtma için küme ve küme iletim kuyruğu eklenmesi***

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir uygulamaya aktarıldıktan sonra. ' den bir uygulamaya aktarıldıktan sonra çözüm, iletileri belirli bir küme kuyruğunda yalıtma için ek bir küme kullanır.

### **Başlamadan önce**

Görevdeki adımlar, Şekil 50 sayfa 338 içinde gösterilen yapılandırmayı değiştirmek için yazılır.

1. Ağ geçidi kuyruk yöneticisi IBM MQ üzerinde olmalıdır.
2. Construct the overlapping clusters shown in IBM MQ kümelerini kullanarak merkeze konuşlandırılan ve mimariye konuşlandırılmış istemci-sunucu uygulaması in “Ağ geçidi kuyruk yöneticisiyle çakışan iki küme oluşturma” sayfa 330 by following the steps in that task.
3. Çözümü ek küme olmadan oluşturmak için “Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtma için küme iletim kuyruğu eklenmesi” sayfa 337 içindeki Şekil 50 sayfa 338 içindeki adımları gerçekleştirin. Bunu, bu görevdeki adımlar için temel olarak kullanın.



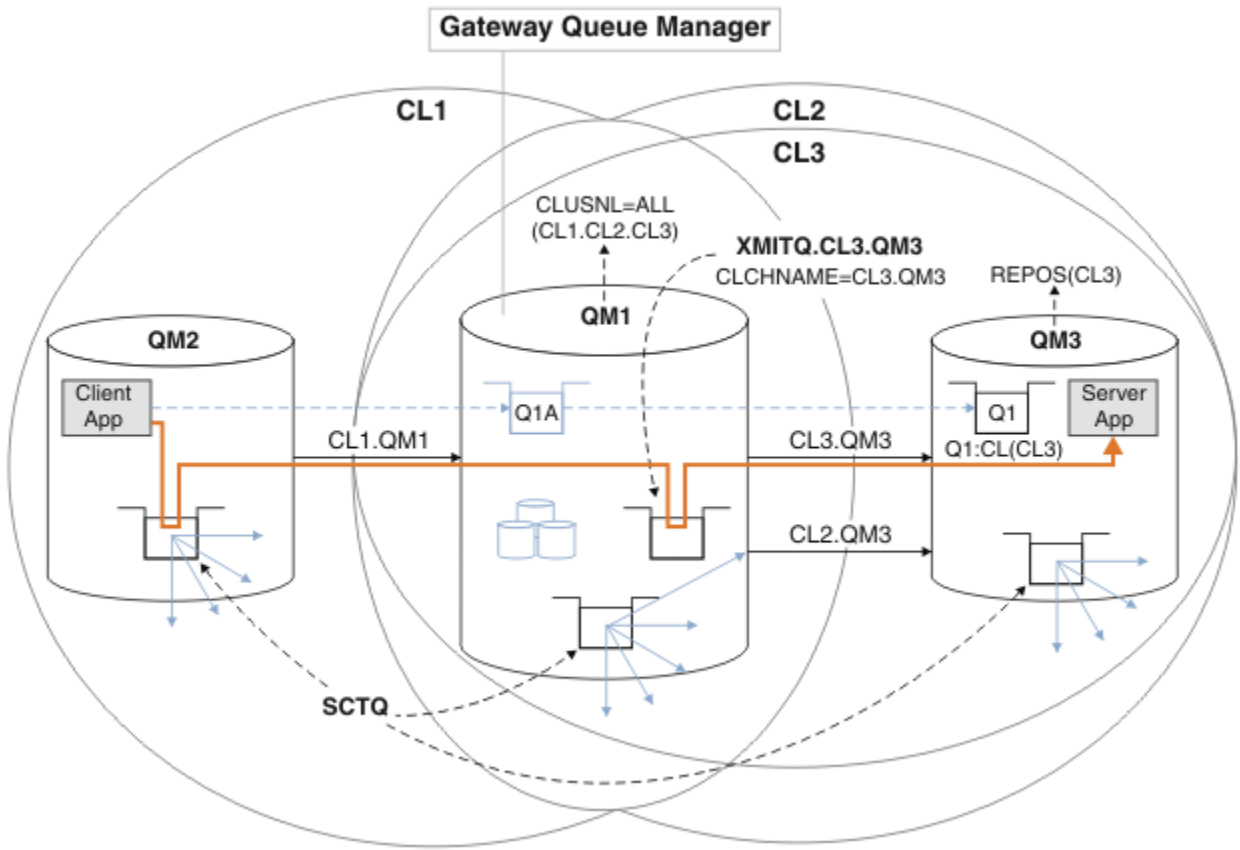
## Bu görev hakkında

Hedef küme kuyruğu, bir kuyruk yöneticisindeki tek küme kuyruğuysa, “Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme iletim kuyruğu eklenmesi” sayfa 337 içindeki tek bir uygulamaya ileti trafiğini yalıtma çözümü çalışır. Eğer değilse, iki seçeneğin var. Kuyruğu farklı bir kuyruk yöneticisine taşıyın ya da kuyruğu, kuyruk yöneticisindeki diğer küme kuyruklarından yalıtın bir küme yarattın.

Bu görev, hedef kuyruğu yalıtma için bir küme ekleme adımları boyunca size yol gösterir. Küme yalnızca bu amaçla eklenir. Uygulamada, küme ve küme adlandırma şemaları tasarlarlarken belirli uygulamaları sistematik olarak yalıtma görevine yaklaşın. Bir kuyruk her yalıtma gerektirdiğinde küme eklenmesi, yönetilebilecek birçok kümeyle sonuçlanır. Bu görevde, QM3 üzerinde Q1 ögesini yalıtma için bir küme CL3 ekleyerek “Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme iletim kuyruğu eklenmesi” sayfa 337 içindeki yapılandırmaı değiştirin. Uygulamalar değişiklik boyunca çalışmaya devam eder.

Yeni ve değiştirilen tanımlar Şekil 51 sayfa 342 içinde vurgulanır. Değişikliklerin özeti şu şekildedir: Yeni bir tam küme havuzu oluşturmanız gerektiği anlamına gelen bir küme oluşturun. Örnekte QM3 , CL3 için tam havuzlardan biri haline getirilmiştir. Ağ geçidi kuyruk yöneticisini yeni kümeye eklemek için QM1 için küme gönderen ve küme alıcı kanalları oluşturun. Q1 tanımını CL3 olarak değiştirmek için değiştirin. Ağ geçidi kuyruk yöneticisindeki küme ad değerini değiştirin ve yeni küme kanalını kullanmak için bir küme iletim kuyruğu ekleyin. Son olarak, Q1A kuyruk diğer adını yeni küme ad listesi olarak değiştirin.

IBM MQ , “Ağ geçidi kuyruk yöneticisinden gönderilen küme iletisi trafiğini yalıtma için küme iletim kuyruğu eklenmesi” sayfa 337 içine eklediğiniz XMITQ . CL2 . QM3 iletim kuyruğundan yeni iletim kuyruğuna XMITQ . CL3 . QM3 otomatik olarak ileti aktaramaz. İletileri, yalnızca her iki iletim kuyruğuna aynı küme gönderen kanalı hizmet verdiği zaman otomatik olarak aktarabilir. Görev, anahtarı el ile gerçekleştirmenin bir yolunu açıklar; bu sizin için uygun olabilir. Aktarma tamamlandığında, QM3 üzerindeki diğer CL2 küme kuyrukları için varsayılan küme iletim kuyruğunu kullanmaya geri dönme seçeneğiniz vardır. Ya da XMITQ . CL2 . QM3' yi kullanmaya devam edebilirsiniz. Varsayılan bir küme iletim kuyruğuna geri dönmeye karar vererseniz, ağ geçidi kuyruk yöneticisi anahtarı sizin için otomatik olarak yönetir.



Şekil 51. Aynı kuyruk yöneticisindeki küme kuyruklarından birine giden ağ geçidi kuyruk yöneticisindeki ileti trafiğini ayırmak için ek bir kümenin kullanılması

## Yordam

1. Kuyruk yöneticilerini QM3 ve QM5 hem CL2 hem de CL3 için havuz yapacak şekilde değiştirin.

Bir kuyruk yöneticisini birden çok kümenin üyesi yapmak için, üyesi olduğu kümeleri tanımlamak üzere bir küme adı listesi kullanılmalıdır.

```
*... On QM3 and QM5
DEFINE NAMELIST(CL23) NAMES(CL2, CL3) REPLACE
ALTER QMGR REPOS(' ') REPOSNL(CL23)
```

2. CL3 için kuyruk yöneticileri QM3 ve QM5 arasındaki kanalları tanımlayın.

```
*... On QM3
DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSRCVR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE

*... On QM5
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
```

3. Ağ geçidi kuyruk yöneticisini CL3' e ekleyin.

QM1 ögesini CL3 ögesine kısmi havuz olarak ekleyerek ağ geçidi kuyruk yöneticisini ekleyin. QM1' e küme gönderen ve küme alıcı kanalları ekleyerek kısmi bir havuz oluşturun.

Ayrıca, ağ geçidi kuyruk yöneticisine bağlı tüm kümelerin ad listesine CL3 ekleyin.

```
*... On QM1
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL3) REPLACE
ALTER NAMELIST(ALL) NAMES(CL1, CL2, CL3)
```

4. QM3 üzerinde CL3 adresine giden iletiler için ağ geçidi kuyruk yöneticisine ( QM1) bir küme iletim kuyruğu ekleyin.

Başlangıçta, iletim kuyruklarını değiştirmeye hazır oluncaya kadar iletim kuyruğundan ileti aktaran küme gönderen kanalını durdurun.

```
*... On QM1
DEFINE QLOCAL(XMITQ.CL3.QM3) USAGE(XMITQ) CLCHNAME(CL3.QM3) GET(DISABLED) REPLACE
```

5. Var olan küme iletim kuyruğundaki iletileri boşaltır XMITQ . CL2 . QM3.

Bu alt yordam, Q1 içindeki iletilerin sırasını, ağ geçidi kuyruk yöneticisine geldikleri sırayla korumayı amaçlar. Kümelerde, ileti sıralaması tam olarak garanti edilmez, ancak olası olur. Garantili ileti sıralaması gerekiyorsa, uygulamaların iletilerin sırasını tanımlaması gerekir; bkz. [İletilerin kuyruktan alınma sırası](#).

- a) QM3 üzerindeki Q1 hedef kuyruğunu CL2 olarak CL3 olarak değiştirin.

```
*... On QM3
ALTER QLOCAL(Q1) CLUSTER(CL3)
```

- b) İletiler teslim edilmeye başlayana kadar XMITQ . CL3 . QM3 ' e bakın.

Q1 to CL3 anahtarı ağ geçidi kuyruk yöneticisine yayıldığında, iletiler XMITQ . CL3 . QM3 ' e teslim edilmeye başlar.

```
*... On QM1
DISPLAY QUEUE(XMITQ.CL3.QM3) CURDEPTH
```

- c) QM3 üzerinde Q1 ' e teslim edilmeyi bekleyen ileti kalmayınca kadar XMITQ . CL2 . QM3 ' yi izleyin.

**Not:** XMITQ . CL2 . QM3 , CL2 üyesi olan QM3 üzerindeki diğer kuyruklara ilişkin iletileri saklıyor olabilir; bu durumda derinlik sıfıra gitmeyebilir.

```
*... On QM1
DISPLAY QUEUE(XMITQ.CL2.QM3) CURDEPTH
```

- d) Yeni küme iletim kuyruğundan alma işlemini etkinleştir, XMITQ . CL3 . QM3

```
*... On QM1
ALTER QLOCAL(XMITQ.CL3.QM3) GET(ENABLED)
```

6. Artık gerekmiyorsa, eski küme iletim kuyruğunu ( XMITQ . CL2 . QM3) kaldırın.

CL2 on QM3 içindeki küme kuyruklarına ilişkin iletiler, ağ geçidi kuyruk yöneticisinde ( QM1) varsayılan küme iletim kuyruğunu kullanmaya geri döner. Varsayılan küme iletim kuyruğu SYSTEM . CLUSTER . TRANSMIT . QUEUE ya da SYSTEM . CLUSTER . TRANSMIT . CL2 . QM3 ' dir. Hangisi, QM1 üzerindeki **DEFCLXQ** kuyruk yöneticisi özneliğinin değerinin SCTQ mu, yoksa CHANNEL mı olduğuna bağlıdır. Kuyruk yöneticisi, bir sonraki küme gönderen kanalı CL2 . QM3 başlatıldığında XMITQ . CL2 . QM3 ' den otomatik olarak ileti aktarır.

- a) XMITQ . CL2 . QM3 iletim kuyruğunu, küme iletim kuyruğu olmaktan normal bir iletim kuyruğuna çevirin.

Bu, herhangi bir küme gönderen kanalla iletim kuyruğunun ilişkilendirmesini bozar. Yanıt olarak, küme gönderen kanal bir sonraki başlatıldığında IBM MQ iletileri XMITQ . CL2 . QM3 ' den varsayılan küme iletim kuyruğuna otomatik olarak aktarır. O zamana kadar, QM3 üzerindeki CL2 iletileri XMITQ . CL2 . QM3 üzerine yerleştirilmeye devam eder.

```
*... On QM1
ALTER QLOCAL(XMITQ.CL2.QM3) CLCHNAME(' ')
```

- b) CL2 . QM3 küme gönderen kanalını durdurun.

Kümeyi gönderen kanalının durdurulması ve yeniden başlatılması, iletilerin XMITQ.CL2.QM3 ' den varsayılan küme iletim kuyruğuna aktarılmasını başlatır. Genellikle, aktarımı başlatmak için kanalı el ile durdurur ve başlatabilirsiniz. Kanal, bağlantı kesme aralığı sona erdikten sonra yeniden başlarsa aktarım otomatik olarak başlar.

```
*... On QM1
STOP CHANNEL (CL2.QM3)
```

Yanıt, komutun kabul edildiği:

AMQ8019: Stop IBM MQ channel accepted.

c) CL2.QM3 kanalının durdurulup durdurulmadığını denetleyin

Kanal durmazsa, **STOP CHANNEL** komutunu FORCE seçeneğiyle yeniden çalıştırabilirsiniz. FORCE seçeneğinin ayarlanmasına örnek olarak, kanal durmazsa ve kanalı eşzamanlamak için diğer kuyruk yöneticisini yeniden başlatamamanız verilebilir.

```
*... On QM1
DISPLAY CHSTATUS (CL2.QM3)
```

Yanıt, kanal durumunun bir özetidir

```
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM3)                CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1413))       CURRENT
RQMNAME (QM3)                    STATUS (STOPPED)
SUBSTATE (MQGET)                 XMITQ (XMITQ.CL2.QM3)
```

d) Kanalı başlatın, CL2.QM3.

```
*... On QM1
START CHANNEL (CL2.QM3)
```

Yanıt, komutun kabul edildiği:

AMQ8018: Start IBM MQ channel accepted.

e) Kanalin başlatıldığını kontrol edin.

```
*... On QM1
DISPLAY CHSTATUS (CL2.QM3)
```

Yanıt, kanal durumunun bir özetidir:

```
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM3)                CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1413))       CURRENT
RQMNAME (QM3)                    STATUS (RUNNING)
SUBSTATE (MQGET)                 XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE/CL2.QM3)
```

f) " AMQ7341 CL2.QM3 : SYSTEM.CLUSTER.TRANSMIT' I AL1N. QUEUE/CL2.QM3 "

g) Küme iletim kuyruğunu ( XMITQ.CL2.QM3) silin.

```
*... On QM1
DELETE QLOCAL (XMITQ.CL2.QM3)
```

## Sonraki adım

Ayrı olarak kümelenmiş kuyruğu sınamak için, kuyruk diğer adını kullanarak QM3 üzerinde QM2 'den Q1 ' e Q1A bir ileti gönderebilirsiniz.

1. Bir ileti koymak için QM2 üzerinde **amqspuT** örnek programını çalıştırın.

```
C:\IBM\MQ>amqspuT Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

2. İletiyi şu adresten almak için **amqsgeT** örnek programını çalıştırın: Q1 on QM3

```
C:\IBM\MQ>amqsgeT Q1 QM3
Sample AMQSGE0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGE0 end
```

## İlgili kavramlar

Erişim denetimi ve birden çok küme iletim kuyruğu

Kümeleme: Birden çok küme iletim kuyruğu kullanan uygulama yalıtımı

[“Küme iletim kuyrukları ve küme gönderen kanallarıyla çalışma” sayfa 289](#)

Kümelenmiş kuyruk yöneticileri arasındaki iletiler küme iletim kuyruklarında saklanır ve küme gönderen kanallar tarafından iletilir. Herhangi bir zamanda, küme gönderen kanal bir iletim kuyruğuyla ilişkilendirilir. Kanalin konfigürasyonunu değiştirirseniz, kanal bir sonraki başlatılışında farklı bir iletim kuyruğuna geçebilir. Bu anahtarın işlenmesi otomatiktir ve işlemseldir.

## İlgili görevler

[Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtım için uzak kuyruk tanımlaması eklenmesi](#)

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir uygulamaya aktarıldıktan sonra. Çözüm, kümelenmiş bir kuyruk uzak tanımı ve ayrı bir gönderen kanalı ve iletim kuyruğunu kullanır.

[Ağ geçidi kuyruk yöneticisinden gönderilen küme ileti trafiğini yalıtım için küme iletim kuyruğu eklenmesi](#)

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir uygulamaya aktarıldıktan sonra. ' den bir uygulamaya aktarıldıktan sonra çözüm, ileti trafiğini kümedeki tek bir kuyruk yöneticisine ayırmak için ek bir küme iletim kuyruğu kullanır.

[İleti trafiğini yalıtım üzere küme iletim kuyruklarını ayırmak için varsayılan değer değiştirilmesi](#)

Varsayılan olarak, bir kuyruk yöneticisinin iletileri bir iletim kuyruğunda kümelenmiş bir kuyruğa ya da konuya ilişkin olarak saklama şeklini değiştirebilirsiniz. Varsayılan değer değiştirilmesi, ağ geçidi kuyruk yöneticisindeki küme iletilerini yalıtmanızı sağlar.

Kümeleme: Küme iletim kuyruklarının nasıl yapılandırılacağını planlama

[“Kümeyle kuyruk yöneticisi eklenmesi: ayrı iletim kuyrukları” sayfa 310](#)

Yarattığınız kümeyle bir kuyruk yöneticisi eklemek için bu yönergeleri izleyin. Küme kuyruklarına ve konularına ilişkin iletiler, birden çok küme iletim kuyruğu kullanılarak aktarılır.

## İleti trafiğini yalıtılmak üzere küme iletim kuyruklarını ayırmak için varsayılan değerin değiştirilmesi

Varsayılan olarak, bir kuyruk yöneticisinin iletileri bir iletim kuyruğunda kümelenmiş bir kuyruğa ya da konuya ilişkin olarak saklama şeklini değiştirebilirsiniz. Varsayılan değerin değiştirilmesi, ağ geçidi kuyruk yöneticisindeki küme iletilerini yalıtmanızı sağlar.

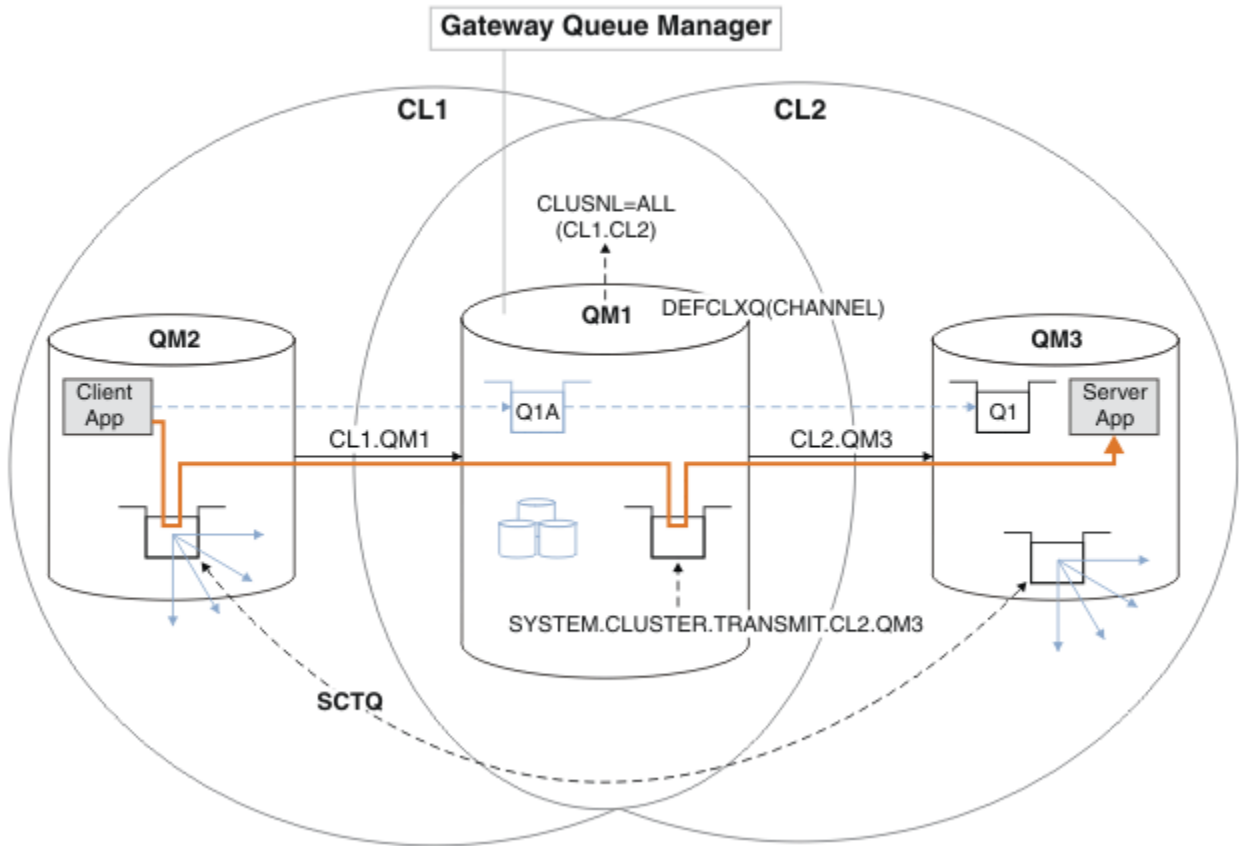
### Başlamadan önce

1. Ağ geçidi kuyruk yöneticisi IBM MQüzerinde olmalıdır.
2. Construct the overlapping clusters shown in IBM MQ kümelerini kullanarak merkeze konuşlandırılan ve mimariye konuşlandırılmış istemci-sunucu uygulaması in "Ağ geçidi kuyruk yöneticisiyle çıkan iki küme oluşturma" sayfa 330 by following the steps in that task.

### Bu görev hakkında

Mimariyi birden çok küme kuyruğuyla uygulamak için ağ geçidi kuyruk yöneticinizin IBM MQüzerinde olması gerekir. Birden çok küme iletim kuyruğu kullanmak için tek yapmanız gereken, ağ geçidi kuyruk yöneticisindeki varsayılan küme iletim kuyruğu tipini değiştirmektir. **DEFCLXQ** on QM1 kuyruk yöneticisi özniteliğinin değerini SCTQ değerinden CHANNEL değerine çevirin; bkz. Şekil 52 sayfa 346. Çizgede bir ileti akışı gösterilir. Diğer kuyruk yöneticilerine ya da diğer kümelere yönelik akışlar için, kuyruk yöneticisi ek kalıcı dinamik küme iletim kuyrukları yaratır. Her küme gönderen kanal, farklı bir küme iletim kuyruğundan ileti aktarır.

Ağ geçidi kuyruk yöneticisini kümelere ilk kez bağlamıyorsanız, değişiklik hemen yürürlüğe girmez. Görev, var olan bir yapılandırmada yapılan bir değişikliğin yönetilmesine ilişkin tipik adımlara ilişkin adımları içerir. Bir kümeyi ilk birleştirdiğinde ayrı küme iletim kuyrukları kullanmak üzere bir kuyruk yöneticisi ayarlamak için bkz. "Kümeyle kuyruk yöneticisi eklenmesi: ayrı iletim kuyrukları" sayfa 310.



Şekil 52. İstemci-sunucu uygulaması, ağ geçidi kuyruk yöneticisinde ayrı küme iletim kuyruklarıyla göbeğe konuşlandırıldı ve mimariyi konuştu.

## Yordam

1. Ağ geçidi kuyruk yöneticisini ayrı küme iletim kuyrukları kullanacak şekilde değiştirin.

```
*... On QM1
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. Ayrı küme iletim kuyruklarına geçin.

Çalışmayan herhangi bir küme gönderen kanal, bir sonraki başlatılışında ayrı küme iletim kuyruklarını kullanmaya geçer.

Çalışan kanalları değiştirmek için kuyruk yöneticisini yeniden başlatın ya da aşağıdaki adımları izleyin:

- a) SYSTEM.CLUSTER.TRANSMIT.QUEUE ile çalışan küme gönderen kanallarını listeleyin.

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

Yanıt, kanal durumu raporlarının listesidir:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1412)) CURRENT
RQMNAME(QM2)             STATUS(RUNNING)
SUBSTATE(MQGET)          XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)             STATUS(RUNNING)
SUBSTATE(MQGET)          XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1415)) CURRENT
RQMNAME(QM5)             STATUS(RUNNING)
SUBSTATE(MQGET)          XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1414)) CURRENT
RQMNAME(QM4)             STATUS(RUNNING)
SUBSTATE(MQGET)          XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

- b) Çalışmakta olan kanalları durdur

Listedeki her kanal için şu komutu çalıştırın:

```
*... On QM1
STOP CHANNEL(ChannelName)
```

Burada *ChannelName*, CL1.QM2, CL1.QM4, CL1.QM3, CL1.QM5' in her birini gösterir.

Yanıt, komutun kabul edildiği:

```
AMQ8019: Stop IBM MQ channel accepted.
```

- c) Hangi kanalların durdurulduğunu izle

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

Yanıt, çalışmakta olan kanalların ve durdurulan kanalların bir listesidir:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1412)) CURRENT
RQMNAME(QM2)             STATUS(STOPPED)
SUBSTATE( )              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)             STATUS(STOPPED)
SUBSTATE( )              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1415)) CURRENT
RQMNAME(QM5)             STATUS(STOPPED)
SUBSTATE( )              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1414)) CURRENT
RQMNAME(QM4)             STATUS(STOPPED)
SUBSTATE( )              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

d) Durdurulan her kanalı başlat.

Çalışmakta olan tüm kanallar için bu adımı yapın. Bir kanal durmazsa, **STOP CHANNEL** komutunu **FORCE** seçeneğiyle yeniden çalıştırabilirsiniz. **FORCE** seçeneğinin ayarlanmasına örnek olarak, kanal durmazsa ve kanalı eşzamanlamak için diğer kuyruk yöneticisini yeniden başlatamamanız verilebilir.

```
*... On QM1
START CHANNEL(CL2.QM5)
```

Yanıt, komutun kabul edildiği:

```
AMQ8018: Start IBM MQ channel accepted.
```

e) Değiştirilmekte olan iletim kuyruklarını izleyin.

```
" AMQ7341 CL2.QM3 : SYSTEM.CLUSTER.TRANSMIT' I AL1N. QUEUE/CL2.QM3 "
```

f) **SYSTEM.CLUSTER.TRANSMIT.QUEUE** ' in artık kullanılmadığını denetleyin

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
DISPLAY QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE) CURDEPTH
```

Yanıt, kanal durumu raporlarının listesi ve **SYSTEM.CLUSTER.TRANSMIT.QUEUE** derinliğidir:

```
AMQ8420: Channel Status not found.
AMQ8409: Display Queue details.
QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE)    TYPE(QLOCAL)
CURDEPTH(0)
```

g) Hangi kanalların başlatıldığını izle

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

Yanıt, kanalların listesidir; bu durumda yeni varsayılan küme iletim kuyruklarıyla çalışır:



```

AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM2)                CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1412))      CURRENT
RQMNAME (QM2)                   STATUS (RUNNING)
SUBSTATE (MQGET)
XMITQ (SYSTEM.CLUSTER.TRANSMIT.CL1.QM2)
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM3)                CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1413))      CURRENT
RQMNAME (QM3)                   STATUS (RUNNING)
SUBSTATE (MQGET)
XMITQ (SYSTEM.CLUSTER.TRANSMIT.CL2.QM3)
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM5)                CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1415))      CURRENT
RQMNAME (QM5)                   STATUS (RUNNING)
SUBSTATE (MQGET)
XMITQ (SYSTEM.CLUSTER.TRANSMIT.CL2.QM5)
AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM4)                CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1414))      CURRENT
RQMNAME (QM4)                   STATUS (RUNNING)
SUBSTATE (MQGET)
XMITQ (SYSTEM.CLUSTER.TRANSMIT.CL1.QM4)

```

## Sonraki adım

1. Otomatik olarak tanımlanan küme iletim kuyruğunu, QM2 'dan Q1 on QM3' a bir ileti göndererek, kuyruk diğer adını Q1A kuyruk diğer adı tanımlamasıyla çözerek sınavın.
  - a. Bir ileti koymak için QM2 üzerinde **amqsput** örnek programını çalıştırın.

```

C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end

```

- b. İletiyi şu adresten almak için **amqsget** örnek programını çalıştırın: Q1 on QM3

```

C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end

```

2. Küme kuyruklarına ilişkin iletilerin kaynaklandığı kuyruk yöneticilerindeki küme kuyrukları için güvenliği yapılandırarak güvenliğin yeniden yapılandırılıp yapılandırılmayacağını göz önünde bulundurun.

## İlgili kavramlar

Erişim denetimi ve birden çok küme iletim kuyruğu

Kümeleme: Birden çok küme iletim kuyruğu kullanan uygulama yalıtımı

## İlgili görevler

Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtım için uzak kuyruk tanımlaması eklenmesi

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir uygulamaya aktarıldıktan sonra. Çözüm, kümelenecek bir kuyruk uzak tanımı ve ayrı bir gönderen kanalı ve iletim kuyruğunu kullanır.

Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtma için küme iletim kuyruğu eklenmesi

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir uygulamaya aktarıldıktan sonra. ' den bir uygulamaya aktarıldıktan sonra çözüm, ileti trafiğini kümedeki tek bir kuyruk yöneticisine ayırmak için ek bir küme iletim kuyruğu kullanır.

Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtma için küme ve küme iletim kuyruğu eklenmesi

Ağ geçidi kuyruk yöneticisini kullanan çakışan kümelerin yapılandırmasını değiştirin. Değişiklik iletileri, diğer küme iletileriyle aynı iletim kuyruğu ya da kanalları kullanılmadan ağ geçidi kuyruk yöneticisinden bir uygulamaya aktarıldıktan sonra. ' den bir uygulamaya aktarıldıktan sonra çözüm, iletileri belirli bir küme kuyruğunda yalıtma için ek bir küme kullanır.

Kümeleme: Küme iletim kuyruklarının nasıl yapılandırılacağını planlama

“Kümeyle kuyruk yöneticisi eklenmesi: ayrı iletim kuyrukları” sayfa 310

Yarattığınız kümeyle bir kuyruk yöneticisi eklemek için bu yönergeleri izleyin. Küme kuyruklarına ve konularına ilişkin iletiler, birden çok küme iletim kuyruğu kullanılarak aktarılır.

## Küme kuyruğunun kuyruk yöneticisinden kaldırılması

Toronto 'daki INVENTQ kuyruğunu devre dışı bırakın. Tüm döküm iletilerini New York 'a gönderin ve boş olduğunda Toronto 'daki INVENTQ kuyruğunu silin.

### Başlamadan önce

**Not:** Bir kümede yapılan değişikliklerin kümeyle yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- INVENTORY kümesi, “Kuyruğu barındıran bir kuyruk yöneticisi eklenmesi” sayfa 314 içinde açıklandığı gibi ayarlandı. Dört kuyruk yöneticisi içerir. LONDON ve NEWYORK her ikisi de tam havuzları tutar. PARIS ve TORONTO kısmi havuzları tutar. Döküm uygulaması, New York ve Toronto 'daki sistemlerde çalışır ve INVENTQ kuyruğundaki iletilerin gelişyle yönlendirilir.
- İş yükünün azalması nedeniyle, artık Toronto 'da stok uygulamasını çalıştırmak istemeyiz. INVENTQ Kuyruk yöneticisi tarafından barındırılan kuyruğu TORONTO devre dışı bırakmak ve TORONTO besleme akışı iletilerinin NEWYORK'taki INVENTQ kuyruğuna gönderilmesini istiyorsunuz.
- Dört sistem arasında ağ bağlantısı vardır.
- Ağ protokolü TCP 'dir.

### Bu görev hakkında

Bir küme kuyruğunu kaldırmak için aşağıdaki adımları izleyin.

### Yordam

1. Kuyruğun artık kullanılabilir olmadığını belirtin.

Bir kuyruğu kümeden kaldırmak için, küme adını yerel kuyruk tanımlamasından kaldırın. INVENTQ on TORONTO ögesini, kümenin geri kalanından erişilemeyecek şekilde değiştirin:

```
ALTER QLOCAL(INVENTQ) CLUSTER(' ')
```

2. Kuyruğun artık kullanılabilir olmadığını denetleyin.

Tam bir havuz kuyruğu yöneticisinde ( LONDON ya da NEWYORK), aşağıdaki komutu vererek kuyruğun artık kuyruk yöneticisi tarafından barındırılmadığını doğrulayın: TORONTO

```
DIS QCLUSTER (INVENTQ)
```

ALTER komutu başarıyla tamamlandıysa, TORONTO sonuçlarda listelenmez.

### 3. Kuyruğu devre dışı bırakın.

Başka iletilerin yazılabilmesi için TORONTO adresindeki INVENTQ kuyruğunu devre dışı bırakın:

```
ALTER QLOCAL(INVENTQ) PUT(DISABLED)
```

Şimdi MQ00\_BIND\_ON\_OPEN kullanılarak bu kuyruğa aktarılan iletiler, gitmeyen iletiler kuyruğuna gider. Tüm uygulamaların iletileri bu kuyruk yöneticisine belirtik olarak kuyruğa koymasını durdurmanız gerekir.

### 4. Boş oluncaya kadar kuyruğu izleyin.

IPPROCS, OPPROCS ve CURDEPTH özneliklerini belirterek DISPLAY QUEUE komutunu kullanarak kuyruğu izleyin ya da IBM üzerinde **WRKMQMSTS** komutunu kullanın. Giriş ve çıkış işlemlerinin sayısı ve kuyruğun yürürlükteki derinliği sıfır olduğunda, kuyruk boş olur.

### 5. Belirsiz ileti olmadığından emin olmak için kanalı izleyin.

INVENTORY .TORONTO kanalında şüpheli ileti olmadığından emin olmak için, diğer kuyruk yöneticilerinin her birinde INVENTORY .TORONTO adlı küme gönderen kanalı izleyin. Her kuyruk yöneticisinden INDOUBT deęiřtirgesini belirterek DISPLAY CHSTATUS komutunu verin:

```
DISPLAY CHSTATUS(INVENTORY.TORONTO) INDOUBT
```

Belirsiz iletiler varsa, devam etmeden önce bunları çözeniz gerekir. Örneęin, RESOLVE kanal komutunu çalıştırmayı ya da kanalı durdurup yeniden başlatmayı deneyebilirsiniz.

### 6. Yerel kuyruğu silin.

TORONTO adresindeki döküm uygulamasına teslim edilecek başka ileti olmadığından emin olduğunuzda, kuyruğu silebilirsiniz:

```
DELETE QLOCAL(INVENTQ)
```

### 7. Artık Toronto 'daki sistemden döküm uygulamasını kaldırabilirsiniz

Uygulamanın kaldırılması, yinelemeyi önler ve sistemde yer tasarrufu sağlar.

## Sonuçlar

Bu görev tarafından ayarlanan küme, önceki görev tarafından ayarlanan kümeye benzer. Fark, INVENTQ kuyruk yöneticisinde artık kuyruk yok TORONTO.

1. adımda kuyruğu hizmet dışı bıraktığınızda, TORONTO kuyruk yöneticisi iki tam havuz kuyruğu yöneticisine bir ileti gönderdi. Onlara durum deęiřiklięini bildirdi. Tam havuz kuyruğu yöneticileri bu bilgileri, INVENTQ ile ilgili bilgilerde güncelleme isteęinde bulunan kümedeki dięer kuyruk yöneticilerine iletir.

Bir kuyruk yöneticisi INVENTQ kuyruğuna bir ileti koyduğunda, güncellenen kısmi havuz INVENTQ kuyruğunun yalnızca NEWYORK kuyruk yöneticisinde kullanılabilir olduğunu gösterir. İleti, NEWYORK kuyruk yöneticisine gönderilir.

## Sonraki adım

Bu görevde, kaldırılacağı tek bir kuyruk vardı ve kaldırılacağı tek bir küme vardı.

Birçok küme adı içeren bir ad melist 'e gönderme yapan birçok kuyruk olduğunu varsayalım. Örneğin, TORONTO kuyruk yöneticisi yalnızca INVENTQürününü değil, PAYROLLQ, SALESQve PURCHASESQöğelerini de barındıracak olabilir. TORONTO , bu kuyrukları tüm uygun kümelerde ( INVENTORY, PAYROLL, SALESve PURCHASES) kullanılabilir duruma getirir. TORONTO kuyruk yöneticisinde küme adlarının ad listesini tanımlayın:

```
DEFINE NAMELIST(TOROLIST)
DESCR('List of clusters TORONTO is in')
NAMES(INVENTORY, PAYROLL, SALES, PURCHASES)
```

Her kuyruk tanımlamasına ad listesi ekleyin:

```
DEFINE QLOCAL(INVENTQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PAYROLLQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(SALESQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PURCHASESQ) CLUSNL(TOROLIST)
```

SALES işlemi PURCHASES işlemi tarafından devralınacağından, tüm bu kuyrukları SALES kümesinden kaldırmak istediğinizi varsayın. Yapmanız gereken tek şey, SALES kümesinin adını kümeden kaldırmak için TOROLIST ad melist ögesini değiştirmektir.

Ad listesindeki kümelerden birinden tek bir kuyruğu kaldırmak istiyorsanız, kalan küme adları listesini içeren bir ad listesi yaratın. Daha sonra, kuyruk tanımlamasını yeni ad listesi kullanacak şekilde değiştirin. PAYROLLQ ögesini INVENTORY kümesinden kaldırmak için:

1. Ad listesi yarat:

```
DEFINE NAMELIST(TOROSHORTLIST)
DESCR('List of clusters TORONTO is in other than INVENTORY')
NAMES(PAYROLL, SALES, PURCHASES)
```

2. PAYROLLQ kuyruk tanımlamasını değiştirin:

```
ALTER QLOCAL(PAYROLLQ) CLUSNL(TOROSHORTLIST)
```

## Bir kümeden kuyruk yöneticisinin kaldırılması: en iyi uygulama

Kuyruk yöneticisinin kümedeki en az bir tam havuzla olağan şekilde iletişim kurabileceği senaryolarda, bir kuyruk yöneticisini kümeden kaldırın.

### Başlamadan önce

Bu yöntem, en az bir tam havuzun kullanılabilir olduğu ve kaldırılmakta olan kuyruk yöneticisi tarafından iletişim kurulabileceği senaryolar için en iyi uygulamadır. Bu yöntem, en az el ile gerçekleştirilen müdahaleyi içerir ve kuyruk yöneticisinin kümeden denetimli olarak geri çekilmesini kararlaştırmasına izin verir. Kaldırılmakta olan kuyruk yöneticisi tam havuzla iletişim kuramazsa, bkz. [“Bir kümeden kuyruk yöneticisinin kaldırılması: Alternatif yöntem” sayfa 354.](#)

### Bu görev hakkında

Bu örnek görev, LONDON kuyruk yöneticisini INVENTORY kümesinden kaldırır. INVENTORY kümesi, [“Kümeye kuyruk yöneticisi eklenmesi” sayfa 308](#) içinde açıklandığı gibi ayarlanır ve [“Küme kuyruğunun kuyruk yöneticisinden kaldırılması” sayfa 350](#) içinde açıklandığı şekilde değiştirilir.

Bir kümeden kuyruk yöneticisi kaldırma işlemi, kuyruk yöneticisi ekleme işleminden daha karmaşıktır.

Bir kuyruk yöneticisi bir kümeye katıldığında, kümenin var olan üyelerinin yeni kuyruk yöneticisi hakkında bilgisi yoktur ve bu nedenle kümeyle etkileşimleri yoktur. Yeni gönderen ve alıcı kanalları, tam bir havuza bağlanabilmesi için birleştirme kuyruk yöneticisinde yaratılmalıdır.

Bir kümeden bir kuyruk yöneticisi kaldırıldığında, kuyruk yöneticisine bağlı uygulamalar kümenin başka bir yerinde bulunan kuyruklar gibi nesnelere kullanıyor olabilir. Ayrıca, kümedeki diğer kuyruk yöneticilerine bağlı uygulamalar, hedef kuyruk yöneticisinde barındırılan nesnelere kullanıyor olabilir. Bu uygulamaların sonucu olarak, yürürlükteki kuyruk yöneticisi kümeye katılmak için kullandığı tam havuz dışındaki küme üyeleriyle iletişim kurmak için ek gönderen kanalları yaratabilir. Kümedeki her kuyruk yöneticisi, diğer küme üyelerini tanımlayan verilerin ön belleğe alınmış bir kopyasına sahiptir. Bu, kaldırılmakta olan bir taneyi içerebilir.

## Yordam

1. Kuyruk yöneticisini kümeden kaldırmadan önce, kuyruk yöneticisinin artık küme için gerekli kaynakları barındırmadığından emin olun:

- Kuyruk yöneticisi tam bir havuz barındırıyor, "[Tam havuzun başka bir kuyruk yöneticisine taşınması](#)" sayfa 319'ünden 1-6 arasındaki adımları tamamlayın. Kaldırılacak kuyruk yöneticisinin tam havuz işlevi farklı bir kuyruk yöneticisine taşınmazsa, yalnızca 5. ve 6. adımların tamamlanması gerekir.
- Kuyruk yöneticisi küme kuyruklarını barındırıyor, "[Küme kuyruğunun kuyruk yöneticisinden kaldırılması](#)" sayfa 350'den 1-7 arasındaki adımları tamamlayın.
- Kuyruk yöneticisi küme konularını barındırıyor, konuları silin (örneğin, `DELETE TOPIC` komutunu kullanarak) ya da diğer anasistemlere taşıyın (bkz. "[Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması](#)" sayfa 429).

**Not:** Bir kümeden bir kuyruk yöneticisini kaldırırsanız ve kuyruk yöneticisi bir küme konusunu içermeye devam ederse, kuyruk yöneticisi konu silininceye kadar kümede kalan kuyruk yöneticilerine yayınları teslim etmeyi denemeye devam edebilir.

2. El ile tanımlanan küme alıcı kanallarını, LONDONkuyruk yöneticisinde kümeden kaldıracak şekilde değiştirin:

```
ALTER CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) CLUSTER(' ')
```

3. El ile tanımlanan küme gönderen kanallarını, LONDONkuyruk yöneticisinde kümeden kaldırmak için değiştirin:

```
ALTER CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) CLUSTER(' ')
```

Kümedeki diğer kuyruk yöneticileri, bu kuyruk yöneticisinin ve küme kaynaklarının artık kümenin bir parçası olmadığını öğrenir.

4. Kümedeki herhangi bir tam havuza akmayı bekleyen ileti kalmayınca kadar, LONDONkuyruk yöneticisinde küme iletim kuyruğunu izleyin.

```
DISPLAY CHSTATUS(INVENTORY.PARIS) XQMSGSA
```

İletiler iletim kuyruğunda kalmaya devam ederse, devam etmeden önce iletilerin neden PARIS ve NEWYORK tam havuzlarına gönderilmediğini belirleyin.

## Sonuçlar

LONDON kuyruk yöneticisi artık kümenin bir parçası değil. Ancak, bağımsız bir kuyruk yöneticisi olarak işlev görür.

## Sonraki adım

Bu değişikliklerin sonucu, kümenin geri kalan üyelerinde aşağıdaki komutu vererek doğrulanabilir:

```
DISPLAY CLUSQMGR(LONDON)
```

Kuyruk yöneticisi, otomatik olarak tanımlanan küme gönderen kanalları duruncaya kadar görüntülenmeye devam eder. Bunun gerçekleşmesini bekleyebilir ya da aşağıdaki komutu vererek etkin yönetim ortamlarını izlemeye devam edebilirsiniz:

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

Bu kuyruk yöneticisine başka ileti teslim edilmeyeceğinden emin olduğunuzda, kümenin geri kalan üyelerinde aşağıdaki komutu vererek LONDON ' e küme gönderen kanallarını durdurabilirsiniz:

```
STOP CHANNEL(INVENTORY.LONDON) STATUS(INACTIVE)
```

Değişiklikler kümeye yayıldıktan ve bu kuyruk yöneticisine başka ileti teslim edildikten sonra, LONDON üzerindeki CLUSRCVR kanalını durdurun ve silin:

```
STOP CHANNEL(INVENTORY.LONDON)  
DELETE CHANNEL(INVENTORY.LONDON)
```

Bu kanal için el ile tanımlanmış bir iletim kuyruğu kullanıyorsa ve CLCHNAME örüntüsü varolan ya da planlanan diğer kanallarla eşleşmiyorsa, iletim kuyruğunu silmek isteyebilirsiniz. Örneğin:

```
DELETE QLOCAL(PARIS.CUSTOM.XMITQ)
```

**Not:** Otomatik tanımlı iletim kuyrukları ya da paylaşılan SYSTEM.CLUSTER.TRANSMIT.QUEUE kullanımda, bu adım gerekli değil.

Kaldırılan kuyruk yöneticisi, “Kümeye kuyruk yöneticisi eklenmesi” sayfa 308’inde açıklandığı gibi daha sonraki bir noktada kümeye yeniden eklenebilir. Kaldırılan kuyruk yöneticisi, kümenin geri kalan üyelerine ilişkin bilgileri 90 güne kadar önbelleğe almaya devam eder. Bu önbelleğin süresi doluncaya kadar beklememeyi tercih ediyorsanız, bu önbellek [“Bir kuyruk yöneticisinin küme öncesi durumuna geri yüklenmesi” sayfa 356’inde açıklandığı gibi zorla kaldırılabilir.](#)

### İlgili görevler

Bir kümeden kuyruk yöneticisinin kaldırılması ( IBM MQ Explorer kullanılarak)

### İlgili başvurular

[ALTER CHANNEL \(kanal ayarlarını değiştir\)](#)

[DISPLAY CHANNEL \(kanal tanımını görüntüle\)](#)

[DISPLAY CHSTATUS \(KANAL DURUMUNU GÖRÜNTÜLE\)](#)

[DISPLAY CLUSQMGR \(küme kuyruğu yöneticileri için kanal bilgilerini görüntüler\)](#)

[STOP CHANNEL \(kanalı durdur\)](#)

### ***Bir kümeden kuyruk yöneticisinin kaldırılması: Alternatif yöntem***

Önemli bir sistem ya da yapılandırma sorunu nedeniyle kuyruk yöneticisinin kümedeki hiçbir tam havuzla iletişim kuramadığı durumlarda, bir kuyruk yöneticisini kümeden kaldırın.

### **Başlamadan önce**

Bir kümeden kuyruk yöneticisini kaldırmanın bu alternatif yöntemi, kaldırılan kuyruk yöneticisini kümeyle bağlayan tüm küme kanallarını el ile durdurur ve siler ve kuyruk yöneticisini kümeden zorla kaldırır. Bu yöntem, kaldırılan kuyruk yöneticisinin tüm havuzlarla iletişim kuramadığı senaryolarda kullanılır. Bunun nedeni (örneğin) kuyruk yöneticisinin çalışmayı durdurması ya da kuyruk yöneticisi ile küme arasında uzun süreli bir iletişim hatası oluşması olabilir. Ters durumda, en sık kullanılan yöntemi kullanın: [“Bir kümeden kuyruk yöneticisinin kaldırılması: en iyi uygulama” sayfa 352.](#)

## Bu görev hakkında

Bu örnek görev, LONDON kuyruk yöneticisini INVENTORY kümesinden kaldırır. INVENTORY kümesi, “Küme kuyruk yöneticisi eklenmesi” sayfa 308’inde açıklandığı gibi ayarlanır ve “Küme kuyruğunun kuyruk yöneticisinden kaldırılması” sayfa 350’inde açıklandığı şekilde değiştirilir.

Bir kümeden kuyruk yöneticisi kaldırma işlemi, kuyruk yöneticisi ekleme işleminden daha karmaşıktır.

Bir kuyruk yöneticisi bir küme katıldığında, kümenin var olan üyelerinin yeni kuyruk yöneticisi hakkında bilgisi yoktur ve bu nedenle kümeyle etkileşimleri yoktur. Yeni gönderen ve alıcı kanalları, tam bir havuza bağlanabilmesi için birleştirme kuyruk yöneticisinde yaratılmalıdır.

Bir kümeden bir kuyruk yöneticisi kaldırıldığında, kuyruk yöneticisine bağlı uygulamalar kümenin başka bir yerinde bulunan kuyruklar gibi nesnelere kullanıyor olabilir. Ayrıca, kümedeki diğer kuyruk yöneticilerine bağlı uygulamalar, hedef kuyruk yöneticisinde barındırılan nesnelere kullanıyor olabilir. Bu uygulamaların sonucu olarak, yürürlükteki kuyruk yöneticisi küme katılmak için kullandığı tam havuz dışındaki küme üyeleriyle iletişim kurmak için ek gönderen kanalları yaratabilir. Kümedeki her kuyruk yöneticisi, diğer küme üyelerini tanımlayan verilerin önbelleğe alınmış bir kopyasına sahiptir. Bu, kaldırılmakta olan bir taneyi içerebilir.

Kuyruk yöneticisinin kümeden düzgün bir şekilde çıkmasını beklemek mümkün olmadığında, bu yordam acil durumlarda uygun olabilir.

## Yordam

1. Kuyruk yöneticisini kümeden kaldırmadan önce, kuyruk yöneticisinin artık küme için gerekli kaynakları barındırmadığından emin olun:

- Kuyruk yöneticisi tam bir havuz barındırıyorsa, “Tam havuzun başka bir kuyruk yöneticisine taşınması” sayfa 319’ünden 1-6 arasındaki adımları tamamlayın. Kaldırılacak kuyruk yöneticisinin tam havuz işlevi farklı bir kuyruk yöneticisine taşınmazsa, yalnızca 5. ve 6. adımların tamamlanması gerekir.
- Kuyruk yöneticisi küme kuyruklarını barındırıyorsa, “Küme kuyruğunun kuyruk yöneticisinden kaldırılması” sayfa 350’den 1-7 arasındaki adımları tamamlayın.
- Kuyruk yöneticisi küme konularını barındırıyorsa, konuları silin (örneğin, `DELETE TOPIC` komutunu kullanarak) ya da diğer anasistemlere taşıyın (bkz. “Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması” sayfa 429).

**Not:** Bir kümeden bir kuyruk yöneticisini kaldırılırsa ve kuyruk yöneticisi bir küme konusunu içermeye devam ederse, kuyruk yöneticisi konu silininceye kadar kümede kalan kuyruk yöneticilerine yayınları teslim etmeyi denemeye devam edebilir.

2. Kümedeki diğer kuyruk yöneticileriyle iletişim kurmak için kullanılan tüm kanalları durdurun. Kuyruk yöneticisindeki CLUSRCVR kanalını LONDONdurdurmak için `MODE (FORCE)` komutunu kullanın. Ters durumda, gönderen kuyruk yöneticisinin kanalı durdurmasını beklemeniz gerekebilir:

```
STOP CHANNEL (INVENTORY.LONDON) MODE (FORCE)
STOP CHANNEL (INVENTORY.TORONTO)
STOP CHANNEL (INVENTORY.PARIS)
STOP CHANNEL (INVENTORY.NEWYORK)
```

3. Kanallar duruncaya kadar LONDONkuyruk yöneticisinde kanal durumlarını izleyin:

```
DISPLAY CHSTATUS (INVENTORY.LONDON)
DISPLAY CHSTATUS (INVENTORY.TORONTO)
DISPLAY CHSTATUS (INVENTORY.PARIS)
DISPLAY CHSTATUS (INVENTORY.NEWYORK)
```

Kanallar durduktan sonra kümedeki diğer kuyruk yöneticilerine başka uygulama iletisi gönderilmez.

4. LONDONkuyruk yöneticisinde el ile tanımlanan küme kanallarını silin:

```
DELETE CHANNEL (INVENTORY .NEWYORK)
DELETE CHANNEL (INVENTORY .TORONTO)
```

5. Kümedeki diğer kuyruk yöneticileri, kaldırılan kuyruk yöneticisi hakkında bilgi sahibi olmaya devam eder ve bu kuyruk yöneticisine ileti göndermeye devam edebilir. Geri kalan kuyruk yöneticilerinden bilgileri temizlemek için, kaldırılan kuyruk yöneticisini kümeden tam havuzlardan birinde ilk durumuna getirin:

```
RESET CLUSTER(INVENTORY) ACTION(FORCEREMOVE) QMNAME(LONDON) QUEUES(YES)
```

Kümede, kaldırılan kuyruk yöneticisiyle aynı ada sahip başka bir kuyruk yöneticisi varsa, kaldırılan kuyruk yöneticisine ilişkin **QMID** değerini belirtin.

## Sonuçlar

LONDON kuyruk yöneticisi artık kümenin bir parçası değil. Ancak, bağımsız bir kuyruk yöneticisi olarak işlev görür.

## Sonraki adım

Bu değişikliklerin sonucu, kümenin geri kalan üyelerinde aşağıdaki komutu vererek doğrulanabilir:

```
DISPLAY CLUSQMGR(LONDON)
```

Kuyruk yöneticisi, otomatik olarak tanımlanan küme gönderen kanalları duruncaya kadar görüntülenmeye devam eder. Bunun gerçekleşmesini bekleyebilir ya da aşağıdaki komutu vererek etkin yönetim ortamlarını izlemeye devam edebilirsiniz:

```
DISPLAY CHANNEL (INVENTORY . LONDON)
```

Değişiklikler kümeye yayıldıktan ve bu kuyruk yöneticisine başka ileti teslim edildikten sonra, LONDON üzerindeki CLUSRCVR kanalını silin:

```
DELETE CHANNEL (INVENTORY . LONDON)
```

Kaldırılan kuyruk yöneticisi, "[Kümeye kuyruk yöneticisi eklenmesi](#)" sayfa 308 içinde açıklandığı gibi daha sonraki bir noktada kümeye yeniden eklenebilir. Kaldırılan kuyruk yöneticisi, kümenin geri kalan üyelerine ilişkin bilgileri 90 güne kadar önbelleğe almaya devam eder. Bu önbelleğin süresi doluncaya kadar beklememeyi tercih ediyorsanız, bu önbellek "[Bir kuyruk yöneticisinin küme öncesi durumuna geri yüklenmesi](#)" sayfa 356 içinde açıklandığı gibi zorla kaldırılabilir.

## İlgili başvurular

[KANAL SILME](#) (kanal silme)

[DISPLAY CHANNEL](#) (kanal tanımını görüntüle)

[DISPLAY CHSTATUS](#) (KANAL DURUMUNU GÖRÜNTÜLE)

[DISPLAY CLUSQMGR](#) (küme kuyruğu yöneticileri için kanal bilgilerini görüntüler)

[STOP CHANNEL](#) (kanalı durdur)

[RESET CLUSTER](#) (küme ilk durumuna getir)

## Bir kuyruk yöneticisinin küme öncesi durumuna geri yüklenmesi

Bir kuyruk yöneticisi bir kümeden kaldırıldığında, geri kalan küme üyeleri hakkında bilgi sahibi olur. Bu bilgi sonunda sona erer ve otomatik olarak silinir. Ancak hemen silmeyi tercih ederseniz, bu konudaki adımları kullanabilirsiniz.



## Başlamadan önce

Kuyruk yöneticisinin kümeden kaldırıldığı ve artık kümede herhangi bir iş gerçekleştirmediği varsayılır. Örneğin, kuyrukları artık kümeden ileti almıyor ve hiçbir uygulama bu kuyruklara ileti gelmesini beklemiyor.

## Bu görev hakkında

Bir kuyruk yöneticisi bir kümeden kaldırıldığında, geri kalan küme üyeleri hakkında 90 güne kadar bilgi sahibi olur. Bunun, özellikle kuyruk yöneticisi kümeye hızlı bir şekilde yeniden katıldıysa, sistem yararları olabilir. Bu bilginin süresi sonunda sona erdiğinde, otomatik olarak silinir. Ancak, bu bilgileri el ile silmeyi tercih edebileceğiniz bazı nedenler vardır. Örneğin:

- Daha önce küme kaynaklarını kullanan bu kuyruk yöneticisindeki her uygulamayı durdurduğınızı doğrulamak isteyebilirsiniz. Geri kalan küme üyelerinin bilgileri sona erinceye kadar, bu tür herhangi bir uygulama bir iletim kuyruğuna yazmaya devam eder. Küme bilgisi silindikten sonra, böyle bir uygulama küme kaynaklarını kullanmayı denediğinde sistem bir hata iletisi oluşturur.
- Kuyruk yöneticisine ilişkin durum bilgilerini görüntülerken, kalan küme üyeleriyle ilgili süre bitimine ilişkin bilgileri görmemeyi tercih edersiniz.

Bu görev, örnek olarak INVENTORY kümesini kullanır. LONDON kuyruk yöneticisi, “[Bir kümeden kuyruk yöneticisinin kaldırılması: en iyi uygulama](#)” sayfa 352’inde açıklandığı gibi INVENTORY kümesinden kaldırılmıştır. Kümenin geri kalan üyelerine ilişkin bilgileri silmek için, LONDON kuyruk yöneticisinde aşağıdaki komutları verin.

## Yordam

1. Kümedeki diğer kuyruk yöneticilerinin tüm belleğini bu kuyruk yöneticisinden kaldır:

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

2. Tüm küme kaynakları gidene kadar kuyruk yöneticisini izleyin:

```
DISPLAY CLUSQMgr(*) CLUSTER(INVENTORY)  
DISPLAY QCLUSTER(*) CLUSTER(INVENTORY)  
DISPLAY TOPIC(*) CLUSTER(INVENTORY)
```

## İlgili kavramlar

[Kümeler](#)

[Küme bileşenleri](#)

## İlgili başvurular

[Kümeleme ve dağıtılmış kuyruğa alma karşılaştırması](#)

## Kuyruk yöneticisinin bakımı

Bakım gerçekleştirmek için bir kümede kuyruk yöneticisini askıya alın ve sürdürün.

## Bu görev hakkında

Zaman zaman, bir kümenin parçası olan bir kuyruk yöneticisinde bakım gerçekleştirmeniz gerekebilir. Örneğin, kuyruklarındaki verilerin yedeklerini almanız ya da yazılıma düzeltmeler uygulamanız gerekebilir. Kuyruk yöneticisi herhangi bir kuyruğu barındırıyorsa, etkinlikleri askıya alınmalıdır. Bakım tamamlandığında, etkinlikleri sürdürülebilir.

## Yordam

1. SUSPEND QMgr **runmqsc** komutunu vererek bir kuyruk yöneticisini askıya alın:

```
SUSPEND QMGR CLUSTER(SALES)
```

SUSPEND **runmqsc** komutu, SALES kümesindeki kuyruk yöneticilerine bu kuyruk yöneticisinin askıya alındığını bildirir.

SUSPEND QMGR komutunun amacı, diğer kuyruk yöneticilerine, mümkünse bu kuyruk yöneticisine ileti göndermekten kaçınmalarını önermektir. Bu, kuyruk yöneticisinin devre dışı olduğu anlamına gelmez. Bu kuyruk yöneticisi tarafından işlenmesi gereken bazı iletiler, örneğin, bu kuyruk yöneticisi kümelenmiş bir kuyruğun tek anasistemi olduğunda, bu iletiye gönderilmeye devam edilir.

Kuyruk yöneticisi askıya alındığında, iş yükü yönetimi yordamları ileti göndermekten kaçınır. Bu kuyruk yöneticisi tarafından işlenmesi gereken iletiler, yerel kuyruk yöneticisi tarafından gönderilen iletileri içerir.

IBM MQ , mümkün olduğunda yerel kuyruk yöneticisini seçmek yerine hangi hedeflerin uygun olduğunu belirlemek için bir iş yükü dengeleme algoritması kullanır.

- a) SUSPEND QMGR komutunda FORCE seçeneğini kullanarak bir kuyruk yöneticisinin askıya alınmasını zorlayın:

```
SUSPEND QMGR CLUSTER(SALES) MODE(FORCE)
```

MODE (FORCE) , kümedeki diğer kuyruk yöneticilerinden gelen tüm kanalları zorla durdurur. MODE (FORCE) belirtmezseniz, varsayılan MODE (QUIESCE) geçerli olur.

2. Gerekli bakım görevlerini gerçekleştirin.
3. RESUME QMGR **runmqsc** komutunu vererek kuyruk yöneticisini sürdürün:

```
RESUME QMGR CLUSTER(SALES)
```

## Sonuçlar

RESUME **runmqsc** komutu, tüm havuzlara kuyruk yöneticisinin yeniden kullanılabilir olduğunu bildirir. Tam havuz kuyruğu yöneticileri, bu bilgileri, bu kuyruk yöneticisiyle ilgili bilgilerde güncelleme isteğinde bulunan diğer kuyruk yöneticilerine dağıtır.

## Küme iletim kuyruğunun bakımı

Küme iletim kuyruklarını kullanılabilir tutmak için her türlü çabayı kullanın. Kümelerin performansı için çok önemlidir. **z/OS** z/OS işletim sistemlerinde, bir küme iletim kuyruğunun INDXTYPE değerini CORRELID olarak ayarlayın.

## Başlamadan önce

- Küme iletim kuyruğunun dolu olmadığından emin olun.
- ALTER **runmqsc** komutunu, devre dışı bırakılması ya da yanlışlıkla devre dışı bırakılması için ayarlamak üzere çalıştırmamaya dikkat edin.
- Küme iletim kuyruğunun **z/OS** (örneğin, z/OS sayfa kümeleri) üzerinde saklandığı ortamın dolu olmadığından emin olun.

## Bu görev hakkında

**z/OS**

Aşağıdaki yordam yalnızca z/OS için geçerlidir.

## Yordam

Küme iletim kuyruğunun INDXTYPE değerini CORRELID olarak ayarlayın.

## Küme kuyruk yöneticisinin yenilenmesi

REFRESH CLUSTER komutunu kullanarak otomatik tanımlı kanalları ve otomatik tanımlı küme nesnelərini yerel havuzdan kaldırabilirsiniz. Hiçbir ileti kaybolmaz.

### Başlamadan önce

IBM Destek Merkeziniz tarafından komutu kullanmanız istenebilir. Dikkatli bir şekilde dikkate alınmadan komutu kullanmayın. Örneğin, **REFRESH CLUSTER** komutunun büyük kümeler için kullanımı, devam ederken kümede kesintiye neden olabilir ve bundan sonra, küme nesneleri otomatik olarak ilgili tüm kuyruk yöneticilerine durum güncellemeleri gönderdiğinde, 27 günlük aralıklarla kümede kesintiye neden olabilir. Bkz. [Clustering: Using REFRESH CLUSTER en iyi uygulamaları](#).

### Bu görev hakkında

Kuyruk yöneticisi, bir kümede yeni bir başlangıç yapabilir. Olağan koşullarda, REFRESH CLUSTER komutunu kullanmanız gerekmez.

## Yordam

Otomatik olarak tanımlanan küme kuyruğu yöneticisini ve kuyruk nesnelərini yerel havuzdan kaldırmak için bir kuyruk yöneticisinden REFRESH CLUSTER **MQSC** komutunu verin.

Komut yalnızca diğer kuyruk yöneticilerine başvuran nesneleri kaldırır, yerel kuyruk yöneticisiyle ilgili nesneleri kaldırmaz. Komut, otomatik tanımlı kanalları da kaldırır. Küme iletim kuyruğunda ileti olmayan ve tam havuz kuyruk yöneticisine bağlı olmayan kanalları kaldırır.

## Sonuçlar

REFRESH CLUSTER komutu, bir kuyruk yöneticisinin tam havuz içeriğine göre soğuk başlatılmasına olanak sağlar. IBM MQ , kuyruklarınızdan veri kaybına uğramamanızı sağlar.

### İlgili bilgiler

[Kümeleme: REFRESH CLUSTER en iyi uygulamalarını kullanma](#)

## Küme kuyruk yöneticisinin kurtarılması

REFRESH CLUSTER **runmqsc** komutunu kullanarak bir kuyruk yöneticisine ilişkin küme bilgilerini güncel olarak getirin. Bir kuyruk yöneticisini anlık yedekten kurtardıktan sonra bu yordamı izleyin.

### Başlamadan önce

Bir küme kuyruk yöneticisini anlık yedekten geri yüklediniz.

### Bu görev hakkında

Kümedeki bir kuyruk yöneticisini kurtarmak için kuyruk yöneticisini geri yükleyin ve REFRESH CLUSTER **runmqsc** komutunu kullanarak küme bilgilerini güncel olarak getirin.

**Not:** Büyük kümeler için, **REFRESH CLUSTER** komutunun kullanılması, devam ederken kümede kesintiye neden olabilir ve bundan sonra küme nesneleri otomatik olarak ilgili tüm kuyruk yöneticilerine durum güncellemelerini gönderdiğinde, 27 günlük aralıklarla kümede kesintiye neden olabilir. Bkz. [Büyük bir kümede yenilenmesi kümenin performansını ve kullanılabilirliğini etkileyebilir](#).

## Yordam

Kuyruk yöneticisinin katıldığı tüm kümeler için geri yüklenen kuyruk yöneticisinde REFRESH CLUSTER komutunu verin.

## Sonraki adım

REFRESH CLUSTER komutunu başka bir kuyruk yöneticisinde çalıştırmanıza gerek yoktur.

## İlgili kavramlar

[Kümeleme: REFRESH CLUSTER en iyi uygulamalarını kullanma](#)

## Küme kanallarının kullanılabilirlik için yapılandırılması

Aralıklı ağ durdurmaları varsa küme kanallarının sorunsuz bir şekilde çalışmasını sağlamak için iyi yapılandırma uygulamalarını izleyin.

## Başlamadan önce

Kümeler, kanalları tanımlama gereksinimini ortadan kaldırıyor, ancak yine de kanalları korumanız gerekiyor. Aynı kanal teknolojisi, dağıtılmış kuyruğa alma işlemi için kullanıldığı gibi, bir kümedeki kuyruk yöneticileri arasında iletişim için de kullanılır. Küme kanalları hakkında bilgi sahibi olmak için aşağıdakiler gibi konulara aşina olmanız gerekir:

- Kanallar nasıl çalışır?
- Durumlarını bulma
- Kanal çıkışlarının kullanılması

## Bu görev hakkında

Aşağıdaki noktalara özel önem vermek isteyebilirsiniz:

## Yordam

Küme kanallarını yapılandırırken aşağıdaki noktaları göz önünde bulundurun

- Çok fazla sağlıklı işletim bildirimi ya da canlı tutma akışı olmayan küme gönderen kanallarda ve küme alıcı kanallarında HBINT ya da KALINT değerlerini seçin. Ağınız bazen yavaşlar ve bu uzunlukta gecikmeler ortaya çıkarsa, yaklaşık 10 saniyeden daha kısa bir süre yanlış hatalar verir.
- BATCHHB değerini, başarısız olan bir kanalda belirsiz durumda olduğu için ayrılmış bir iletiye neden olma penceresini azaltacak şekilde ayarlayın. Toplu iş daha uzun süre doldurulursa, başarısız bir kanalda belirsiz bir toplu iş oluşma olasılığı daha yüksektir. Kanaldaki ileti trafiği, ileti ayırmaları arasında uzun sürelerle dağınıksa, başarısız bir toplu iş daha olasıdır.
- Bir kanalın küme gönderen ucu başarısız olursa ve sağlıklı işletim bildirimi ya da canlı tutma işlemi arızayı saptamadan önce yeniden başlatmayı denerse bir sorun ortaya çıkar. Kanalın küme alıcı ucu etkin olarak kaldıysa, kanal göndericinin yeniden başlatılması reddedilir. Hatayı önlemek için, küme gönderen kanal yeniden başlatma girişiminde bulunduğu küme alıcı kanalının sonlandırılmasını ve yeniden başlatılmasını sağlayın.

### **AçıkIBM MQ for z/OS**

**ALTER QMGR** üzerindeki **ADOPTMCA** ve **ADOPTCHK** parametrelerini kullanarak etkin kalan kanalın küme alıcı ucundaki sorunu denetleyin.

### **AçıkÇoklu platformlar**

**qm.ini** dosyasında ya da Windows Kayıt dosyasında **AdoptNewMCA**, **AdoptNewMCATimeout** ve **AdoptNewMCACheck** özniteliklerini kullanarak etkin kalan kanalın küme alıcı ucundaki sorunu denetleyin.

## Örnek

Bu ayarların hem IBM MQ for z/OS hem de IBM MQ for Multiplatforms üzerinde nasıl uygulanacağına ilişkin örnekler için bkz. [“Önerilen ayarlar” sayfa 226](#).

## Dağıtılmış ağlara ilişkin zamanuyumsuz komutların tamamının denetlenip tamamlanmadığının denetlenmesi

Birçok komut, dağıtılmış bir ağda kullanıldığında zamanuyumsuzdur. Komuta ve ağ durumuna bağlı olarak, komutun tamamlanması çok uzun sürebilir. Kuyruk yöneticisi tamamlanınca bir ileti yayınlamaz; bu nedenle, komutun tamamlandığını denetlemenin başka yolları da gerekir.

### Bu görev hakkında

Bir kümede yapacağınız hemen hemen her yapılandırma değişikliği zamanuyumsuz olarak tamamlanır. Bunun nedeni, kümeler içinde çalışan iç yönetim ve güncelleme döngüleridir. Yayınlama/abone olma sıradüzenleri için, abonelikleri etkileyen yapılandırma değişikliklerinin zamanuyumsuz olarak tamamlanması muhtemeldir. Bu, komutun adından her zaman belli değildir.

Aşağıdaki MQSC komutlarının tümü zamanuyumsuz olarak tamamlanabilir. Bu komutların her biri bir PCF eşdeğerine sahiptir ve çoğu IBM MQ Explorer içinden de kullanılabilir. İş yükü olmayan küçük bir ağ üzerinde çalıştırıldığında, bu komutlar genellikle birkaç saniye içinde tamamlanır. Bununla birlikte, daha büyük ve daha yoğun ağlar için durum böyle değildir. **REFRESH CLUSTER** komutu, özellikle aynı anda birden çok kuyruk yöneticilerine verildiğinde çok daha uzun sürebilir.

Bu komutların tamamlandığını güvenmek için, uzak kuyruk yöneticilerindeki beklenen nesnelere var olup olmadığını denetleyin.

### Yordam

- [ALTER QMGR](#)

[ALTER QMGR PARENT](#) komutunda, istenen üst ilişkinin durumunu izlemek için `DISPLAY PUBSUB TYPE(PARENT) ALL` komutunu kullanın.

[ALTER QMGR REPOS](#) ve [ALTER QMGR REPOSNL](#) komutları için, tamamlandığını doğrulamak üzere `DISPLAY CLUSQMGR QMTYPE` komutunu kullanın.

- [KANALI TANIMLAYIN](#), [KANALI DEĞİŞTİRVE](#) [KANALI SIL](#)

[ALTER CHANNEL](#) parametresizselgesinde listelenen tüm parametreler için, değişiklikler kümeye yayıldığında izlemek için `DISPLAY CLUSQMGR` komutunu kullanın.

- [DEFINE NAMELIST](#), [ALTER NAMELIST](#) ve [DELETE NAMELIST](#).

Bir **QMGR** nesnesinin **CLUSNL** özneliğinde **NAMELIST** kullanıyorsanız, bir kuyruk ya da küme kanalı o nesneyi etkileyebilir. Etkilenen nesne için uygun şekilde izleyin.

`SYSTEM.QPUBSUB.QUEUE.NAMELIST` üzerinde yapılan değişiklikler, bir yayınlama/abone olma sıradüzeninde yetkili sunucu aboneliklerinin oluşturulmasını ya da iptal edilmesini etkileyebilir. Bunu izlemek için `DISPLAY SUB SUBTYPE(PROXY)` komutunu kullanın.

- [DEFINE kuyrukları](#), [ALTER kuyrukları](#) ve [DELETE kuyrukları](#).

[DISPLAY QUEUE](#) komutu tarafından döndürülebilir parametreler `çizelgesinde` listelenen tüm parametreler için, değişiklikler kümeye yayıldığında izlemek için `DISPLAY QCLUSTER` komutunu kullanın.

- [DEFINE SUB](#) ve [DELETE SUB](#)

Bir konu dizgisinde ilk aboneliği tanımladığınızda, bir yayınlama/abone olma sıradüzeninde ya da yayınlama/abone olma kümesinde yetkili sunucu abonelikleri yaratabilirsiniz. Benzer şekilde, bir konu dizgisindeki son aboneliği sildiğinizde, bir yayınlama/abone olma sıradüzeninde ya da yayınlama/abone olma kümesinde yetkili sunucu aboneliklerini iptal edebilirsiniz.

Bir aboneliği tanımlayan ya da silen bir komutun tamamlanıp tamamlanmadığını denetlemek için, beklenen yetkili sunucu aboneliğinin dağıtılmış ağdaki diğer kuyruk yöneticilerine var olup olmadığını denetleyin. Bir kümede *doğrudan yöneltme* kullanıyorsanız, beklenen yetkili sunucu aboneliğinin kümedeki diğer kısmi havuzlarda var olup olmadığını denetleyin. Bir kümede *konu anasistem yöneltmesi* kullanıyorsanız, beklenen yetkili sunucu aboneliğinin eşleşen konu anasistemlerinde var olup olmadığını denetleyin. Aşağıdaki MQSC komutunu kullanın:

```
DISPLAY SUB(*) SUBTYPE(PROXY)
```

Bir kümede ya da sıradüzende yayınlandığında, aşağıdaki eşdeğer abone olma ve MQI aboneliğini kaldırma çağrılarını için aynı denetimi kullanın:

- MQSUBkomutunu kullanarak abone olun.
- MQCO\_REMOVE\_SUB ile MQCLOSE komutunu kullanarak abonelikten çıkın.
- KONU TANIMLAYIN, ALTER TOPICve DELETE TOPIC

Kümelenmiş bir konuyu tanımlayan, değiştiren ya da silen bir komutun tamamlandığını denetlemek için, konuyu kümedeki diğer kısmi havuzlarda görüntüleyin ( *doğrudan yöneltme* kullanıyorsanız) ya da diğer konu anasistemlerinde ( *konu anasistemi yöneltmesi* kullanıyorsanız).

DISPLAY TOPIC komutuyla döndürülebilir parametrelerçizelgesinde listelenen tüm parametreler için, değişiklikler kümeye yayıldığında izlemek için DISPLAY TCLUSTER komutunu kullanın.

**Not:**

- **CLUSTER** parametresi, bir yayınlama/abone olma kümesinde yetkili sunucu aboneliklerinin yaratılmasını ya da iptal edilmesini etkileyebilir.
- **PROXYSUB** ve **SUBSCOPE** parametreleri, bir yayınlama/abone olma sıradüzeninde yetkili sunucu aboneliklerinin oluşturulmasını ya da iptal edilmesini ya da küme yayınlanmasını/abone olunmasını etkileyebilir.
- Bunu izlemek için DISPLAY SUB SUBTYPE(PROXYSUB) komutunu kullanın.
- KÜMEYİ YENİLE

**REFRESH CLUSTER** komutunu çalıştırıyorsanız, küme komut kuyruğu derinliğini yoklayın. Nesneleri aramadan önce sıfıra ulaşmasını ve sıfırda kalmasını bekleyin.

1. Küme komut kuyruğu derinliğinin sıfır olup olmadığını denetlemek için aşağıdaki MQSC komutunu kullanın.

```
DISPLAY QL(SYSTEM.CLUSTER.COMMAND.QUEUE) CURDEPTH
```

2. Kuyruk derinliği sıfıra ulaşınca kadar denetimi yineleyin ve sonraki denetimde sıfır olarak kalır.

**REFRESH CLUSTER** komutu nesnelere kaldırır ve yeniden yaratır ve büyük yapılandırmalarda tamamlanması önemli bir zaman alabilir. Bkz. REFRESH CLUSTER yayınlama/abone olma kümeleri için önemli noktalar.

- QMGR ' YI YENİLE TIP (PROXYSUB)

**REFRESH QMGR TYPE(PROXYSUB)** komutunun tamamlandığını denetlemek için, yetkili sunucu aboneliklerinin dağıtılmış ağdaki diğer kuyruk yöneticilerine düzeltilip düzeltilmediğini denetleyin. Bir kümede *doğrudan yöneltme* kullanıyorsanız, yetkili sunucu aboneliklerinin kümedeki diğer kısmi havuzlarda düzeltilip düzeltilmediğini denetleyin. Bir kümede *konu anasistem yöneltmesi* kullanıyorsanız, beklenen yetkili sunucu aboneliklerinin eşleşen konu anasistemlerinde düzeltilip düzeltilmediğini denetleyin. Aşağıdaki MQSC komutunu kullanın:

```
DISPLAY SUB(*) SUBTYPE(PROXYSUB)
```

- KÜMEYİ SIFIRLAYIN

**RESET CLUSTER** komutunun tamamlanıp tamamlanmadığını denetlemek için DISPLAY CLUSQMGRCOMUTUNU kullanın.

- [QMGR 'YI SIFIRLAYIN TIPI \(PUBSUB\)](#)

**RESET QMGR** komutunun tamamlanıp tamamlanmadığını denetlemek için DISPLAY PUBSUB TYPE (PARENT | CHILD) komutunu kullanın.

**Not: RESET QMGR** komutu, yayınlama/abone olma sıradüzeninde ya da yayınlama/abone olma kümesinde yetkili sunucu aboneliklerinin iptal edilmesine neden olabilir. Bunu izlemek için DISPLAY SUB SUBTYPE (PROXYSUB) komutunu kullanın.

- Ayrıca, komutlar tamamlandığında, sıfır kuyruk derinliğine doğru eğilimli diğer sistem kuyruklarını da izlemek isteyebilirsiniz.  
Örneğin, SYSTEM . INTER . QMGR . CONTROL kuyruğunu ve SYSTEM . INTER . QMGR . FANREQ kuyruğunu izlemek isteyebilirsiniz. Bkz. [Kümelere içinde yetkili sunucu aboneliği trafiğini izleme ve Yayınlama/abone olma ağlarında üreticileri ve tüketicileri dengeleme](#).

## Sonraki adım

Bu denetimler, zamanuyumsuz bir komutun tamamlandığını doğrulamazsa, bir hata oluşmuş olabilir. Araştırmak için, önce komutun verildiği kuyruk yöneticisine ilişkin günlüğü denetleyin, ardından (küme için) kümenin tam havuz günlüklerini denetleyin.

## İlgili başvurular

 z/OS üzerinde CLUSTER komutlarının zamanuyumsuz davranışı

## İletileri kümelere/kümelere yönlendirme

Kümelere dış kuyruk yöneticilerine ve diğer kümelere bağlamak için kuyruk diğer adlarını, kuyruk yöneticisi diğer adlarını ve uzak kuyruk tanımlamalarını kullanın.

İletilerin kümelere/kümelere yönlendirilmesine ilişkin ayrıntılar için aşağıdaki alt konulara bakın:

### İlgili kavramlar

#### Kümelere

#### Bir kümenin bileşenleri

“Kuyruk yöneticisi diğer adları ve kümelere” sayfa 377

Bir kümeye ileti gönderirken ya da gönderirken kuyruk yöneticilerinin adını ve bir kümeye gönderilen iş yükü dengeleme iletilerini gizlemek için kuyruk yöneticisi diğer adlarını kullanın.

“Kuyruk diğer adları ve kümelere” sayfa 380

Bir küme kuyruğunun adını gizlemek, bir kuyruğu kümelemek, farklı öznitelikleri benimsemek ya da farklı erişim denetimlerini benimsemek için kuyruk diğer adlarını kullanın.

“Yanıt kuyruğu diğer adları ve kümelere” sayfa 380

Yanıtlama kuyruğu diğer adı tanımı, yanıt bilgilerine ilişkin diğer adları belirlemek için kullanılır.

Yanıtlanacak kuyruk diğer adı tanımlamaları, dağıtılmış bir kuyruğa alma ortamındaki gibi kümelere birlikte kullanılabilir.

### İlgili görevler

“Kuyruk yöneticisi kümesinin yapılandırılması” sayfa 283

Kümelere, hem ilk yapılandırmayı hem de devam eden yönetimi basitleştiren bir şekilde birbirine bağlanan kuyruk yöneticileri için bir mekanizma sağlar. Küme bileşenleri tanımlayabilir ve kümelere oluşturabilir ve yönetebilirsiniz.

“Yeni bir küme ayarlama” sayfa 297

Örnek kümeyi ayarlamak için bu yönergeleri izleyin. Ayrı yönergeler, kümenin TCP/IP, LU 6.2 üzerinde ve tek bir iletim kuyruğuyla ya da birden çok iletim kuyruğuyla ayarlanmasını açıklar. Bir kuyruk yöneticisinden diğerine ileti göndererek kümeyi sınavın.

### İlgili başvurular

[Kümeleme ve dağıtılmış kuyruğa alma karşılaştırması](#)

## ***Kümeye istek/yanıt yapılandırılması***

Bir kümenin dışındaki bir kuyruk yöneticisinden bir istek/yanıt iletisi yolu yapılandırın. Kümeyle iletişim yolu olarak bir ağ geçidi kuyruk yöneticisini kullanarak kümenin iç ayrıntılarını gizleyin.

### **Başlamadan önce**

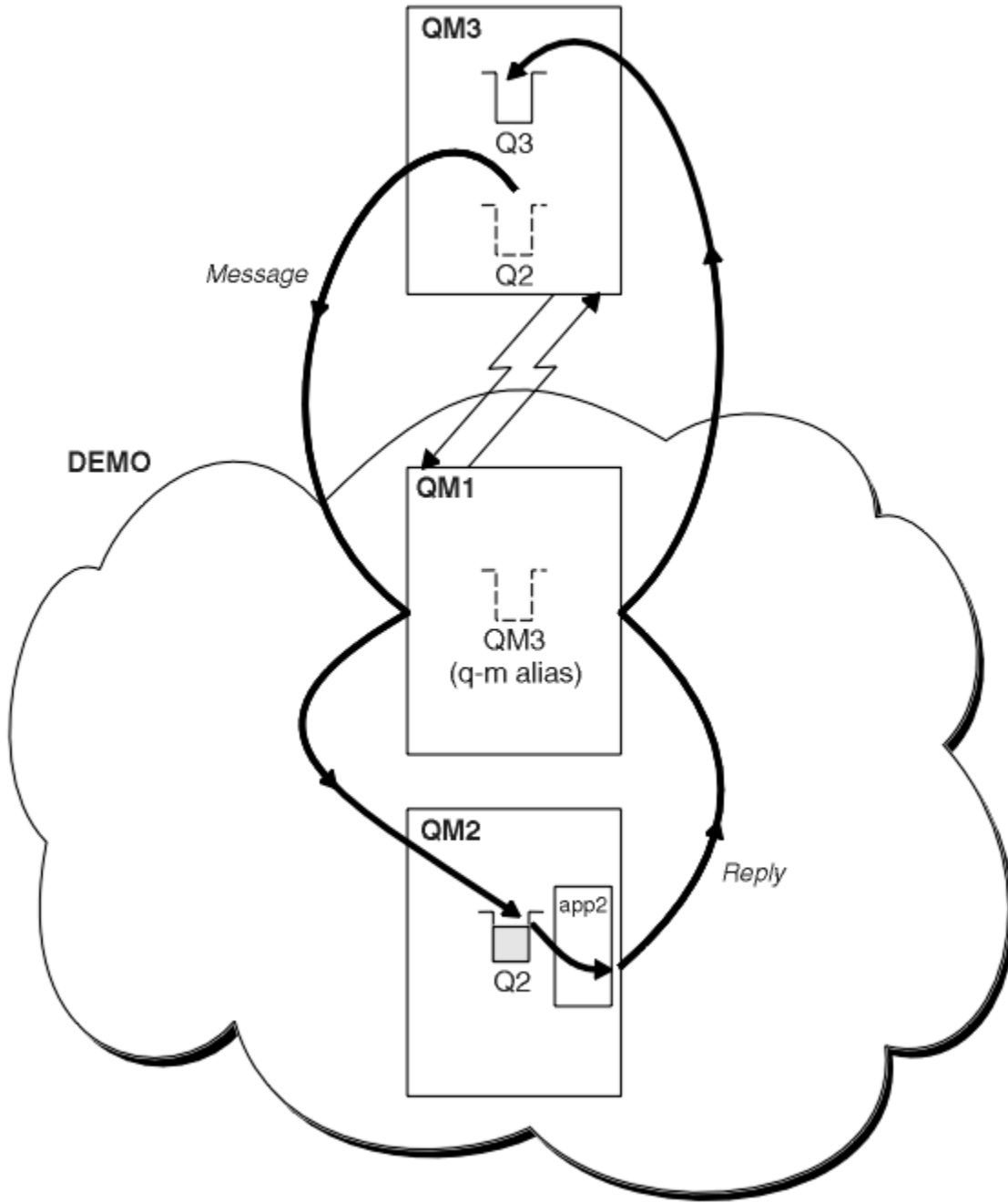
Şekil 53 sayfa 365 , DEM0adlı kümenin dışında olan QM3 adlı bir kuyruk yöneticisini gösterir. QM3 , kümeleri desteklemeyen bir IBM MQ ürününde bir kuyruk yöneticisi olabilir. QM3 , aşağıdaki gibi tanımlanan Q3adlı bir kuyruğu barındırır:

```
DEFINE QLOCAL(Q3)
```

Kümenin içinde QM1 ve QM2adlı iki kuyruk yöneticisi vardır. QM2 , aşağıdaki gibi tanımlanan Q2adlı bir küme kuyruğunu barındırır:

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO)
```





Şekil 53. Kümenin dışına bir kuyruk yöneticisinden yerleştirme

### Bu görev hakkında

İstek ve yanıt iletilerine ilişkin yolu ayarlamak için yordamdaki önerileri izleyin.

### Yordam

1. İstek iletilisini kümeye gönderin.

Kümenin dışındaki kuyruk yöneticisinin, kümenin içindeki Q2 at QM2kuyruğuna bir iletiyi nasıl koyduğunu göz önünde bulundurun. Kümenin dışındaki bir kuyruk yöneticisinin, iletileri koyduğu kümedeki her bir kuyruk için bir QREMOTE tanımlaması olmalıdır.

- a) QM3üzerinde Q2 için bir uzak kuyruk tanımlayın.

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(QM2) XMITQ(QM1)
```

QM3 bir kümenin parçası olmadığından, dağıtılmış kuyruğa alma tekniklerini kullanarak iletişim kurulmalıdır. Bu nedenle, QM1'e bir gönderen kanalı ve bir iletim kuyruğu da olmalıdır. QM1 için karşılık gelen bir alıcı kanal gerekir. Kanallar ve iletim kuyrukları [Şekil 53 sayfa 365](#) içinde açık bir şekilde gösterilmez.

Örnekte, QM3 adresindeki bir uygulama, Q2 adresine ileti göndermek için bir MQPUT çağırısı yayımlar. QREMOTE tanımı, iletinin QM1 iletim kuyruğundan ileti alan gönderen kanalı kullanılarak QM2 adresindeki Q2 adresine yönlendirilmesine neden olur.

## 2. Kümeden yanıt iletilisini alın.

Kümenin dışındaki bir kuyruk yöneticisine yanıt vermek üzere bir dönüş yolu yaratmak için kuyruk yöneticisi diğer adını kullanın. Ağ geçidi (QM1), kümenin dışındaki kuyruk yöneticisi için bir kuyruk yöneticisi diğer adını (QM3) tanıtır. Küme özniteliğini QM3 için bir kuyruk yöneticisi diğer adı tanımlamasına ekleyerek, küme içindeki kuyruk yöneticilerine QM3 değerini tanıtır. Kuyruk yöneticisi diğer adı tanımlaması uzak kuyruk tanımlaması gibi, ancak boş RNAME.

a) QM1 üzerinde QM3 için bir kuyruk yöneticisi diğer adı tanımlayın.

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

Yanıtları QM1 'dan QM3' e iletmek için kullanılan iletim kuyruğuna ilişkin ad seçimini göz önünde bulundurmalıyız. QREMOTE tanımında örtük olarak XMITQ özniteliği, iletim kuyruğunun adının QM3 olduğunu belirtir. Ancak QM3 , kuyruk yöneticisi diğer adını kullanarak kümenin geri kalanına duyurmayı beklediğimiz adla aynıdır. IBM MQ , hem iletim kuyruğuna hem de kuyruk yöneticisi diğer adına aynı adı vermenize izin vermez. Bir çözüm, iletileri kuyruk yöneticisi diğer adına farklı bir adla QM3 ' e iletmek için bir iletim kuyruğu yaratmaktır.

b) QREMOTE tanımında iletim kuyruğu adını belirtin.

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO) XMITQ(QM3.XMIT)
```

Yeni kuyruk yöneticisi diğer adı, QM3 . XMIT adlı yeni iletim kuyruğunu QM3 kuyruk yöneticisi diğer adıyla eşler. Basit ve doğru bir çözüm, ama tamamen tatmin edici değil. Bu, hedef kuyruk yöneticisiyle aynı adı verilen iletim kuyruklarına ilişkin adlandırma kuralını bozdu. İletim kuyruğu adlandırma kuralını koruyan alternatif çözümler var mı?

İstekte bulunan kişi, QM3'den gönderilen istek iletilisinde QM3 ' i yanıt kuyruğu yöneticisi adı olarak varsayılan olarak geçirdiğinden sorun ortaya çıkar. QM2 üzerindeki sunucu, yanıtlarında QM3 adresine QM3 yanıtlanacak kuyruk yöneticisi adını kullanır. The solution required QM1 to advertise QM3 as the queue manager alias to return reply messages to and prevented QM1 from using QM3 as the name of the transmission queue.

Yanıt kuyruğu yöneticisi adı olarak QM3 'in varsayılan olarak sağlanması yerine, QM3 üzerindeki uygulamaların yanıt iletileri için QM1 ' e bir yanıt kuyruğu yöneticisi diğer adı geçirmeleri gerekir. Ağ geçidi kuyruk yöneticisi QM1 , iletim kuyruğunun adıyla çakışmayı önlemek amacıyla, QM3 yerine QM3 ' e yanıt vermek için kuyruk yöneticisi diğer adını duyurur.

c) QM1 üzerinde QM3 için bir kuyruk yöneticisi diğer adı tanımlayın.

```
DEFINE QREMOTE(QM3.ALIAS) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

Yapılandırma komutlarında iki değişiklik yapılması gerekir.

i) Şu anda QM1 adresindeki QREMOTE , QM3 . ALIAS kuyruk yöneticisi diğer adını kümenin geri kalanına tanıtır ve gerçek kuyruk yöneticisinin adıyla eşler QM3 . QM3 , yanıt kuyruklarının geri gönderileceği iletim kuyruğunun adıdır QM3

ii) İstemci uygulaması, istek iletisini oluştururken yanıt kuyruğu yöneticisinin adı olarak QM3 . ALIAS belirtmelidir. İstemci uygulamasına iki yoldan birini kullanarak QM3 . ALIAS sağlayabilirsiniz.

- MQMDiçinde MQPUT tarafından oluşturulan yanıt kuyruğu yöneticisi adı alanında QM3 . ALIAS kodu. Yanıtlar için dinamik bir kuyruk kullanıyorsanız, bunu bu şekilde yapmalısınız.
- Yanıtın gönderileceği kuyruk adını sağlarken yanıt kuyruğu yerine, bir yanıt kuyruğu diğer adı ( Q3 . ALIAS) kullanın.

```
DEFINE QREMOTE(Q3.ALIAS) RNAME(Q3) RQMNAME(QM3.ALIAS)
```

## Sonraki adım

**Not: AMQSREQ0** ile yanıt kuyruğu diğer adlarının kullanımını gösteremezsiniz. Parametre 3 'te sağlanan kuyruk adını ya da varsayılan SYSTEM . SAMPLE . REPLY model kuyruğunu kullanarak yanıt kuyruğunu açar. MQPUT için yanıtlanacak kuyruk yöneticisi diğer adını adlamak üzere, yanıtlanacak kuyruk diğer adını içeren başka bir parametre sağlayan örneği değiştirmeniz gerekir.

### İlgili kavramlar

Kuyruk yöneticisi diğer adları ve kümeleri

Bir kümeye ileti gönderirken ya da gönderirken kuyruk yöneticilerinin adını ve bir kümeye gönderilen iş yükü dengeleme iletilerini gizlemek için kuyruk yöneticisi diğer adlarını kullanın.

Yanıt kuyruğu diğer adları ve kümeleri

Yanıtlama kuyruğu diğer adı tanımı, yanıt bilgilerine ilişkin diğer adları belirlemek için kullanılır.

Yanıtlanacak kuyruk diğer adı tanımlamaları, dağıtılmış bir kuyruğa alma ortamındaki gibi kümelerle birlikte kullanılabilir.

Kuyruk diğer adları ve kümeleri

Bir küme kuyruğunun adını gizlemek, bir kuyruğu kümelemek, farklı öznitelikleri benimsemek ya da farklı erişim denetimlerini benimsemek için kuyruk diğer adlarını kullanın.

### İlgili görevler

Kümeden istek/yanıt yapılandırılması

Kümeden kümenin dışındaki bir kuyruk yöneticisine bir istek/yanıt iletisi yolu yapılandırın. Küme içindeki bir kuyruk yöneticisinin bir ağ geçidi kuyruk yöneticisini kullanarak küme dışında nasıl iletişim kuracağına ilişkin ayrıntıları gizleyin.

Kümenin dışından iş yükü dengelemeyi yapılandırma

Bir kümenin dışındaki bir kuyruk yöneticisinden küme kuyruğunun herhangi bir kopyasına bir ileti yolu yapılandırın. Sonuç, kümenin dışından küme kuyruğunun her bir eşgörünümüne gelen iş yükü dengeleme isteklerinin sağlanmasıdır.

Kümeler arasındaki ileti yollarının yapılandırılması

Bir ağ geçidi kuyruk yöneticisini kullanarak kümeleri birbirine bağlayın. Ağ geçidi kuyruk yöneticisinde küme kuyruğu ya da küme kuyruğu yöneticisi diğer adlarını tanımlayarak kuyrukları ya da kuyruk yöneticilerini tüm kümeler tarafından görülebilir yapın.

“Küme hedef kuyruk yöneticisinin adının gizlenmesi” sayfa 367

Bir iletiyi, kuyruk yöneticisini adlandırmadan kümedeki herhangi bir kuyruk yöneticisinde tanımlı bir küme kuyruğuna yönlendirin.

Küme hedef kuyruk yöneticisinin adının gizlenmesi

Bir iletiyi, kuyruk yöneticisini adlandırmadan kümedeki herhangi bir kuyruk yöneticisinde tanımlı bir küme kuyruğuna yönlendirin.

## Başlamadan önce

- Küme içindeki kuyruk yöneticilerinin adlarını, küme dışındaki kuyruk yöneticilerine göstermekten kaçının.

- Küme içinde bir kuyruk barındıran kuyruk yöneticisine yapılan başvuruların çözülmesi, iş yükü dengeleme esnekliğini ortadan kaldırır.
  - Ayrıca, kümedeki bir kuyruğu barındıran bir kuyruk yöneticisini değiştirmenizi de zorlaştırır.
  - Diğer bir seçenek, RQMNAME yerine küme denetimcisi tarafından sağlanan bir kuyruk yöneticisi diğer adını koymaktır.
  - “Küme hedef kuyruk yöneticisinin adının gizlenmesi” sayfa 367 içinde, bir kümenin dışındaki bir kuyruk yöneticisini küme içindeki kuyruk yöneticilerinin yönetiminden eşlemek için bir kuyruk yöneticisi diğer adının kullanılması açıklanmaktadır.
- Ancak, iletim kuyruklarını adlandırmanın önerilen yolu, onlara hedef kuyruk yöneticisinin adını vermektir. İletim kuyruğunun adı, kümedeki bir kuyruk yöneticisinin adını gösterir. Hangi kuralı izleyeceğinizi seçmek zorundasınız. İletim kuyruğunu, kuyruk yöneticisi adını ya da küme adını kullanarak adlandırmayı seçebilirsiniz:

#### **İletim kuyruğunu ağ geçidi kuyruk yöneticisi adını kullanarak adlandır**

Ağ geçidi kuyruk yöneticisi adının bir küme dışındaki kuyruk yöneticilerine açıklanması, küme kuyruk yöneticisi adlarının gizlenmesi kuralı için makul bir kural dışı durumdur.

#### **Kümenin adını kullanarak iletim kuyruğunu adlandır**

İletim kuyruklarını hedef kuyruk yöneticisi adıyla adlandırma kurallarına uymuyorsanız, küme adını kullanın.

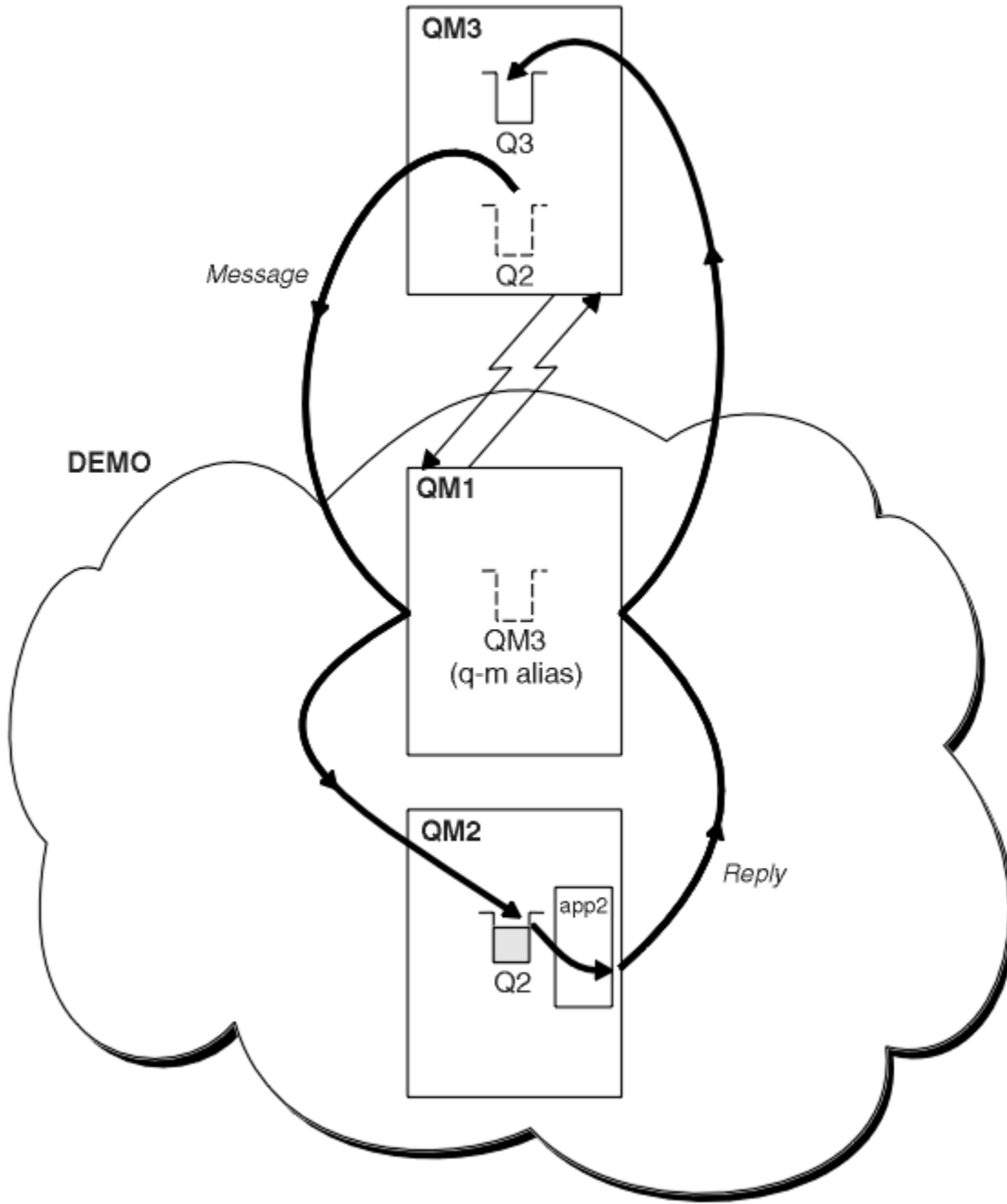
### **Bu görev hakkında**

Küme içindeki hedef kuyruk yöneticisinin adını gizlemek için “Kümeye istek/yanıt yapılandırılması” sayfa 364 görevini değiştirin.

### **Yordam**

Örnekte, bkz. Şekil 54 sayfa 369, QM1 ağ geçidi kuyruk yöneticisinde DEMO adlı bir kuyruk yöneticisi diğer adı tanımlayın:

```
DEFINE QREMOTE(DEMO) RNAME(' ') RQMNAME(' ')
```



Şekil 54. Kümenin dışına bir kuyruk yöneticisinden yerleştirme

QM1 üzerindeki QREMOTE tanımlaması, kuyruk yöneticisi diğer adını DEMO ağ geçidi kuyruk yöneticisinin tanımasını sağlar. QM3, Kuyruk yöneticisi, gerçek bir kuyruk yöneticisi adı kullanmak yerine, DEMO üzerindeki küme kuyruklarına ileti göndermek için DEMO kuyruk yöneticisi diğer adını kullanabilir.

Bir kümeye bağlanan iletim kuyruğuna ad vermek için küme adını kullanma kuralını benimserseniz, Q2 uzak kuyruk tanımı şöyle olur:

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(DEMO) XMIT(DEMO)
```

## **Sonuçlar**

DEMO üzerinde Q2 için yönlendirilen iletiler, DEMO iletim kuyruğuna yerleştirilir. İletim kuyruğundan, gönderen kanal tarafından ağ geçidi kuyruk yöneticisine ( QM1) aktarılır. Ağ geçidi kuyruk yöneticisi, iletileri kümede Q2küme kuyruğunu barındıran herhangi bir kuyruk yöneticisine yönlendirir.

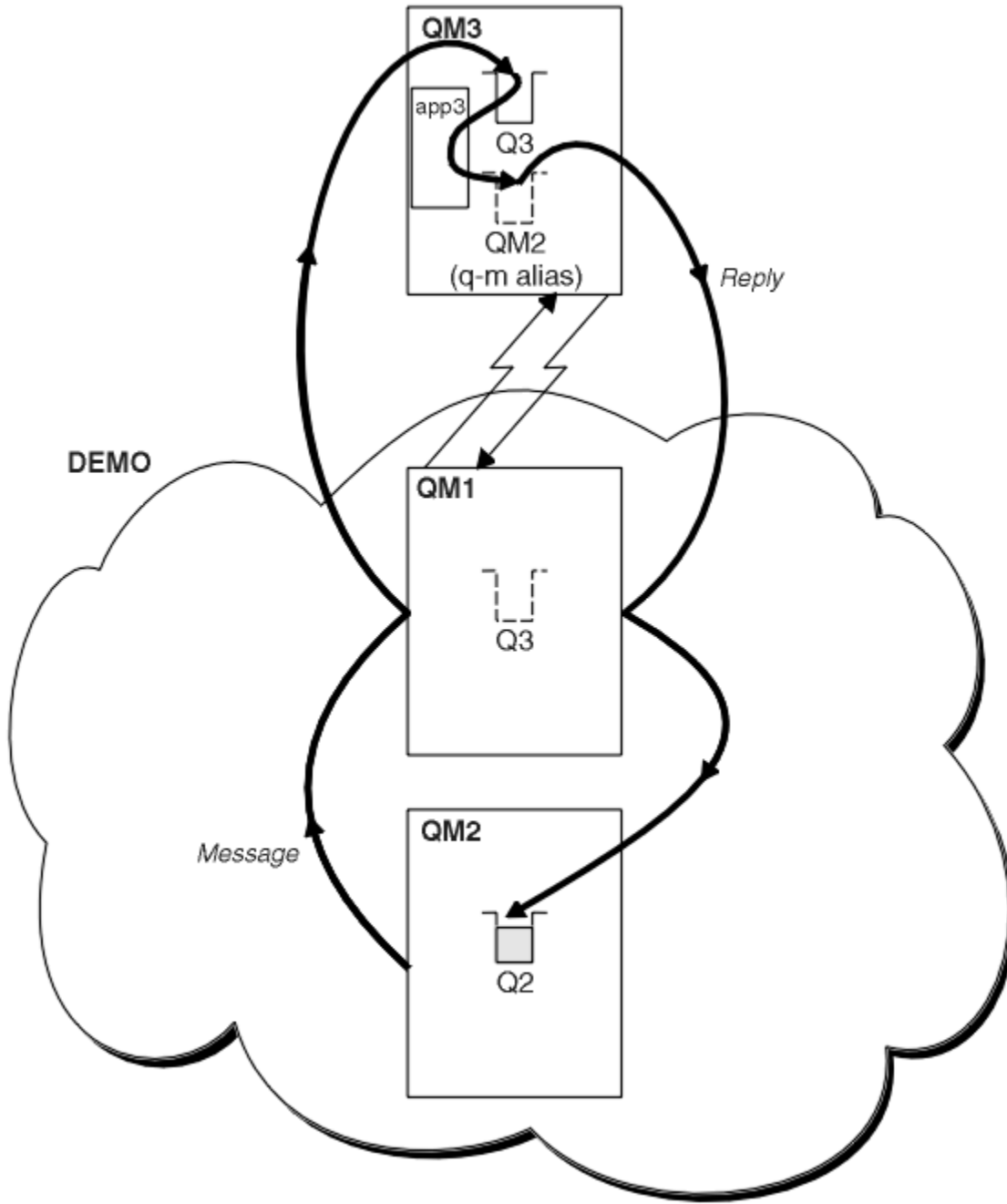
### ***Kümeden istek/yanıt yapılandırılması***

Kümeden kümenin dışındaki bir kuyruk yöneticisine bir istek/yanıt iletisi yolu yapılandırın. Küme içindeki bir kuyruk yöneticisinin bir ağ geçidi kuyruk yöneticisini kullanarak küme dışında nasıl iletişim kuracağına ilişkin ayrıntıları gizleyin.

## **Başlamadan önce**

Şekil 55 sayfa 371 , kümenin içindeki bir kuyruk yöneticisini ( QM2) gösterir DEMO. Kümenin dışındaki kuyruk yöneticisinde barındırılan bir kuyruğa ( Q3) istek gönderir. Yanıtlar, kümenin içindeki Q2 ( QM2 ) adresine döndürülür.

Kümenin dışındaki kuyruk yöneticisiyle iletişim kurmak için, kümenin içindeki bir ya da daha çok kuyruk yöneticisi ağ geçidi işlevi görür. Bir ağ geçidi kuyruk yöneticisinin, küme dışındaki kuyruk yöneticileriyle iletişim yolu vardır. Örnekte QM1 ağ geçididir.



Şekil 55. Kümenin dışında bir kuyruk yöneticisine yerleştirme

### Bu görev hakkında

İstek ve yanıt iletilerinin yolunu ayarlamak için yönergeleri izleyin

### Yordam

1. Kümeden istek iletilisini gönderin.

Kümenin içindeki QM2kuyruk yöneticisinin QM3adresindeki, kümenin dışındaki Q3 kuyruğuna bir iletiyi nasıl koyduğunu göz önünde bulundurun.

- a) Kümeye Q3 uzak kuyruğu yayınlamak için QM1 üzerinde bir QREMOTE tanımlaması yaratın.

```
DEFINE QREMOTE(Q3) RNAME(Q3) RQMNAME(QM3) CLUSTER(DEMO)
```

Ayrıca, kümenin dışındaki kuyruk yöneticisine bir gönderici kanalı ve bir iletim kuyruğu da vardır. QM3 ' in karşılık gelen bir alıcı kanalı var. Kanallar Şekil 55 sayfa 371 içinde gösterilmez.

QM2 üzerindeki bir uygulama, hedef kuyruğu ve yanıtların gönderileceği kuyruğu belirten bir MQPUT çağrısı yayınlar. Hedef kuyruk Q3 ve yanıt kuyruğu Q2.

İleti, kuyruk adını QM3adresindeki Q3 olarak çözmek için uzak kuyruk tanımlamasını kullanan QM1adresine gönderilir.

## 2. Yanıt iletisini kümenin dışındaki kuyruk yöneticisinden alın.

Kümenin dışındaki bir kuyruk yöneticisinin, ileti gönderdiği kümedeki her kuyruk yöneticisi için bir kuyruk yöneticisi diğer adı olmalıdır. Kuyruk yöneticisi diğer adı, ağ geçidi kuyruk yöneticisine iletim kuyruğunun adını da belirtmelidir. Bu örnekte QM3 , QM2için bir kuyruk yöneticisi diğer adı tanımlaması gerektirir:

a) QM3 üzerinde QM2 kuyruk yöneticisi diğer adı yaratılması

```
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) XMITQ(QM1)
```

QM3 , QM1 ve QM1 için bir gönderici kanalı ve iletim kuyruğuna da karşılık gelen bir alıcı kanal gerekir.

QM3 üzerindeki **app3** uygulaması, bir MQPUT çağrısı yayınlayarak ve Q2 kuyruk adını ve QM2kuyruk yöneticisi adını belirterek QM2' e yanıt gönderebilir.

## Sonraki adım

Bir kümeden birden çok rota tanımlayabilirsiniz.

### İlgili kavramlar

Kuyruk yöneticisi diğer adları ve kümeleri

Bir kümeye ileti gönderirken ya da gönderirken kuyruk yöneticilerinin adını ve bir kümeye gönderilen iş yükü dengeleme iletilerini gizlemek için kuyruk yöneticisi diğer adlarını kullanın.

Yanıt kuyruğu diğer adları ve kümeleri

Yanıtlama kuyruğu diğer adı tanımlama, yanıt bilgilerine ilişkin diğer adları belirlemek için kullanılır.

Yanıtlanacak kuyruk diğer adı tanımlamaları, dağıtılmış bir kuyruğa alma ortamındaki gibi kümelerle birlikte kullanılabilir.

Kuyruk diğer adları ve kümeleri

Bir küme kuyruğunun adını gizlemek, bir kuyruğu kümelemek, farklı öznitelikleri benimsemek ya da farklı erişim denetimlerini benimsemek için kuyruk diğer adlarını kullanın.

### İlgili görevler

Kümeyle istek/yanıt yapılandırılması

Bir kümenin dışındaki bir kuyruk yöneticisinden bir istek/yanıt iletili yolu yapılandırın. Kümeyle iletişim yolu olarak bir ağ geçidi kuyruk yöneticisini kullanarak kümenin iç ayrıntılarını gizleyin.

Kümenin dışından iş yükü dengelemeyi yapılandırma

Bir kümenin dışındaki bir kuyruk yöneticisinden küme kuyruğunun herhangi bir kopyasına bir iletili yolu yapılandırın. Sonuç, kümenin dışından küme kuyruğunun her bir eşgörünümüne gelen iş yükü dengeleme isteklerinin sağlanmasıdır.

Kümeler arasındaki ileti yollarının yapılandırılması

Bir ağ geçidi kuyruk yöneticisini kullanarak kümeleri birbirine bağlayın. Ağ geçidi kuyruk yöneticisinde küme kuyruğu ya da küme kuyruğu yöneticisi diğer adlarını tanımlayarak kuyrukları ya da kuyruk yöneticilerini tüm kümeler tarafından görülebilir yapın.

### **Kümenin dışından iş yükü dengelemeyi yapılandırma**

Bir kümenin dışındaki bir kuyruk yöneticisinden küme kuyruğunun herhangi bir kopyasına bir iletili yolu yapılandırın. Sonuç, kümenin dışından küme kuyruğunun her bir eşgörünümüne gelen iş yükü dengeleme isteklerinin sağlanmasıdır.



## Başlamadan önce

Örneği, “Kümeyle istek/yanıt yapılandırılması” sayfa 364’inde Şekil 53 sayfa 365’inde gösterildiği gibi yapılandırın.

## Bu görev hakkında

Bu senaryoda, kümenin dışındaki QM3 in Şekil 56 sayfa 374’te kuyruk yöneticisi, istekleri Q2 kuyruğuna gönderir. Q2, QM2 ve QM4 küme içinde DEMO olmak üzere iki kuyruk yöneticisinde barındırılır. Her iki kuyruk yöneticisi de iş yükü dengelemeyi kullanmak için NOTFIXED varsayılan bağ tanımlama seçeneğiyle yapılandırılır. Kümenin dışındaki kuyruk yöneticisi olan QM3’den gelen istekler, Q2’un herhangi bir eşgörünümüne QM1 aracılığıyla gönderilir.

QM3 bir kümenin parçası değildir ve dağıtılmış kuyruğa alma tekniklerini kullanarak iletişim kurar. QM1’e bir gönderen kanalı ve bir iletim kuyruğu olmalıdır. QM1 için karşılık gelen bir alıcı kanal gerekir. Kanallar ve iletim kuyrukları Şekil 56 sayfa 374’te açık bir şekilde gösterilmez.

Yordam, “Kümeyle istek/yanıt yapılandırılması” sayfa 364’indeki Şekil 53 sayfa 365’inde örneği genişletir.

## Yordam

1. QM3’ünde Q2 için bir QREMOTE tanımlaması oluşturun.

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(Q3) XMITQ(QM1)
```

QM3’in iletileri koyduğu kümedeki her kuyruk için bir QREMOTE tanımlaması yaratın.

2. QM1’ünde bir kuyruk yöneticisi diğer adı Q3 yaratın.

```
DEFINE QREMOTE(Q3) RNAME(' ') RQMNAME(' ')
```

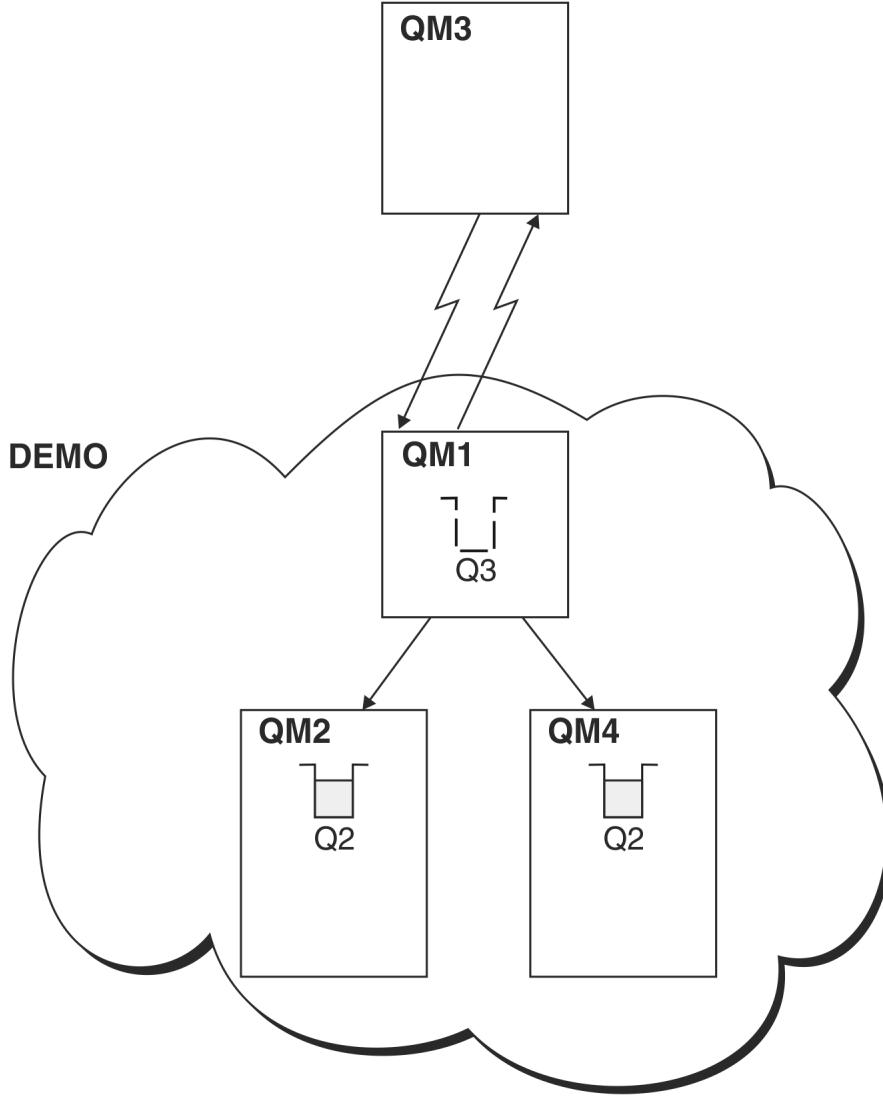
Q3 gerçek bir kuyruk yöneticisi adı değil. Kümedeki bir kuyruk yöneticisi diğer adı tanımlamasının adı. Bu ad, kuyruk yöneticisi diğer adı Q3 ile boşluk, ' '

3. QM2 ve QM4’un her birinde Q2 adlı bir yerel kuyruk tanımlayın.

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO) DEFBIND(NOTFIXED)
```

4. QM1, ağ geçidi kuyruk yöneticisinde özel tanımlama yok.

## Sonuçlar



Şekil 56. Kümenin dışına bir kuyruk yöneticisinden yerleştirme

QM3 adresindeki bir uygulama, Q2 adresine ileti göndermek için bir MQPUT çağrısı yayınladığında, QM3 ile ilgili QREMOTE tanımlaması iletilinin ağ geçidi kuyruk yöneticisi aracılığıyla yönlendirilmesine neden olur QM1. QM1 iletiyi aldığı anda, iletilinin Q2 adlı bir kuyruk için kullanılmaya devam ettiğinin farkında olur ve ad çözümlemesi gerçekleştirir. QM1, yerel tanımlarını denetler ve Q2 için herhangi bir tanım bulamaz. QM1 daha sonra küme yapılandırmasını denetler ve kümede Q2 'un iki eşgörünümünün farkında olduğunu bulur DEMO. QM1 artık, iletileri QM2 ve QM4 üzerinde bulunan Q2 eşgörünümleri arasında dağıtmak için iş yükü dengeleme özelliğini kullanabilir.

### İlgili kavramlar

#### Kuyruk yöneticisi diğer adları ve kümeleri

Bir kümeye ileti gönderirken ya da gönderirken kuyruk yöneticilerinin adını ve bir kümeye gönderilen iş yükü dengeleme iletilerini gizlemek için kuyruk yöneticisi diğer adlarını kullanın.

#### Yanıt kuyruğu diğer adları ve kümeleri

Yanıtlama kuyruğu diğer adı tanımları, yanıt bilgilerine ilişkin diğer adları belirlemek için kullanılır.

Yanıtlanacak kuyruk diğer adı tanımlamaları, dağıtılmış bir kuyruğa alma ortamındaki gibi kümelerle birlikte kullanılabilir.

#### Kuyruk diğer adları ve kümeleri

Bir küme kuyruğunun adını gizlemek, bir kuyruğu kümelemek, farklı öznitelikleri benimsemek ya da farklı erişim denetimlerini benimsemek için kuyruk diğer adlarını kullanın.

#### Ad çözümlemesi

##### **İlgili görevler**

###### Kümeyle istek/yanıt yapılandırılması

Bir kümenin dışındaki bir kuyruk yöneticisinden bir istek/yanıt iletisi yolu yapılandırın. Kümeyle iletişim yolu olarak bir ağ geçidi kuyruk yöneticisini kullanarak kümenin iç ayrıntılarını gizleyin.

###### Kümeden istek/yanıt yapılandırılması

Kümeden kümenin dışındaki bir kuyruk yöneticisine bir istek/yanıt iletisi yolu yapılandırın. Küme içindeki bir kuyruk yöneticisinin bir ağ geçidi kuyruk yöneticisini kullanarak küme dışında nasıl iletişim kuracağına ilişkin ayrıntıları gizleyin.

###### Kümeler arasındaki ileti yollarının yapılandırılması

Bir ağ geçidi kuyruk yöneticisini kullanarak kümeleri birbirine bağlayın. Ağ geçidi kuyruk yöneticisinde küme kuyruğu ya da küme kuyruğu yöneticisi diğer adlarını tanımlayarak kuyrukları ya da kuyruk yöneticilerini tüm kümeler tarafından görülebilir yapın.

##### **İlgili başvurular**

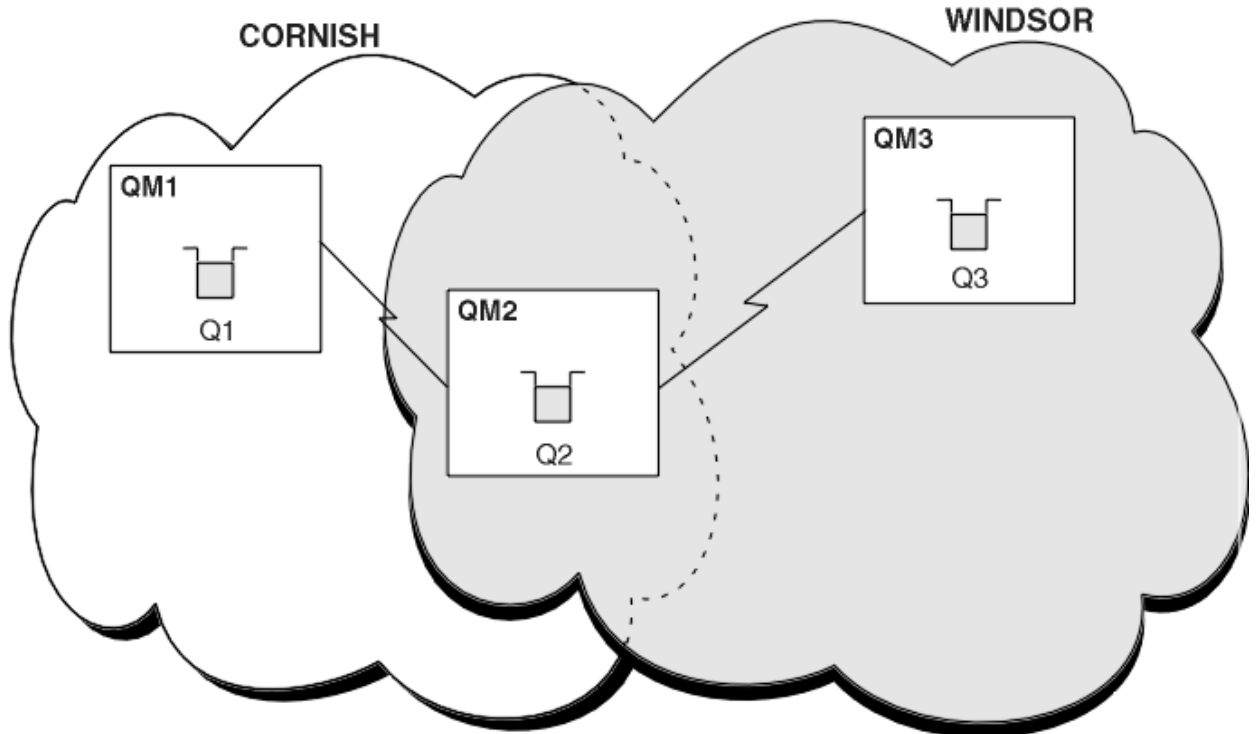
###### Kuyruk adı çözümlemesi

##### ***Kümeler arasındaki ileti yollarının yapılandırılması***

Bir ağ geçidi kuyruk yöneticisini kullanarak kümeleri birbirine bağlayın. Ağ geçidi kuyruk yöneticisinde küme kuyruğu ya da küme kuyruğu yöneticisi diğer adlarını tanımlayarak kuyrukları ya da kuyruk yöneticilerini tüm kümeler tarafından görülebilir yapın.

#### **Bu görev hakkında**

Tüm kuyruk yöneticilerinizi tek bir büyük kümede gruplamak yerine, daha küçük kümeleriniz olabilir. Her kümede, köprü görevi gören bir ya da daha çok kuyruk yöneticisi vardır. Bunun avantajı, kümeler genelinde kuyruk ve kuyruk yöneticisi adlarının görünürlüğünü kısıtlayabilmendir. Bkz. Üst üste binen kümeler. Ad çakışmalarını önlemek ya da yerel adlandırma kurallarına uymak üzere kuyruk ve kuyruk yöneticilerinin adlarını değiştirmek için diğer adları kullanın.



Şekil 57. Kümeler arasında köprü

Şekil 57 sayfa 375 içinde aralarında bir köprü bulunan iki küme gösterilmektedir. Birden fazla köprü olabilir.

Aşağıdaki yordamı kullanarak kümeleri yapılandırın:

## Yordam

1. Define a cluster queue, Q1 on QM1.

```
DEFINE QLOCAL(Q1) CLUSTER(CORNISH)
```

2. Define a cluster queue, Q3 on QM3.

```
DEFINE QLOCAL(Q3) CLUSTER(WINDSOR)
```

3. Her iki kümenin adlarını içeren CORNISHWINDSOR on QM2adlı bir ad listesi oluşturun.

```
DEFINE NAMELIST(CORNISHWINDSOR) DESCR('CornishWindsor namelist')  
NAMES(CORNISH, WINDSOR)
```

4. QM2 üzerinde Q2 küme kuyruğunun tanımlanması

```
DEFINE QLOCAL(Q2) CLUSNL(CORNISHWINDSOR)
```

## Sonraki adım

QM2 , her iki kümenin bir üyesidir ve aralarındaki köprüdür. Köprüde görünür kılmak istediğiniz her kuyruk için köprüde bir QALIAS tanımlaması gerekir. Örneğin, Şekil 57 sayfa 375 içinde, QM2 üzerinde aşağıdakilere gereksinim duyarsınız:

```
DEFINE QALIAS(MYQ3) TARGET(Q3) CLUSTER(CORNISH) DEFBIND(NOTFIXED)
```

Kuyruk diğer adını kullanarak, CORNISH içindeki bir kuyruk yöneticisine bağlı bir uygulama (örneğin, QM1), Q3' e bir ileti gönderebilir. Q3 ögesine MYQ3 olarak başvuruda bulunur. İleti, QM3 adresindeki Q3 adresine yönlendirilir.

Bir kuyruğu açtığınızda, DEFBIND değerini NOTFIXED ya da QDEF olarak ayarlamanız gerekir. DEFBIND varsayılan olarak bırakılırsa OPEN, kuyruk yöneticisi diğer ad tanımlamasını onu barındıran köprü kuyruk yöneticisine çözer. Köprü mesajı iletmiyor.

Görünür kılmak istediğiniz her kuyruk yöneticisi için bir kuyruk yöneticisi diğer adı tanımlaması gerekir. Örneğin, QM2 üzerinde aşağıdakilere gereksinim duyarsınız:

```
DEFINE QREMOTE(QM1) RNAME(' ') RQMNAME(QM1) CLUSTER(WINDSOR)
```

WINDSOR içindeki herhangi bir kuyruk yöneticisine bağlı bir uygulama (örneğin, QM3), MQOPEN çağrısında QM1 ögesini belirttik olarak adlandırarak QM1 üzerindeki herhangi bir kuyruğa bir ileti yerleştirebilir.

## İlgili kavramlar

### Kuyruk yöneticisi diğer adları ve kümeleri

Bir kümeye ileti gönderirken ya da gönderirken kuyruk yöneticilerinin adını ve bir kümeye gönderilen iş yükü dengeleme iletilerini gizlemek için kuyruk yöneticisi diğer adlarını kullanın.

### Yanıt kuyruğu diğer adları ve kümeleri

Yanıtlama kuyruğu diğer adı tanımı, yanıt bilgilerine ilişkin diğer adları belirlemek için kullanılır. Yanıtlanacak kuyruk diğer adı tanımlamaları, dağıtılmış bir kuyruğa alma ortamındaki gibi kümelerle birlikte kullanılabilir.

### Kuyruk diğer adları ve kümeleri

Bir küme kuyruğunun adını gizlemek, bir kuyruğu kümelemek, farklı öznitelikleri benimsemek ya da farklı erişim denetimlerini benimsemek için kuyruk diğer adlarını kullanın.

### **İlgili görevler**

#### **Kümeyle istek/yanıt yapılandırılması**

Bir kümenin dışındaki bir kuyruk yöneticisinden bir istek/yanıt iletisi yolu yapılandırın. Kümeyle iletişim yolu olarak bir ağ geçidi kuyruk yöneticisini kullanarak kümenin iç ayrıntılarını gizleyin.

#### **Kümeden istek/yanıt yapılandırılması**

Kümeden kümenin dışındaki bir kuyruk yöneticisine bir istek/yanıt iletisi yolu yapılandırın. Küme içindeki bir kuyruk yöneticisinin bir ağ geçidi kuyruk yöneticisini kullanarak küme dışında nasıl iletişim kuracağına ilişkin ayrıntıları gizleyin.

#### **Kümenin dışından iş yükü dengelemeyi yapılandırma**

Bir kümenin dışındaki bir kuyruk yöneticisinden küme kuyruğunun herhangi bir kopyasına bir ileti yolu yapılandırın. Sonuç, kümenin dışından küme kuyruğunun her bir eşgörünümüne gelen iş yükü dengeleme isteklerinin sağlanmasıdır.

### ***Kuyruk yöneticisi diğer adları ve kümeleri***

Bir kümeyle ileti gönderirken ya da gönderirken kuyruk yöneticilerinin adını ve bir kümeyle gönderilen iş yükü dengeleme iletilerini gizlemek için kuyruk yöneticisi diğer adlarını kullanın.

Boş RNAME ile uzak kuyruk tanımlaması kullanılarak yaratılan kuyruk yöneticisi diğer adlarının beş kullanımı vardır:

#### **İleti gönderirken kuyruk yöneticisi adını yeniden eşleme**

Bir MQOPEN çağrısında belirtilen kuyruk yöneticisi adını başka bir kuyruk yöneticisine yeniden eşlemek için kuyruk yöneticisi diğer adı kullanılabilir. Bir küme kuyruğu yöneticisi olabilir. Örneğin, bir kuyruk yöneticisinin kuyruk yöneticisi diğer adı tanımlaması olabilir:

```
DEFINE QREMOTE(YORK) RNAME(' ') RQMNAME(CLUSQM)
```

YORK, CLUSQMadlı kuyruk yöneticisi için diğer ad olarak kullanılabilir. Kuyruk yöneticisinde bu tanımlamayı yapan bir uygulama YORKkuyruk yöneticisine bir ileti gönderdiğinde, yerel kuyruk yöneticisi adı CLUSQMolarak çözer. Yerel kuyruk yöneticisi CLUSQMolarak adlandırılmazsa, iletiyi CLUSQM' e taşıyacak küme iletim kuyruğuna yerleştirir. Ayrıca, iletim üstbilgisini YORKyerine CLUSQM olarak değiştirir.

**Not:** Tanımlama yalnızca, tanımlamayı yapan kuyruk yöneticisine uygulanır. Diğer adı kümenin tamamına yayınlamak için uzak kuyruk tanımlamasına CLUSTER özniteliğini eklemeniz gerekir. Daha sonra, YORK için yönlendirilen diğer kuyruk yöneticilerinden gelen iletiler CLUSQM' e gönderilir.

#### **İleti gönderirken iletim kuyruğunun değiştirilmesi ya da belirlenmesi**

Diğer ad, bir kümeyle küme olmayan bir sistemle birleştirmek için kullanılabilir. Örneğin, ITALY kümesindeki kuyruk yöneticileri, kümenin dışındaki PALERMOadlı kuyruk yöneticisiyle iletişim kurabilir. İletişim kurmak için, kümedeki kuyruk yöneticilerinden birinin ağ geçidi işlevi görmesi gerekir. Ağ geçidi kuyruk yöneticisinden şu komutu verin:

```
DEFINE QREMOTE(ROME) RNAME(' ') RQMNAME(PALERMO) XMITQ(X) CLUSTER(ITALY)
```

Komut bir kuyruk yöneticisi diğer adı tanımlamasıdır. It defines and advertises ROME as a queue manager over which messages from any queue manager in the cluster ITALY can multi-hop to reach their destination at PALERMO. Kuyruk yöneticisi adı ROME olarak ayarlanmış olarak açılan bir kuyruğa konan iletiler, kuyruk yöneticisi diğer adı tanımlamasıyla birlikte ağ geçidi kuyruk yöneticisine gönderilir. Bu ileti ulaştıktan sonra, iletiler X iletim kuyruğuna yerleştirilir ve küme olmayan kanallar tarafından kuyruk yöneticisine taşınır PALERMO.

Bu örnekte ROME adının seçimi önemli değildir. QREMOTE ve RQMNAME değerleri aynı olabilir.

## İleti alırken hedefin belirlenmesi

Bir kuyruk yöneticisi bir ileti aldığında, iletim üstbilgisinden hedef kuyruk ve kuyruk yöneticisinin adını alır. İletim üstbilgisindeki kuyruk yöneticisiyle aynı adı taşıyan bir kuyruk yöneticisi diğer adı tanımlamasını arar. Bir tane bulursa, iletim üstbilgisindeki kuyruk yöneticisi adı için kuyruk yöneticisi diğer adı tanımındaki RQMNAME yerine kullanılır.

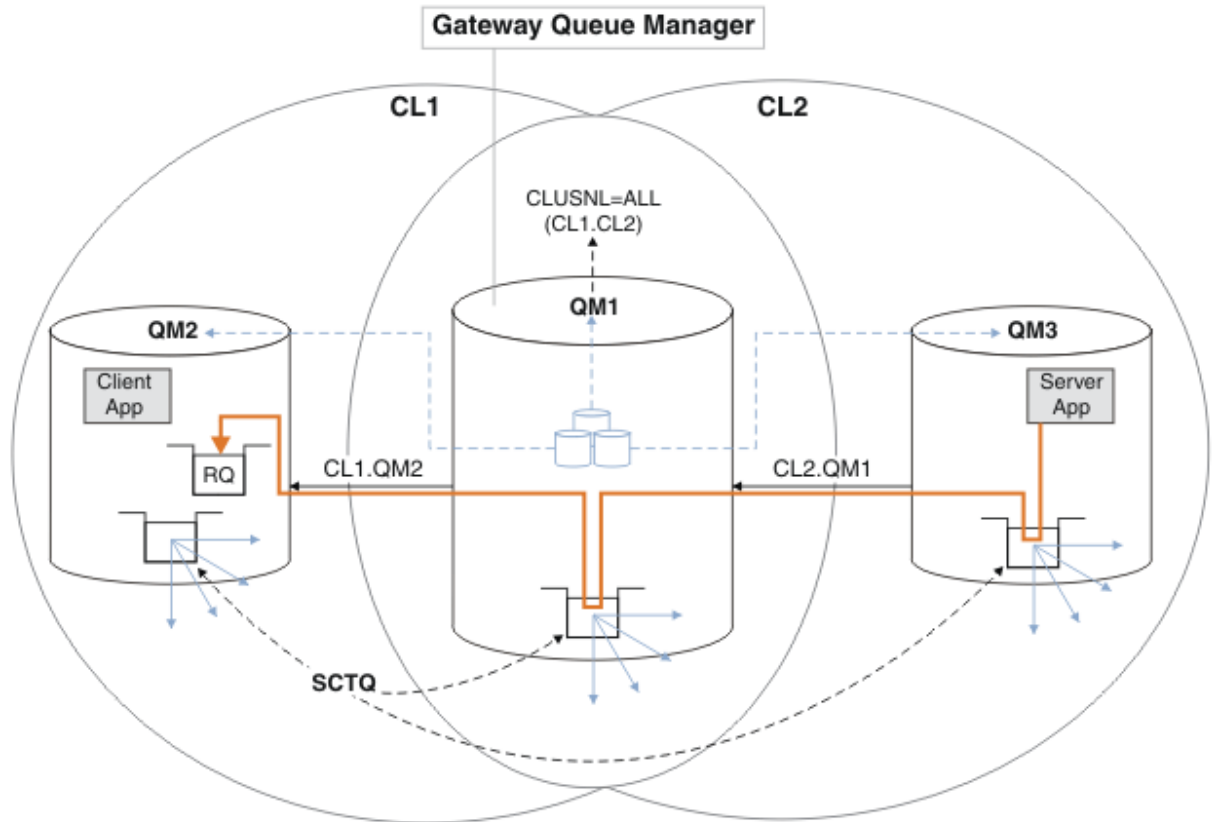
Bir kuyruk yöneticisi diğer adını bu şekilde kullanmanın iki nedeni vardır:

- İletileri başka bir kuyruk yöneticisine yönlendirmek için
- Kuyruk yöneticisi adını, yerel kuyruk yöneticisiyle aynı olacak şekilde değiştirmek için

## İletileri farklı kümelerdeki kuyruk yöneticileri arasında yönlendirmek için bir ağ geçidi kuyruk yöneticisinde kuyruk yöneticisi diğer adlarını kullanma.

Bir uygulama, kuyruk yöneticisi diğer adını kullanarak farklı bir kümedeki bir kuyruğa ileti gönderebilir. Kuyruğun bir küme kuyruğu olması gerekmez. Kuyruk tek bir kümede tanımlandı. Uygulama, farklı bir kümedeki bir kuyruk yöneticisine bağlı. Bir ağ geçidi kuyruk yöneticisi iki kümeyi bağlar. Kuyruk kümelenmiş olarak tanımlanmazsa, doğru yöneltmenin gerçekleşmesi için uygulamanın kuyruk adını ve bir kümelenmiş kuyruk yöneticisi diğer adını kullanarak kuyruğu açması gerekir. Bir yapılandırma örneği için, şekil 1 içinde gösterilen yanıt ileti akışının alındığı "Ağ geçidi kuyruk yöneticisiyle çakışan iki küme oluşturma" sayfa 330 başlıklı konuya bakın.

Çizge, yanıt iletisi tarafından alınan yolu, RQ olarak adlandırılan geçici bir dinamik kuyruğa geri gösterir. QM3' e bağlı sunucu uygulaması, QM2 kuyruk yöneticisi adını kullanarak yanıt kuyruğunu açar. Kuyruk yöneticisi adı QM2 , QM1 üzerinde kümelenmiş kuyruk yöneticisi diğer adı olarak tanımlanır. QM3 , yanıt iletisini QM1' e yönlendirir. QM1 , iletiyi QM2' e yönlendirir.



Şekil 58. Yanıt iletisini farklı bir kümeye döndürmek için kuyruk yöneticisi diğer adının kullanılması

Yönlendirmenin çalışma şekli şu şekildedir. Her kümedeki her kuyruk yöneticisinin QM1 üzerinde bir kuyruk yöneticisi diğer adı tanımlaması vardır. Diğer adlar tüm kümelerde kümelenir. Her bir diğer addan bir kuyruk yöneticisine gelen gri çizgili oklar, her kuyruk yöneticisi diğer adının kümelerin en az birinde gerçek bir kuyruk yöneticisine çözümlendiğini gösterir. Bu durumda, QM2

diğer adı hem kümede CL1 hem de CL2kümelenir ve CL1içindeki QM2 gerçek kuyruk yöneticisine çözümlenir. Sunucu uygulaması, yanıt iletisini RQkuyruk adını kullanarak ve QM2kuyruk yöneticisine yanıt vererek yaratır. QM2 kuyruk yöneticisi diğer ad tanımlaması QM1 kümesinde CL2 ve QM2 kuyruk yöneticisinde CL2tanımlı olmadığı için ileti QM1 olarak yöneltilir. İleti hedef kuyruk yöneticisine gönderilemediğinden, diğer ad tanımlamasına sahip kuyruk yöneticisine gönderilir.

QM1 places the message on the cluster transmission queue on QM1 for transferal to QM2. QM1 , QM2 için QM1 üzerindeki kuyruk yöneticisi diğer adı tanımlaması QM2 'i gerçek hedef kuyruk yöneticisi olarak tanımladığından, iletiyi QM2 ' e yöneltir. Diğer ad tanımlamaları yalnızca gerçek tanımlamalara gönderme yapabildiği için tanımlama çevrimsel değildir; diğer ad kendisini gösteremez. Hem QM1 hem de QM2 aynı kümede ( CL1) olduğundan, gerçek tanımlama QM1tarafından çözümlenir. QM1 , CL1havuzundan QM2 ile ilgili bağlantı bilgilerini bulur ve iletiyi QM2' e yönlendirir. İletinin QM1tarafından yeniden yönlendirilmesi için, sunucu uygulaması yanıt kuyruğunu DEFBIND seçeneği MQBND\_BIND\_NOT\_FIXEDolarak ayarlanmış şekilde açmış olmalıdır. Sunucu uygulaması yanıt kuyruğunu MQBND\_BIND\_ON\_OPENseçeneğiyle açtıysa, ileti yeniden yönlendirilmez ve gitmeyen iletiler kuyruğuna gönderilir.

### **Kümenin dışından gelen iş yükü dengeleme iletileri için kümeye ağ geçidi olarak bir kuyruk yöneticisinin kullanılması.**

Kümedeki birden çok kuyruk yöneticisinde EDINBURGH adlı bir kuyruk tanımlarsınız. Kümeleme mekanizmasının, kümenin dışından o kuyruğa gelen iletilere ilişkin iş yükünü dengelemesi için kullanılır.

Kümenin dışından gelen bir kuyruk yöneticisi, kümedeki bir kuyruk yöneticisine bir iletim kuyruğu ve gönderen kanalı gerektirir. Bu kuyruğa ağ geçidi kuyruk yöneticisi adı verilir. Varsayılan iş yükü dengeleme mekanizmasından yararlanmak için aşağıdaki kurallardan biri geçerli olmalıdır:

- Ağ geçidi kuyruk yöneticisi, EDINBURGH kuyruğunun bir eşgörünümünü içermemelidir.
- Ağ geçidi kuyruk yöneticisi, ALTER QMGRüzerinde CLWLUSEQ (ANY) belirtiyor.

Bir kümenin dışından iş yükü dengeleme örneği için bkz. [“Kümenin dışından iş yükü dengelemeyi yapılandırma” sayfa 372](#)

### **İlgili kavramlar**

Yanıt kuyruğu diğer adları ve kümeleri

Yanıtlama kuyruğu diğer adı tanımlaması, yanıt bilgilerine ilişkin diğer adları belirlemek için kullanılır.

Yanıtlanacak kuyruk diğer adı tanımlamaları, dağıtılmış bir kuyruğa alma ortamındaki gibi kümelerle birlikte kullanılabilir.

Kuyruk diğer adları ve kümeleri

Bir küme kuyruğunun adını gizlemek, bir kuyruğu kümelemek, farklı öznitelikleri benimsemek ya da farklı erişim denetimlerini benimsemek için kuyruk diğer adlarını kullanın.

### **İlgili görevler**

Kümeyle istek/yanıt yapılandırılması

Bir kümenin dışındaki bir kuyruk yöneticisinden bir istek/yanıt iletisi yolu yapılandırın. Kümeyle iletişim yolu olarak bir ağ geçidi kuyruk yöneticisini kullanarak kümenin iç ayrıntılarını gizleyin.

Kümeden istek/yanıt yapılandırılması

Kümeden kümenin dışındaki bir kuyruk yöneticisine bir istek/yanıt iletisi yolu yapılandırın. Küme içindeki bir kuyruk yöneticisinin bir ağ geçidi kuyruk yöneticisini kullanarak küme dışında nasıl iletişim kuracağına ilişkin ayrıntıları gizleyin.

Kümenin dışından iş yükü dengelemeyi yapılandırma

Bir kümenin dışındaki bir kuyruk yöneticisinden küme kuyruğunun herhangi bir kopyasına bir ileti yolu yapılandırın. Sonuç, kümenin dışından küme kuyruğunun her bir eşgörünümüne gelen iş yükü dengeleme isteklerinin sağlanmasıdır.

Kümeler arasındaki ileti yollarının yapılandırılması

Bir ağ geçidi kuyruk yöneticisini kullanarak kümeleri birbirine bağlayın. Ağ geçidi kuyruk yöneticisinde küme kuyruğu ya da küme kuyruğu yöneticisi diğer adlarını tanımlayarak kuyrukları ya da kuyruk yöneticilerini tüm kümeler tarafından görülebilir yapın.

## Yanıt kuyruğu diğer adları ve kümeleri

Yanıtlama kuyruğu diğer adı tanımlama, yanıt bilgilerine ilişkin diğer adları belirlemek için kullanılır. Yanıtlanacak kuyruk diğer adı tanımlamaları, dağıtılmış bir kuyruğa alma ortamındaki gibi kümelerle birlikte kullanılabilir.

Örneğin:

- VENICE kuyruk yöneticisindeki bir uygulama, MQPUT çağrısıyla PISA kuyruk yöneticisine bir ileti gönderir. Uygulama, ileti tanımlayıcısında aşağıdaki yanıt kuyruğu bilgilerini sağlar:

```
ReplyToQ='QUEUE'  
ReplyToQMGr=''
```

- QUEUE adresine gönderilen yanıtların PISAadresindeki OTHERQ adresinden alınabilmesi için, VENICE üzerinde yanıtlanacak kuyruk diğer adı olarak kullanılan bir uzak kuyruk tanımlaması yaratın. Diğer ad yalnızca, yaratıldığı sistemde geçerlidir.

```
DEFINE QREMOTE(Queue) RNAME(OTHERQ) RQMNAME(PISA)
```

RQMNAME ve QREMOTE , RQMNAME bir küme kuyruğu yöneticisi olsa da aynı adları belirtebilir.

### İlgili kavramlar

Kuyruk yöneticisi diğer adları ve kümeleri

Bir kümeye ileti gönderirken ya da gönderirken kuyruk yöneticilerinin adını ve bir kümeye gönderilen iş yükü dengeleme iletilerini gizlemek için kuyruk yöneticisi diğer adlarını kullanın.

Kuyruk diğer adları ve kümeleri

Bir küme kuyruğunun adını gizlemek, bir kuyruğu kümelemek, farklı öznitelikleri benimsemek ya da farklı erişim denetimlerini benimsemek için kuyruk diğer adlarını kullanın.

### İlgili görevler

Kümeyle istek/yanıt yapılandırılması

Bir kümenin dışındaki bir kuyruk yöneticisinden bir istek/yanıt iletisi yolu yapılandırın. Kümeyle iletişim yolu olarak bir ağ geçidi kuyruk yöneticisini kullanarak kümenin iç ayrıntılarını gizleyin.

Kümeden istek/yanıt yapılandırılması

Kümeden kümenin dışındaki bir kuyruk yöneticisine bir istek/yanıt iletisi yolu yapılandırın. Küme içindeki bir kuyruk yöneticisinin bir ağ geçidi kuyruk yöneticisini kullanarak küme dışında nasıl iletişim kuracağına ilişkin ayrıntıları gizleyin.

Kümenin dışından iş yükü dengelemeyi yapılandırma

Bir kümenin dışındaki bir kuyruk yöneticisinden küme kuyruğunun herhangi bir kopyasına bir ileti yolu yapılandırın. Sonuç, kümenin dışından küme kuyruğunun her bir eşgörünümüne gelen iş yükü dengeleme isteklerinin sağlanmasıdır.

Kümeler arasındaki ileti yollarının yapılandırılması

Bir ağ geçidi kuyruk yöneticisini kullanarak kümeleri birbirine bağlayın. Ağ geçidi kuyruk yöneticisinde küme kuyruğu ya da küme kuyruğu yöneticisi diğer adlarını tanımlayarak kuyrukları ya da kuyruk yöneticilerini tüm kümeler tarafından görülebilir yapın.

## Kuyruk diğer adları ve kümeleri

Bir küme kuyruğunun adını gizlemek, bir kuyruğu kümelemek, farklı öznitelikleri benimsemek ya da farklı erişim denetimlerini benimsemek için kuyruk diğer adlarını kullanın.

QALIAS tanımlaması, kuyruğun tanınacağı bir diğer ad yaratmak için kullanılır. Çeşitli nedenlerden ötürü bir diğer ad yaratabilirsiniz:

- Farklı bir kuyruk kullanmaya başlamak istiyorsunuz, ancak uygulamalarınızı değiştirmek istemiyorsunuz.
- Uygulamaların, iletileri koydukları kuyruğun gerçek adını bilmesini istemezsiniz.
- Kuyruğun tanımlandığı yerden farklı bir adlandırma kuralınız olabilir.
- Uygulamalarınızın kuyruğa gerçek adıyla değil, yalnızca diğer adıyla erişme yetkisi olabilir.



DEFINE QALIAS komutunu kullanarak bir kuyruk yöneticisinde QALIAS tanımlaması yaratın. Örneğin, şu komutu çalıştırın:

```
DEFINE QALIAS(PUBLIC) TARGET(LOCAL) CLUSTER(C)
```

Komut, Ckümesindeki kuyruk yöneticilerine PUBLIC adlı bir kuyruk tanıtır. PUBLIC , LOCALadlı kuyruğa çözülen bir diğer addır. PUBLIC ' e gönderilen iletiler, LOCALadlı kuyruğa yöneltilir.

Bir kuyruk adını küme kuyruğuna çözümlenmek için kuyruk diğer adı tanımlamasını da kullanabilirsiniz. Örneğin, şu komutu çalıştırın:

```
DEFINE QALIAS(PRIVATE) TARGET(PUBLIC)
```

Bu komut, bir kuyruk yöneticisinin kümenin başka bir yerinde PUBLICadıyla yayınlanan bir kuyruğa erişmek için PRIVATE adını kullanmasını sağlar. Bu tanımlama CLUSTER özneliğini içermediğinden, yalnızca onu yapan kuyruk yöneticisi için geçerlidir.

### İlgili kavramlar

Kuyruk yöneticisi diğer adları ve kümeleri

Bir kümeye ileti gönderirken ya da gönderirken kuyruk yöneticilerinin adını ve bir kümeye gönderilen iş yükü dengeleme iletilerini gizlemek için kuyruk yöneticisi diğer adlarını kullanın.

Yanıt kuyruğu diğer adları ve kümeleri

Yanıtlama kuyruğu diğer adı tanımlama, yanıt bilgilerine ilişkin diğer adları belirlemek için kullanılır.

Yanıtlanacak kuyruk diğer adı tanımlamaları, dağıtılmış bir kuyruğa alma ortamındaki gibi kümelerle birlikte kullanılabilir.

### İlgili görevler

Kümeyle istek/yanıt yapılandırılması

Bir kümenin dışındaki bir kuyruk yöneticisinden bir istek/yanıt iletileri yolu yapılandırın. Kümeyle iletişim yolu olarak bir ağ geçidi kuyruk yöneticisini kullanarak kümenin iç ayrıntılarını gizleyin.

Kümeden istek/yanıt yapılandırılması

Kümeden kümenin dışındaki bir kuyruk yöneticisine bir istek/yanıt iletileri yolu yapılandırın. Küme içindeki bir kuyruk yöneticisinin bir ağ geçidi kuyruk yöneticisini kullanarak küme dışında nasıl iletişim kuracağına ilişkin ayrıntıları gizleyin.

Kümenin dışından iş yükü dengelemeyi yapılandırma

Bir kümenin dışındaki bir kuyruk yöneticisinden küme kuyruğunun herhangi bir kopyasına bir ileti yolu yapılandırın. Sonuç, kümenin dışından küme kuyruğunun her bir eşgörünümüne gelen iş yükü dengeleme isteklerinin sağlanmasıdır.

Kümeler arasındaki ileti yollarının yapılandırılması

Bir ağ geçidi kuyruk yöneticisini kullanarak kümeleri birbirine bağlayın. Ağ geçidi kuyruk yöneticisinde küme kuyruğu ya da küme kuyruğu yöneticisi diğer adlarını tanımlayarak kuyrukları ya da kuyruk yöneticilerini tüm kümeler tarafından görülebilir yapın.

## İş yükü yönetimi için kümeleri kullanma

Bir kümedeki farklı kuyruk yöneticilerindeki bir kuyruğun birden çok eşgörünümünü tanımlayarak, kuyruğa hizmet işini birden çok sunucuya dağıtabilirsiniz. Hata durumunda iletilerin farklı bir kuyruk yöneticisine yeniden kuyruğa alınmasını önleyebilecek birkaç etmen vardır.

Sistem yönetimini azaltmak için kümeler ayarlamasının yanı sıra, birden çok kuyruk yöneticisinin aynı kuyruğun eşgörünümünü bulundurduğu kümeler de yaratabilirsiniz.

Kümenizi, içindeki kuyruk yöneticileri birbirinin klonları olacak şekilde düzenleyebilirsiniz. Her kuyruk yöneticisi aynı uygulamaları çalıştırabilir ve aynı kuyruklara ilişkin yerel tanımlamaları olabilir.

► **z/OS** Örneğin, z/OS koşut sistem şebekesinde, klonlanan uygulamalar paylaşılan Db2 ya da Virtual Storage Access Method (VSAM) veritabanındaki verilere erişebilir. Bir uygulamanın birkaç eşgörünümüne

sahip olarak iş yükünü kuyruk yöneticileriniz arasında dağıtabilirsiniz. Uygulamanın her eşgörünümü iletileri alır ve diğerlerinden bağımsız olarak çalışır.

Kümeleri bu şekilde kullanmanın avantajları şunlardır:

- Kuyruklarınızın ve uygulamalarınızın kullanılabilirliğini artırır.
- Daha hızlı ileti çıkışı.
- Ağınızdaki iş yükünün daha eşit bir şekilde dağıtılması.

Belirli bir kuyruğun eşgörünümünü barındıran kuyruk yöneticilerinden herhangi biri, söz konusu kuyruk için hedeflenmiş iletileri işleyebilir ve uygulamalar ileti gönderirken kuyruk yöneticisine ad vermezler. Bir küme aynı kuyruğun birden çok eşgörünümünü içeriyorsa, IBM MQ iletiyi yönetmek için bir kuyruk yöneticisi seçer. Kuyruk yöneticisinin ve kuyruğun kullanılabilirliğine ve kuyruk yöneticileriyle, kuyruklarıyla ve kanallarla ilişkili küme iş yüküne özgü özniteliklere dayalı olarak uygun hedefler seçilir. Bkz. [Kümeler içinde iş yükü dengeleme](#).

**z/OS** IBM MQ for z/OS' de, kuyruk paylaşım gruplarındaki kuyruk yöneticileri, küme kuyruklarını paylaşılan kuyruklar olarak barındıracak. Paylaşılan küme kuyrukları, aynı kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından kullanılabilir. Örneğin, [Aynı kuyruğun birden çok eşgörünümüne sahip bir küme](#) içinde ya da her iki kuyruk yöneticisi QM2 ve QM4 bir paylaşılan kuyruk yöneticisi olabilir. Her birinin Q3kuyruğu için bir tanımlaması vardır. QM4 ile aynı kuyruk paylaşım grubundaki kuyruk yöneticilerinden herhangi biri, paylaşılan kuyruğa konan bir iletiyi okuyabilir Q3. Her kuyruk paylaşım grubu, her biri aynı verilere erişimi olan en fazla 32 kuyruk yöneticisi içerebilir. Kuyruk paylaşımı, iletilerinizin verimini önemli ölçüde artırır.

İş yükü yönetimine ilişkin küme yapılandırmalarıyla ilgili daha fazla bilgi için aşağıdaki alt başlıklara bakın:

### **İlgili kavramlar**

[Kümeleme ve dağıtılmış kuyruğa alma karşılaştırması](#)

[Dağıtılmış kuyruğa alma ve kümeler](#)

[Bir kümenin bileşenleri](#)

[Küme kanalları](#)

[MQPUT için bir küme kuyruğu geçersiz kılındıysa ne olur?](#)

[Bir küme gönderen kanalında iş yükü dengeleme kümesi çalışmıyor](#)

[“İletileri kümelere/kümelere yönlendirme” sayfa 363](#)

Kümeleri dış kuyruk yöneticilerine ve diğer kümelere bağlamak için kuyruk diğer adlarını, kuyruk yöneticisi diğer adlarını ve uzak kuyruk tanımlamalarını kullanın.

### **İlgili görevler**

[Küme iş yükü çıkışlarının yazılması ve derlenmesi](#)

[“Kuyruk yöneticisi kümesinin yapılandırılması” sayfa 283](#)

Kümeler, hem ilk yapılandırmayı hem de devam eden yönetimi basitleştiren bir şekilde birbirine bağlanan kuyruk yöneticileri için bir mekanizma sağlar. Küme bileşenleri tanımlayabilir ve kümeleri oluşturabilir ve yönetebilirsiniz.

[“Yeni bir küme ayarlama” sayfa 297](#)

Örnek kümeyi ayarlamak için bu yönergeleri izleyin. Ayrı yönergeler, kümenin TCP/IP, LU 6.2üzerinde ve tek bir iletim kuyruğuyla ya da birden çok iletim kuyruğuyla ayarlanmasını açıklar. Bir kuyruk yöneticisinden diğerine ileti göndererek kümeyi sınavın.

[“Kümenin dışından iş yükü dengelemeyi yapılandırma” sayfa 372](#)

Bir kümenin dışındaki bir kuyruk yöneticisinden küme kuyruğunun herhangi bir kopyasına bir ileti yolu yapılandırın. Sonuç, kümenin dışından küme kuyruğunun her bir eşgörünümüne gelen iş yükü dengeleme isteklerinin sağlanmasıdır.

### **İlgili başvurular**

[Küme Kuyruğu İzleme örnek programı \(AMQSCLM\)](#)

### **Bir kuyruğun birden çok eşgörünümüne sahip bir küme örneği**

Bir kuyruğun birden çok eşgörünümüne sahip bir kümenin bu örneğinde, iletiler kuyruğun farklı eşgörünümüne yöneltilir. Bir iletiyi kuyruğun belirli bir eşgörünümüne zorlayabilir ve kuyruk yöneticilerinden birine ileti dizisi göndermeyi seçebilirsiniz.

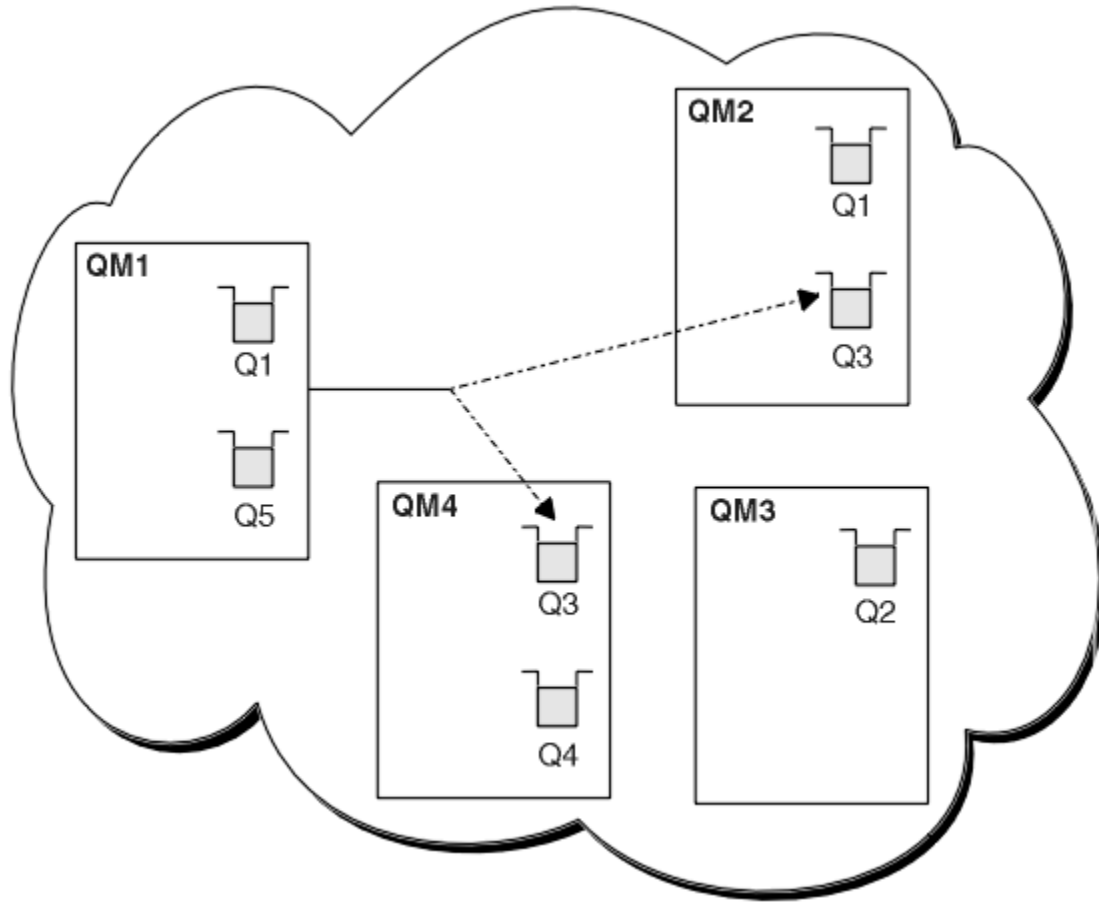
Şekil 59 sayfa 383 , kuyruk için birden çok tanımlama bulunan bir kümeyi gösterir Q3. QM1 adresindeki bir uygulama Q3' a bir ileti gönderirse, hangi Q3 örneğinin iletiyi işleyeceğini mutlaka bilemez. Bir uygulama QM2 ya da QM4 , üzerinde çalışıyorsa ve yerel Q3yönetim ortamları varsa, varsayılan olarak Q3 yerel yönetim ortamı açılır. CLWLUSEQ kuyruk özniteliği ayarlanarak, kuyruğun yerel eşgörünümü, kuyruğun uzak bir örneğiyle aynı işlem görür.

MQOPEN seçeneği DefBind , hedef kuyruk yöneticisinin MQOPEN çağrısı yayınlandığında mı, yoksa iletinin iletim kuyruğundan mı aktarılacağını denetler.

DefBind için MQBND\_BIND\_NOT\_FIXED değerini belirlerseniz, ileti iletilirken kullanılacak bir kuyruk eşgörünümüne gönderilebilir. Bu, aşağıdaki sorunları önler:

- İleti hedef kuyruk yöneticisine ulaştığında hedef kuyruk kullanılamaz.
- Kuyruğun durumu değişti.
- İleti bir küme kuyruğu diğer adı kullanılarak kondu ve küme kuyruğu diğer adının eşgörünümünün tanımlandığı kuyruk yöneticisinde hedef kuyruk kuyruğu eşgörünümü yok.

Bu sorunlar iletim sırasında saptanırsa, hedef kuyruğun başka bir eşgörünümü aranır ve ileti yeniden yönlendirilir. Kuyruğun kullanılabilir bir eşgörünümü yoksa, ileti teslim almamış iletiler kuyruğuna yerleştirilir.



Şekil 59. Aynı kuyruğun birden çok eşgörünümüne sahip bir küme

İletilerin yeniden yönlendirilmesini önleyebilecek bir faktör, iletilerin MQBND\_BIND\_ON\_OPEN ile sabit bir kuyruk yöneticisine ya da kanala atanması olabilir. MQOPEN üzerinde bağ tanımlanan iletiler hiçbir zaman başka bir kanala yeniden yerleştirilmez. Ayrıca, ileti yeniden konumunun yalnızca bir küme kanalı

gerçekten başarısız olduğunda gerçekleştiğini unutmayın. Kanal zaten başarısız olursa yeniden tahsis gerçekleşmez.

Hedef kuyruk yöneticisi hizmet dışı kaldıysa, sistem iletiyi yeniden yönlendirmeyi dener. Bu durumda, iletinin kaybedilme riskini çalıştırarak ya da bir yineleme oluşturarak iletinin bütünlüğünü etkilemez. Bir kuyruk yöneticisi başarısız olursa ve bir iletiyi şüphe içinde bırakırsa, bu ileti yeniden yönlendirilmez.

**z/OS** IBM MQ for z/OS' ta, ileti yeniden yerleştirme işlemi tamamlanincaya kadar kanal tamamen durmaz. Kip FORCE ya da TERMINATE olarak ayarlanmış olarak kanalın durdurulması işlemi kesintiye uğratar; bunu yaparsanız, bazı BIND\_NOT\_FIXED iletileri başka bir kanala önceden yeniden tahsis edilmiş ya da iletiler sıralı olmayabilir.

**Not:** **z/OS**

1. Aynı kuyruğun birden çok eşgörünümüne sahip bir küme oluşturmadan önce, iletilerinizin birbirine bağımlı olmadığından emin olun. Örneğin, belirli bir sırada ya da aynı kuyruk yöneticisi tarafından işlenmesi gerekir.
2. Aynı kuyruğun farklı eşgörünümlerine ilişkin tanımlamaları aynı yapın. Ters durumda, farklı MQINQ çağrılarında farklı sonuçlar elde edin.

### İlgili kavramlar

Uygulama programlama ve kümeleri

Aynı kuyruktaki birden çok eşgörünümünden yararlanmak için programlama değişiklikleri yapmanız gerekmez. Ancak, kuyruğun aynı örneğine bir ileti dizisi gönderilmedikçe, bazı programlar doğru çalışmaz.

### İlgili görevler

Yerel olarak kuyruk barındıran bir kuyruk yöneticisi eklenmesi

Paris ve New York 'ta stok uygulama sistemini çalıştırmak için ek kapasite sağlamak üzere bir INVENTQ eşgörünümü eklemek üzere bu yönergeleri izleyin.

Kümede iki ağ kullanılması

İki farklı ağ olan TOKYO içine yeni bir mağaza eklemek için bu yönergeleri izleyin. Her ikisinin de Tokyo 'daki kuyruk yöneticisiyle iletişim kurmak için kullanılabilir olması gerekir.

Kümede birincil ve ikincil ağ kullanılması

Bir ağı birincil ağ, diğerini yedek ağ yapmak için bu yönergeleri izleyin. Birincil ağla ilgili bir sorun varsa yedek ağı kullanın.

Yedek olarak işlev görececek bir kuyruk ekleme

Şu anda New York 'ta çalışan envanter sistemi için Chicago 'da yedek sağlamak üzere bu yönergeleri izleyin. Chicago sistemi yalnızca New York sistemiyle ilgili bir sorun olduğunda kullanılır.

Kullanılan kanal sayısının kısıtlanması

Çeşitli kuyruk yöneticilerine bir fiyat denetimi uygulaması kurulduğunda her bir sunucunun çalıştırdığı etkin kanal sayısını sınırlamak için bu yönergeleri izleyin.

Bir kuyruğu barındıran daha güçlü bir kuyruk yöneticisi ekleme

Los Angeles 'ta ve New York 'ta envanter sistemini çalıştırarak ek kapasite sağlamak için bu yönergeleri izleyin. Burada Los Angeles, New York 'tan iki kat daha fazla ileti işleyebiliyor.

### ***Yerel olarak kuyruk barındıran bir kuyruk yöneticisi eklenmesi***

Paris ve New York 'ta stok uygulama sistemini çalıştırmak için ek kapasite sağlamak üzere bir INVENTQ eşgörünümü eklemek üzere bu yönergeleri izleyin.

### **Başlamadan önce**

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- INVENTORY kümesi, Kümeye yeni bir kuyruk yöneticisi eklenmesikonusunda açıklandığı gibi ayarlandı. Üç kuyruk yöneticisi içerir; LONDON ve NEWYORK her ikisi de tam havuzu tutar, PARIS kısmi bir havuzu

tutar. Döküm uygulaması, NEWYORK kuyruk yöneticisine bağlı olarak New York 'taki sistemde çalışır. Uygulama, INVENTQ kuyruğundaki iletilerin gelişyle yönlendirilir.

- Paris ve New York 'ta envanter uygulama sistemini çalıştırmak için ek kapasite sağlamak üzere bir INVENTQ eşgörünümü eklemek istiyoruz.

## Bu görev hakkında

Bir kuyruğu yerel olarak barındıran bir kuyruk yöneticisi eklemek için aşağıdaki adımları izleyin.

### Yordam

1. PARIS kuyruk yöneticisini değiştirin.

Paris 'teki uygulamanın Paris 'teki INVENTQ ve New York 'taki uygulamayı kullanması için kuyruk yöneticisine bilgi vermeliyiz. PARIS ' de aşağıdaki komutu verin:

```
ALTER QMGR CLWLUSEQ(ANY)
```

2. İleti yakınlıkları için stok uygulamasını gözden geçirin.

Devam etmeden önce, stok uygulamasının ileti işleme sırasına bağımlı olmadığından emin olun. Daha fazla bilgi için bkz. [İleti yakınlıkları](#).

3. Paris 'teki sisteme döküm uygulamasını kurun.
4. INVENTQküme kuyruğunu tanımlayın.

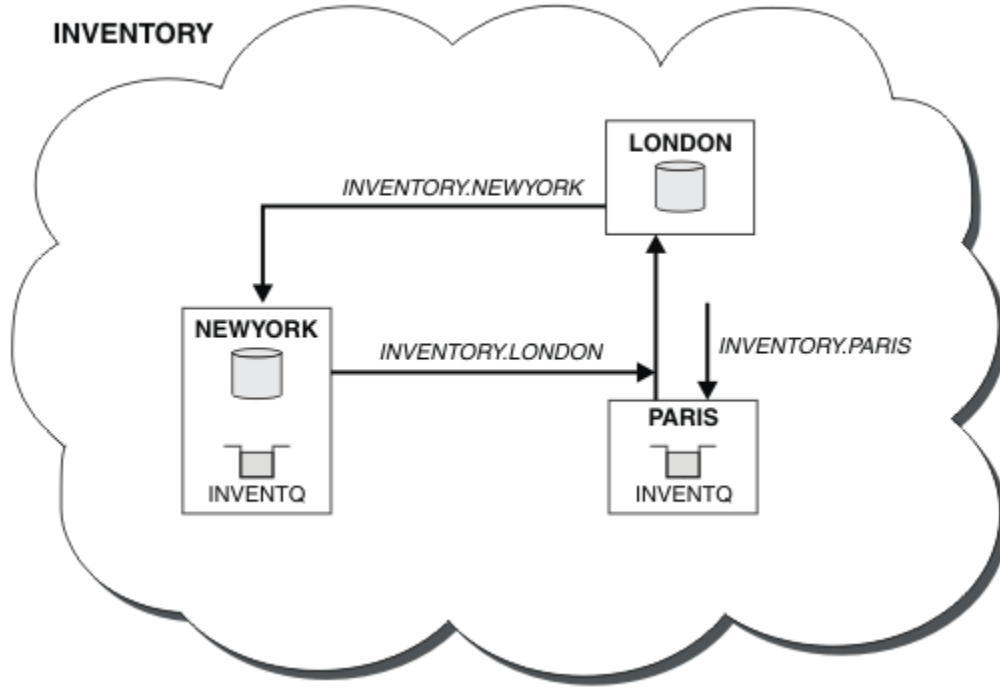
NEWYORK kuyruk yöneticisi tarafından barındırılan INVENTQ kuyruğu da PARIS tarafından barındırılır. PARIS kuyruk yöneticisinde aşağıdaki gibi tanımlayın:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

Artık tüm tanımları tamamladınız, henüz tamamlamadıysanız, kanal başlatıcısını IBM MQ for z/OS üzerinde başlatın. Tüm altyapılarda, PARIS kuyruk yöneticisinde bir dinleyici programı başlatın. Dinleyici, gelen ağ isteklerini dinler ve gerektiğinde küme-alıcı kanalını başlatır.

### Sonuçlar

[Şekil 60 sayfa 386](#) , bu görev tarafından ayarlanan kümeyi gösterir.



Şekil 60. Üç kuyruk yöneticisiyle INVENTORY kümesi

Bu kümede yapılan değişiklik, siz kuyruk yöneticilerini NEWYORK ya da LONDONdeğiştirmeden gerçekleştirildi. Bu kuyruk yöneticilerindeki tüm havuzlar, PARISadresindeki INVENTQ ' e ileti gönderebilmeleri için gereksinim duydukları bilgilerle otomatik olarak güncellenir.

## Sonraki adım

INVENTQ kuyruğu ve döküm uygulaması artık kümedeki iki kuyruk yöneticisinde barındırılır. Bu, kullanılabilirliğini artırır, iletilerin verimini hızlandırır ve iş yükünün iki kuyruk yöneticisi arasında dağıtılmasını sağlar. Herhangi bir kuyruk yöneticisi LONDON, NEWYORK, PARIS tarafından INVENTQ 'e yerleştirilen iletiler, iş yükünün dengelenmesi için dönüşümlü olarak PARIS ya da NEWYORK' e yönlendirilir.

## İlgili kavramlar

Bir kuyruğun birden çok eşgörünümüne sahip bir küme örneği

Bir kuyruğun birden çok eşgörünümüne sahip bir kümenin bu örneğinde, iletiler kuyruğun farklı eşgörünümlerine yönlendirilir. Bir iletiyi kuyruğun belirli bir eşgörünümüne zorlayabilir ve kuyruk yöneticilerinden birine ileti dizisi göndermeyi seçebilirsiniz.

Uygulama programlama ve kümeleri

Aynı kuyruktaki birden çok eşgörünümünden yararlanmak için programlama değişiklikleri yapmanız gerekmez. Ancak, kuyruğun aynı örneğine bir ileti dizisi gönderilmedikçe, bazı programlar doğru çalışmaz.

## İlgili görevler

Kümede iki ağ kullanılması

İki farklı ağ olan TOKYO içine yeni bir mağaza eklemek için bu yönergeleri izleyin. Her ikisinin de Tokyo 'daki kuyruk yöneticisiyle iletişim kurmak için kullanılabilir olması gerekir.

Kümede birincil ve ikincil ağ kullanılması

Bir ağı birincil ağ, diğerini yedek ağ yapmak için bu yönergeleri izleyin. Birincil ağla ilgili bir sorun varsa yedek ağı kullanın.

Yedek olarak işlev görecektir bir kuyruk ekleme

Şu anda New York 'ta çalışan envanter sistemi için Chicago 'da yedek sağlamak üzere bu yönergeleri izleyin. Chicago sistemi yalnızca New York sistemiyle ilgili bir sorun olduğunda kullanılır.

Kullanılan kanal sayısının kısıtlanması

Çeşitli kuyruk yöneticilerine bir fiyat denetimi uygulaması kurulduğunda her bir sunucunun çalıştığı etkin kanal sayısını sınırlamak için bu yönergeleri izleyin.

Bir kuyruğu barındıran daha güçlü bir kuyruk yöneticisi ekleme

Los Angeles 'ta ve New York 'ta envanter sistemini çalıştırarak ek kapasite sağlamak için bu yönergeleri izleyin. Burada Los Angeles, New York 'tan iki kat daha fazla ileti işleyebiliyor.

### **Kümede iki ağ kullanılması**

İki farklı ağ olan TOKYO içine yeni bir mağaza eklemek için bu yönergeleri izleyin. Her ikisinin de Tokyo 'daki kuyruk yöneticisiyle iletişim kurmak için kullanılabilir olması gerekir.

### **Başlamadan önce**

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- INVENTORY kümesi, "Kümede kuyruk yöneticisi eklenmesi" başlıklı konuda açıklandığı gibi ayarlandı. Üç kuyruk yöneticisi içerir; LONDON ve NEWYORK her ikisi de tam havuzu tutar, PARIS kısmi bir havuzu tutar. Döküm uygulaması, NEWYORK kuyruk yöneticisine bağlı olarak New York 'taki sistemde çalışır. Uygulama, INVENTORY kuyruğundaki iletilerin gelişyle yönlendirilir.
- İki farklı ağın bulunduğu TOKYO içine yeni bir mağaza ekleniyor. Her ikisinin de Tokyo 'daki kuyruk yöneticisiyle iletişim kurmak için kullanılabilir olması gerekir.

### **Bu görev hakkında**

Bir kümede iki ağ kullanmak için aşağıdaki adımları izleyin.

### **Yordam**

1. TOKYO 'nin önce hangi tam havuza başvuruda bulunacağına karar verin.

Bir kümedeki her kuyruk yöneticisi, kümeyle ilgili bilgi toplamak için tam havuzlardan birine ya da diğerine başvurmalıdır. Kendi kısmi havuzunu oluşturur. Seçtiğiniz havuzun özel bir önemi yoktur. Bu örnekte NEWYORK seçilir. Yeni kuyruk yöneticisi kümeye katıldıktan sonra, her iki havuzla da iletişim kurar.

2. CLUSRCVR kanallarını tanımlayın.

Bir kümedeki her kuyruk yöneticisinin, ileti alabileceği bir küme alıcısı tanımlaması gerekir. Bu kuyruk yöneticisinin her ağ üzerinde iletişim kurabilmesi gerekir.

```
DEFINE CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME('TOKYO.NETB.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel using network B for TOKYO')
```

```
DEFINE CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME('TOKYO.NETA.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel using network A for TOKYO')
```

3. Kuyruk yöneticisinde CLUSSDR kanalı tanımlanması TOKYO .

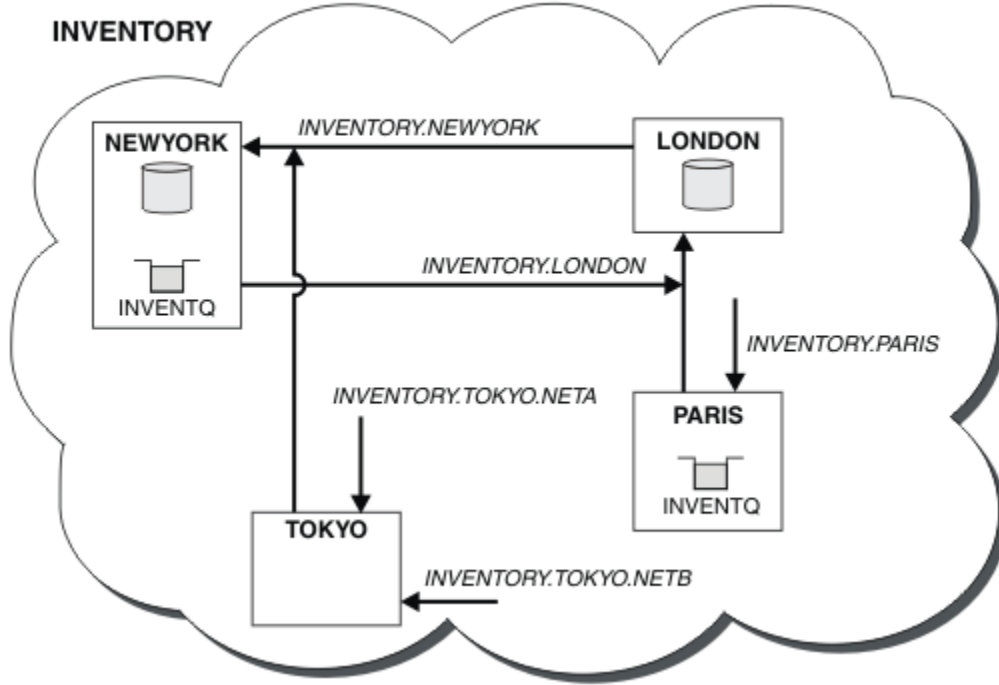
Bir kümedeki her kuyruk yöneticisinin, ilk tam havuzuna ileti gönderebildiği bir küme gönderen kanalı tanımlaması gerekir. Bu durumda NEWYORK' i seçtik, bu nedenle TOKYO aşağıdaki tanımlı gerektirir:

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
channel from TOKYO to repository at NEWYORK')
```

Tüm tanımları tamamladıktan sonra, henüz tamamlamadıysanız, kanal başlatıcısını IBM MQ for z/OS üzerinde başlatın. Tüm altyapılarda, PARIS kuyruk yöneticisinde bir dinleyici programı başlatın. Dinleyici programı gelen ağ isteklerini dinler ve gerektiğinde küme-alıcı kanalını başlatır.

## Sonuçlar

Şekil 61 sayfa 388 , bu görev tarafından ayarlanan kümeyi gösterir.



Şekil 61. Dört kuyruk yöneticisiyle INVENTORY kümesi

Yalnızca üç tanımlama yaparak, iki farklı ağ rotası bulunan TOKYO kuyruk yöneticisini kümeye ekledik.

### İlgili kavramlar

Bir kuyruğun birden çok eşgörünümüne sahip bir küme örneği

Bir kuyruğun birden çok eşgörünümüne sahip bir kümenin bu örneğinde, iletiler kuyruğun farklı eşgörünümlerine yöneltilir. Bir iletiyi kuyruğun belirli bir eşgörünümüne zorlayabilir ve kuyruk yöneticilerinden birine ileti dizisi göndermeyi seçebilirsiniz.

### Uygulama programlama ve kümeleri

Aynı kuyruktaki birden çok eşgörünümünden yararlanmak için programlama değişiklikleri yapmanız gerekmez. Ancak, kuyruğun aynı örneğine bir ileti dizisi gönderilmedikçe, bazı programlar doğru çalışmaz.

### İlgili görevler

Yerel olarak kuyruk barındıran bir kuyruk yöneticisi eklenmesi

Paris ve New York 'ta stok uygulama sistemini çalıştırmak için ek kapasite sağlamak üzere bir INVENTQ eşgörünümü eklemek üzere bu yönergeleri izleyin.

### Kümede birincil ve ikincil ağ kullanılması

Bir ağı birincil ağ, diğerini yedek ağ yapmak için bu yönergeleri izleyin. Birincil ağla ilgili bir sorun varsa yedek ağı kullanın.

### Yedek olarak işlev görecektir bir kuyruk ekleme

Şu anda New York 'ta çalışan envanter sistemi için Chicago 'da yedek sağlamak üzere bu yönergeleri izleyin. Chicago sistemi yalnızca New York sistemiyle ilgili bir sorun olduğunda kullanılır.

### Kullanılan kanal sayısının kısıtlanması

Çeşitli kuyruk yöneticilerine bir fiyat denetimi uygulaması kurulduğunda her bir sunucunun çalıştırdığı etkin kanal sayısını sınırlamak için bu yönergeleri izleyin.

### Bir kuyruğu barındıran daha güçlü bir kuyruk yöneticisi ekleme

Los Angeles 'ta ve New York 'ta envanter sistemini çalıştırarak ek kapasite sağlamak için bu yönergeleri izleyin. Burada Los Angeles, New York 'tan iki kat daha fazla ileti işleyebiliyor.



[“Kümeye kuyruk yöneticisi eklenmesi” sayfa 308](#)

Yarattığınız kümeye bir kuyruk yöneticisi eklemek için bu yönergeleri izleyin. Küme kuyruklarına ve konularına ilişkin iletiler, tek küme iletim kuyruğu SYSTEM . CLUSTER . TRANSMIT . QUEUE kullanılarak aktarılır.

### ***Kümede birincil ve ikincil ağ kullanılması***

Bir ağı birincil ağ, diğerini yedek ağ yapmak için bu yönergeleri izleyin. Birincil ağla ilgili bir sorun varsa yedek ağı kullanın.

### **Başlamadan önce**

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- INVENTORY kümesi, [“Kümede iki ağ kullanılması” sayfa 387](#) içinde açıklandığı gibi ayarlandı. Dört kuyruk yöneticisi içerir; LONDON ve NEWYORK her ikisi de tam havuzları tutar; PARIS ve TOKYO kısmi havuzları tutar. Döküm uygulaması, NEWYORK kuyruk yöneticisine bağlı olarak New York 'ta sistemde çalışır. TOKYO kuyruk yöneticisinin iletişim kurabileceği iki farklı ağı vardır.
- Ağlardan birini birincil ağ, başka bir ağı da yedek ağ yapmak istiyorsunuz. Birincil ağla ilgili bir sorun varsa, yedekleme ağını kullanmayı planlıyorsunuz.

### **Bu görev hakkında**

Bir kümedeki birincil ve ikincil ağı yapılandırmak için NETPRTY özniteliğini kullanın.

### **Yordam**

TOKYO üzerinde var olan CLUSRCVR kanallarını değiştirin.

Ağ A kanalının birincil kanal olduğunu ve ağ B kanalının ikincil kanal olduğunu belirtmek için aşağıdaki komutları kullanın:

- a) ALTER CHANNEL (INVENTORY.TOKYO.NETA) CHLTYPE (CLUSRCVR) NETPRTY (2) DESCR ('Main cluster-receiver channel for TOKYO')
- b) ALTER CHANNEL (INVENTORY.TOKYO.NETB) CHLTYPE (CLUSRCVR) NETPRTY (1) DESCR ('Backup cluster-receiver channel for TOKYO')

### **Sonraki adım**

Kanalı farklı ağ öncelikleriyle yapılandırarak, artık birincil ağınız ve ikincil ağınız olan kümede tanımladınız. Bu kanalları kullanan kümedeki kuyruk yöneticileri, kullanılabilir olduğunda birincil ağı otomatik olarak kullanır. Birincil ağ kullanılmadığında, kuyruk yöneticileri ikincil ağı kullanmak için hata durumunda yedek sisteme geçiş.

### **İlgili kavramlar**

[Bir kuyruğun birden çok eşgörünümüne sahip bir küme örneği](#)

[Bir kuyruğun birden çok eşgörünümüne sahip bir kümenin bu örneğinde, iletiler kuyruğun farklı eşgörünümlerine yöneltilir. Bir iletiyi kuyruğun belirli bir eşgörünümüne zorlayabilir ve kuyruk yöneticilerinden birine ileti dizisi göndermeyi seçebilirsiniz.](#)

[Uygulama programlama ve kümeleri](#)

Aynı kuyruktaki birden çok eşgörünümünden yararlanmak için programlama değişiklikleri yapmanız gerekmez. Ancak, kuyruğun aynı örneğine bir ileti dizisi gönderilmedikçe, bazı programlar doğru çalışmaz.

### **İlgili görevler**

[Yerel olarak kuyruk barındıran bir kuyruk yöneticisi eklenmesi](#)

[Paris ve New York 'ta stok uygulama sistemini çalıştırmak için ek kapasite sağlamak üzere bir INVENTQ eşgörünümü eklemek üzere bu yönergeleri izleyin.](#)

[Kümede iki ağ kullanılması](#)

İki farklı ağ olan TOKYO içine yeni bir mağaza eklemek için bu yönergeleri izleyin. Her ikisinin de Tokyo 'daki kuyruk yöneticisiyle iletişim kurmak için kullanılabilir olması gerekir.

#### Yedek olarak işlev görecek bir kuyruk ekleme

Şu anda New York 'ta çalışan envanter sistemi için Chicago 'da yedek sağlamak üzere bu yönergeleri izleyin. Chicago sistemi yalnızca New York sistemiyle ilgili bir sorun olduğunda kullanılır.

#### Kullanılan kanal sayısının kısıtlanması

Çeşitli kuyruk yöneticilerine bir fiyat denetimi uygulaması kurulduğunda her bir sunucunun çalıştırdığı etkin kanal sayısını sınırlamak için bu yönergeleri izleyin.

#### Bir kuyruğu barındıran daha güçlü bir kuyruk yöneticisi ekleme

Los Angeles 'ta ve New York 'ta envanter sistemini çalıştırarak ek kapasite sağlamak için bu yönergeleri izleyin. Burada Los Angeles, New York 'tan iki kat daha fazla ileti işleyebiliyor.

### **Yedek olarak işlev görecek bir kuyruk ekleme**

Şu anda New York 'ta çalışan envanter sistemi için Chicago 'da yedek sağlamak üzere bu yönergeleri izleyin. Chicago sistemi yalnızca New York sistemiyle ilgili bir sorun olduğunda kullanılır.

## **Başlamadan önce**

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- INVENTORY kümesi, “Kümeye kuyruk yöneticisi eklenmesi” sayfa 308’inde açıklandığı gibi ayarlandı. Üç kuyruk yöneticisi içerir; LONDON ve NEWYORK her ikisi de tam havuzu tutar, PARIS kısmi bir havuzu tutar. Döküm uygulaması, NEWYORK kuyruk yöneticisine bağlı olarak New York 'taki sistemde çalışır. Uygulama, INVENTQ kuyruğundaki iletilerin gelişyle yönlendirilir.
- Chicago 'da, şu anda New York 'ta çalışan envanter sistemi için bir yedek sağlamak üzere yeni bir mağaza kuruluyor. Chicago sistemi sadece New York sistemiyle ilgili bir sorun olduğunda kullanılır.

## **Bu görev hakkında**

Yedek olarak işlev görecek bir kuyruk eklemek için aşağıdaki adımları izleyin.

### **Yordam**

1. CHICAGO ' nin önce hangi tam havuza başvuruda bulunacağına karar verin.

Bir kümedeki her kuyruk yöneticisi, kümeyle ilgili bilgi toplamak için tam havuzlardan birine ya da diğerine başvurmalıdır. Kendi kısmi havuzunu oluşturur. Belirli bir kuyruk yöneticisi için seçtiğiniz havuz özel bir önem taşımaz. Bu örnekte NEWYORK seçilir. Yeni kuyruk yöneticisi kümeye katıldıktan sonra, her iki havuzla da iletişim kurar.

2. CLUSRCVR kanalını tanımlayın.

Bir kümedeki her kuyruk yöneticisinin, ileti alabileceği bir küme alıcısı tanımlaması gerekir. CHICAGO üzerinde aşağıdakileri tanımlayın:

```
DEFINE CHANNEL(INVENTORY.CHICAGO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(CHICAGO.CMSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel for CHICAGO')
```

3. CHICAGO kuyruk yöneticisinde bir CLUSSDR kanalı tanımlayın.

Bir kümedeki her kuyruk yöneticisinin, ilk tam havuzuna ileti gönderebildiği bir küme gönderen kanalı tanımlaması gerekir. Bu durumda NEWYORK 'i seçtik, bu nedenle CHICAGO aşağıdaki tanımı gerektirir:

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
channel from CHICAGO to repository at NEWYORK')
```

4. Var olan küme kuyruğunu değiştirin INVENTQ.

NEWYORK kuyruk yöneticisi tarafından zaten barındırılan INVENTQ , kuyruğun ana eşgörünümüdür.

```
ALTER QLOCAL (INVENTQ) CLWLPRTY (2)
```

5. İleti yakınlıkları için stok uygulamasını gözden geçirin.

Devam etmeden önce, stok uygulamasının ileti işleme sırasına bağımlı olmadığından emin olun.

6. Döküm uygulamasını sisteme CHICAGO içinde kurun.

7. Yedek küme kuyruğunun tanımlanması INVENTQ

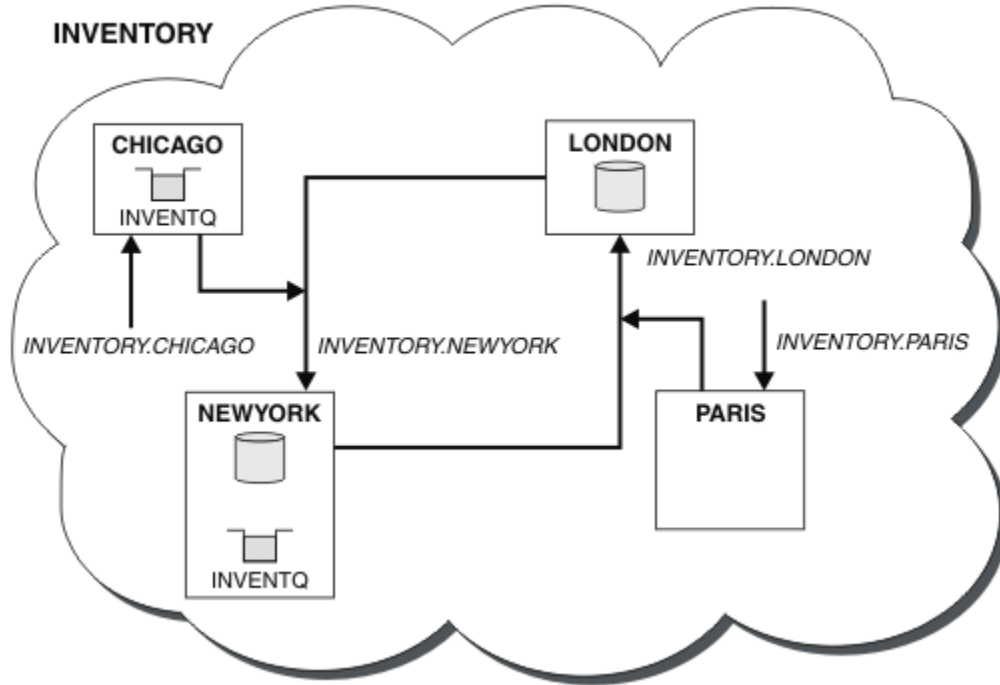
NEWYORK kuyruk yöneticisi tarafından barındırılan INVENTQ , CHICAGO tarafından yedek olarak da barındırılır. CHICAGO kuyruk yöneticisinde aşağıdaki gibi tanımlayın:

```
DEFINE QLOCAL (INVENTQ) CLUSTER (INVENTORY) CLWLPRTY (1)
```

Tüm tanımları tamamladıktan sonra, henüz tamamlamadıysanız, kanal başlatıcısını IBM MQ for z/OS üzerinde başlatın. Tüm altyapılarda, CHICAGO kuyruk yöneticisinde bir dinleyici programı başlatın. Dinleyici programı gelen ağ isteklerini dinler ve gerektiğinde küme-alıcı kanalını başlatır.

## Sonuçlar

Şekil 62 sayfa 391 , bu görev tarafından ayarlanan kümeyi gösterir.



Şekil 62. Dört kuyruk yöneticisiyle INVENTORY kümesi

INVENTQ kuyruğu ve döküm uygulaması artık kümedeki iki kuyruk yöneticisinde barındırılır. CHICAGO kuyruk yöneticisi bir yedektir. INVENTQ 'e yerleştirilen iletiler, CHICAGO' e gönderildiklerinde kullanılmadıkları sürece NEWYORK ' e yönlendirilir.

### Not:

Uzak kuyruk yöneticisinin kullanılabilirliği, o kuyruk yöneticisine ilişkin kanalın durumuna bağlıdır. Kanallar başladığında, durum birkaç kez değişir ve bazı durumlar küme iş yükü yönetimi algoritmasına göre daha az tercih edilir. Uygulamada bu, daha yüksek öncelikli (birincil) hedeflere kanallar başlatılırken düşük öncelikli (yedek) hedeflerin seçilebileceği anlamına gelir.

Yedekleme hedefine ileti gitmediğinden emin olmanız gerekiyorsa, CLWLPRTY komutunu kullanmayın. Ayrı kuyruklar kullanmayı ya da CLWLRANK komutunu, birincil anahtardan yedek sisteme el ile geçiş ile kullanmayı düşünün.

## İlgili kavramlar

Bir kuyruğun birden çok eşgörünümüne sahip bir küme örneği

Bir kuyruğun birden çok eşgörünümüne sahip bir kümenin bu örneğinde, iletiler kuyruğun farklı eşgörünümüne yöneltilir. Bir iletiyi kuyruğun belirli bir eşgörünümüne zorlayabilir ve kuyruk yöneticilerinden birine ileti dizisi göndermeyi seçebilirsiniz.

Uygulama programlama ve kümeleri

Aynı kuyruktaki birden çok eşgörünümünden yararlanmak için programlama değişiklikleri yapmanız gerekmez. Ancak, kuyruğun aynı örneğine bir ileti dizisi gönderilmedikçe, bazı programlar doğru çalışmaz.

## İlgili görevler

Yerel olarak kuyruk barındıran bir kuyruk yöneticisi eklenmesi

Paris ve New York 'ta stok uygulama sistemini çalıştırmak için ek kapasite sağlamak üzere bir INVENTQ eşgörünümü eklemek üzere bu yönergeleri izleyin.

Kümede iki ağ kullanılması

İki farklı ağ olan TOKYO içine yeni bir mağaza eklemek için bu yönergeleri izleyin. Her ikisinin de Tokyo 'daki kuyruk yöneticisiyle iletişim kurmak için kullanılabilir olması gerekir.

Kümede birincil ve ikincil ağ kullanılması

Bir ağı birincil ağ, diğerini yedek ağ yapmak için bu yönergeleri izleyin. Birincil ağla ilgili bir sorun varsa yedek ağı kullanın.

Kullanılan kanal sayısının kısıtlanması

Çeşitli kuyruk yöneticilerine bir fiyat denetimi uygulaması kurulduğunda her bir sunucunun çalıştırdığı etkin kanal sayısını sınırlamak için bu yönergeleri izleyin.

Bir kuyruğu barındıran daha güçlü bir kuyruk yöneticisi ekleme

Los Angeles 'ta ve New York 'ta envanter sistemini çalıştırarak ek kapasite sağlamak için bu yönergeleri izleyin. Burada Los Angeles, New York 'tan iki kat daha fazla ileti işleyebiliyor.

## ***Kullanılan kanal sayısının kısıtlanması***

Çeşitli kuyruk yöneticilerine bir fiyat denetimi uygulaması kurulduğunda her bir sunucunun çalıştırdığı etkin kanal sayısını sınırlamak için bu yönergeleri izleyin.

## Başlamadan önce

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- Çeşitli kuyruk yöneticilerine bir fiyat denetimi uygulaması kurulacak. Kullanılmakta olan kanal sayısını düşük bir sayıyla tutmak için, her sunucunun çalıştırdığı etkin kanal sayısı sınırlıdır. Uygulama, PRICEQ kuyruğundaki iletilerin gelişle yönlendirilir.
- Dört sunucu kuyruk yöneticisi fiyat denetimi uygulamasını barındırıyor. İki sorgu kuyruğu yöneticisi, bir fiyatı sorgulamak için PRICEQ ' e ileti gönderir. İki kuyruk yöneticisi daha tam havuz olarak yapılandırıldı.

## Bu görev hakkında

Kullanılan kanal sayısını sınırlamak için aşağıdaki adımları izleyin.

### Yordam

1. İki tam havuz seçin.

Fiyat denetimi kümeniz için tam havuz olacak iki kuyruk yöneticisi seçin. Bunlara REPOS1 ve REPOS2adı verilir.

Şu komutu verin:

```
ALTER QMGR REPOS (PRICECHECK)
```

2. Her kuyruk yöneticisinde bir CLUSRCVR kanalı tanımlayın.

Kümedeki her kuyruk yöneticisinde, bir küme alıcı kanalı ve bir küme gönderen kanalı tanımlayın. Hangisinin önce tanımlandığı önemli değil.

```
DEFINE CHANNEL (PRICECHECK.SERVE1) CHLTYPE (CLUSRCVR) TRPTYPE (TCP)  
CONNNAME (SERVER1.COM) CLUSTER (PRICECHECK) DESCR ('Cluster-receiver channel')
```

3. Her kuyruk yöneticisinde bir CLUSSDR kanalı tanımlayın.

Her kuyruk yöneticisinde, o kuyruk yöneticisini tam havuz kuyruğu yöneticilerinden birine ya da diğerine bağlamak için bir CLUSSDR tanımlaması yapın.

```
DEFINE CHANNEL (PRICECHECK.REPOS1) CHLTYPE (CLUSSDR) TRPTYPE (TCP)  
CONNNAME (REPOS1.COM) CLUSTER (PRICECHECK) DESCR ('Cluster-sender channel to  
repository queue manager')
```

4. Fiyat denetimi uygulamasını kurun.
5. PRICEQ kuyruğunu tüm sunucu kuyruğu yöneticilerine tanımlayın.

Her biri için aşağıdaki komutu verin:

```
DEFINE QLOCAL (PRICEQ) CLUSTER (PRICECHECK)
```

6. Sorgular tarafından kullanılan kanal sayısını sınırla

Sorgu kuyruğu yöneticilerimizde, her biri için aşağıdaki komutları vererek, kullanılan etkin kanal sayısını kısıtlarız:

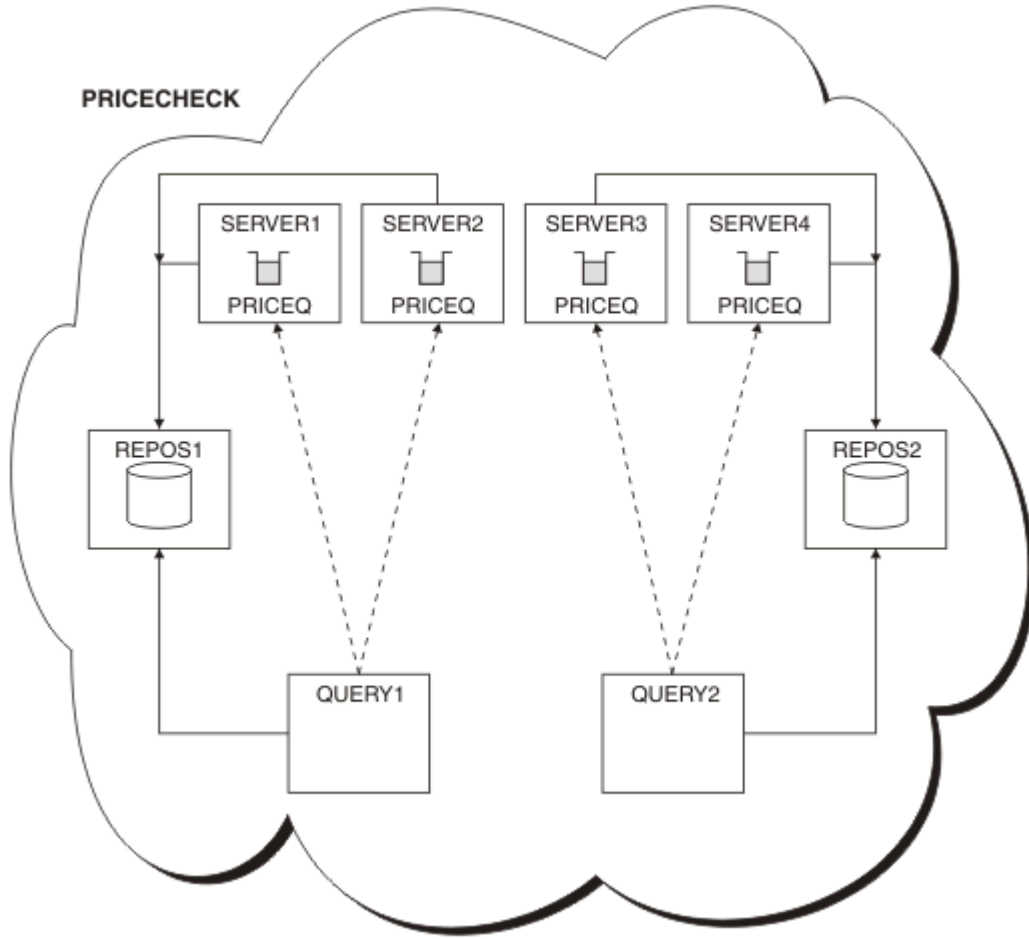
```
ALTER QMGR CLWLMRUC (2)
```

7. Henüz başlatmadıysanız, kanal başlatıcıyı IBM MQ for z/OS üzerinde başlatın. Tüm platformlarda bir dinleyici programı başlatın.

Dinleyici programı gelen ağ isteklerini dinler ve gerektiğinde küme-alıcı kanalını başlatır.

## Sonuçlar

[Şekil 63 sayfa 394](#) , bu görev tarafından ayarlanan kümeyi gösterir.



Şekil 63. Dört sunucu kuyruğu yöneticisi, iki havuz ve iki sorgu kuyruğu yöneticisi içeren PRICECHECK kümesi

PRICECHECK kümesinde kullanılabilir olan PRICEQ kuyruğunun dört eşgörünümü olsa da, her sorgulama kuyruk yöneticisi bunlardan yalnızca ikisini kullanır. Örneğin, QUERY1 kuyruk yöneticisinin yalnızca SERVER1 ve SERVER2 kuyruk yöneticilerine yönelik etkin kanalları vardır. SERVER1 kullanılmazsa, QUERY1 kuyruk yöneticisi başka bir kuyruk yöneticisini kullanmaya başlar; örneğin, SERVER3.

### İlgili kavramlar

Bir kuyruğun birden çok eşgörünümüne sahip bir küme örneği

Bir kuyruğun birden çok eşgörünümüne sahip bir kümenin bu örneğinde, iletiler kuyruğun farklı eşgörünümlerine yöneltilir. Bir iletiyi kuyruğun belirli bir eşgörünümüne zorlayabilir ve kuyruk yöneticilerinden birine ileti dizisi göndermeyi seçebilirsiniz.

Uygulama programlama ve kümeleri

Aynı kuyruktaki birden çok eşgörünümden yararlanmak için programlama değişiklikleri yapmanız gerekmez. Ancak, kuyruğun aynı örneğine bir ileti dizisi gönderilmedikçe, bazı programlar doğru çalışmaz.

### İlgili görevler

Yerel olarak kuyruk barındıran bir kuyruk yöneticisi eklenmesi

Paris ve New York 'ta stok uygulama sistemini çalıştırmak için ek kapasite sağlamak üzere bir INVENTQ eşgörünümü eklemek üzere bu yönergeleri izleyin.

Kümede iki ağ kullanılması

İki farklı ağ olan TOKYO içine yeni bir mağaza eklemek için bu yönergeleri izleyin. Her ikisinin de Tokyo 'daki kuyruk yöneticisiyle iletişim kurmak için kullanılabilir olması gerekir.

Kümede birincil ve ikincil ağ kullanılması

Bir ağı birincil ağ, diğerini yedek ağ yapmak için bu yönergeleri izleyin. Birincil ağla ilgili bir sorun varsa yedek ağı kullanın.

#### Yedek olarak işlev görecek bir kuyruk ekleme

Şu anda New York 'ta çalışan envanter sistemi için Chicago 'da yedek sağlamak üzere bu yönergeleri izleyin. Chicago sistemi yalnızca New York sistemiyle ilgili bir sorun olduğunda kullanılır.

#### Bir kuyruğu barındıran daha güçlü bir kuyruk yöneticisi ekleme

Los Angeles 'ta ve New York 'ta envanter sistemini çalıştırarak ek kapasite sağlamak için bu yönergeleri izleyin. Burada Los Angeles, New York 'tan iki kat daha fazla ileti işleyebiliyor.

### ***Bir kuyruğu barındıran daha güçlü bir kuyruk yöneticisi ekleme***

Los Angeles 'ta ve New York 'ta envanter sistemini çalıştırarak ek kapasite sağlamak için bu yönergeleri izleyin. Burada Los Angeles, New York 'tan iki kat daha fazla ileti işleyebiliyor.

## **Başlamadan önce**

**Not:** Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Senaryo:

- INVENTORY kümesi, “Kümeye kuyruk yöneticisi eklenmesi” sayfa 308’inde açıklandığı gibi ayarlandı. Üç kuyruk yöneticisi içerir: LONDON ve NEWYORK her ikisi de tam havuzları tutar, PARIS kısmi bir havuzu tutar ve INVENTQ' den gelen iletileri yerleştirir. Döküm uygulaması, NEWYORK kuyruk yöneticisine bağlı olarak New York 'ta çalışır. Uygulama, INVENTQ kuyruğundaki iletilerin gelişyle yönlendirilir.
- Los Angeles 'ta yeni bir mağaza kuruluyor. Ek kapasite sağlamak için New York 'un yanı sıra Los Angeles 'ta da envanter sistemini çalıştırmak istiyorsunuz. Yeni kuyruk yöneticisi, New York 'un iki katı kadar iletiyi işleyebilir.

## **Bu görev hakkında**

Bir kuyruğu barındıran daha güçlü bir kuyruk yöneticisi eklemek için aşağıdaki adımları izleyin.

### **Yordam**

1. LOSANGELES ' nin önce hangi tam havuza başvuruda bulunacağına karar verin.
2. Bir kümedeki her kuyruk yöneticisi, kümeyle ilgili bilgi toplamak için tam havuzlardan birine ya da diğerine başvurmalıdır. Kendi kısmi havuzunu oluşturur. Seçtiğiniz havuzun özel bir önemi yoktur. Bu örnekte NEWYORK seçilir. Yeni kuyruk yöneticisi kümeye katıldıktan sonra, her iki havuzla da iletişim kurar.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from LOSANGELES to repository at NEWYORK')
```

3. LOSANGELESkuyruk yöneticisinde CLUSRCVR kanalını tanımlayın.

Bir kümedeki her kuyruk yöneticisi, ileti alabileceği bir küme-alıcı kanalı tanımlamalıdır. LOSANGELESüzerinde aşağıdakileri tanımlayın:

```
DEFINE CHANNEL(INVENTORY.LOSANGELES) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(LOSANGELES.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager LOSANGELES')
CLWLWGHT(2)
```

Küme alıcı kanalı, INVENTORYkümesindeki diğer kuyruk yöneticilerinden ileti almak için kuyruk yöneticisinin kullanılabilirliğini tanıtır. CLWLWGHT değerinin iki olarak ayarlanması, Los Angeles kuyruk yöneticisinin New York 'tan iki kat daha fazla döküm ileti almasını sağlar ( NEWYORK kanalı bir olarak ayarlandığında).

4. NEWYORKkuyruk yöneticisinde CLUSRCVR kanalını değiştirin.

Los Angeles kuyruk yöneticisinin, New York 'tan iki kat daha fazla döküm iletisi aldığından emin olun. Küme-alıcı kanalının tanımını değiştirin.

```
ALTER CHANNEL (INVENTORY.NEWYORK) CHLTYPE (CLUSRCVR) CLWLWGHT (1)
```

5. İleti yakınlıkları için stok uygulamasını gözden geçirin.

Devam etmeden önce, stok uygulamasının ileti işleme sırasına bağımlı olmadığından emin olun.

6. Döküm uygulamasını Los Angeles 'ta sisteme kur

7. INVENTQküme kuyruğunu tanımlayın.

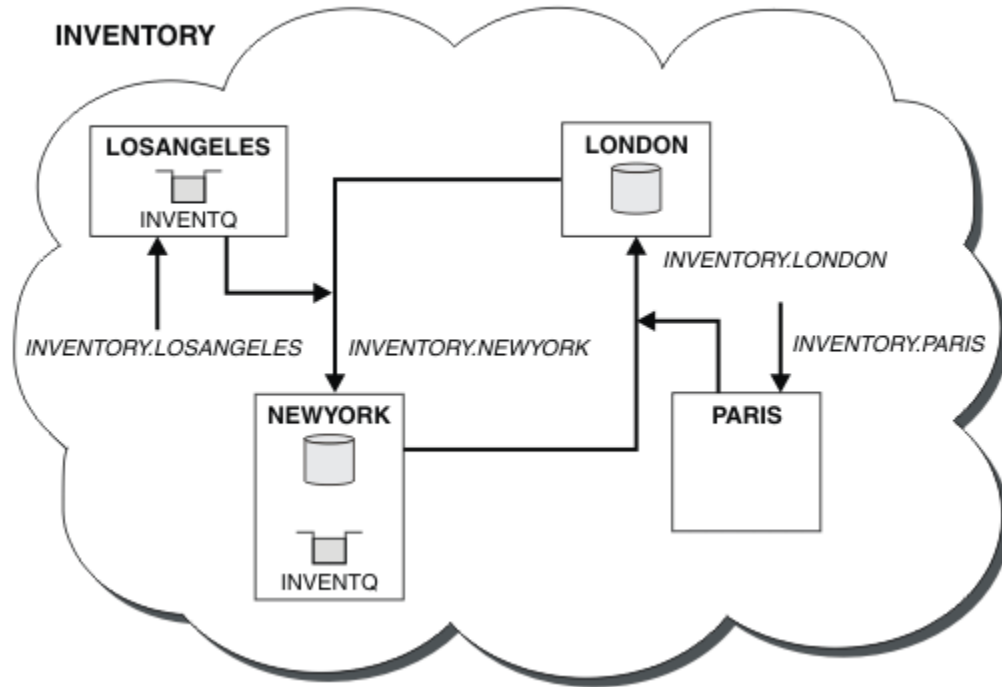
NEWYORK kuyruk yöneticisi tarafından barındırılan INVENTQ kuyruğu da LOSANGELEStarafından barındırılır. LOSANGELES kuyruk yöneticisinde aşağıdaki gibi tanımlayın:

```
DEFINE QLOCAL (INVENTQ) CLUSTER (INVENTORY)
```

Tüm tanımları tamamladıktan sonra, henüz tamamlamadıysanız, kanal başlatıcısını IBM MQ for z/OSüzerinde başlatın. Tüm altyapılarda, LOSANGELESkuyruk yöneticisinde bir dinleyici programı başlatın. Dinleyici programı gelen ağ isteklerini dinler ve gerektiğinde küme-alıcı kanalını başlatır.

## Sonuçlar

“Bir kuyruğu barındıran daha güçlü bir kuyruk yöneticisi ekleme” sayfa 395 , bu görev tarafından ayarlanan kümeyi gösterir.



Şekil 64. Dört kuyruk yöneticisiyle INVENTORY kümesi

Kümede yapılan bu değişiklik, kuyruk yöneticilerini LONDON ve PARISdeğiştirmeniz gerekmeden gerçekleştirildi. Bu kuyruk yöneticilerindeki havuzlar, LOSANGELESadresindeki INVENTQ ' e ileti gönderebilmeleri için gereksinim duydukları bilgilerle otomatik olarak güncellenir.

## Sonraki adım

INVENTQ kuyruk ve döküm uygulaması, kümedeki iki kuyruk yöneticisinde barındırılır. Yapılandırma, kullanılabilirliğini artırır, iletilerin verimini hızlandırır ve iş yükünün iki kuyruk yöneticisi arasında dağıtılmasına olanak sağlar. INVENTQ ' e LOSANGELES ya da NEWYORK tarafından konan iletiler, mümkün



olduğunda yerel kuyruk yöneticisindeki eşgörünüm tarafından işlenir. LONDON ya da PARIS tarafından konan iletiler LOSANGELES ya da NEWYORK'e yöneltilir ve LOSANGELES' e iki kat daha fazla ileti gönderilir.

### **İlgili kavramlar**

Bir kuyruğun birden çok eşgörünümüne sahip bir küme örneği

Bir kuyruğun birden çok eşgörünümüne sahip bir kümenin bu örneğinde, iletiler kuyruğun farklı eşgörünümüne yöneltilir. Bir iletiyi kuyruğun belirli bir eşgörünümüne zorlayabilir ve kuyruk yöneticilerinden birine ileti dizisi göndermeyi seçebilirsiniz.

Uygulama programlama ve kümeleri

Aynı kuyruktaki birden çok eşgörünümünden yararlanmak için programlama değişiklikleri yapmanız gerekmez. Ancak, kuyruğun aynı örneğine bir ileti dizisi gönderilmedikçe, bazı programlar doğru çalışmaz.

### **İlgili görevler**

Yerel olarak kuyruk barındıran bir kuyruk yöneticisi eklenmesi

Paris ve New York 'ta stok uygulama sistemini çalıştırmak için ek kapasite sağlamak üzere bir INVENTQ eşgörünümü eklemek üzere bu yönergeleri izleyin.

Kümede iki ağ kullanılması

İki farklı ağ olan TOKYO içine yeni bir mağaza eklemek için bu yönergeleri izleyin. Her ikisinin de Tokyo 'daki kuyruk yöneticisiyle iletişim kurmak için kullanılabilir olması gerekir.

Kümede birincil ve ikincil ağ kullanılması

Bir ağ birincil ağ, diğerini yedek ağ yapmak için bu yönergeleri izleyin. Birincil ağla ilgili bir sorun varsa yedek ağ kullanın.

Yedek olarak işlev görecek bir kuyruk ekleme

Şu anda New York 'ta çalışan envanter sistemi için Chicago 'da yedek sağlamak üzere bu yönergeleri izleyin. Chicago sistemi yalnızca New York sistemiyle ilgili bir sorun olduğunda kullanılır.

Kullanılan kanal sayısının kısıtlanması

Çeşitli kuyruk yöneticilerine bir fiyat denetimi uygulaması kurulduğunda her bir sunucunun çalıştırdığı etkin kanal sayısını sınırlamak için bu yönergeleri izleyin.

### **Uygulama programlama ve kümeleri**

Aynı kuyruktaki birden çok eşgörünümünden yararlanmak için programlama değişiklikleri yapmanız gerekmez. Ancak, kuyruğun aynı örneğine bir ileti dizisi gönderilmedikçe, bazı programlar doğru çalışmaz.

Uygulamalar, MQOPEN çağrılarını kullanarak bir kuyruk açabilir. Uygulamalar, iletileri açık bir kuyruğa koymak için MQPUT aramasını kullanır. Uygulamalar, MQPUT1 çağrılarını kullanarak açık olmayan bir kuyruğa tek bir ileti yerleştirebilirler.

Aynı kuyruğun birden çok eşgörünümüne sahip kümeleri ayarlarsanız, belirli bir uygulama programlamasıyla ilgili dikkate alınacak noktalar yoktur. Ancak, kümelemenin iş yükü yönetimi özelliklerinden yararlanmak için uygulamalarınızı değiştirmeniz gerekebilir. Aynı kuyrukta birden çok tanımlama bulunan bir ağ oluşturursanız, ileti yakınlıkları için uygulamalarınızı gözden geçirin.

Örneğin, aralarında soru ve yanıt biçiminde akan bir dizi iletiye dayanan iki uygulamanız olduğunu varsayın. Muhtemelen yanıtların, bir soru gönderen aynı kuyruk yöneticisine geri dönmesini istiyorsunuz. İş yükü yönetimi yordamının, iletileri yanıt kuyruğunun bir kopyasını barındıran herhangi bir kuyruk yöneticisine göndermemesi önemlidir.

İletilerin sırayla işlenmesini gerektiren uygulamalarınız olabilir (örneğin, sırayla alınması gereken ileti grupları gönderen bir veritabanı eşleme uygulaması). Bölümlenmiş iletilerin kullanılması da bir benzerlik sorununa neden olabilir.

### **Hedef kuyruğun yerel ya da uzak bir sürümünün açılması**

Kuyruk yöneticisinin hedef kuyruğun yerel ya da uzak bir sürümünü kullanmayı nasıl seçtiğini unutmayın.

1. Kuyruk yöneticisi, iletileri okumak ya da kuyruğun özniteliklerini ayarlamak için hedef kuyruğun yerel sürümünü açar.

2. Kuyruk yöneticisi, aşağıdaki koşullardan en az biri doğruysa, iletilerin yazılacağı hedef kuyruğun herhangi bir eşgörünümünü açar:

- Hedef kuyruğun yerel bir sürümü yok.
- Kuyruk yöneticisi, ALTER QMGRüzerinde CLWLUSEQ (ANY) belirtiyor.
- Kuyruk yöneticisindeki kuyruk CLWLUSEQ (ANY) belirtiyor.

### **İlgili kavramlar**

Bir kuyruğun birden çok eşgörünümüne sahip bir küme örneği

Bir kuyruğun birden çok eşgörünümüne sahip bir kümenin bu örneğinde, iletiler kuyruğun farklı eşgörünümlerine yöneltilir. Bir iletiyi kuyruğun belirli bir eşgörünümüne zorlayabilir ve kuyruk yöneticilerinden birine ileti dizisi göndermeyi seçebilirsiniz.

### **İlgili görevler**

Yerel olarak kuyruk barındıran bir kuyruk yöneticisi eklenmesi

Paris ve New York 'ta stok uygulama sistemini çalıştırmak için ek kapasite sağlamak üzere bir INVENTO eşgörünümü eklemek üzere bu yönergeleri izleyin.

Kümede iki ağ kullanılması

İki farklı ağ olan TOKYO içine yeni bir mağaza eklemek için bu yönergeleri izleyin. Her ikisinin de Tokyo 'daki kuyruk yöneticisiyle iletişim kurmak için kullanılabilir olması gerekir.

Kümede birincil ve ikincil ağ kullanılması

Bir ağı birincil ağ, diğerini yedek ağ yapmak için bu yönergeleri izleyin. Birincil ağla ilgili bir sorun varsa yedek ağı kullanın.

Yedek olarak işlev görecektir bir kuyruk ekleme

Şu anda New York 'ta çalışan envanter sistemi için Chicago 'da yedek sağlamak üzere bu yönergeleri izleyin. Chicago sistemi yalnızca New York sistemiyle ilgili bir sorun olduğunda kullanılır.

Kullanılan kanal sayısının kısıtlanması

Çeşitli kuyruk yöneticilerine bir fiyat denetimi uygulaması kurulduğunda her bir sunucunun çalıştığı etkin kanal sayısını sınırlamak için bu yönergeleri izleyin.

Bir kuyruğu barındıran daha güçlü bir kuyruk yöneticisi ekleme

Los Angeles 'ta ve New York 'ta envanter sistemini çalıştırarak ek kapasite sağlamak için bu yönergeleri izleyin. Burada Los Angeles, New York 'tan iki kat daha fazla ileti işleyebiliyor.

### *İleti yakınlıkları işlenmektedir*

Mesaj yakınlıkları nadiren iyi programlama tasarımının bir parçasıdır. Kümelemeyi tam olarak kullanmak için ileti yakınlıkları kaldırılmalıdır. İleti benzerliklerini kaldıramıyorsanız, aynı kanal ve aynı kuyruk yöneticisini kullanarak ilgili iletilerin teslim edilmesini zorlayabilirsiniz.

İleti yakınlıkları olan uygulamalarınız varsa, kümeleri kullanmaya başlamadan önce bu yakınlıkları kaldırın.

İleti yakınlıklarının kaldırılması, uygulamaların kullanılabilirliğini artırır. Bir uygulama, bir kuyruk yöneticisine ileti yakınlıkları olan bir ileti kümesi gönderir. Toplu işin yalnızca bir kısmı alındıktan sonra kuyruk yöneticisi başarısız olur. Gönderen kuyruk yöneticisinin, daha fazla ileti göndermeden önce tamamlanmamış ileti kümesini kurtarmasını ve işlemlerini beklemesi gerekir.

İleti yakınlıklarının kaldırılması, uygulamaların ölçeklenebilirliğini de artırır. Yakınlıkları olan bir ileti kümesi, sonraki iletileri beklerken hedef kuyruk yöneticisindeki kaynakları kilitleyebilir. Bu kaynaklar uzun süre kilitli kalarak diğer uygulamaların işlerini yapmasını önleyebilir.

Ayrıca, ileti yakınlıkları, küme iş yükü yönetimi yordamlarının kuyruk yöneticisi için en iyi seçimi yapmasını önler.

Yakınlıkları kaldırmak için aşağıdaki olasılıkları göz önünde bulundurun:

- İletilerde durum bilgilerini taşıma
- Db2 veritabanı gibi herhangi bir kuyruk yöneticisinin erişebileceği geçici olmayan saklama alanında durum bilgilerinin tutulması
- Birden çok kuyruk yöneticisi tarafından erişilebilmesi için salt okunur verileri eşleme

İleti yakınlıklarınızı kaldırmak için uygulamalarınızı değiştirmek uygun değilse, soruna ilişkin bir dizi olası çözüm vardır.

### **MQOPEN çağrısında belirli bir hedefi adlandır**

Her MQOPEN çağrısında uzak kuyruk adını ve kuyruk yöneticisi adını belirtin ve bu nesne tanıtcısı kullanılarak kuyruğa konan tüm iletiler aynı kuyruk yöneticisine (yerel kuyruk yöneticisi olabilir) gider.

Her MQOPEN çağrısında uzak kuyruk adının ve kuyruk yöneticisi adının belirtilmesi olumsuz yönlere sahiptir:

- İş yükü dengeleme gerçekleştirilmez. Küme iş yükü dengeleme avantajlarından yararlanmayın.
- Hedef kuyruk yöneticisi uzaksa ve birden çok kanal varsa, iletiler farklı rotalar alabilir ve ileti sırası korunmaya devam eder.
- Kuyruk yöneticinizin, hedef kuyruk yöneticisiyle aynı adı taşıyan bir iletim kuyruğu tanımı varsa, iletiler küme iletim kuyruğunda değil, o iletim kuyruğunda olur.

### **Yanıt kuyruğu yöneticisi alanında kuyruk yöneticisi adını döndür**

Bir toplu işte ilk iletiyi alan kuyruk yöneticisinin, adını yanıtında döndürmesine izin verin. Bunu, ileti tanımlayıcısının ReplyToQMGr alanını kullanarak yapar. Gönderme ucundaki kuyruk yöneticisi, yanıtlanacak kuyruk yöneticisi adını alabilir ve sonraki tüm iletilerde bu adı belirtebilir.

Yarıta ReplyToQMGr bilgilerinin kullanılması dezavantajlara sahiptir:

- İstekte bulunan kuyruk yöneticisi, ilk iletisine yanıt beklemelidir
- Sonraki iletileri göndermeden önce ReplyToQMGr bilgilerini bulmak ve kullanmak için ek kod yazmanız gerekir.
- Kuyruk yöneticisine birden çok rota varsa, iletilerin sırası korunmayabilir

### **MQOPEN çağrısında MQOO\_BIND\_ON\_OPEN seçeneğini ayarlayın**

MQOPEN çağrısında MQOO\_BIND\_ON\_OPEN seçeneğini kullanarak tüm iletilerinizi aynı hedefe konmaya zorlayın. Gruptaki tüm iletilerin aynı hedefte işlendiğinden emin olmak için kümelerle ileti grupları kullanılırken MQOO\_BIND\_ON\_OPEN ya da MQOO\_BIND\_ON\_GROUP belirtilmelidir.

Bir kuyruk açarak ve MQOO\_BIND\_ON\_OPEN belirleyerek, bu kuyruğa gönderilen tüm iletileri kuyruğun aynı eşgörünümüne gönderilmeye zorlayabilirsiniz. MQOO\_BIND\_ON\_OPEN , tüm iletileri aynı kuyruk yöneticisine ve aynı rotaya bağlar. Örneğin, aynı hedefe giden bir IP rotası ve NetBIOS rotası varsa, kuyruk açıldığında bunlardan biri seçilir ve bu seçim, elde edilen nesne tanıtcısı kullanılarak aynı kuyruğa konan tüm iletiler için kabul edilir.

MQOO\_BIND\_ON\_OPEN seçeneğini belirleyerek tüm iletilerin aynı hedefe yönlendirilmesini zorlayabilirsiniz. Bu nedenle, ileti yakınlığı olan uygulamalar kesintiye uğramaz. Hedef kullanılmıyorsa, iletiler yeniden kullanılabilir oluncaya kadar iletim kuyruğunda kalır.

MQOO\_BIND\_ON\_OPEN , bir kuyruğu açtığınızda nesne tanımlayıcısında kuyruk yöneticisi adı belirtildiğinde de geçerlidir. Adı belirtilen kuyruk yöneticisine birden çok rota olabilir. Örneğin, birden çok ağ yolu olabilir ya da başka bir kuyruk yöneticisi bir diğer ad tanımlamış olabilir. MQOO\_BIND\_ON\_OPEN değini belirlerseniz, kuyruk açıldığında bir rota seçilir.

**Not:** Bu önerilen tekniktir. Ancak, kuyruk yöneticisinin bir küme kuyruğu için diğer ad yayınladığı çok sekmeli bir yapılanışta çalışmaz. Ayrıca, uygulamaların farklı ileti grupları için aynı kuyruk yöneticisinde farklı kuyruklar kullandığı durumlarda da yardımcı olur.

MQOPEN çağrısında MQOO\_BIND\_ON\_OPEN belirtmenin bir alternatifi, kuyruk tanımlarınızı değiştirmektir. Kuyruk tanımlarınızda DEFBIND(OPEN) değini belirtin ve MQOO\_BIND\_AS\_Q\_DEF varsayılan değine MQOPEN çağrısında DefBind seçeneğine izin verin.

## MQOPEN çağrısında MQ00\_BIND\_ON\_GROUP seçeneğini ayarlayın

MQOPEN aramasında MQ00\_BIND\_ON\_GROUP seçeneğini kullanarak bir gruptaki tüm iletilerinizi aynı hedefe konmaya zorlayın. Gruptaki tüm iletilerin aynı hedefte işlendiğinden emin olmak için kümelerle ileti grupları kullanılırken MQ00\_BIND\_ON\_OPEN ya da MQ00\_BIND\_ON\_GROUP belirtilmelidir.

Bir kuyruk açarak ve MQ00\_BIND\_ON\_GROUP belirleyerek, bu kuyruğa gönderilen bir gruptaki tüm iletileri kuyruğun aynı eşgörünümüne gönderilmeye zorlayabilirsiniz. MQ00\_BIND\_ON\_GROUP , bir gruptaki tüm iletileri aynı kuyruk yöneticisine ve aynı rotaya bağlar. Örneğin, aynı hedefe giden bir IP rotası ve bir NetBIOS rotası varsa, bunlardan biri kuyruk açıldığında seçilir ve bu seçim, elde edilen nesne tanıtıcısı kullanılarak aynı kuyruğa konan gruptaki tüm iletiler için geçerli olur.

MQ00\_BIND\_ON\_GROUP değerini belirleyerek, bir gruptaki tüm iletileri aynı hedefe yönlendirmeye zorlayabilirsiniz. Bu nedenle, ileti yakınlığı olan uygulamalar kesintiye uğramaz. Hedef kullanılmıyorsa, iletiler yeniden kullanılabilir oluncaya kadar iletim kuyruğunda kalır.

MQ00\_BIND\_ON\_GROUP , bir kuyruğu açtığınızda nesne tanımlayıcısında kuyruk yöneticisi adı belirtildiğinde de geçerlidir. Adı belirtilen kuyruk yöneticisine birden çok rota olabilir. Örneğin, birden çok ağ yolu olabilir ya da başka bir kuyruk yöneticisi bir diğer ad tanımlamış olabilir. MQ00\_BIND\_ON\_GROUP değerini belirlerseniz, kuyruk açıldığında bir rota seçilir.

MQ00\_BIND\_ON\_GROUP ' in etkili olması için MQPUT üzerinde MQPMO\_LOGICAL\_ORDER koyma seçeneğini eklemeniz gerekir. İletiyeye ilişkin MQMD ' de **GroupId** değerini MQGI\_NONE olarak ayarlayabilir ve iletilerin MQMD **MsgFlags** alanına aşağıdaki ileti işaretlerini eklemelisiniz:

- Gruptaki son ileti: MQMF\_LAST\_MSG\_IN\_GROUP
- Gruptaki diğer tüm iletiler: MQMF\_MSG\_IN\_GROUP

MQ00\_BIND\_ON\_GROUP belirtildiyse, ancak iletiler gruplanmamışsa, davranış MQ00\_BIND\_NOT\_FIXED ile eşdeğerdir.

**Not:** Bu, bir gruptaki iletilerin aynı hedefe gönderilmesini sağlamak için önerilen tekniktir. Ancak, kuyruk yöneticisinin bir küme kuyruğu için diğer ad yayınladığı çok sekmeli bir yapılanışta çalışmaz.

MQOPEN çağrısında MQ00\_BIND\_ON\_GROUP belirtmenin bir alternatifi, kuyruk tanımlarınızı değiştirmektir. Kuyruk tanımlarınızda DEFBIND(GROUP) değerini belirtin ve MQ00\_BIND\_AS\_Q\_DEF varsayılan değerine MQOPEN çağrısında DefBind seçeneğine izin verin.

## Özelleştirilmiş bir küme iş yükü çıkış programı yaz

Uygulamalarınızı değiştirmek yerine, bir küme iş yükü çıkış programı yazarak ileti yakınlıkları sorununu engelleyebilirsiniz. Küme iş yükü çıkış programı yazmak kolay değildir ve önerilen bir çözüm değildir. Programın, iletilerin içeriğini inceleyerek benzerliği tanıyacak şekilde tasarlanması gerekir. Benzeşimi tanıdıktan sonra, program iş yükü yönetimi yardımcı programını ilgili tüm iletileri aynı kuyruk yöneticisine yönlendirmeye zorlamak zorunda kalırdı.

Multi

## Bir örnek kümenin yapılandırılması

Tek tip kümeler, uygulamaların ölçek ve kullanılabilirlik için tasarlanmasını sağlar ve bu tek tip küme içindeki herhangi bir kuyruk yöneticisine bağlanabilir.

### Başlamadan önce

Kümeleme girişine ilişkin bilgi için bkz. [Kümeler](#). Tek biçimli kümelere giriş için bkz. [“Tek biçimli kümeler hakkında” sayfa 401](#).

### Bu görev hakkında

Tek tip kümeler, kuyruk yöneticileri arasındaki iletişim ve kuyruklar arasındaki iş yükünü dengeleme için IBM MQ kümelemesini kullanır. Ancak bunlar, aşağıdaki şekillerde tipik IBM MQ kümelerinden farklıdır:

- Tek biçimli kümeler genellikle kümede daha az sayıda kuyruk yöneticilerine sahiptir. 10 'dan fazla kuyruk yöneticisiyle tek tip küme yaratmamalısınız.
- Kümenin her üyesi neredeyse aynı yapılandırmaya sahiptir.
- Küme genellikle tek bir uygulama ya da ilgili uygulamalar grubu tarafından kullanılır.
- Kümeye bağlanan uygulama eşgörünümlerinin sayısı, kuyruk yöneticilerinin sayısından büyük ya da bu sayıya eşit olmalıdır.

Otomatik yapılandırma ve otomatik kümeleme desteğini kullanarak hem tek tip küme oluşturmayı basitleştirebilir, hem de daha sonra tek tip küme üyeleri arasında yapılandırmayı aynı tutabilir.

## Yordam

- [Tek tip kümeler hakkında bilgi edinin](#)
- [Tek tip küme yaratılması](#)
- [Tek tip küme yaratılması](#)
- [Tek Tip Kümeden Kuyruk Yöneticisinin Askıya Alınması](#)

### Multi

## Tek biçimli kümeler hakkında

Tek tip küme konuşlandırmasının amacı, uygulamaların ölçek ve kullanılabilirlik için tasarlanabilmesidir ve tek tip küme içindeki herhangi bir kuyruk yöneticisine bağlanabilir. Bu, belirli bir kuyruk yöneticisine olan bağımlılığı kaldırır ve ileti alışverişi trafiğinin daha iyi kullanılabilirliği ve iş yükü dengelemesi ile sonuçlanır.

### z/OS

Tek biçimli kümeler IBM MQ for z/OS üzerinde kullanılamaz; kuyruk paylaşım grupları, tek biçimli bir kümenin birçok yeteneğini sağlar.

Tek biçimli kümeler, yüksek kullanılabilirlikli ve yatay olarak ölçeklenmiş küçük kuyruk yöneticileri derlemi sağlayan belirli bir IBM MQ kümesi kalıbıdır. Bu kuyruk yöneticileri, bir uygulamanın onlarla tek bir grup olarak etkileşimde bulunabilmesi için neredeyse aynı şekilde yapılandırılır. Bu, uygulama eşgörünümlerinin kuyruk yöneticilerine eşit olarak dağıtılmasını otomatik olarak sağlayarak, kümedeki her kuyruk yöneticisinin kullanıldığından emin olmayı kolaylaştırır.

Tek tip kümeler, bir yöneticinin bağımsız, birbirine bağlı kuyruk yöneticilerinden oluşan bir grup oluşturmak ve yönetmek için geçmesi gereken bazı el ile adımları kaldırır. Bazı istemci bağlantısı mantığını istemciden kuyruk yöneticisine taşırlar; burada uygulama etkinliği düzeylerine ilişkin bilgiler, istemcilerde hangi kuyruk yöneticilerine bağlanmaları gerektiğine ilişkin kararları bildirebilir.

Otomatik yapılandırma ve otomatik kümeleme desteğini kullanarak, hem tek tip bir kümenin ilk oluşturulmasını basitleştirebilir, hem de tek tip küme üyeleri arasında yapılandırmayı aynı tutabilir. Bu yeteneği kullanırken, bir yapılanış kütüğü kümeyi, diğeri de birörnek kümedeki tüm kuyruk yöneticilerine uygulanacak MQSC yapılanışını gösterir. Her kuyruk yöneticisi yeniden başlatıldığında, yapılandırma yeniden uygulanır ve küme otomatik olarak oluşur. Bu özelliğin kullanılmasıyla ilgili daha fazla ayrıntı için bkz. [“Tek tip küme yaratılması” sayfa 414](#).

Tek tip bir kümeden tam olarak yararlanmak için, her bir uygulama, tercihen kuyruk yöneticileri kadar çok sayıda yönetim ortamı ile birden çok eşleşen yönetim ortamlarına da ölçeklenmelidir.

Boyutu ne olursa olsun, birden çok yetenek sağlayan bir IBM MQ kümesi:

- Kümedeki herhangi bir üye tarafından keşfedilebilen tüm kümeleme kaynaklarının dizini
- Otomatik kanal oluşturma ve bağlantılık
- İleti iş yükü dengelemesi kullanılarak birden çok eşleşen kuyruk arasında yatay ölçekleme
- Kullanılabilirliğe dayalı olarak dinamik ileti yönlendirmesi

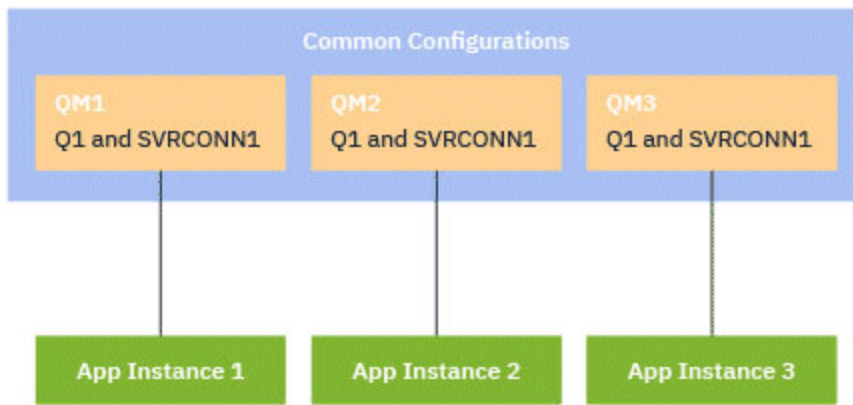
Tek tip kümeler, kuyruk yöneticileri arasındaki iletişim ve kuyruklar arasındaki iş yükünü dengeleme için IBM MQ kümelemesini kullanır. Ancak bunlar, aşağıdaki şekillerde tipik IBM MQ kümelerinden farklıdır:

- Tek biçimli kümeler genellikle kümede daha az sayıda kuyruk yöneticilerine sahiptir. 10 'dan fazla kuyruk yöneticisiyle tek tip küme yaratmamalısınız.

- Kümenin her üyesi neredeyse aynı yapılandırmaya sahiptir.
- Küme genellikle tek bir uygulama ya da ilgili uygulamalar grubu tarafından kullanılır.
- Kümeye bağlanan uygulama eşgörünümlerinin sayısı, kuyruk yöneticilerinin sayısından büyük ya da bu sayıya eşit olmalıdır.

Tek tip küme örüntüsünde, kümedeki tüm kuyruk yöneticileri aynı ileti sistemi hizmetlerini sunar. Örneğin, tüm küme üyelerini aynı yerel kuyruklara sahip olacak şekilde yapılandırabilir ve istemci uygulamalarının kümenin herhangi bir üyesine bağlanmasına izin verebilirsiniz. Aynı sunucu bağlantı kanalları ve büyük olasılıkla aynı yetki kayıtları, kanal doğrulama kuralları vb. de tanımlanmış olabilir. Ancak, küme üyelerinin nesnelere ve yapılandırmada bazı farklılıklar olabilir. Örneğin, bazı uygulamalar bir kuyruk yöneticisine bağliken geçici dinamik kuyruklar yaratabilir. Ayrıca, bazı yapılandırma güncellemeleri belirli bir süre boyunca üyeler arasında kullanıma sunulabilir; örneğin, yeni ya da güncellenmiş sertifikalar. Normal IBM MQ kümelerinde olduğu gibi, iki kuyruk yöneticisi bunları tam havuz kuyruğu yöneticileri yapmak için ek yapılandırma gerektirir.

Aşağıdaki çizge, kuyruk yöneticilerinin benzer yapılandırmalara sahip olduğunu göstermektedir. Q1 adlı aynı kuyruğu ve aynı sunucu bağlantı kanalını SVRCONN1 tanımlar.



Sunucu bağlantısı kanal adları aynı olan birden çok kuyruk yöneticisinin tek bir istemci-kanal tanımlama çizelgesiyle (CCDT) çalışabilmesi için, IBM MQ 9.1.2'inde tanımlanmış güncellenmiş CCDT biçimini kullanmanız gerektiğini unutmayın. Bkz. “JSON biçimi CCDT ' nin yapılandırılması” sayfa 44.

## Uygulama adları ve uygulama örnekleri

**DISPLAY CONN(\*) TYPE CONN** komutunun **APPLTAG** özniteliği olarak bir uygulama adı görüntülenir. IBM MQ 9.1.2' den uygulama adının ayarlanma şeklinde bir değişiklik vardır.

Bir uygulamanın eşgörünümlü, o uygulama için bir *yürütme birimi* sağlayan, yakından ilişkili bağlantılar kümesidir. Genellikle bu, bir dizi iş parçacığı ve ilişkili IBM MQ bağlantısına sahip olabilen tek bir işletim sistemi şilesidir.

Uygulama adı ve uygulama eşgörünümleriyle ilgili ek bilgi için [Uygulama geliştirme kavramları](#) başlıklı konuya bakın.

## Yeniden bağlanabilir istemciler

Yeniden bağlanabilir istemciler eşit bir iş yükü dağıtımına ulaşmak için taşınabilir; ancak, tanımlamaya göre, yeniden bağlanamayan bir istemci farklı bir kuyruk yöneticisine yeniden bağlanamaz. Ancak, yeniden bağlanamayan bir istemciyi tek tip bir kümeye bağlamak için yine de iyi bir neden olabilir: Örneğin, istemci bir tür kalıcı durum yarattığından ve uygulamanın her bir kuyruk yöneticisinde çalıştığından emin olmak için başka bir mekanizma kullanıldığından.



## Yerel olarak bağı uygulamalar

Tek tip kümelerin, yerel olarak bağı uygulamalar yerine istemci uygulamaları olarak bağılanan IBM MQ uygulamalarına sahip olması beklenir. Yerel olarak bağı uygulamaların tek tip küme üyelerine bağılanması engellenmez, ancak tek tip kümeler, kümenin başka bir üyesine bağılanamadıkları için yerel olarak bağı uygulamalarla iş yükü dağıtımını bile gerçekleştiremez.

### İlgili görevler

Desteklenen programlama dillerinde uygulama adının belirtilmesi

Multi

### Otomatik uygulama dengeleme

Otomatik uygulama dengelemesi, IBM MQ tek tip kümesinin rasgele hale getirme ya da uygulamaların belirli kuyruk yöneticilerine el ile sabitlenmesi yerine küme genelinde uygulama dağıtımını yakından yönetmesini sağlayarak uygulama dağıtımını ve kullanılabilirliğini büyük ölçüde geliştirir.

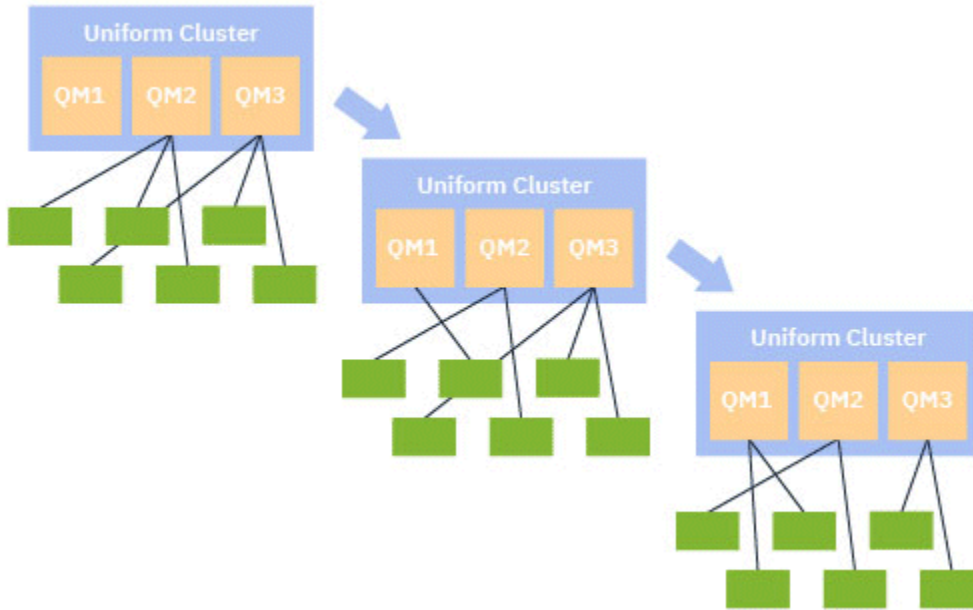
IBM MQ 9.2.0' den bir dizi kümeli kuyruk yöneticisi arasında otomatik dengeleme, C, JMS, IBM MQ .NET, XMS .NET şeklinde yazılan uygulamalar için desteklenir.

Aynı uygulamanın kuyruk yöneticileriyle aynı sayıda eşgörünümlü varsa, tek tip küme her kuyruk yöneticisinin bağı uygulamanın en az bir eşgörünümlüne sahip olmasını sürekli olarak sağlar.

Uygulamalar, bir kuyruk yöneticisine belirli bir benzerliği kaldırabilir ve bunun yerine, tek tip kümedeki kuyruk yöneticileri grubuna bağılanırlığı güvenle rasgele hale getirmek için bir istemci kanal tanımlama çizelgesi (CCDT) kullanabilir. Uygulamalar bunu aşağıdaki nedenlerle yapabilir:

- Yeterli sayıda tüketen uygulama örneği olduğunda, her zaman uygulamanın iletileri işlediği bir yönetim ortamı vardır.
- Bir kuyruk yöneticisini durdurduğunuzda, bağı uygulama eşgörünümleri kümedeki diğer kuyruk yöneticilerine eşit olarak dağıtılır.
- Bir kuyruk yöneticisini başlattığınızda, kümedeki diğer kuyruk yöneticilerine bağı uygulama eşgörünümleri, yeni başlatılan kuyruk yöneticisini içerecek şekilde otomatik olarak yeniden dengelenir.

Bu, tek tip kümenin sürekli olarak uygulamaların en iyi şekilde dağıtılmasını, planlanmış ve planlanmamış kesintilerde bile ileti işlemeyi en üst düzeye çıkarmasını sağladığı anlamına gelir.



Otomatik dengeleme sağlamak için, tek tip kümedeki kuyruk yöneticileri belirli aralıklarla kendi aralarında bilgi paylaşırlar. Bunu, konu ağacının ayrılmış \$SYS/MQ dalı altındaki sistem konularındaki meta verileri yayınlamaya yapar. Tek tip kümedeki her kuyruk yöneticisi, diğer kuyruk yöneticileri tarafından yayınlanan iletilere abone olur ve tek tip kümedeki uygulamaların durumunun bir resmini oluşturur.

Kuyruk yöneticileri, tüm kümede istemci uygulamalarının dağıtımını izler. Belirli bir kuyruk yöneticisine bağlı uygulama sayısı kümenin dengesiz olduğunu saptayabilecek kadar düşük olduğunda, kuyruk yöneticisi bir sistem konusunda bir isteği kümedeki diğer kuyruk yöneticilerinden birine yayınlar.

İleti alındığında, hedef kuyruk yöneticisi istemci uygulamalarından birinin istekte bulunan kuyruk yöneticisine yeniden yönlendirmesini ister. İstemci uygulaması yeniden yönlendirme isteğini alır, bağlantısını kapatır ve istekte bulunan kuyruk yöneticisine yeniden bağlanır. Bu otomatik dengeleme mekanizması uygulama için saydamdır. Daha fazla bilgi için bkz. "[Otomatik dengeleme nasıl çalışır?](#)" sayfa 404.

Birbirine bağlı uygulamalarla ilgili meta verileri periyodik olarak dağıtarak, tek tip küme, istemci uygulamalarının zaman içinde kuyruk yöneticilerine geniş dengeli bir oranına ulaşabilir. Hızlı ardışık yeniden yönlendirme olaylarının oluşmasını önlemek için otomatik dengeleme algoritması, yeniden yönlendirme isteklerinin yapıldığı hızı sınırlar.

Bir kümedeki kuyruk yöneticilerindeki uygulamaların yürürlükteki durumunu izleyebilir ve uygulama yönetim ortamlarını izleyebilirsiniz. Daha fazla bilgi için bkz. [İzleme uygulaması dengeleme. Uygulama dengeleme sorun gidermekonusunda](#) açıklandığı gibi uygulama dengeleme ile ilgili çeşitli sorunları da çözebilirsiniz.

Yeniden dengeleme yalnızca uzun bağlantı süresi olan uygulamalar için yararlıdır. Kısa bağlantı süresi olan istemci uygulamalarınız varsa (örneğin, farklı kuyruk yöneticilerine düzenli olarak bağlanmak ve bağlantı kesmek için yazılan istemci uygulamaları), bunları yeniden bağlanmaz olarak yapılandırmanız gerekir. Bu işlem, kuyruk yöneticilerinin dengelemeye çalıştığı uygulama kümesinden bunları kaldırır.

### İlgili kavramlar

"[Otomatik dengeleme otomatik yeniden bağlanma özelliğini nasıl kullanır?](#)" sayfa 406

IBM MQ 9.2.0' den, tek tip küme otomatik dengelemesi, var olan IBM MQotomatik yeniden bağlanma özelliğinde yapılan geliştirmelerden yararlanır.

### **Multi** Otomatik dengeleme nasıl çalışır?

Tek tip kümede, istemci bağlantıları uygulama adına dayalı olarak gruplanır. Aynı uygulama adını kullanarak tek tip kümenin herhangi bir üyesine bağlanan uygulamalar, aynı uygulama adını kullanan diğer uygulamalarla eşdeğer olarak kabul edilir.

Otomatik dengeleme, uygulama eşgörünümlerinin kümenin üyelerine eşit olarak dağıtılmasını sağlar; daha fazla bilgi için bkz. "[Uygulama adları ve uygulama örnekleri](#)" sayfa 402 . Bir kuyruk yöneticisine ya da tek tip bir kümeyle bağlı bir ya da daha çok uygulamanın ve uygulama eşgörünümlerinin durumunu görüntülemek için [DISPLAY APSTATUS](#) komutunu kullanın.

Örneğin, bir sigorta isteği uygulamasının tüm eşgörünümlerini "INSURANCE.REQUESTS". Bu uygulamadaki ilgili bağlantılar, uygun olduğu durumlarda eşgörünümler halinde otomatik olarak gruplandırılır ve tüm dengeleme eşgörünüm temelinde gerçekleştirilir.

Uygulamanın yeni eşgörünümleri tek tip kümenin bir üyesine bağlandığında, otomatik dengeleme algoritması hangi kuyruk yöneticilerinin en az INSURANCE.REQUESTS ve bu kuyruk yöneticilerine bazı bağlantıları yeniden yönlendirir.

Otomatik dengeleme yalnızca aşağıdaki durumlarda etkinleştirilir:

- Kanalın SHARECNV değeri sıfırdan büyük.
- Aşağıdakilerden biri doğru:
  - İstemci uygulaması MQCNO\_RECONNECT belirtiyor
  - mqclient.ini kütüğü **Defrecon=**YES

**Not:** Kuyruk yöneticisi benzerliğine sahip uygulamalar (örneğin, sürekli abonelik ya da dinamik bir kuyruk yanıtı nedeniyle) güvenli bir şekilde yeniden dengelenemez ve MQCNO\_RECONNECT\_QMGR ya da yeniden bağlanma seçeneği kullanılmamalıdır.

Bir istemci diğer bir kuyruk yöneticisine yeniden yönlendirildiğinde, yeni hedefe ilişkin bağlantı bilgilerini bulmak için her zamanki gibi yerel istemci kanal tanımlama çizelgelerini (CCDT) kullanır. Bu nedenle, otomatik dengeleme işleminin sorunsuz ve verimli çalışması için müşterilerin, ilk bağlantıları dengelemek



için kullanılan herhangi bir kuyruk yöneticisi grubunun yanı sıra tek tip kümenin her bir üyesi için bir giriş içeren bir CCDT kullanması önemlidir.

JSON biçimi CCDT kullanımı, aynı sunucu bağlantısı adını kullanan birden çok bağlantıya izin verdiği için bunu basitleştirir. Daha fazla bilgi için bkz. "[JSON biçimi CCDT 'nin yapılandırılması](#)" sayfa 44.

## İlgili kavramlar

"Otomatik dengeleme otomatik yeniden bağlanma özelliğini nasıl kullanır?" sayfa 406

IBM MQ 9.2.0' den, tek tip küme otomatik dengelemesi, var olan IBM MQ otomatik yeniden bağlanma özelliğinde yapılan geliştirmelerden yararlanıyor.

## ALW **JMS uygulamalarının otomatik olarak dengelenmesi**

Jakarta Messaging 3.0 ya da Java Message Service 2.0 uygulamaları otomatik olarak dengelendiğinde, JMS uygulamalarının oluşturmuş olduğu temel IBM MQ bağlantıları grupları birlikte taşınır.

**V 9.3.0** IBM MQ 9.3.0'lanağından, ActivationSpecs yapılandırılırken **dynamicallyBalanced** özelliği kullanılabilir. Bu özellik, bir MDB 'nin tek tip bir kümede uygulama dengeleme işleminin bir parçası olarak farklı bir kuyruk yöneticisinden ileti almasının istenip istenemeyeceğini belirtir. Daha fazla bilgi için bkz. [Gelen iletişim için kaynak bağdaştırıcısının yapılandırılması](#).

JMS bağlantılarının işlenmesi için tek tip kümeler, *uygulama eşgörünümü* kavramına sahiptir. JMS için bir *uygulama eşgörünümü*, JMS Connection ve ilişkili tüm JMS Oturumları olarak tanımlanır.

İstemci bağlantısında, JMS Connection 'a karşılık gelen benzersiz bir bağlantı etiketi ayrılır ve daha sonra, bu JMS Connection tarafından yaratılan JMS Oturumlarına karşılık gelen istemci bağlantılarına aynı etiket uygulanır.

Örneğin, bir çift istemci uygulaması tek bir etkin kuyruk yöneticisine (Kuyruk Yöneticisi 1) sahip tek bir kümeyle karşı JMS uygulamalarını çalıştırıyorsa:

- Client 1, "App1" uygulama adını ayarladığı bir bağlantı üreticisi yaratır ve bir JMS Connection ve üç JMS Oturumu yaratır. İstemci 1, Kuyruk Yöneticisi 1 'de her biri aynı bağlantı etiketini paylaşan dört istemci bağlantısı yaratır ve bu, tek bir "App1" eşgörünümü olarak değerlendirilir.
- İstemci 2, "App1" uygulama adını ayarladığı bir bağlantı üreticisi de yaratır ve bir JMS Connection ve iki JMS Oturumu yaratır. İstemci 2, her biri aynı bağlantı etiketini (İstemci 1 'e atanandan farklı) paylaşan üç istemci bağlantısı oluşturur ve bu, "App1" ögesinin tek, ayrı bir eşgörünümü olarak değerlendirilir.
- Bu nedenle, kuyruk yöneticisi iki "App1" eşgörünümü görür

Otomatik dengeleme gerçekleştirildiğinde, uygulama eşgörünümleri taşınır. Kuyruk yöneticisi bir uygulama yönetim ortamı (aynı bağlantı etiketini paylaşan istemci bağlantıları grubu) seçer ve yönetim ortamının farklı bir kuyruk yöneticisine taşınmasını ister. İstemci kodu isteği alır ve ilgili tüm bağlantılarını ( JMS Connection ve ilişkili JMS Oturumları ile ilgili) yeni kuyruk yöneticisine taşınmasını sağlar.

Örneğin, önceden anahatlarıyla belirtilen uygulama yönetim ortamları kümesini alın ve bir örnek kümede yeni bir kuyruk yöneticisinin (Kuyruk Yöneticisi 2) başladığını varsayın.

Kuyruk Yöneticisi 2 'de iş yok, ancak Kuyruk Yöneticisi 1 'de 2 "App1" eşgörünümü var; Kuyruk Yöneticisi 2, Kuyruk Yöneticisi 1 'in bir "App1" eşgörünümünü Kuyruk Yöneticisi 2 'ye aktarmasını istiyor.

Kuyruk Yöneticisi 1, taşınacak bir "App1" eşgörünümü seçer. Örneğin amaçları açısından, Müşteri 1 tarafından yaratılan eşgörünümü seçtiğini varsayın.

- Kuyruk Yöneticisi 1, "App1" eşgörünümünü QM2 taşımak için İstemci 1 'e bir istek gönderir.
- İstemci, Kuyruk Yöneticisi 1 ile var olan dört istemci bağlantısını kapatır ve Kuyruk Yöneticisi 2 ile dört yeni bağlantı yaratır.
- İşlenmekte olan kısa bir duraklama dışında, JMS Connection ve JMS Oturumları olağan durumda rahatsız edilmemelidir.

## Not:

Bir uygulama örneği taşındığı sırada belirli işlemler devam ediyorsa, bir uygulama JMS kural dışı durumu alabilir.

JMS kural dışı durumunun, hatanın nedenini saptamak için neden kodunun alınabileceği bağlantılı bir IBM MQ kural dışı durumu olur.

Beklenen neden kodları şunlardır:

#### **MQRC\_CALL\_INTERRUPTED**

Bu durum, örneğin, kalıcı (JMS ' de varsayılan) bir ileti bir eşitleme noktasının dışına konduğunda, ancak işlem yeniden bağlantı tarafından kesildiğinde ortaya çıkar.

#### **MQRC\_BACKED\_OUT**

Bu durum, örneğin, bir iletiyi bir eşitleme noktasına koyma girişimi yeniden bağlantı tarafından kesildiğinde ortaya çıkar.

#### **İlgili kavramlar**

[“Otomatik dengeleme nasıl çalışır?” sayfa 404](#)

Tek tip kümede, istemci bağlantıları uygulama adına dayalı olarak gruplanır. Aynı uygulama adını kullanarak tek tip kümenin herhangi bir üyesine bağlanan uygulamalar, aynı uygulama adını kullanan diğer uygulamalarla eşdeğer olarak kabul edilir.

[“Otomatik dengeleme otomatik yeniden bağlanma özelliğini nasıl kullanır?” sayfa 406](#)

IBM MQ 9.2.0' den, tek tip küme otomatik dengelemesi, var olan IBM MQotomatik yeniden bağlanma özelliğinde yapılan geliştirmelerden yararlanıyor.

#### **Multi Otomatik dengeleme otomatik yeniden bağlanma özelliğini nasıl kullanır?**

IBM MQ 9.2.0' den, tek tip küme otomatik dengelemesi, var olan IBM MQotomatik yeniden bağlanma özelliğinde yapılan geliştirmelerden yararlanıyor.

IBM MQ 9.2.0öncesi IBM MQ sürümlerinde, otomatik yeniden bağlanma özelliği, sağlanan bağlantı ayrıntılarına, genellikle bir bağlantı adı listesine ya da bir İstemci Kanal Tanımlama Çizelgesine (CCDT) dayalı olarak, bir kuyruk yöneticisinin ya da farklı bir kuyruk yöneticisine otomatik olarak yeniden bağlanır.

IBM MQ istemcisi, bazı durumlarda uygulamanın gerçekleştiğinin farkında olmadan yeniden bağlantıyı sessiz bir şekilde gerçekleştirir. Hangi kuyruk yöneticisine yeniden bağlanma kararı, bir bağlantı adı listesindeki bağlantı adları sırasına ya da CCDT ' deki iş yükü dengeleme yapılandırmasına tamamen bağlı.

IBM MQ 9.2.0 ' den istemcinin yeniden bağlanması gereken kuyruk yöneticisine ilişkin bir ipucu içeren bir istemciye yeniden bağlanma isteği gönderilebilir. Kuyruk yöneticisi hatası ya da **endmqm -r**komutunu yayınlayan denetimci gibi birçok yeniden bağlanma senaryosunda, ipucu bilgisinde bir kuyruk yöneticisi adı bulunmaz ve otomatik yeniden bağlanma davranışı şu anda olduğu gibi çalışır.

Ancak, tek tip bir küme yapılandırdıysanız, otomatik uygulama dengeleme, dengeli bir kümeyi elde etmek için düzenli aralıklarla istemcilere yeniden bağlanma istekleri gönderir. Bu durumlarda, tek tip küme, istemci bağlantılarının en az bağlantısı olan kuyruk yöneticilerine taşınmasını sağlamak için yeniden bağlanma ipucunda bir kuyruk yöneticisi adı belirtir.

Otomatik dengelemenin çalışması için aşağıdakiler önemlidir:

- IBM MQ uygulamaları, bağlantı bilgilerini almak için CCDT ' leri kullanır.
- CCDT ' ler birörnek kümedeki her kuyruk yöneticisi için bir giriş içerir

Böyle bir durum yoksa, kümenin tüm üyelerindeki uygulamaları otomatik olarak dengelemesi mümkün değildir.

Bir uygulama, IBM MQ 9.2.0' dan önceki bir IBM MQ istemcisi sürümünü kullanıyorsa ve otomatik istemci yeniden bağlantısını destekleyecek şekilde yapılandırıldıysa, tek tip küme tarafından yeniden bağlanma adımlarından geçmesi için bir istek gönderilebilir.

İstemciden belirli bir kuyruk yöneticisine yeniden bağlanması istenmez, bunun yerine diğer yeniden bağlanma olayları için geçerli olan yeniden bağlanma mantığı dizisinden geçer. İstemcilerin, kümenin her bir üyesi için eşit ağırlıklı girişler içeren CCDT ' leri kullanacak şekilde yapılandırıldığından emin olarak, IBM MQ 9.2.0 ' den önce istemci uygulamalarının tek biçimli kümede eşit ağırlıklı olarak dağıtılmasını sağlamak mümkündür.

Uygulamalar, ek yönetim ortamı gerektiren bir kuyruk yöneticisine bağlanmadan önce birkaç yeniden bağlanma girişiminde bulunabilir ve bu nedenle, uygulamaların küme genelinde eşit olarak dağıtılmasını sağlamanın daha az verimli bir yoludur. Otomatik dengeleme bu ortamlarda daha uzun sürebilir.

## IBM MQ istemcileri otomatik istemci yeniden bağlantısını desteklemiyor

Bir uygulama, IBM MQ istemcisinin otomatik istemci yeniden bağlantısını desteklemeyen bir sürümünü kullanıyorsa, uygulama MQI çağrısından bir hata dönüş kodu alabilir.

Uygulamanız hataları işlemek ve el ile yeniden bağlantı gerçekleştirmek üzere tasarlanmamışsa, bu uygulamalar için otomatik dengelemeyi devre dışı bırakmak gerekebilir.

**Not:** Yeniden bağlanabilir olarak tanımlanan herhangi bir uygulama için otomatik dengeleme etkinleştirilir; yani, uygulamanın etkin bağlantı seçeneklerinde MQCNO\_RECONNECT vardır.

### İlgili görevler

“Yeni bir birörnek küme oluşturma” sayfa 415  
Yeni bir birörnek kümeyi nasıl oluşturduğunuz.

## V 9.3.0 Multi **Tek tip kümelerde uygulama yeniden dengelemesini etkileme**

Otomatik uygulama dengeleme (tek tip kümelerin bir özelliği) ile, bir uygulama bağlantısının yaşam çevriminin herhangi bir noktasında alternatif bir kuyruk yöneticisine taşınması istenebilir.

## Giriş

IBM MQ 9.3.0' den dengeleme algoritması, uygulama akışındaki kesintiyi en aza indirmek için uygulamaların durumunu otomatik olarak dikkate almaya çalışır. Bu, uygulama tipi ya da bu uygulama tarafından gerçekleştirilen IBM MQ etkinliği kalıbı hakkında IBM MQ ek bilgi vererek belirli uygulamalara ya da uygulama eşgörünümlerine uyacak şekilde ayarlanabilir.

Genellikle, bir istemci uygulamasını geliştiren ya da devreye alan kişinin bu kalıbı anlaması ve kuyruk yöneticisine bu bilgileri sağlaması olasıdır (bkz. [Esnek ve ölçeklenebilir istemci uygulamalarını devreye alma](#)), ancak aynı zamanda bir yönetici tarafından da ayarlanabilir.

Kuyruk yöneticisi makul bir süre içinde uygulamaların eşit dağılımını sağlayamazsa, uygulama bağlantılarının IBM MQ akışında uygun bir zaman beklemeden diğer kuyruk yöneticilerine yeniden dengelenebileceğini unutmayın.

Bu, gereksinimleri karşılayacak şekilde ayarlanabilir. Uygulamaların eşit dağıtımını hızlı bir şekilde gerçekleştirmek daha önemliyse, bir uygulamayı yeniden dengelemek için uygun bir zaman bulmak üzere ürünü daha az bekleyecek şekilde yapılandırabilirsiniz. Alternatif olarak, uygulamalarda kesintiyi önlemek daha önemliyse, ürünü her zaman uygulamayı taşımak için uygun bir süre bekleyecek şekilde yapılandırmak mümkündür.

Daha fazla genel bakış bilgileri için [Esnek ve ölçeklenebilir istemci uygulamalarını devreye alma](#) başlıklı konuya bakın.

.NET uygulamaları için daha fazla bilgi için bkz. [“.NET uygulamasında uygulama yeniden dengelemeyi etkileme” sayfa 410](#).

XMS.NET uygulamaları, daha fazla bilgi için [ConnectionFactory](#) başlıklı konuya bakın.

V 9.3.4 JMS Uygulamaları için daha fazla bilgi için bkz. [“IBM MQ classes for JMS uygulamasında uygulama yeniden dengelemeyi etkileme” sayfa 411](#).

## Varsayılan uygulama dengeleme davranışı

Varsayılan olarak, bir uygulamanın kuyruk yöneticisiyle etkileşiminin hareket/iş birimi durumu tüm uygulamalar için dikkate alınır.

Yerel hareketler için, otomatik uygulama dengeleme, bir harekette yer alan uygulamalara yeniden dengeleme isteklerinin verilmesini önler. Bu, bir uygulamanın, yapılandırılmış yeniden dengeleme

zamanaşımını ya da gerçek bir kesintiye ulaşması, böyle bir dönüş koduna neden olabileceği için, bir geri dönüş kodu alma olasılığını ortadan kaldırmaz; ancak bu, uygulamaların genellikle bir işlemin ortasında yeniden bağlanmaları istenmediği anlamına gelir.

Yeni bir hareketi hemen önceki işlemi tamamladıktan sonra başlatan uygulamalar için, yeniden dengeleme tamamlanırken yeni işlemdeki ilk çağrı için bir gecikme olabilir. Bu, otomatik uygulama dengeleme özelliğinin, tek tip bir kümedeki kuyruk yöneticileri arasında eşit bir uygulama dağılımı elde etmesini sağlar.

Daha uzun süreli hareketler kullanan uygulamalarınız varsa, yeniden dengeleme zamanaşımı değerini artırmayı ya da bu kısıtlamayı tamamen devre dışı bırakmayı düşünebilirsiniz. See [“Dengeleme davranışının yapılandırılması” sayfa 408](#) for links on how to control this in the MQI and .NET, or 'Designing client applications for fault tolerance and scalability' for the code level equivalent.

## İstek-yanıt dengeleme

Uygulama tipi **Request-Reply** olarak belirtildiğinde, uygulama yönetim ortamının gerçekleştirdiği her PUT için bir GET yanıtı beklenir. Uygulama eşgörünümü birden çok iş parçacığı içeriyorsa ya da toplu işlerde isteklerle ve yanıtlarla ilgileniyorsa, herhangi bir zamanda birden çok istek ve yanıt kullanılabilir.

Gönderilen isteklerin sayısı alınan yanıt sayısına eşit oluncaya ya da zamanaşımının arka plan değeri aşılıncaya kadar uygulama taşınmaya uygun olarak kabul edilmez.

Bu kural dışı durum, bir istek iletisi için ileti süre bitimi yapılandırıldığında ortaya çıktı. Yanıtların istek iletisinin süre bitim aralığı içinde alınması gerektiği varsayılır ve tüm istek iletilerinin süresi dolduğunda, dengeleme algoritması taşınmaya uygun eşgörünüm dikkate alınmadan önce ek yanıtları beklemez.

Birden çok istek varsa, yalnızca gönderilen istek iletileri arasındaki en son süre bitimi dikkate alınır. Anlamlı süre bitimi değerleri kullanımdayken, beklenen istek/yanıt süre bitimi zaman aralığının kısaltılmasını önlemek için uygulamaya ilişkin **Timeout** dengeleme parametresini en az gönderilen ileti süre bitimi kadar yüksek bir değer olacak şekilde yapılandırmanız gerekir.

Önceki örüntü yalnızca, bekleyen isteklerin olmadığı dönemler olmasını bekleyen uygulamalar için uygundur. Örneğin, sürekli ileti gönderen ve alan karmaşık çok iş parçacıklı uygulamalar, bu örüntü altında yeniden dengelemeye hiçbir zaman uygun olmayabilir.

### Notlar:

- Belirli istekleri ve yanıtları ilintilendirmek için herhangi bir girişimde bulunulmaz; bu nedenle, uçuş iletilerinde yer alan bir toplu iş içinde daha önceki bir yanıtın süresi dolarsa, uygulama, dengelemeye hak kazanmadan önce en son isteğin süresinin dolmasını bekleyebilir.
- Özellikle, benzer nedenlerle sınırsız süre sonu ve süresi dolan iletiler birleştirildiğinde özen gereklidir.

Sınırlı bir süre bitimine sahip istek iletileri varsa ve yeni iletiler sınırsız süre bitimiyle gönderilirse, sınırsız süre bitimi, geçerli en son süre bitimini yerine getirmeye devam eden dengeleme algoritması tarafından *dikkate alınmaz*.

Aksi takdirde, daha önce süresi dolmuş olan yanıtlar, uygulamanın taşınmaya uygun olmasını önleyebilir. Buna karşılık, sınırsız süre bitimi zamanaşımı yanıtları varsa, ancak daha sonra süresi dolan istekler gönderilirse, bekleme süresi en uzun (sınırlı) süre bitimine indirgenir.

Genel olarak, bir geliştiricinin ya da yöneticinin doğru bir şekilde izlemesi ya da tanımlaması için yeniden dengeleme uygunluğu güçleştiği için, tek bir uygulama eşgörünümünün dengeli bir uygulamada hem süresi dolan hem de süresi dolmayan istek iletileri göndermesini önlemelisiniz.

- Yalnızca gönderen uygulama tarafından belirtilen süre sonu (örneğin, MQI 'da MQMD.**Expiry** değeri) yanıtların ne kadar süreyle bekleneceği saptanırken dikkate alınır. Bu değerle yapılan sonraki değişiklikler (örneğin, CAPEXPY kullanımı) bekleme süresini etkilemez.

## Dengeleme davranışının yapılandırılması

IBM MQ uygulamalarının yeniden dengelenmesini tam olarak etkilemek için, belirli istemci uygulama ortamları, kullanılmakta olan ileti sistemi örüntüsüyle ilgili bağlantı sırasında bilgi sağlayabilir.

Bu bilgiler, *Dengeleme Seçenekleri* olarak adlandırılan yeni bir yapıda sağlanır.

MQI için bkz. [“MQI kullanılarak dengeleme davranışının yapılandırılması” sayfa 409.](#)

Bu yapının .NET istemcisi eşdeğeri için bkz. [“.NET uygulamasında uygulama yeniden dengelemeyi etkileme” sayfa 410.](#)

**V 9.3.4** Bu seçenekleri ayarlamaya ilişkin JMS yaklaşımı için bkz. [“IBM MQ classes for JMS uygulamasında uygulama yeniden dengelemeyi etkileme” sayfa 411 .](#)

Diğer istemci ortamları şu anda bağlantı sırasında bu yapının sağlanmasını desteklemiyor.

**V 9.3.0** **Multi** *MQI kullanılarak dengeleme davranışının yapılandırılması*  
IBM MQ uygulamalarının yeniden dengelenmesini tam olarak etkilemek için, belirli istemci uygulama ortamları, kullanılmakta olan ileti sistemi örüntüsüyle ilgili bağlantı sırasında bilgi sağlayabilir.

MQI 'da dengeleme seçenekleri yapısı [MQBNO](#) olarak bilinir.

Programınızda *Dengeleme Seçenekleri* sağlanmazsa, destekleyici istemciler bu bilgileri istemci uygulamasıyla birlikte konuşturulan `client.ini` dosyasında [Application](#) stanza ya da [ApplicationDefaults](#) stanza ' da türetir.

**Not:** [Application](#) sürümü, bu seçeneklerin hangi uygulama için geçerli olduğunu tanımlamak üzere bir **Name** alanı içermesi dışında, bu bölmeler aynıdır.

İki biçimden biri sağlanırsa, belirtik olarak ayarlanmamışsa `none` olarak kabul edilen **BalanceOptions** dışında tüm alanların var olması gerekir.

Seçeneklerin sağlanmasına ilişkin tercih sırası:

1. CONNX üzerinde uygulama tarafından bir MQBNO yapısı sağlanır ve bütünüyle kullanılır
2. Ya da varsa, eşleşen [Application](#) kısmı yalnızca bir tane oluşturmak için kullanılır
3. Ya da varsa, [ApplicationDefaults](#) kısmı yalnızca bir tane oluşturmak için kullanılır
4. Ya da bu bağlantı için MQBNO akışı yok.

MQBNO yapısından ya da `client.ini` dosyasından üç temel bilgi parçası sağlayabilirsiniz:

1. **ApplicationType** ya da uygulama kalıbı.

Bu alan, IBM MQ bu uygulamanın katıldığı IBM MQ etkinliğinin genel örüntüsünü gösterir.

Üç tip uygulama desteklenir:

#### **Basit**

“Varsayılan uygulama dengeleme davranışı” [sayfa 407](#) içinde açıklanan varsayılanlardan başka belirli bir kural uygulanmamalıdır.

#### **İstek-Yanıt**

Her MQPUT çağrısından sonra, yanıt iletisi için eşleşen bir MQGET çağrısı beklenir. Daha fazla ayrıntı için bkz. [“İstek-yanıt dengeleme” sayfa 408 .](#)

#### **Yönetilen istemci**

Yeniden dengeleme istekleri her zaman hemen istemciye dağıtılır; bu istekler, uygun gördüğü bir noktada yeniden dengelenir; örneğin, JEE kaynak bağdaştırıcısı bu şekilde kaydolur.

2. Yeniden dengeleme, uygulama etkinliğini kesintiye uğratabilir. **Timeout**
3. Belirli **BalanceOptions**

## **Uygulamanızın ne zaman yeniden dengelenebileceğine ilişkin örnekler**

### **Örnek 1**

İletileri eşitleme noktası altına koyan ve bir MQCMIT çağrısı yayınlayarak ileti kümesini kesinleştiren bir uygulama yazdınız. MQCMIT çağrısı tamamlandığında, uygulama iletileri yeni bir eşitleme noktası altına yerleştirmeye başlar.

## Önerilen IBM MQ yapılandırması

Varsayılan seçenekler yeterli

### Sonuç

Yapılandırılan hareket sayısı karşılandıktan sonra, bir MQCMIT çağrısı başarılı olduktan (ya da başarısız olduktan) sonra uygulama eşgörünümü taşınır.

Varsayılan olarak, bir ileti kümesi 10 saniyeyi aşarsa, yeniden dengeleme istenirse bu işlem geriye işlenebilir. İşlemlerin düzenli olarak bu sınırı aşmasını bekliyorsanız ve buna izin verilmesini gerektiriyorsanız, **Timeout** ' yi uygun şekilde genişletebilirsiniz.

## Örnek 2

Bir iletiyi küme kuyruğu eşgörünümüne koyan bir uygulama yazdınız ve başka bir uygulama isteği işledikten sonra yerel bir geçici dinamik kuyruğa bir iletiyle yanıt verdi. İstek yerel kuyruktan yıkıcı bir şekilde okunduğunda, uygulama bir sonraki istek iletisini koyar.

## Önerilen IBM MQ yapılandırması

Tip ögesini MQBNO\_BALTYPE\_REQREP olarak ayarlayın

### Sonuç

Bir uygulama MQGET çağrısını tamamladığında uygulama eşgörünümü taşınır; bu noktada uygulama yönetim ortamı başka bir kuyruk yöneticisine taşınır. Sonraki MQPUT çağrıları yeni kuyruk yöneticisinde gerçekleştirilir.

MQBNO

ApplicationType



.NET uygulamasında uygulama yeniden dengelemeyi etkileme

IBM MQ 9.3.0' den, kuyruk yöneticisine bağlanmak için MQQueueManager sınıfını kullandığınızda uygulamadaki bir HASH çizelgesini kullanarak dengeleme seçeneği özelliklerini ayarlayabileceğiniz ek sabitler vardır.

.NET içinde uygulama dengelemeyi etkilemek için kullandığınız sabitler şunlardır:

### Uygulama tipi yeniden dengeleniyor

Dengeleme işleminin tipi; **MQC.BALANCING\_APPLICATION\_TYPE\_PROPERTY** değişmeziyle gösterilir

- MQBNO yapısının **ApplicationType** alanını ayarlamak için bu özelliği kullanmalısınız.

Tamsayı tipindeki değerleri ayarlamanız gerekir ve olası değerler şunlardır:

#### **MQC.BALANCING\_APPLICATION\_TYPE\_SIMPLE**

Basit dengeleme; "Tek tip kümelerde uygulama yeniden dengelemesini etkileme" sayfa 407 içinde açıklananlara ek olarak belirli kurallar uygulanmaz. Bu varsayılan değerdir.

#### **MQC.BALANCING\_APPLICATION\_TYPE\_REQUEST\_REPLY**

İstek-Yanıt dengeleme; her **MQPUT** aramasından sonra, bir yanıt iletisi için eşleşen bir **MQGET** çağrısı beklenir. Böyle bir ileti alınıncaya ya da **EXPIRY** istek iletisi aşıncaya kadar dengeleme geciktirildi.

Yeniden bağlantı uygulama tarafından etkinleştirildiyse ve bu özellik ayarlanmamışsa,

**MQC.BALANCING\_APPLICATION\_TYPE\_SIMPLE** kullanılır.

### Yeniden dengeleme seçenekleri

**MQC.BALANCING\_OPTIONS\_PROPERTY** değişmeziyle gösterilen, düzenleyen uygulama tarafından ayarlanan dengeleme seçenekleri

- MQBNO yapısının **BalanceOptions** alanını ayarlamak için bu özelliği kullanmalısınız.

Tamsayı tipindeki değerleri ayarlamanız gerekir ve olası değerler şunlardır:

#### **MQC.BALANCING\_OPTIONS\_NONE**

Seçenek belirlenmedi. Bu varsayılan değerdir

## **MQC.BALANCING\_OPTIONS\_IGNORE\_TRANSACTIONS**

Bu seçeneğin ayarlanması, bir hareketin ortasında bile uygulamaların yeniden dengelenmesini sağlar.

Yeniden bağlantı uygulama tarafından etkinleştirildiyse ve bu özellik ayarlanmamışsa, **MQC.BALANCING\_OPTIONS\_NONE** kullanılır.

### **Yeniden dengeleme zamanlaşımı**

Yeniden dengeleme uygulama etkinliğini kesintiye uğratabilir; **MQC.BALANCING\_TIMEOUT\_PROPERTY** değişmeziyle gösterilir

- MQBNO yapısının Zamanlaşımı alanını ayarlamak için bu özelliği kullanmalısınız.

Tamsayı tipindeki değerleri ayarlamamız gerekir ve olası değerler şunlardır:

#### **MQC.BALANCING\_TIMEOUT\_AS\_DEFAULT**

Ayarlanan varsayılan zamanlaşımı değeri. Bu varsayılan değerdir

#### **MQC.BALANCING\_TIMEOUT\_IMMEDIATE**

Anında zamanlaşımı oluştu


#### **MQC.BALANCING\_TIMEOUT\_NEVER**

Zamanlaşımı oluşmuyor

**Not:** Yalnızca tanımlı değerlerden bir değer ya da 0-999999999 saniye arasında bir değer belirtmeniz gerekir.

Esnek ve ölçeklenebilir istemci uygulamalarının devreye alınması

MQBNO

 *IBM MQ classes for JMS uygulamasında uygulama yeniden dengelemeyi etkileme*

IBM MQ 9.3.4' de, bir **ConnectionFactory** üzerinde dengeleme seçeneği özelliklerini ayarlayabileceğiniz ek sabitler vardır. Bu sabitler yalnızca **WMQ\_PROVIDER\_VERSION** , 7 olarak ayarlandığında geçerlidir. Tek tip kümedeki Request\_reply uygulamaları, yanıtların kaçırılması olasılığına izin vermelidir.

- “Kullanılabilir sabitler” sayfa 411.
- “REQUEST\_REPLY uygulamalarının dengelenmesinde kaybolan iletilere ilişkin potansiyel” sayfa 412.

### **Kullanılabilir sabitler**

IBM MQ classes for JMS içinde uygulama dengelemeyi etkilemek için kullandığınız sabitler şunlardır:

#### **Uygulama tipi yeniden dengeleniyor**

Dengeleme işleminin tipi; **WMQConstants.WMQ\_BALANCING\_APPLICATION\_TYPE** değişmeziyle gösterilir

- MQBNO yapısının **ApplicationType** alanını ayarlamak için bu özelliği kullanmalısınız.

Tamsayı tipinde değerler ayarlamalısınız. Olası değerler şunlardır:

#### **WMQConstants.WMQ\_BALANCING\_APPLICATION\_TYPE\_SIMPLE (varsayılan)**

Basit dengeleme; “Tek tip kümelerde uygulama yeniden dengelemesini etkileme” sayfa 407 içinde açıklananlara ek olarak belirli kurallar uygulanmaz.

#### **WMQConstants.WMQ\_BALANCING\_APPLICATION\_TYPE\_REQUEST\_REPLY**

İstek-Yanıt dengeleme; her **MQPUT** aramasından sonra, bir yanıt ileti için eşleşen bir **MQGET** çağrısı beklenir. Böyle bir ileti alınıncaya ya da **EXPIRY** istek ileti aşıncaya kadar dengeleme geciktirildi.

Yeniden bağlantı uygulama tarafından etkinleştirildiyse ve bu özellik ayarlanmamışsa, **WMQConstants.WMQ\_BALANCING\_APPLICATION\_TYPE\_SIMPLE** kullanılır.

### **Yeniden dengeleme seçenekleri**

**WMQConstants.WMQ\_BALANCING\_OPTIONS** değişmeziyle gösterilen, düzenleyen uygulama tarafından ayarlanan dengeleme seçenekleri



- MQBNO yapısının **BalanceOptions** alanını ayarlamak için bu özelliği kullanmalısınız.

Tamsayı tipinde değerler ayarlamalısınız. Olası değerler şunlardır:

**WMQConstants.WMQ\_BALANCING\_OPTIONS\_NONE (varsayılan)**

Seçenek belirlenmedi.

**WMQConstants.WMQ\_BALANCING\_OPTIONS\_IGNORE\_TRANSACTIONS**

Bu seçeneğin ayarlanması, bir hareketin ortasında bile uygulamaların yeniden dengelenmesini sağlar.

Yeniden bağlantı uygulama tarafından etkinleştirildiyse ve bu özellik ayarlanmamışsa,

**WMQConstants.WMQ\_BALANCING\_OPTIONS\_NONE** kullanılır.

### Yeniden dengeleme zaman aşımı

Yeniden dengeleme işlemi uygulama etkinliğini kesintiye uğratabilir;

**WMQConstants.WMQ\_BALANCING\_TIMEOUT** değişmeziyle gösterilir

- MQBNO yapısının **Timeout** alanını ayarlamak için bu özelliği kullanmalısınız.

Tamsayı tipinde değerler ayarlamalısınız. Olası değerler şunlardır:

**WMQConstants.WMQ\_BALANCING\_TIMEOUT\_AS\_DEFAULT (varsayılan)**

Ayarlanan varsayılan zaman aşımı değeri. Varsayılan olarak bu değer 10 saniyedir.

**WMQConstants.WMQ\_BALANCING\_TIMEOUT\_IMMEDIATE**

Anında zaman aşımı oluşur.

**WMQConstants.WMQ\_BALANCING\_TIMEOUT\_NEVER**

Zaman aşımı oluşmaz.

**1 ile 999999999 arasında bir değer**

Bu, saniye cinsinden bir değeri temsil eder.

**Not:** Yalnızca tanımlı değerlerden bir değer ya da 0-999999999 saniye arasında bir değer belirtmeniz gerekir.

Bu özellikler, JMSAdmin ya da IBM MQ Explorer arabirimlerini kullanan bağlantı üreticileri JNDI gösterimlerinde de ayarlanabilir.

## REQUEST\_REPLY uygulamalarının dengelenmesinde kaybolan iletilere ilişkin potansiyel

IBM MQ classes for JMS (ve IBM MQ classes for Jakarta Messaging) içinde, istek/yanıt işlevselliği, yanıtın gönderilip gönderilmeyeceğini belirlemek için yanıt veren uygulama tarafından kullanılan istek iletilerinde **JMSReplyTo** özelliği ayarlanarak uygulanır. JMS terimlerinde **JMSReplyTo** özelliği bir **Destination**.

Bu, IBM MQ işlemlerine çevrildiğinde, **JMSReplyTo** özelliği, belirli bir kuyruk yöneticisindeki bir kuyruğu tanıtan tam olarak nitelenmiş bir kuyruk URI 'si olarak gönderilir.

Yeniden dengeleme bağlantılarının işlenmesinin zamanuyumsuz niteliği nedeniyle, **JMSReplyTo** özelliği tam olarak nitelenmiş bir URI 'ye çevrildikten sonra, ancak istek iletileri istek kuyruğuna konmadan önce yeniden bağlantı başlatılabilir. Bu koşullarda, yanıt veren uygulama özgün kuyruk yöneticisindeki özgün yanıt kuyruğuna yanıt gönderebilir; ancak, istekte bulunan uygulama yeni kuyruk yöneticisinde yanıt bekliyor olabilir.

Bu nedenle, tek tip bir kümedeki Request\_reply uygulamaları, yanıtların kaçırılması olasılığına izin vermemelidir.

Esnek ve ölçeklenebilir istemci uygulamalarının devreye alınması

MQBNO-Dengeleme seçenekleri

### **Multi** **Tek biçimli kümeler için sınırlamalar ve dikkat edilecek noktalar**

Tek tip kümeleri yapılandırırken dikkate alınması gereken sınırlamalar ve diğer noktalar.

**Not:** Tek tip kümeleri yapılandırırken genel gereksinimler için bkz. “Yeni bir birörnek küme oluşturma” sayfa 415.



## Kuyruk yöneticileri arasındaki tekdüzenin önemi

Varsayılan olarak, kendisini reconnectable olarak bildiren herhangi bir uygulama, herhangi bir zamanda tek tip bir kümedeki diğer bir kuyruk yöneticisine yeniden dengelenebilir. Başka bir deyişle, bu tür uygulamaların gerektirdiği herhangi bir kaynak (örneğin, kuyruk, konu ya da yetki kaydı), bir örnek kümedeki tüm kuyruk yöneticilerine bildirilmelidir.

Kuyruk yöneticisi yapılandırmanın tutarlılığı yok. Benzer bir yapılandırmaya sahip olacak şekilde kümenin üyelerini yapılandırmak sistem yöneticiniz tarafından sağlanır.

Ancak, küme için nesnelere tanımlayan MQSC komut dosyalarını paylaşmak için [Başlangıçta bir MQSC komut dosyasından otomatik yapılandırma yeteneğini](#) kullanarak tutarlılığı destekleyebilir ve bu nedenle hepsinin aynı tanımlamalara sahip olduğundan emin olabilirsiniz. Daha fazla bilgi için bkz. [“Yeni bir birörnek küme oluşturma” sayfa 415.](#)

Bu tekdüzelik, küme için tam havuz kuyruğu yöneticilerini kapsayacak şekilde genişletir. Geleneksel IBM MQ kümeleri için genellikle en iyi uygulama olarak kabul edilse de, tek tip bir kümedeki tüm havuzları bağımsız sistemlere ayırmak için model, tam havuzların diğer düğümlerle birlikte küme ve süreç uygulaması iş yüklerine tam olarak katılmasıdır.

## Çakışan tek tip kümeler ve geleneksel IBM MQ kümeleri

Tek tip bir küme kuyruk yöneticisi en çok bir tek tip kümeye katılabilir ve herhangi bir sayıda standart IBM MQ kümesinin üyesi de olabilir. Tek tip kümenin daha geniş kümede tek bir kuyruk yöneticisi olarak hareket ettiğini düşünmek yararlı olabilir.

Tek tip bir küme kuyruk yöneticisi, yalnızca tek tip kümenin kendisi için tam havuz görevi görmelidir. Tek tip bir kümeye ait olan, ancak daha geniş bir geleneksel IBM MQ kümesine ait olabilecek herhangi bir kuyruk yöneticisi, tek tip kümenin dışında bir havuz olarak kullanılamaz. Daha fazla bilgi için [Tam havuzları tutmak için küme kuyruğu yöneticilerini seçme](#) başlıklı konuya bakın.

Tek bir tam havuz kuyruk yöneticisini tek bir kümeyle değiştirmek için, tam havuzu üzerinde devam eden uygulama çalışmasından ayırın ve yalnızca uygulama işini tek tip kümeye taşıyın.

Tek tip kümeler için otomatik tanımlamalar kullanıyorsanız, küme kanalları diğer kümelerde kullanılmak üzere paylaşamaz; başka bir şekilde, **CLUSTER** öznitelikli otomatik kümeye ayarlarsanız **CLUSNL** öznitelikli boş olur.

## Uygulama dengeleme ile ilgili önemli noktalar

Uygulama eşgörünümleri, özellikle aşağıdaki durumlarda, her zaman eşit dengelenmez:

- Kümedeki kuyruk yöneticilerinden daha az uygulama eşgörünümü olduğunda.
- İstemci uygulamaları kümeye bağlandıktan ya da kümeden ayrıldıktan kısa bir süre sonra.

İstemci uygulamalarının çok sık yeniden dengelenmesini önlemek için, özellikle uygulama bağlantıları gelip gidiyorken, tek tip kümenin istemci uygulamalarının ne sıklıkta yeniden dengelenmesini istediğinde sınırlar belirlenir. Yüksek bağlantı ya da bağlantı kesme etkinliğinden sonra, kalan uygulama eşgörünümlerinin tek tip kümede eşit olarak dengelenmesi birkaç dakika sürebilir.

Daha fazla bilgi için bkz. [Uygulama dengeleme sorun giderme.](#)

## Uygulama yakınlıkları

Tüm uygulamalar, tek tip bir kümede otomatik yeniden dengeleme için uygun değildir. Yalnızca **MQCNO\_RECONNECT** değerini belirten uygulamalar yeniden dengelenir. Belirli bir kuyruk yöneticisine benzeşimi olan uygulamaların **MQCNO\_NO\_RECONNECT** seçeneğini ya da **MQCNO\_RECONNECT\_Q\_MGR** seçeneğini belirtmesi gerekir. İkincisi, HA hata durumunda yedek sisteme geçmeye izin verir, ancak yeniden dengelemeye izin vermez.

Bir kuyruk yöneticisiyle örtülü benzerlik yaratan uygulama örnekleri:

- Sürekli abonelikler oluşturan uygulamalar.

- Örneğin, yanıt iletileri almak için kalıcı dinamik kuyruklar oluşturan uygulamalar.
- Sıkı ileti sıralaması beklenen ya da bir sıradaki tüm iletileri gerektiren uygulamalar aynı uygulama örneği ya da her ikisi tarafından işlenir.

Bu uygulamalar **MQCNO\_RECONNECT** yerine **MQCNO\_NO\_RECONNECT** ya da **MQCNO\_RECONNECT\_Q\_MGR** seçeneklerini belirtmelidir.

Daha fazla bilgi için [Yeniden yapılandırma seçenekleri](#) başlıklı konuya bakın.

## İleti kullanılabilirliği

Uygulama dengeleme, başarısız ya da geçici olarak kullanılamayan kuyruk yöneticilerine ilişkin bağlantıları yeniden dengeleyebilirken, tek biçimli kümeler, üyeleri arasında ileti verilerini eşlememektedir. Veri kullanılabilirliği için, bir düğüm arızalanırsa, tek tip kümenin her bir üyesi de yüksek kullanılabilirlikli olacak şekilde yapılandırılmalıdır. Birçok veri eşleme ve yüksek düzeyde kullanılabilirlik çözümü mevcuttur ve maksimum hizmet ve veri kullanılabilirliği için tek tip kümelerle birleştirilebilir, örneğin:

- Kapsayıcı düzenlemesi tarafından otomatik olarak yeniden başlatılan bir kapsayıcı örneğini destekleyen eşlenmiş depolama. Daha fazla bilgi için bkz. [Tek aksaklığa dayanıklı kuyruk yöneticisi](#).
- RDQM kuyruk yöneticileri. Daha fazla bilgi için bkz. [RDQM yüksek kullanılabilirlik](#).
- Çok eşgörünümlü kuyruk yöneticileri. Daha fazla bilgi için bkz. [Çok eşgörünümlü kuyruk yöneticileri](#).
- Yerli HA. Daha fazla bilgi için bkz. [Yerel HA](#).
- IBM MQ Appliance HA. Daha fazla bilgi için bkz. [Yüksek kullanılabilirlik](#).

## Tek tip kümelerin ölçeklenebilirliği ve performansı

Tek tip bir kümedeki kuyruk yöneticileri arasında uygulama durumunun daha yakın bütünleştirilmesini ve paylaşılmasını sağlamak için geleneksel IBM MQ kümesinden daha yüksek bir iletişim düzeyi gerekir. Bu nedenle, ek iletişimin performans üzerinde zararlı etkisi olduğundan, tek bir kümedeki çok sayıda kuyruk yöneticisinin ölçeklenmesi önerilmez.

Hem performans hem de yönetim nedenleriyle, tek tip bir kümenin, bir dizi ilgili uygulamaya ileti sistemi sağlayan, ancak bir kuruluş genelinde tek bir ileti sistemi hizmeti olmayan tek bir geleneksel kuyruk yöneticisi olarak hareket etmesi tercih edilir. Bu örnekte, 10 'a kadar küçük sayılar, kuyruk yöneticileri genellikle çok sayıda istemci uygulaması bağlantısını desteklemek için yeterlidir. Uygulama dengeleme, küçük sayılarla (örneğin, 3 kuyruk yöneticisi) başlamayı ve daha fazla kuyruk yöneticisi ekleyerek büyümeyi kolaylaştırır.



**Uyarı:** Önerilen özelliklere sahip olmayan bir kümede, özellikle de çok sayıda kuyruk yöneticisi içeren kümelerin kullanımının tek tip küme davranışının etkinleştirilmesi, önemli bir performans etkisine sahip olabilir.

### İlgili kavramlar

[“Otomatik uygulama dengeleme” sayfa 403](#)

Otomatik uygulama dengelemesi, IBM MQ tek tip kümesinin rasgele hale getirme ya da uygulamaların belirli kuyruk yöneticilerine el ile sabitlenmesi yerine küme genelinde uygulama dağıtımını yakından yönetmesini sağlayarak uygulama dağıtımını ve kullanılabilirliğini büyük ölçüde geliştirir.



## Tek tip küme yaratılması

Otomatik yapılandırma ve otomatik kümeleme desteğini kullanarak hem tek tip bir kümenin ilk oluşturulmasını basitleştirebilir, hem de daha sonra tek tip küme üyeleri arasındaki yapılandırmayı aynı tutabilir.

## Başlamadan önce

Tek tip bir küme oluşturmadan önce [“Tek biçimli kümeler için sınırlamalar ve dikkat edilecek noktalar” sayfa 412](#) başlıklı konuyu okumalısınız.

## Bu görev hakkında

Belirli bir IBM MQ kümesinin, `qm.ini` dosyasında en az **Type=Tek biçimli** ve **ClusterName=< tek biçimli küme adı >**olan AutoCluster için bir bölüm sağlayarak tek biçimli küme olarak kabul edileceğini belirtirsiniz.

İsteğe bağlı olarak, *otomatik küme yaratma* özelliğini kullanarak aynı `.ini` kısmı aracılığıyla temel IBM MQ kümesini yapılandırabilirsiniz. Kümeyi ayarlamak için bu otomatik küme desteğini kullanırken, kümeyi ve tam havuzlarını açıklayan bir yapılandırma dosyası sağlarsanız.

Başlatılmakta olan kuyruk yöneticisi tam havuzlardan biri olarak listeleniyorsa, otomatik olarak tam havuz yapılır. Benzer şekilde, küme alıcı kanalı tanımlandığında, küme gönderen kanalları tam havuza ya da havuzlara otomatik olarak tanımlanır.

## Yordam

Tek tip küme gerektiren ek işlemlerden yararlanmak için aşağıdaki adımlardan birini tamamlamanız gerekir:

- [Var olan bir kümeyi bir örnek kümeye dönüştürme](#), “Tek biçimli kümeler hakkında” sayfa 401’inde açıklanan kalıbı karşılar.
- [Bu amaçla yeni bir tek küme oluşturun](#).

### **ALW** Yeni bir bir örnek küme oluşturma

Yeni bir bir örnek kümeyi nasıl oluşturduğunuz.

## Yordam

1. Kümenin tam havuzlar açısından nasıl görünmesini istediğinizi açıklayan bir dosya oluşturun. Herhangi bir kümeye gelince, iki tam havuz, kümeyle ilgili merkezi bilgi depoları olarak işlev görür. Özellikle, bu kümedeki iki tam havuz için adları ve bağlantı adlarını tanımlamanız gerekir.

**Not:** Bu, yaratılmakta olan herhangi bir şeyden (kuyruk yöneticileri de içinde olmak üzere) önce gerçekleştirilir ve aşağıdaki işlem, bu kuyruk yöneticilerinin oluşturulmasını içerir.

Örneğin, kuyruk yöneticisi üyeleri QMA, QMB, QMC ve QMD ile UNICLUS adında tek tip bir küme oluşturduğunuzu varsayın. Bu örnekte, QMA ve QMB, kısmi havuzlar olarak QMC ve QMD ile tam havuzlar olacaktır. Örnek bir yapılandırma dosyası, `uniclus.ini`:

```
AutoCluster:
  Repository2Conname=QMA.dnsname(1414)
  Repository2Name=QMA
  Repository1Conname=QMB.dnsname(1414)
  Repository1Name=QMB
  ClusterName=UNICLUS
  Type=Uniform
```

**RepositoryNConname** alanları, küme göndericileri (CLUSDR) tanımlamak için diğer küme üyeleri için *conname* özniteliği olarak kullanılır ve çok eşgörünümlü bir kuyruk yöneticisi için bir bağlantı listesi olabilir ve isteğe bağlı olarak kapıyı içerebilir.

2. Tüm küme üyelerine uygulanmasını istediğiniz MQSC tanımlamalarını içeren örnek bir yapılandırma kütüğü (`uniclus.mqsc`) yaratın.

Bu dosyada, otomatik küme adının CLUSTER özniteliği (genellikle + AUTOCL + insert yoluyla) ve + QMNAME + insert değerini içeren bir kanal adı olan bir küme alıcı kanalı (CLUSRCVR) tanımlaması olan zorunlu bir satır vardır.

Bu, tek tip kümenin diğer üyelerinin her bir kuyruk yöneticisine nasıl bağlandığını ve diğer kuyruk yöneticilerine nasıl bağlanacağını şablonu olarak nasıl kullanıldığını açıklar. Örnek bir tanımlama şöyle olabilir:

```
define channel('+AUTOCL+_QMNAME+') chltype(clusrcvr) trptype(tcp)
conname(+CONNAME+) cluster('+AUTOCL+') replace
```

Otomatik kümeler yapılandırıldığında, bir küme alıcı kanalının tanımlaması, tanımın tek biçimli kümedeki tüm kuyruk yöneticileriyle aynı olmasını sağlamak için CLUSTER, CONNAME ve CHANNEL alanlarında bazı ek eklemeler kullanılabilir. Buna aşağıdakiler dahildir:

**+ AUTOCL +**

Otomatik küme adı

**+ QMNAME +**

Yaratılmakta olan kuyruk yöneticisinin adı

**+ CONNAME +**

Bağlantı adı parametre dizisinde kullanılmak üzere, kuyruk yöneticisi oluşturma sırasında, **-iv** parametresi kullanılarak ya da Variables qm.ini kısmı kullanılarak tanımlanan bir değişken. Değişkenin adı herhangi bir değer olabilir.

Kanal adlarının 20 karakterle sınırlı olduğunu unutmayın; bu nedenle, hem eklemeler hem de eklemeler değiştirildiğinde, değer bu sınırlamaya uyması gerekir. Örnek bir dosya aşağıdaki gibi görünebilir:

```
*#####
* Compulsory section for all uniform cluster queue managers
*#####
define channel('+AUTOCL+_QMNAME+') chltype(clusrcvr) trptype(tcp) conname(+CONNAME+)
cluster('+AUTOCL+') replace
*
*#####
* Configuration for all queue managers
*#####
define QL(APPQ) maxdepth(99999999) replace
define QL(APPQ2) maxdepth(99999999) replace
define channel(CLIENCHL) chltype(svrconn) trptype(tcp) replace
```

3. Bu iki kütüğü, tek tip bir küme üyesini barındıracak her makinede kullanılabilir kılın.

Örneğin, /shared/uniclus.ini ve /shared/uniclus.mqsc.

4. Bu makinelerin her birinde kuyruk yöneticisini yaratın.

Komut satırına şu komutu girin:

- a. Beklenen kapıda bir dinleyici başlatma isteği
- b. Otomatik küme kuruluşu dosyasını (uniclus.ini) gösteren otomatik INI yapılandırması isteği (**-ii**)
- c. Birörnek küme için CLUSRCVR tanımlamasını içeren MQSC yapılanış kütüğünü gösteren otomatik MQSC yapılanışı isteği (**-ic**).
- d. Bu kuyruk yöneticisine ilişkin bir CONNAME.

QMA için anasistemde:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMA.dnsname(1414) QMA
strmqm QMA
```

Tek biçimli kümedeki her kuyruk yöneticisi, hemen hemen aynı bir komut satırıyla yaratılır; tam ve kısmi havuz arasındaki tüm farklar, tek biçimli bir küme için otomatik olarak işlenir.

QMB için anasistemde:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMB.dnsname(1414) QMB
strmqm QMB
```

QMC anasisteminde:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMC.dnsname(1414) QMC
strmqm QMC
```

QMD anasisteminde:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMD.dnsname(1414) QMD
strmqm QMD
```

Otomatik olarak ne olur:

Kuyruk yöneticisi başlatıldıkça, `uniclus.ini` dosyasındaki tanımlar `qm.ini` dosyasına uygulanır. Daha fazla bilgi için bkz "[Başlangıçta qm.ini otomatik yapılandırması](#)" sayfa 97. Bu, **AutoCluster** tanımlamasını `qm.ini` dosyasına ekler.

Kuyruk yöneticisi **AutoCluster** içinde tam havuzlardan biri olarak adlandırılırsa, ALTER QMGR REPOS (**ClusterName**) MQSC komutunun verilmesine benzer şekilde otomatik olarak tam havuza dönüştürülür; tersi durumda, ALTER QMGR REPOS (!) MQSC komutunun verilmesine benzer şekilde kısmi bir havuza dönüştürülür.

Otomatik küme için küme alıcı kanalının tanımlanması işlendiğinde, küme gönderen kanalları bu kuyruk yöneticisinden **AutoCluster** bölümündeki tüm havuzlara tanımlanır (bu, tam havuzlardan biriyse yerel kuyruk yöneticisi dışında). Bu gönderen kanallar, tanımlanan yerel küme alıcısından tüm ortak kanal özniteliklerini devralır.



**Uyarı:** Kanallar el ile müdahaleye gerek kalmadan oluşturulsa da, bunlar başka bir kanal tanımı için görüntülenebilecek ve yönetilebilecek yönetim kanalı nesnelere aittir. Bu nesnelere, ileti trafiğini yönlendirmek için küme tarafından geçici ve isteğe bağlı olarak oluşturulan 'otomatik tanımlı' küme gönderen kanallarıyla karıştırmamalısınız.

## Sonraki adım

### Tek tip küme ayarının doğrulanması

**ClusterName** parametresi doğru ayarlandığında ve kuyruk yöneticisi adı belirtilen kümenin bir üyesi olduğunda, kümenin artık bir tek küme olarak tanımlandığını doğrulamak için AMQ9883 iletisi yayınlanır.

Daha sonra otomatik uygulama dengeleme gibi tek tip küme yeteneklerini kullanabilirsiniz. Kuyruk yöneticisi başlatılırken bu parametre ayarlandıysa, ancak ad geçerli bir IBM MQ küme adı değilse, ad yoksayılr ve AMQ9882 hata iletisi yayınlanır.

Ad geçerli bir küme adıysa, ancak belirtilen küme için küme kanalı yoksa, kuyruk yöneticisi hata günlüğüne AMQ9881 uyarı iletisi gönderilir. Bu ileti, denetimcinizin bu durumu saptamasına ve düzeltmesine olanak sağlar.

### Otomatikleştirilmiş küme ayarının doğrulanması

Tek tip kümeyi ayarlamak için otomatik küme desteğini kullandıysanız, tam havuz olarak belirtilen kuyruk yöneticilerinin `runmqsc` komutlarını kullanarak doğru şekilde yapılandırıldığını doğrulayabilirsiniz:

```
QMA:
  1 : dis qmgr repos
AMQ8408I: Display Queue Manager details.
      QMNAME(QMA)                REPOS(UNICLUS)
```

Kısmi havuzlar havuz olarak yapılandırılmamışken:

```
QMC:
  1 : dis qmgr repos
AMQ8408I: Display Queue Manager details.
      QMNAME(QMC)                REPOS( )
```

Ayrıca, yapılandırma MQSC dosyasındaki kanal adını kullanarak her kuyruk yöneticisinden diğer tam havuzlara küme gönderen kanallarının (CLUSSDR) yapılandırıldığını da görebilirsiniz:

```
QMA:
  1 : dis chl(UNICLUS*) conname
AMQ8414I: Display Channel details.
CHANNEL(UNICLUS_QMA)                CHLTYPE(CLUSRCVR)
CONNAME(QMA.dnsname(1414))
AMQ8414I: Display Channel details.
CHANNEL(UNICLUS_QMB)                CHLTYPE(CLUSSDR)
CONNAME(QMB.dnsname(1414))

QMC:
  1 : dis chl(UNICLUS*) conname
AMQ8414I: Display Channel details.
CHANNEL(UNICLUS_QMA)                CHLTYPE(CLUSSDR)
CONNAME(QMA.dnsname(1414))
AMQ8414I: Display Channel details.
CHANNEL(UNICLUS_QMB)                CHLTYPE(CLUSSDR)
CONNAME(QMB.dnsname(1414))
AMQ8414I: Display Channel details.
CHANNEL(UNICLUS_QMC)                CHLTYPE(CLUSRCVR)
CONNAME(QMC.dnsname(1414))
```

## İlgili kavramlar

[“Tek biçimli kümeler hakkında” sayfa 401](#)

Tek tip küme konuşlandırmasının amacı, uygulamaların ölçek ve kullanılabilirlik için tasarlanabilmesidir ve tek tip küme içindeki herhangi bir kuyruk yöneticisine bağlanabilir. Bu, belirli bir kuyruk yöneticisine olan bağımlılığı kaldırır ve ileti alışverişi trafiğinin daha iyi kullanılabilirliği ve iş yükü dengelemesi ile sonuçlanır.

**z/OS** Tek biçimli kümeler IBM MQ for z/OS üzerinde kullanılamaz; kuyruk paylaşım grupları, tek biçimli bir kümenin birçok yeteneğini sağlar.

[“Tek biçimli kümeler için sınırlamalar ve dikkat edilecek noktalar” sayfa 412](#)

Tek tip kümeleri yapılandırırken dikkate alınması gereken sınırlamalar ve diğer noktalar.

## **ALW** Var olan bir kümeyi bir örnek kümeye dönüştürme

Var olan bir kümeyi tek bir kümeye dönüştürmek için bu yordamı kullanabilirsiniz.

## Bu görev hakkında

Var olan bir kümeyi tek bir kümeye dönüştürürseniz, tüm küme üyelerinde kuyruk yöneticileri arasında dengeleyen uygulamaları desteklemek için gereken tanımlamanın var olduğundan emin olmanız gerekir.

## Yordam

1. Tüm kuyruk yöneticilerine uzak (kümelenmiş) yayınlama aboneliği de içinde olmak üzere IBM MQ yayınlama aboneliğini etkinleştirin.  
Bu, tek tip küme işlevselliği için bir önkoşuldur; bu nedenle, kuyruk yöneticisinin PSMODE ve PSCLUS özniteliklerinin her ikisinin de ENABLED varsayılan değerine ayarlandığından emin olmanız gerekir.
2. qm. ini kütüğüne, küme kanalları gibi MQSC nesne tanımlamalarınızda kullanıldığı şekilde, IBM MQ kümesinin adına bir **AutoCluster** kısmı ekleyin.  
Örneğin, kümenin adı UNICLUS ise, qm. ini dosyalarınıza aşağıdaki gibi AutoCluster (Otomatik Küme) kısmı ekleyin ya da değiştirin:

```
AutoCluster:
  ClusterName=UNICLUS
  Type=Uniform
```

3. Yeni ayarı uygulamak için kuyruk yöneticilerini yeniden başlatın.
4. Otomatik yapılandırmayı, tüm tek tip küme üyelerinin başlatma sırasında aynı yapılandırmaya sahip olmasını sağlamak için bir mekanizma olarak kullanmayı düşünün.  
Daha fazla ayrıntı için bkz. [Başlangıçta bir MQSC komut dosyasından otomatik yapılandırma](#) .

## İlgili kavramlar

“Tek biçimli kümeler hakkında” sayfa 401

Tek tip küme konuşlandırmasının amacı, uygulamaların ölçek ve kullanılabilirlik için tasarlanabilmesidir ve tek tip küme içindeki herhangi bir kuyruk yöneticisine bağlanabilir. Bu, belirli bir kuyruk yöneticisine olan bağımlılığı kaldırır ve ileti alışverişi trafiğinin daha iyi kullanılabilirliği ve iş yükü dengelemesi ile sonuçlanır.

**z/OS** Tek biçimli kümeler IBM MQ for z/OS üzerinde kullanılamaz; kuyruk paylaşım grupları, tek biçimli bir kümenin birçok yeteneğini sağlar.

“Tek biçimli kümeler için sınırlamalar ve dikkat edilecek noktalar” sayfa 412

Tek tip kümeleri yapılandırırken dikkate alınması gereken sınırlamalar ve diğer noktalar.

## Multi Otomatik küme yapılandırmasının kullanılması

IBM MQ yapılandırmasını, qm.ini yapılandırma bilgilerini değiştirerek otomatik yapılandırmayı etkinleştirecek şekilde yapılandırın.

**Not:** Tek biçimli kümeler için yalnızca AutoCluster (Otomatik Küme) kılımını kullanabilirsiniz.

## Yapılandırılacak stanzas

Aşağıdaki kısmı değiştirebilirsiniz:

### AutoConfig

qm.ini dosyasında tanımlanır. Kuyruk yöneticisi başlatıldığında, hangi otomatik yapılanış dosyalarının uygulanacağını belirler.

Tek biçimli kümeler kullanılırken aynı küme yapılandırmasını dağıtmak için bu mekanizmayı kullanmalısınız.

### AutoCluster

qm.ini dosyasında tanımlanır. Kuyruk yöneticisi kümenin otomatik bir kümenin üyesi olup olmadığını tanımlamaya başladığında kullanılır ve kümenin tam havuzlarını tanımlayabilir.

### Değişkenler

qm.ini dosyasında tanımlanır. Bazı kuyruk yöneticisi değişkenlerini içerir.

## AutoConfig kısmına ilişkin öznitelikler

AutoConfig kısmında aşağıdaki iki öznitelige izin verilir:

### MQSCConfig=< Yol >

Yol, her kuyruk yöneticisi başlangıcında, tüm dosyaların \*.mqsc kuyruk yöneticisine uygulandığı bir dizinin yolu ya da tam dosya yolu olabilir.

Daha fazla bilgi için bkz. [Başlangıçta MQSC komut dosyasından otomatik yapılandırma](#).

### IniConfig=< Yol >

Yol, her kuyruk yöneticisi başlangıcında, tüm \*.ini dosyalarının qm.ini dosyasına uygulandığı bir dizinin yolu ya da tam dosya yoludur.

Daha fazla bilgi için bkz. [“Başlangıçta qm.ini otomatik yapılandırması” sayfa 97](#).

Bu öznitelikler, tek biçimli kümelerin ayarlanmasının bir parçası olarak sık sık kullanılır. Daha fazla bilgi için bkz. [“Yeni bir birörnek küme oluşturma” sayfa 415](#).

Örnek kısmı:

```
AutoConfig:
MQSCConfig=C:\MQ_Configuration\uniclus.mqsc
IniConfig=C:\MQ_Configuration\uniclus.ini
```

## AutoCluster (Otomatik Küme) kısmına ilişkin öznitelikler

AutoCluster (Otomatik Küme) kısmı için aşağıdaki öznitelikler zorunludur:

### Tip = *Tek biçimli*

Otomatik kümenin tipini belirtir ve geçerli tek seçenek tek bir kümeyi gösteren *Tek biçimli* seçeneğidir.

### ClusterName=< *Dizgi*>

Otomatik küme adı olan kümenin adı.

Yukarıdaki özniteliklerin varlığı, tek tip kümeler için uygulama dengelemesini sağlar. Daha fazla ayrıntı için bkz. "[Otomatik uygulama dengeleme](#)" sayfa 403 .

Ayrıca, küme bu kısımda açıklandıysa, bir kümenin basitleştirilmiş kurulumu da gerçekleştirilebilir. Daha fazla bilgi için bkz. "[Yeni bir birörnek küme oluşturma](#)" sayfa 415. Bu seçenek kullanıldığında, iki kuyruk yöneticisine ad verebilir ve bu otomatik kümeye ilişkin tam havuzlar için bağlantı adlarını verebilirsiniz.

AutoCluster (Otomatik Küme) kısmı için aşağıdaki öznitelikler isteğe bağlıdır, ancak bunları çift olarak sağlamanız gerekir:

### HavuzAdı1 =< *Dizgi*>

Bu, otomatik kümedeki ilk tam havuzun kuyruk yöneticisi adıdır. Bu, bu kuyruk yöneticisinin adı ya da başka bir ad olabilir.

### Repository1Conname=< *Bağlantı Adı Dizesi* >

Bu, otomatik küme üyelerinin bu kuyruk yöneticisine nasıl bağlanmaları gerektiğine ilişkin bağlantı adı (CONNAME) değeridir.

Buna ek olarak, küme için ikinci bir tam havuz tanımlayabilirsiniz:

### Repository2Name=< *Dizgi*>

### Repository2Conname=< *Bağlantı adı dizgisi* >

Örnek kısmı:

```
AutoCluster:
  Repository2Conname=myFR1.hostname(1414)
  Repository2Name=QMFR1
  Repository1Conname= myFR2.hostname(1414)
  Repository1Name=QMFR2
  ClusterName=UNICLUS
  Type=Uniform
```

## Değişkenler kısmı için öznitelikler

Öznitelik alanında bir attribute=value çifti geçerlidir. Bunlar, bir kuyruk yöneticisi yarattığınızda **crtmqm** komutundaki **-iv** komut satırı seçeneği kullanılarak sağlanabilir.

CONNAME otomatik küme yapılandırması sırasında Değişkenler kısmında listelenen öznitelikleri ve bir küme alıcı kanalının kanal adı MQSC alanlarını kullanabilirsiniz.

## Bir kuyruk yöneticisinin tek tip bir kümeden askıya alınması

Tek tip bir kümenin olağan çalışması sırasında, yeniden bağlanabilir istemci uygulaması yönetim ortamları, kümedeki herhangi bir kuyruk yöneticisine herhangi bir zamanda otomatik olarak yeniden dengelenebilir. Uygulamaların belirli bir süre boyunca (örneğin, bakım işlemleri ya da sorun saptama sırasında) belirli bir kuyruk yöneticisine bağlanmasını önlemek istiyorsanız, SUSPEND QMGR komutunu kullanın.

**SUSPEND QMGR CLUSTER** (*birörnek küme adı*) komutunu verin

Bir IBM MQ kümesinden askıya almanın olağan etkilerine ek olarak, tek bir kümede SUSPEND komutu yeniden bağlanabilir uygulamaların bu kuyruk yöneticisine yeniden dengelenmesini de önler.

Kuyruk yöneticisiyle var olan bu tür bağlantılar, komut yayınlandığında kümedeki diğer kullanılabilir kuyruk yöneticileriyle hemen yeniden dengelenir.

### Notlar:

- Kuyruk yöneticileri bir kümeden askıya alındığında, **DIS APSTATUS** , kendi durum girdisi için her zaman ACTIVE (YES) gösteren yerel kuyruk yöneticisi dışında, bunları ACTIVE (NO) olarak gösterir.



- Tek tip kümedeki tüm kuyruk yöneticileri askıya alındıysa, uygulamalar askıya alınmış kuyruk yöneticilerinden birine ya da daha fazlasına bağlı kalır.

Kuyruk yöneticisine yeni bağlantıların eklenmesini önlemek için, istemci uygulamalarınız tarafından kullanılan sunucu bağlantısı kanallarını ya da kanalları (örneğin, aşağıdaki **runmqsc** komutunu vererek) durdurmalısınız:

```
STOP CHANNEL(surconn channel name)
```

Örneğin, bu kanallar bakım aralığı sırasında gerekli olan yönetim uygulamalarını bağlamak için de kullanılıyorsa, bu mümkün olmayabilir. Bu nedenle, askıya alınan kuyruk yöneticisi belirli aralıklarla bağlı yeniden bağlanabilir uygulamaları denetler

Yeniden bağlanabilir uygulamalar varsa, bunlar kümedeki diğer kullanılabilir kuyruk yöneticileriyle yeniden dengelenir. Artık askıya alınan kuyruk yöneticisinde bakım gerçekleştirilebilir.

**Not:** Taşınabilir olarak kabul edilmeyen uygulamalar, ilk komutlardan ya da sonraki yeniden taramalardan etkilenmez ve askıya alınan kuyruk yöneticisine bağlı kalır; daha fazla ayrıntı için bkz. [MOVCOUNT](#) .

Askıya alınmış bir kuyruk yöneticisini sürdürmek için:

1. Gerekliyse, aşağıdaki komutu vererek yeni uygulama bağlantılarını kabul etmeye devam etmek için sunucu bağlantı kanalını başlatın:

```
START CHANNEL(surconn channel name)
```

2. Aşağıdaki **runmqsc** komutunu verin:

```
RESUME QMGR CLUSTER(uniform cluster name)
```

Kuyruk yöneticisi, tek tip kümenin geri kalanıyla iletişimi sürdürür ve dengeyi geri yüklemek için gerekirse, yeniden bağlanabilir istemci uygulaması yönetim ortamları bu kuyruk yöneticisine yeniden yönlendirilir.

## Yayınlama/abone olma ileti sisteminin yapılandırılması

Kuyruğa alınan yayınlama/abone olma durumunu başlatabilir, durdurabilir ve görüntüleyebilirsiniz. Ayrıca, akışları ekleyebilir ve kaldırabilir ve bir aracı sıradüzenine kuyruk yöneticileri ekleyebilir ve silebilirsiniz.

### Yordam

- Kuyruğa alınan yayınlama/abone olma denetimi hakkında daha fazla bilgi için aşağıdaki alt konulara bakın:
  - [“Kuyruğa alınan yayınlama/abone olma iletisi özniteliklerini ayarlama” sayfa 421](#)
  - [“Kuyruğa alınan yayınlama/abone olma başlatılıyor” sayfa 423](#)
  - [“Kuyruğa alınan yayınlama/abone olma durduruluyor” sayfa 423](#)
  - [“Akış eklenmesi” sayfa 424](#)
  - [“Akış silinmesi” sayfa 425](#)
  - [“Abonelik noktası eklenmesi” sayfa 425](#)
  - [“Yayınlama/abone olma ağlarında konu alanlarını birleştirme” sayfa 433](#)

## Kuyruğa alınan yayınlama/abone olma iletisi özniteliklerini ayarlama

Kuyruk yöneticisi özniteliklerini kullanarak bazı yayınlama/abone olma iletisi özniteliklerinin davranışını denetleyebilirsiniz. qm . in i dosyasının *Aracı* kıtasında denetleyeceğiniz diğer öznitelikler.

### Bu görev hakkında

Şu yayınlama/abone olma özniteliklerini ayarlayabilirsiniz: Ayrıntılar için bkz. [Kuyruk yöneticisi parametreleri](#)

Çizelge 27. Yapılanış değiştirgelerini yayınla/abone ol

Açıklama	MQSC değiştirgesi adı
Komut iletisi yeniden deneme sayısı	<b>PSRTYCNT</b>
Teslim edilemeyen komut giriş iletisini at	<b>PSNPMSG</b>
Teslim edilemeyen komut yanıt iletisini izleyen davranış	<b>PSNPRES</b>
Komut iletilerini eşitleme noktası altında işle	<b>PSSYNCPT</b>

Aracı kısmı, aşağıdaki yapılanış ayarlarını yönetmek için kullanılır:

- `PersistentPublishRetry=yes | force`

Evetseçeneğini belirlerseniz, kuyruğa alınan yayınlama/abone olma arabirimi aracılığıyla kalıcı bir iletinin yayınlanması başarısız olursa ve olumsuz yanıt istenmezse, yayınlama işlemi yeniden denenir.

Olumsuz bir yanıt iletisi istediyseniz, olumsuz yanıt gönderilir ve başka bir yeniden deneme gerçekleşmez.

`Zorlaseçeneğini` belirlerseniz, kuyruğa alınan yayınlama/abone olma arabirimi aracılığıyla kalıcı bir iletinin yayınlanması başarısız olursa, yayınlama işlemi başarıyla işleninceye kadar yeniden denenir. Negatif yanıt gönderilmez.

- `NonPersistentPublishRetry= evet | force`

Evetseçeneğini belirlerseniz, kuyruğa alınan yayınlama/abone olma arabirimi aracılığıyla kalıcı olmayan bir iletinin yayınlanması başarısız olursa ve olumsuz yanıt istenmezse, yayınlama işlemi yeniden denenir.

Olumsuz bir yanıt iletisi istediyseniz, olumsuz yanıt gönderilir ve başka bir yeniden deneme gerçekleşmez.

`Zorlaseçeneğini` belirlediyseniz, kuyruğa alınan yayınlama/abone olma arabirimi aracılığıyla kalıcı olmayan bir iletinin yayınlanması başarısız olursa, yayınlama işlemi başarıyla işleninceye kadar yeniden denenir. Negatif yanıt gönderilmez.

**Not:** Kalıcı olmayan ileteler için bu işlevselliği etkinleştirmek istiyorsanız, `NonPersistentPublishRetry` değerini ayarlamamanın yanı sıra, **PSSYNCPT** kuyruk yöneticisi özneteliğinin Evetolarak ayarlandığından da emin olmanız gerekir.

Bunun yapılması, `STREAM` kuyruğundaki **MQGET** artık syncpoint altında gerçekleştiği için, kalıcı olmayan yayınları işleme performansı üzerinde de etkili olabilir.

- `PublishBatchSize =sayı`

Aracı olağan durumda, yayınlama iletelerini eşitleme noktası içinde işler. Her yayını tek tek kesinleştirmek verimsiz olabilir ve bazı durumlarda aracı birden çok yayınlama iletisini tek bir iş biriminde işleyebilir. Bu değiştirge, tek bir iş biriminde işlenebilecek yayınlama iletisi sayısı üst sınırını belirtir.

`PublishBatchSize` için varsayılan değer 5 'tir.

- `PublishBatchAralık =sayı`

Aracı olağan durumda, yayınlama iletelerini eşitleme noktası içinde işler. Her yayını tek tek kesinleştirmek verimsiz olabilir ve bazı durumlarda aracı birden çok yayınlama iletisini tek bir iş biriminde işleyebilir. Bu parametre, bir toplu işteki ilk iletiyle aynı toplu işte bulunan sonraki yayınlar arasındaki süre üst sınırını (milisaniye cinsinden) belirtir.

0 toplu iş aralığı, iletelerin hemen kullanılabilmesi koşuluyla, en çok `PublishBatchSize` iletilerinin işlenebildiğini gösterir.

`PublishBatchInterval` için varsayılan değer sıfırdır.

## Yordam

Yayınlama/abone olma davranışını denetleyen kuyruk yöneticisi özniteliklerini değiştirmek için IBM MQ Explorer, programlanabilir komutlar ya da **runmqsc** komutunu kullanın.

## Örnek

```
QMGR PSNPRES DEĞİŞTİR (GÜVENLİ)
```

## Kuyruğa alınan yayınlama/abone olma başlatılıyor

Kuyruk yöneticisinin PSMODE özniteliğini ayarlayarak, kuyruğa alınmış yayınlama/abone olma işlemini başlatabilirsiniz.

### Başlamadan önce

Üç yayınlama/abone olma kipini anlamak için [PSMODE](#) açıklamasını okuyun:

- DERLEME
- DEVRE DIŞI
- Etkinleştirildi

### Bu görev hakkında

QMGR PSMODE özniteliğini, kuyruğa alınan yayınlama/abone olma arabirimini (aracı olarak da bilinir) ya da yayınlama/abone olma altyapısını (Sürüm 7 yayınlama/abone olma olarak da bilinir) ya da her ikisini birden başlatacaktır. Kuyruğa alınmış yayınlama/abone olma işlemini başlatmak için PSMODE değerini ENABLED olarak ayarlamanız gerekir. Varsayılan değer ETKİNLEŞTİRİLMİŞ'tir.

## Yordam

Arabirim önceden etkinleştirilmemişse, kuyruğa alınan yayınlama/abone olma arabirimini etkinleştirmek için IBM MQ Explorer ya da **runmqsc** komutunu kullanın.

## Örnek

```
ALTER QMGR PSMODE (ENABLED)
```

### Sonraki adım

IBM MQ , kuyruğa alınan yayınlama/abone olma komutlarını ve yayınlama/abone olma İleti Kuyruğu Arabirimi (MQI) çağrılarını işler.

## Kuyruğa alınan yayınlama/abone olma durduruluyor

Kuyruk yöneticisinin PSMODE özniteliğini ayarlayarak, kuyruğa alınan yayınlama/abone olma işlemini durdurursanız.

### Başlamadan önce

Üç yayınlama/abone olma kipini anlamak için [PSMODE](#) açıklamasını okuyun:

- ŞİRKET
- DEVRE DIŞI
- Etkinleştirildi

### Bu görev hakkında

QMGR PSMODE özniteliğini, kuyruğa alınan yayınlama/abone olma arabirimini (aracı olarak da bilinir) ya da yayınlama/abone olma altyapısını (Sürüm 7 yayınlama/abone olma olarak da bilinir) ya da her

ikisini birden durduracak şekilde ayarlayın. Kuyruğa alınan yayınlama/abone olma işlemini durdurmak için PSMODE ögesini COMPATolarak ayarlamanız gerekir. Yayınlama/abone olma motorunu tamamen durdurmak için PSMODE seçeneğini DISABLEDolarak ayarlayın.

## Yordam

Kuyruğa alınan yayınlama/abone olma arabirimini geçersiz kılmak için IBM MQ Explorer ya da **runmqsc** komutunu kullanın.

## Örnek

```
ALTER QMGR PSMODE (COMPAT)
```

## Akış eklenmesi

Uygulamalar arasında veri yalıtımına izin vermek ya da IBM MQ yayınlama/abone olma sıradüzenleriyle işlemler arası işlemlere izin vermek için akışları el ile ekleyebilirsiniz.

## Başlamadan önce

Yayınlama/abone olma akışlarını çalıştırma şeklini tanıyın. Bkz. [Akışlar ve konular](#).

## Bu görev hakkında

Bu adımları gerçekleştirmek için **runmqsc**ya da IBM MQ Explorer PCF komutunu kullanın.

**Not:** Adım 1 ve 2 'yi istediğiniz sırada gerçekleştirebilirsiniz. Yalnızca adım 1 ve 2 tamamlandıktan sonra adım 3 'ü gerçekleştirin.

## Yordam

1. Önceki IBM MQsürümünde akışla aynı ada sahip bir yerel kuyruk tanımlayın.
2. IBM MQ' un önceki sürümündeki akışla aynı ada sahip yerel bir konu tanımlayın.
3. Kuyruğun adını ad listesine ekle, SYSTEM . QPUBSUB . QUEUE . NAMELIST
4. Yayınlama/abone olma sıradüzenindeki IBM MQ ' un sonraki sürümündeki tüm kuyruk yöneticileri için bu işlemi yineleyin.

## ekleme 'Sport'

Akış paylaşma örneğinde 'Sport' , önceki sürüm kuyruğu yöneticileri ve sonraki sürüm IBM MQ kuyruk yöneticileri aynı yayınlama/abone olma sıradüzeninde çalışıyor. Önceki sürüm kuyruğu yöneticileri 'Sport' adlı bir akışı paylaşıyor. Bu örnekte, önceki sürüm kuyruğu yöneticileri akışıyla paylaşılan 'Sport' konu dizisiyle 'Sport' 'Sport' adlı daha sonraki sürüm kuyruğu yöneticilerinde bir kuyruk ve konu nasıl yaratılacağı gösterilmektedir.

Daha sonraki bir sürüm kuyruk yöneticisi yayınlama uygulaması, konu dizesi 'Soccer/Results' ile 'Sport' konusunda yayınlama, sonuçta elde edilen konu dizisini yaratır 'Sport/Soccer/Results' . Daha sonraki sürüm kuyruğu yöneticileriyle ilgili olarak, 'Sport' konusuna abone olan 'Soccer/Results' konu dizesi yayını alır.

Daha önceki sürüm kuyruk yöneticileriyle ilgili olarak, 'Sport' akımını yayınlamak için 'Soccer/Results' konu dizisini içeren aboneler yayını alır.

```
runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
define qlocal('Sport')
  1 : define qlocal('Sport')
AMQ8006: IBM MQ queue created.
define topic('Sport') topicstr('Sport')
  2 : define topic('Sport') topicstr('Sport')
AMQ8690: IBM MQ topic created.
alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
```

```
'SYSTEM.BROKER.ADMIN.STREAM')
3 : alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
'SYSTEM.BROKER.ADMIN.STREAM')
AMQ8551: IBM MQ namelist changed.
```

**Not: alter namelist** komutuna eklediğiniz yeni adların yanı sıra namelist nesnesinde var olan adları da sağlamanız gerekir.

## Sonraki adım

Akışla ilgili bilgiler, sıradüzendeki diğer araçlara iletilir.

Sıradüzendeki her IBM MQ kuyruk yöneticisini el ile yapılandırmanız gerekir.

## Akış silinmesi

Bir akışı IBM MQ kuyruk yöneticisinden silebilirsiniz.

### Başlamadan önce

Bir akışı silmeden önce akışa kalan abonelik olmadığından emin olmanız ve akışı kullanan tüm uygulamaları susturmanız gerekir. Yayınlar silinen bir akışa akmaya devam ederse, sistemi temiz bir çalışma durumuna geri döndürmek için çok fazla yönetim çalışması gerekir.

### Yordam

1. Bu akışı barındıran tüm bağlı araçları bulun.
2. Tüm araçlarda akışa olan tüm abonelikleri iptal edin.
3. Kuyruğu (akımla aynı ada sahip) ad en alt sırasından ( SYSTEM.QPUBSUB.QUEUE.NAMELIST) kaldırın.
4. Akışla aynı ada sahip tüm iletileri kuyruktan silin ya da silin.
5. Akımla aynı ada sahip kuyruğu silin.
6. İlişkili konu nesnesini silin.

### Sonraki adım

Akışı barındıran diğer IBM MQ kuyruk yöneticilerinin tümünde 3-5 arasındaki adımları yineleyin.

## Abonelik noktası eklenmesi

IBM Integration Bus ' un önceki bir sürümünden yeni bir abonelik noktasıyla geçirdiğiniz, kuyruğa alınmış bir yayınlama/abone olma uygulamasını genişletme.

### Başlamadan önce

1. Abonelik noktasının SYSTEM.QPUBSUB.SUBPOINT.NAMELIST içinde önceden tanımlı olup olmadığını denetleyin.
2. Abonelik noktasıyla aynı ada sahip bir konu nesnesi ya da konu dizgisi olup olmadığını denetleyin.

### Bu görev hakkında

IBM MQ, uygulamalar abonelik noktalarını kullanmaz, ancak abonelik noktası geçiş mekanizmasını kullanarak var olan uygulamalarla birlikte çalışabilirler.

**Önemli:** Abonelik noktası geçiş mekanizması IBM MQ 8.0içinden kaldırıldı. Var olan uygulamalarınızı yeni düzeye geçirmeniz gerekiyorsa, en son sürüme geçmeden önce, ürün sürümünüze ilişkin belgelerde açıklanan yordamları gerçekleştirmeniz gerekir.

IBM MQsürümleri için yazılan tümleşik yayınlama/abone olma uygulamalarını kullanmak için abonelik noktaları eklemeye gerek yoktur.

## Yordam

1. Abonelik noktasının adını SYSTEM.QPUBSUB.SUBPOINT.NAMELIST' e ekleyin.
  - z/OSsistemlerinde **NLTYPE** , varsayılan değer olan NONE' dır.
  - Adımı, aynı yayınlama/abone olma topolojisine bağlı her kuyruk yöneticisinde yineleyin.
2. Abonelik noktasının adıyla eşleşen bir konu dizgisiyle, tercihen abonelik noktasının adını veren bir konu nesnesi ekleyin.
  - Abonelik noktası bir kümedeyse, konu nesnesini küme konusu anasisteminde küme konusu olarak ekleyin.
  - Abonelik noktasının adıyla aynı konu dizisine sahip bir konu nesnesi varsa, var olan konu nesnesini kullanın. Var olan bir konuyu yeniden kullanarak abonelik noktasının sonuçlarını anlamanız gerekir. Var olan konu var olan bir uygulamanın parçasıysa, aynı adı taşıyan iki konu arasındaki çakışmaları çözmeniz gerekir.
  - Abonelik noktasıyla aynı ada sahip, ancak farklı bir konu dizisine sahip bir konu nesnesi varsa, farklı bir ada sahip bir konu oluşturun.
3. **Topic** özniteliği WILDCARD değerini BLOCKolarak ayarlayın.
  - # ya da \* aboneliklerinin engellenmesi, abonelik noktalarına genel arama karakteri aboneliklerini yalıtır, bkz. [Genel arama karakterleri ve abonelik noktaları](#).
4. Konu nesnesinde gerekli olan öznitelikleri ayarlayın.

## Örnek

Örnekte, USD ve GBP olmak üzere iki abonelik noktası ekleyen bir **runmqsc** komut dosyası gösterilmektedir.

```
DEFINE TOPIC(USD) TOPICSTR(USD)
DEFINE TOPIC(GBP) TOPICSTR(GBP) WILDCARD(BLOCK)
ALTER NL(SYSTEM.QPUBSUB.SUBPOINT.NAMELIST) NAMES(SYSTEM.BROKER.DEFAULT.SUBPOINT, USD, GBP)
```

## Not:

1. **ALTER** komutu kullanılarak eklenen abonelik noktaları listesine varsayılan abonelik noktasını ekleyin. **ALTER** , ad alanında var olan adları siler.
2. Ad değerini değiştirmeden önce konuları tanımlayın. Kuyruk yöneticisi yalnızca, kuyruk yöneticisi başladığında ve ad listesi değiştirildiğinde ad listesi durumunu denetler.

## Dağıtılmış yayınlama/abone olma ağlarının yapılandırılması

Dağıtılmış bir yayınlama/abone olma topolojisine bağlanan kuyruk yöneticileri ortak bir birleşik konu alanını paylaşır. Bir kuyruk yöneticisinde yaratılan abonelikler, topolojideki başka bir kuyruk yöneticisine bağlı bir uygulama tarafından yayınlanan iletileri alabilir.

Kuyruk yöneticilerini kümelerde ya da sıradüzenlerde birbirine bağlayarak oluşturulan konu alanlarının kapsamını denetleyebilirsiniz. Bir yayınlama/abone olma kümesinde, konu alanının kümeye yayılacak her dalı için bir konu nesnesi 'kümelenmiş' olmalıdır. Bir sıradüzeninde, her kuyruk yöneticisi sıradüzeninde 'üst ögesini' tanımlayacak şekilde yapılandırılmalıdır.

Her bir yayın ve aboneliğin yerel mi, yoksa genel mi olduğunu seçerek topoloji içindeki yayınların ve aboneliklerin akışını daha fazla denetleyebilirsiniz. Yerel yayınlar ve abonelikler, yayınlayıcının ya da abonenin bağlı olduğu kuyruk yöneticisinin ötesine yayılmaz.

## İlgili kavramlar

[Dağıtılmış yayınlama/abone olma ağları](#)

[Yayın kapsamı](#)

[Abonelik kapsamı](#)

[Konu alanları](#)

## İlgili görevler

[Küme konularının tanımlanması](#)

## Yayınlama/abone olma kümesinin yapılandırılması

Kuyruk yöneticisinde bir konu tanımlayın. Konuyu bir küme konusu yapmak için **CLUSTER** özelliğini ayarlayın. Bu konuya ilişkin yayınlar ve abonelikler için kullanılacak yönlendirmeyi seçmek üzere **CLROUTE** özelliğini ayarlayın.

### Başlamadan önce

Bazı küme yapılandırılmaları, doğrudan yönlendirilmiş yayınlama/abone olma ek giderlerini karşılayamaz. Bu yapılandırmayı kullanmadan önce, [Yayınlama/abone olma kümeleri tasarlamak](#) konusunda ayrıntılı olarak açıklanan önemli noktaları ve seçenekleri keşfedin.

Bir kümede yapılan değişikliklerin kümeye yayılabilmesi için en az bir tam havuz her zaman kullanılabilir olmalıdır. Bu görevi başlatmadan önce havuzlarınızın kullanılabilir olduğundan emin olun.

Ayrıca bkz. [Yayınlama/abone olma kümeleri için yönlendirme: Davranışta notlar](#).

Senaryo:

- **INVENTORY** kümesi, “Kümeye kuyruk yöneticisi eklenmesi” sayfa 308’inde açıklandığı gibi ayarlandı. Üç kuyruk yöneticisi içerir; LONDON ve NEWYORK her ikisi de tam havuzu tutar, PARIS kısmi bir havuzu tutar.

### Bu görev hakkında

Bir kümedeki kuyruk yöneticisinde bir konu tanımladığınızda, konunun bir küme konusu olup olmadığını ve (öyleyse) bu konuya ilişkin yayınlara ve aboneliklere ilişkin küme içindeki yönlendirmeyi belirtmeniz gerekir. Konuyu bir küme konusu yapmak için, küme adıyla **TOPIC** nesnesinde **CLUSTER** özelliğini yapılandırabilirsiniz. Kümedeki bir kuyruk yöneticisinde bir küme konusu tanımlayarak, konuyu tüm kümenin kullanımına sunuyorsunuz. Küme içinde kullanılacak ileti yöneltmesini seçmek için, **TOPIC** nesnesinde **CLROUTE** özelliğini aşağıdaki değerlerden birine ayarlayın:

- **DIRECT**
- **TOPICHOST**

Varsayılan olarak, konu yöneltme **DIRECT**’dir. Bu, IBM MQ 8.0’ den önceki tek seçenektir. Bir kuyruk yöneticisinde doğrudan yönlendirilmiş kümelendirilmiş bir konu yapılandırıldığında, kümedeki tüm kuyruk yöneticileri kümedeki diğer tüm kuyruk yöneticilerini bilir. Yayınlama ve abone olma işlemleri gerçekleştirirken, her kuyruk yöneticisi kümedeki başka bir kuyruk yöneticisine doğrudan bağlanabilir. Bkz. [Doğrudan yönlendirilmiş yayınlama/abone olma kümeleri](#).

IBM MQ 8.0’ dan bunun yerine konu yönlendirmesini **TOPICHOST** olarak yapılandırabilirsiniz. Konu anasistemi yöneltme özelliğini kullandığınızda, kümedeki tüm kuyruk yöneticileri, yönlendirilen konu tanımlamasını barındıran küme kuyruğu yöneticilerini (yani, konu nesnesini tanımladığınız kuyruk yöneticilerini) tanır. Yayınlama ve abone olma işlemleri gerçekleştirirken, kümedeki kuyruk yöneticileri doğrudan birbirine değil, yalnızca bu konu anasistem kuyruk yöneticilerine bağlanır. Konu anasistem kuyruk yöneticileri, yayınların yayınlandığı kuyruk yöneticilerinden, eşleşen aboneliklere sahip kuyruk yöneticilerine yönlendirmekten sorumludur. Bkz. [Konu anasistemi tarafından yönlendirilen yayınlama/abone olma kümeleri](#).

**Not:** Bir konu nesnesi kümelendikten sonra ( **CLUSTER** özelliğini ayarlayarak ) **CLROUTE** özelliğinin değerini değiştiremezsiniz. Değeri değiştirebilmeniz için nesnenin kümelendirilmemiş ( **CLUSTER** , ' ' olarak ayarlanmış ) olması gerekir. Bir konuyu kümelemeyi geri alma, konu tanımlamasını yerel konuya dönüştürür; bu da yayınların uzak kuyruk yöneticilerine aboneliğe teslim edilmediği bir dönemle sonuçlanır; bu değişiklik gerçekleştirilirken dikkate alınmalıdır. Başka bir kuyruk yöneticisinden küme konusuyla aynı ada sahip küme dışı bir konu tanımlamanın etkisiz başlıklı konuya bakın. **CLROUTE** özelliğinin değerini kümeli olarak değiştirmeyi denerseniz, sistem bir MQRCCF\_CLROUTE\_NOT\_ALTERABLE kural dışı durumu oluşturur.

## Yordam

1. Konunuzu barındıracak bir kuyruk yöneticisi seçin.

Herhangi bir küme kuyruğu yöneticisi bir konuyu barındıracak. Üç kuyruk yöneticilerinden ( LONDON, NEWYORK ya da PARIS) birini seçin ve TOPIC nesnesinin özelliklerini yapılandırın. Doğrudan yöneltmeyi kullanmayı planlıyorsanız, hangi kuyruk yöneticisini seçeceğinizi işletimsel olarak fark etmez. Konu anasistem yönlendirmesini kullanmayı planlıyorsanız, seçilen kuyruk yöneticisinin yayınları yönlendirmeye ilişkin ek sorumlulukları vardır. Bu nedenle, konu anasistemi yönlendirmesi için, daha güçlü sistemlerinizin birinde barındırılan ve iyi ağ bağlantısına sahip bir kuyruk yöneticisi seçin.

2. Kuyruk yöneticisinde konu tanımlanması.

Konuyu bir küme konusu yapmak için, konuyu tanımlarken küme adını ekleyin ve bu konuya ilişkin yayınlar ve abonelikler için kullanmak istediğiniz yönlendirmeyi ayarlayın. Örneğin, LONDON kuyruk yöneticisinde doğrudan yöneltme kümesi konusu yaratmak için konuyu aşağıdaki gibi yaratın:

```
DEFINE TOPIC(INVENTORY) TOPICSTR('/INVENTORY') CLUSTER(INVENTORY) CLROUTE(DIRECT)
```

Kümedeki bir kuyruk yöneticisinde bir küme konusu tanımlayarak, konuyu tüm kümenin kullanımına sunuyorsunuz.

**CLROUTE**kullanımı hakkında daha fazla bilgi için bkz. [DEFINE TOPIC \(CLROUTE\)](#) ve [Routing for publish/abone olma kümeleri: Notes on behavior](#).

## Sonuçlar

Küme, konuya ilişkin yayınları ve abonelikleri almaya hazır.

## Sonraki adım

Bir konu anasistemi tarafından yöneltilecek yayınlar/abone olma kümesi yapılandırdıysanız, büyük olasılıkla bu konu için ikinci bir konu anasistemi eklemek istersiniz. Bkz. [“Konu anasistemi tarafından yönlendirilen kümeye ek konu anasistemleri eklenmesi” sayfa 430.](#)

Birkaç ayrı yayınlar/abone olma kümeniz varsa (örneğin, kuruluşunuz coğrafi olarak dağınık olduğu için), bazı küme konularını tüm kümelere yaymak isteyebilirsiniz. Bunu, bir sıradüzendeki kümeleri bağlayarak yapabilirsiniz. Bkz. [“Birden çok kümenin konu alanlarını birleştirme” sayfa 435.](#) Ayrıca, bir kümeden diğerine hangi yayınların akmasını denetleyebilirsiniz. Bkz. [“Birden çok kümede konu alanlarını birleştirme ve yalıtma” sayfa 437.](#)

## İlgili kavramlar

[Yayınlar ve abonelik kapsamlarını birleştirme](#)

IBM WebSphere MQ 7.0 ' den itibaren yayın ve abonelik kapsamı, kuyruk yöneticileri arasındaki yayınların akışını belirlemek için bağımsız olarak çalışır.

[Yayınlar/abone olma ağlarında konu alanlarını birleştirme](#)

Bir kuyruk yöneticisinin konu alanını, bir yayınlar/abone olma kümesindeki ya da sıradüzendeki diğer kuyruk yöneticileriyle birleştirin. Yayınlar/abone olma kümelerini birleştirin ve kümeleri sıradüzenerle yayınlar/abone olun.

## İlgili görevler

[Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması](#)

Yöneltilecek ya da doğrudan yöneltilecek konu kümelerinde, bir kuyruk yöneticisinin devreden çıkarılması sırasında ya da bir küme kuyruk yöneticisinin başarısız olması ya da önemli bir süre için uygun olmaması nedeniyle bir küme konusu tanımlamasını taşımanız gerekebilir.

[Konu anasistemi tarafından yönlendirilen kümeye ek konu anasistemleri eklenmesi](#)

Bir konu anasistemi tarafından yönlendirilen yayınlar/abone olma kümesinde, bu kuyruk yöneticilerindeki aynı kümelenebilir konu nesnesini tanımlayarak yayınları aboneliklere yönlendirmek için birden çok kuyruk yöneticisi kullanılabilir. Bu, kullanılabilirliği ve iş yükü dengelemeyi geliştirmek için kullanılabilir. Aynı küme konusu nesnesi için ek bir konu anasistemi eklediğinizde, yayınların yeni



konu anasistemi aracılığıyla yöneltilmeye ne zaman başlayacağını denetlemek için **PUB** parametresini kullanabilirsiniz.

Kuyruk yöneticisinin yayınlama/abone olma sıradüzenine bağlanması

Alt kuyruk yöneticisini sıradüzendeki üst kuyruk yöneticisine bağlayabilirsiniz. Alt kuyruk yöneticisi zaten başka bir sıradüzenin ya da kümenin üyesiye, bu bağlantı sıradüzenleri birleştirir ya da kümeyi sıradüzeniyile birleştirir.

Bir kuyruk yöneticisinin yayınlama/abone olma sıradüzeninden bağlantısının kesilmesi

Yayınlama/abone olma sıradüzenindeki bir üst kuyruk yöneticisiyle alt kuyruk yöneticisinin bağlantısını kesin.

Yayınlama/abone olma kümeleri tasarlanması

Dağıtılmış yayınlama/abone olma sorunlarının giderilmesi

Kümelenmiş yayınlama/abone olmayı engelleme

## **Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması**

Yöneltilebilir ya da doğrudan yöneltilebilir konu kümelerinde, bir kuyruk yöneticisinin devreden çıkarılması sırasında ya da bir küme kuyruk yöneticisinin başarısız olması ya da önemli bir süre için uygun olmaması nedeniyle bir küme konusu tanımlamasını taşımanız gerekebilir.

### **Bu görev hakkında**

Bir kümedeki aynı küme konusu nesnesine ilişkin birden çok tanımlamanız olabilir. Bu, konu anasistemi tarafından yönlendirilen bir küme için olağan bir durum ve doğrudan yönlendirilen bir küme için olağan dışı bir durumdur. Daha fazla bilgi için [Aynı adı taşıyan birden çok küme konusu tanımlaması](#) başlıklı konuya bakın.

Bir küme konusu tanımlamasını, yayınların akışını kesmeden kümedeki farklı bir kuyruk yöneticisine taşımak için aşağıdaki adımları tamamlayın. Yordam, bir tanımlama QM1 kuyruk yöneticisinden QM2 kuyruk yöneticisine taşır.

### **Yordam**

1. QM2 üzerinde küme konusu tanımının bir kopyasını oluşturun.

Doğrudan yöneltme için, tüm öznitelikleri QM1 tanımlamasıyla eşleşecek şekilde ayarlayın.

Konu anasistem yönlendirmesi için, başlangıçta yeni konu anasistemini **PUB (DISABLED)** olarak tanımlayın. Bu, QM2 'nin kümedeki abonelikleri öğrenmesini sağlar, ancak yayınları yönlendirmeyi başlatmaz.

2. Bilgilerin kümede yayılmasını bekleyin.

Yeni küme konusu tanımlamasının tam havuz kuyruğu yöneticileri tarafından kümedeki tüm kuyruk yöneticilerine yayılmasını bekleyin. Her küme üyesindeki küme konularını görüntülemek için **DISPLAY CLUSTER** komutunu kullanın ve QM2 kaynaklı bir tanımlama denetleyin.

Konu anasistem yönlendirmesi için, QM2 üzerindeki yeni konu anasisteminin tüm abonelikleri öğrenmesini bekleyin. QM2 ile QM1 tarafından bilinen yetkili sunucu aboneliklerini karşılaştırın. Bir kuyruk yöneticisinde yetkili sunucu aboneliklerini görüntülemek için aşağıdaki **runmqsc** komutunu vermeniz gerekir:

```
DISPLAY SUB(*) SUBTYPE(PROXY)
```

3. Konu anasistem yönlendirmesi için, QM2 üzerindeki konu anasistemini **PUB (ENABLED)** olarak yeniden tanımlayın ve QM1 üzerindeki konu anasistemini **PUB (DISABLED)** olarak yeniden tanımlayın.

QM2 üzerindeki yeni konu anasistemi artık diğer kuyruk yöneticilerindeki tüm abonelikleri öğrendiğine göre, konu anasistemi yayınları yönlendirmeye başlayabilir.

QM1 aracılığıyla ileti trafiğini susturmak için PUB (DISABLED) ayarını kullanarak, küme konu tanımlamasını sildiğinizde QM1 aracılığıyla hiçbir yayın eğitilmediğinden emin olun.

#### 4. Küme konusu tanımlamasını QM1 içinden silin.

Kuyruk yöneticisi kullanılabiliriyorsa, tanımlamayı QM1 içinden silebilirsiniz. Tersi durumda, QM1 yeniden başlatılincaya ya da zorla kaldırılincaya kadar her iki tanımla da çalışmanız gerekir.

QM1 uzun süre kullanılamaz durumdaysa ve bu süre içinde QM2 üzerinde kümelenmiş konu tanımını değiştirmeniz gerekirse, QM2 tanımı QM1 tanımından daha yenidir ve bu nedenle genellikle geçerli olur.

Bu dönemde, QM1 ve QM2 üzerindeki tanımlar arasında farklılıklar varsa, hatalar her iki kuyruk yöneticisinin hata günlüklerine yazılır ve çakışan küme konusu tanımı hakkında sizi uyarır.

QM1 hiçbir zaman kümeye geri dönmezse, örneğin bir donanım arızasından sonra beklenmeyen kullanımdan kaldırma nedeniyle, son çare olarak kuyruk yöneticisini zorla çıkarmak için **RESET CLUSTER** komutunu kullanabilirsiniz. **RESET CLUSTER**, hedef kuyruk yöneticisinde bulunan tüm konu nesnelerini otomatik olarak siler.

### İlgili kavramlar

#### Yayınlama ve abonelik kapsamlarını birleştirme

IBM WebSphere MQ 7.0 ' den itibaren yayın ve abonelik kapsamı, kuyruk yöneticileri arasındaki yayınların akışını belirlemek için bağımsız olarak çalışır.

#### Yayınlama/abone olma ağlarında konu alanlarını birleştirme

Bir kuyruk yöneticisinin konu alanını, bir yayınlama/abone olma kümesindeki ya da sıradüzenindeki diğer kuyruk yöneticileriyle birleştirin. Yayınlama/abone olma kümelerini birleştirin ve kümeleri sıradüzenlerle yayınlayın/abone olun.

### İlgili görevler

#### Yayınlama/abone olma kümesinin yapılandırılması

Kuyruk yöneticisinde bir konu tanımlayın. Konuyu bir küme konusu yapmak için **CLUSTER** özelliğini ayarlayın. Bu konuya ilişkin yayınlar ve abonelikler için kullanılacak yönlendirmeyi seçmek üzere **CLROUTE** özelliğini ayarlayın.

#### Konu anasistemi tarafından yönlendirilen kümeye ek konu anasistemleri eklenmesi

Bir konu anasistemi tarafından yönlendirilen yayınlama/abone olma kümesinde, bu kuyruk yöneticilerindeki aynı kümelenmiş konu nesnesini tanımlayarak yayınları aboneliklere yönlendirmek için birden çok kuyruk yöneticisi kullanılabilir. Bu, kullanılabilirliği ve iş yükü dengelemeyi geliştirmek için kullanılabilir. Aynı küme konusu nesnesi için ek bir konu anasistemi eklediğinizde, yayınların yeni konu anasistemi aracılığıyla yöneltilmeye ne zaman başlayacağını denetlemek için **PUB** parametresini kullanabilirsiniz.

#### Kuyruk yöneticisinin yayınlama/abone olma sıradüzenine bağlanması

Alt kuyruk yöneticisini sıradüzendeki üst kuyruk yöneticisine bağlayabilirsiniz. Alt kuyruk yöneticisi zaten başka bir sıradüzenin ya da kümenin üyesiye, bu bağlantı sıradüzenleri birleştirir ya da kümeyi sıradüzenleriyle birleştirir.

#### Bir kuyruk yöneticisinin yayınlama/abone olma sıradüzeninden bağlantısının kesilmesi

Yayınlama/abone olma sıradüzenindeki bir üst kuyruk yöneticisiyle alt kuyruk yöneticisinin bağlantısını kesin.

### **Konu anasistemi tarafından yönlendirilen kümeye ek konu anasistemleri eklenmesi**

Bir konu anasistemi tarafından yönlendirilen yayınlama/abone olma kümesinde, bu kuyruk yöneticilerindeki aynı kümelenmiş konu nesnesini tanımlayarak yayınları aboneliklere yönlendirmek için birden çok kuyruk yöneticisi kullanılabilir. Bu, kullanılabilirliği ve iş yükü dengelemeyi geliştirmek için kullanılabilir. Aynı küme konusu nesnesi için ek bir konu anasistemi eklediğinizde, yayınların yeni konu anasistemi aracılığıyla yöneltilmeye ne zaman başlayacağını denetlemek için **PUB** parametresini kullanabilirsiniz.

## Başlamadan önce

Birden çok kuyruk yöneticisine aynı küme konusu nesnesi tanımlanması yalnızca, konu anasistemi tarafından yönlendirilen bir küme için işlevsel olarak yararlıdır. Doğrudan yöneltilen bir kümede eşleşen birden çok konu tanımlanması, bunun davranışını değiştirmez. Bu görev yalnızca konu anasistemi tarafından yönlendirilen kümeler için geçerlidir.

Bu görevde, özellikle aşağıdaki kısımlar olmak üzere Aynı adı taşıyan birden çok küme konusu tanımlaması makalesini okuduğunuz varsayılmıştır:

- Bir konu anasistemi tarafından yönlendirilen kümede birden çok küme konusu tanımlaması
- PUB parametresi için özel işleme

## Bu görev hakkında

Bir kuyruk yöneticisi yöneltilen bir konu anasistemi yapıldığında, öncelikle kümede abone olunan tüm ilgili konuların varlığını öğrenmelidir. Yayınlar, ek konu anasistemi eklendiği sırada bu konulara yayınlanıyorsa ve bir yayın, anasistem kümedeki diğer kuyruk yöneticilerine aboneliklerin varlığını öğrenmeden önce yeni anasisteme yöneltilmişse, yeni anasistem o yayını bu aboneliklere iletmez. Bu, aboneliklerin yayınları kaçırmasına neden olur.

Yayınlar, küme konusu nesnesi **PUB** değiştirgesini belirttik olarak DISABLED değerine ayarlanan konu anasistem kuyruğu yöneticileri aracılığıyla yöneltilmez; bu nedenle, ek konu anasistemi ekleme işlemi sırasında hiçbir aboneliğe ilişkin yayınların eksik olmadığından emin olmak için bu ayarı kullanabilirsiniz.

**Not:** Bir kuyruk yöneticisi PUB (DISABLED) olarak tanımlanmış bir küme konusunu içerirken, o kuyruk yöneticisine bağlı yayıncılar ileti yayınlamamaz ve bu kuyruk yöneticisindeki eşleşen abonelikler, kümedeki diğer kuyruk yöneticilerine yayınlanan yayınları almaz. Bu nedenle, aboneliklerin bulunduğu ve yayınlama uygulamalarının bağlandığı kuyruk yöneticilerine konu anasistemi tarafından yönlendirilen konuların tanımlanmasına dikkat edilmelidir.

## Yordam

1. Yeni bir konu anasistemi yapılandırın ve başlangıçta yeni konu anasistemini PUB (DISABLED) olarak tanımlayın.

Bu, yeni konu anasisteminin kümedeki abonelikleri öğrenmesini sağlar, ancak yayınları yönlendirmeyi başlatmaz.

Konu anasistemini yapılandırma hakkında bilgi için bkz. "Yayınlama/abone olma kümesinin yapılandırılması" sayfa 427.

2. Yeni konu anasisteminin tüm abonelikleri ne zaman öğrendiğini belirleyin.

Bunu yapmak için, yeni konu anasisteminin tanıdığı yetkili sunucu aboneliklerini ve var olan konu anasisteminin tanıdığı abonelikleri karşılaştırın. Yetkili sunucu aboneliklerini görüntülemenin bir yolu şu **runmqsc** komutunu vermektir: `DISPLAY SUB(*) SUBTYPE (PROXY)`

3. Yeni konu anasistemini PUB (ENABLED) olarak yeniden tanımlayın.

Yeni konu anasistemi diğer kuyruk yöneticilerindeki tüm abonelikleri öğrendikten sonra, konu yayınları yönlendirmeye başlayabilir.

## İlgili kavramlar

Yayınlama ve abonelik kapsamlarını birleştirme

IBM WebSphere MQ 7.0 ' den itibaren yayın ve abonelik kapsamı, kuyruk yöneticileri arasındaki yayınların akışını belirlemek için bağımsız olarak çalışır.

Yayınlama/abone olma ağlarında konu alanlarını birleştirme

Bir kuyruk yöneticisinin konu alanını, bir yayınlama/abone olma kümesindeki ya da sıradüzenindeki diğer kuyruk yöneticileriyle birleştirin. Yayınlama/abone olma kümelerini birleştirin ve kümeleri sıradüzenlerle yayınlayın/abone olun.

## İlgili görevler

Yayınlama/abone olma kümesinin yapılandırılması

Kuyruk yöneticisinde bir konu tanımlayın. Konuyu bir küme konusu yapmak için **CLUSTER** özelliğini ayarlayın. Bu konuya ilişkin yayınlar ve abonelikler için kullanılacak yönlendirmeyi seçmek üzere **CLROUTE** özelliğini ayarlayın.

Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması

Yöneltilen ya da doğrudan yöneltilen konu kümelerinde, bir kuyruk yöneticisinin devreden çıkarılması sırasında ya da bir küme kuyruk yöneticisinin başarısız olması ya da önemli bir süre için uygun olmaması nedeniyle bir küme konusu tanımlamasını taşımanız gerekebilir.

Kuyruk yöneticisinin yayınlama/abone olma sıradüzenine bağlanması

Alt kuyruk yöneticisini sıradüzendeki üst kuyruk yöneticisine bağlayabilirsiniz. Alt kuyruk yöneticisi zaten başka bir sıradüzenin ya da kümenin üyesiye, bu bağlantı sıradüzenleri birleştirir ya da kümeyi sıradüzeniyile birleştirir.

Bir kuyruk yöneticisinin yayınlama/abone olma sıradüzeninden bağlantısının kesilmesi

Yayınlama/abone olma sıradüzenindeki bir üst kuyruk yöneticisiyle alt kuyruk yöneticisinin bağlantısını kesin.

## Yayınlama ve abonelik kapsamlarını birleştirme

IBM WebSphere MQ 7.0 ' den itibaren yayın ve abonelik kapsamı, kuyruk yöneticileri arasındaki yayınların akışını belirlemek için bağımsız olarak çalışır.

Yayınlar, bir yayınlama/abone olma topolojisine bağlı tüm kuyruk yöneticilerine ya da yalnızca yerel kuyruk yöneticisine akabilir. Benzer şekilde, yetkili sunucu abonelikleri için de. Bir abonelikte eşleşen yayınlar, bu iki akışın birleşimiyle yönetilir.

Yayınların ve aboneliklerin kapsamı QMGR ya da ALL olabilir. Bir yayıncı ve bir abone aynı kuyruk yöneticisine bağlıysa, kapsam ayarları abonenin o yayıncıdan aldığı yayınları etkilemez.

Yayıncı ve abone farklı kuyruk yöneticilerine bağlıysa, uzak yayınları almak için her iki ayarın da ALL olması gerekir.

Yayıncıların farklı kuyruk yöneticilerine bağlı olduğunu varsayalım. Bir abonenin herhangi bir yayıncıdan yayın almasını istiyorsanız, abonelik kapsamını ALL olarak ayarlayın. Daha sonra her yayıncı için yayınlarının kapsamını yayıncının yerel aboneleri ile sınırlayıp sınırlamamaya karar verebilirsiniz.

Abonelerin farklı kuyruk yöneticilerine bağlı olduğunu varsayalım. Bir yayıncıdaki yayınların tüm abonelere gönderilmesini istiyorsanız, yayın kapsamını ALL olarak ayarlayın. Bir abonenin yayınları yalnızca aynı kuyruk yöneticisine bağlı bir yayıncıdan almasını istiyorsanız, abonelik kapsamını QMGR olarak ayarlayın.

## Örnek: futbol sonuçları hizmeti

Futbol liginde bir takım olduğunuzu varsayalım. Her ekibin, bir yayınlama/abone olma kümesindeki diğer tüm ekiplere bağlı bir kuyruk yöneticisi vardır.

Ekipler, `Football/result/Home team name/Away team name` konusunu kullanarak evlerinde oynanan tüm oyunların sonuçlarını yayınlıyor. İtalik dizgiler değişken konu adlarıdır ve yayın eşleştirmenin sonucudur.

Ayrıca her kulüp, `Football/myteam/Home team name/Away team name` konu dizisini kullanarak kulüp için sonuçları yeniden yayınlar.

Her iki konu da tüm kümede yayınlanır.

Aşağıdaki abonelikler, herhangi bir takımın hayranlarının sonuçlara üç ilginç şekilde abone olabilmeleri için lig tarafından ayarlandı.

SUBSCOPE (QMGR) ile küme konularını ayarlayabileceğinize dikkat edin. Konu tanımlamaları kümenin her üyesine yayılır, ancak aboneliğin kapsamı yalnızca yerel kuyruk yöneticisidir. Bu nedenle, her kuyruk yöneticisindeki aboneler aynı abonelikten farklı yayınlar alır.

## Tüm sonuçları al

```
DEFINE TOPIC(A) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(ALL)
```

## Tüm ana sayfa sonuçlarını al

```
DEFINE TOPIC(B) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(QMGR)
```

Abonelik QMGR kapsamına sahip olduğundan, yalnızca ana alanda yayınlanan sonuçlar eşleştirilir.

## Tüm ekip sonuçlarını al

```
DEFINE TOPIC(C) TOPICSTR('Football/myteam/') CLUSTER SUBSCOPE(QMGR)
```

Aboneliğin QMGR kapsamı olduğundan, yalnızca yerel olarak yeniden yayınlanan yerel ekip sonuçları eşleştirilir.

## İlgili kavramlar

Yayınlama/abone olma ağlarında konu alanlarını birleştirme

Bir kuyruk yöneticisinin konu alanını, bir yayınlama/abone olma kümesindeki ya da sıradüzenindeki diğer kuyruk yöneticileriyle birleştirin. Yayınlama/abone olma kümelerini birleştirin ve kümeleri sıradüzenlerle yayınlayın/abone olun.

Dağıtılmış yayınlama/abone olma ağları

Yayın kapsamı

Abonelik kapsamı

## İlgili görevler

Yayınlama/abone olma kümesinin yapılandırılması

Kuyruk yöneticisinde bir konu tanımlayın. Konuyu bir küme konusu yapmak için **CLUSTER** özelliğini ayarlayın. Bu konuya ilişkin yayınlar ve abonelikler için kullanılacak yönlendirmeyi seçmek üzere **CLROUTE** özelliğini ayarlayın.

Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması

Yöneltilen ya da doğrudan yöneltilen konu kümelerinde, bir kuyruk yöneticisinin devreden çıkarılması sırasında ya da bir küme kuyruk yöneticisinin başarısız olması ya da önemli bir süre için uygun olmaması nedeniyle bir küme konusu tanımlamasını taşımanız gerekebilir.

Konu anasistemi tarafından yönlendirilen kümeye ek konu anasistemleri eklenmesi

Bir konu anasistemi tarafından yönlendirilen yayınlama/abone olma kümesinde, bu kuyruk yöneticilerindeki aynı kümelenebilir konu nesnesini tanımlayarak yayınları aboneliklere yönlendirmek için birden çok kuyruk yöneticisi kullanılabilir. Bu, kullanılabilirliği ve iş yükü dengelemeyi geliştirmek için kullanılabilir. Aynı küme konusu nesnesi için ek bir konu anasistemi eklediğinizde, yayınların yeni konu anasistemi aracılığıyla yönetilmeye ne zaman başlayacağını denetlemek için **PUB** parametresini kullanabilirsiniz.

Kuyruk yöneticisinin yayınlama/abone olma sıradüzenine bağlanması

Alt kuyruk yöneticisini sıradüzendeki üst kuyruk yöneticisine bağlayabilirsiniz. Alt kuyruk yöneticisi zaten başka bir sıradüzenin ya da kümenin üyesiye, bu bağlantı sıradüzenleri birleştirir ya da kümeyi sıradüzenleriyle birleştirir.

Bir kuyruk yöneticisinin yayınlama/abone olma sıradüzeninden bağlantısının kesilmesi

Yayınlama/abone olma sıradüzenindeki bir üst kuyruk yöneticisiyle alt kuyruk yöneticisinin bağlantısını kesin.

## Yayınlama/abone olma ağlarında konu alanlarını birleştirme

Bir kuyruk yöneticisinin konu alanını, bir yayınlama/abone olma kümesindeki ya da sıradüzenindeki diğer kuyruk yöneticileriyle birleştirin. Yayınlama/abone olma kümelerini birleştirin ve kümeleri sıradüzenlerle yayınlayın/abone olun.

**CLUSTER, PUBSCOPE** ve **SUBSCOPE** özniteliklerinin yapı taşlarını, yayınlama/abone olma kümelerini ve yayınlama/abone olma sıradüzenlerini kullanarak farklı yayınlama/abone olma konu alanları oluşturabilirsiniz.

Tek bir kuyruk yöneticisinden yayınlama/abone olma kümesine ölçekleme örneğinden başlayarak, aşağıdaki senaryolar farklı yayınlama/abone olma topolojilerini göstermektedir.

### **İlgili kavramlar**

Yayınlama ve abonelik kapsamlarını birleştirme

IBM WebSphere MQ 7.0 ' den itibaren yayın ve abonelik kapsamı, kuyruk yöneticileri arasındaki yayınların akışını belirlemek için bağımsız olarak çalışır.

Dağıtılmış yayınlama/abone olma ağları

Konu alanları

### **İlgili görevler**

Yayınlama/abone olma kümesinin yapılandırılması

Kuyruk yöneticisinde bir konu tanımlayın. Konuyu bir küme konusu yapmak için **CLUSTER** özelliğini ayarlayın. Bu konuya ilişkin yayınlar ve abonelikler için kullanılacak yönlendirmeyi seçmek üzere **CLROUTE** özelliğini ayarlayın.

Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması

Yöneltilen ya da doğrudan yöneltilen konu kümelerinde, bir kuyruk yöneticisinin devreden çıkarılması sırasında ya da bir küme kuyruk yöneticisinin başarısız olması ya da önemli bir süre için uygun olmaması nedeniyle bir küme konusu tanımlamasını taşımanız gerekebilir.

Konu anasistemi tarafından yönlendirilen kümeye ek konu anasistemleri eklenmesi

Bir konu anasistemi tarafından yönlendirilen yayınlama/abone olma kümesinde, bu kuyruk yöneticilerindeki aynı kümelenebilir konu nesnesini tanımlayarak yayınları aboneliklere yönlendirmek için birden çok kuyruk yöneticisi kullanılabilir. Bu, kullanılabilirliği ve iş yükü dengelemeyi geliştirmek için kullanılabilir. Aynı küme konusu nesnesi için ek bir konu anasistemi eklediğinizde, yayınların yeni konu anasistemi aracılığıyla yönetilmeye ne zaman başlayacağını denetlemek için **PUB** parametresini kullanabilirsiniz.

Kuyruk yöneticisinin yayınlama/abone olma sıradüzenine bağlanması

Alt kuyruk yöneticisini sıradüzendeki üst kuyruk yöneticisine bağlayabilirsiniz. Alt kuyruk yöneticisi zaten başka bir sıradüzenin ya da kümenin üyesiye, bu bağlantı sıradüzenleri birleştirir ya da kümeyi sıradüzeniyle birleştirir.

Bir kuyruk yöneticisinin yayınlama/abone olma sıradüzeninden bağlantısının kesilmesi

Yayınlama/abone olma sıradüzenindeki bir üst kuyruk yöneticisiyle alt kuyruk yöneticisinin bağlantısını kesin.

Küme konularının tanımlanması

### ***Yayınlama/abone olma kümesinde tek bir konu alanı yaratılması***

Bir yayınlama/abone olma sistemini birden çok kuyruk yöneticileriyle çalışacak şekilde ölçeklendirin. Her yayınlama/abone olma sistemine tek bir özdeş konu alanı sağlamak için bir yayınlama/abone olma kümesi kullanın.

### **Başlamadan önce**

Tek bir sürüm 7 kuyruk yöneticisine yayınlama/abone olma sistemi uyguladınız.

SYSTEM.BASE.TOPIC özniteliklerini edinmeye güvenmek yerine, her zaman kendi kök konularına sahip konu alanları oluşturun. Yayınlama/abone olma sisteminizi bir kümeye kadar ölçeklendirirseniz, kök konularınızı küme konuları olarak, küme konusu anasistemi üzerinde tanımlayabilir ve ardından tüm konularınız kümede paylaşılır.

### **Bu görev hakkında**

Şimdi sistemi, daha fazla yayıncı ve aboneyi destekleyecek şekilde ölçeklendirmek ve her konuyu küme içinde görmek istiyorsunuz.

## Yordam

1. Yayınlama/abone olma sistemiyle kullanılacak bir küme oluşturun.  
Var olan bir geleneksel kümeniz varsa, performans nedenleriyle yeni yayınlama abone olma sistemi için yeni bir küme oluşturmanız daha iyi olur. Her iki kümenin küme havuzları için aynı sunucuları kullanabilirsiniz
2. Küme konusu anasistemi olarak bir kuyruk yöneticisi (büyük olasılıkla havuzlardan biri) seçin.
3. Yayınlama/abone olma kümesinde görünür olacak her konunun bir yönetim konusu nesnesine çözümlendiğinden emin olun.  
Yayınlama/abone olma kümesini adlandırma **CLUSTER** özneteliğini ayarlayın.

## Sonraki adım

Yayıncı ve abone uygulamalarını kümedeki kuyruk yöneticilerine bağlayın.

**CLUSTER** özneteliğine sahip yönetim konusu nesnelere oluşturun. Konular kümede de yayılır. Yayıncı ve abone programları, kümedeki farklı kuyruk yöneticilerine bağlanarak davranışlarının değiştirilmemesi için denetim konularını kullanır.

SYSTEM.BASE.TOPIC ' in her kuyruk yöneticisinde bir küme konusu gibi davranması gerekiyorsa, bunu her kuyruk yöneticisinde değiştirmeniz gerekir.

### İlgili kavramlar

[Dağıtılmış yayınlama/abone olma ağları](#)

[Konu alanları](#)

### İlgili görevler

[Birden çok kümenin konu alanlarını birleştirme](#)

Birden çok kümeyi kapsayan konu alanları oluşturun. Bir kümedeki bir konuyu yayınlayın ve başka bir kümede konuya abone olun.

[Birden çok kümede konu alanlarını birleştirme ve yalıtma](#)

Bazı konu alanlarını belirli bir kümede yalıtın ve tüm bağlı kümelerde erişilebilir hale getirmek için diğer konu alanlarını birleştirin.

[Birden çok kümede konu alanlarını yayınlama ve bunlara abone olma](#)

Çakışan kümeleri kullanarak birden çok kümedeki konuları yayınlayın ve bunlara abone olun. Kümelerdeki konu alanları çakışmadığı sürece bu tekniği kullanabilirsiniz.

[Küme konularının tanımlanması](#)

### ***Birden çok kümenin konu alanlarını birleştirme***

Birden çok kümeyi kapsayan konu alanları oluşturun. Bir kümedeki bir konuyu yayınlayın ve başka bir kümede konuya abone olun.

### **Başlamadan önce**

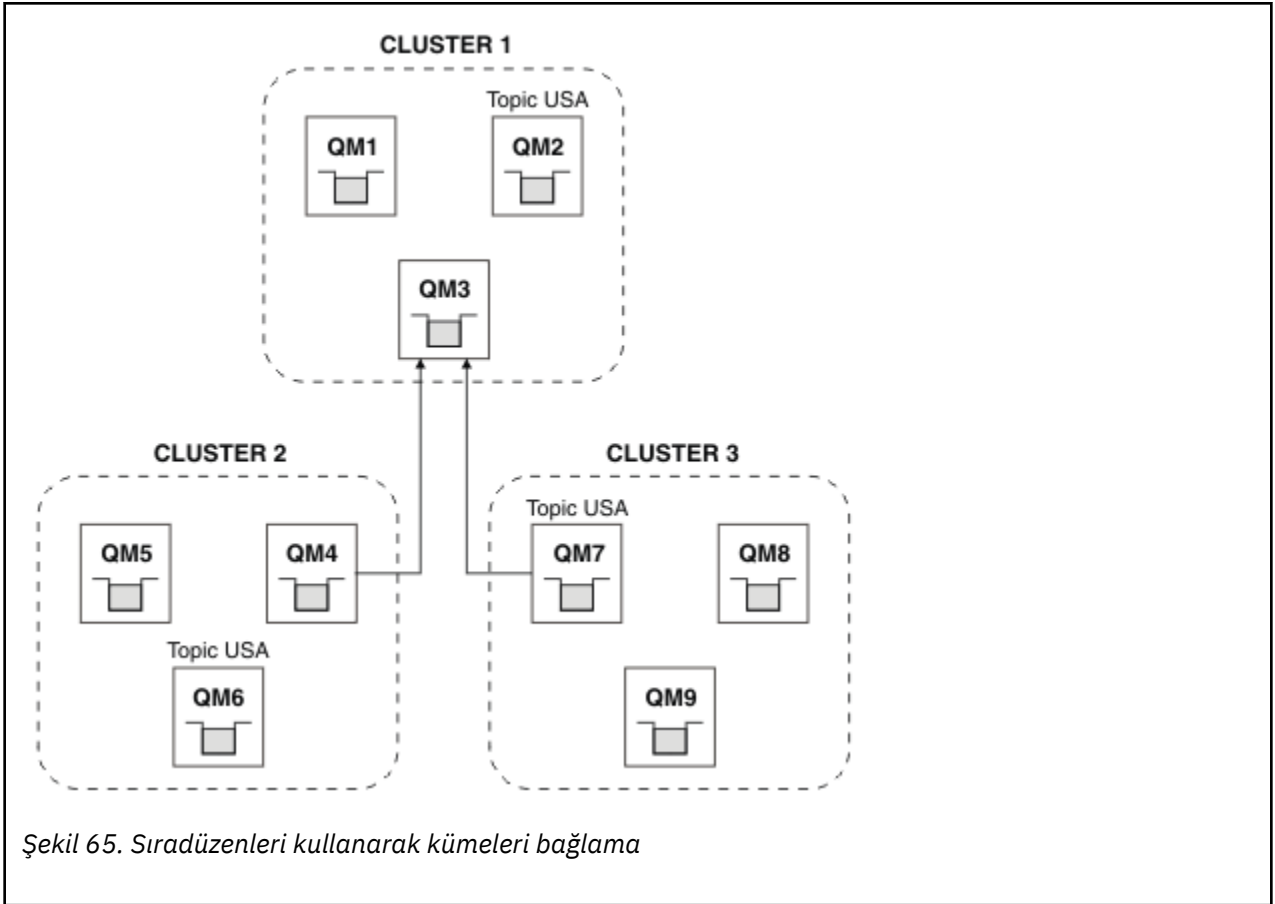
Bu görev, var olan doğrudan yönlendirilmiş yayınlama/abone olma kümeleriniz olduğunu ve bazı küme konularını tüm kümelere yaymak istediğinizi varsayar.

**Not:** Bunu, anasistem tarafından yönlendirilen yayınlama/abone olma kümeleri için yapamazsınız.

### **Bu görev hakkında**

Yayıncıyı bir kümeden diğerine yaymak için kümeleri bir sıradüzende birleştirmeniz gerekir; bkz. [Şekil 65 sayfa 436](#). Sıradüzensel bağlantılar, bağlı kuyruk yöneticileri arasında abonelikleri ve yayıncıyı yayacaktır ve kümeler, kümeler arasında değil, her küme içinde küme konularını yayacaktır.

Bu iki mekanizmanın birleşimi, küme konularını tüm kümeler arasında yayacaktır. Küme konusu tanımlamalarını her kümede yinelemeniz gerekir.



Şekil 65. Sıradüzenleri kullanarak kümeleri bağlama

Aşağıdaki adımlar, kümeleri bir sıradüzene bağlar.

### Yordam

1. Her iki yönde QM3 ve QM4 ve QM3 ve QM7 ile bağlantı kurmak için iki gönderen alıcı kanal kümesi oluşturun. Bir sıradüzeni bağlamak için küme yerine geleneksel gönderen alıcı kanallarını ve iletim kuyruklarını kullanmanız gerekir.
2. Hedef kuyruk yöneticilerinin adlarını içeren üç iletim kuyruğu yaratın. Bir nedenden ötürü, iletim kuyruğu adı olarak hedef kuyruk yöneticisinin adını kullanamazsanız, kuyruk yöneticisi diğer adlarını kullanın.
3. Gönderen kanallarını tetikleme için iletim kuyruklarını yapılandırın.
4. QM3, QM4 ve QM7 **PSMODE** ögesinin ENABLE olarak ayarlanmasını denetleyin.
5. QM4 ve QM7 için **PARENT** özniteliğini QM3 olarak değiştirin.
6. Kuyruk yöneticileri arasındaki üst-alt ilişkisinin her iki yönde de etkin olup olmadığını denetleyin.
7. **CLUSTER** ( 'CLUSTER 1' ), **CLUSTER** ( 'CLUSTER 2' ) ve **CLUSTER** ( 'CLUSTER 3' ) özniteliğiyle USA denetim konusunu yaratın. Küme 1, 2 ve 3 'teki üç küme konusu anasistem kuyruk yöneticilerinin her birinde. Küme konusu anasisteminin sıradüzenel olarak bağlı bir kuyruk yöneticisi olması gerekmez.

### Sonraki adım

Artık Şekil 65 sayfa 436 içinde USA küme konusunu yayınlatabilir ya da bu konuya abone olabilirsiniz. Yayın abonelikleri, üç kümede de yayıncılara ve abonelere akar.

USA ögesini diğer kümelerde küme konusu olarak oluşturmadığınızı varsayın. USA yalnızca QM7 üzerinde tanımlanmışsa, USA yayınları ve abonelikleri QM7, QM8, QM9 ve QM3 arasında değiştirilir. QM7, QM8, QM9 üzerinde çalışan yayıncılar ve aboneler, USA yönetim konusunun özniteliklerini devralır. QM3 üzerindeki yayıncılar ve aboneler, QM3 üzerinde SYSTEM . BASE . TOPIC özniteliklerini devralır.



Ayrıca bkz. [“Birden çok kümede konu alanlarını birleştirme ve yalıtma” sayfa 437.](#)

## **İlgili kavramlar**

[Dağıtılmış yayınlama/abone olma ağları](#)

[Konu alanları](#)

## **İlgili görevler**

[Yayınlama/abone olma kümesinde tek bir konu alanı yaratılması](#)

Bir yayınlama/abone olma sistemini birden çok kuyruk yöneticileriyle çalışacak şekilde ölçeklendirin.

Her yayınlayıcıya ve aboneye tek bir özdeş konu alanı sağlamak için bir yayınlama/abone olma kümesi kullanın.

[Birden çok kümede konu alanlarını birleştirme ve yalıtma](#)

Bazı konu alanlarını belirli bir kümede yalıtın ve tüm bağlı kümelerde erişilebilir hale getirmek için diğer konu alanlarını birleştirin.

[Birden çok kümede konu alanlarını yayınlama ve bunlara abone olma](#)

Çakışan kümeleri kullanarak birden çok kümedeki konuları yayınlayın ve bunlara abone olun. Kümelerdeki konu alanları çakışmadığı sürece bu tekniği kullanabilirsiniz.

[Küme konularının tanımlanması](#)

## ***Birden çok kümede konu alanlarını birleştirme ve yalıtma***

Bazı konu alanlarını belirli bir kümede yalıtın ve tüm bağlı kümelerde erişilebilir hale getirmek için diğer konu alanlarını birleştirin.

## **Başlamadan önce**

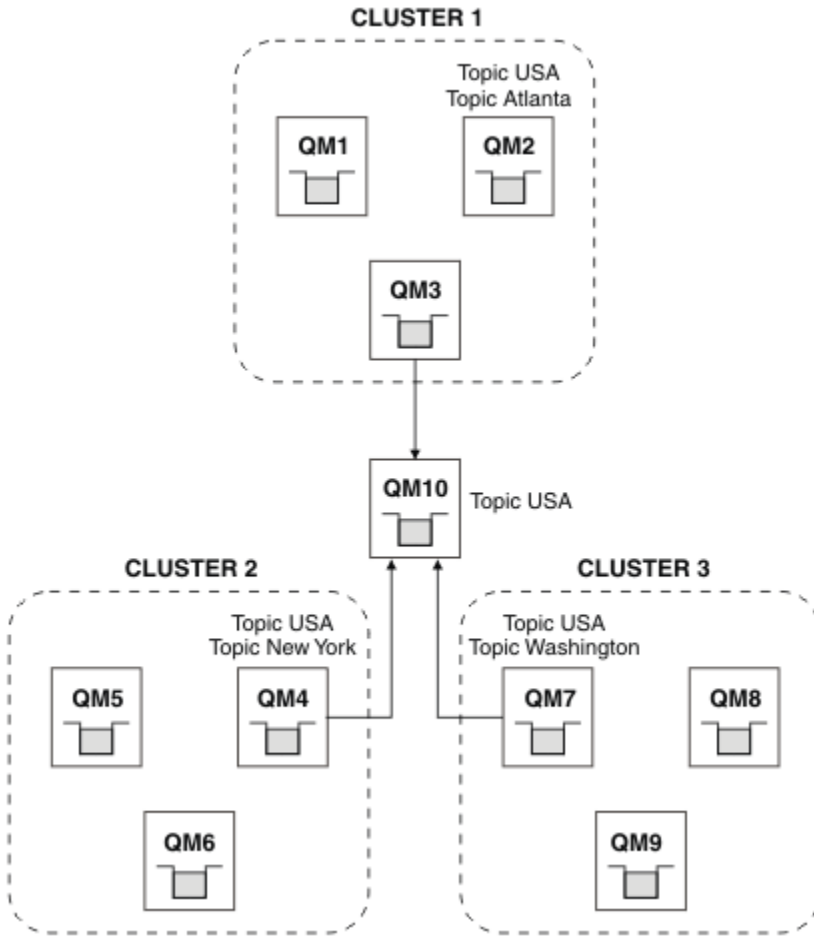
[“Birden çok kümenin konu alanlarını birleştirme” sayfa 435](#) konusunu inceleyin. Köprü olarak ek bir kuyruk yöneticisi eklemeyen, gereksinimleriniz için yeterli olabilir.

**Not:** Bu görevi yalnızca doğrudan yöneltilebilir yayınlama/abone olma kümelerini kullanarak tamamlayabilirsiniz. Bunu, konu anasistemi tarafından yönlendirilen kümeleri kullanarak yapamazsınız.

## **Bu görev hakkında**

A potential improvement on the topology shown in [Şekil 65 sayfa 436](#) in [“Birden çok kümenin konu alanlarını birleştirme” sayfa 435](#) is to isolate cluster topics that are not shared across all the clusters.

Kümelerin hiçbirinde olmayan bir köprü kuyruk yöneticisi oluşturarak kümeleri yalıtın; bkz. [Şekil 66 sayfa 438](#). Hangi yayınların ve aboneliklerin bir kümeden diğerine akabileceğini süzmek için köprü oluşturan kuyruk yöneticisini kullanın.



Şekil 66. Köprülü kümeler

Diğer kümelerdeki köprüde gösterilmesini istemediğiniz küme konularını yalıtılmak için köprüyü kullanın. Şekil 66 sayfa 438’inde USA , tüm kümelerde paylaşılan bir küme konusudur ve Atlanta, New York ve Washington , her biri yalnızca bir kümede paylaşılan küme konularını içerir.

Aşağıdaki yordamı kullanarak yapılandırmanızı modelleyin:

### Yordam

1. Tüm SYSTEM.BASE.TOPIC konu nesnelerini **SUBSCOPE** ( QMGR ) ve **PUBSCOPE** ( QMGR ) tüm kuyruk yöneticilerine.  
**SUBSCOPE** ( ALL ) belirttik olarak ayarlanmadıkça, hiçbir konu (küme konuları da dahil) diğer kuyruk yöneticilerine yayılmaz. ve **PUBSCOPE** ( ALL ) (kümeyle ilgili konularınızın kök konusuna bakın).
2. Her bir kümede paylaşılmasını istediğiniz üç küme konusu anasistem kuyruk yöneticisindeki konuları **CLUSTER** (küme adı) **SUBSCOPE** ( ALL ) öznitelikleriyle tanımlayın ve **PUBSCOPE** ( ALL ).  
Bazı küme konularının tüm kümeler arasında paylaşılmasını istiyorsanız, kümelerin her birinde aynı konuyu tanımlayın. Küme özniteliği olarak her kümenin küme adını kullanın.
3. Tüm kümeler arasında paylaşılmasını istediğiniz küme konuları için, konuları köprü kuyruk yöneticisinde ( QM10 ) **SUBSCOPE** ( ALL ) ve **PUBSCOPE** ( ALL ) öznitelikleriyle yeniden tanımlayın.

### Örnek

Şekil 66 sayfa 438’örneğinde, yalnızca USA ögesinden devralınan konular üç küme arasında yayılır.

## Sonraki adım

**SUBSCOPE** ( ALL ) ile köprü kuyruk yöneticisinde tanımlanan konulara ilişkin abonelikler ve **PUBSCOPE** ( ALL ) kümeler arasında yayılır.

Öznitelikler **CLUSTER** (*kümeadı*), **SUBSCOPE** ( ALL ) ile her kümede tanımlanan konulara ilişkin abonelikler ve **PUBSCOPE** ( ALL ) her küme içinde yayılır.

Diğer abonelikler bir kuyruk yöneticisi için yereldir.

### İlgili kavramlar

Dağıtılmış yayınlama/abone olma ağları

Konu alanları

Yayın kapsamı

Abonelik kapsamı

### İlgili görevler

Yayınlama/abone olma kümesinde tek bir konu alanı yaratılması

Bir yayınlama/abone olma sistemini birden çok kuyruk yöneticileriyle çalışacak şekilde ölçeklendirin. Her yayınlamaya ve aboneye tek bir özdeş konu alanı sağlamak için bir yayınlama/abone olma kümesi kullanın.

Birden çok kümenin konu alanlarını birleştirme

Birden çok kümeyi kapsayan konu alanları oluşturun. Bir kümedeki bir konuyu yayınlayın ve başka bir kümede konuya abone olun.

Birden çok kümede konu alanlarını yayınlama ve bunlara abone olma

Çakışan kümeleri kullanarak birden çok kümedeki konuları yayınlayın ve bunlara abone olun. Kümelerdeki konu alanları çakışmadığı sürece bu tekniği kullanabilirsiniz.

Küme konularının tanımlanması

### ***Birden çok kümede konu alanlarını yayınlama ve bunlara abone olma***

Çakışan kümeleri kullanarak birden çok kümedeki konuları yayınlayın ve bunlara abone olun. Kümelerdeki konu alanları çakışmadığı sürece bu tekniği kullanabilirsiniz.

## Başlamadan önce

Kümeler arasındaki kesişimlerde bazı kuyruk yöneticileriyle birden çok geleneksel küme oluşturun.

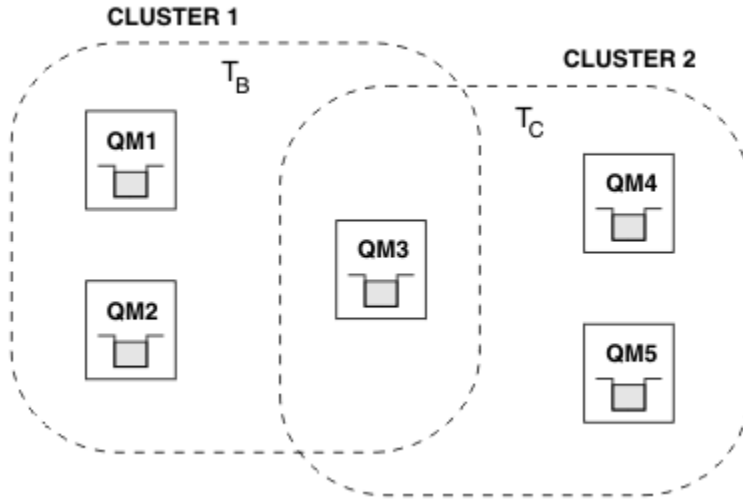
## Bu görev hakkında

Çeşitli nedenlerden ötürü kümelerle çakışmayı seçmiş olabilirsiniz.

1. Sınırlı sayıda yüksek kullanılabilirlikli sunucunuz ya da kuyruk yöneticiniz var. Tüm küme havuzlarını ve küme konusu anasistemlerini bu havuzlara konuşlandırmaya karar verirsiniz.
2. Ağ geçidi kuyruk yöneticileri kullanılarak bağlı olan geleneksel kuyruk yöneticisi kümeleriniz var. Yayınlama/abone olma uygulamalarını aynı küme topolojisine konuşlandırmak istiyorsunuz.
3. Birkaç bağımsız yayınlama/abone olma uygulamanız var. Performans nedenleriyle, yayınlama/abone olma kümelerini küçük tutmak ve geleneksel kümelerden ayrı tutmak daha iyidir. Uygulamaları farklı kümelere konuşlandırmaya karar verdiniz. Ancak, izleme uygulamasının yalnızca bir kopyasını lisansladığınız için, tek bir kuyruk yöneticisindeki tüm yayınlama/abone olma uygulamalarını da izlemek istiyorsunuz. Bu kuyruk yöneticisinin, tüm kümelerdeki küme konularına ilişkin yayınlara erişimi olmalıdır.

Konularınızın örtüşmeyen konu alanlarında tanımlandığından emin olarak, konuları çakışan yayınlama/abone olma kümelerinde konuşlandırabilirsiniz, bkz. [Şekil 67 sayfa 440](#). Konu alanları çakıştıysa, çakışan kümelere konuşlandırma sorunlara neden olur.

Yayınlama/abone olma kümeleri çakıştığı için, çakışan kuyruk yöneticilerini kullanarak herhangi bir konu alanını yayınlatabilir ve bunlara abone olabilirsiniz.



Şekil 67. Çakışan kümeler, çakışmayan konu alanları

## Yordam

Konu alanlarının çakışmamasını sağlamak için bir yöntem oluşturun.

Örneğin, konu alanlarının her biri için benzersiz bir kök konu tanımlayın. Kök konular kümesi konularını yapın.

- DEFINE TOPIC(B) TOPICSTR('B') CLUSTER('CLUSTER 1') ...
- DEFINE TOPIC(C) TOPICSTR('C') CLUSTER('CLUSTER 2') ...

## Örnek

QM3 ' e bağlı Şekil 67 sayfa 440 yayıncılarında ve abonesinde T<sub>B</sub> ya da T<sub>C</sub> yayınlatabilir ya da abone olabilir

## Sonraki adım

Her iki kümedeki konuları kullanan yayıncıları ve aboneleri çakışan kuyruk yöneticilerine bağlayın.

Yalnızca belirli bir kümedeki konuları üst üste binmeyen kuyruk yöneticilerine kullanması gereken yayıncılar ve aboneler arasında bağlantı kurun.

## İlgili kavramlar

Dağıtılmış yayınlama/abone olma ağları

Konu alanları

## İlgili görevler

Yayınlama/abone olma kümesinde tek bir konu alanı yaratılması

Bir yayınlama/abone olma sistemini birden çok kuyruk yöneticileriyle çalışacak şekilde ölçeklendirin.

Her yayıncıya ve aboneye tek bir özdeş konu alanı sağlamak için bir yayınlama/abone olma kümesi kullanın.

Birden çok kümenin konu alanlarını birleştirme

Birden çok kümeyi kapsayan konu alanları oluşturun. Bir kümedeki bir konuyu yayınlayın ve başka bir kümede konuya abone olun.

Birden çok kümede konu alanlarını birleştirme ve yalıtma

Bazı konu alanlarını belirli bir kümede yalıtın ve tüm bağlı kümelerde erişilebilir hale getirmek için diğer konu alanlarını birleştirin.

Küme konularının tanımlanması

## Kuyruk yöneticisinin yayınlama/abone olma sıradüzenine bağlanması

Alt kuyruk yöneticisini sıradüzendeki üst kuyruk yöneticisine bağlayabilirsiniz. Alt kuyruk yöneticisi zaten başka bir sıradüzenin ya da kümenin üyesiye, bu bağlantı sıradüzenleri birleştirir ya da kümeyi sıradüzenine birleştirir.

### Başlamadan önce

1. Bir yayınlama/abone olma sıradüzenindeki kuyruk yöneticilerinin benzersiz kuyruk yöneticisi adları olmalıdır.
2. Bir yayınlama/abone olma sıradüzeni, "kuyruğa alınan yayınlama/abone olma" kuyruk yöneticisi özelliğine dayanır. Bu, hem üst hem de alt kuyruk yöneticisinde etkinleştirilmelidir. Bkz. "[Kuyruğa alınan yayınlama/abone olma başlatılıyor](#)" sayfa 423.
3. Yayınlama/abone olma ilişkisi, kuyruk yöneticisi gönderen ve alıcı kanallarına dayanır. Kanalları oluşturmanın iki yolu vardır:
  - Bir IBM MQ kümesine hem üst hem de alt kuyruk yöneticilerini ekleyin. Bkz. "[Kümeye kuyruk yöneticisi eklenmesi](#)" sayfa 308.
  - Alt kuyruk yöneticisinden üst öğeye ve üst öğeden alt öğeye bir gönderen/alıcı kanal çifti oluşturun. Her kanal, hedef kuyruk yöneticisiyle aynı ada sahip bir iletim kuyruğu ya da hedef kuyruk yöneticisiyle aynı ada sahip bir kuyruk yöneticisi diğer adı kullanmalıdır. Noktadan noktaya iletişim kanal bağlantısı kurulmasına ilişkin ek bilgi için bkz. "[IBM MQ dağıtılmış kuyruğa alma teknikleri](#)" sayfa 189.

Her kanal yapılandırması tipi üzerinde bir sıradüzen yapılandırma örnekler için aşağıdaki yayınlama/abone olma sıradüzeni senaryoları kümesine bakın:

- [Senaryo 1: Kuyruk yöneticisi adı diğer adıyla noktadan noktaya iletişim kanallarının kullanılması](#)
- [Senaryo 2: İletim kuyruğu ve uzak kuyruk yöneticisi için aynı adı taşıyan noktadan noktaya iletişim kanallarının kullanılması](#)
- [Senaryo 3: Kuyruk yöneticisi eklemek için küme kanalı kullanılması](#)

### Bu görev hakkında

Alt öğeleri üst öğelere bağlamak için ALTER QMGR PARENT (*PARENT\_NAME*) **runmqsc** komutunu kullanın. Bu yapılandırma, alt kuyruk yöneticisinde gerçekleştirilir; burada *PARENT\_NAME* , üst kuyruk yöneticisinin adıdır.

### Yordam

QMGR ÜST ÖĞESİNİ DEĞİŞTİR (*PARENT\_NAME*)

### Örnek

İlk örnekte, QM2 kuyruk yöneticisinin QM1'un alt öğesi olarak nasıl ekleneceği gösterilmektedir; daha sonra, kuyruk yöneticisinin başarıyla ETKİN' un **STATUS** alt öğesi olduğunu doğrulamak için QM2 sorgusu:

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2
alter qmgr parent(QM1)
  1 : alter qmgr parent(QM1)
AMQ8005: IBM MQ queue manager changed.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.
      QMNAME(QM2)                TYPE(LOCAL)
      STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
      QMNAME(QM1)                TYPE(PARENT)
      STATUS(ACTIVE)
```

Sonraki örnekte, bağlantıları için QM1 sorgulanırken elde edilen sonuç gösterilmektedir:

```
C:\Documents and Settings\Admin>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.      TYPE(LOCAL)
      QMNAME(QM1)
      STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.      TYPE(CHILD)
      QMNAME(QM2)
      STATUS(ACTIVE)
```

**STATUS ACTIVE** olarak görünmezse, alt ve üst öge arasındaki kanalların doğru şekilde yapılandırılıp yapılandırılmadığını ve çalışıp çalışmadığını denetleyin. Olası hatalar için her iki kuyruk yöneticisi hata günlüğünü de denetleyin.

## Sonraki adım

Varsayılan olarak, bir kuyruk yöneticisindeki yayıncılar ve aboneler tarafından kullanılan konular, sıradüzendeki diğer kuyruk yöneticilerindeki yayıncılarla ve abonelerle paylaşılır. Yönetilen konular, **SUBSCOPE** ve **PUBSCOPE** konu özelliklerini kullanarak paylaşım düzeyini denetleyecek şekilde yapılandırılabilir. Bkz. [“Dağıtılmış yayınlama/abone olma ağlarının yapılandırılması” sayfa 426.](#)

### **İlgili kavramlar**

[Yayınlama ve abonelik kapsamalarını birleştirme](#)

IBM WebSphere MQ 7.0 ' den itibaren yayın ve abonelik kapsamı, kuyruk yöneticileri arasındaki yayınların akışını belirlemek için bağımsız olarak çalışır.

[Yayınlama/abone olma ağlarında konu alanlarını birleştirme](#)

Bir kuyruk yöneticisinin konu alanını, bir yayınlama/abone olma kümesindeki ya da sıradüzendeki diğer kuyruk yöneticileriyle birleştirin. Yayınlama/abone olma kümelerini birleştirin ve kümeleri sıradüzenlerle yayınlayın/abone olun.

### **İlgili görevler**

[Yayınlama/abone olma kümesinin yapılandırılması](#)

Kuyruk yöneticisinde bir konu tanımlayın. Konuyu bir küme konusu yapmak için **CLUSTER** özelliğini ayarlayın. Bu konuya ilişkin yayınlar ve abonelikler için kullanılacak yönlendirmeyi seçmek üzere **CLROUTE** özelliğini ayarlayın.

[Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması](#)

Yönetilen ya da doğrudan yönetilen konu kümelerinde, bir kuyruk yöneticisinin devreden çıkarılması sırasında ya da bir küme kuyruk yöneticisinin başarısız olması ya da önemli bir süre için uygun olmaması nedeniyle bir küme konusu tanımlamasını taşımanız gerekebilir.

[Konu anasistemi tarafından yönlendirilen kümeye ek konu anasistemleri eklenmesi](#)

Bir konu anasistemi tarafından yönlendirilen yayınlama/abone olma kümesinde, bu kuyruk yöneticilerindeki aynı kümelenebilir konu nesnesini tanımlayarak yayınları aboneliklere yönlendirmek için birden çok kuyruk yöneticisi kullanılabilir. Bu, kullanılabilirliği ve iş yükü dengelemeyi geliştirmek için kullanılabilir. Aynı küme konusu nesnesi için ek bir konu anasistemi eklediğinizde, yayınların yeni konu anasistemi aracılığıyla yönetilmeye ne zaman başlayacağını denetlemek için **PUB** parametresini kullanabilirsiniz.

[Bir kuyruk yöneticisinin yayınlama/abone olma sıradüzeninden bağlantısının kesilmesi](#)

Yayınlama/abone olma sıradüzenindeki bir üst kuyruk yöneticisiyle alt kuyruk yöneticisinin bağlantısını kesin.

### **İlgili başvurular**

[Akışlar ve konular](#)

[PUBSUB 'I GÖRÜNTÜLE](#)

[İleti alışverişi yayınlama/abone ol](#)

## Bir kuyruk yöneticisinin yayınlama/abone olma sıradüzeninden bağlantısının kesilmesi

Yayınlama/abone olma sıradüzenindeki bir üst kuyruk yöneticisiyle alt kuyruk yöneticisinin bağlantısını kesin.

### Bu görev hakkında

Bir kuyruk yöneticisinin aracı sıradüzeniyle bağlantısını kesmek için **ALTER QMGR** komutunu kullanın. İsteddiğiniz zaman bir kuyruk yöneticisinin bağlantısını kesebilirsiniz.

Kuyruk yöneticileri arasındaki bağlantı çalışırken üst öğeyi güncelleme isteği gönderilir.

### Yordam

```
ALTER QMGR PARENT( ' ')
```

### Örnek

```
C:\Documents and Settings\Admin>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2.
  1 : alter qmgr parent(' ')
AMQ8005: IBM MQ queue manager changed.
  2 : display pubsub type(child)
AMQ8147: IBM MQ object not found.
display pubsub type(parent)
  3 : display pubsub type(parent)
AMQ8147: IBM MQ object not found.
```

### Sonraki adım

Artık gerekli olmayan akışları, kuyrukları ve el ile tanımlanan kanalları silebilirsiniz.

#### İlgili kavramlar

[Yayınlama ve abonelik kapsamlarını birleştirme](#)

IBM WebSphere MQ 7.0 'den itibaren yayın ve abonelik kapsamı, kuyruk yöneticileri arasındaki yayınların akışını belirlemek için bağımsız olarak çalışır.

[Yayınlama/abone olma ağlarında konu alanlarını birleştirme](#)

Bir kuyruk yöneticisinin konu alanını, bir yayınlama/abone olma kümesindeki ya da sıradüzenindeki diğer kuyruk yöneticileriyle birleştirin. Yayınlama/abone olma kümelerini birleştirin ve kümeleri sıradüzenlerle yayınlayın/abone olun.

#### İlgili görevler

[Yayınlama/abone olma kümesinin yapılandırılması](#)

Kuyruk yöneticisinde bir konu tanımlayın. Konuyu bir küme konusu yapmak için **CLUSTER** özelliğini ayarlayın. Bu konuya ilişkin yayınlar ve abonelikler için kullanılacak yönlendirmeyi seçmek üzere **CLROUTE** özelliğini ayarlayın.

[Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması](#)

Yöneltilen ya da doğrudan yöneltilen konu kümelerinde, bir kuyruk yöneticisinin devreden çıkarılması sırasında ya da bir küme kuyruk yöneticisinin başarısız olması ya da önemli bir süre için uygun olmaması nedeniyle bir küme konusu tanımlamasını taşımanız gerekebilir.

[Konu anasistemi tarafından yönlendirilen kümeye ek konu anasistemleri eklenmesi](#)

Bir konu anasistemi tarafından yönlendirilen yayınlama/abone olma kümesinde, bu kuyruk yöneticilerindeki aynı kümelenebilir konu nesnesini tanımlayarak yayınları aboneliklere yönlendirmek için birden çok kuyruk yöneticisi kullanılabilir. Bu, kullanılabilirliği ve iş yükü dengelemeyi geliştirmek için kullanılabilir. Aynı küme konusu nesnesi için ek bir konu anasistemi eklediğinizde, yayınların yeni konu anasistemi aracılığıyla yönetilmeye ne zaman başlayacağını denetlemek için **PUB** parametresini kullanabilirsiniz.

[Kuyruk yöneticisinin yayınlama/abone olma sıradüzenine bağlanması](#)

Alt kuyruk yöneticisini sıradüzendeki üst kuyruk yöneticisine bağlayabilirsiniz. Alt kuyruk yöneticisi zaten başka bir sıradüzenin ya da kümenin üyesiye, bu bağlantı sıradüzenleri birleştirir ya da kümeyi sıradüzeniyle birleştirir.

## ALW Birden çok kuruluşu yapılandırma

Aynı sistemde birden çok kuruluş kullanırken, kuruluşları ve kuyruk yöneticilerini yapılandırmanız gerekir.

### Bu görev hakkında

Bu bilgiler AIX, Linux, and Windows için geçerlidir.

### Yordam

- Kuruluşlarınızı yapılandırmak için aşağıdaki bağlantılardaki bilgileri kullanın:
  - [“Birincil kuruluşun değiştirilmesi” sayfa 451](#)
  - [“Bir kuyruk yöneticisinin kuruluşla ilişkilendirilmesi” sayfa 452](#)
  - [“Birden çok kuruluş ortamında uygulamaların bağlanması” sayfa 444](#)

## ALW Birden çok kuruluş ortamında uygulamaların bağlanması

AIX, Linux, and Windows sistemlerinde IBM MQ kitaplıkları yüklüyse, IBM MQ uygun kitaplıkları otomatik olarak kullanır; siz daha fazla işlem yapmanız gerekmeden bu kitaplıkları kullanır. IBM MQ , uygulamanın bağlı olduğu kuyruk yöneticisiyle ilişkili kuruluştan kitaplıkları kullanır.

Uygulamaların IBM MQ' e bağlanma şeklini açıklamak için aşağıdaki kavramlar kullanılır:

### Bağlantı kurma

Uygulama derlendiğinde, uygulama çalıştırıldığında yüklenen işlev dışı aktarılarını almak için uygulama IBM MQ kitaplıklarına bağlanır.

### Yükleniyor

Uygulama çalıştırıldığında, IBM MQ kitaplıkları bulunur ve yüklenir. Kitaplıkları bulmak için kullanılan özel mekanizma işletim sistemine ve uygulamanın oluşturulma şekline göre değişir. Birden çok kuruluş ortamında kitaplıkların nasıl bulunacağına ve yükleneceğine ilişkin ek bilgi için bkz. [“IBM MQ kitaplıkları yükleniyor” sayfa 445.](#)

### Bağlantı kuruluyor

Uygulama çalışan bir kuyruk yöneticisine bağlandığında (örneğin, bir MQCONN ya da MQCONNX çağrısı kullanarak), yüklenen IBM MQ kitaplıklarını kullanarak bağlanır.

Bir sunucu uygulaması bir kuyruk yöneticisine bağlandığında, yüklenen kitaplıkların kuyruk yöneticisiyle ilişkili kuruluştan gelmesi gerekir. Bir sistemde birden çok kuruluşta bu sınırlama, işletim sisteminin yüklenecek IBM MQ kitaplıklarını bulmak için kullandığı mekanizmayı seçerken yeni zorluklar ortaya çıkarır:

- Bir kuyruk yöneticisiyle ilişkili kuruluşu değiştirmek için **setmqm** komutu kullanıldığında, yüklenmesi gereken kitaplıklar değişir.
- Bir uygulama farklı kuruluşlara ait birden çok kuyruk yöneticilerine bağlandığında, birden çok kitaplık kümesinin yüklenmesi gerekir.

Ancak, IBM MQ, kitaplıklar bulunuyorsa ve yüklendiyse IBM MQ , daha fazla işlem yapmanız gerekmeden uygun kitaplıkları yükler ve kullanır. Uygulama bir kuyruk yöneticisine bağlandığında, IBM MQ kuyruk yöneticisinin ilişkilendirildiği kuruluştan kitaplıkları yükler.

Geçiş senaryoları ve birden çok kuruluşla uygulama bağlantısı AIX, Linux, and Windows üzerinde çoklu kuruluş kuyruk yöneticisi birlikte var olma içinde daha ayrıntılı olarak ele alınır.

IBM MQ kitaplıklarını yükleme hakkında daha fazla bilgi için bkz. [“IBM MQ kitaplıkları yükleniyor” sayfa 445.](#)



## Destek ve kısıtlamalar

Aşağıdaki IBM MQ kitaplıklarından herhangi biri bulunuyorsa ve yüklendiyse, ürün uygun kitaplıkları otomatik olarak yükleyebilir ve kullanabilir:

- C sunucusu kitaplıkları
- C++ sunucu kitaplıkları
- XA sunucusu kitaplıkları
- COBOL sunucu kitaplıkları
- COM + sunucu kitaplıkları
- Yönetilmeyen kipte .NET

IBM MQ , bağ tanımlama kipinde Java ve JMS uygulamaları için uygun kitaplıkları otomatik olarak yükler ve kullanır.

Birden çok kuruluş kullanan uygulamalar için bazı kısıtlamalar vardır. Daha fazla bilgi için [“Birden çok kuruluş kullanan uygulamalara ilişkin kısıtlamalar” sayfa 448](#) başlıklı konuya bakın.

### İlgili kavramlar

[“Birden çok kuruluş kullanan uygulamalara ilişkin kısıtlamalar” sayfa 448](#)

Birden çok kuruluş ortamında CICS sunucu kitaplıkları, hızlı yol bağlantıları, ileti tanıtıcıları ve çıkışları kullanılırken kısıtlamalar vardır.

[“IBM MQ kitaplıkları yükleniyor” sayfa 445](#)

IBM MQ kitaplıklarını nasıl yükleyeceğinize karar verirken, ortamınız, var olan uygulamalarınızı değiştirip değiştiremeyeceğiniz, birincil kuruluş isteyip istemediğiniz, IBM MQ ' un kurulu olup olmadığı ve IBM MQ konumunun değişip değişmeyeceği gibi çeşitli etkenleri göz önünde bulundurmanız gerekir.

### İlgili görevler

Birincil kuruluş seçilmesi

[“Birincil kuruluşun değiştirilmesi” sayfa 451](#)

Bir kuruluşu birincil kuruluş olarak ayarlamak ya da kaldırmak için **setmqinst** komutunu kullanabilirsiniz.

[“Bir kuyruk yöneticisinin kuruluşla ilişkilendirilmesi” sayfa 452](#)

Bir kuyruk yöneticisi oluşturduğunuzda, **crtmqm** komutunu yayınlayan kuruluşla otomatik olarak ilişkilendirilir. AIX, Linux, and Windows' da, **setmqm** komutunu kullanarak bir kuyruk yöneticisiyle ilişkili kuruluşu değiştirebilirsiniz.

## ALW IBM MQ kitaplıkları yükleniyor

IBM MQ kitaplıklarını nasıl yükleyeceğinize karar verirken, ortamınız, var olan uygulamalarınızı değiştirip değiştiremeyeceğiniz, birincil kuruluş isteyip istemediğiniz, IBM MQ ' un kurulu olup olmadığı ve IBM MQ konumunun değişip değişmeyeceği gibi çeşitli etkenleri göz önünde bulundurmanız gerekir.

IBM MQ kitaplıklarının nasıl yerleştirileceği ve yükleneceği kuruluş ortamınıza bağlıdır:

- AIX and Linux sistemlerinde, varsayılan konuma bir IBM MQ sürümünün kopyası kurulursa, var olan uygulamalar önceki sürümlerle aynı şekilde çalışmaya devam eder. Ancak, uygulamaların /usr/ libiçindeki simgesel bağlantılara gereksinimi varsa, birincil kuruluş olması için bir IBM MQ sürümü kuruluşu seçmeniz ya da simgesel bağlantıları el ile oluşturmanız gerekir.
- IBM MQ varsayılan olmayan bir konuma kurulduysa, doğru kitaplıkların yüklenmesi için var olan uygulamalarınızı değiştirmeniz gerekebilir.

IBM MQ kitaplıklarının nasıl bulunabileceği ve yüklenebileceği, var olan uygulamaların kitaplıkları yüklemek üzere nasıl ayarlandığına da bağlıdır. Kitaplıkların nasıl yüklenebileceğine ilişkin daha fazla bilgi için bkz.

[“İşletim sistemi kitaplığı yükleme mekanizmaları” sayfa 447.](#)




En iyi şekilde, işletim sistemi tarafından yüklenen IBM MQ kitaplığının, kuyruk yöneticisinin ilişkilendirildiği kitaplık olduğundan emin olmanız gerekir.

IBM MQ kitaplıklarını yükleme yöntemleri altyapıya göre değişir ve her yöntemin yararları ve dezavantajları vardır.

Çizelge 28. Kitaplıkları yüklemeye ilişkin seçeneklerin avantajları ve dezavantajları

Hizmet olarak sunulan	Seçenek	Avantajları	Dezavantajları
<p>Linux</p> <p>AIX</p> <p>AIX and Linux sistemleri</p>	<p>Uygulamanın gömülü yürütme ortamı arama yolunu (RPath) ayarlayın ya da değiştirin.</p> <p>Bu seçenek, uygulamayı yeniden derleyip bağlamanızı gerektirir. Uygulamaları derleme ve bağlama hakkında daha fazla bilgi için bkz. <a href="#">Prosedür uygulaması oluşturma</a>.</p>	<ul style="list-style-type: none"> <li>Değişikliğin kapsamı açık.</li> </ul>	<ul style="list-style-type: none"> <li>Uygulamayı yeniden derleyebilmeli ve bağlayabilmelisiniz.</li> <li>IBM MQ konumu değişirse, RPath ögesini değiştirmeniz gerekir.</li> </ul>
<p>AIX and Linux sistemleri</p>	<p><code>LD_LIBRARY_PATH</code> ortam değişkenini, <code>-k</code> ya da <code>-l</code> seçeneğiyle <code>setmqenvya</code> da <code>crtmqenvkomutunu</code> kullanarak ayarlayın. (</p> <p>AIX işletim sistemlerinde bu ortam değişkeni <code>LIBPATH</code> değişkenidir.</p>	<ul style="list-style-type: none"> <li>Var olan uygulamalarda değişiklik yapılması gerekmez.</li> <li>Bir uygulamadaki gömülü RPaths 'i geçersiz kılar.</li> <li>IBM MQ konumu değişirse, değişkeni kolayca değiştirebilirsiniz.</li> </ul>	<ul style="list-style-type: none"> <li><code>setuid</code> ve <code>setgid</code> uygulamaları ya da başka şekillerde oluşturulan uygulamalar, güvenlik nedeniyle <code>LD_LIBRARY_PATH</code> değerini yoksayabilir.</li> <li>Ortama özgü olduğundan, uygulamanın çalıştırıldığı her ortamda ayarlanmalıdır.</li> <li><code>LD_LIBRARY_PATH</code> değişkenine dayalı diğer uygulamalar üzerinde olası bir etki.</li> <li>Linux: Uygulamayı oluşturmak için kullanılan derleyici, <code>LD_LIBRARY_PATH</code> kullanımı devre dışı bırakabilir. Daha fazla bilgi için bkz. <a href="#">Linux için Runtime bağlantısına ilişkin önemli noktalar</a>.</li> </ul>
<p>Windows</p> <p>Windows sistemleri</p>	<p><code>setmqenvya</code> da <code>crtmqenvkomutunu</code> kullanarak <code>PATH</code> değişkenini ayarlayın.</p>	<ul style="list-style-type: none"> <li>Var olan uygulamalar için değişiklik gerekmez.</li> <li>IBM MQ konumu değişirse, değişkeni kolayca değiştirebilirsiniz.</li> </ul>	<ul style="list-style-type: none"> <li>Ortama özgü olduğundan, uygulamanın çalıştırıldığı her ortamda ayarlanmalıdır.</li> <li>Diğer uygulamalar üzerinde olası etki.</li> </ul>

Çizelge 28. Kitaplıkları yüklemeye ilişkin seçeneklerin avantajları ve dezavantajları (devamı var)

Hizmet olarak sunulan	Seçenek	Avantajları	Dezavantajları
 AIX, Linux, and Windows sistemleri	Birincil kuruluğu IBM MQya da daha sonraki bir kuruluşa ayarlayın. Bkz. “Birincil kuruluğun değiştirilmesi” sayfa 451.  Birincil kuruluşa ilgili daha fazla bilgi için bkz. <a href="#">Birincil kuruluş seçilmesi</a> .	<ul style="list-style-type: none"><li>Var olan uygulamalar için değişiklik gerekmez.</li><li>IBM MQ konumu değişirse, birincil kuruluğu kolayca değiştirebilirsiniz.</li><li>Önceki IBM MQsürümlerine benzer davranış verir.</li></ul>	<ul style="list-style-type: none"><li> Linux</li><li> AIX and Linux: /usr/lib varsayılan arama yolunda değilse çalışmaz.</li></ul>

## Linux için kitaplık yüklemeyle ilgili önemli noktalar

### Linux

Bazı gcc sürümleri (örneğin, sürüm 3.2.x) kullanılarak derlenen uygulamalarda, `LD_LIBRARY_PATH` ortam değişkeni kullanılarak geçersiz kılınamayan gömülü bir RPath olabilir. `readelf -d applicationName` komutunu kullanarak bir uygulamanın etkilenip etkilenmediğini belirleyebilirsiniz. RPATH simgesi varsa ve RUNPATH simgesi yoksa, RPath geçersiz kılınmaz.

## İşletim sistemi kitaplığı yükleme mekanizmaları

Windows sistemlerinde, kitaplıkları bulmak için birden çok dizin aranır:

- Uygulamanın yüklendiği dizin.
- Yürürlükteki dizin.
- `PATH` ortam değişkenindeki dizinler (hem genel `PATH` değişkeni hem de yürürlükteki kullanıcının `PATH` değişkeni).

### Linux

### AIX

AIX and Linux sistemlerinde, yüklenecek kitaplıkları bulmak için kullanılan birkaç yöntem vardır:

- `LD_KITAPLIĞI_YOLU` ortam değişkeninin kullanılması ( AIXüzerinde `LIBPATH` da). Bu değişken ayarlanırsa, gerekli IBM MQ kitaplıkları için aranan bir dizin kümesini tanımlar. Bu dizinlerde herhangi bir kitaplık bulunursa, diğer yöntemler kullanılarak bulunabilecek kitaplıklar tercih edilir.
- Gömülü arama yolu (RPath) kullanılıyor. Uygulama, IBM MQ kitaplıklarını aramak için bir dizin kümesi içerebilir. `LD_LIBRARY_PATH` ayarlanmazsa ya da gerekli kitaplıklar değişken kullanılarak bulunamazsa, kitaplıklar için RPath aranır. Var olan uygulamalarınız bir RPath kullanıyorsa, ancak uygulamayı yeniden derleyip bağlayamıyorsanız, IBM MQ ' i varsayılan konuma kurmanız ya da kitaplıkları bulmak için başka bir yöntem kullanmanız gerekir.
- Varsayılan kitaplık yolu kullanılıyor. `LD_LIBRARY_PATH` değişkeni ve RPath yerleri arandıktan sonra IBM MQ kitaplıkları bulunamazsa, varsayılan kitaplık yolu aranır. Genellikle bu yol `/usr/lib` ya da `/usr/lib64` içerir. Varsayılan kitaplık yolu arandıktan sonra kitaplıklar bulunamazsa, eksik bağımlılıklar nedeniyle uygulama başlatılamaz.

Uygulamalarınızın gömülü bir arama yolu olup olmadığını öğrenmek için işletim sistemi düzeneklerini kullanabilirsiniz. Örneğin:

-  AIX: `dump`
-  Linux: `readelf`

## İlgili kavramlar

“Birden çok kuruluş kullanan uygulamalara ilişkin kısıtlamalar” sayfa 448

Birden çok kuruluş ortamında CICS sunucu kitaplıkları, hızlı yol bağlantıları, ileti tanıtıcıları ve çıkışları kullanılırken kısıtlamalar vardır.

[“Birden çok kuruluş ortamında uygulamaların bağlanması” sayfa 444](#)

AIX, Linux, and Windows sistemlerinde IBM MQ kitaplıkları yüklüyse, IBM MQ uygun kitaplıkları otomatik olarak kullanır; siz daha fazla işlem yapmanız gerekmeden bu kitaplıkları kullanır. IBM MQ , uygulamanın bağlandığı kuyruk yöneticisiyle ilişkili kuruluştan kitaplıkları kullanır.

### **İlgili görevler**

[Birincil kuruluş seçilmesi](#)

[“Birincil kuruluşun değiştirilmesi” sayfa 451](#)

Bir kuruluşu birincil kuruluş olarak ayarlamak ya da kaldırmak için **setmqinst** komutunu kullanabilirsiniz.

[“Bir kuyruk yöneticisinin kuruluşla ilişkilendirilmesi” sayfa 452](#)

Bir kuyruk yöneticisi oluşturduğunuzda, **crtmqm** komutunu yayınlayan kuruluşla otomatik olarak ilişkilendirilir. AIX, Linux, and Windows' da, **setmqm** komutunu kullanarak bir kuyruk yöneticisiyle ilişkili kuruluşu değiştirebilirsiniz.

## **ALW Birden çok kuruluş kullanan uygulamalara ilişkin kısıtlamalar**

Birden çok kuruluş ortamında CICS sunucu kitaplıkları, hızlı yol bağlantıları, ileti tanıtıcıları ve çıkışları kullanılırken kısıtlamalar vardır.

### **CICS sunucu kitaplıkları**

CICS sunucu kitaplıklarını kullanıyorsanız, IBM MQ sizin için doğru kitaplık düzeyini otomatik olarak seçmez. Uygulamalarınızı derlemeli ve uygulamanın bağlandığı kuyruk yöneticisi için uygun kitaplık düzeyine bağlamalısınız. Daha fazla bilgi için bkz. [Building libraries for TXSeries for Multiplatforms version 5](#).

### **İleti tanıtıcıları**

MQHC\_UNASSOCIATED\_HCONN özel değerini kullanan ileti tanıtıcıları, bir süreçte yüklenen ilk kuruluşla sınırlıdır. İleti tanıtıcısı belirli bir kuruluş tarafından kullanılamıyorsa, neden kodu MQRC\_HMSG\_NOT\_AVAILABLE döndürülür.

Bu kısıtlama ileti özelliklerini etkiler. Bir kuruluştaki kuyruk yöneticisinden ileti özelliklerini almak ve bunları farklı bir kuruluşta kuyruk yöneticisine koymak için ileti tanıtıcılarını kullanamazsınız. İleti tanıtıcılarıyla ilgili ek bilgi için [MQCRTMH-İleti tanıtıcısı yaratılması](#) başlıklı konuya bakın.

### **Çıkışlar**

Birden çok kuruluş ortamında, var olan çıkışlar IBM MQ kuruluşlarıyla kullanılmak üzere güncellenmelidir. **crtmqcvx** komutu kullanılarak oluşturulan veri dönüştürme çıkışları, güncellenen komut kullanılarak yeniden oluşturulmalıdır.

Tüm çıkışlar MQIEP yapısı kullanılarak yazılmalı, IBM MQ kitaplıklarını bulmak için gömülü RPATH kullanılamaz ve IBM MQ kitaplıklarına bağlanamaz. Daha fazla bilgi için bkz. [AIX, Linux, and Windows](#) üzerinde yazma çıkışları ve kurulabilir hizmetler.

### **Hızlı yol**

Birden çok kuruluşu olan bir sunucuda, IBM MQ ile hızlı yol bağlantısı kullanan uygulamalar aşağıdaki kurallara uymalıdır:

1. Kuyruk yöneticisi, uygulamanın IBM MQ yürütme zamanı kitaplıklarını yüklediği kuruluşla aynı kuruluşla ilişkilendirilmelidir. Uygulama, farklı bir kuruluşla ilişkilendirilmiş bir kuyruk yöneticisine hızlı yol bağlantısı kullanmamalıdır. Bağlantı kurma girişimi bir hatayla ve neden kodu MQRC\_INSTALLATION\_MISMATCH ile sonuçlanır.

2. Uygulamanın IBM MQ çalıştırma zamanı kitaplıklarını yüklediği kuruluşla aynı kuruluşla ilişkili bir kuyruk yöneticisine hızlı olmayan bir yol bağlanması, aşağıdaki koşullardan biri doğru olmadığı sürece uygulamanın hızlı yola bağlanmasını önler:
  - Uygulama, aynı kuruluşla ilişkilendirilmiş bir kuyruk yöneticisine ilk bağlantısını hızlı yol bağlantısı olarak yapar.
  - AMQ\_SINGLE\_INSTALLATION ortam değişkeni ayarlı.
3. Hızlı olmayan bir yolun IBM MQ kuruluşuyla ilişkilendirilmiş bir kuyruk yöneticisine bağlanması, bir uygulamanın hızlı yola bağlanıp bağlanamayacağını etkilemez.

AMQ\_SINGLE\_INSTALLATION kümesiyle, bir kuyruk yöneticisiyle hızlı yol bağlantısı kurabilirsiniz. Aksi takdirde hemen hemen aynı kısıtlamalar geçerlidir:

- Kuruluş, IBM MQ yürütme zamanı kitaplıklarının yüklendiği kuruluşla aynı olmalıdır.
- Aynı işlemdeki her bağlantı aynı kuruluşla kurulmalıdır. Farklı bir kuruluşla ilişkilendirilmiş bir kuyruk yöneticisine bağlanma girişiminde bulunursanız, bağlantı MQRC\_INSTALLATION\_MISMATCHneden koduyla başarısız olur. AMQ\_SINGLE\_INSTALLATION ayarı ile bu sınırlamanın yalnızca hızlı yol bağlantıları için değil, tüm bağlantılar için de geçerli olduğunu unutmayın.
- Hızlı yol bağlantılarıyla yalnızca bir kuyruk yöneticisine bağlanın.

### İlgili başvurular

[MQCONN-Connect kuyruk yöneticisi \(genişletilmiş\)](#)

[MQIEP yapısı](#)

[2583 \(0A17\) \(RC2583\): MQRC\\_INSTALLATION\\_MISMATCH](#)

[2587 \(0A1B\) \(RC2587\): MQRC\\_HMSG\\_NOT\\_AVAILABLE](#)

[2590 \(0A1E\) \(RC2590\): MQRC\\_FASTPATH\\_NOT\\_ALLOWED](#)

## Birden çok kuruluş ortamında .NET uygulamalarının bağlanması

Varsayılan olarak, uygulamalar birincil kuruluşta .NET düzeneklerini kullanır. Birincil kuruluş yoksa ya da birincil kuruluş düzeneklerini kullanmak istemiyorsanız, uygulama yapılandırma dosyasını ya da *DEVPATH* ortam değişkenini güncellemeniz gerekir.

Sistemde birincil kuruluş varsa, bu kuruluşun .NET düzenekleri ve ilke dosyaları, genel düzenek önbelleğine (GAC) kaydedilir. Diğer tüm kuruluşlara ilişkin .NET düzenekleri her kuruluşun kuruluş yolunda bulunabilir, ancak düzenekler GAC 'ye kayıtlı değildir. Bu nedenle, varsayılan olarak uygulamalar birincil kuruluşta .NET yapıbirimleri kullanılarak çalışır. Aşağıdaki durumlardan herhangi biri doğruysa uygulama yapılandırma dosyasını güncellemeniz gerekir:

- Birincil kuruluşunuz yok.
- Uygulamanın birincil kuruluş düzeneklerini kullanmasını istemezsiniz.
- Birincil kuruluş, uygulamanın derlendiği sürümden daha düşük bir IBM MQ sürümüdür.

Uygulama yapılandırma dosyasının nasıl güncelleneceğine ilişkin bilgi için bkz. "[Uygulama yapılandırma dosyasını kullanarak .NET uygulamalarının bağlanması](#)" sayfa 449.

Aşağıdaki durum doğruysa, *DEVPATH* ortam değişkenini güncellemeniz gerekir:

- Uygulamanızın birincil olmayan bir kuruluşta gelen yapıbirimleri kullanmasını istiyorsunuz, ancak birincil kuruluş, birincil olmayan kuruluşla aynı sürümde.

*DEVPATH* değişkeninin güncellenmesiyle ilgili daha fazla bilgi için bkz. "[DEVPATH kullanarak .NET uygulamalarının bağlanması](#)" sayfa 450.

## Uygulama yapılandırma dosyasını kullanarak .NET uygulamalarının bağlanması

Uygulama yapılandırma dosyası içinde, uygulamaları birincil kuruluşta olmayan yapıbirimleri kullanacak şekilde yeniden yönlendirmek için çeşitli etiketler ayarlamanız gerekir.

Aşağıdaki tabloda, .NET uygulamalarının belirli derlemleri kullanarak bağlanmasına izin vermek için uygulama yapılandırma dosyasında yapılması gereken belirli değişiklikler gösterilmektedir:

Çizelge 29. Uygulamaların belirli düzenekleri kullanacak şekilde yapılandırılması		
	Daha önceki bir IBM MQ sürümüyle derlenen uygulamalar	Daha sonraki bir IBM MQ sürümüyle derlenen uygulamalar
Bir uygulamayı daha sonraki bir sürüm IBM MQ birincil kuruluşuyla çalıştırmak için. (GAC ' deki sonraki sürüm düzenekleri):	Herhangi bir değişiklik gerekmez	Herhangi bir değişiklik gerekmez
Bir uygulamayı daha önceki bir sürüm IBM MQ birincil kuruluşuyla çalıştırmak için. (GAC ' de önceki sürüm derlemleri):	Herhangi bir değişiklik gerekmez	Uygulama yapılandırma dosyasında: <ul style="list-style-type: none"> <li>GAC ' de bulunan düzeneklerin önceki sürümünün kullanımını belirtmek için <i>bindingRedirect</i> etiketini kullanın.</li> </ul>
Bir uygulamayı daha sonraki bir IBM MQ birincil olmayan kuruluş sürümüyle çalıştırmak için. (kuruluş klasöründeki sonraki sürüm yapıbirimleri):	Uygulama yapılandırma dosyasında: <ul style="list-style-type: none"> <li>Sonraki sürüm düzeneklerinin yerini göstermek için <i>codebase</i> etiketini kullanın.</li> <li>Sonraki sürüm derlemelerinin kullanımını belirtmek için <i>bindingRedirect</i> etiketini kullanın.</li> </ul>	Uygulama yapılandırma dosyasında: <ul style="list-style-type: none"> <li>Sonraki sürüm düzeneklerinin yerini göstermek için <i>codebase</i> etiketini kullanın.</li> </ul>
Birincil olmayan IBM MQ kuruluşunun önceki bir sürümüyle bir uygulamayı çalıştırmak için. (kuruluş klasöründeki önceki sürüm yapıbirimleri):	Uygulama yapılandırma dosyasında: <ul style="list-style-type: none"> <li>Önceki sürüm yapıbirimleri konumunu göstermek için <i>codebase</i> etiketini kullanın.</li> <li><i>publisherpolicy Apply=no</i> etiketini dahil et</li> </ul>	Uygulama yapılandırma dosyasında: <ul style="list-style-type: none"> <li>Önceki sürüm yapıbirimleri konumunu göstermek için <i>codebase</i> etiketini kullanın.</li> <li>Önceki sürüm düzeneklerinin kullanımını belirtmek için <i>bindingRedirect</i> etiketini kullanın.</li> <li><i>publisherpolicy Apply=no</i> etiketini dahil et</li> </ul>

Örnek bir uygulama yapılandırma dosyası *NonPrimaryRedirect.config*, *MQ\_INSTALLATION\_PATH\tools\dotnet\samples\base* klasöründe teslim edilir. Bu dosya, birincil olmayan herhangi bir kuruluşun IBM MQ kuruluş yoluyla değiştirilebilir. Dosya, *linkedConfiguration* etiketi kullanılarak doğrudan diğer yapılandırma dosyalarına da eklenebilir. *nmqsget.exe.config* ve *nmqspu.exe.config* için örnekler sağlanır. Her iki örnek de *linkedConfiguration* etiketini kullanır ve *NonPrimaryRedirect.config* dosyasını içerir.

## DEVPATH kullanarak .NET uygulamalarının bağlanması

Yapıbirimleri *DEVPATH* ortam değişkenini kullanarak bulabilirsiniz. *DEVPATH* değişkeniyle belirtilen yapıbirimleri, GAC ' deki tüm yapıbirimleri tercih etmek için kullanılır. Bu değişkenin ne zaman kullanılacağı hakkında daha fazla bilgi için *DEVPATH* ile ilgili Microsoft belgelerine bakın.

Yapıbirimleri *DEV* ortam değişkenini kullanarak bulmak için, *DEV* değişkenini kullanmak istediğiniz yapıbirimleri içeren klasöre ayarlamamız gerekir. Daha sonra, uygulama yapılandırma dosyasını güncelleme ve aşağıdaki çalıştırma zamanı yapılandırma bilgilerini eklememiz gerekir:

```
<configuration>
<runtime>
<developmentMode developerInstallation="true" />
</runtime>
</configuration>
```

### İlgili kavramlar

“Birden çok kuruluş ortamında uygulamaların bağlanması” sayfa 444

AIX, Linux, and Windows sistemlerinde IBM MQ kitaplıkları yüklüyse, IBM MQ uygun kitaplıkları otomatik olarak kullanır; siz daha fazla işlem yapmanız gerekmeden bu kitaplıkları kullanır. IBM MQ , uygulamanın bağlandığı kuyruk yöneticisiyle ilişkili kuruluştan kitaplıkları kullanır.

Birden çok kuruluş

### İlgili görevler

Birincil kuruluş seçilmesi

kullanma.NET

## ALW Birincil kuruluşun değiştirilmesi

Bir kuruluşu birincil kuruluş olarak ayarlamak ya da kaldırmak için **setmqinst** komutunu kullanabilirsiniz.

### Bu görev hakkında

Bu görev AIX, Linux, and Windows için geçerlidir.

Birincil kuruluş, sistem genelinde gerekli konumların başvurduğu kuruluştur. Birincil kuruluşla ilişkin ek bilgi ve birincil kuruluşunuzu seçmeye ilişkin dikkat edilmesi gereken noktalar için [Birincil kuruluşun seçilmesibaşlıklı konuya](#) bakın.

**Windows** Windows üzerindeki kuruluş işlemi sırasında, kuruluşun birincil kuruluş olacağını belirtebilirsiniz.

**Linux** **AIX** AIX and Linux sistemlerinde, kuruluşu birincil kuruluş olarak ayarlamak için kuruluştan sonra bir **setmqinst** komutu vermeniz gerekir.

### Yordam

- Bir kuruluşu birincil kuruluş olarak ayarlamak için aşağıdaki adımları tamamlayın:
  - a) Aşağıdaki komutu girerek bir kuruluşun birincil kuruluş olup olmadığını denetleyin:

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

Burada *MQ\_INSTALLATION\_PATH* , bir IBM MQ kuruluşunun kuruluş yoludur.

- b) Var olan bir IBM MQ kuruluşu birincil kuruluş olarak ayarlanırsa, sonraki adıma geçmeden önce [ayarı kaldırın](#) .

- c) Uygun yetkiyle oturum açtığınızdan emin olun:

– **Linux** **AIX** AIX and Linux üzerinde kök olarak.

– **Windows** Windows sistemlerinde Administrators (Yöneticiler) grubunun bir üyesi olarak.

- d) Aşağıdaki komutlardan birini girin:

– Birincil kuruluşu, birincil kuruluş olmasını istediğiniz kuruluş yolunu kullanarak ayarlamak için:

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -p MQ_INSTALLATION_PATH
```

- Birincil kuruluşu, birincil kuruluş olmasını istediğiniz kuruluşun adını kullanarak ayarlamak için:

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -n installationName
```

#### e) **Windows**

Windows sistemlerinde sistemi yeniden başlatın.

- Bir kuruluşu birincil kuruluş olarak kaldırmak için aşağıdaki adımları tamamlayın:
  - a) Aşağıdaki komutu girerek hangi kuruluşun birincil kuruluş olduğunu denetleyin:

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

Burada `MQ_INSTALLATION_PATH` , bir IBM MQ kuruluşunun kuruluş yoludur.

- b) Uygun yetkiyle oturum açtığınızdan emin olun:

- **Linux** **AIX** AIX and Linux üzerinde kök olarak.

- **Windows** Windows sistemlerinde Administrators (Yöneticiler) grubunun bir üyesi olarak.

- Aşağıdaki komutlardan birini girin:
  - Kuruluş yolunu kullanarak birincil kuruluşun ayarını kaldırmak için artık birincil kuruluş olmak istemeyiniz:

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -p MQ_INSTALLATION_PATH
```

- Kuruluşun adını kullanarak birincil kuruluşun ayarını kaldırmak için artık birincil kuruluş olmak istemeyiniz:

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -n installationName
```

### İlgili görevler

[Birincil kuruluş kaldırma, yükseltme ve sürdürme](#)

[Kuruluş adı seçilmesi](#)

### İlgili başvurular

[Yalnızca Windows üzerinde birincil kuruluşla kullanılabilen özellikler](#)

[AIX and Linux üzerinde birincil kuruluş dış kitaplık ve denetim komutu bağlantıları](#)

[setmqinst](#)

## ALW

## Bir kuyruk yöneticisinin kuruluşla ilişkilendirilmesi

Bir kuyruk yöneticisi oluşturduğunuzda, **crtmqm** komutunu yayınlayan kuruluşla otomatik olarak ilişkilendirilir. AIX, Linux, and Windows' da, **setmqm** komutunu kullanarak bir kuyruk yöneticisiyle ilişkili kuruluşu değiştirebilirsiniz.

### Bu görev hakkında

Kuyruk yöneticisinin yalnızca o kuruluştan gelen komutlarla yönetilebilmesi için, kuyruk yöneticisinin ilişkilendirildiği kuruluş. Üç temel kural dışı durum vardır:

- **setmqm** , kuyruk yöneticisiyle ilişkili kuruluşu değiştirir. Bu komut, kuyruk yöneticisinin ilişkilendirildiği kuruluştan değil, kuyruk yöneticisiyle ilişkilendirmek istediğiniz kuruluştan verilmelidir. **setmqm** komutuyla belirtilen kuruluş adı, komutun verildiği kuruluşla eşleşmelidir.
- **strmqm** , kuyruk yöneticisiyle ilişkili kuruluştan yayınlanmalı.



- **dspmqr** , yalnızca **dspmqr** komutuyla aynı kuruluşla ilişkili kuyruk yöneticileriyle değil, bir sistemdeki tüm kuyruk yöneticileriyle ilgili bilgileri görüntüler. **dspmqr -o installation** komutu, hangi kuyruk yöneticilerinin hangi kuruluşlarla ilişkilendirildiği hakkında bilgi görüntüler.

HA ortamları için **addmqinf** komutu, kuyruk yöneticisini **addmqinf** komutunun verildiği kuruluşla otomatik olarak ilişkilendirir. **strmqm** komutu **addmqinf** komutuyla aynı kuruluştan verildiği sürece, başka bir ayar gerekmez. Kuyruk yöneticisini farklı bir kuruluş kullanarak başlatmak için önce **setmqm** komutunu kullanarak ilişkili kuruluşu değiştirmeniz gerekir.

Bir kuyruk yöneticisini bir kuruluşla ilişkilendirmek istediğinizde, **setmqm** komutunu aşağıdaki şekillerde kullanabilirsiniz:

- Tek tek kuyruk yöneticilerinin IBM MQ' un eşdeğer sürümleri arasında taşınması. Örneğin, bir kuyruk yöneticisinin testten üretim sistemine taşınması.
- Tek tek kuyruk yöneticilerinin daha eski bir IBM MQ sürümünden daha yeni bir IBM MQ sürümüne geçirilmesi. Kuyruk yöneticilerini sürümler arasında geçirmenin bilmeniz gereken çeşitli etkileri vardır. Geçiş hakkında daha fazla bilgi için bkz. [Bakım ve geçiş](#).

## Yordam

1. Kuyruk yöneticisiyle ilişkilendirilmiş kuruluştan **endmqm** komutunu kullanarak kuyruk yöneticisini durdurun.
2. Bu kuruluştan **setmqm** komutunu kullanarak kuyruk yöneticisini başka bir kuruluşla ilişkilendirin. Örneğin, QMB kuyruk yöneticisini Installation2 adlı bir kuruluşla ilişkilendirilecek şekilde ayarlamak için Installation2: komutunu girin.

```
MQ_INSTALLATION_PATH/bin/setmqm -m QMB -n Installation2
```

Burada *MQ\_INSTALLATION\_PATH* , Installation2 ' in kurulu olduğu yoldur.

3. Kuyruk yöneticisiyle ilişkilendirilmiş kuruluştan **strmqm** komutunu kullanarak kuyruk yöneticisini başlatın.

Bu komut, gerekli kuyruk yöneticisi geçişini gerçekleştirir ve kuyruk yöneticisinin kullanıma hazır olmasına neden olur.

## Sonraki adım

Bir kuyruk yöneticisinin ilişkilendirildiği kuruluş silindiye ya da kuyruk yöneticisi durum bilgisi kullanılamıyorsa, **setmqm** komutu kuyruk yöneticisini başka bir kuruluşla ilişkilendiremez. Bu durumda aşağıdaki işlemleri gerçekleştirin:

1. Sisteminizdeki diğer kuruluşları görmek için **dspmqrinst** komutunu kullanın.
2. Başka bir kuruluş belirtmek için **mq.s.ini** içindeki `QueueManager` kısmına ilişkin `InstallationName` alanını el ile değiştirin.
3. Kuyruk yöneticisini silmek için o kuruluştan **dlmqm** komutunu kullanın.

## İlgili kavramlar

[“Sistemdeki IBM MQ kuruluşlarını bulma” sayfa 454](#)

Bir sistemde birden çok IBM MQ kuruluşu varsa, hangi sürümlerin kurulu olduğunu ve bunların nerede olduğunu denetleyebilirsiniz.

[“IBM MQ yapılandırma dosyası, mq.s.ini” sayfa 85](#)

IBM MQ yapılandırma kütüğü ( `mq.s.ini` ), düğümdeki tüm kuyruk yöneticileriyle ilgili bilgileri içerir. Kuruluş sırasında otomatik olarak yaratılır.

## İlgili görevler

[Birincil kuruluş seçilmesi](#)

## İlgili başvurular

[addmqinf](#)

[dismq](#)  
[mqinst](#)  
[sonmqm](#)  
[setmqm](#)  
[strmqm](#)

## ALW Sistemdeki IBM MQ kuruluşlarını bulma

Bir sistemde birden çok IBM MQ kuruluşu varsa, hangi sürümlerin kurulu olduğunu ve bunların nerede olduğunu denetleyebilirsiniz.

Sisteminizdeki IBM MQ kuruluşlarını bulmak için aşağıdaki yöntemleri kullanabilirsiniz:

- IBM MQ ' in kurulduğu yeri sorgulamak için platform kuruluş araçlarını kullanın. Daha sonra, IBM MQ kuruluşundan **dspmqr** komutunu kullanın. Aşağıdaki komutlar, IBM MQ ' in kurulduğu yeri sorgulamak için kullanabileceğiniz komutlara örnektir:

- **AIX** AIX sistemlerinde **lspp** komutunu kullanabilirsiniz:

```
lspp -R ALL -l mqm.base.runtime
```

- **Linux** Linux sistemlerinde **rpm** komutunu kullanabilirsiniz:

```
rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}\t%{INSTPREFIXES}\n" | grep MQSeriesRuntime
```

- **Windows** Windows sistemlerinde **wmic** komutunu kullanabilirsiniz. Bu komut wmic istemcisini kurabilir:

```
wmic product where "(Name like '%MQ%') AND (not Name like '%bitSupport')" get Name, Version, InstallLocation
```

- **Linux** **AIX** AIX and Linux sistemlerinde, IBM MQ ' in nereye kurulduğunu öğrenmek için aşağıdaki komutu verin:

```
cat /etc/opt/mqm/mqinst.ini
```

Daha sonra, IBM MQ kuruluşundan **dspmqr** komutunu kullanın.

- **Windows** Sistemdeki kuruluşlara ilişkin ayrıntıları görüntülemek için, 32 bit Windows üzerinde aşağıdaki komutu verin:

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation" /s
```

- **Windows** 64 bit Windows üzerinde şu komutu verin:

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE Wow6432Node\IBM\WebSphere MQ\Installation" /s
```

### İlgili başvurular

[mqver](#)  
[mqinst](#)  
[Birden çok kuruluş](#)

# Yüksek kullanılabilirliğin, kurtarmanın ve yeniden başlatmanın yapılandırılması

Bir kuyruk yöneticisinin başarısız olması durumunda ve sunucu ya da depolama hatasından sonraki iletileri kurtararak uygulamalarınızı yüksek kullanılabilirlikli kılabilirsiniz.

## Bu görev hakkında

**z/OS** z/OS üzerinde, platformda yüksek düzeyde kullanılabilirlik yerleşik hale getirmiştir. Bkz. [Paylaşılan kuyruklar ve kuyruk paylaşım grupları](#).

**Multi** 'Çoklu platformlar' ta, istemci yeniden bağlantısını kullanarak bir kuyruk yöneticisi grubu arasında otomatik olarak ya da bir kuyruk yöneticisi hatasından sonra çok eşgörsümlü bir kuyruk yöneticisinin yeni etkin eşgörsümlüme geçiş yaparak istemci uygulaması kullanılabilirliğini artırabilirsiniz. Otomatik istemci yeniden bağlantısı IBM MQ classes for Javatarafından desteklenmez. Çok eşgörsümlü bir kuyruk yöneticisi, birden çok sunucuda tek bir kuyruk yöneticisi olarak çalışacak şekilde yapılandırıldı. Sunucu uygulamalarını bu kuyruk yöneticisine konuşturabilirsiniz. Etkin yönetim ortamını çalıştıran sunucu başarısız olursa, yürütme otomatik olarak farklı bir sunucuda aynı kuyruk yöneticisinin yedek yönetim ortamına değiştirilir. Sunucu uygulamalarını kuyruk yöneticisi hizmetleri olarak çalışacak şekilde yapılandırırsanız, bunlar, bir yedek yönetim ortamı etkin olarak çalışan kuyruk yöneticisi yönetim ortamı olduğunda yeniden başlatılır.

Multiplatforms üzerinde sunucu uygulaması kullanılabilirliğini artırmanın başka bir yolu da, sunucu uygulamalarının bir kuyruk yöneticisi kümesindeki birden çok bilgisayara konuşturulmasıdır. IBM WebSphere MQ 7.1 ' den başlayarak, sorunlar çözülmüncedey kadar sorunlara neden olan küme hatası kurtarma işlemlerini yeniden çalıştırır. Bkz. [z/OS dışındaki sunucularda küme hatası kurtarmada yapılan değişiklikler](#). IBM MQ for Multiplatforms ürününü, aşağıdakiler gibi platforma özgü bir kümeleme çözümünün bir parçası olarak da yapılandırabilirsiniz:

- Microsoft Küme Sunucusu
- **IBM i** IBM i üzerinde HA kümeleri
- **Linux** **AIX** PowerHA for AIX (önceki adıyla AIX üzerinde HACMP ) ve diğer UNIX and Linux kümeleme çözümleri

**Linux** Linux sistemlerinde, yüksek düzeyde kullanılabilirlik ya da olağanüstü durumdan kurtarma çözümleri uygulamak için eşlenmiş veri kuyruğu yöneticilerini (RDIMM ' ler) yapılandırabilirsiniz. Yüksek kullanılabilirlik için, üç Linux sunucusundan oluşan bir gruptaki her düğümde aynı kuyruk yöneticisinin eşgörsümleri yapılandırılır. Bu üç eşgörsümlüden biri etkin eşgörsümlüdür. Etkin kuyruk yöneticisindeki veriler zamanuyumlu olarak diğer iki örneğe eşlenir; bu nedenle, bir hata durumunda bu eşgörsümlerden biri devralabilir. Olağanüstü durumdan kurtarma için, bir kuyruk yöneticisi, bir yerdeki birincil düğümde çalışır ve bu kuyruk yöneticisinin ikincil eşgörsümlü farklı bir yerdeki bir kurtarma düğümünde bulunur. Veriler, birincil eşgörsümlü ile ikincil eşgörsümlü arasında eşlenir ve birincil düğüm bir nedenle kaybolursa, ikincil eşgörsümlü birincil örneğe dönüştürülebilir ve başlatılabilir.

**CP4I** Yerel HA, konteynerleri hedefleyen yüksek kullanılabilirlikli bir çözümdür. Yerel HA, farklı düğümlerde çalışan bir kuyruk yöneticisinin üç eşgörsümlümlü güncel tutmak için günlük eşlemesini kullanır. Bir eşgörsümlü aynı anda etkindir ve iletileri işler. Etkin kuyruk yöneticisi, günlük güncellemelerini diğer iki yönetim ortamı için göndererek bunları güncel tutmasını sağlar. Etkin eşgörsümlü başarısız olursa, eşleme örneklerinden biri otomatik olarak etkin rolü devralır.

**MQ Appliance** Yüksek düzeyde kullanılabilirlik ya da olağanüstü durumdan kurtarma çözümü için bir diğer seçenek de bir çift IBM MQ aygıtının devreye alınmasına ilişkin seçenektir. IBM MQ Appliance belgelerinde [High Availability \(Yüksek Kullanılabilirlik\)](#) ve [Disaster Recovery \(Olağanüstü Durum Kurtarma\)](#) başlıklı konuya bakın.

Bir ileti sistemi, sisteme girilen iletilerin hedeflerine teslim edilmesini sağlar. IBM MQ , **dspmqzte** komutunu kullanarak bir kuyruk yöneticisinden diğerine hareket ederken iletinin rotasını izleyebilir. Bir

sistem başarısız olursa, hata tipine ve sistemin konfigürasyonunun nasıl tanımlandığına bağlı olarak iletiler çeşitli şekillerde kurtarılabilir. IBM MQ , iletilerin alınmasını, iletilmesini ve teslimini işleme alan kuyruk yöneticilerinin etkinliklerinin kurtarma günlüklerini tutar.Üç tip kurtarma için bu günlükleri kullanır:

1. IBM MQ ' yi planlı bir şekilde durdurduğunuzda *kurtarmayı yeniden başlatın*.
2. *Hata kurtarma*, bir hata durduğunda IBM MQ.
3. *Ortam kurtarma*(hasarlı nesnelere geri yüklemek için).

Her durumda, kurtarma işlemi kuyruk yöneticisini, kuyruk yöneticisinin durduğu andaki durumuna geri yükler; ancak, hareket halindeki hareketler geriye işlenir ve kuyruk yöneticisinin durduğu sırada hareket halindeki tüm güncellemeler kuyruklardan kaldırılır. Kurtarma, tüm kalıcı iletileri geri yükler; kalıcı olmayan iletiler işlem sırasında kaybolabilir.



**DİKKAT:** Kurtarma günlüklerini farklı bir işletim sistemine taşıyamazsınız.

## Otomatik istemci yeniden bağlantısı

Birkaç bileşen yapılandırarak, istemci uygulamalarınızın herhangi bir ek kod yazmadan otomatik olarak yeniden bağlanmasını sağlayabilirsiniz.

Otomatik istemci yeniden bağlantısı *yerleşik*. Bağlantı, istemci uygulama programının herhangi bir noktasında otomatik olarak geri yüklenir ve nesnelere açma tanıtıcılarının tümü geri yüklenir.

Buna karşılık, el ile yeniden bağlantı, istemci uygulamasının MQCONN ya da MQCONNXkullanarak yeniden bağlantı oluşturmasını ve nesnelere yeniden açmasını gerektirir. Otomatik istemci yeniden bağlantısı, tüm istemci uygulamaları için değil, pek ya: Sayısız uygulamalar için uygundur.

Çizelge 30 sayfa 457 içinde, bir istemci iş istasyonuna kurulması gereken en eski IBM MQ istemcisi desteği yayını listelenir. Bir uygulamanın otomatik istemci yeniden bağlantısını kullanabilmesi için istemci iş istasyonlarını bu düzeylerden birine yükseltmeniz gerekir. Çizelge 31 sayfa 457 içinde otomatik istemci yeniden bağlantısını etkinleştirmek için diğer gereksinimler listelenir.

Yeniden bağlanma seçeneklerine program erişimiyle, bir istemci uygulaması yeniden bağlanma seçeneklerini ayarlayabilir. JMS ve XMS istemcileri dışında, bir istemci uygulamasının yeniden bağlanma seçeneklerine erişimi varsa, yeniden bağlanma olaylarını işlemek için bir olay işleyici de yaratabilir.

Var olan bir istemci uygulaması, yeniden derleme ve bağlantı oluşturma olmadan yeniden bağlantı desteğinden yararlanabilir:

- JMS dışı bir istemci için, yeniden bağlanma seçeneklerini ayarlamak üzere `mqclient.ini` ortam değişkenini `DefRecon` ayarlayın. Bir kuyruk yöneticisine bağlanmak için `CCDT` kullanın. İstemci çok eşgörümlü bir kuyruk yöneticisine bağlanacaksa, `CCDT` ' deki etkin ve yedek kuyruk yöneticisi yönetim ortamlarının ağ adreslerini belirtin. Eşlenmiş bir veri kuyruğu yöneticisi ya da IBM MQ Appliance ürünündeki bir HA kuyruk yöneticisi için, yapılandırmayı kolaylaştırmak üzere hem etkin hem de beklemedeki kuyruk yöneticileri tarafından kullanılan bir kayan IP adresi belirtebilirsiniz.
- JMS istemcisi için, bağlantı üreticisi yapılandırmasında yeniden bağlanma seçeneklerini belirleyin. Bir Java EE sunucusunun EJB taşıyıcısı içinde çalışırken, MDB 'ler IBM MQ kaynak bağdaştırıcısının etkinleştirme belirtilmeleriyle (ya da WebSphere Application Serverinde çalışıyorsa dinleyici kapılarıyla) sağlanan yeniden bağlanma düzeneğini kullanarak IBM MQ ' e yeniden bağlanabilir. Ancak, uygulama bir MDB değilse (ya da web taşıyıcısında çalışıyorsa), bu senaryoda otomatik istemci yeniden bağlantısı desteklenmediğinden, uygulamanın kendi yeniden bağlanma mantığını gerçekleştirmesi gerekir. IBM MQ kaynak bağdaştırıcısı, iletilerin iletiyle yönlendirilen Bean 'lere teslim edilmesi için bu yeniden bağlanma yeteneğini sağlar; ancak, sunucu uygulamaları gibi diğer Java EE öğeleri kendi yeniden bağlantılarını gerçekleştirmelidir.

**Not:** Otomatik istemci yeniden bağlantısı IBM MQ classes for Javatarafından desteklenmez.

Çizelge 30. Desteklenen istemciler

İstemci arabirimi	Müşteri	Yeniden bağlanma seçeneklerine program erişimi	Yeniden bağlantı desteği
İleti Sistemi API 'leri	C, C + +, COBOL, Unmanaged Visual Basic, XMS ( Windowsüzerinde yönetilmeyen XMS )	7.0.1	7.0.1
	JMS (JSE ve Java EE istemci taşıyıcısı ve yönetilen kaplar)	7.0.1.3	7.0.1.3
	IBM MQ classes for Java	Desteklenmiyor	Desteklenmiyor
	Yönetilen XMS ve yönetilen .NET istemcileri: C#, Visual Basic,	7.1	7.1
Diğer API 'ler	Windows Communication Foundation (Yönetilmeyen <sup>1</sup> )	Desteklenmiyor	7.0.1
	Windows Communication Foundation (Yönetilen <sup>1</sup> )	Desteklenmiyor	Desteklenmiyor
	Eksen 1	Desteklenmiyor	Desteklenmiyor
	Eksen 2	Desteklenmiyor	7.0.1.3
	HTTP (web 2.0)	Desteklenmiyor	7.0.1.3

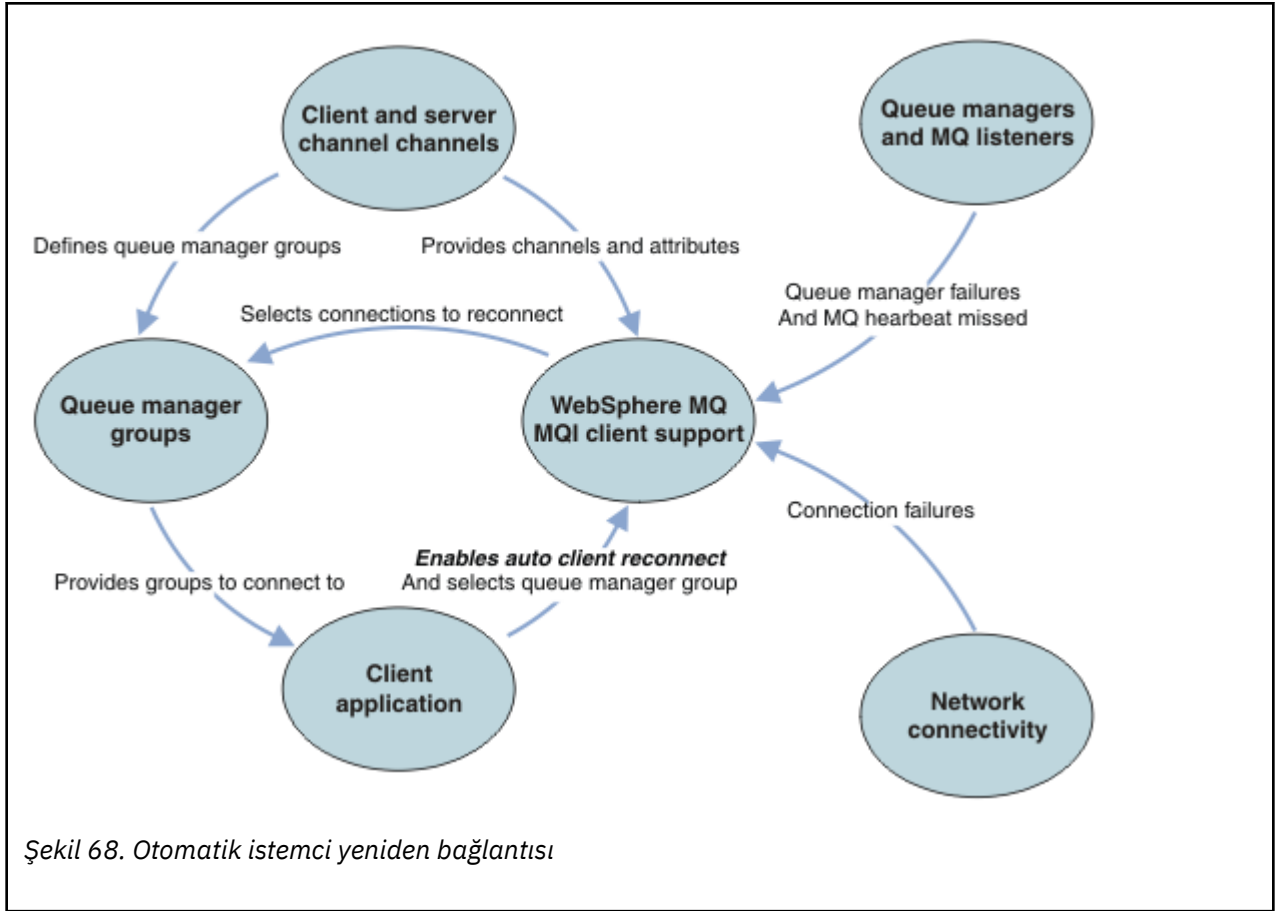
1. WCF bağ tanımlama yapılanışında yönetilen ya da yönetilmeyen kipi ayarlayın.

Otomatik yeniden bağlanma aşağıdaki yapılandırma gereksinimlerine sahiptir:

Çizelge 31. Otomatik yeniden bağlantı yapılandırma gereksinimleri

Bileşen	Gereksinim	Gereksinimi karşılamamanın etkisi
IBM MQ MQI client kurulumu	Bakınız Çizelge 30 sayfa 457	MQRC_OPTIONS_ERROR
IBM MQ Sunucu Kuruluşu	Düzy 7.0.1	MQRC_OPTIONS_ERROR
Kanal	SHARECNV > 0	MQRC_ENVIRONMENT_ERROR
Uygulama ortamı	İş parçacığı olmalıdır	MQRC_ENVIRONMENT_ERROR
MQI	Aşağıdakilerden biri: <ul style="list-style-type: none"> <li>MQCONN with MQCNO Options set to MQCNO_RECONNECT or MQCNO_RECONNECT_Q_MGR.</li> <li>Defrecon=YES  QMGR içinde mqclient.ini</li> <li>JMS içinde bağlantı üreticisinin <b>CLIENTRECONNECTOPTIONS</b> özelliğini ayarlayın.</li> </ul>	Bir bağlantı kesildiğinde ya da kuyruk yöneticisi sona erdiğinde ya da başarısız olduğunda MQCC_FAILED .

Şekil 68 sayfa 458 , istemcinin yeniden bağlanmasında yer alan bileşenler arasındaki ana etkileşimleri gösterir.



## İstemci uygulaması

İstemci uygulaması bir IBM MQ MQI client. Bir JMS istemcisi için otomatik istemci yeniden bağlantısına ilişkin ayrıntılar için [Otomatik JMS istemci yeniden bağlantısının kullanılması](#) başlıklı konuya bakın.

- Varsayılan olarak istemciler otomatik olarak yeniden bağlanmaz. MQCONNX MQCNO Seçenek MQCNO\_RECONNECT ya da MQCNO\_RECONNECT\_Q\_MGR'ı tanımlayarak otomatik istemci yeniden bağlantısını etkinleştirin.
- Birçok uygulama, ek kodlama olmadan otomatik yeniden bağlantıdan yararlanabilecek şekilde yazılır. mqclient.ini yapılandırma dosyasının kanal bölümündeki DefRecon özneliğini ayarlayarak, kodlama değişikliği yapmadan var olan programlar için otomatik yeniden bağlantıyı etkinleştirin.
- Aşağıdaki üç seçenektten birini kullanın:
  - Mantığın yeniden bağlantıdan etkilenmeyecek şekilde programı değiştirin. Örneğin, eşitleme noktası içinde MQI çağrılarını vermeniz ve geriletmiş işlemleri yeniden göndermeniz gerekebilir. Zamanuysuz tüketiciler, bir işlem geriletildiyse, bu müşterilerin 'askıya alınıp alınmadıklarını' kontrol etmelidir.
  - Yeniden bağlantıyı saptamak için bir olay işleyici ekleyin ve bağlantı yeniden kurulduğunda istemci uygulamasının durumunu geri yükleyin.
  - Otomatik yeniden bağlantıyı etkinleştirmeyin: Bunun yerine, istemcinin bağlantısını kesin ve aynı kuyruk yöneticisi grubunda çalışan başka bir kuyruk yöneticisi yönetim ortamı bulmak için yeni bir MQCONN ya da MQCONNX MQI çağrısı yapın.

Bu üç seçenекle ilgili daha fazla ayrıntı için bkz. ["Uygulama kurtarma" sayfa 545.](#)

- Aynı adı taşıyan bir kuyruk yöneticisine yeniden bağlanma, bir kuyruk yöneticisinin aynı yönetim ortamına yeniden bağlandığını garanti etmez.

Aynı kuyruk yöneticisinin bir eşgörünümüne yeniden bağlanmak için MQCNO seçeneğini MQCNO\_RECONNECT\_Q\_MGR kullanın.

- Bir istemci, yeniden bağlantı durumunu bilebilir diye bir olay işleyiciyi kaydedebilir. Olay işleyicide geçirilen MQHCONN kullanılmıyor. Aşağıdaki neden kodları verilmiştir:

#### **MQRC\_RECONNECTING**

Bağlantı başarısız oldu ve sistem yeniden bağlanmaya çalışıyor. Birden çok yeniden bağlanma girişiminde bulunursanız, birden çok MQRC\_RECONNECTING olayı alırsınız.

#### **MQRC\_RECONNECTED**

Yeniden bağlantı kuruldu ve tüm tanıtıcıları başarıyla yeniden oluşturuldu.

#### **MQRC\_RECONNECT\_FAILED**

Yeniden bağlantı başarısız oldu.

#### **MQRC\_RECONNECT\_QMID\_MISMATCH**

Yeniden bağlanabilir bir bağlantı belirtildi MQCNO\_RECONNECT\_Q\_MGR ve bağlantı farklı bir kuyruk yöneticisine yeniden bağlanmayı denedi.

#### **MQRC\_RECONNECT\_Q\_MGR\_REQD**

An option, such as MQMO\_MATCH\_MSG\_TOKEN in an MQGET call, was specified in the client program that requires reconnection to the same queue manager.

- Yeniden bağlanabilir bir istemci, yalnızca bağlandıktan sonra otomatik olarak yeniden bağlanabilir. Yani, MQCONNX çağrısı başarısız olursa yeniden denenmez. Örneğin, 2543 - MQRC\_STANDBY\_Q\_MGR dönüş kodunu MQCONNX' den alırsanız, kısa bir gecikmeden sonra çağrıyı yeniden yayınlayın.

#### **MQRC\_RECONNECT\_UYUMSUZ**

Uygulama MQPMO\_LOGICAL\_ORDER (MQPUT ve MQPUT1ile) ya da MQGMO\_LOGICAL\_ORDER (MQGET ile) yeniden bağlanma seçenekleri ayarlandığında. Neden kodunun döndürülmesinin nedeni, uygulamaların bu tür durumlarda yeniden bağlanma kullanmadığından emin olmaktır.

#### **MQRC\_CALL\_INTERRUPTED**

Kesintiye uğratma çağrısının yürütülmesi sırasında bağlantı kesildiğinde ve istemci yeniden bağlandığında bu neden kodu döndürülür. Eşitleme noktasının dışındaki bir kalıcı iletinin MQPUT işlemi, uygulamaya aynı neden kodunun döndürülmesine neden olur.

## **Yüksek kullanılabilirlik kuyruk yöneticileri**

Yüksek kullanılabilirlikli kuyruk yöneticilerinin etkin bir yönetim ortamı ve bir kuyruk yöneticisinin bir ya da daha çok yedek yönetim ortamı vardır. Etkin kuyruk yöneticisi yedek kuyruk yöneticileriyle uyumlulaştırılır; böylece, etkin yönetim ortamı başarısız olursa yedek veritabanı otomatik olarak devralabilir. Yüksek kullanılabilirlikli kuyruk yöneticileri sağlamak için birçok farklı çözüm vardır, bkz. "[Yüksek kullanılabilirlikli yapılandırmalar](#)" sayfa 466.

Yüksek kullanılabilirlikli bir kuyruk yöneticisi yedek yönetim ortamını etkinleştirdikten sonra, otomatik istemci yeniden bağlantısını kullanarak IBM MQ MQI client uygulamalarının yeniden başlatılmasını basitleştirebilirsiniz.

Yüksek kullanılabilirlikli bir kuyruk yöneticisinin yedek yönetim ortamı genellikle etkin yönetim ortamı için farklı bir ağ adresidir. İstemci bağlantı tanımlama çizelgesine (CCDT) her iki yönetim ortamının ağ adreslerini ekleyin. **CONNAME** parametresi için bir ağ adresleri listesi sağlayın ya da CCDT ' de kuyruk yöneticisi için birden çok satır tanımlayın. Eşleşmiş veri kuyruğu yöneticileri ve IBM MQ Appliance yüksek kullanılabilirlik kuyruk yöneticileri, etkin ya da beklemedeki kuyruk yöneticileriyle kullanılmak üzere tek bir adres belirtmeniz için kayan IP adreslerini destekler.

## **Kuyruk yöneticisi grupları**

Genellikle IBM MQ MQI clients , bir kuyruk yöneticisi grubundaki herhangi bir kuyruk yöneticisine yeniden bağlanır. Bazen bir IBM MQ MQI client ' in yalnızca aynı kuyruk yöneticisine yeniden bağlanmasını istersiniz. Kuyruk yöneticisine benzeşimi olabilir.

İstemci uygulamasının her zaman aynı adı taşıyan bir kuyruk yöneticisine mi, aynı kuyruk yöneticisine mi, yoksa istemci bağlantı çizelgesinde aynı QMNAME değeriyle tanımlanmış bir kuyruk yöneticisi kümesine mi bağlanacağını ve yeniden bağlanacağını seçebilirsiniz.

- İstemci kanal tanımındaki kuyruk yöneticisi adı özniteliği QMNAME, bir kuyruk yöneticisi grubunun adıdır.



- İstemci uygulamanızda, MQCONN ya da MQCONNX QmgrName değıştirgesinin değeri bir kuyruk yöneticisi adına ayarlarsanız, istemci yalnızca o adı taşıyan kuyruk yöneticilerine bağlanır. Kuyruk yöneticisi adına yıldız işareti (\*) eklerseniz, istemci kuyruk yöneticisi grubundaki herhangi bir kuyruk yöneticisine aynı QMNAME değeriyle bağlanır. Tam açıklama için bkz. [CCDT ' deki kuyruk yöneticisi grupları](#).

İstemcinin farklı bir kuyruk yöneticisine yeniden bağlanmasını önleyebilirsiniz. MQCNO seçeneğini ( MQCNO\_RECONNECT\_Q\_MGR) ayarlayın. IBM MQ MQI client farklı bir kuyruk yöneticisine yeniden bağlanırsa başarısız olur. MQCNO seçeneğini ayarlarsanız, MQCNO\_RECONNECT\_Q\_MGR, aynı kuyruk yöneticisi grubuna diğer kuyruk yöneticilerini eklemeyin. Yeniden bağlandığı kuyruk yöneticisi, bağlı olduğu kuyruk yöneticisiyle aynı değilse, istemci bir hata döndürür.

## Kuyruk paylaşım grupları

**z/OS** z/OS kuyruk paylaşım gruplarına otomatik istemci yeniden bağlantısı, yeniden bağlantı için diğer ortamlarla aynı mekanizmaları kullanır. İstemci, özgün bağlantı için yapılandırılanla aynı kuyruk yöneticisi seçimine yeniden bağlanır. Örneğin, istemci kanal tanımlama çizelgesini kullanırken, denetimci çizelgedeki tüm girişlerin aynı z/OS kuyruk paylaşım grubuna çözülmesini sağlamalıdır.

## İstemci ve sunucu kanalı tanımları

İstemci ve sunucu kanalı tanımlamaları, bir istemci uygulamasının yeniden bağlanabileceği kuyruk yöneticisi gruplarını tanımlar. Tanımlar, yeniden bağlantıların seçilmesini ve zamanlamasını ve güvenlik gibi diğer etkenleri yönetir; ilgili konulara bakın. Yeniden bağlantı için göz önünde bulundurulması gereken en ilgili kanal öznitelikleri iki grupta listelenir:

### İstemci bağlantısı öznitelikleri

#### **Bağlantı benzerliği (AFFINITY) AFFINITY**

Bağlantı benzerliği.

#### **İstemci kanalı ağırlığı (CLNTWGHT) CLNTWGHT**

Müşteri kanalı ağırlığı.

#### **Bağlantı adı (CONNAME) CONNAME**

Bağlantı bilgileri.

#### **sağlık işareti aralığı (HBINT) HBINT**

Sağlıklı işletim aralığı. Sunucu bağlantı kanalında sağlıklı işletim bildirim aralığını ayarlayın.

#### **Canlı Tutma Aralığı (KAINT) KAINT**

Canlı tutma aralığı. Sunucu bağlantı kanalında canlı tutma aralığını ayarlayın.

**z/OS** KAINT ' in yalnızca z/OS için geçerli olduğunu unutmayın.

#### **Kuyruk yöneticisi adı (QMNAME) QMNAME**

Kuyruk yöneticisi adı.

### Sunucu bağlantısı öznitelikleri

#### **sağlık işareti aralığı (HBINT) HBINT**

Sağlıklı işletim aralığı. İstemci bağlantı kanalında sağlıklı işletim bildirim aralığını ayarlayın.

#### **Canlı Tutma Aralığı (KAINT) KAINT**

Canlı tutma aralığı. İstemci bağlantı kanalında canlı tutma aralığını ayarlayın.

**z/OS** KAINT ' in yalnızca z/OS için geçerli olduğunu unutmayın.

KAINT bir ağ katmanı sağlıklı işletim bildirim, HBINT ise istemci ile kuyruk yöneticisi arasındaki IBM MQ sağlıklı işletim bildirim. Bu sağlıklı işletim sinyallerini daha kısa bir süreye ayarlamak iki amaca hizmet eder:

1. Bağlantı üzerinde etkinlik benzetimi yaparak, etkin olmayan bağlantıları kapatmaktan sorumlu olan ağ katmanı yazılımının bağlantınızı kapatma olasılığı daha düşüktür.
2. Bağlantı kapalıysa, kopan bağlantı saptanmadan önceki gecikme kısaltılır.

Varsayılan TCP/IP canlı tutma aralığı iki saattir. KAINT ve HBINT özniteliklerini daha kısa bir süreye ayarlamayı düşünün. Bir ağın olağan davranışının otomatik yeniden bağlanma gereksinimlerine uygun



olduğunu varsaymayın. Örneğin, bazı güvenlik duvarları 10 dakika kadar kısa bir süre sonra etkin olmayan bir TCP/IP bağlantısını kapatabilir.

## Ağ bağlantırlığı

Yalnızca ağ tarafından IBM MQ MQI client ' e iletilen ağ hataları, istemcinin otomatik yeniden bağlanma yeteneği tarafından işlenir.

- İletim tarafından otomatik olarak gerçekleştirilen yeniden bağlantılar IBM MQ için görünmez.
- HBINT 'in ayarlanması, IBM MQ' in göremeyeceği ağ hatalarıyla başa çıkmanıza yardımcı olur.

## Kuyruk yöneticileri ve IBM MQ dinleyicileri

İstemci yeniden bağlantısı, sunucu hatası, kuyruk yöneticisi hatası, ağ bağlantırlığı hatası ve bir denetimci başka bir kuyruk yöneticisi eşgörünümüne geçtiğinde tetiklenir.

- Çok eşgörünümlü bir kuyruk yöneticisi kullanıyorsanız, denetimi etkin kuyruk yöneticisi yönetim ortamından yedek yönetim ortamına geçirdiğinizde istemcinin yeniden bağlanmasının ek bir nedeni ortaya çıkar.
- Varsayılan **endmqm** komutu kullanılarak bir kuyruk yöneticisinin sona erdirilmesi, otomatik istemci yeniden bağlantısını tetiklemez. Otomatik istemci yeniden bağlantısı istemek için **endmqm** komutuna -r seçeneğini ya da kapattıktan sonra yedek kuyruk yöneticisi yönetim ortamına aktarmak için -s seçeneğini ekleyin.

## IBM MQ MQI client otomatik yeniden bağlantı desteği

IBM MQ MQI clientinde otomatik istemci yeniden bağlanma desteğini kullanırsanız, istemci uygulaması otomatik olarak yeniden bağlanır ve kuyruk yöneticisine yeniden bağlanmak için bir MQCONN ya da MQCONNX MQI çağrısı yayınlamadan işlemeye devam eder.

- Otomatik istemci yeniden bağlantısı aşağıdaki oluşumlardan biriyle tetiklenir:
  - Kuyruk yöneticisi hatası
  - Bir kuyruk yöneticisinin sona erdirilmesi ve **endmqm** komutunda -r, reconnect, seçeneğinin belirtilmesi
- MQCONNX MQCNO seçenekleri, otomatik istemci yeniden bağlantısını etkinleştirip etkinleştirmedeğinizi denetler. Seçenekler [Yeniden yapılandırma seçenekleri](#) içinde açıklanır.
- Otomatik istemci yeniden bağlantısı, bağlantı tanıtıcısını ve tanıtıcıları diğer açık nesnelere geri yüklemek için uygulamanız adına MQI çağrılarını verir; böylece, programınız bozuk bağlantıdan kaynaklanan MQI hatalarını işledikten sonra olağan işlemeye devam edebilir. Bkz. "[Otomatik olarak yeniden bağlanan bir istemcinin kurtarılması](#)" sayfa 547.
- Bağlantı için bir kanal çıkış programı yazdıysanız, çıkış bu ek MQI çağrılarını alır.
- Yeniden bağlanma başladığında ve bittiğinde tetiklenen bir yeniden bağlanma olayı işleyicisini kaydedebilirsiniz.

Hedeflenen yeniden bağlantı süresi bir dakikadan fazla olmasa da, bir kuyruk yöneticisinin yönetilmesi gereken çok sayıda kaynağı olabileceği için yeniden bağlantı daha uzun sürebilir. Bu süre içinde, bir istemci uygulaması IBM MQ kaynaklarına ait olmayan kilitleri tutuyor olabilir. Bir istemcinin yeniden bağlantı için bekleyeceği süreyi sınırlamak üzere yapılandırabileceğiniz bir zamanaşımı değeri vardır. Değer (saniye) `mqclient.ini` dosyasında ayarlanır.

```
Channels:  
MQReconnectTimeout = 1800
```

Zamanaşımı süresi dolduktan sonra yeniden bağlanma girişiminde bulunulmaz. Sistem zamanaşımı süresinin dolduğunu saptadığında, bir MQRC\_RECONNECT\_FAILED hatası döndürür.

## İlgili kavramlar

[Yeniden bağlanabilir istemciler](#)

## İlgili görevler

[Kuyruk yöneticisinin durdurulması](#)

z/OS

## Konsol ileti izlemesi

IBM MQ for z/OS işletim sistemi üzerinde, kuyruk yöneticisi ya da kanal başlatıcısı tarafından yayınlanan ve özellikle önemli kabul edilmesi gereken bir dizi bilgi iletisi vardır. Bu iletiler kendi içinde bir sorunu göstermez, ancak adresleme gerekebilecek olası bir sorunu gösterdikleri için izlemede yararlı olabilir.

Bu konsol iletilerinin varlığı, bir kullanıcı uygulamasının sayfa kümesine çok sayıda ileti koyduğunu da gösterebilir; bu da daha büyük bir sorunun belirtisi olabilir:

- Denetimsiz döngü gibi, PUTs iletilerinin gönderildiği kullanıcı uygulamasıyla ilgili bir sorun.
- Kuyruktaki iletileri GETlayan bir kullanıcı uygulaması artık çalışmıyor.

## İzlenecek konsol iletileri

Aşağıdaki liste, daha büyük sorunları gösterebilecek iletileri özetlemektedir. Bu iletilerin sistem otomasyonu ile izlenmesinin gerekli olup olmadığını belirleyin ve olası sorunların etkili bir şekilde takip edilebilmesi için uygun belgeleri sağlayın.

### **CSQI004I: csect-adı DIKKATE AL kuyruk-adı BY dizin-tipi FOR bağlantı-tipi CONNECTION bağlantı-adı, num-msgs MESSAGES ATLANDI**

- Kuyruk yöneticisi, tanımlanmış dizini olmayan bir kuyruktan ileti tanıtıcısı ya da ilinti tanıtıcısı temelinde ileti alan bir uygulama saptadı.
- Yerel kuyruk nesnesini ( *kuyruk-adı* ), INDXTYPE özneliğini *dizin-tipi* değerine sahip olacak şekilde değiştirerek, tanımlanan kuyruk için bir dizin oluşturmayı düşünün.

### **CSQI031I: böcek-adı SAYFA KÜMESİNİN YENİ KAPSAMI PSID BAŞARIYLA BIÇIMLENDİ**

- Bu sayfa kümesine ayrılmış kuyrukların derinliğini denetleyin.
- İletilerin işlenememesinin nedenini araştırın.

### **CSQI041I: csect-name JOB iş adı USER userid HAD ERROR HAD PAGE SET psid**

- Sayfa kümesinin kuyruk yöneticisine ayrılıp ayrılmadığını belirleyin.
- Sayfa kümesinin durumunu belirlemek için bir **DISPLAY USAGE** komutu verin.
- Ek hata iletileri için kuyruk yöneticisi iş günlüğünü denetleyin.

### **CSQI045I: csect-name Günlük RBA rba' ye ulaştı. Günlüğü sıfırlamayı planla**

- Uygun bir zamanda kuyruk yöneticisini durdurmayı ve günlükleri sıfırlamayı planlayın.
- Kuyruk yöneticiniz 6 baytlık günlük RNA kullanıyorsa, kuyruk yöneticisini 8 baytlık günlük RNA ' larını kullanacak şekilde dönüştürmeyi düşünün.

### **CSQI046E: csect-name Günlük RBA rba' ye ulaştı. Günlük ilk duruma getirme işlemi gerçekleştir**

- Uygun bir zamanda kuyruk yöneticisini durdurmayı ve günlükleri sıfırlamayı planlayın.
- Kuyruk yöneticiniz 6 baytlık günlük RNA kullanıyorsa, kuyruk yöneticisini 8 baytlık günlük RNA ' larını kullanacak şekilde dönüştürmeyi düşünün.

### **CSQI047E: csect-name Günlük RBA rba' ye ulaştı. Kuyruk yöneticisini durdur ve günlükleri ilk durumuna getir**

- Kuyruk yöneticisini hemen durdurun ve günlükleri ilk durumuna getirin.
- Kuyruk yöneticiniz 6 baytlık günlük RNA kullanıyorsa, kuyruk yöneticisini 8 baytlık günlük RNA ' larını kullanacak şekilde dönüştürmeyi düşünün.

#### **CSQJ004I: ACTIVE LOG COPY n INACTIVE, LOG IN SINGLE MODE, ENDRBA= ttt**

- Kuyruk yöneticisi 'single' (tek) günlüğe kaydetme kipini etkinleştirdi. Bu genellikle bir günlük boşaltma sorununun göstergesidir.
- Etkin ve arşiv günlüklerinin çift yönlülük ayarlarını belirlemek için bir **DISPLAY LOG** komutu verin. Bu görüntü, boşaltma işlemi gerektiren etkin günlüklerin sayısını da gösterir.
- Ek hata iletileri için kuyruk yöneticisi iş günlüğünü denetleyin

#### **CSQJ031D: csect-name, Günlük RBA Aralığı Sıfırlanmalıdır. Başlatmaya Devam Etme İçin 'Y' Yanıtını Verin Ya da 'N'**

- Kuyruk yöneticisini durdurun ve günlükleri en kısa zamanda ilk durumuna getirin ve günlükleri ilk durumuna getirin.
- Kuyruk yöneticiniz 6 baytlık günlük RNA kullanıyorsa, kuyruk yöneticisini 8 baytlık günlük RNA ' larını kullanacak şekilde dönüştürmeyi düşünün.

#### **CSQJ032E: csect-name alert-lvl - max-rba günlük rba aralığının sonuna yaklaşıyor. Geçerli Günlük RBA: current-rba.**

- Kuyruk yöneticisini durdurmaya ve günlükleri en kısa zamanda ilk durumuna getirmeyi planlayın.
- Kuyruk yöneticiniz 6 baytlık günlük RNA kullanıyorsa, kuyruk yöneticisini 8 baytlık günlük RNA ' larını kullanacak şekilde dönüştürmeyi düşünün.

#### **CSQJ110E: SON KOPYAN ETKİN GÜNLÜK VERİ KÜMESİ Nnn Yüzde Dolu**

- Günlük boşaltma işlemiyle ilgili bekleyen istekleri belirlemek üzere bir görüntü isteği gerçekleştirerek diğer bekleyen boşaltma görevlerini tamamlamak için gereken adımları gerçekleştirin. İstekleri karşılamak için gerekli işlemi gerçekleştirin ve boşaltmanın devam etmesine izin verin.
- Yeterli etkin günlük veri kümesi olup olmadığını göz önünde bulundurun. Gerekirse, DEFINE LOG komutunu kullanarak dinamik olarak ek günlük veri kümeleri ekleyebilirsiniz.

#### **CSQJ111A: ETKİN GÜNLÜK VERİ KÜMELERİNDE ALAN KALMADI**

- Günlük boşaltma işlemiyle ilgili bekleyen istek olmadığından emin olmak için bir görüntüleme isteği gerçekleştirin. İstekleri karşılamak için gerekli işlemi gerçekleştirin ve boşaltmanın devam etmesine izin verin.
- Yeterli etkin günlük veri kümesi olup olmadığını göz önünde bulundurun. Gerekirse, DEFINE LOG komutunu kullanarak dinamik olarak ek günlük veri kümeleri ekleyebilirsiniz.
- Gecikme, boşaltma için gerekli bir kaynağın olmamasından kaynaklanıyorsa, boşaltma işleminin tamamlanabilmesi için gerekli kaynağın kullanılabilir kılınması ve böylece günlüğe kaydetmenin devam etmesine izin verilmesi gerekir. Bu koşuldan kurtarmaya ilgili bilgi için Günlük sorunlarını arşivlebaşlıklı konuya bakın.

#### **CSQJ114I: ARŞİV VERİ KÜMESİNDE HATA OLUŞTU, BOŞALMA İŞLEME YALNIZCA BİR ARŞİV VERİ KÜMESİYLE DEVAM EDİYOR**

- Ek hata iletileri için kuyruk yöneticisi iş günlüğünü denetleyin.
- Arşiv günlüğünün ikinci bir kopyasını alın ve BSDS ' nizi el ile güncelleyin.

#### **CSQJ115E: OFFLOAD BAŞARISIZ OLDU, BİR ARŞİV VERİ KÜMESİ AYRILAMADI**

CSQJ103E ya da CSQJ073E ile ilgili hata durumu bilgilerini gözden geçirin. Veri kümesi ayırma hatasına neden olan koşulu düzelterek, yeniden denemede boşaltma işleminin gerçekleşmesini sağlayın.

#### **CSQJ136I: BAĞLANTI İÇİN MANYETİK BANT BİRİMİ AYIRILAMIYOR-ID= xxxx İLİŞKİ-ID= yyyyyy, m AYRILMIŞ N İZİN VERİLDİ**

- Ek hata iletileri için kuyruk yöneticisi iş günlüğünü denetleyin.

**CSQJ151I: csect-name ERROR READING RBA rrr, CONNECTION-ID= xxxx ILINTI-ID= yyyyyy REASON CODE= ccc**

- Ek iletiler için kuyruk yöneticisi iş günlüğünü denetleyin.
- Hangi bağlantının etkinliğini kesinleştirmediyini belirlemek için bir **DISPLAY CONN** komutu verin.
- Uygulamanın güncellemelerini kesinleştirebildiğinden emin olun.

**CSQJ160I: LONG-RUNNING UOW FOUND, URID= urid CONNECTION NAME= name**

- Ek iletiler için kuyruk yöneticisi iş günlüğünü denetleyin.
- Hangi bağlantının etkinliğini kesinleştirmediyini belirlemek için bir **DISPLAY CONN** komutu verin.
- Uygulamanın güncellemelerini kesinleştirebildiğinden emin olun.

**CSQJ161I: n OFFLOADS, URID= urid CONNECTION NAME= name**

- Sayfa kümesinin kuyruk yöneticisine ayrılıp ayrılmadığını belirleyin.
- Sayfa kümesinin durumunu belirlemek için bir **DISPLAY USAGE** komutu verin.
- Ek iletiler için kuyruk yöneticisi iş günlüğünü denetleyin.

**CSQP011E: SAYFA KÜMESİNE İLİŞKİN BAĞLANTI HATASI DURUMU ret kodu psid**

- Bu sayfa kümesine ayrılmış kuyrukların derinliğini denetleyin.
- İletilerin işlenememesinin nedenini araştırın.

**CSQP013I: csect-name NEW KAPLAM FOR PAGE SET psid. YENİ KAPSAM ŞİMDİ BİÇİMLENDİRİLECEK**

- Bu sayfa kümesine ayrılmış kuyrukların derinliğini denetleyin.
- İletilerin işlenememesinin nedenini araştırın.
- Kuyrukların başka bir sayfa kümesine yerleştirilmesi gerekip gerekmediğini belirleyin.
- Birim doluyorsa, sayfa için çok birimli bir veri kümesi oluşturmanız gerekip gerekmediğini belirleyin. Sayfa kümesi zaten çok birimli ise, kullanılmakta olan depolama grubuna daha fazla birim ekleyebilirsiniz. Bir kez daha yer açıldıktan sonra, sayfa kümesi **EXPAND** yöntemini **SYSTEM** olarak ayarlayarak genişletmeyi yeniden deneyin. Yeniden deneme gerekiyorsa, **EXPAND** ögesini **SYSTEM** olarak değiştirin ve daha sonra, olağan ayarınıza geri dönün.

**CSQP014E: csect-name EXPANSION FAILED FOR PAGE SET psid. UZATILMASINA İLİŞKİN GELECEKTEKİ İSTEKLER REDDEDİLECEK**

- Bu sayfa kümesine ayrılmış kuyrukların derinliğini denetleyin.
- İletilerin işlenememesinin nedenini araştırın.
- Kuyrukların başka bir sayfa kümesine yerleştirilmesi gerekip gerekmediğini belirleyin.

**CSQP016E: BÖCEK-ADı SAYFA KÜMESİ PSID KAPSAM SAYISI ÜST SINIRINA ULAŞTI. YENİDEN GENİŞLETİLEMEZ**

- Bu sayfa kümesine ayrılmış kuyrukların derinliğini denetleyin.
- İletilerin işlenememesinin nedenini araştırın.

**CSQP017I: csect-adi SAYFA KÜMESİ İÇİN GENİŞLETME BAŞLADI psid**

IBM MQ'indeki İş Birimlerinin durumunu belirlemek için DISPLAY THREAD komutlarını verin.

**CSQP047E: Kullanılmayan sayfa kümeleri sorunlara neden olabilir-bu durumu düzeltmek için işlem yapın**

- Sistem programcısı yanıtını izleyin.

**CSQQ008I: nn kurtarma birimleri qqqq kuyruk yöneticisinde hala şüpheli.**

- Ölü mektup kuyruğunuzun durumunu araştırın. Gitmeyen iletiler kuyruğunun PUT devre dışı bırakılmadığından emin olun.
- Gitmeyen iletiler kuyruğunun MAXMSG sınırında olmadığından emin olun.

### **CSQQ113I: *psb-adı bölge-tnt* Bu ileti işlenemiyor**

- CSQINPX hatasının nedenini saptamak için CSQOUTX veri kümesini denetleyin.
- Bazı komutlar işlenmeyebilir.

### **CSQX035I: *csect-name* Kuyruk yöneticisine bağlantı *qmgr-name* durduruluyor ya da bozuk, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)**

- Hatanın nedenini saptamak için MQRC ' yi denetleyin.
- Bu kodlar, IBM MQ for z/OS iletilerinde, tamamlama ve neden kodlarındabelgelenmiştir.

### **CSQX032I: *csect-name* Başlatma komutu işleyicisi sonlandırıldı**

- Hatanın nedenini saptamak için MQRC ' yi denetleyin.
- Bu kodlar, IBM MQ for z/OS iletilerinde, tamamlama ve neden kodlarındabelgelenmiştir.

### **CSQX048I: *csect-name* ad için ileti dönüştürülemiyor, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)**

- TCP/IP hatasının nedenini belirlemek için iş günlüğünü denetleyin.
- TCP/IP adres alanında hata olup olmadığını denetleyin.

### **CSQX234I: *csect-name* Dinleyici durduruldu, TRPTYPE=*trptype* INDISP=*atma***

- Dinleyici durmazsa, **STOP** komutuyla TCP/IP adres alanında hata olup olmadığını denetleyin.
- Sistem programcısı yanıtını izleyin.

### **CSQX407I: *csect-name* Küme kuyruğu *q-adı* tanımlamaları tutarsız**

- Küme içindeki birden çok küme kuyruğunun değerleri tutarsız. Farklılıkları araştırın ve çözün.

### **CSQX411I: *csect-name* Havuz yöneticisi durduruldu**

- Havuz yöneticisi bir hata nedeniyle durduysa, iş günlüğünde ileti olup olmadığını denetleyin.

### **CSQX417I: *csect-name* Kaldırılan kuyruk yöneticisi için gönderenler kaldı *qmgr-name***

- Sistem programcısı yanıtını izleyin.

### **CSQX418I: *csect-name* Küme için tek bir havuz *küme\_adı***

- Yüksek düzeyde kullanılabilirlik için kümeler, iki tam havuz ile yapılandırılmalıdır.

### **CSQX419I: *böcek-adı* Küme yok *küme\_adı***

- Sistem programcısı yanıtını izleyin.

### **CSQX420I: *csect-name* Küme için havuz yok *küme\_adı***

- Sistem programcısı yanıtını izleyin.

### **CSQX448E: *csect-name* Havuz yöneticisi hatalar nedeniyle durduruluyor. *n* saniye içinde yeniden başlat**

- Sistem programcısı yanıtını izleyin.

Bu ileti, SYSTEM.CLUSTER.COMMAND.QUEUE , şu komut kullanılarak etkinleştirilir:

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
```

Kuyruğu etkinleştirmeden önce, yayınlanmakta olan ilk CSQX448E iletisinden önce, havuz yöneticisinin sona ermesine neden olan sorunu çözmek için el ile müdahale gerekebilir.

### **CSQX548E: *csect-name* Yerel ileti kuyruğu, kanal *kanal-adı* neden=*mqrc* (*mqrc-text*)**

- Sistem programcısı yanıtını izleyin.

### **'func' işlevini kullanarak CSQX788I: *csect-adı* adres için DNS araması *adres n* saniye sürdü**

- Sistem programcısı yanıtını izleyin.

## **CSQY225E: böcek-adı Kuyruk yöneticisi, çubuğun üstündeki yerel saklama alanının kritik derecede altında**

- Kuyruk yöneticisi, çubuğun üzerindeki sanal saklama alanının kritik derecede altında çalışıyor. Durumu hafifletmek ve kuyruk yöneticisinin olası olağandışı sona erdirilmesini önlemek için işlem yapılmalıdır.

## **CSQ5038I: csect-name Hizmet görevi-görevi hh.mm.ss.nnnnnn' den bu yana yanıt vermiyor. Db2 ile ilgili sorunları denetleyin**

- Sistem programcısı yanıtını izleyin.

## **Yüksek kullanılabilirlikli yapılandırmalar**

IBM MQ kuyruk yöneticilerinizi yüksek kullanılabilirlikli (HA) bir yapılandırmada çalıştırmak istiyorsanız, kuyruk yöneticilerinizi yüksek kullanılabilirlikli bir yöneticiyle çalışacak şekilde ayarlayabilirsiniz; örneğin, PowerHA for AIX (önceki adıyla HACMP) ya da Microsoft Cluster Service (MSCS) ya da IBM MQ çok eşgörünümlü kuyruk yöneticileriyle. Linux sistemlerinde, yüksek kullanılabilirlik sağlamak için çekirdek tabanlı bir grup kullanan, eşlenmiş veri kuyruğu yöneticilerini (RDIMM 'ler) de devreye alabilirsiniz. Başka bir seçenek olan Native HA, kapsayıcı devreye alımlarını hedefliyor.

**MQ Appliance** Yüksek düzeyde kullanılabilirlik ya da olağanüstü durumdan kurtarma çözümü için bir diğer seçenek de bir çift IBM MQ aygıtının devreye alınmasına ilişkin seçenektir. IBM MQ Appliance belgelerinde **High Availability (Yüksek Kullanılabilirlik)** ve **Disaster Recovery (Olağanüstü Durum Kurtarma)** başlıklı konuya bakın.

Aşağıdaki yapılandırma tanımlarını bilmeniz gerekir:

### **Kuyruk yöneticisi kümeleri**

Bir ya da daha fazla bilgisayardaki iki ya da daha fazla kuyruk yöneticisi grubu, otomatik ara bağlantı sağlar ve yük dengeleme ve yedeklilik için kuyruklar arasında paylaşılmasına izin verir. IBM WebSphere MQ 7.1 ' den başlayarak, sorunlar çözülmünceye kadar sorunlara neden olan küme hatası kurtarma işlemlerini yeniden çalıştırır.

### **HA kümeleri**

HA kümeleri, diskler ve ağlar gibi iki ya da daha fazla bilgisayardan ve kaynaktan oluşan gruplardır. Bu gruplar birbirine bağlanır ve biri arızalanırsa, HACMP (AIX and Linux) ya da MSCS (Windows) gibi yüksek kullanılabilirlikli bir yöneticinin *hata durumunda yedek sisteme geçiş* gerçekleştirileceği şekilde yapılandırılır. Hata durumunda yedek sisteme geçiş, uygulamaların durum verilerini arızalı bilgisayardan kümedeki başka bir bilgisayara aktarır ve işlemlerini burada yeniden başlatır. Bu, HA kümesi içinde çalışan hizmetlerin yüksek düzeyde kullanılabilirliğini sağlar. IBM MQ kümeleri ve HA kümeleri arasındaki ilişki "[HA kümelerinin kuyruk yöneticisi kümelerine ilişkisi](#)" sayfa 467'inde açıklanmıştır.

### **Çok eşgörünümlü kuyruk yöneticileri**

İki ya da daha çok bilgisayarda yapılandırılan aynı kuyruk yöneticisinin eşgörünümleri. Birden çok eşgörünüm başlatılarak, bir eşgörünüm etkin eşgörünüm olur ve diğer eşgörünümler yedek olur. Etkin yönetim ortamı başarısız olursa, farklı bir bilgisayarda çalışan bir yedek yönetim ortamı otomatik olarak devralır. Çok eşgörünümlü kuyruk yöneticilerini kullanarak, HACMP ya da MSCS gibi bir küme teknolojisi gerektirmeden IBM MQ tabanlı yüksek kullanılabilirlikli ileti sistemi sistemlerinizi yapılandırabilirsiniz. HA kümeleri ve çok eşgörünümlü kuyruk yöneticileri, kuyruk yöneticilerini yüksek kullanılabilirlikli hale getirmenin alternatif yollarıdır. Çok eşgörünümlü bir kuyruk yöneticisini bir HA kümesine koyarak bunları birleştirmeyin.

### **Yüksek kullanılabilirlikli eşlenmiş veri kuyruğu yöneticileri (HA RDIMM 'ler)**

Üç Linux sunucusu grubundaki her bir düğümde yapılandırılan aynı kuyruk yöneticisinin eşgörünümleri. Bu üç eşgörünümünden biri etkin eşgörünümdür. Etkin kuyruk yöneticisindeki veriler zamanuyumlu olarak diğer iki örneğe eşlenir; bu nedenle, bir hata durumunda bu eşgörünümlerden biri devralabilir. Sunucuların gruplanması Pacemaker tarafından ve kopyalama DRBD tarafından denetlenir.

### **Olağanüstü durumdan kurtarma eşlenmiş veri kuyruğu yöneticileri (DR RDIMM 'leri)**

Bir kuyruk yöneticisi, bir yerdeki birincil düğümde çalışır; bu kuyruk yöneticisinin ikincil eşgörünümü farklı bir yerdeki kurtarma düğümünde bulunur. Veriler, birincil eşgörünüm ile ikincil eşgörünüm arasında eşlenir ve birincil düğüm bir nedenle kaybolursa, ikincil eşgörünüm birincil örneğe dönüştürülebilir ve başlatılabilir. Her iki düğüm de Linux sunucusu olmalıdır. Eşleme DRBD tarafından denetlenir.

### **Olağanüstü durum kurtarma/yüksek kullanılabilirlik eşlenmiş veri kuyruğu yöneticileri (DR/HA RDM)**

Bir sitede yüksek kullanılabilirlik grubunda çalışan, ancak ilk grubu kullanılamaz duruma getiren bir olağanüstü durum oluşursa başka bir yerdeki başka bir yüksek kullanılabilirlik grubuna geçilebilen bir eşlenmiş veri kuyruğu yöneticisi (RDQM) yapılandırabilirsiniz. Bu, DR/HA RDQM olarak bilinir.

#### **CP4I Yerel HA**

Yerel HA, IBM MQ' in kapsayıcı devreye alımlarını hedefleyen yüksek düzeyde kullanılabilirlik çözümdür. Yerel HA, farklı düğümlerde çalışan bir kuyruk yöneticisinin üç eşgörünümünü güncel tutmak için günlük eşlemesini kullanır. Bir eşgörünüm aynı anda etkindir ve iletileri işler. Etkin kuyruk yöneticisi, günlük güncellemelerini diğer iki yönetim ortamı için göndererek bunları güncel tutmasını sağlar. Etkin eşgörünüm başarısız olursa, eşleme örneklerinden biri otomatik olarak etkin rolü devralır.

### **Çok eşgörünümlü kuyruk yöneticileri ile HA kümeleri arasındaki farklar**

Çok eşgörünümlü kuyruk yöneticileri ve HA kümeleri, kuyruk yöneticileriniz için yüksek düzeyde kullanılabilirlik elde etmenin alternatif yollarıdır. İşte iki yaklaşım arasındaki farkları vurgulayan bazı noktalar.

Çok eşgörünümlü kuyruk yöneticileri aşağıdaki özellikleri içerir:

- IBM MQ ile bütünleştirilmiş temel hata durumunda yedek sisteme geçiş desteği
- Yüksek düzeyde kullanılabilirlik kümesinden daha hızlı yedek sisteme geçiş
- Basit yapılandırma ve işletim
- ile bütünleştirmeIBM MQ Explorer

Çok eşgörünümlü kuyruk yöneticilerine ilişkin sınırlamalar şunlardır:

- Yüksek düzeyde kullanılabilir, yüksek performanslı ağa bağlı depolama gereklidir
- Kuyruk yöneticisi başarısız olduğunda IP adresini değiştirdiğinden daha karmaşık ağ yapılandırması

HA kümeleri aşağıdaki özellikleri içerir:

- Uygulama sunucusu ya da veritabanı gibi birden çok kaynağı koordine etme yeteneği
- İkidenden fazla düğümden oluşan kümeler de dahil olmak üzere daha esnek yapılandırma seçenekleri
- İşletmen müdahalesi olmadan birden çok kez hata durumunda yedek sisteme geçiş yapılabilir
- Hata durumunda yedek sisteme geçişin bir parçası olarak kuyruk yöneticisinin IP adresinin devralınması

HA kümelerinin sınırlamaları şunlardır:

- Ek ürün satın alma ve beceri gereklidir
- Kümenin düğümleri arasında değiştirilebilen diskler gereklidir
- HA kümelerinin yapılandırması nispeten karmaşıktır
- Hata durumunda yedek sisteme geçiş tarihsel olarak oldukça yavaştır, ancak en son HA küme ürünleri bunu iyileştirmektedir
- Kuyruk yöneticileri gibi kaynakları izlemek için kullanılan komut dosyalarında eksiklikler varsa, gereksiz hata durumunda yedek sisteme geçiş oluşabilir.

### **HA kümelerinin kuyruk yöneticisi kümelerine ilişkisi**

Kuyruk yöneticisi kümeleri, kuyruk yöneticisi küme kuyruklarının kullanılabilir eşgörünümünde iletilerin yük dengelemesini sağlar. Bu, bir kuyruk yöneticisinin başarısız olmasının ardından ileti alışverişini uygulamaları, bir kuyruk yöneticisi küme kuyruğunun yürürlükte kalan eşgörünümüne ileti göndermeye



ve erişmeye devam edebildiği için, tek bir kuyruk yöneticisinden daha yüksek kullanılabilirlik sunar. Ancak, kuyruk yöneticisi kümeleri yeni iletileri bir kümedeki kullanılabilir kuyruk yöneticilerine otomatik olarak yönlendirse de, kullanılmayan bir kuyruk yöneticisinde kuyruğa alınan iletiler, kuyruk yöneticisi yeniden başlatılıncaya kadar kullanılamaz. Bu nedenle, kuyruk yöneticisi kümeleri tek başına tüm ileti verilerinin yüksek kullanılabilirliğini sağlamaz ya da kuyruk yöneticisi hatasının otomatik olarak saptanmasını ve kuyruk yöneticisi yeniden başlatma ya da yedek sisteme geçiş işleminin otomatik olarak tetiklenmesini sağlar. Yüksek Kullanılabilirlik (HA) kümeleri bu özellikleri sağlar. İki tip küme, iyi bir etki yaratmak için birlikte kullanılabilir. Kuyruk yöneticisi kümelerine giriş için bkz. [Kümeleri tasarlama](#).

## İlgili kavramlar

Linux > MQ Adv. > CD IBM MQ Advanced container için yüksek kullanılabilirlik

## Linux > AIX AIX and Linux üzerinde HA kümeleri

IBM MQ ürününü AIX and Linux platformlarında yüksek kullanılabilirlikli (HA) bir kümeyle kullanabilirsiniz: örneğin, PowerHA for AIX (eski adıyla HACMP), Veritas Cluster Server, HP Serviceguard ya da Red Hat Cluster Suiteçeren bir Red Hat Enterprise Linux kümesi.

Bu bölümde, "HA kümesi yapılandırılması" sayfa 468, HA kümelerinin kuyruk yöneticisi kümeleriyle ilişkisi, "IBM MQ müşterileri" sayfa 469ve "HA kümesinde çalışan IBM MQ" sayfa 469tanıtılır ve adımlar boyunca size yol gösterir ve bir HA kümesiyle kuyruk yöneticilerini yapılandırmak için uyarlayabileceğiniz örnek komut dosyaları sağlanır.

Bu bölümde açıklanan yapılandırma adımlarıyla ilgili yardım almak için ortamınıza özgü HA kümesi belgelerine bakın.

## HA kümesi yapılandırılması

Bu bölümde *düğüm* terimi, bir işletim sistemini ve HA yazılımını çalıştıran varlığa atıfta bulunmak için kullanılır; "bilgisayar", "sistem" ya da "makine" ya da "bölüm" ya da "blade" bu kullanımda eşanlamlı olarak kabul edilebilir. Tüm küme düğümlerinin IBM MQ iş yükünü çalıştırdığı karşılıklı devralma da dahil olmak üzere yedekleme ya da devralma yapılandırmalarının ayarlanmasına yardımcı olmak için IBM MQ komutunu kullanabilirsiniz.

*Yedek* yapılandırma, bir düğümün yalnızca yedek düğüm olarak hareket ederken, diğer düğümün çalışacağı en temel HA kümesi yapılandırmasıdır. Yedek düğüm iş gerçekleştirmez ve boşta olarak adlandırılır; bu yapılandırma bazen *soğuk yedekleme* olarak adlandırılır. Böyle bir yapılandırma, yüksek düzeyde donanım yedekliliği gerektirir. Donanımı ekonomik hale getirmek için, bu yapılandırmayı tek bir yedek düğüme sahip birden çok işçi düğümüne sahip olacak şekilde genişletebilirsiniz. Bunun nedeni, yedek düğümün başka bir işçi düğümünün işini devralabilmektir. Bu yapılandırmaya yedek yapılandırma olarak ve bazen "N+1" yapılandırması olarak atıfta bulunulur.

*devralma* yapılandırması, düğüm arızası durumunda tüm düğümlerin bazı işleri gerçekleştirebileceği ve kritik işlerin devralınabileceği daha gelişmiş bir yapılandırma.

*Tek taraflı devralma* yapılandırması, beklemedeki bir düğümün bazı ek, kritik olmayan ve taşınabilir olmayan işleri gerçekleştirdiği yapılandırma. Bu yapılandırma, beklemedeki yapılandırmaya benzer, ancak beklemedeki düğüm tarafından gerçekleştirilmekte olan (kritik olmayan) bir iş.

*Ortak devralma* yapılandırması, tüm düğümlerin yüksek kullanılabilirlikli (taşınabilir) işler gerçekleştirdikleri bir yapılandırma. Bu tip HA kümesi yapılandırması, tüm düğümlerin etkin olarak kritik iş yükünü işlediğini belirtmek için bazen "Etkin/Etkin" olarak da adlandırılır.

Genişletilmiş yedekleme yapılandırmasıyla ya da devralma yapılandırmalarından biriyle, diğer düğümlerin işini devralabilecek bir düğüme yerleştirebilecek en yüksek yükü göz önünde bulundurmanız önemlidir. Bu tür bir düğümün, kabul edilebilir bir performans düzeyini korumak için yeterli kapasiteye sahip olması gerekir.



## HA kümelerinin kuyruk yöneticisi kümelerine ilişkisi

Kuyruk yöneticisi kümeleri, yönetimi azaltır ve kuyruk yöneticisi küme kuyruklarının eşgörünümünde iletilerin yük dengelemesini sağlar. Ayrıca, bir kuyruk yöneticisinin başarısız olmasının ardından, ileti sistemi uygulamaları bir kuyruk yöneticisi küme kuyruğunun yürürlükte kalan eşgörünümüne erişmeye devam edebildiğinden, tek bir kuyruk yöneticisinden daha yüksek kullanılabilirlik sunar. Ancak, kuyruk yöneticisi kümeleri tek başına kuyruk yöneticisi hatasının otomatik olarak algılanmasını ve kuyruk yöneticisi yeniden başlatma ya da yedek sisteme geçiş işleminin otomatik olarak tetiklenmesini sağlamaz. HA kümeleri bu özellikleri sağlar. İki tip küme, iyi bir etki yaratmak için birlikte kullanılabilir.

## IBM MQ müşterileri

Yeniden başlatma ya da devralma işlemine tabi olabilecek bir kuyruk yöneticisiyle iletişim kuran IBM MQ istemcileri, bozuk bir bağlantıyı tolere etmek için yazılmalı ve sürekli olarak yeniden bağlanmayı denemelidir. IBM MQ, bağlantı kullanılabilirliği ve iş yükü dengelemeye yardımcı olan İstemci Kanal Tanımlama Çizelgesi 'nin (CCDT) işlenmesine ilişkin özellikleri içerir; ancak bunlar, yedek sisteme geçiş sistemiyle çalışırken doğrudan ilgili değildir.

İşlemsel işlevsellik, istemci aynı kuyruk yöneticisine bağlı olduğu sürece, IBM MQ MQI client ' in iki aşamalı hareketlere katılmasını sağlar. İşlem işlevselliği, kuyruk yöneticileri listesinden seçim yapmak için IP yük dengeleyici gibi teknikleri kullanamaz. Bir HA ürünü kullandığınızda, bir kuyruk yöneticisi üzerinde çalıştığı düğümün kimliğini (adını ve adresini) korur; böylece, işlem işlevselliği HA denetimi altındaki kuyruk yöneticileriyle birlikte kullanılabilir.

## HA kümesinde çalışan IBM MQ

Tüm HA kümeleri, hata durumunda yedek sisteme geçiş birimi kavramına sahip olur. Bu, yüksek kullanılabilirlikli hizmeti oluşturan tüm kaynakları içeren bir tanımlama kümeidir. Yedek sisteme geçiş birimi, hizmetin kendisini ve bağlı olduğu diğer tüm kaynakları içerir.

Yüksek düzeyde kullanılabilirlik çözümleri, bir yedek sisteme geçiş birimi için farklı terimler kullanır:

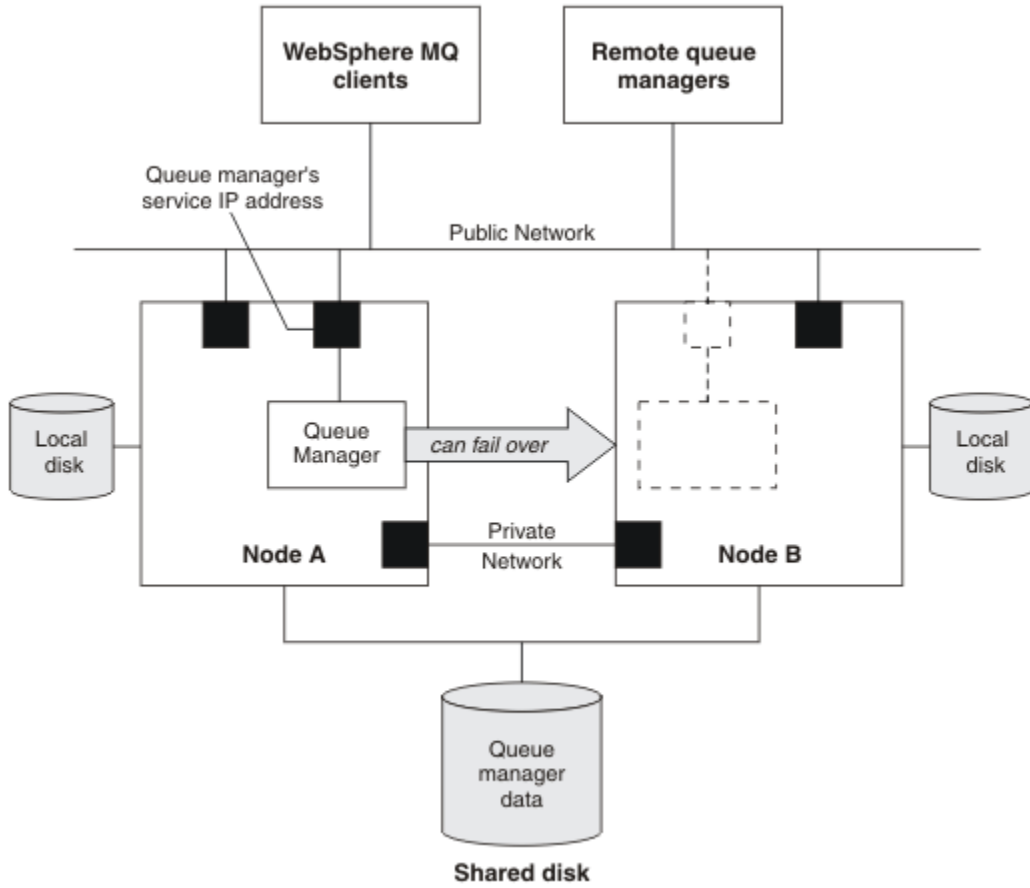
- PowerHA ' da AIX için hata durumunda yedek sisteme geçiş birimine *kaynak grubu* denir.
- Veritas Cluster Server 'da *hizmet grubu* olarak bilinir.
- Serviceguard 'da buna *paket* denir.

Bu konuda, hata durumunda yedek sisteme geçiş birimi anlamına gelen *kaynak grubu* terimi kullanılır.

IBM MQ için en küçük yedek sisteme geçiş birimi bir kuyruk yöneticisidir. Genellikle, kuyruk yöneticisini içeren kaynak grubu, özel olarak kaynak grubu tarafından kullanılmak üzere ayrılmış bir birim grubundaki ya da disk grubundaki paylaşılan diskleri ve kuyruk yöneticisine bağlanmak için kullanılan IP adresini de içerir. Ayrıca, aynı kaynak grubundaki bir dinleyici ya da tetikleyici izleyicisi gibi diğer IBM MQ kaynakları ayrı kaynaklar olarak ya da kuyruk yöneticisinin kendisinin denetimi altında da içerilebilir.

Bir HA kümesinde kullanılacak bir kuyruk yöneticisinin verileri ve kümedeki düğümler arasında paylaşılan disklerdeki günlükleri olmalıdır. HA kümesi, bir defada kümedeki yalnızca bir düğümün disklere yazabilmesini sağlar. HA kümesi, kuyruk yöneticisinin durumunu izlemek için bir izleme komut dosyası kullanabilir.

Kuyruk yöneticisiyle ilgili veriler ve günlükler için tek bir paylaşılan disk kullanılabilir. Ancak, bağımsız olarak boyutlandırılabilirlikleri ve ayarlanabilirlikleri için ayrı paylaşılan dosya sistemlerinin kullanılması normal bir uygulamadır.



Şekil 69. HA kümesi

Şekil 1, iki düğümlü bir HA kümesini gösterir. HA kümesi, bir kaynak grubunda tanımlanmış bir kuyruk yöneticisinin kullanılabilirliğini yönetir. Yalnızca bir düğüm (düğüm A) şu anda bir kuyruk yöneticisi çalıştırdığından, bu etkin/pasif ya da soğuk yedekleme yapılandırmasıdır. Kuyruk yöneticisi, verileri ve günlük dosyalarıyla paylaşılan bir diskte yaratıldı. Kuyruk yöneticisinin, HA kümesi tarafından da yönetilen bir hizmet IP adresi vardır. Kuyruk yöneticisi, paylaşılan diske ve hizmet IP adresine bağlıdır. HA kümesi kuyruk yöneticisini düğüm A 'dan düğüm B' ye doğru başarısız olursa, önce kuyruk yöneticisinin bağımlı kaynaklarını düğüm B 'ye taşır ve kuyruk yöneticisini başlatır.

HA kümesi birden çok kuyruk yöneticisi içeriyorsa, HA küme yapılandırmanız, hata durumunda yedek sisteme geçiş işleminden sonra aynı düğümde iki ya da daha çok kuyruk yöneticisinin çalışmasıyla sonuçlanabilir. HA kümesindeki her kuyruk yöneticisine, herhangi bir zamanda etkin olduğu herhangi bir küme düğümünde kullandığı kendi kapı numarası atanmalıdır.

Genel olarak, HA kümesi kök kullanıcı olarak çalışır. IBM MQ, mqm kullanıcısı olarak çalışır. IBM MQ yönetimi, mqm grubunun üyelerine verilir. mqm kullanıcısı ve grubunun her ikisinin de tüm HA kümesi düğümlerinde var olduğundan emin olun. Kullanıcı kimliği ve grup kimliği kümede tutarlı olmalıdır. Kök kullanıcı tarafından IBM MQ yönetimine izin verilmez; komut dosyalarını başlatan, durduran ya da izleyen komut dosyaları mqm kullanıcısına geçmelidir.

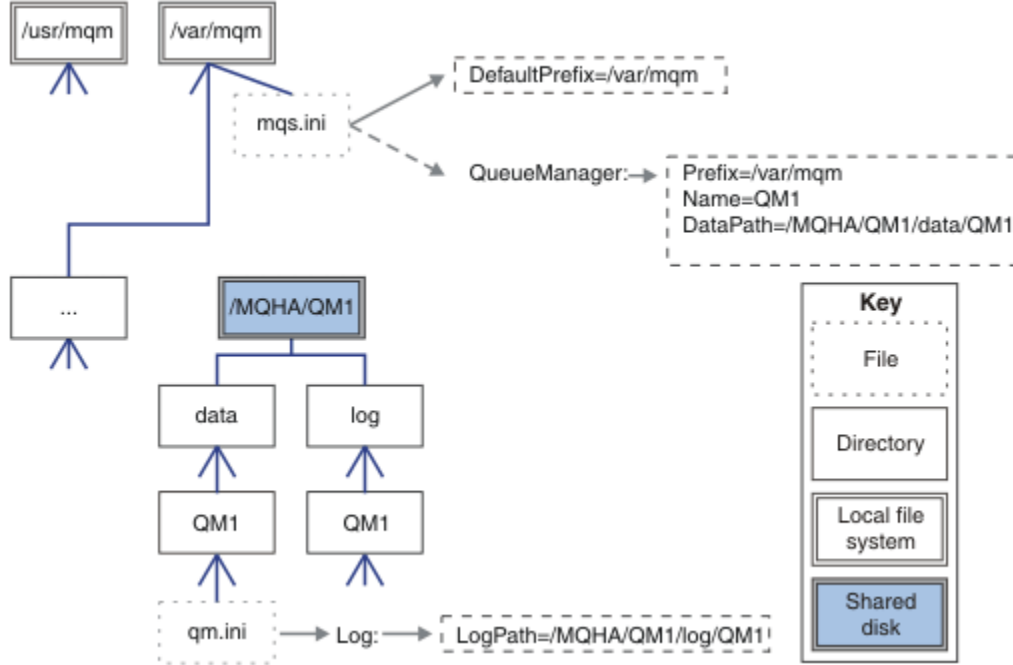
**Not:** IBM MQ tüm düğümlere doğru şekilde kurulmalıdır; ürün yürütülür dosyalarını paylaşamazsınız.

### Linux AIX **AIX and Linux üzerinde paylaşılan disklerin yapılandırılması**

Bir HA kümesindeki IBM MQ kuyruk yöneticisi, veri kütüklerinin ve günlük kütüklerinin paylaşılan bir diskteki ortak adlandırılmış uzak kütük sistemlerinde olmasını gerektirir.

## Bu görev hakkında

Şekil 1, HA kümesindeki bir kuyruk yöneticisi için olası bir yerleşim düzenini göstermektedir. Kuyruk yöneticisinin verileri ve günlük dizinleri, /MQHA/QM1 dizinine bağlanan paylaşılan diskte bulunur. Bu disk, hata durumunda yedek sisteme geçiş gerçekleştirildiğinde HA kümesinin düğümleri arasında değiştirilir; böylece veriler, kuyruk yöneticisi yeniden başlatıldığında kullanılabilir olur. mqm . ini dosyası, QM1 kuyruk yöneticisine ilişkin bir bölüme sahiptir. qm . ini dosyasındaki Günlük kısmı, LogPath için bir değere sahiptir.



Şekil 70. data ve log dizinlerini paylaşılan

## Yordam

1. Kuyruk yöneticisinin dosya sistemlerine ilişkin bağlama noktalarının adlarını belirleyin.  
Örneğin, kuyruk yöneticisinin veri dosyaları için /MQHA/qmqname/data ve günlük dosyaları için /MQHA/qmqname/log .
2. Kuyruk yöneticisinin verilerini ve günlük dosyalarını içerecek bir birim grubu (ya da disk grubu) oluşturun.  
Bu birim grubu, kuyruk yöneticisiyle aynı kaynak grubundaki yüksek kullanılabilirlik (HA) kümesi tarafından yönetilir.
3. Birim grubunda kuyruk yöneticisinin verileri ve günlük dosyaları için dosya sistemleri oluşturun.
4. Sırayla her bir düğüm için, dosya sistemlerine ilişkin bağlama noktalarını oluşturun ve dosya sistemlerinin bağlanabildiğinden emin olun.  
mqm kullanıcısı bağlama noktalarına sahip olmalıdır.

## Linux AIX AIX and Linux üzerinde HA kümesi kuyruk yöneticisi yaratılması

Yüksek kullanılabilirlikli bir kümede kuyruk yöneticisini kullanmaya yönelik ilk adım, düğümlerden birinde kuyruk yöneticisini yaratmaktır.

## Bu görev hakkında

Bir HA kümesinde kullanılmak üzere bir kuyruk yöneticisi yaratmak için, önce kümede kuyruk yöneticisinin yaratılacağı düğümlerden birini seçmeniz ve daha sonra, bu düğümde aşağıdaki adımları tamamlamanız gerekir.

## Yordam

1. Kuyruk yöneticisinin dosya sistemlerini düğüme bağlayın.
2. **crtmqm** komutunu kullanarak kuyruk yöneticisini yaratın.

Örneğin:

```
crtmqm -md /MQHA/qmgrname/data -ld /MQHA/qmgrname/log qmgrname
```

3. **strmqm** komutunu kullanarak kuyruk yöneticisini el ile başlatın.
4. Kuyruk ve kanal yaratma gibi kuyruk yöneticisinin ilk yapılandırmasını tamamlayın ve kuyruk yöneticisi başlatıldığında bir dinleyiciyi otomatik olarak başlatacak şekilde kuyruk yöneticisini belirleyin.
5. **endmqm** komutunu kullanarak kuyruk yöneticisini durdurun.
6. **addmqinf** komutunu görüntülemek için **dspmqrinf** komutunu kullanın:

```
dspmqrinf -o command qmgrname
```

Burada qmgrname , kuyruk yöneticisinin adıdır.

**addmqinf** komutunu kullanma hakkında daha fazla bilgi için bkz. [“AIX and Linux üzerindeki diğer HA küme düğümlerine kuyruk yöneticisi yapılandırması ekleniyor” sayfa 472.](#)

**addmqinf** komutu, aşağıdaki örneğe benzer bir şekilde görüntülenir:

```
addmqinf -sQueueManager -vName=qmgrname -vDirectory=qmgrname \  
-vPrefix=/var/mqm -vDataPath=/MQHA/qmgrname/data/qmgrname
```

7. Görüntülenen komutu dikkatli bir şekilde not edin.
8. Kuyruk yöneticisinin dosya sistemlerini kaldır.

## Sonraki adım

Artık [“AIX and Linux üzerindeki diğer HA küme düğümlerine kuyruk yöneticisi yapılandırması ekleniyor” sayfa 472](#) içinde açıklanan adımları tamamlamaya hazırsınız.

## **AIX and Linux üzerindeki diğer HA küme düğümlerine kuyruk yöneticisi yapılandırması ekleniyor**

Kuyruk yöneticisi yapılanış bilgilerini HA kümesindeki diğer düğümlere eklemeniz gerekir.

## Başlamadan önce

Bu görevi tamamlamadan önce, [“AIX and Linux üzerinde HA kümesi kuyruk yöneticisi yaratılması” sayfa 471](#) içindeki adımları tamamlamış olmanız gerekir. Kuyruk yöneticisini yarattıktan sonra, diğer düğümlerin her birinde aşağıdaki adımları tamamlayarak, HA kümesindeki diğer düğümlerin her birine kuyruk yöneticisine ilişkin yapılanış bilgilerini eklemeniz gerekir.

## Bu görev hakkında

Bir HA kümesinde kullanılmak üzere bir kuyruk yöneticisi yarattığınızda, önce kümede kuyruk yöneticisinin yaratılacağı düğümlerden birini seçmeniz gerekir (bkz. [“AIX and Linux üzerinde HA kümesi kuyruk yöneticisi yaratılması” sayfa 471](#)).

## Yordam

1. Kuyruk yöneticisi dosya sistemlerini sisteme bağlayın.
2. Kuyruk yöneticisi yapılanış bilgilerini düğüme ekleyin.  
Yapılandırma bilgilerini eklemenin iki yolu vardır:
  - /var/mqm/mqs.ini ögesini doğrudan düzenleyerek.

- “AIX and Linux üzerinde HA kümesi kuyruk yöneticisi yaratılması” sayfa 471’inde adım 6 ‘da **dspmqrinf** komutu tarafından görüntülenen **addmqinf** komutunu çalıştırarak.

3. Yapılandırmayı doğrulamak için kuyruk yöneticisini başlatın ve durdurun.

Kuyruk yöneticisini başlatmak ve durdurmak için kullanılan komutlar, **addmqinf** komutuyla aynı IBM MQ kuruluşundan verilmelidir. Kuyruk yöneticisini, şu anda kuyruk yöneticisiyle ilişkilendirilmiş olandan farklı bir kuruluştan başlatmak ve durdurmak için, önce **setmqm** komutunu kullanarak kuyruk yöneticisiyle ilişkili kuruluşu ayarlamanız gerekir. Daha fazla bilgi için bkz. [setmqm](#).

4. Kuyruk yöneticisi dosya sistemlerini kaldır.

Linux

AIX

## **AIX and Linux üzerinde bir HA kümesi kuyruk yöneticisini başlatmak için örnek kabuk komut dosyaları**

Kuyruk yöneticisi HA kümesinde bir kaynak olarak gösterilir. HA kümesinin kuyruk yöneticisini başlatabilmesi ve durdurabilmesi gerekir. Çoğu durumda, kuyruk yöneticisini başlatmak için bir kabuk komut dosyası kullanabilirsiniz. Bu komut dosyalarını, bir ağ dosya sistemi kullanarak ya da yerel disklerin her birine kopyalayarak kümedeki tüm düğümlerde aynı konumda kullanılabilir kılmamız gerekir.

**Not:** Başarısız olan bir kuyruk yöneticisini yeniden başlatmadan önce, uygulamalarınızın kuyruk yöneticisinin o yönetim ortamıyla bağlantısını kesmeniz gerekir. Bunu yapmazsanız, kuyruk yöneticisi doğru olarak yeniden başlamayabilir.

Burada uygun kabuk betiklerinin örnekleri verilmiştir. Bunları gereksinimlerinize göre uyarlayabilir ve HA kümenizin denetimi altında kuyruk yöneticisini başlatmak için kullanabilirsiniz.

Aşağıdaki kabuk komut dosyası, kuyruk yöneticisinin başarıyla başlatılabilmesi için HA kümesi kullanıcısından mqm kullanıcılarına nasıl geçileceğine ilişkin bir örnektir:

```
#!/bin/ksh
# A simple wrapper script to switch to the mqm user.
su mqm -c name_of_your_script $*
```

Aşağıdaki kabuk komut dosyası, kuyruk yöneticisinin yürürlükteki durumu hakkında herhangi bir varsayımda bulunmadan bir kuyruk yöneticisinin nasıl başlatılacağına ilişkin bir örnektir. Kuyruk yöneticisine ait olan işlemleri sona erdirmek için son derece ani bir yöntem kullandığını unutmayın:

```
#!/bin/ksh
#
# This script robustly starts the queue manager.
#
# The script must be run by the mqm user.
#
# The only argument is the queue manager name. Save it as QM variable
QM=$1
if [ -z "$QM" ]
then
echo "ERROR! No queue manager name supplied"
exit 1
fi
# End any queue manager processes which might be running.
srchstr="(|-m)$QM *.*$"
for process in amqzmuc0 amqzma0 amqfcbx amqfcpub amqpcsea amqzlaa0 \
amqzlsa0 runmqchi runmqlsr amqcrista amqirmfa amqimppa \
amqzfuma amqzmuf0 amqzmur0 amqzmgr0
do
ps -ef | tr "\t" " " | grep $process | grep -v grep | \
egrep "$srchstr" | awk '{print $2}' | \
xargs kill -9 > /dev/null 2>&1
done
# It is now safe to start the queue manager.
# The stmqm command does not use the -x flag.
stmqm ${QM}
```

Diğer ilgili programları başlatmak için komut dosyasını değiştirebilirsiniz.

## Linux AIX **AIX and Linux üzerinde bir HA küme kuyruk yöneticisini durdurmak için örnek kabuk komut dosyası**

Çoğu durumda, bir kuyruk yöneticisini durdurmak için kabuk komut dosyası kullanabilirsiniz. Burada uygun kabuk betiklerinin örnekleri verilmiştir. Bunları gereksinimlerinize göre uyarlayabilir ve HA kümenizin denetimi altında kuyruk yöneticisini durdurmak için kullanabilirsiniz.

Aşağıdaki komut dosyası, kuyruk yöneticisinin yürürlükteki durumuna ilişkin varsayımlar yapmadan bir kuyruk yöneticisinin hemen nasıl durdurulacağına ilişkin bir örnektir. Komut dosyası mqm kullanıcısı tarafından çalıştırılmalıdır. Bu nedenle, kullanıcıyı HA kümesi kullanıcısından mqm 'e geçirmek için bu komut dosyasının bir kabuk komut dosyasına sarılması gerekebilir. ([“AIX and Linux üzerinde bir HA kümesi kuyruk yöneticisini başlatmak için örnek kabuk komut dosyaları” sayfa 473](#) içinde örnek bir kabuk komut dosyası sağlanır.)

```
#!/bin/ksh
#
# The script ends the QM by using two phases, initially trying an immediate
# end with a time-out and escalating to a forced stop of remaining
# processes.
#
# The script must be run by the mqm user.
#
# There are two arguments: the queue manager name and a timeout value.
QM=$1
TIMEOUT=$2

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

if [ -z "$TIMEOUT" ]
then
  echo "ERROR! No timeout specified"
  exit 1
fi

for severity in immediate brutal
do
  # End the queue manager in the background to avoid
  # it blocking indefinitely. Run the TIMEOUT timer
  # at the same time to interrupt the attempt, and try a
  # more forceful version. If the brutal version fails,
  # nothing more can be done here.

  echo "Attempting ${severity} end of queue manager '${QM}'"
  case $severity in
    immediate)
      # Minimum severity of endmqm is immediate which severs connections.
      # HA cluster should not be delayed by clients
      endmqm -i ${QM} &
      ;;
    brutal)
      # This is a forced means of stopping queue manager processes.

      srchstr="(|-m)$QM *.*$"
      for process in amqzmuc0 amqzma0 amqfcxba amqfcpub amqpcsea amqzlaa0 \
        amqzlsa0 runmqchi runmqlsr amqcrista amqirmfa amqrmppa \
        amqzfuma amqmuf0 amqzmur0 amqzmgr0
      do
        ps -ef | tr "\t" " " | grep $process | grep -v grep | \
          egrep "$srchstr" | awk '{print $2}' | \
            xargs kill -9 > /dev/null 2>&1
      done
    esac

    TIMED_OUT=yes
    SECONDS=0
    while (( $SECONDS < ${TIMEOUT} ))
```

```

do
  TIMED_OUT=yes
  i=0
  while [ $i -lt 5 ]
  do
    # Check for execution controller termination
    srchstr="( |-m)$QM *.*$"
    cnt=`ps -ef | tr "\t" " " | grep amqzma0 | grep -v grep | \
    egrep "$srchstr" | awk '{print $2}' | wc -l`
    i=`expr $i + 1`
    sleep 1
    if [ $cnt -eq 0 ]
    then
      TIMED_OUT=no
      break
    fi
  done

  if [ ${TIMED_OUT} = "no" ]
  then
    break
  fi

  echo "Waiting for ${severity} end of queue manager '${QM}'"
  sleep 1
done # timeout loop

if [ ${TIMED_OUT} = "yes" ]
then
  continue      # to next level of urgency
else
  break         # queue manager is ended, job is done
fi

done # next phase

```

**Not:** Belirli bir kuyruk yöneticisi için hangi işlemlerin çalıştığına bağlı olarak, bu komut dosyasında bulunan kuyruk yöneticisi işlemlerinin listesi tam bir liste olmayabilir ya da o kuyruk yöneticisi için çalışan işlemlerden daha fazla işlem içerebilir:

```

for process in amqzmc0 amqzma0 amqfcxba amqfcpub amqpcsea amqzlaa0 \
  amqzlsa0 runmqchi runmqlsr amqcrsta amqirmfa amqimppa \
  amqzfuma amqmuf0 amqzmur0 amqzmgr0

```

Bir süreç, hangi özelliğin yapılandırıldığına ve belirli bir kuyruk yöneticisi için hangi işlemlerin çalıştığına bağlı olarak bu listeye eklenebilir ya da bu listeden çıkarılabilir. Süreçlerin ve işlemlerin belirli bir sırayla durdurulmasına ilişkin bilgilerin tam bir listesi için [UNIX ve Linux üzerinde kuyruk yöneticisinin el ile durdurulması](#) başlıklı konuya bakın.

## Linux AIX **AIX and Linux üzerinde bir HA küme kuyruk yöneticisinin izlenmesi**

Yüksek kullanılabilirlikli (HA) kümenin, kuyruk yöneticisinin durumunu düzenli aralıklarla izlemesini sağlamak olağan bir durumdur. Çoğu durumda, bunun için bir kabuk komut dosyası kullanabilirsiniz. Burada uygun kabuk betiklerinin örnekleri verilmiştir. Bu komut dosyalarını gereksinimlerinize göre uyarlayabilir ve bunları ortamınıza özgü ek izleme denetimleri yapmak için kullanabilirsiniz.

Bir sistemde birden çok IBM MQ kuruluşu olabilir. Birden çok kuruluşla ilgili daha fazla bilgi için bkz. [Birden çok kuruluş](#). İzleme komut dosyasını birden çok kuruluşta kullanmayı planlıyorsanız, bazı ek adımları gerçekleştirmeniz gerekebilir. Birincil kuruluşunuz varsa, komut dosyasını kullanmak için `MQ_INSTALLATION_PATH` belirtmenize gerek yoktur. Ters durumda, `MQ_INSTALLATION_PATH` 'in doğru tanımlandığından emin olmak için aşağıdaki adımları kullanın:

1. Bir kuyruk yöneticisine ilişkin doğru `MQ_INSTALLATION_PATH` ögesini tanımlamak için IBM MQ kuruluşunda `crtmqenv` komutunu kullanın:

```
crtmqenv -m qmname
```

Bu komut, `qmname` ile belirtilen kuyruk yöneticisi için doğru `MQ_INSTALLATION_PATH` değerini döndürür.

2. İzleme komut dosyasını uygun *qmname* ve *MQ\_INSTALLATION\_PATH* değıştirmeleriyle çalıştırın.

**Not:** PowerHA for AIX , kuyruk yöneticisine ilişkin izleme programına değıştirge sağlamak için bir yol sağlamaz. Her kuyruk yöneticisi için, kuyruk yöneticisi adını saran ayrı bir izleme programı yaratmanız gerekir. Aşağıda, kuyruk yöneticisi adını kapsüllemek için AIX üzerinde kullanılan bir komut dosyası örneğı verilmiştir:

```
#!/bin/ksh
su mqm -c name_of_monitoring_script qmname MQ_INSTALLATION_PATH
```

Burada *MQ\_INSTALLATION\_PATH* , *qmname* kuyruk yöneticisinin ilişkilendirildiğı IBM MQ kuruluşunun yolunu belirten isteğe bağı bir parametredir.

Aşağıdaki komut dosyası, **runmqsc** ' in askıda kalma olasılığına karşı sağlam değildir. Genellikle, HA kümeleri bir asılı izleme komut dosyasını bir hata olarak görür ve bu olasılığa karşı sağlamdır.

Ancak komut dosyası, kuyruk yöneticisinin başlangıç durumunda olmasına izin verir. Bunun nedeni, HA kümesinin başlatılır başlatılmaz kuyruk yöneticisini izlemeye başlaması için yaygın olmasıdır. Bazı HA kümeleri, kaynaklar için başlangıç aşaması ile çalıştırma aşaması arasında ayırım sağlar, ancak başlangıç aşamasının süresinin yapılandırılması gerekir. Kuyruk yöneticisini başlatmak için geçen süre, yapması gereken iş miktarına bağı olduğundan, bir kuyruk yöneticisini başlatmak için geçmesi gereken süre üst sınırını seçmek zordur. Çok düşük bir değer seçerseniz, HA kümesi, başlatma işlemi tamamlanmadığında kuyruk yöneticisinin başarısız olduğunu varsayar. Bu, sonsuz bir hata durumunda yedek sisteme geçiş sırasına neden olabilir.

Bu komut dosyası mqm kullanıcısı tarafından çalıştırılmalıdır; bu nedenle, kullanıcıyı HA kümesi kullanıcısından mqm 'e geçirmek için bu komut dosyasının bir kabuk komut dosyasına sarılması gerekebilir ( "[AIX and Linux üzerinde bir HA kümesi kuyruk yöneticisini başlatmak için örnek kabuk komut dosyaları](#)" sayfa 473 içinde örnek bir kabuk komut dosyası sağlanır):

```
#!/bin/ksh
#
# This script tests the operation of the queue manager.
#
# An exit code is generated by the runmqsc command:
# 0 => Either the queue manager is starting or the queue manager is running and responds.
#     Either is OK.
# >0 => The queue manager is not responding and not starting.
#
# This script must be run by the mqm user.
QM=$1
MQ_INSTALLATION_PATH=$2

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

if [ -z "$MQ_INSTALLATION_PATH" ]
then
  # No path specified, assume system primary install or MQ level < 7.1.0.0
  echo "INFO: Using shell default value for MQ_INSTALLATION_PATH"
else
  echo "INFO: Prefixing shell PATH variable with $MQ_INSTALLATION_PATH/bin"
  PATH=$MQ_INSTALLATION_PATH/bin:$PATH
fi

# Test the operation of the queue manager. Result is 0 on success, non-zero on error.
echo "ping qmgr" | runmqsc ${QM} > /dev/null 2>&1
pingresult=$?

if [ $pingresult -eq 0 ]
then # ping succeeded

  echo "Queue manager '${QM}' is responsive"
  result=0

else # ping failed

  # Don't condemn the queue manager immediately, it might be starting.
```



```

srchstr="(|-m)$QM *.*$"
cnt=`ps -ef | tr "\t" " " | grep strmqm | grep "$srchstr" | grep -v grep \
| awk '{print $2}' | wc -l`
if [ $cnt -gt 0 ]
then
# It appears that the queue manager is still starting up, tolerate
echo "Queue manager '${QM}' is starting"
result=0
else
# There is no sign of the queue manager starting
echo "Queue manager '${QM}' is not responsive"
result=$pingresult
fi
fi
exit $result

```

Linux

AIX

## Kuyruk yöneticisini AIX and Linux üzerinde HA küme denetimi

### altına alma

Kuyruk yöneticisinin IP adresi ve paylaşılan diskleriyle, HA kümesinin denetimi altında kuyruk yöneticisini yapılandırmanız gerekir.

### Bu görev hakkında

Kuyruk yöneticisini HA kümesinin denetimi altına almak için, kuyruk yöneticisini ve ilişkili tüm kaynaklarını içerecek bir kaynak grubu tanımlamanız gerekir.

### Yordam

1. Kuyruk yöneticisini, kuyruk yöneticisinin birimini ya da disk grubunu ve kuyruk yöneticisinin IP adresini içeren kaynak grubunu yaratın.  
IP adresi, bilgisayarın IP adresi değil, sanal bir IP adresidir.
2. HA kümesinin, küme düğümleri arasındaki kaynakları doğru şekilde değiştirdiğini ve kuyruk yöneticisini denetlemeye hazır olduğunu doğrulayın.

Linux

AIX

## AIX and Linux üzerinde bir HA kümesi kuyruk yöneticisinin

### silinmesi

Kuyruk yöneticisini çalıştırmak için artık gerekli olmayan bir düğümden kuyruk yöneticisini kaldırmak isteyebilirsiniz.

### Bu görev hakkında

Kuyruk yöneticisini HA kümesindeki bir düğümden kaldırmak için yapılandırma bilgilerini kaldırmanız gerekir.

### Yordam

1. HA kümesinin artık bu düğümde kuyruk yöneticisini etkinleştirmeyi denemeyecek şekilde düğümü HA kümesinden kaldırın.
2. Kuyruk yöneticisinin yapılanış bilgilerini kaldırmak için aşağıdaki **rmvmqinf** komutunu kullanın:  
`rmvmqinf qmgrname`
3. İsteğe bağlı: Kuyruk yöneticisini tamamen silmek için **dltmqm** komutunu kullanın.

**Önemli:** **dltmqm** komutunu kullanarak kuyruk yöneticisinin silinmesiyle kuyruk yöneticisinin verilerinin ve günlük dosyalarının tamamen silindiğini unutmayın.

Kuyruk yöneticisini sildiğinizde, diğer düğümlerden kalan yapılandırma bilgilerini kaldırmak için **rmvmqinf** komutunu kullanabilirsiniz.

Sanal sunucuların yedek sisteme geçmesini desteklemek için MSCS ' nin tanıtılması ve ayarlanması. MSCS, Windows Server Failover Clustering (WSFC) olarak da bilinir.

Bu bilgiler yalnızca IBM MQ for Windows için geçerlidir.

**Not:** Windows Server 2016' den Microsoft Cluster Service (MSCS) için yeni ad Windows Server Failover Clustering (WSFC) olur.

MSCS/WSFC, sunucuları bir kümeye bağlamanızı sağlayarak verilerin ve uygulamaların kullanılabilirliğini artırmanızı ve sistemi yönetmenizi kolaylaştırır. MSCS/WSFC, sunucu ya da uygulama hatalarını otomatik olarak saptayabilir ve kurtarabilir.

MSCS/WSFC, uygulamalara, web sitelerine, yazdırma kuyruklarına ya da dosya paylaşımlarına (örneğin, disk spinleri, dosyaları ve IP adresleri) karşılık gelen sanal sunucuların yedek sisteme geçmesini destekler.

*Yedek Sisteme Geçiş* , MSCS/WSFC ' nin kümedeki bir bilgisayardaki bir uygulamada bir hata saptaması ve kesintiye uğrayan uygulamayı düzgün bir şekilde kapatması, durum verilerini diğer bilgisayara aktarması ve uygulamayı yeniden başlatması sürecidir.

Yedek sisteme geçiş kümelerini yapılandırma ve kullanma hakkında bilgi için alt konulara bakın.

Microsoft Cluster Service (MSCS) kümeleri, birbirine bağlı ve biri arızalanırsa, MSCS bir *hata durumunda yedek sisteme geçiş* gerçekleştirecek şekilde yapılandırılmış iki ya da daha fazla bilgisayardan oluşan gruplardır. Bu şekilde, uygulamaların durum verileri hatalı bilgisayardan kümedeki başka bir bilgisayara aktarılır ve burada yeniden başlatılır.

**Not:** Windows Server 2016' den Microsoft Cluster Service (MSCS) için yeni ad Windows Server Failover Clustering (WSFC) olur.

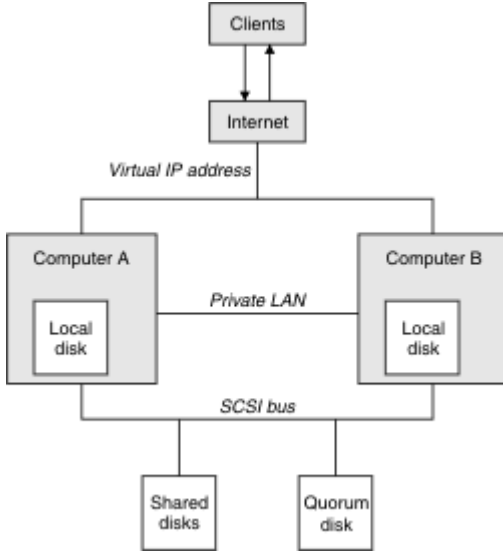
[“Yüksek kullanılabilirlikli yapılandırmalar” sayfa 466](#) , MSCS kümeleri, çok eşgörünümlü kuyruk yöneticileri ve IBM MQ kümeleri arasında bir karşılaştırma içerir.

Bu bölümde ve alt konularında, *kümeteri* tek başına kullanıldığında **her zaman** bir MSCS kümesi anlamına gelir. Bu, bu kılavuzda başka bir yerde açıklanan bir IBM MQ kümesinden farklıdır.

İki makineli bir küme, *sanal IP adresi* kullanılarak istemci erişimi için bir ağa ortaklaşa bağlanan iki bilgisayardan (örneğin, A ve B) oluşur. Bunlar, bir ya da daha çok özel ağ tarafından birbirine bağlanmış da olabilir. A ve B, her birinde sunucu uygulamaları için en az bir disk paylaşır. Yedek bağımsız disk dizisi olması gereken başka bir paylaşılan disk de vardır ( *RAID* ) MSCS ' nin özel kullanımı için Düzey 1; bu, *çekirdek* diski olarak bilinir. MSCS, donanımın ve yazılımın doğru çalışıp çalışmadığını denetlemek için her iki bilgisayarı da izler.

Bunun gibi basit bir kurulumda, her iki bilgisayarda da tüm uygulamalar yüklenmiş, ancak yalnızca A bilgisayarı canlı uygulamalarla çalışıyor; B bilgisayarı çalışıyor ve bekliyor. A bilgisayarında çeşitli sorunlardan biriyle karşılaşırsa, MSCS kesintiye uğrayan uygulamayı düzgün bir şekilde kapatır, durum verilerini diğer bilgisayara aktarır ve uygulamayı yeniden başlatır. Bu, *hata durumunda yedek sisteme geçiş* olarak bilinir. Uygulamalar, MSCS ile tam olarak etkileşimde bulunmaları ve hata durumunda yedek sisteme geçmeleri için *kümeye duyarlı* hale getirilebilirler.

İki bilgisayarlı bir küme için tipik bir ayar, [Şekil 71 sayfa 479](#) içinde gösterildiği gibi olur.



Şekil 71. İki bilgisayardan MSCS kümesi

Her bir bilgisayar, MSCS denetimi altında paylaşılan diske erişebilir, ancak aynı anda yalnızca bir adet olabilir. Hata durumunda yedek sisteme geçiş durumunda, MSCS diğer bilgisayara erişimi değiştirir. Paylaşılan diskin kendisi genellikle bir RAID 'dir, ancak olması gerekmez.

Her bir bilgisayar istemci erişimi için dış ağa bağlıdır ve her birinin bir IP adresi vardır. Ancak, bu kümeyle iletişim kuran bir dış istemci yalnızca bir *sanal IP adresinin* farkında ve MSCS, küme içindeki IP trafiğini uygun şekilde yönlendirir.

MSCS ayrıca iki bilgisayar arasında, örneğin sağlıklı işletim sinyalini kullanarak durumlarını izlemek ve veritabanlarını eşitlemek için, bir ya da daha fazla özel bağlantı üzerinden ya da genel ağ üzerinden kendi iletişimini gerçekleştirir.

### Windows MSCS kümelemesi için IBM MQ 'nin ayarlanması

Kuyruk yöneticisini MSCS 'ye yedek sisteme geçiş birimi yaparak IBM MQ 'u kümeleme için yapılandırabilirsiniz. Bir kuyruk yöneticisini MSCS 'ye kaynak olarak tanımlarsınız; bu kaynak kuyruk yöneticisini izleyebilir ve bir sorun varsa kümedeki başka bir bilgisayara aktarabilir.

**Not:** Windows Server 2016' den Microsoft Cluster Service (MSCS) için yeni ad Windows Server Failover Clustering (WSFC) olur.

Sistemini bunun için ayarlamak üzere, kümedeki her bir bilgisayara IBM MQ 'i kurarak başlayabilirsiniz.

Kuyruk yöneticisi IBM MQ kuruluş adıyla ilişkilendirildiğinden, kümedeki tüm bilgisayarlardaki IBM MQ kuruluş adı aynı olmalıdır. Bkz. [Kurma ve kaldırma](#).

Kuyruk yöneticilerinin kendileri, yalnızca bunları yarattığınız bilgisayarda bulunmaları gerekir. Hata durumunda yedek sisteme geçiş durumunda, MSCS diğer bilgisayarda kuyruk yöneticilerini başlatır. Ancak, kuyruk yöneticilerinin günlük ve veri kütükleri yerel bir sürücüde değil, küme tarafından paylaşılan bir diskte bulunmalıdır. Yerel bir sürücüde önceden kurulmuş bir kuyruk yöneticiniz varsa, IBM MQ ile verilen bir aracı kullanarak bu kuyruk yöneticisini yeni düzeye geçirebilirsiniz; bkz. [“Kuyruk yöneticisinin MSCS depolama alanına taşınması”](#) sayfa 482. MSCS ile kullanılmak üzere yeni kuyruk yöneticileri oluşturmak istiyorsanız, bkz. [“MSCS ile kullanılmak üzere kuyruk yöneticisi yaratılması”](#) sayfa 481.

Kuruluştan ve geçişten sonra MSCS 'nin kuyruk yöneticilerinizi haberdar etmesi için MSCS Cluster Administrator olanağını kullanın; bkz. [“Kuyruk yöneticisinin MSCS denetimi altına konması”](#) sayfa 483.

Bir kuyruk yöneticisini MSCS denetiminden kaldırmaya karar vererseniz, [“Kuyruk yöneticisinin MSCS denetiminden kaldırılması”](#) sayfa 489 içinde açıklanan yordamı kullanın.

## Windows Kurulum simetrisi ve MSCS

Bir uygulama bir düğümden diğerine geçtiğinde, düğümden bağımsız olarak aynı şekilde davranmalıdır. Bunu sağlamanın en iyi yolu, ortamların aynı olmasını sağlamaktır.

Yapabiliyorsa, her bir bilgisayarda aynı donanıma, işletim sistemi yazılımına, ürün yazılımına ve yapılandırmaya sahip bir küme kurun. Özellikle, iki bilgisayarda kurulu olan tüm gerekli yazılımların sürüm, bakım düzeyi, SupportPacs, yollar ve çıkışlar açısından aynı olduğundan ve "[MSCS güvenliği](#)" sayfa 480 içinde açıklandığı gibi ortak bir ad alanı (güvenlik ortamı) olduğundan emin olun.

## Windows MSCS güvenliği

Başarılı MSCS güvenliği için bu yönergeleri izleyin.

Yönergeler aşağıdaki gibidir:

- Kümedeki her bir bilgisayarda aynı yazılım kuruluşlarına sahip olduğunuzdan emin olun.
- Küme genelinde ortak bir ad alanı (güvenlik ortamı) oluşturun.
- *Küme sahibi* kullanıcı hesabının bir etki alanı hesabı olduğu bir etki alanının MSCS kümesi üyelerinin düğümlerini oluşturun.
- Kümedeki diğer kullanıcı hesaplarını, her iki düğümde de kullanılabilir olacak şekilde etki alanı hesaplarını da yapın. Zaten bir etki alanınız varsa ve IBM MQ ile ilgili hesaplar etki alanı hesaplarıdır. Şu anda bir etki alanınız yoksa, küme düğümlerine ve ilgili hesaplara hitap etmek için bir *mini etki alanı* ayarlamayı düşünebilirsiniz. Amacınız, iki bilgisayardan oluşan kümenizi tek bir bilgi işlem kaynağı gibi göstermek.

Bir bilgisayarda yerel olan bir hesabın diğerinde olmadığını unutmayın. Diğer bilgisayarda aynı ada sahip bir hesap oluştursanız bile, güvenlik tanıtıcısı (SID) farklıdır; bu nedenle, uygulamanız diğer düğüme taşındığında, izinler o düğümde bulunmaz.

IBM MQ MSCS desteği, hata durumunda yedek sisteme geçiş ya da taşıma sırasında, kuyruk yöneticisi nesnelerini içeren tüm dosyaların hedef düğümde eşdeğer izinlere sahip olmasını sağlar. Kod, denetimcilerin ve mqm gruplarının ve SYSTEM hesabının tam denetime sahip olup olmadığını ve Everyone ' un eski düğümde okuma erişimi olup olmadığını belirttik olarak denetler ve hedef düğümüne bu izin eklenir.

IBM MQ Hizmetinizi çalıştırmak için bir etki alanı hesabı kullanabilirsiniz. Kümedeki her bir bilgisayarda yerel mqm grubunda bulunduğundan emin olun.

## Windows MSCS ile birden çok kuyruk yöneticisinin kullanılması

Bir bilgisayarda birden çok kuyruk yöneticisi çalıştırıyorsanız, bu ayarlardan birini seçebilirsiniz.

Tuzaklar aşağıdaki gibidir:

- Tek bir gruptaki tüm kuyruk yöneticileri. Bu yapılandırmada, herhangi bir kuyruk yöneticisinde bir sorun oluşursa, gruptaki tüm kuyruk yöneticileri grup olarak diğer bilgisayara geçiş işlemi gerçekleştirir.
- Her gruptaki tek bir kuyruk yöneticisi. Bu yapılandırmada, kuyruk yöneticisiyle ilgili bir sorun oluşursa, diğer kuyruk yöneticilerini etkilemeden diğer bilgisayara tek başına geçer.
- İlk iki setupun karışımı.

## Windows Küme kipleri ve MSCS

Windows üzerinde IBM MQ ile bir küme sistemini çalıştırabileceğiniz iki kip vardır: Etkin/Edilgen ya da Etkin/Etkin.

**Not:** Microsoft Transaction Server (COM +) ile birlikte MSCS kullanıyorsanız, Etkin/Etkin kipini kullanamazsınız.

## Etkin/Edilgen kip

Etkin/Edilgen kipte, A bilgisayarında çalışan uygulama vardır ve B bilgisayarında yedekleme vardır, yalnızca MSCS bir sorun saptadığında kullanılır.

Bu kipi tek bir paylaşılan diskle kullanabilirsiniz, ancak herhangi bir uygulama hata durumunda yedek sisteme geçiş neden olursa, **tüm** uygulamaların bir grup olarak aktarılması gerekir (çünkü aynı anda yalnızca bir bilgisayar paylaşılan diske erişebilir).

MSCS 'yi A ile *tercih edilen* bilgisayar olarak yapılandırabilirsiniz. Daha sonra, A bilgisayarı onarıldığında ya da değiştirildiğinde ve yeniden düzgün çalıştığında, MSCS bunu algılar ve uygulamayı otomatik olarak A bilgisayarına geri çevirir.

Birden çok kuyruk yöneticisi çalıştırıyorsanız, her biri için ayrı bir paylaşılan disk kullanmayı düşünün. Daha sonra her kuyruk yöneticisini MSCS 'de ayrı bir gruba koyun. Bu şekilde, herhangi bir kuyruk yöneticisi diğer kuyruk yöneticilerini etkilemeden diğer bilgisayara geçebilir.

## Etkin/Etkin kip

Etkin/Etkin kipte, A ve B bilgisayarlarının her ikisinin de çalışan uygulamaları vardır ve her bir bilgisayardaki gruplar diğer bilgisayarı yedek olarak kullanacak şekilde ayarlanmıştır. A bilgisayarında bir hata saptanırsa, MSCS durum verilerini B bilgisayarına aktarır ve uygulamayı yeniden başlatır. B bilgisayar daha sonra kendi uygulamasını ve A ' larını çalıştırır.

Bu kurulum için en az iki paylaşılan disk gerekir. MCS 'yi A' nın uygulamaları için tercih edilen bilgisayar olarak, B 'yi B' nin uygulamaları için tercih edilen bilgisayar olarak yapılandırabilirsiniz. Hata durumunda yedek sisteme geçiş ve onarımdan sonra, her uygulama otomatik olarak kendi bilgisayarına geri döner.

IBM MQ için bu, örneğin, her biri kendi bilgisayarının tam gücünü kullanarak A ve B ' nin her birinde bir tane olmak üzere iki kuyruk yöneticisi çalıştırabilirsiniz anlamına gelir. A bilgisayarındaki bir hatadan sonra, her iki kuyruk yöneticisi de B bilgisayarında çalışacaktır. Bu, tek bir bilgisayarın gücünün paylaşılması ve büyük miktarlarda veriyi hızla işleme yeteneğinin azalması anlamına gelir. Ancak, A 'daki hatayı bulup onarıken kritik uygulamalarınız hala kullanılabilir olacaktır.

## Windows MSCS ile kullanılmak üzere kuyruk yöneticisi yaratılması

Bu yordam, yeni bir kuyruk yöneticisinin Microsoft Cluster Service (MSCS) denetiminin hazırlanması ve yerleştirilmesi için uygun bir şekilde yaratılmasını sağlar.

**Not:** Windows Server 2016' den Microsoft Cluster Service (MSCS) için yeni ad Windows Server Failover Clustering (WSFC) olur.

Kuyruk yöneticisini tüm kaynaklarıyla yerel bir sürücüde yarattıktan sonra günlük dosyalarını ve veri dosyalarını paylaşılan bir diske geçirerek başlayabilirsiniz. (Bu işlemi tersine çevirebilirsiniz.) Kaynakları paylaşılan bir sürücüde olan bir kuyruk yöneticisi yaratmayı **denemeyin** .

Bir komut isteminden ya da IBM MQ Explorer' den MSCS ile kullanmak üzere iki şekilde bir kuyruk yöneticisi oluşturabilirsiniz. Bir komut istemi kullanmanın yararı, kuyruk yöneticisinin *durdurulmuş* olması ve MSCS için hazır *el ile başlatma* olarak ayarlanmasıdır. ( IBM MQ Explorer otomatik olarak yeni bir kuyruk yöneticisini başlatır ve yaratıldıktan sonra otomatik başlatmaya ayarlar. Bunu değiştirmek zorundasınız.)

## Komut isteminden kuyruk yöneticisi yaratılması

MSCS ile kullanılmak üzere bir komut isteminden kuyruk yöneticisi yaratmak için aşağıdaki adımları izleyin:

1. MQSPREFIX ortam değişkeninin yerel bir sürücüye (örneğin, C : \IBM MQ) gönderme yapma üzere ayarlandığından emin olun. Bunu değiştirerseniz, Sistem hesabının değişikliği seçmesi için makineyi yeniden başlatın. Değişkeni ayarlamazsanız, kuyruk yöneticisi kuyruk yöneticileri için IBM MQ varsayılan dizininde yaratılır.
2. **crtmqm** komutunu kullanarak kuyruk yöneticisini yaratın. Örneğin, varsayılan dizinde mscs\_test adlı bir kuyruk yöneticisi yaratmak için şunu kullanın:

```
crtmqm mscs_test
```

3. “Kuyruk yöneticisinin MSCS depolama alanına taşınması” sayfa 482 başlıklı konuya geçin.

## IBM MQ Explorer kullanılarak kuyruk yöneticisi yaratılması

MSCS ile kullanılmak üzere IBM MQ Explorer komutunu kullanarak bir kuyruk yöneticisi oluşturmak için aşağıdaki adımları izleyin:

1. Başlat menüsünden IBM MQ Explorer komutunu başlatın.
2. Navigator görünümünde, ağaç düğümlerini genişleterek Kuyruk Yöneticileri ağaç düğümünü bulun.
3. Kuyruk Yöneticileri ağaç düğümünü sağ tıklayın ve **Yeni > Kuyruk Yöneticisi** seçeneğini belirleyin. Create Queue Manager (Kuyruk Yöneticisi Yarat) panosu görüntülenir.
4. İletişim kutusunu tamamlayın (Adım 1), ardından **İleri >** düğmesini tıklayın.
5. İletişim kutusunu tamamlayın (Adım 2), ardından **İleri >** düğmesini tıklayın.
6. Kuyruk Yöneticisini Başlat ve Sunucu Bağlantısı Kanalı Yarat seçeneklerinin belirlenmemesini sağlayarak iletişim kutusunu (Adım 3) tamamlayın ve **İleri >** düğmesini tıklayın.
7. İletişim kutusunu (Adım 4) tamamlayın ve **Son** düğmesini tıklayın.
8. [“Kuyruk yöneticisinin MSCS depolama alanına taşınması”](#) sayfa 482 başlıklı konuya geçin.

### Windows

### **Kuyruk yöneticisinin MSCS depolama alanına taşınması**

Bu yordam, var olan bir kuyruk yöneticisini Microsoft Cluster Service (MSCS) denetimine yerleştirmek için uygun olacak şekilde yapılandırır.

Bunu gerçekleştirmek için, günlük dosyalarını ve veri dosyalarını, bir hata durumunda diğer bilgisayarın kullanımına sunmak üzere paylaşılan disklere taşıyabilirsiniz. Örneğin, var olan kuyruk yöneticisinin C:\WebSphere MQ\Log\QMname ve C:\WebSphere MQ\qmgrs\QMname gibi yolları olabilir.



**Uyarı:** Dosyaları elle taşımayı denemeyin; bu konuda açıklandığı gibi IBM MQ MSCS Desteği 'nin bir parçası olarak sağlanan yardımcı programı kullanın.

Taşınmakta olan kuyruk yöneticisi TLS bağlantıları kullanıyorsa ve TLS anahtar havuzu yerel makinedeki kuyruk yöneticisi veri dizinindeyse, anahtar havuzu kuyruk yöneticisinin geri kalanıyla birlikte paylaşılan diske taşınır. Varsayılan olarak, TLS anahtar havuzu konumunu (SSLKEYR) belirten kuyruk yöneticisi özniteliği, kuyruk yöneticisi veri dizini altındaki MQ\_INSTALLATION\_PATH\qmgrs\QMGRNAME\ssl\keyolarak ayarlanır. MQ\_INSTALLATION\_PATH, IBM MQ 'in kurulu olduğu üst düzey dizini gösterir. hamvmqm komutu bu kuyruk yöneticisi özniteliğini değiştirmez. Bu durumda, IBM MQ Explorer ya da MQSC komutunu ALTER QMGR kullanarak SSLKEYR kuyruk yöneticisi özniteliğini, yeni TLS anahtar havuzu dosyasını gösterecek şekilde değiştirmeniz gerekir.

**Not:** Windows Server 2016' den Microsoft Cluster Service (MSCS) için yeni ad Windows Server Failover Clustering (WSFC) olur.

Yordam aşağıdaki gibidir:

1. Kuyruk yöneticisini kapatın ve hata olmadığını denetleyin.
2. Kuyruk yöneticisinin günlük dosyaları ya da kuyruk dosyaları önceden paylaşılan bir diskte saklanıyorsa, bu yordamın geri kalanını atlayın ve doğrudan [“Kuyruk yöneticisinin MSCS denetimi altına konması”](#) sayfa 483 başlıklı konuya geçin.
3. Kuyruk dosyalarının ve günlük dosyalarının tam ortam yedeğini alın ve yedeği güvenli bir yerde saklayın (bunun neden önemli olduğunu görmek için bkz. [“Kuyruk yöneticisi günlük dosyaları”](#) sayfa 492).
4. Önceden uygun bir paylaşılan disk kaynağınız varsa, 6. adıma ilerleyin. Ters durumda, kuyruk yöneticisi günlük dosyalarını ve veri (kuyruk) dosyalarını saklamak için yeterli kapasiteye sahip *paylaşılan disk* tipinde bir kaynak yaratmak için MSCS Cluster Administrator olanağını kullanın.
5. Paylaşılan diski bir küme düğümünden diğerine taşımak için MSCS Cluster Administrator olanağını kullanarak deneyin ve yeniden deneyin.
6. Paylaşılan diskin, kuyruk yöneticisi günlüğünün ve veri dosyalarının yerel olarak saklandığı küme düğümünde çevrimiçi olduğundan emin olun.
7. Kuyruk yöneticisini aşağıdaki gibi taşımak için yardımcı programı çalıştırın:

```
hamvmqm /m qmname /dd " e: \  
IBM MQ " /ld " e: \  
IBM MQ \log"
```

*qmname* için kuyruk yöneticisi adınızı, *eiçin* paylaşılan disk sürücüsü adınızı ve *IBM MQ* için seçtiğiniz dizini değiştirerek. Dizinler önceden yoksa yaratılır.

8. Kuyruk yöneticisini sınavın ve IBM MQ Explorer komutunu kullanarak çalıştığından emin olun. Örneğin:
  - a. Kuyruk yöneticisi ağaç düğümünü sağ tıklatın ve **Başlat** seçeneğini belirleyin. Kuyruk yöneticisi başlar.
  - b. Kuyruklar ağaç düğümünü sağ tıklatın ve **Yeni > Yerel Kuyruk ...**, ve kuyruğa bir ad verin.
  - c. **Bitir**'i tıklatın.
  - d. Kuyruğu sağ tıklatın ve **Test İletisi Koy ...** seçeneğini belirleyin. Put Test Message (Sınama İletisi Koy) panosu görüntülenir.
  - e. Bir ileti metni yazın ve **Test İletisi Koy** seçeneğini tıklatın ve panoyu kapatın.
  - f. Kuyruğu sağ tıklatın ve **İletilere Göz At ...** seçeneğini belirleyin. Message Browser (İleti Tarayıcısı) panosu görüntülenir.
  - g. İletinizin kuyrukta olduğundan emin olun ve **Kapat**'ı tıklatın. Message Browser (İleti Tarayıcısı) panosu kapanır.
  - h. Kuyruğu sağ tıklatın ve **İletileri Temizle ...** seçeneğini belirleyin. Kuyruktaki iletiler temizlenir.
  - i. Kuyruğu sağ tıklatın ve **Sil ...** seçeneğini belirleyin. Bir onay panosu görüntülenir, **Tamam**'ı tıklatın. Kuyruk silindi.
  - j. Kuyruk yöneticisi ağaç düğümünü sağ tıklatın ve **Durdur ...** seçeneğini belirleyin. End Queue Manager (Kuyruk Yöneticisini Sona Erdir) panosu görüntülenir.
  - k. **Tamam**'ı tıklatın. Kuyruk yöneticisi durur.
9. IBM MQ Yöneticisi olarak, kuyruk yöneticisinin başlatma özneliğinin el ile olarak ayarlandığından emin olun. IBM MQ Explorer içinde, kuyruk yöneticisi özellikleri panosunda Başlatma alanını manual olarak ayarlayın.
10. ["Kuyruk yöneticisinin MSCS denetimi altına konması" sayfa 483](#) başlıklı konuya geçin.

### **Windows** *Kuyruk yöneticisinin MSCS denetimi altına konması*

Önkoşul görevleri de içinde olmak üzere, kuyruk yöneticisinin Microsoft Cluster Service (MSCS) denetimi altına yerleştirilmesi.

**Not:** Windows Server 2016' den Microsoft Cluster Service (MSCS) için yeni ad Windows Server Failover Clustering (WSFC) olur.

### **Bir kuyruk yöneticisini MSCS/WSFC denetimine yerleştirmeden önce**

Bir kuyruk yöneticisini MSCS/WSFC denetimine yerleştirmeden önce aşağıdaki adımları tamamlayın:

1. IBM MQ ve MSCS/WSFC desteğinin kümedeki her iki makineye de kurulduğundan ve her bir bilgisayardaki yazılımın ["MSCS kümelemesi için IBM MQ ' nin ayarlanması" sayfa 479](#) içinde açıklandığı şekilde aynı olduğundan emin olun.
2. IBM MQ ' i tüm küme düğümlerinde MSCS kaynak tipi olarak kaydetmek için **haretyp** yardımcı programını kullanın. Bkz. ["MSCS yardımcı programları için destek" sayfa 494](#).
3. Henüz yapmadıysanız, [MSCS/WSFC ile kullanılmak üzere bir kuyruk yöneticisi oluşturun](#).
4. Kuyruk yöneticisini yarattıysanız ya da zaten varsa, yordamı ["Kuyruk yöneticisinin MSCS depolama alanına taşınması" sayfa 482](#) içinde gerçekleştirdiğinizden emin olun.
5. Kuyruk yöneticisi çalışıyorsa, bir komut istemi ya da IBM MQ Explorer kullanarak bunu durdurun.
6. Bu konuda aşağıdaki Windows yordamlarından birine geçmeden önce paylaşılan sürücülerin MSCS/WSFC çalışmasını sınavın.



## Windows Server 2012, 2016, 2019 ya da 2022

Bir kuyruk yöneticisini Windows Server 2012 ya da sonraki bir sunucuda MSCS/WSFC denetimi altına yerleştirmek için aşağıdaki yordamı kullanın:

1. Kuyruk yöneticisini barındıran küme düğümü bilgisayarında oturum açın ya da küme yönetimi izinlerine sahip bir kullanıcı olarak uzak bir iş istasyonunda oturum açın ve kuyruk yöneticisini barındıran küme düğümüne bağlanın.
2. Hata Durumunda Yedek Sisteme Geçiş Küme Yönetimi aracını başlatın.
3. **Failover Cluster Management > Connect Cluster ...** (Hata Durumunda Yedek Sisteme Geçiş Kümesi Yönetimi > Kümeyi Bağla ...) bir kümeye bağlantı açmak için.
4. Windows' in önceki sürümlerinde MSCS Cluster Administrator 'da kullanılan grup şemasının tersine, Failover Cluster Management aracı hizmet ve uygulama kavramını kullanır. Yapılandırılmış bir hizmet ya da uygulama, bir uygulamanın kümelenebilmesi için gereken tüm kaynakları içerir. WSFC altında bir kuyruk yöneticisini aşağıdaki gibi yapılandırabilirsiniz:

- a. Yapılanış sihirbazını başlatmak için kümeyi farenin sağ düğmesiyle tıklatın ve **Rol Yapılandır** seçeneğini belirleyin.
- b. "Select Service or Application" (Hizmet ya da Uygulama Seç) panosunda **Other Server** (Diğer Sunucu) seçeneğini belirleyin.
- c. İstemci erişim noktası olarak uygun bir IP adresi seçin.

Bu adres, istemciler ve diğer kuyruk yöneticileri tarafından *sanal* kuyruk yöneticisine bağlanmak için kullanılacak kullanılmayan bir IP adresi olmalıdır. Bu IP adresi, her iki düğümün olağan (statik) adresi değildir; bu, aralarında *kayan* ek bir adrestir. WSFC, bu adresin yönlendirmesini işlese de, adrese ulaşılabilirliğini **doğrulamaz** .

- d. Kuyruk yöneticisi tarafından özel kullanım için bir depolama aygıtı atayın. Bu aygıtın atanması için önce bir kaynak örneği olarak yaratılması gerekir.

Hem günlükleri hem de kuyruk dosyalarını depolamak için bir sürücü kullanabilir ya da bunları sürücüler arasında bölebilirsiniz. Her iki durumda da, her kuyruk yöneticisinin kendi paylaşılan diski varsa, bu kuyruk yöneticisi tarafından kullanılan tüm sürücülerin bu kuyruk yöneticisine özel olduğundan, yani sürücülere başka hiçbir şeyin güvenmediğinden emin olun. Ayrıca, kuyruk yöneticisinin kullandığı her sürücü için bir kaynak eşgörünümü yarattığınızdan emin olun.

Bir sürücünün kaynak tipi, kullandığınız SCSI desteğine bağlıdır; SCSI bağdaştırıcısı yönergelerinize bakın. Paylaşılan sürücülerin her biri için önceden gruplar ve kaynaklar olabilir. Bu durumda, her sürücü için kaynak yönetim ortamı yaratmanıza gerek yoktur. Bunu yürürlükteki grubundan, kuyruk yöneticisi için yaratılan gruba taşıyın.

Her bir sürücü kaynağı için olası sahipleri her iki düğümüne ayarlayın. Bağımlı kaynakları yok olarak ayarlayın.

- e. "Kaynak Tipi Seç" panosunda **MQSeries MSCS** kaynağını seçin.
- f. Sihirbazda kalan adımları tamamlayın.

5. Kaynağı çevrimiçi duruma getirmeden önce MQSeries MSCS kaynağı için ek yapılandırma gerekir:

- a. 'Yeni MQSeries MSCS' adlı bir kaynak içeren yeni tanımlanan hizmeti seçin.
- b. MQ kaynağında **Özellikler** ögesini sağ tıklatın.
- c. Kaynağı yapılandır:

- Name ; Hangi kuyruk yöneticisi için olduğunu saptamanızı kolaylaştıran bir ad seçin.
- Run in a separate Resource Monitor ; daha iyi yalıtım için
- Possible owners ; her iki düğümü de ayarla
- Dependencies ; Bu kuyruk yöneticisine ilişkin sürücüyü ve IP adresini ekleyin.

**Uyarı:** Bu bağımlılıkların eklenmemesi, IBM MQ ' in hata durumunda yedek sisteme geçiş sırasında kuyruk yöneticisi durumunu yanlış küme diskine yazmayı denediği anlamına gelir. Birçok



işlem aynı anda bu diske yazma girişiminde bulunduğundan, bazı IBM MQ işlemlerinin çalışması engellenebilir.

• **Parameters** ; Aşağıdaki gibi:

- **QueueManagerName** (gerekli); bu kaynağın denetleyeceği kuyruk yöneticisinin adı. Bu kuyruk yöneticisi yerel bilgisayarda var olmalıdır.
- **PostOnlineCommand** (isteğe bağlı); kuyruk yöneticisi kaynağı çevrimdışıdan çevrimiçi durumuna her değiştiğinde çalıştırılacak bir program belirtebilirsiniz. Daha fazla ayrıntı için bkz. [“PostOnlineKomutu ve MSCS ' de PreOfflineKomutu” sayfa 493.](#)
- **PreOfflineCommand** (isteğe bağlı); kuyruk yöneticisi kaynağı çevrimiçi durumundan çevrimdışı durumuna her değiştiğinde çalıştırılacak bir program belirtebilirsiniz. Daha fazla ayrıntı için bkz. [“PostOnlineKomutu ve MSCS ' de PreOfflineKomutu” sayfa 493.](#)

**Not:** *looksAlive* yoklama aralığı, varsayılan 5000 ms değerine ayarlanır. *isAlive* yoklama aralığı 60000 ms varsayılan değerine ayarlanır. Bu varsayılan değerler yalnızca kaynak tanımlaması tamamlandıktan sonra değiştirilebilir. Daha fazla ayrıntı için bkz. [“MSCS ' de looksAlive ve isAlive yoklaması” sayfa 489.](#)

- d. İsteğe bağlı olarak, tercih edilen bir düğüm ayarlayın (ancak [“MSCS ' de tercih edilen düğümlerin kullanılması” sayfa 493](#) içindeki yorumları not edin)
  - e. *Yedek Sisteme Geçiş İlkesi* varsayılan olarak duyarlı değerlere ayarlanır, ancak *Hata Durumunda Yedek Sisteme Geçiş* ve *Hata Durumunda Yedek Sisteme Geçiş Grubu ' nu denetleyen eşikler ve dönemler*, kuyruk yöneticisine yerleştirilen yüklerle eşleşecek şekilde ayarlanabilir.
6. Kuyruk yöneticisini MSCS Cluster Administrator 'da çevrimiçi duruma getirerek ve bir test iş yüküne özneleyerek test edin. Bir test kuyruğu yöneticisiyle deneyler yapıyorsanız, IBM MQ Explorer 'ı kullanın. Örneğin:
- a. Kuyruklar ağaç düğümünü sağ tıklayın ve **Yeni > Yerel Kuyruk ...**, ve kuyruğa bir ad verin.
  - b. **Bitir**'i tıklayın. Kuyruk yaratılır ve içerik görünümünde görüntülenir.
  - c. Kuyruğu sağ tıklayın ve **Test İletisi Koy ...**seçeneğini belirleyin. Put Test Message (Sınama İletisi Koy) panosu görüntülenir.
  - d. Bir ileti metni yazın ve **Test İletisi Koy**seçeneğini tıklayın ve panoyu kapatın.
  - e. Kuyruğu sağ tıklayın ve **İletilere Göz At ...**seçeneğini belirleyin. Message Browser (İleti Tarayıcısı) panosu görüntülenir.
  - f. İletinizin kuyruқта olduğundan emin olun ve **Kapat**'ı tıklayın. Message Browser (İleti Tarayıcısı) panosu kapanır.
  - g. Kuyruğu sağ tıklayın ve **İletileri Temizle ...**seçeneğini belirleyin. Kuyruktaki iletiler temizlenir.
  - h. Kuyruğu sağ tıklayın ve **Sil ...**seçeneğini belirleyin. Bir onay panosu görüntülenir, **Tamam**'ı tıklayın. Kuyruk silindi.
7. Kuyruk yöneticisinin MSCS Küme Denetimcisi 'ni kullanarak çevrimdışı ve yeniden çevrimiçi duruma getirilip getirilebileceğini denetleyin.
8. Yedek sisteme geçiş simülasyonu yap.
- MSCS Cluster Administrator 'da, kuyruk yöneticisini içeren grubu sağ tıklayın ve **Move Group**seçeneğini belirleyin. Bu işlem birkaç dakika sürebilir. (Başka zamanlarda bir kuyruk yöneticisini hızlı bir şekilde başka bir düğüme taşımak istiyorsanız, [“Kuyruk yöneticisinin MSCS depolama alanına taşınması” sayfa 482](#)başlıklı konudaki yordamı izleyin.) Ayrıca, farenin sağ düğmesini tıklayıp **Initiate Failure** ; İşlem (yerel yeniden başlatma ya da yedek sisteme geçiş) geçerli duruma ve yapılandırma ayarlarına bağlıdır.

## Windows Server 2008 (Sunucu 2008)

Bir kuyruk yöneticisini Windows Server 2008 'de MSCS denetimi altına yerleştirmek için aşağıdaki yordamı kullanın:

1. Kuyruk yöneticisini barındıran küme düğümü bilgisayarında oturum açın ya da küme yönetimi izinlerine sahip bir kullanıcı olarak uzak bir iş istasyonunda oturum açın ve kuyruk yöneticisini barındıran küme düğümüne bağlanın.
2. Hata Durumunda Yedek Sisteme Geçiş Küme Yönetimi aracını başlatın.
3. **Failover Cluster Management > Manage A Cluster ...** (Yedek Sisteme Geçiş Kümesi Yönetimi > Kümeyi Yönet ...) seçeneklerini sağ tıklayın. bir kümeye bağlantı açmak için.
4. Windows' in önceki sürümlerinde MSCS Cluster Administrator 'da kullanılan grup şemasının tersine, Failover Cluster Management aracı hizmet ve uygulama kavramını kullanır. Yapılandırılmış bir hizmet ya da uygulama, bir uygulamanın kümelenebilmesi için gereken tüm kaynakları içerir. MSCS altında bir kuyruk yöneticisini aşağıdaki gibi yapılandırabilirsiniz:
  - a. **Hizmetler ve Uygulamalar > Hizmet ya da Uygulama Yapılandır ...** öğelerini sağ tıklayın. Yapılandırma sihirbazını başlatmak için.
  - b. **Select Service or Application** (Hizmet ya da Uygulama Seç) panosunda **Other Server** (Diğer Sunucu) seçeneğini belirleyin.
  - c. İstemci erişim noktası olarak uygun bir IP adresi seçin.

Bu adres, istemciler ve diğer kuyruk yöneticileri tarafından *sanal* kuyruk yöneticisine bağlanmak için kullanılacak kullanılmayan bir IP adresi olmalıdır. Bu IP adresi, her iki düğümün olağan (statik) adresi değildir; bu, aralarında *kayan* ek bir adrestir. MSCS bu adresin yönlendirmesini işlese de, adrese ulaşabildiğini **doğrulamaz** .
  - d. Kuyruk yöneticisi tarafından özel kullanım için bir depolama aygıtı atayın. Bu aygıtın atanması için önce bir kaynak örneği olarak yaratılması gerekir.

Hem günlükleri hem de kuyruk dosyalarını depolamak için bir sürücü kullanabilir ya da bunları sürücüler arasında bölebilirsiniz. Her iki durumda da, her kuyruk yöneticisinin kendi paylaşılan diski varsa, bu kuyruk yöneticisi tarafından kullanılan tüm sürücülerin bu kuyruk yöneticisine özel olduğundan, yani sürücülere başka hiçbir şeyin güvenmediğinden emin olun. Ayrıca, kuyruk yöneticisinin kullandığı her sürücü için bir kaynak eşgörünümü yarattığınızdan emin olun.

Bir sürücünün kaynak tipi, kullandığınız SCSI desteğine bağlıdır; SCSI bağdaştırıcısı yönergelerinize bakın. Paylaşılan sürücülerin her biri için önceden gruplar ve kaynaklar olabilir. Bu durumda, her sürücü için kaynak yönetim ortamı yaratmanıza gerek yoktur. Bunu yürürlükteki grubundan, kuyruk yöneticisi için yaratılan gruba taşıyın.

Her bir sürücü kaynağı için olası sahipleri her iki düğümüne ayarlayın. Bağımlı kaynakları yok olarak ayarlayın.
  - e. **Kaynak Tipi Seç** panosunda **MQSeries MSCS** kaynağını seçin.
  - f. Sihirbazda kalan adımları tamamlayın.
5. Kaynağı çevrimiçi duruma getirmeden önce MQSeries MSCS kaynağı için ek yapılandırma gerekir:
  - a. 'Yeni MQSeries MSCS' adlı bir kaynak içeren yeni tanımlanan hizmeti seçin.
  - b. MQ kaynağında **Özellikler** öğesini sağ tıklayın.
  - c. Kaynağı yapılandır:
    - Name ; Hangi kuyruk yöneticisi için olduğunu saptamanızı kolaylaştıran bir ad seçin.
    - Run in a separate Resource Monitor ; daha iyi yalıtım için
    - Possible owners ; her iki düğümü de ayarla
    - Dependencies ; Bu kuyruk yöneticisine ilişkin sürücüyü ve IP adresini ekleyin.

**Uyarı:** Bu bağımlılıkların eklenmemesi, IBM MQ ' in hata durumunda yedek sisteme geçiş sırasında kuyruk yöneticisi durumunu yanlış küme diskine yazmayı denediği anlamına gelir. Birçok işlem aynı anda bu diske yazma girişiminde bulunduğundan, bazı IBM MQ işlemlerinin çalışması engellenebilir.

    - Parameters ; Aşağıdaki gibi:

- QueueManagerName (gerekli); bu kaynağın denetleyeceği kuyruk yöneticisinin adı. Bu kuyruk yöneticisi yerel bilgisayarda var olmalıdır.
- PostOnlineCommand (isteğe bağlı); kuyruk yöneticisi kaynağı çevrimdışıdan çevrimiçi durumuna her değiştiğinde çalıştırılacak bir program belirtebilirsiniz. Daha fazla ayrıntı için bkz. [“PostOnlineKomutu ve MSCS ' de PreOfflineKomutu” sayfa 493.](#)
- PreOfflineCommand (isteğe bağlı); kuyruk yöneticisi kaynağı çevrimiçi durumundan çevrimdışı durumuna her değiştiğinde çalıştırılacak bir program belirtebilirsiniz. Daha fazla ayrıntı için bkz. [“PostOnlineKomutu ve MSCS ' de PreOfflineKomutu” sayfa 493.](#)

**Not:** *looksAlive* yoklama aralığı, varsayılan 5000 ms değerine ayarlanır. *isAlive* yoklama aralığı 60000 ms varsayılan değerine ayarlanır. Bu varsayılan değerler yalnızca kaynak tanımlaması tamamlandıktan sonra değiştirilebilir. Daha fazla ayrıntı için bkz. [“MSCS ' de looksAlive ve isAlive yoklaması” sayfa 489.](#)

- d. İsteğe bağlı olarak, tercih edilen bir düğüm ayarlayın (ancak [“MSCS ' de tercih edilen düğümlerin kullanılması” sayfa 493](#) içindeki yorumları not edin)
  - e. *Yedek Sisteme Geçiş İlkesi* varsayılan olarak duyarlı değerlere ayarlanır, ancak *Hata Durumunda Yedek Sisteme Geçiş ve Hata Durumunda Yedek Sisteme Geçiş Grubu ' nu* denetleyen eşikler ve dönemler, kuyruk yöneticisine yerleştirilen yüklerle eşleşecek şekilde ayarlanabilir.
6. Kuyruk yöneticisini MSCS Cluster Administrator 'da çevrimiçi duruma getirerek ve bir test iş yüküne özneleyerek test edin. Bir test kuyruğu yöneticisiyle deneyler yapıyorsanız, IBM MQ Explorer 'ı kullanın. Örneğin:
- a. Kuyruklar ağaç düğümünü sağ tıklayın ve **Yeni > Yerel Kuyruk ...**, ve kuyruğa bir ad verin.
  - b. **Bitir**'i tıklayın. Kuyruk yaratılır ve içerik görünümünde görüntülenir.
  - c. Kuyruğu sağ tıklayın ve **Test İletisi Koy ...**seçeneğini belirleyin. **Put Test Message** (Sınama İletisi Koy) panosu görüntülenir.
  - d. Bir ileti metni yazın ve **Test İletisi Koy**seçeneğini tıklayın ve panoyu kapatın.
  - e. Kuyruğu sağ tıklayın ve **İletilere Göz At ...**seçeneğini belirleyin. **Message Browser** (İleti Tarayıcısı) panosu görüntülenir.
  - f. İletinizin kuyrukta olduğundan emin olun ve **Kapat**' ı tıklayın. **Message Browser** (İleti Tarayıcısı) panosu kapanır.
  - g. Kuyruğu sağ tıklayın ve **İletileri Temizle ...**seçeneğini belirleyin. Kuyruktaki iletiler temizlenir.
  - h. Kuyruğu sağ tıklayın ve **Sil ...**seçeneğini belirleyin. Bir onay panosu görüntülenir, **Tamam**' ı tıklayın. Kuyruk silindi.
7. Kuyruk yöneticisinin MSCS Küme Denetimcisi 'ni kullanarak çevrimdışı ve yeniden çevrimiçi duruma getirilip getirilebileceğini denetleyin.
8. Yedek sisteme geçiş simülasyonu yap.

MSCS Cluster Administrator 'da, kuyruk yöneticisini içeren grubu sağ tıklayın ve **Move Group**seçeneğini belirleyin. Bu işlem birkaç dakika sürebilir. (Başka zamanlarda bir kuyruk yöneticisini hızlı bir şekilde başka bir düğüme taşımak istiyorsanız, [“Kuyruk yöneticisinin MSCS depolama alanına taşınması” sayfa 482](#)başlıklı konudaki yordamı izleyin.) Ayrıca, farenin sağ düğmesini tıklayıp **Initiate Failure** ; İşlem (yerel yeniden başlatma ya da yedek sisteme geçiş) geçerli duruma ve yapılandırma ayarlarına bağlıdır.

## Windows 2003

Bir kuyruk yöneticisini Windows 2003üzerinde MSCS denetimi altına yerleştirmek için aşağıdaki yordamı kullanın:

1. Kuyruk yöneticisini barındıran küme düğümü bilgisayarında oturum açın ya da küme yönetimi izinlerine sahip bir kullanıcı olarak uzak bir iş istasyonunda oturum açın ve kuyruk yöneticisini barındıran küme düğümüne bağlanın.
2. MSCS Küme Denetimcisini başlatın.
3. Kümeyle bir bağlantı açın.

4. Kuyruk yöneticisine ilişkin kaynakları içermek için kullanılacak bir MSCS grubu yaratın. Grubu, hangi kuyruk yöneticisiyle ilişkili olduğu belli olacak şekilde adlandırın. Her grup, “MSCS ile birden çok kuyruk yöneticisinin kullanılması” sayfa 480 içinde açıklandığı gibi birden çok kuyruk yöneticisi içerebilir.

Kalan tüm adımlar için grubu kullanın.

5. Kuyruk yöneticisinin kullandığı her bir SCSI mantıksal sürücüsü için bir kaynak örneği oluşturun.

Hem günlükleri hem de kuyruk dosyalarını depolamak için bir sürücü kullanabilir ya da bunları sürücüler arasında bölebilirsiniz. Her iki durumda da, her kuyruk yöneticisinin kendi paylaşılan diski varsa, bu kuyruk yöneticisi tarafından kullanılan tüm sürücülerin bu kuyruk yöneticisine özel olduğundan, yani sürücülere başka hiçbir şeyin güvenmediğinden emin olun. Ayrıca, kuyruk yöneticisinin kullandığı her sürücü için bir kaynak eşgörünümü yarattığınızdan emin olun.

Bir sürücünün kaynak tipi, kullandığınız SCSI desteğine bağlıdır; SCSI bağdaştırıcısı yönergelerinize bakın. Paylaşılan sürücülerin her biri için önceden gruplar ve kaynaklar olabilir. Bu durumda, her sürücü için kaynak yönetim ortamı yaratmanıza gerek yoktur. Bunu yürürlükteki grubundan, kuyruk yöneticisi için yaratılan gruba taşıyın.

Her bir sürücü kaynağı için olası sahipleri her iki düğüme ayarlayın. Bağımlı kaynakları yok olarak ayarlayın.

6. IP adresi için bir kaynak örneği oluşturun.

Bir IP adresi kaynağı yaratın (kaynak tipi *IP adresi*). Bu adres, istemciler ve diğer kuyruk yöneticileri tarafından *sanal* kuyruk yöneticisine bağlanmak için kullanılacak kullanılmayan bir IP adresi olmalıdır. Bu IP adresi, her iki düğümün olağan (statik) adresi değildir; bu, aralarında *kayan* ek bir adrestir. MSCS bu adresin yönlendirmesini işlese de, adrese ulaşılabildiğini **doğrulamaz**.

7. Kuyruk yöneticisi için bir kaynak eşgörünümü yaratın.

*IBM MQ MSCS* tipinde bir kaynak oluşturun. Sihirbaz, aşağıdakiler de içinde olmak üzere çeşitli öğeler için sizden bilgi ister:

- Name ; Hangi kuyruk yöneticisi için olduğunu saptamanızı kolaylaştıran bir ad seçin.
- Add to group ; oluşturduğunuz grubu kullanma
- Run in a separate Resource Monitor ; daha iyi yalıtım için
- Possible owners ; her iki düğümü de ayarla
- Dependenc ies ; Bu kuyruk yöneticisine ilişkin sürücüyü ve IP adresini ekleyin.

**Uyarı:** Bu bağımlılıkların eklenmemesi, IBM MQ ' in hata durumunda yedek sisteme geçiş sırasında kuyruk yöneticisi durumunu yanlış küme diskinde yazmayı denediği anlamına gelir. Birçok işlem aynı anda bu diske yazma girişiminde bulunduğundan, bazı IBM MQ işlemlerinin çalışması engellenebilir.

- Parameters ; Aşağıdaki gibi:

- QueueManagerName (gerekli); bu kaynağın denetleyeceği kuyruk yöneticisinin adı. Bu kuyruk yöneticisi yerel bilgisayarda var olmalıdır.
- PostOnlineCommand (isteğe bağlı); kuyruk yöneticisi kaynağı çevrimdışıdan çevrimiçi durumuna her değiştiğinde çalıştırılacak bir program belirtebilirsiniz. Daha fazla ayrıntı için bkz. “PostOnlineKomutu ve MSCS ' de PreOfflineKomutu” sayfa 493.
- PreOfflineCommand (isteğe bağlı); kuyruk yöneticisi kaynağı çevrimiçi durumundan çevrimdışı durumuna her değiştiğinde çalıştırılacak bir program belirtebilirsiniz. Daha fazla ayrıntı için bkz. “PostOnlineKomutu ve MSCS ' de PreOfflineKomutu” sayfa 493.

**Not:** *looksAlive* yoklama aralığı, varsayılan 5000 ms değerine ayarlanır. *isAlive* yoklama aralığı 30000 ms varsayılan değerine ayarlanır. Bu varsayılan değerler yalnızca kaynak tanımlaması tamamlandıktan sonra değiştirilebilir. Daha fazla ayrıntı için bkz. “MSCS ' de looksAlive ve isAlive yoklaması” sayfa 489.

8. İsteğe bağlı olarak, tercih edilen bir düğüm ayarlayın (ancak “MSCS ' de tercih edilen düğümlerin kullanılması” sayfa 493 içindeki yorumları not edin)

9. *Yedek Sisteme Geçiş İlkesi* (grubun özelliklerinde tanımlandığı şekilde) varsayılan olarak duyarlı değerlere ayarlanır, ancak *Hata Durumunda Yedek Sisteme Geçiş ve Hata Durumunda Yedek Gruba Geçiş* öğelerini denetleyen eşikleri ve dönemleri, kuyruk yöneticisine yerleştirilen yüklerle eşleşecek şekilde ayarlayabilirsiniz.
10. Kuyruk yöneticisini MSCS Cluster Administrator 'da çevrimiçi duruma getirerek ve bir test iş yüküne özneyerek test edin. Bir test kuyruğu yöneticisiyle deneyler yapıyorsanız, IBM MQ Explorer 'ı kullanın. Örneğin:
- Kuyruklar ağaç düğümünü sağ tıklayın ve **Yeni > Yerel Kuyruk ...**, ve kuyruğa bir ad verin.
  - Bitir**'i tıklayın. Kuyruk yaratılır ve içerik görünümünde görüntülenir.
  - Kuyruğu sağ tıklayın ve **Test İletisi Koy ...**seçeneğini belirleyin. **Put Test Message** (Sınama İletisi Koy) panosu görüntülenir.
  - Bir ileti metni yazın ve **Test İletisi Koy**seçeneğini tıklayın ve panoyu kapatın.
  - Kuyruğu sağ tıklayın ve **İletilere Göz At ...**seçeneğini belirleyin. **Message Browser** (İleti Tarayıcısı) panosu görüntülenir.
  - İletinizin kuyrukta olduğundan emin olun ve **Kapat** 'ı tıklayın. **Message Browser** (İleti Tarayıcısı) panosu kapanır.
  - Kuyruğu sağ tıklayın ve **İletileri Temizle ...**seçeneğini belirleyin. Kuyruktaki iletiler temizlenir.
  - Kuyruğu sağ tıklayın ve **Sil ...**seçeneğini belirleyin. Bir onay panosu görüntülenir, **Tamam** 'ı tıklayın. Kuyruk silindi.
11. Kuyruk yöneticisinin MSCS Küme Denetimcisi 'ni kullanarak çevrimdışı ve yeniden çevrimiçi duruma getirilip getirilebileceğini denetleyin.
12. Yedek sisteme geçiş simülasyonu yap.
- MSCS Cluster Administrator 'da, kuyruk yöneticisini içeren grubu sağ tıklayın ve Move Groupseçeneğini belirleyin. Bu işlem birkaç dakika sürebilir. (Başka zamanlarda bir kuyruk yöneticisini hızlı bir şekilde başka bir düğüme taşımak istiyorsanız, "[Kuyruk yöneticisinin MSCS depolama alanına taşınması](#)" sayfa 482başlıklı konudaki yordamı izleyin.) Ayrıca, farenin sağ düğmesini tıklayıp **Initiate Failure** ; İşlem (yerel yeniden başlatma ya da yedek sisteme geçiş) geçerli duruma ve yapılandırma ayarlarına bağlıdır.

### **Windows** **MSCS ' de looksAlive ve isAlive yoklaması**

*looksAlive* ve *isAlive* , Microsoft Cluster Service (MSCS) tarafından sağlanan kitaplık koduna geri çağrılan aralıklardır ve kaynağın kendi çalışma durumunu belirlemek için denetleme gerçekleştirmesini istediği isteklerdir. Bu sonuç, MSCS ' nin kaynağı devreden çıkarmayı deneyip denemeyeceğini belirler.

**Not:** Windows Server 2016' den Microsoft Cluster Service (MSCS) için yeni ad Windows Server Failover Clustering (WSFC) olur.

*looksAlive* aralığının her defasında (varsayılan 5000 ms), kuyruk yöneticisi kaynağı, durumunun tatmin edici olup olmadığını belirlemek üzere kendi denetimini gerçekleştirmek için çağrılır.

*isAlive* aralığının her geçişinde (varsayılan 30000 ms), kaynağın doğru çalışıp çalışmadığını saptamak için başka bir denetim gerçekleştirmesi için kuyruk yöneticisi kaynağına başka bir çağrı yapılır. Bu, iki kaynak tipi denetimi düzeyini etkinleştirir.

- Kaynağın çalışıp çalışmadığını belirlemek için *looksAlive* durum denetimi.
- Kuyruk yöneticisi kaynağının etkin olup olmadığını belirleyen daha önemli bir *isAlive* denetimi.

Kuyruk yöneticisi kaynağının etkin olmadığı belirlenirse, MSCS, diğer ileri düzey MSCS seçeneklerine dayalı olarak, kaynak ve ilişkili bağımlı kaynaklar için kümedeki başka bir düğümde yedek sisteme geçiş işlemini tetikler. Daha fazla bilgi için bkz. [MSCS belgeleri](#).

### **Windows** **Kuyruk yöneticisinin MSCS denetiminden kaldırılması**

Kuyruk yöneticilerini Microsoft Cluster Service (MSCS) denetiminden kaldırabilir ve el ile yönetime geri döndürebilirsiniz.

**Not:** Windows Server 2016' den Microsoft Cluster Service (MSCS) için yeni ad Windows Server Failover Clustering (WSFC) olur.

Bakım işlemleri için MSCS denetiminden kuyruk yöneticilerini kaldırmanız gerekmez. MSCS Küme Deneticisi 'ni kullanarak bir kuyruk yöneticisini geçici olarak çevrimdışı duruma alarak bunu yapabilirsiniz. Bir kuyruk yöneticisinin MSCS denetiminden kaldırılması daha kalıcı bir değişikliktir; yalnızca, MSCS ' nin kuyruk yöneticisi üzerinde daha fazla denetime sahip olmasını istemediğinize karar verirsiniz bunu yapın.

Kaldırılan kuyruk yöneticisi TSL bağlantılarını kullanıyorsa, IBM MQ Explorer ya da MQSC komutunu ALTER QMGRkullanarak, yerel dizindeki TLS anahtar havuzu dosyasını gösterecek şekilde kuyruk yöneticisi öznitelğini (SSLKEYR) değiştirmeniz gerekir.

Yordam şöyledir:

1. Kuyruk yöneticisi kaynağını MSCS Küme Deneticisi 'ni kullanarak çevrimdışı duruma getirin; açıklamalar için bkz. [“MSCS ' den bir kuyruk yöneticisini çevrimdışı duruma getirme” sayfa 490](#)
2. Kaynak somut örneğini yok edin. Bu, kuyruk yöneticisini yok etmez.
3. İsteğe bağlı olarak, kuyruk yöneticisi dosyalarını paylaşılan sürücülerden yerel sürücülere geri geçirin. Bunu yapmak için bkz. [“Kuyruk yöneticisinin MSCS depolamasından geri döndürülmesi” sayfa 490.](#)
4. Kuyruk yöneticisini sınavın.

## MSCS ' den bir kuyruk yöneticisini çevrimdışı duruma getirme

Bir kuyruk yöneticisini MSCS ' den çevrimdışı duruma getirmek için aşağıdaki adımları izleyin:

1. MSCS Küme Deneticisini başlatın.
2. Kümeyle bir bağlantı açın.
3. Groupsya da Windows 2012 kullanıyorsanız Role seçeneğini belirleyin ve taşınacak kuyruk yöneticisini içeren grubu açın.
4. Kuyruk yöneticisi kaynağını seçin.
5. Sağ tıklayın ve Offline seçeneğini belirleyin.
6. Tamamlanmasını bekleyin.

## Kuyruk yöneticisinin MSCS depolamasından geri döndürülmesi

Bu yordam, kuyruk yöneticisini bilgisayarının yerel sürücüsüne, yani *normal* IBM MQ bir kuyruk yöneticisine dönüşecek şekilde yapılandırır. Bunu gerçekleştirmek için günlük dosyalarını ve veri dosyalarını paylaşılan disklerden taşıyabilirsiniz. Örneğin, var olan kuyruk yöneticisinin E : \WebSphere MQ \log \QMname ve E : \WebSphere MQ \qmgrs \QMname gibi yolları olabilir. Dosyaları elle taşımayı denemeyin; IBM MQ MSCS Desteği 'nin bir parçası olarak sağlanan **hamvmqm** yardımcı programını kullanın:

1. Kuyruk dosyalarının ve günlük dosyalarının tam ortam yedeğini alın ve yedeği güvenli bir yerde saklayın (bunun neden önemli olduğunu görmek için bkz. [“Kuyruk yöneticisi günlük dosyaları” sayfa 492](#) ).
2. Hangi yerel sürücünün kullanılacağına karar verin ve kuyruk yöneticisi günlük dosyalarını ve veri (kuyruk) dosyalarını saklamak için yeterli kapasiteye sahip olduğundan emin olun.
3. Dosyaların bulunduğu paylaşılan diskin, kuyruk yöneticisi günlüğünün ve veri dosyalarının taşınacağı küme düğümünde çevrimiçi olduğundan emin olun.
4. Kuyruk yöneticisini aşağıdaki gibi taşımak için yardımcı programı çalıştırın:

```
hamvmqm /m qmname /dd " c:\
IBM MQ " /ld "c:\
IBM MQ \log"
```

*qmname* için kuyruk yöneticisi adınızı, için yerel disk sürücüsü adınızı ve *IBM MQ* için seçtiğiniz dizini değiştirerek (dizinler yoksa yaratılır).

5. Kuyruk yöneticisini sınavın ve çalıştığından emin olun (bkz. [“Kuyruk yöneticisinin MSCS depolama alanına taşınması” sayfa 482](#)).

### **Windows** *MSCS kullanımına ilişkin ipuçları ve öneriler*

Bu bölümde, Microsoft Cluster Service (MSCS) için IBM MQ desteğini etkili bir şekilde kullanmanıza yardımcı olacak bazı genel bilgiler yer alır.

**Not:** Windows Server 2016' den Microsoft Cluster Service (MSCS) için yeni ad Windows Server Failover Clustering (WSFC) olur.

Bir kuyruk yöneticisini bir makineden diğerine geçirmesi ne kadar sürer? Bu, büyük ölçüde kuyruk yöneticisindeki iş yükü miktarına ve trafik karışımına bağlıdır; örneğin, bunun ne kadarının kalıcı, eşitleme noktası içinde ve hatadan önce ne kadar kesinleştirildiğine. IBM sınamaları, hata durumunda yedek sisteme geçiş ve onarım sonrası geri dönüş sürelerini yaklaşık bir dakika olarak verdi. Bu çok hafif yüklenmiş bir kuyruk yöneticisindeydi ve gerçek saatler yüke bağlı olarak önemli ölçüde değişecektir.

### **Windows** *MSCS 'nin çalıştığını doğrulama*

Çalışan bir MSCS kümeniz olduğundan emin olmak için aşağıdaki adımları izleyin.

“MSCS ile kullanılmak üzere kuyruk yöneticisi yaratılması” sayfa 481 ile başlayan görev açıklamalarında, kaynakları yaratabileceğiniz, yeni düzeye geçirebileceğiniz ve yok edebileceğiniz çalışan bir MSCS kümeniz olduğu varsayılmıştır. Böyle bir kümeniz olduğundan emin olmak istiyorsanız:

1. MSCS Cluster Administrator olanağını kullanarak bir grup oluşturun.
2. Bu grup içinde, sistem saatini (yol adı C:\winnt\system32\clock.exe ve C:\çalışma dizini) belirterek soysal bir uygulama kaynağı eşgörünümü yaratın.
3. Kaynağı çevrimiçi duruma getirebildiğinizden, kaynağı içeren grubu diğer düğüme taşıyabildiğinizden ve kaynağı çevrimdışı yapabildiğinizden emin olun.

### **Windows** *El ile başlatma ve MSCS*

MSCS tarafından yönetilen bir kuyruk yöneticisi için, başlatma özniteliğini el ile olarak ayarlamamız gerekir. Bu, IBM MQ MSCS desteğinin, kuyruk yöneticisini hemen başlatmadan MQSeries hizmetini yeniden başlatmasını sağlar.

IBM MQ MSCS desteğinin, izleme ve denetim gerçekleştirebilmesi için hizmeti yeniden başlatabilmesi gerekir; ancak, hangi kuyruk yöneticilerinin ve hangi makinelerde çalıştıklarının denetiminde kalması gerekir. Ek bilgi için bkz. [“Kuyruk yöneticisinin MSCS depolama alanına taşınması” sayfa 482](#).

### **Windows** *MSCS ve kuyruk yöneticileri*

MSCS kullanılırken kuyruk yöneticileriyle ilgili dikkat edilecek noktalar.

## **Diğer düğüme eşleşen bir kuyruk yöneticisi yaratılması**

Kümelemenin IBM MQ ile çalışması için düğüm B ' de A düğümündeki her biri için aynı kuyruk yöneticisine gereksinim duyarsınız. Ancak, ikincisini belirttik olarak yaratmanıza gerek yoktur. Bir düğüme kuyruk yöneticisi yaratabilir ya da hazırlayabilir, [“Kuyruk yöneticisinin MSCS depolama alanına taşınması” sayfa 482](#) konusunda açıklandığı gibi diğer düğüme taşıyabilir ve o düğüme tam olarak çoğaltılabilir.

## **Varsayılan kuyruk yöneticileri**

MSCS denetimi altında varsayılan bir kuyruk yöneticisi kullanmayın. Bir kuyruk yöneticisinin varsayılan değeri yapan bir özelliği yoktur; IBM MQ kendi kaydını tutar. Hata durumunda yedek sisteme geçiş sırasında diğer bilgisayara varsayılan olarak ayarlanmış bir kuyruk yöneticisini taşırsanız, bu, orada varsayılan değer olmaz. Tüm uygulamalarınızın belirli kuyruk yöneticilerine ad temelinde gönderme yapmasını sağlayın.



## Kuyruk yöneticisinin silinmesi

Bir kuyruk yöneticisi düğümü taşıdıktan sonra, ayrıntıları her iki bilgisayardaki kayıta da bulunur. Silmek istediğinizde, bunu bir bilgisayarda olağan biçimde yapın ve diğer bilgisayardaki kaydı temizlemek için [“MSCS yardımcı programları için destek” sayfa 494](#) içinde açıklanan yardımcı programı çalıştırın.

## Var olan kuyruk yöneticileri için destek

Kuyruk yöneticisi günlük dosyalarını ve kuyruk dosyalarını, iki makine arasında paylaşılan SCSI veriyolunda bulunan bir diske yerleştirebilmeniz koşuluyla, var olan bir kuyruk yöneticisini MSCS denetimi altına alabilirsiniz (bkz. [Şekil 71 sayfa 479](#)). MCS kaynağı yaratılırken kuyruk yöneticisini kısa bir süre çevrimdışı kılmanız gerekir.

Yeni bir kuyruk yöneticisi yaratmak istiyorsanız, MSCS ' den bağımsız olarak yaratın, sınavın ve MSCS denetiminin altına koyun. Bkz:

- [“MSCS ile kullanılmak üzere kuyruk yöneticisi yaratılması” sayfa 481](#)
- [“Kuyruk yöneticisinin MSCS depolama alanına taşınması” sayfa 482](#)
- [“Kuyruk yöneticisinin MSCS denetimi altına konması” sayfa 483](#)

## MSCS ' ye hangi kuyruk yöneticilerinin yönetileceğini söyleme

MSCS Küme Denetimcisi 'ni kullanarak, MSCS denetimi altına hangi kuyruk yöneticilerinin yerleştirileceğini seçerek, bu tür her kuyruk yöneticisi için bir kaynak eşgörünümü yaratabilirsiniz. Bu işlem, yönetim ortamının yönetmesini istediğiniz kuyruk yöneticisini seçebileceğiniz kaynakların listesini sunar.

## Kuyruk yöneticisi günlük dosyaları

Bir kuyruk yöneticisini MSCS depolama alanına taşıdığınızda, günlük ve veri dosyalarını paylaşılan bir diske taşıyabilirsiniz (örneğin, [“Kuyruk yöneticisinin MSCS depolama alanına taşınması” sayfa 482](#) ).

Taşımadan önce, kuyruk yöneticisini temiz bir şekilde kapatmanız ve veri dosyalarının ve günlük dosyalarının tam yedeğini almanız önerilir.

## Birden çok kuyruk yöneticisi

IBM MQ MSCS desteği, her makinede birden çok kuyruk yöneticisi çalıştırmanızı ve tek tek kuyruk yöneticilerini MSCS denetimi altına almanızı sağlar.

**Windows** *Kümeleri yönetmek için her zaman MSCS kullan*

Denetim komutlarını ya da IBM MQ Explorerolanağını kullanarak, MSCS denetimi altındaki herhangi bir kuyruk yöneticisinde doğrudan başlatma ve durdurma işlemlerini gerçekleştirmeye çalışmayın. Bunun yerine, kuyruk yöneticisini çevrimiçi duruma getirmek ya da çevrimdışı duruma getirmek için MSCS Cluster Administrator olanağını kullanın.

MSCS Cluster Administrator olanağının kullanılması, MSCS raporlamasının kuyruk yöneticisinin çevrimdışı olduğunu bildirmesinin neden olabileceği karışıklığı kısmen önler; ancak, MSCS denetiminin dışında başlatmış olmanız gerekir. Daha da önemlisi, MSCS kullanılmadan bir kuyruk yöneticisinin durdurulması, MSCS tarafından hata olarak algılanır ve diğer düğüme geçiş başlatılır.

**Windows** *MSCS ' de Etkin/Etkin kipte çalışma*

MSCS kümesindeki her iki bilgisayar da kuyruk yöneticilerini Etkin/Etkin kipinde çalıştırabilir. Tamamen boşta duran bir makinenin bekleme kipine girmesine gerek yoktur (ancak isterseniz Etkin/Edilgen Kipinde çalışabilirsiniz).

İş yükünü çalıştırmak için her iki makineyi de kullanmayı planlıyorsanız, küme iş yükünün tamamını tatmin edici bir performans düzeyinde çalıştırmak için her birine yeterli kapasite (işlemci, bellek, ikincil depolama) sağlayın.



**Not:** Microsoft Transaction Server (COM +) ile birlikte MSCS kullanıyorsanız, Etkin/Etkin kipi **kullanamazsınız** . Bunun nedeni, MSCS ve COM + ile IBM MQ kullanmak için:

- IBM MQ COM + desteğini kullanan uygulama bileşenleri, COM + ' un bir parçası olan Dağıtılmış Hareket Koordinatörü (DTC) ile aynı bilgisayarda çalışmalıdır.
- Kuyruk yöneticisi aynı bilgisayarda da çalışmalıdır.
- DTC bir MSCS kaynağı olarak yapılandırılmalıdır ve bu nedenle herhangi bir zamanda kümedeki bilgisayarlardan yalnızca birinde çalışabilir.

#### **Windows** *PostOnlineKomutu ve MSCS ' de PreOfflineKomutu*

IBM MQ MSCS desteğini diğer sistemlerle bütünleştirmek için bu komutları kullanın. Bunları, IBM MQ komutlarını yayınlamak ve bazı kısıtlamaları kullanmak için kullanabilirsiniz.

Değiştirgelelerde IBM MQ MSCStipindeki bir kaynağa ilişkin bu komutları belirtin. Bunları, IBM MQ MSCS desteğini diğer sistemlerle ya da yordamlarla bütünleştirmek için kullanabilirsiniz. Örneğin, posta iletisi gönderen, çağrı aygıtını etkinleştiren ya da başka bir izleme sistemi tarafından yakalanacak başka bir uyarı biçimi oluşturan bir programın adını belirleyebilirsiniz.

PostOnlineKomut, kaynak çevrimdışıdan çevrimiçi duruma geçtiğinde çağrılır; PreOfflineKomutu, çevrimiçinden çevrimdışına değişiklik için çağrılır. Bu komutlar çağrıldığında, varsayılan olarak Windows sistem dizininden çalıştırılır. IBM MQ , 32 bit kaynak izleme işlemi kullandığından, Windows 64 bit sistemlerinde bu, \Windows\system32 dizini yerine \Windows\SysWOW64 dizinidir. Daha fazla bilgi için, Windows x64 ortamındaki dosya yeniden yönlendirmesiyle ilgili Microsoft belgelerine bakın. Her iki komut da MSCS Küme Hizmetini çalıştırmak için kullanılan kullanıcı hesabı altında çalışır ve zamanuyumsuz olarak çağrılır; IBM MQ MSCS desteği devam etmeden önce bunların tamamlanmasını beklemez. Bu, diğer küme işlemlerini engelleme ya da geciktirme riskini ortadan kaldırır.

IBM MQ komutlarını yayınlamak (örneğin, İstekte Bulunan kanallarını yeniden başlatmak) için de bu komutları kullanabilirsiniz. Ancak, komutlar kuyruk yöneticisinin durumu değiştiği anda çalıştırılır, bu nedenle komutlar uzun süreli işlevler gerçekleştirmeyi amaçlamaz ve kuyruk yöneticisinin yürürlükteki durumu hakkında varsayımda bulunmamalıdır; kuyruk yöneticisi çevrimiçi duruma getirildikten hemen sonra, bir yönetici çevrimdışı bir komut verdi.

Kuyruk yöneticisinin durumuna bağlı olan programları çalıştırmak istiyorsanız, MSCS *Generic Application* kaynak tipinin eşgörünümlerini yaratmayı, bunları kuyruk yöneticisi kaynağıyla aynı MSCS grubuna yerleştirmeyi ve kuyruk yöneticisi kaynağına bağımlı kılmayı düşünebilirsiniz.

#### **Windows** *MSCS ' de tercih edilen düğümlerin kullanılması*

Her kuyruk yöneticisi için *tercih edilen düğüm* yapılandırmak üzere MSCS ' de Etkin/Etkin kip kullanılırken yararlı olabilir. Bununla birlikte, genel olarak tercih edilen bir düğümü ayarlamak değil, el ile onarım sonrası geri dönüş işlemine güvenmek daha iyidir.

Diğer bazı durumsuz kaynakların aksine, bir kuyruk yöneticisinin bir düğümden diğerine geçiş (ya da geri dönüş) işlemi biraz zaman alabilir. Gereksiz kesintileri önlemek için, bir kuyruk yöneticisini geri döndürmeden önce kurtarılan düğümü sınavın. Bu, *immediate* onarım sonrası geri dönüş ayarının kullanılmasını önler. Onarım sonrası geri dönüşü günün belirli saatleri arasında gerçekleşecek şekilde yapılandırabilirsiniz.

Büyük olasılıkla en güvenli yol, düğümün tam olarak kurtarıldığından emin olduğunuzda kuyruk yöneticisini gereken düğüme el ile geri taşımaktır. Bu, *preferred* node seçeneğinin kullanılmasını önler.

#### **Windows** *MSCS üzerine kuruluş yaptığınızda COM + hataları*

IBM MQ yeni kurulan bir MSCS kümesine kurulduğunda, Uygulama Olay günlüğünde Kaynak COM + ve Olay Tanıtıcısı 4691 ile ilgili bir hata bulabilirsiniz.

Bu, Microsoft Distributed Transaction eşgüdümçüsü (MSDTC) böyle bir ortamda çalışacak şekilde yapılandırılmadığında, IBM MQ ürününü bir Microsoft Cluster Server (MSCS) ortamında çalıştırmaya çalıştığınız anlamına gelir. Kümelenmiş bir ortamda MSDTC ' un yapılandırılmasına ilişkin bilgi için Microsoft belgelerine bakın.

## Windows MSCS yardımcı programları için destek

Bir komut isteminde çalıştırabileceğiniz, IBM MQ support for Microsoft Cluster Service (MSCS) yardımcı programlarının listesi.

**Not:** Windows Server 2016' den Microsoft Cluster Service (MSCS) için yeni ad Windows Server Failover Clustering (WSFC) olur.

MSCS için IBM MQ desteği aşağıdaki yardımcı programları içerir:

### Kaynak tipini kaydet/kaydını kaldır

haregtyp.exe

IBM MQ MSCS kaynak tipinin kaydını *kaldırdıktan* sonra artık bu tipte kaynak yaratamazsınız. Küme içinde bu tipte eşgörünümlerin varsa, MSCS, bir kaynak tipinin kaydını kaldırmanızı izin vermez:

1. MSCS Cluster Administrator olanağını kullanarak, MSCS denetimi altında çalışan kuyruk yöneticilerini [“MSCS ' den bir kuyruk yöneticisini çevrimdışı duruma getirme” sayfa 490](#) başlıklı konuda açıklandığı gibi çevrimdışı duruma getirin.
2. MSCS Cluster Administrator olanağını kullanarak kaynak eşgörünümlerini silin.
3. Bir komut isteminde, aşağıdaki komutu girerek kaynak tipinin kaydını geri alın:

```
haregtyp /u
```

Tipi *kaydetmek* istiyorsanız (ya da daha sonra yeniden kaydettirmek), bir komut isteminde aşağıdaki komutu girin:

```
haregtyp /r
```

MSCS kitaplıklarını başarıyla kaydettikten sonra, IBM MQ kuruluşundan bu yana sistemi yeniden başlatmanız gerekir.

### Kuyruk yöneticisini MSCS depolama alanına taşıma

hamvmqm.exe

Bkz. [“Kuyruk yöneticisinin MSCS depolama alanına taşınması” sayfa 482.](#)

### Düğümde bir kuyruk yöneticisinin silinmesi

hadl1mqm.exe

Kümenizde bir kuyruk yöneticisi olduğu, bir düğümden başka bir düğüme taşındığı ve şimdi yok etmek istediğiniz durumu göz önünde bulundurun. Şu anda bulunduğu düğümden silmek için IBM MQ Gezgini'ni kullanın. Kayıt girişleri diğer bilgisayarda var olmaya devam ediyor. Bunları silmek için, o bilgisayardaki bir komut istemine aşağıdaki komutu girin:

```
hadl1mqm /m qmname
```

Burada qmname , kaldırılacak kuyruk yöneticisinin adıdır.

### Kurulum ayrıntılarını denetle ve kaydet

amqmsysn.exe

Bu yardımcı program, IBM MQ MSCS Desteği kurulumunuzun tüm ayrıntılarını gösteren bir iletişim kutusu sunar; örneğin, IBM desteğini aramanız durumunda istenebilir. Ayrıntıları bir dosyaya kaydetme seçeneği vardır.

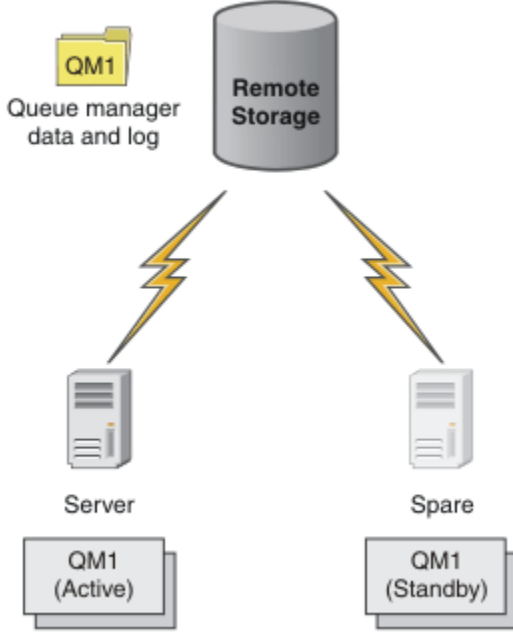
## Multi

## Çok eşgörünümlü kuyruk yöneticileri

Çok eşgörünümlü kuyruk yöneticileri, farklı sunucularda yapılandırılan aynı kuyruk yöneticisinin eşgörünümleridir. Kuyruk yöneticisinin bir yönetim ortamı etkin yönetim ortamı olarak tanımlandı ve başka bir yönetim ortamı yedek yönetim ortamı olarak tanımlandı. Etkin yönetim ortamı başarısız olursa, çok eşgörünümlü kuyruk yöneticisi yedek sunucuda otomatik olarak yeniden başlatılır.

## Örnek çok eşgörümlü kuyruk yöneticisi yapılandırması

Şekil 72 sayfa 495 içinde, QM1kuyruk yöneticisine ilişkin çok eşgörümlü bir yapılandırma örneği gösterilmektedir. IBM MQ , biri yedek olan iki sunucuya kurulur. Bir kuyruk yöneticisi ( QM1) yaratıldı. Bir QM1 eşgörümlü etkin ve tek bir sunucuda çalışıyor. QM1 'in diğer örneği diğer sunucuda yedekte çalışıyor, etkin işlem yapmıyor, ancak etkin eşgörümlü başarısız olursa, QM1' in etkin eşgörümlü devralmaya hazır. (Çok eşgörümlü bir yapılanışta kuyruk yöneticisinin tek bir etkin yönetim ortamı ve tek bir yedek yönetim ortamı olabilir.)



Şekil 72. Çok eşgörümlü kuyruk yöneticisi

Bir kuyruk yöneticisini çok eşgörümlü kuyruk yöneticisi olarak kullanmak istediğinizde, **crtmqm** komutunu kullanarak sunuculardan birinde tek bir kuyruk yöneticisi yaratın, kuyruk yöneticisi verilerini ve günlükleri paylaşılan ağ depolamasına yerleştirin. Diğer sunucuda, kuyruk yöneticisini yeniden yaratmak yerine, **addmqinf** komutunu kullanarak, kuyruk yöneticisi verilerine ve ağ depolamasındaki günlüklere ilişkin bir başvuru yaratın.

Artık kuyruk yöneticisini sunuculardan herhangi birinden çalıştırabilirsiniz. Sunucuların her biri aynı kuyruk yöneticisi verilerine ve günlüklere başvurur; yalnızca bir kuyruk yöneticisi vardır ve aynı anda yalnızca bir sunucuda etkindir.

Kuyruk yöneticisi tek bir eşgörümlü kuyruk yöneticisi olarak ya da çok eşgörümlü bir kuyruk yöneticisi olarak çalışabilir. Her iki durumda da kuyruk yöneticisinin yalnızca bir eşgörümlü çalışıyor, istekler işleniyor. Fark, çok eşgörümlü bir kuyruk yöneticisi olarak çalışırken, kuyruk yöneticisinin etkin yönetim ortamını çalıştırmayan sunucu yedek yönetim ortamı olarak çalışır; etkin sunucu başarısız olursa, etkin yönetim ortamından otomatik olarak devralmaya hazırdır.

Hangi eşgörümlü ilk olarak etkin olacağı üzerinde sahip olduğunuz tek denetim, iki sunucuda kuyruk yöneticisini başlatma sırasındadır. Kuyruk yöneticisi verilerine okuma/yazma kilitleri alan ilk yönetim ortamı etkin yönetim ortamı olur.

Etkin yönetim ortamını, başladıktan sonra, denetimi yedek veritabanına aktarmak için yedek sisteme geçiş seçeneğini kullanarak etkin yönetim ortamını durdurarak, etkin yönetim ortamını diğer sunucuyla değiştirebilirsiniz.

QM1 etkin örneği, çalışırken paylaşılan kuyruk yöneticisi verilerine ve günlük klasörlerine dışlayıcı erişime sahiptir. QM1 yedek yönetim ortamı etkin yönetim ortamı başarısız olduğunda saptar ve etkin yönetim ortamı olur. QM1 verilerini ve etkin eşgörümlü tarafından bırakıldıkları durumdaki günlükleri devralır ve istemcilerden ve kanallardan gelen yeniden bağlantıları kabul eder.

Etkin yönetim ortamı, yedek veritabanının devralınmasıyla sonuçlanan çeşitli nedenlerden ötürü başarısız olabilir:

- Etkin kuyruk yöneticisi eşgörünümünü barındıran sunucu başarısız oldu.
- Etkin kuyruk yöneticisi eşgörünümünü barındıran sunucu ile dosya sistemi arasında bağlantı kurulamaması.
- IBM MQ tarafından saptanan kuyruk yöneticisi işlemlerinin yanıt verilebilirliği. Bu işlem kuyruk yöneticisini kapatır.

Kuyruk yöneticisi yapılanış bilgilerini birden çok sunucuya ekleyebilir ve etkin/yedek sunucu çifti olarak çalışacak iki sunucu seçebilirsiniz. Toplam iki eşgörünüm sınırı vardır. İki yedek yönetim ortamınız ve bir etkin yönetim ortamınız olamaz.

## **Yüksek düzeyde kullanılabilirlik çözümü oluşturmak için gereken ek bileşenler**

Çok eşgörünümlü kuyruk yöneticisi, yüksek kullanılabilirlikli bir çözümün bir parçasıdır. Yararlı bir yüksek düzeyde kullanılabilirlik çözümü oluşturmak için bazı ek bileşenlere gereksinim duyarsınız.

- Etkin kuyruk yöneticisi eşgörünümünü çalıştırmayı devralan bilgisayara IBM MQ bağlantıları aktarmak için istemci ve kanal yeniden bağlantısı.
- Kilitleri doğru şekilde yöneten ve ortam ve dosya sunucusu hatasına karşı koruma sağlayan yüksek performanslı bir paylaşılan ağ dosya sistemi (NFS).

**Önemli:** NFS sürücüsünde bakım gerçekleştirilmeden önce, ortamınızda çalışan tüm çok eşgörünümlü kuyruk yöneticisi yönetim ortamlarını durdurmanız gerekir. Bir NFS arızası durumunda, kurtarılacak kuyruk yöneticisi yapılandırma yedeklerinizin olduğundan emin olun.

- Temel altyapıdaki tek hata noktalarını ortadan kaldırmak için esnek ağlar ve güç kaynakları.
- Hata durumunda yedek sisteme geçişi tolere eden uygulamalar. Özellikle, hareket işleme uygulamalarının davranışına ve IBM MQ kuyruklarına göz atayan uygulamalara dikkat etmeniz gerekir.
- Etkin ve yedek yönetim ortamlarının çalıştığından emin olmak ve başarısız olan etkin yönetim ortamlarını yeniden başlatmak için etkin ve yedek yönetim ortamlarının izlenmesi ve yönetilmesi. Çok eşgörünümlü kuyruk yöneticileri otomatik olarak yeniden başlatılsa da, yedek yönetim ortamlarınızın çalıştığından, devralmaya hazır olduğundan ve başarısız olan yönetim ortamlarının yeni yedek yönetim ortamları olarak yeniden çevrimiçi duruma getirildiğinden emin olun.

IBM MQ MQI clients ve kanallar etkin duruma geldiğinde otomatik olarak yedek kuyruk yöneticisine yeniden bağlanır. Yeniden bağlantıyla ilgili daha fazla bilgi ve yüksek kullanılabilirlikli bir çözümdeki diğer bileşenler ilgili konularda bulunabilir. Otomatik istemci yeniden bağlantısı IBM MQ classes for Javataarafından desteklenmez.

## **Desteklenen platformlar**

z/OS dışı herhangi bir altyapıda çok eşgörünümlü bir kuyruk yöneticisi yaratabilirsiniz.

MQI istemcileri için otomatik istemci yeniden bağlantısı desteklenir.

### **Çok eşgörünümlü kuyruk yöneticisi yarat**

Çok eşgörünümlü bir kuyruk yöneticisi yaratın, kuyruk yöneticisini bir sunucuda yaratın ve IBM MQ ' i başka bir sunucuda yapılandırın. Çok eşgörünümlü kuyruk yöneticileri kuyruk yöneticisi verilerini ve günlüklerini paylaşır.

Çok eşgörünümlü bir kuyruk yöneticisi yaratmaya ilişkin çabaların çoğu, paylaşılan kuyruk yöneticisi verilerini ve günlük dosyalarını ayarlama görevidir. Ağ depolamasındaki paylaşılan dizinleri oluşturmalı ve dizinleri ağ paylaşımlarını kullanarak diğer sunucuların kullanımına açmalısınız. Bu görevlerin, AIX and Linux sistemlerinde *root* (kök) gibi bir denetim yetkisi bulunan biri tarafından gerçekleştirilmesi gerekir. Adımlar aşağıdaki gibidir:

1. Veri ve günlük dosyaları için paylaşımlar oluşturun.
2. Kuyruk yöneticisini tek bir sunucuda yaratın.

3. Kuyruk yöneticisi yapılanış verilerini toplamak ve panoya kopyalamak için ilk sunucuda **dspmqlnf** komutunu çalıştırın.
4. İkinci sunucuda kuyruk yöneticisi yapılanışını yaratmak için, kopyalanan verilerle **addmqinf** komutunu çalıştırın.

Kuyruk yöneticisini ikinci sunucuda yeniden yaratmak için **crtmqm** komutunu çalıştırmayın.

## Dosya erişim denetimi

Diğer tüm sunuculardaki mqm kullanıcı ve grubunun paylaşımlara erişme iznine sahip olduğunu göz kulak olmanız gerekir.

AIX and Linux işletim sisteminde, mqm uid ve gid öğelerini tüm sistemlerde aynı yapmanız gerekir. mqm için ortak bir uid ve gid ayarlamak üzere her bir sistemde /etc/passwd dosyasını düzenlemeniz ve daha sonra, sisteminizi yeniden başlatmanız gerekebilir.

Microsoft Windows üzerinde, kuyruk yöneticisi işlemlerini çalıştıran kullanıcı kimliğinin, kuyruk yöneticisi verilerini ve günlük dosyalarını içeren dizinler için tam denetim iznine sahip olması gerekir. İzni iki şekilde yapılandırabilirsiniz:

1. Diğer güvenlik birincil kullanıcısı olarak genel grubu olan bir kuyruk yöneticisi yaratın. Genel gruba, kuyruk yöneticisi verilerini ve günlük dosyalarını içeren dizinlere tam denetim erişimi yetkisi verin; bkz. [“Windows üzerinde paylaşılan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğinin sağlanması” sayfa 525](#). Kuyruk yöneticisini çalıştıran kullanıcı kimliğini genel grubun üyesi yapın. Yerel bir kullanıcıyı genel bir grubun üyesi yapamazsınız, bu nedenle kuyruk yöneticisi işlemleri bir etki alanı kullanıcı kimliği altında çalıştırılmalıdır. Etki alanı kullanıcı kimliği, mqm yerel grubunun bir üyesi olmalıdır. [“Windows üzerinde etki alanı iş istasyonlarında ya da sunucularda çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 499](#) görevi, bu şekilde güvenli kılınan dosyaları kullanarak çok eşgörünümlü bir kuyruk yöneticisinin nasıl ayarlanacağını gösterir.
2. Yerel mqm grubunun "domain local" etki alanı kapsamına sahip olması için etki alanı denetleyicisinde bir kuyruk yöneticisi oluşturun. Dosya paylaşımını yerel mqm etki alanıyla güvenli hale getirin ve aynı etki alanı yerel mqm grubu altında bir kuyruk yöneticisinin tüm yönetim ortamlarında kuyruk yöneticisi işlemlerini çalıştırın. [“Windows etki alanı denetleyicilerinde çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 515](#) görevi, bu şekilde güvenli kılınan dosyaları kullanarak çok eşgörünümlü bir kuyruk yöneticisinin nasıl ayarlanacağını gösterir.

## Yapılandırma bilgileri

Her bir sunucuyla ilgili IBM MQ kuyruk yöneticisi yapılanış bilgilerini değiştirerek, gereken sayıda kuyruk yöneticisi yönetim ortamı yapılandırın. Her sunucuda uyumlu bir düzeltme düzeyinde aynı IBM MQ sürümü kurulu olmalıdır. **dspmqlnf** ve **addmqinf** komutları, ek kuyruk yöneticisi yönetim ortamlarını yapılandırmanıza yardımcı olur. Alternatif olarak, mqs.ini ve qm.ini dosyalarını doğrudan düzenleyebilirsiniz. [“Linux üzerinde çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 537](#), [“Windows üzerinde etki alanı iş istasyonlarında ya da sunucularda çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 499](#) ve [“Windows etki alanı denetleyicilerinde çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 515](#) konuları, çok eşgörünümlü bir kuyruk yöneticisinin nasıl yapılandırılacağını gösteren örneklerdir.

AIX, Linux, and Windows sistemlerinde, tek bir mqs.ini dosyasını ağ paylaşımına yerleştirerek ve **AMQ\_MQS\_INI\_LOCATION** ortam değişkenini onu gösterecek şekilde ayarlayarak paylaşabilirsiniz.

## Sınırlamalar

1. Yalnızca aynı işletim sistemine, mimariye ve endianness sahip sunucularda aynı kuyruk yöneticisinin birden çok örneğini yapılandırın. Örneğin, her iki makine de 32 bit ya da 64 bit olmalıdır.
2. Tüm IBM MQ kuruluşları 7.0.1 yayın düzeyinde ya da daha yüksek bir yayın düzeyinde olmalıdır.
3. Genellikle, etkin ve beklemedeki kuruluşlar aynı bakım düzeyinde tutulur. Tüm kuruluşları birlikte büyütmeniz gerekip gerekmediğini denetlemek için her bir büyütme için bakım yönergelerine bakın.

Etkin ve pasif kuyruk yöneticilerine ilişkin bakım düzeylerinin aynı olması gerektiğini unutmayın.

4. Kuyruk yöneticisi verilerini ve günlüklerini yalnızca aynı IBM MQ kullanıcı, grup ve erişim denetimi mekanizmasıyla yapılandırılmış kuyruk yöneticileri arasında paylaşın. **IBM i** Örneğin, bir Linux sunucusunda ayarlanan ağ paylaşımı, AIX and Linux kuyruk yöneticileri için ayrı kuyruk yöneticisi verileri ve günlükleri içerebilir, ancak IBM tarafından kullanılan kuyruk yöneticisi verilerini içeremez.

**IBM i** Paylaşımlar farklı olduğu sürece IBM i ve AIX and Linux sistemleri için aynı ağ depolaması üzerinde birden çok paylaşım oluşturabilirsiniz. Farklı paylaşımlara farklı sahipler verebilirsiniz. Kısıtlama, AIX and Linux ve IBM arasındaki IBM MQ kullanıcıları ve grupları için kullanılan farklı adların bir sonucudur. Kullanıcının ve grubun aynı uid ve gid grubuna sahip olması kısıtlamayı gevşetmez.

5. AIX and Linux sistemlerinde, paylaşılan dosya sistemini ağ depolamasındaki donanımla, kesilebilir, yumuşak bir bağlamayla yapılandırın. Sabit kesilebilir bir bağlama, kuyruk yöneticisini bir sistem çağrısı tarafından kesilinceye kadar askıda kalmaya zorlar. Yazılım bağlamaları, sunucu hatasından sonra veri tutarlılığını garanti etmez.

6. Paylaşılan günlük ve veri dizinleri bir FAT ya da NFSv3 dosya sisteminde saklanamaz. Windows üzerindeki çok eşgörümlü kuyruk yöneticileri için, ağa bağlı depolamaya Windows ağları tarafından kullanılan Common Internet File System (CIFS) iletişim kuralı tarafından erişilmelidir.

7. **z/OS** z/OS , çok eşgörümlü kuyruk yöneticilerini desteklemez. Kuyruk paylaşım gruplarını kullanın.

Yeniden bağlanabilir istemciler z/OS kuyruk yöneticileriyle çalışır.

**Windows** *Windows etki alanları ve çok eşgörümlü kuyruk yöneticileri*

Windows üzerinde çok eşgörümlü bir kuyruk yöneticisi, verilerinin ve günlüklerinin paylaşılmasını gerektirir. Paylaşım, farklı sunucularda ya da iş istasyonlarında çalışan kuyruk yöneticisinin tüm eşgörümleri için erişilebilir olmalıdır. Kuyruk yöneticilerini yapılandırın ve bir Windows etki alanının parçası olarak paylaşın. Kuyruk yöneticisi bir etki alanı iş istasyonunda ya da sunucusunda ya da etki alanı denetleyicisinde çalışabilir.

**Önemli:** Varsayılan olarak, Windows 10 sürüm 1607 ve Windows Server 2016 ile başlayan bilgisayarlar, Windows ürününün önceki sürümlerinden daha kısıtlayıcıdır.

Bu değişiklik, Güvenlik Hesapları Yöneticisi 'ne (SAM) uzak çağrı yapmasına izin verilen istemcileri sınırlar ve kuyruk yöneticileri başlatılmadığı için IBM MQ ögesini etkileyebilir. IBM MQ bir etki alanı hesabı olarak yapılandırıldığında, SAM 'ye erişim IBM MQ ' in çalışması için kritik önem taşır.

Çok eşgörümlü bir kuyruk yöneticisini yapılandırmadan önce, kuyruk yöneticisi verilerine ve günlük dosyalarına erişimin nasıl denetleneceğini gözden geçirmek için [“Windows üzerinde paylaşılmayan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğini sağlayın” sayfa 528](#) ve [“Windows üzerinde paylaşılan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğini sağlanması” sayfa 525](#) başlıklı bilgileri okuyun. Konular eğitimdir; Windows etki alanında çok eşgörümlü bir kuyruk yöneticisi için doğrudan paylaşılan dizinler ayarlamaya gitmek istiyorsanız, bkz. [“Windows üzerinde etki alanı iş istasyonlarında ya da sunucularda çok eşgörümlü kuyruk yöneticisi yaratılması” sayfa 499](#).

## **Etki alanı iş istasyonlarında ya da sunucularda çok eşgörümlü kuyruk yöneticisini çalıştır**

IBM WebSphere MQ 7.1' den, çok eşgörümlü kuyruk yöneticileri bir etki alanının üyesi olan bir iş istasyonunda ya da sunucuda çalışır. Windows üzerinde çok eşgörümlü bir kuyruk yöneticisini çalıştırmak için bir etki alanı denetleyicisi, bir dosya sunucusu ve aynı etki alanına bağlı aynı kuyruk yöneticisini çalıştıran iki iş istasyonu ya da sunucu gerekir.

Bir etki alanındaki herhangi bir sunucuda ya da iş istasyonunda çok eşgörümlü bir kuyruk yöneticisinin çalıştırılmasını mümkün kılan değişiklik, artık ek bir güvenlik grubuyla bir kuyruk yöneticisi yaratabilmendir. Ek güvenlik grubu, -a deęiřtirgesinde **crtnqm** komutunda geçirilir. Kuyruk yöneticisi verilerini ve günlüklerini içeren dizinlerin güvenliğini grupla birlikte sağlarsınız. Kuyruk yöneticisi işlemlerini çalıştıran kullanıcı kimliği bu grubun bir üyesi olmalıdır. Kuyruk yöneticisi dizinlere eriştiğinde,

Windows kullanıcı kimliğinin dizinlere erişmesi için gereken izinleri denetler. Hem gruba hem de kullanıcı kimliği etki alanı kapsamına izin vererek, kuyruk yöneticisi işlemlerini çalıştıran kullanıcı kimliğinin genel gruptaki kimlik bilgileri var. Kuyruk yöneticisi farklı bir sunucuda çalışıyorsa, kuyruk yöneticisi işlemlerini çalıştıran kullanıcı kimliği aynı kimlik bilgilerine sahip olabilir. Kullanıcı kimliğinin aynı olması gerekmez. Yerel mqm grubunun bir üyesinin yanı sıra alternatif güvenlik grubunun da üyesi olması gerekir.

Çok eşgörünümlü bir kuyruk yöneticisi yaratılmasıyla ilgili ayrıntılar için bkz. [“Windows üzerinde etki alanı iş istasyonlarında ya da sunucularda çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 499](#) .

Etki alanını ve etki alanı sunucularını ve iş istasyonlarını yapılandırmak için birden çok adım gerekir. Windows ' in bir kuyruk yöneticisi tarafından veri ve günlük dizinlerine erişimi nasıl yetkilendirdiğini anlamamız gerekir. Kuyruk yöneticisi işlemlerinin günlük dosyalarına erişme yetkisinin nasıl olduğundan emin değilseniz ve veri dosyaları [“Windows üzerinde paylaşılmayan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğini sağlayın” sayfa 528](#) konusunu okur. Bu konuda, gerekli adımları anlamamıza yardımcı olacak iki görev bulunur. Görevler [“Yerel mqm grubu tarafından yetkilendirilen veri ve günlük dosyalarının okunması ve yazılması” sayfa 530](#) ve [“Alternatif bir yerel güvenlik grubu tarafından yetkilendirilen veri ve günlük dosyalarının okunması ve yazılması” sayfa 533](#) ' dir. [“Windows üzerinde paylaşılan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğinin sağlanması” sayfa 525](#) başlıklı başka bir konuda, diğer güvenlik grubuyla kuyruk yöneticisi verilerini ve günlük dosyalarını içeren paylaşılan dizinlerin nasıl güvenliğinin sağlanacağı açıklanmaktadır. Konu, Windows etki alanı oluşturmak, bir dosya paylaşımı yaratmak, IBM MQ for Windows kurmak ve bir kuyruk yöneticisini paylaşımı kullanacak şekilde yapılandırmak için dört görevi içerir. Görevler aşağıdaki gibidir:

1. [“Windows üzerinde Active Directory ve DNS etki alanı oluşturma” sayfa 503](#).
2. [“IBM MQ ' in Windows etki alanındaki bir sunucuya ya da iş istasyonuna kurulması” sayfa 506](#).
3. [“Windows üzerinde kuyruk yöneticisi verileri ve günlük dosyaları için paylaşılan dizin oluşturma” sayfa 509](#).
4. [“Alternatif bir genel güvenlik grubu tarafından yetkilendirilen paylaşılan verileri ve günlük dosyalarını okuma ve yazma” sayfa 512](#).

Daha sonra etki alanını kullanarak [“Windows üzerinde etki alanı iş istasyonlarında ya da sunucularda çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 499](#) görevini yapabilirsiniz. Bilginizi bir üretim etki alanına aktarmadan önce çok eşgörünümlü bir kuyruk yöneticisi ayarlamayı keşfetmek için bu görevleri gerçekleştirin.

## **Etki alanı denetleyicilerinde çok eşgörünümlü kuyruk yöneticisini çalıştır**

Kuyruk yöneticisi verilerinin güvenliği mqm etki alanı grubuyla sağlanabilir. [“Windows üzerinde paylaşılan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğinin sağlanması” sayfa 525](#) başlıklı konuda anlatıldığı gibi, iş istasyonlarında ya da sunucularda yerel mqm grubuyla güvenli dizinleri paylaşamazsınız. Ancak etki alanı denetleyicilerinde tüm grup ve birincil kullanıcılar etki alanı kapsamına sahip olur. IBM MQ for Windows ' i bir etki alanı denetleyicisine kurarsanız, kuyruk yöneticisi verileri ve günlük dosyaları, paylaşılabilir mqm etki alanı grubuyla güvenceye alınır. Etki alanı denetleyicilerinde çok eşgörünümlü bir kuyruk yöneticisi yapılandırmak için [“Windows etki alanı denetleyicilerinde çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 515](#) görevindeki adımları izleyin.

### **İlgili bilgiler**

[Yetkilendirme ve Erişim Denetiminin Yönetilmesi](#)

[Windows Server küme düğümlerini etki alanı denetleyicileri olarak kullanma](#)

**Windows** *Windows üzerinde etki alanı iş istasyonlarında ya da sunucularda çok eşgörünümlü kuyruk yöneticisi yaratılması*

Bir örnek, bir iş istasyonunda ya da Windows etki alanının bir parçası olan bir sunucuda Windows üzerinde çok eşgörünümlü bir kuyruk yöneticisinin nasıl ayarlanacağını gösterir. Sunucunun etki alanı denetleyicisi olması gerekmez. Kurulum, üretim ölçeği olmak yerine, ilgili kavramları gösterir. Örnek, Windows Server 2008 'e dayalıdır. Adımlar, Windows Server 'in diğer sürümlerinde farklı olabilir.



Üretim ölçeği yapılandırmasında, yapılandırmayı var olan bir etki alanına uyarlamamız gerekebilir. Örneğin, farklı paylaşımları yetkilendirmek ve kuyruk yöneticilerini çalıştıran kullanıcı kimliklerini gruplamak için farklı etki alanı grupları tanımlayabilirsiniz.

Örnek yapılandırma üç sunucudan oluşur:

#### ***sun***

Bir Windows Server 2008 etki alanı denetleyicisi. *Sun, marsve venus*çeren *wmq.example.com* etki alanının sahibi olur. Şekil amacıyla, dosya sunucusu olarak da kullanılır.

#### ***mars***

İlk IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. Çok eşgörünümlü kuyruk yöneticisinin *QMGR*adlı bir eşgörünümünü içerir.

#### ***venus***

İkinci IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. *QMGR*adlı çok eşgörünümlü kuyruk yöneticisinin ikinci eşgörünümünü içerir.

Örnekteki italik adları, seçtiğiniz adlarla değiştirin.

## **Başlamadan önce**

Windowsüzerinde, kuyruk yöneticisi verilerini ve günlük dosyalarını saklamayı planladığınız dosya sistemini doğrulamanız gerekmez. Paylaşılan dosya sistemi davranışının doğrulanmasıdenetim yordamı AIX and Linux için geçerlidir. Windowsüzerinde, denetimler her zaman başarılı olur.

Aşağıdaki görevlerdeki adımları gerçekleştirin. Görevler etki alanı denetleyicisini ve etki alanını oluşturur, IBM MQ for Windows ürününü tek bir sunucuya kurar ve veri ve günlük dosyaları için dosya paylaşımı yaratır. Var olan bir etki alanı denetleyicisini yapılandırıyorsanız, yeni bir Windows Server 2008 'deki adımları denemeniz yararlı olabilir. Adımları etki alanınıza uyarlayabilirsiniz.

1. “Windows üzerinde Active Directory ve DNS etki alanı oluşturma” sayfa 503.
2. “IBM MQ ' in Windows etki alanındaki bir sunucuya ya da iş istasyonuna kurulması” sayfa 506.
3. “Windows üzerinde kuyruk yöneticisi verileri ve günlük dosyaları için paylaşılan dizin oluşturma” sayfa 509.
4. “Alternatif bir genel güvenlik grubu tarafından yetkilendirilen paylaşılan verileri ve günlük dosyalarını okuma ve yazma” sayfa 512.

## **Bu görev hakkında**

Bu görev, bir etki alanı denetleyicisini ve etki alanındaki iki sunucuyu kuyruk yöneticisinin eşgörünümlerini çalıştıracak şekilde yapılandırmak için kullanılan görevlerden biridir. Bu görevde, *venus*adlı ikinci bir sunucuyu, *QMGR*kuyruk yöneticisinin başka bir eşgörünümünü çalıştıracak şekilde yapılandırır. Kuyruk yöneticisinin ikinci eşgörünümünü ( *QMGR*) yaratmak ve çalıştığını sınamak için bu görevdeki adımları izleyin.

Bu görev, önceki bölümdeki dört görevden ayrıdır. Tek bir eşgörünüm kuyruk yöneticisini çok eşgörünümlü bir kuyruk yöneticisine dönüştüren adımları içerir. Diğer tüm adımlar, tek ya da çok eşgörünümlü kuyruk yöneticileri için ortaktır.

## **Yordam**

1. IBM MQ for Windows' yi çalıştırmak için ikinci bir sunucu yapılandırın.
  - a) İkinci bir etki alanı sunucusu oluşturmak için “IBM MQ ' in Windows etki alanındaki bir sunucuya ya da iş istasyonuna kurulması” sayfa 506 görevindeki adımları gerçekleştirin. Bu görev sırasında ikinci sunucu *venus*olarak adlandırılır.

**İpucu:** İki sunucunun her birinde IBM MQ için aynı kuruluş varsayılanlarını kullanarak ikinci kuruluşu oluşturun. Varsayılanlar farklıysa, IBM MQ yapılandırma dosyasındaki *mq.s.in* i *QMGR QueueManager* kısmına ilişkin Prefix (Önek) ve *InstallationName* (Kuruluş Adı) değişkenlerini uyarlamamız gerekebilir. Değişkenler, her sunucudaki her kuruluş ve kuyruk yöneticisi için farklı



olabilen yollara başvurmaktadır. Yollar her sunucuda aynı kalıyorsa, çok eşgörunümlü bir kuyruk yöneticisi yapılandırmak daha kolaydır.

2. *venus* üzerinde ikinci bir *QMGR* örneği oluşturun.

- a) *mars* üzerinde *QMGR* yoksa, bunu oluşturmak için “[Alternatif bir genel güvenlik grubu tarafından yetkilendirilen paylaşılan verileri ve günlük dosyalarını okuma ve yazma](#)” sayfa 512 görevini gerçekleştirin
- b) Prefix (Önek) ve InstallationName deęiřtirgelerinin deęerlerinin *venus* için doęru olup olmadığını denetleyin.

*mars*' da **dspmqrinf** komutunu çalıştırın:

```
dspmqrinf QMGR
```

Sistemin yapması gereken:

```
QueueManager:  
Name=QMGR  
Directory=QMGR  
Prefix=C:\ProgramData\IBM\MQ  
DataPath=\\sun\wmq\data\QMGR  
InstallationName=Installation1
```

- c) **QueueManager** kütasının makine tarafından okunabilir biçimini panoya kopyalayın.

*mars* üzerinde, -o command deęiřtirgesiyle **dspmqrinf** komutunu yeniden çalıştırın.

```
dspmqrinf -o command QMGR
```

Sistemin yapması gereken:

```
addmqinf -s QueueManager -v Name=QMGR  
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\MQ"  
-v DataPath=\\sun\wmq\data\QMGR
```

- d) *venus* üzerinde, *venus* üzerinde kuyruk yöneticisinin bir eşgörunümünü yaratmak için **addmqinf** komutunu panodan çalıştırın.

Gerekliyse, Prefix (Önek) ya da InstallationName deęiřtirgelerindeki farklılıkları karşılamak için komutu ayarlayın.

```
addmqinf -s QueueManager -v Name=QMGR  
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\MQ"  
-v DataPath=\\sun\wmq\data\QMGR
```

IBM MQ configuration information added.

3. Yedek yönetim ortamlarına izin veren *QMGR* kuyruk yöneticisini *venus* üzerinde başlatın.

- a) *QMGR* on *mars* işaretinin durdurulduęunu denetleyin.

*mars*' da **dspmqr** komutunu çalıştırın:

```
dspmqr -m QMGR
```

Sistem yanıtı, kuyruk yöneticisinin nasıl durdurulduęuna baęlıdır; örneęin:

```
C:\Users\Administrator>dspmqr -m QMGR
QMNAME(QMGR) STATUS(Ended immediately)
```

b) *venus* işletim sistemlerinde, standları izin veren **stirmqm** komutunu *QMGR* başlatmak için çalıştırın:

```
stirmqm -x QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

## Sonuçlar

Çok eşgönlü kuyruk yöneticisi geçişleri sınamak için aşağıdaki adımları izleyin:

1. *mars* işletim sistemlerinde, yedek takıma izin veren *QMGR* ' i başlatmak için **stirmqm** komutunu çalıştırın:

```
stirmqm -x QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
A standby instance of queue manager 'QMGR' has been started.
The active instance is running elsewhere.
```

2. *venus* üzerinde **endmqm** komutunu çalıştırın:

```
endmqm -r -s -i QMGR
```

*venus* üzerindeki sistem yanıtı:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended, permitting switchover to
a standby instance.
```

Ve *mars* üzerinde:

```
dspmqr
QMNAME(QMGR) STATUS(Running as standby)
C:\Users\wmquser2>dspmqr
QMNAME(QMGR) STATUS(Running as standby)
```

```
C:\Users\wmquser2>dspmq  
QMNAME(QMGR) STATUS(Running)
```

## Sonraki adım

Çok eşgörünümlü bir kuyruk yöneticisini örnek programları kullanarak doğrulamak için bkz. [“Windows üzerinde çok eşgörünümlü kuyruk yöneticisinin doğrulanması” sayfa 522.](#)

### **Windows** *Windows üzerinde Active Directory ve DNS etki alanı oluşturma*

Bu görev, *sun* adlı bir Windows 2008 etki alanı denetleyicisinde *wmq.example.com* etki alanını yaratır. Etki alanındaki Domain *mqm* genel grubunu doğru haklarla ve tek bir kullanıcıyla yapılandırır.

Üretim ölçeği yapılandırmasında, yapılandırmayı var olan bir etki alanına uyarlamamız gerekebilir. Örneğin, farklı paylaşımları yetkilendirmek ve kuyruk yöneticilerini çalıştıran kullanıcı kimliklerini gruplamak için farklı etki alanı grupları tanımlayabilirsiniz.

Örnek yapılandırma üç sunucudan oluşur:

#### ***sun***

Bir Windows Server 2008 etki alanı denetleyicisi. *Sun, marsve venus* içeren *wmq.example.com* etki alanının sahibi olur. Şekil amacıyla, dosya sunucusu olarak da kullanılır.

#### ***mars***

İlk IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. Çok eşgörünümlü kuyruk yöneticisinin *QMGR* adlı bir eşgörünümünü içerir.

#### ***venus***

İkinci IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. *QMGR* adlı çok eşgörünümlü kuyruk yöneticisinin ikinci eşgörünümünü içerir.

Örnekteki italik adları, seçtiğiniz adlarla değiştirin.

## Başlamadan önce

1. Görev adımları, kurulu ancak herhangi bir rolle yapılandırılmamış bir Windows Server 2008 ile tutarlıdır. Var olan bir etki alanı denetleyicisini yapılandırıyorsanız, yeni bir Windows Server 2008 'deki adımları denemeniz yararlı olabilir. Adımları etki alanınıza uyarlayabilirsiniz.

## Bu görev hakkında

Bu görevde, yeni etki alanı denetleyicisinde bir Active Directory (Etkin Dizin) ve DNS etki alanı yaratırsınız. Daha sonra, IBM MQ ürününü etki alanına katılan diğer sunuculara ve iş istasyonlarına kurmaya hazır olarak yapılandırın. Bir Windows etki alanı oluşturmak için Active Directory 'yi kurma ve yapılandırma konusunda bilginiz yoksa görevi izleyin. Çok eşgörünümlü kuyruk yöneticisi yapılanışı yaratmak için bir Windows etki alanı yaratmalısınız. Bu görevin, bir Windows etki alanını yapılandırmanız için size en iyi şekilde yol göstermesi amaçlanmamıştır. Çok eşgörünümlü kuyruk yöneticilerini bir üretim ortamında konuşlandırmak için Windows belgelerine bakın.

Görev sırasında aşağıdaki adımları gerçekleştirmeniz gerekir:

1. Active Directory'yi kurun.
2. Bir etki alanı ekleyin.
3. Etki alanını DNS 'ye ekleyin.
4. Domain *mqm* adlı genel grubu yaratın ve doğru hakları verin.
5. Bir kullanıcı ekleyin ve bunu Domain *mqm* genel grubunun bir üyesi yapın.

Bu görev, kuyruk yöneticisi verilerine ve günlük dosyalarına erişimi gösteren ilgili görevler kümesinden biridir. Görevler, seçtiğiniz bir dizinde saklanan verileri ve günlük dosyalarını okuma ve yazma yetkisi olan bir kuyruk yöneticisinin nasıl yaratılacağını gösterir. Görevle birlikte, [“Windows etki alanları ve çok eşgörünümlü kuyruk yöneticileri” sayfa 498.](#)

Görevin amaçları açısından etki alanı denetleyicisi anasistem adı *sun*, iki IBM MQ sunucusu *mars* ve *venus* olarak adlandırılır. Etki alanına *wmq.example.com* adı verilir. Görevdeki tüm italik adları kendi seçtiğiniz adlarla değiştirebilirsiniz.

## Yordam

1. *sun* etki alanı denetleyicisinde yerel ya da Workgroup yöneticisi olarak oturum açın.  
Sunucu önceden bir etki alanı denetleyicisi olarak yapılandırıldıysa, etki alanı yöneticisi olarak oturum açmanız gerekir.
2. Active Directory Etki Alanı Hizmetleri sihirbazını çalıştırın.
  - a) **Başlat > Çalıştır ...** düğmesini tıklatın. *dcpromo* yazın ve **Tamam'** i tıklatın.  
Active Directory ikili dosyaları önceden kurulu değilse, Windows dosyaları otomatik olarak kurar.
3. Sihirbazın ilk penceresinde, **Use advanced mode installation** (Gelişmiş kip kuruluşunu kullan) onay kutusunu temizleyin. **Next > Next (İleri)** düğmesini tıklatın ve **Create a new domain in a new forest > Next** (Yeni ormanda yeni bir etki alanı oluştur) seçeneğini tıklatın.
4. **Orman kök etki alanının FQDN 'si** alanına *wmq.example.com* yazın. **İleri'**yi tıklatın.
5. Set Forest Functional Level (Orman İşlevsel Düzeyini Ayarla) penceresinde, **Forest işlev düzeyleri > Next** (İleri) listesinden **Windows Server 2003** ya da daha sonraki bir yayın düzeyini seçin.  
IBM MQ tarafından desteklenen en eski Windows Server düzeyi Windows Server 2003 'tür.
6. İsteğe bağlı: Etki Alanı İşlevsel Düzeyini Ayarla penceresinde, **Etki Alanı işlevsel düzeyleri > Sonraki** listesinden **Windows Server 2003** ya da daha üstünü seçin.  
Bu adım yalnızca Orman İşlevsel Düzeyini **Windows Server 2003** olarak ayarladıysanız gereklidir.
7. Ek seçenek olarak **DNS sunucusu** seçildiğinde Ek Etki Alanı Denetleyicisi Seçenekleri penceresi açılır. Uyarı penceresini temizlemek için **Next (İleri)** ve **Yes (Evet)** düğmesini tıklatın.  
**İpucu:** Bir DNS sunucusu önceden kuruluysa, bu seçenek size sunulmaz. Bu görevi tam olarak izlemek istiyorsanız, bu etki alanı denetleyicisinden tüm rolleri kaldırın ve yeniden başlatın.
8. Database, Log Files ve SYSVOL dizinlerini değiştirmeden bırakın; **Next** (İleri) düğmesini tıklatın.
9. Directory Services Restore Mode Administrator Password (Dizin Hizmetleri Geri Yükleme Kipi Yönetici Parolası) penceresindeki **Password** (Parola) ve **Confirm password** (Parolayı onayla) alanlarına bir parola yazın. **Next > Next** (İleri) düğmesini tıklatın. Son sihirbaz penceresinde **Reboot on tamamlanınca yeniden yükle** seçeneğini belirleyin.
10. Etki alanı denetleyicisi yeniden önyüklendiğinde, *wmq\Administrator* olarak oturum açın.  
Sunucu yöneticisi otomatik olarak başlatılır.
11. *wmq.example.com\Users* klasörünü aç
  - a) **Server Manager > Roller > Active Directory Domain Services > wmq.example.com > Kullanıcılar'** i açın.
12. **Kullanıcılar > Yeni > Grup** ögesini sağ tıklatın.
  - a) **Grup adı** alanına bir grup adı yazın.  
**Not:** Tercih edilen grup adı *Domain\_mqm*. Tam olarak gösterildiği gibi yazın.
    - *Domain\_mqm* grubunun çağrılmasının, bir etki alanı iş istasyonundaki ya da sunucusundaki Prepare IBM MQ Wizard davranışı değiştirilir. Prepare IBM MQ Wizard 'in etki alanındaki her yeni IBM MQ kuruluşunda *Domain\_mqm* grubunu yerel *mqm* grubuna otomatik olarak eklemesine neden olur.
    - İş istasyonlarını ya da sunucuları, *Domain\_mqm* genel grubu olmayan bir etki alanına kurabilirsiniz. Bunu yaparsanız, *Domain\_mqm* grubuyla aynı özelliklere sahip bir grup tanımlamanız gerekir. Bu grubu ya da bu grubun üyesi olan kullanıcıları, IBM MQ 'un bir etki alanında kurulu olduğu her yerde yerel *mqm* grubunun üyeleri yapmalısınız. Etki alanı kullanıcılarını birden çok gruba yerleştirebilirsiniz. Her biri ayrı olarak yönetmek istediğiniz bir kuruluş kümesine karşılık gelen birden çok etki alanı grubu oluşturun. Yönettikleri kuruluşlara göre etki alanı kullanıcılarını farklı etki alanı gruplarına ayırın. Her bir etki alanı grubunu ya da

grubunu yerel mqm farklı IBM MQ kuruluş grubuna ekleyin. Yalnızca belirli bir yerel mqm grubunun üyesi olan etki alanı gruplarındaki etki alanı kullanıcıları, o kuruluş için kuyruk yöneticileri yaratabilir, bunları yönetebilir ve çalıştırabilir.

- Bir etki alanındaki bir iş istasyonuna ya da sunucuya IBM MQ ürününü kurarken aday gösterdiğiniz etki alanı kullanıcısı, Domain mqm grubunun ya da Domain mqm grubuyla aynı özelliklerle tanımladığınız diğer bir grubun üyesi olmalıdır.

b) **Genel** 'i **Grup kapsamı** olarak tıklatın ya da **Evrensel** olarak değiştirin. **Güvenlik** seçeneğini **Grup tipi** olarak tıklatın. **Tamam**'i tıklatın.

13. Domain mqm genel grubunun haklarına **İzin Ver Grup üyeliğini oku** ve **İzin Ver groupMembershipSAM ' I Oku** haklarını ekleyin.

- Server Manager işlem çubuğunda **Görüntüle** > **Gelişmiş özellikler** seçeneğini tıklatın.
- Server Manager gezinme ağacında **Users** (Kullanıcılar) seçeneğini tıklatın.
- Kullanıcılar penceresinde **Etki alanı mqm** > **Özellikler** ögesini sağ tıklatın.
- Güvenlik** > **Gelişmiş** > **Ekle ...** düğmesini tıklatın. Domain mqm yazın ve **Adları denetle** > **Tamam** düğmesini tıklatın.

**Ad** alanı, Domain mqm (*domain name* \ Domain mqm) dizisiyle önceden doldurulur.

- Özellikler**'i tıklatın. **Apply to** (Uygula) listesinde **Descendant User Objects** (Alt Kullanıcı Nesneleri) seçeneğini belirleyin.
- İzinler** listesinden **Grup üyeliğini oku** ve **Oku groupMembershipSAM İzin ver** onay kutularını seçin; **Tamam** > **Uygula** > **Tamam** > **Tamam** düğmesini tıklatın.

14. Domain mqm genel grubuna iki ya da daha fazla kullanıcı ekleyin.

Örnekte bir kullanıcı (*wmquser1*) IBM MQ hizmetini çalıştırır ve diğer kullanıcı (*wmquser2*) etkileşimli olarak kullanılır.

Etki alanı yapılışında diğer güvenlik grubunu kullanan bir kuyruk yöneticisi yaratmak için etki alanı kullanıcısı gereklidir. Bir yönetici **crtmqm** komutunu çalıştırma yetkisine sahip olsa da, kullanıcı kimliğinin yönetici olması yeterli değildir. Denetimci olabilecek etki alanı kullanıcısı, diğer güvenlik grubunun yanı sıra yerel mqm grubunun da üyesi olmalıdır.

Bu örnekte, Domain mqm genel grubunun *wmquser1* ve *wmquser2* üyelerini oluşturmanız gerekir. Prepare IBM MQ Wizard, Domain mqm ürününü sihirbazın çalıştırıldığı yerel mqm grubunun bir üyesi olarak otomatik olarak yapılandırır.

Tek bir bilgisayarda her IBM MQ kuruluşu için IBM MQ hizmetini çalıştırmak üzere farklı bir kullanıcı sağlamanız gerekir. Aynı kullanıcıları farklı bilgisayarlarda yeniden kullanabilirsiniz.

- Server Manager gezinme ağacında **Kullanıcılar** > **Yeni** > **Kullanıcı** seçeneğini tıklatın.
- Yeni Nesne-Kullanıcı penceresinde, **Kullanıcı oturum açma adı** alanına *wmquser1* yazın. **Ad** alanına *WebSphere* ve **Soyadı** alanına *MQ1* yazın. **İleri**'yi tıklatın.
- Parola** ve **Parolayı onayla** alanlarına bir parola yazın ve **Kullanıcı sonraki oturum açmada parolayı değiştirmeli** onay kutusunun işaretini kaldırın. **İleri** > **Sondüğmesini** tıklatın.
- Users** (Kullanıcılar) penceresinde **WebSphere MQ** > **Add to a group ...** (Gruba ekle ...) seçeneğini sağ tıklatın. Domain mqm yazın ve **Adları Denetle** > **Tamam** > **Tamam** düğmesini tıklatın.
- WebSphere MQ2** ögesini *wmquser2* olarak eklemek için **a** - **d** arasındaki adımları yineleyin.

15. IBM MQ hizmet olarak çalıştırılıyor.

Hizmet olarak IBM MQ komutunu çalıştırmanız gerekiyorsa ve etki alanı kullanıcısına (etki alanı yöneticinizden edindiğiniz) hizmet olarak çalışması için erişim vermeniz gerekiyorsa, aşağıdaki yordamı gerçekleştirin:

- Başlat** > **Çalıştır ...** seçeneklerini tıklatın.  
secpo1.msc komutunu yazın ve **Tamam** düğmesini tıklatın.
- Güvenlik Ayarları** > **Yerel İlkeler** > **Kullanıcı Hakları Atamaları** öğelerini açın.  
İlkeler listesinde **Hizmet olarak oturum aç** > **Özellikler** seçeneğini sağ tıklatın.
- Kullanıcı ya da Grup Ekle ...** düğmesini tıklatın.

Etki alanı yöneticinizden aldığınız kullanıcının adını yazın ve **Adları Denetle** seçeneğini tıklatın.

- d) Windows Güvenlik penceresi tarafından istenirse, yeterli yetkisi olan bir hesap kullanıcısının ya da yöneticisinin kullanıcı adını ve parolasını yazın ve **Tamam > Uygula > Tamamdüğmesini** tıklatın. Yerel Güvenlik İlkesi penceresini kapatın.

**Not:** Windows Server 2008 ve Windows Server 2012 'de Kullanıcı Hesabı Denetimi (UAC) varsayılan olarak etkindir.

UAC özelliği, Administrators (Yöneticiler) grubunun üyesi olsalar bile, kullanıcıların belirli işletim sistemi olanaklarında gerçekleştirebilecekleri işlemleri kısıtlar. Bu kısıtlamayı aşmak için uygun adımları atmanız gerekir.

## Sonraki adım

Sonraki göreve ( [“IBM MQ ' in Windows etki alanındaki bir sunucuya ya da iş istasyonuna kurulması” sayfa 506](#)) ilerleyin.

### İlgili görevler

**Windows** [IBM MQ ' in Windows etki alanındaki bir sunucuya ya da iş istasyonuna kurulması](#)

**Windows** [Windows üzerinde kuyruk yöneticisi verileri ve günlük dosyaları için paylaşılan dizin oluşturma](#)

**Windows** [Alternatif bir genel güvenlik grubu tarafından yetkilendirilen paylaşılan verileri ve günlük dosyalarını okuma ve yazma](#)

**Windows** [IBM MQ ' in Windows etki alanındaki bir sunucuya ya da iş istasyonuna kurulması](#)

Bu görevde, IBM MQ ürününü *wmq.example.com* Windows etki alanındaki bir sunucuya ya da iş istasyonuna kurar ve yapılandırır.

Üretim ölçeği yapılandırmasında, yapılandırmayı var olan bir etki alanına uyarlamanız gerekebilir. Örneğin, farklı paylaşımları yetkilendirmek ve kuyruk yöneticilerini çalıştıran kullanıcı kimliklerini gruplamak için farklı etki alanı grupları tanımlayabilirsiniz.

Örnek yapılandırma üç sunucudan oluşur:

#### ***sun***

Bir Windows Server 2008 etki alanı denetleyicisi. *Sun, marsve venus*çeren *wmq.example.com* etki alanının sahibi olur. Şekil amacıyla, dosya sunucusu olarak da kullanılır.

#### ***mars***

İlk IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. Çok eşgörünümlü kuyruk yöneticisinin *QMGR*adlı bir eşgörünümünü içerir.

#### ***venus***

İkinci IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. *QMGR*adlı çok eşgörünümlü kuyruk yöneticisinin ikinci eşgörünümünü içerir.

Örnekteki italik adları, seçtiğiniz adlarla değiştirin.

## Başlamadan önce

**Önemli:** Varsayılan olarak, Windows 10 sürüm 1607 ve Windows Server 2016 ile başlayan bilgisayarlar, Windowsürününün önceki sürümlerinden daha kısıtlayıcıdır.

Bu değişiklik, Güvenlik Hesapları Yöneticisi 'ne (SAM) uzak çağrı yapmasına izin verilen istemcileri sınırlar ve kuyruk yöneticileri başlatılmadığı için IBM MQ ögesini etkileyebilir. IBM MQ bir etki alanı hesabı olarak yapılandırıldığında, SAM 'ye erişim IBM MQ ' in çalışması için kritik önem taşır.

1. *wmq.example.com* etki alanı için bir etki alanı denetleyicisi ( *sun*) oluşturmak üzere [“Windows üzerinde Active Directory ve DNS etki alanı oluşturma” sayfa 503](#) içindeki adımları gerçekleştirin. İtalik adları yapılandırmanıza uyacak şekilde değiştirin.

2. IBM MQ üzerinde çalıştırabileceğiniz diğer Windows sürümleri için bkz. Windows sistemlerinde donanım ve yazılım gereksinimleri .

## Bu görev hakkında

Bu görevde, *wmq.example.com* etki alanının bir üyesi olarak *mars* adlı bir Windows Server 2008 'i yapılandırırınız. IBM MQürününü kurar ve kuruluşu, *wmq.example.com* etki alanının bir üyesi olarak çalışacak şekilde yapılandırırınız.

Bu görev, kuyruk yöneticisi verilerine ve günlük dosyalarına erişimi gösteren ilgili görevler kümesinden biridir. Görevler, seçtiğiniz bir dizinde saklanan verileri ve günlük dosyalarını okuma ve yazma yetkisi olan bir kuyruk yöneticisinin nasıl yaratılacağını gösterir. Görevle birlikte, "Windows etki alanları ve çok eşgörümlü kuyruk yöneticileri" sayfa 498.

Görevin amaçları açısından etki alanı denetleyicisi anasistem adı *sun*, iki IBM MQ sunucusu *mars* ve *venus* olarak adlandırılır. Etki alanına *wmq.example.com* adı verilir. Görevdeki tüm italik adları kendi seçtiğiniz adlarla değiştirebilirsiniz.

## Yordam

1. Etki alanı denetleyicisini ( *sun.wmq.example.com* ) DNS sunucusu olarak *mars* ' e ekleyin.
  - a) *mars*' da *mars\Administrator* olarak oturum açın ve **Start**(Başlat) seçeneğini tıklayın.
  - b) **Ağ > Özellikler > Ağ bağlantılarını yönet** seçeneğini sağ tıklayın.
  - c) Ağ bağdaştırıcısını sağ tıklayın ve **Özellikler** seçeneğini belirleyin.

Sistem, bağlantının kullandığı öğeleri listeleyen Yerel Ağ Bağlantısı Özellikleri penceresi ile yanıt verir.
  - d) Local Area Connection Properties (Yerel Ağ Bağlantısı Özellikleri) penceresindeki öğeler listesinden **Internet Protocol Version 4** ya da **Internet Protocol IBM WebSphere MQ 6** (Internet İletişim Kuralı) seçeneğini belirleyin. **Özellikler > Gelişmiş ...** seçeneğini tıklayın. ve **DNS** sekmesini tıklayın.
  - e) DNS sunucusu adreslerinin altında **Ekle ...** düğmesini tıklayın.
  - f) DNS sunucusu olan etki alanı denetleyicisinin IP adresini yazın ve **Ekledüğmesini** tıklayın.
  - g) **Bu DNS soneklerini ekle > Ekle ...** düğmesini tıklayın.
  - h) *wmq.example.com* yazın ve **Ekledüğmesini** tıklayın.
  - i) **Bu bağlantı için DNS soneki** alanına *wmq.example.com* yazın.
  - j) **Bu bağlantının adresini DNS ' de kaydet** ve **Bu bağlantının sonekini DNS kaydında kullan** seçeneğini belirleyin. **Tamam > Tamam > Kapat** düğmesini tıklayın.
  - k) Bir komut penceresi açın ve TCP/IP ayarlarını gözden geçirmek için **ipconfig /all** komutunu yazın.
2. *mars* üzerinde, bilgisayarı *wmq.example.com* etki alanına ekleyin.
  - a) **Start** (Başlat) düğmesini tıklayın.
  - b) **Bilgisayar > Özellikler** seçeneğini sağ tıklayın. Bilgisayar adı, etki alanı ve çalışma grubu ayarları bölümünde **Ayarları değiştir** seçeneğini tıklayın.
  - c) System Properties (Sistem Özellikleri) pencerelerinde **Change ...**(Değiştir ...) düğmesini tıklayın.
  - d) Domain (Etki Alanı) seçeneğini tıklayın, *wmq.example.com* yazın ve **OK**(Tamam) düğmesini tıklayın.
  - e) Bilgisayarın etki alanına katılmasına izin veren etki alanı denetleyicisi yöneticisinin **Kullanıcı adı** ve **Parola** değerini yazın ve **Tamam** düğmesini tıklayın.
  - f) "*wmq.example.com* etki alanına hoş geldiniz" iletisine yanıt olarak **Tamam > Tamam > Kapat > Şimdi Yeniden Başlat** düğmesini tıklayın.
3. Bilgisayarın *wmq.example.com* etki alanının bir üyesi olup olmadığını denetleyin
  - a) *sun* işletim sistemi üzerinde, etki alanı denetleyicisinde *wmq\Administrator* olarak oturum açın.

- b) **Server Manager > Active Directory Domain Services > wmq.example.com > Computers** ' ı açın ve Computers (Bilgisayarlar) penceresinde *mars* dosyasını denetleyin.
4. IBM MQ for Windows ürününü *mars* üzerine kurun.
- IBM MQ for Windows kuruluş sihirbazını çalıştırma hakkında daha fazla bilgi için bkz. [IBM MQ sunucusunu Windows üzerine kurma](#) .
- a) *mars*' da yerel yönetici olarak oturum açın, *mars\Administrator*.
- b) IBM MQ for Windows kuruluş ortamında **Setup** komutunu çalıştırın.
- IBM MQ Launchpad uygulaması başlar.
- c) Önkoşul olan yazılımın kurulu olup olmadığını denetlemek için **Software Requirements** (Yazılım Gereksinimleri) seçeneğini tıklattın.
- d) Bir etki alanı kullanıcı kimliğini yapılandırmak için **Ağ Yapılandırması > Evet** seçeneğini tıklattın.
- [“Windows üzerinde Active Directory ve DNS etki alanı oluşturma” sayfa 503](#) görevi, bu görev kümesi için bir etki alanı kullanıcı kimliğini yapılandırır.
- e) **IBM MQ Installation** (Kuruluş) seçeneğini tıklattın, bir kuruluş dili seçin ve IBM MQ Installer düğmesini tıklattın.
- f) Lisans sözleşmesini onaylayın ve varsayılan yapılandırmayı kabul etmek için **Next > Next > Install** (İleri) düğmesini tıklattın. Kuruluşun tamamlanmasını bekleyin ve **Finish** (Son) düğmesini tıklattın.
- Kuruluşun adını değiştirmek, farklı bileşenler kurmak, kuyruk yöneticisi verileri ve günlükleri için farklı bir dizin yapılandırmak ya da farklı bir dizine kurmak isteyebilirsiniz. Bu durumda, **Tipik** yerine **Özel** ' i tıklattın.
- IBM MQ kurulur ve kuruluş programı Prepare IBM MQ Wizard' ı başlatır.
- Önemli:** Sihirbazı henüz çalıştırmayın.
5. IBM MQ hizmetini çalıştıracak kullanıcıyı **Hizmet olarak çalıştır** hakkıyla yapılandırın.
- Yerel mqm grubunun mu, Domain mqm grubunun mu, yoksa IBM MQ hizmetini sağla çalıştıracak kullanıcının mı yapılandırılacağını seçin. Örnekte, kullanıcıya hak verirsiniz.
- a) **Başlat > Çalıştır ...** düğmesini tıklattın, **secp01.msc** komutunu yazın ve **OK** (Tamam) düğmesini tıklattın.
- b) **Güvenlik Ayarları > Yerel İlkeler > Kullanıcı Hakları Atamaları** ' nı açın. İlkeler listesinde **Hizmet olarak oturum aç > Özellikler** seçeneğini sağ tıklattın.
- c) **Kullanıcı ya da Grup Ekle ...** düğmesini tıklattın. ve *wmquser1* yazın ve **Adları Denetle** ' yi tıklattın
- d) *wmq\Administrator* etki alanı yöneticisinin kullanıcı adını ve parolasını yazın ve **Tamam > Uygula > Tamam** düğmesini tıklattın. Yerel Güvenlik İlkesi penceresini kapatın.
6. Prepare IBM MQ Wizard komutunu çalıştırın.
- Daha fazla bilgi için bkz. [IBM MQ ürününü Prepare IBM MQ Wizard ile yapılandırma](#).
- a) IBM MQ Installer, Prepare IBM MQ Wizard programını otomatik olarak çalıştırır.
- Sihirbazı el ile başlatmak için, **Başlat > Tüm programlar > IBM MQ** klasöründe Prepare IBM MQ Wizard kısayolunu bulun. Çoklu kuruluş yapılandırmasında IBM MQ kuruluşuna karşılık gelen kısayolu seçin.
- b) **Next** (İleri) düğmesini tıklattın ve **Yes** (Evet) düğmesini tıklatarak "Tanımlama (Tanımla) sorusuna yanıt olarak ağda bir Windows 2000 ya da daha sonraki bir etki alanı denetleyicisi olup olmadığını belirleyin" sorusunu tıklattın.
- c) Windows etki alanı kullanıcıları için ilk Configuring IBM MQ for Windows (Yapılandırma) penceresinde **Yes > Next** (Evet) seçeneğini tıklattın.
- d) İkinci Configuring IBM MQ for Windows for Windows domain users (etki alanı kullanıcıları için yapılandırma) penceresinde, **Domain** (Etki Alanı) alanına *wmq* yazın. **User name** (Kullanıcı adı) alanına *wmquser1* yazın ve ayarladıysanız, parolayı **Password** (Parola) alanına yazın. **İleri**' yi tıklattın.



Sihirbaz, IBM MQ ' i *wmquser1* ile yapılandırır ve başlatır.

- e) Sihirbazın son sayfasında, gerek duyduğunuz onay kutularını seçin ya da temizleyin ve **Sondüğmesini** tıklayın.

## Sonraki adım

1. Kuruluş ve yapılandırmanın doğru çalıştığını doğrulamak için [“Yerel mqm grubu tarafından yetkilendirilen veri ve günlük dosyalarının okunması ve yazılması” sayfa 530](#) görevini yapın.
2. Çok eşgörünümlü bir kuyruk yöneticisinin veri ve günlük dosyalarını saklamak üzere bir dosya paylaşımı yapılandırmak için [“Windows üzerinde kuyruk yöneticisi verileri ve günlük dosyaları için paylaşılan dizin oluşturma” sayfa 509](#) görevini yapın.

## İlgili görevler

**Windows** [Windows üzerinde Active Directory ve DNS etki alanı oluşturma](#)

**Windows** [Windows üzerinde kuyruk yöneticisi verileri ve günlük dosyaları için paylaşılan dizin oluşturma](#)

**Windows** [Alternatif bir genel güvenlik grubu tarafından yetkilendirilen paylaşılan verileri ve günlük dosyalarını okuma ve yazma](#)

## İlgili başvurular

[Bir IBM MQ Windows Hizmeti için gerekli kullanıcı hakları](#)

**Windows** [Windows üzerinde kuyruk yöneticisi verileri ve günlük dosyaları için paylaşılan dizin oluşturma](#)

Bu görev, kuyruk yöneticisi verilerine ve günlük dosyalarına erişimi gösteren ilgili görevler kümesinden biridir. Görevler, seçtiğiniz bir dizinde saklanan verileri ve günlük dosyalarını okuma ve yazma yetkisi olan bir kuyruk yöneticisinin nasıl yaratılacağını gösterir.

Üretim ölçeği yapılandırmasında, yapılandırmayı var olan bir etki alanına uyarlamamız gerekebilir. Örneğin, farklı paylaşımları yetkilendirmek ve kuyruk yöneticilerini çalıştıran kullanıcı kimliklerini gruplamak için farklı etki alanı grupları tanımlayabilirsiniz.

Örnek yapılandırma üç sunucudan oluşur:

### **sun**

Bir Windows Server 2008 etki alanı denetleyicisi. *Sun, mars ve venus* çeren *wmq.example.com* etki alanının sahibi olur. Şekil amacıyla, dosya sunucusu olarak da kullanılır.

### **mars**

İlk IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. Çok eşgörünümlü kuyruk yöneticisinin *QMGR* adlı bir eşgörünümünü içerir.

### **venus**

İkinci IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. *QMGR* adlı çok eşgörünümlü kuyruk yöneticisinin ikinci eşgörünümünü içerir.

Örnekteki italik adları, seçtiğiniz adlarla değiştirin.

## Başlamadan önce

1. Bu görevi tam olarak belgelendiği şekilde yapmak için, [“Windows üzerinde Active Directory ve DNS etki alanı oluşturma” sayfa 503](#) etki alanı denetleyicisinde *sun.wmq.example.com* etki alanını oluşturmak için *sun* görevindeki adımları gerçekleştirin. İtalik adları yapılandırmanıza uyacak şekilde değiştirin.

## Bu görev hakkında

Bu görev, kuyruk yöneticisi verilerine ve günlük dosyalarına erişimi gösteren ilgili görevler kümesinden biridir. Görevler, seçtiğiniz bir dizinde saklanan verileri ve günlük dosyalarını okuma ve yazma yetkisi olan bir kuyruk yöneticisinin nasıl yaratılacağını gösterir. Görevle birlikte, [“Windows etki alanları ve çok eşgörünümlü kuyruk yöneticileri” sayfa 498](#).

Görevde, bir veri ve günlük dizini içeren bir paylaşım ve paylaşım erişim yetkisi vermek için bir genel grup yaratırsınız. Paylaşımı yetkilendiren genel grubun adını, -a değiştirilmesinde **cdmcm** komutuna geçirir. Genel grup, bu paylaşımın kullanıcılarını diğer paylaşımların kullanıcılarından ayırma esnekliği sağlar. Bu esnekliğe gerek duymuyorsanız, yeni bir genel grup oluşturmak yerine Domain mcm grubuyla paylaşımı yetkilendirin.

Bu görevde paylaşım için kullanılan genel gruba *wmqha* adı verilir ve paylaşım *wmq* adı verilir. They are defined on the domain controller *sun* in the Windows domain *wmq.example.com*. Paylaşım, *wmqha* genel grubu için tam denetim izinlerine sahiptir. Görevdeki italik adları seçtiğiniz adlarla değiştirin.

Bu görevin amaçları açısından etki alanı denetleyicisi, dosya sunucusuyla aynı sunucudur. Pratik uygulamalarda, performans ve kullanılabilirlik için dizin ve dosya hizmetlerini farklı sunucular arasında bölün.

Kuyruk yöneticisinin altında çalıştığı kullanıcı kimliğini iki grubun üyesi olacak şekilde yapılandırmanız gerekir. Bir IBM MQ sunucusundaki yerel mcm grubunun ve *wmqha* genel grubunun bir üyesi olmalıdır.

Bu görev kümesinde, kuyruk yöneticisi bir hizmet olarak çalıştığında *wmquser1* kullanıcı kimliği altında çalışır, bu nedenle *wmquser1*, *wmqha* in bir üyesi olmalıdır. Kuyruk yöneticisi etkileşimli olarak çalıştığında, *wmquser2* kullanıcı kimliği altında çalışır; bu nedenle *wmquser2*, *wmqha* un bir üyesi olmalıdır. Hem *wmquser1* hem de *wmquser2*, Domain mcm genel grubunun üyeleridir. Domain mcm, *mars* ve *venus* IBM MQ sunucularındaki yerel mcm grubunun bir üyesidir. Bu nedenle *wmquser1* ve *wmquser2*, her iki IBM MQ sunucusunda yerel mcm grubunun üyeleridir.

## Yordam

1. Etki alanı denetleyicisinde, *sun.wmq.example.com* etki alanı yöneticisi olarak oturum açın.
2. *wmqha* genel grubunu oluşturun.
  - a) **Server Manager** > **Roller** > **Active Directory Domain Services** > *wmq.example.com* > **Kullanıcılar** i açın.
  - b) *wmq.example.com*\Users klasörünü aç
  - c) **Kullanıcılar** > **Yeni** > **Grup** öğesini sağ tıklayın.
  - d) **Grup adı** alanına *wmqha* yazın.
  - e) **Genel** seçeneğini, **Grup tipi** olarak **Grup kapsamı** ve **Güvenlik** olarak tıklayın. **Tamam**'i tıklayın.
3. Etki alanı kullanıcılarını *wmquser1* ve *wmquser2* genel grubuna (*wmqha*) ekleyin.
  - a) Server Manager gezinme ağacında **Kullanıcılar** 'i tıklayın ve kullanıcılar listesinde *wmqha* > **Özellikler** ' i sağ tıklayın.
  - b) *wmqha* Özellikler penceresinde Üyeler sekmesini tıklayın.
  - c) **Ekle ...** düğmesini tıklayın; *wmquser1* ; *wmquser2* yazın ve **Adları Denetle** > **Tamam** > **Uygula** > **Tamam** düğmesini tıklayın.
4. Kuyruk yöneticisi verilerini ve günlük dosyalarını içerecek dizin ağacını yaratın.
  - a) Bir komut istemi açın.
  - b) Komutu yazın:

```
md c:\wmq\data, c:\wmq\logs
```
5. *wmqha* genel grubuna, *c:\wmq* dizinleri için tam denetim izni ve paylaşım yetkisi verin.
  - a) Windows Explorer 'da **c:\wmq** > **Properties**(Özellikler) öğesini sağ tıklayın.
  - b) **Güvenlik** sekmesini tıklayın ve **Gelişmiş** > **Düzenle ...** seçeneğini tıklayın.
  - c) **Bu nesnenin iyesinden edinilebilir izinleri ekle** onay kutusunun işaretini kaldırın. Windows Security (Güvenlik) penceresinde **Copy** (Kopyala) seçeneğini tıklayın.
  - d) **İzin girişleri** listesinden Kullanıcılar için satırları seçin ve **Kaldır** ' i tıklayın. SYSTEM, Administrators ve CREATOR OWNER satırlarını **İzin girişleri** listesinde bırakın.

- e) **Ekle ...**düğmesini tıklatın, ve *wmq* genel grubunun adını yazın. **Adları Denetle > Tamam**düğmesini tıklatın.
- f) *wmq* için İzin Girdisi penceresinde, **İzinler**listesinden **Tam Denetim** seçeneğini belirleyin.
- g) **Tamam > Uygula > Tamam > Tamam > Tamam** düğmesini tıklatın.
- h) Windows Explorer 'da **c: \wmq > Paylaş ...**öğesini sağ tıklatın.
- i) **Gelişmiş Paylaşım ...** seçeneğini tıklatın. ve **Bu klasörü paylaş** onay kutusunu seçin. Paylaşım adını *wmq*olarak bırakın.
- j) **İzinler > Ekle ...**, ve *wmq* genel grubunun adını yazın. **Adları Denetle > Tamam**düğmesini tıklatın.
- k) **Grup ya da kullanıcı adları**listesinde *wmqha* öğesini seçin. **wmqha için izinler** listesindeki **Tam Denetim** onay kutusunu işaretleyin; **Uygula** simgesini tıklatın.
- l) **Grup ya da kullanıcı adları**listesinde *Administrators* öğesini seçin. **Yöneticiler için izinler** listesindeki **Tam Denetim** onay kutusunu işaretleyin; **Uygula > -Tamam. > -Tamam. > Kapat** simgesini tıklatın.

## Sonraki adım

IBM MQ sunucularının her birinden paylaşılan dizinleri okuyabildiğinizi ve bu dizinlere dosya yazabildiğinizi denetleyin. IBM MQ Hizmet kullanıcı kimliğini *wmquser1* ve etkileşimli kullanıcı kimliğini (*wmquser2*) denetleyin.

1. Uzak masaüstü kullanıyorsanız, *mars* üzerindeki Remote Desktop Users yerel grubuna *wmq\wmquser1* ve *wmquser2* eklemelisiniz.
  - a. *mars* ' da *wmq\Administrator* olarak oturum aç
  - b. Yerel Kullanıcılar ve Gruplar penceresini açmak için **lusrmgr.msc** komutunu çalıştırın.
  - c. **Gruplar**düğmesini tıklatın. **Uzak Masaüstü Kullanıcıları > Özellikler > Ekle ...**seçeneğini sağ tıklatın. *wmquser1* ; *wmquser2* yazın ve **Adları Denetle** seçeneğini tıklatın.
  - d. *wmq\Administratoretki* alanı yöneticisinin kullanıcı adını ve parolasını yazın ve **Tamam > Uygula > Tamam**düğmesini tıklatın.
  - e. Yerel Kullanıcılar ve Gruplar penceresini kapatın.
2. *mars* ' da *wmq\wmquser1*olarak oturum açın.
  - a. Bir Windows Explorer penceresi açın ve `\\sun\wmq` yazın.  
Sistem, *sun.wmq.example.com* üzerinde *wmq* paylaşımını açarak yanıt verir ve veri ve günlük dizinlerini listeler.
  - b. Veri alt dizininde bir dosya oluşturarak, içerik ekleyerek, okuyarak ve silerek *wmquser1* izinlerini denetleyin.
3. *mars* ' da *wmq\wmquser2*olarak oturum açın ve denetimleri yineleyin.
4. Paylaşılan verileri ve günlük dizinlerini kullanmak üzere bir kuyruk yöneticisi yaratmak için sonraki görevi yapın; bkz. "Alternatif bir genel güvenlik grubu tarafından yetkilendirilen paylaşılan verileri ve günlük dosyalarını okuma ve yazma" sayfa 512.

## İlgili görevler

- Windows** [Windows üzerinde Active Directory ve DNS etki alanı oluşturma](#)
- Windows** [IBM MQ ' in Windows etki alanındaki bir sunucuya ya da iş istasyonuna kurulması](#)
- Windows** [Alternatif bir genel güvenlik grubu tarafından yetkilendirilen paylaşılan verileri ve günlük dosyalarını okuma ve yazma](#)

## Windows

*Alternatif bir genel güvenlik grubu tarafından yetkilendirilen paylaşılan verileri ve günlük dosyalarını okuma ve yazma*

Bu görev, **crtmqm** komutunda -a işaretinin nasıl kullanılacağını gösterir. -a işareti, kuyruk yöneticisinin diğer güvenlik grubunu kullanarak uzak bir dosya paylaşımında günlük ve veri dosyalarına erişmesini sağlar.

Üretim ölçeği yapılandırmasında, yapılandırmayı var olan bir etki alanına uyarlamamız gerekebilir. Örneğin, farklı paylaşımları yetkilendirmek ve kuyruk yöneticilerini çalıştıran kullanıcı kimliklerini gruplamak için farklı etki alanı grupları tanımlayabilirsiniz.

Örnek yapılandırma üç sunucudan oluşur:

### **sun**

Bir Windows Server 2008 etki alanı denetleyicisi. *Sun, marsve venus* içeren *wmq.example.com* etki alanının sahibi olur. Şekil amacıyla, dosya sunucusu olarak da kullanılır.

### **mars**

İlk IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. Çok eşgörünümlü kuyruk yöneticisinin *QMGR* adlı bir eşgörünümünü içerir.

### **venus**

İkinci IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. *QMGR* adlı çok eşgörünümlü kuyruk yöneticisinin ikinci eşgörünümünü içerir.

Örnekteki italik adları, seçtiğiniz adlarla değiştirin.

## **Başlamadan önce**

Aşağıdaki görevlerdeki adımları gerçekleştirin. Görevler etki alanı denetleyicisini ve etki alanını oluşturur, IBM MQ for Windows ürününü tek bir sunucuya kurar ve veri ve günlük dosyaları için dosya paylaşımı yaratır. Var olan bir etki alanı denetleyicisini yapılandırıyorsanız, yeni bir Windows Server 2008 'deki adımları denemeniz yararlı olabilir. Adımları etki alanınıza uyarlayabilirsiniz.

1. [“Windows üzerinde Active Directory ve DNS etki alanı oluşturma”](#) sayfa 503.
2. [“IBM MQ ' in Windows etki alanındaki bir sunucuya ya da iş istasyonuna kurulması”](#) sayfa 506.
3. [“Windows üzerinde kuyruk yöneticisi verileri ve günlük dosyaları için paylaşılan dizin oluşturma”](#) sayfa 509.

## **Bu görev hakkında**

Bu görev, kuyruk yöneticisi verilerine ve günlük dosyalarına erişimi gösteren ilgili görevler kümesinden biridir. Görevler, seçtiğiniz bir dizinde saklanan verileri ve günlük dosyalarını okuma ve yazma yetkisi olan bir kuyruk yöneticisinin nasıl yaratılacağını gösterir. Görevle birlikte, [“Windows etki alanları ve çok eşgörünümlü kuyruk yöneticileri”](#) sayfa 498.

Bu görevde, verilerini ve günlüklerini bir dosya sunucusundaki uzak bir dizinde saklayan bir kuyruk yöneticisi yaratırsınız. Bu örneğin amaçları açısından, dosya sunucusu etki alanı denetleyicisiyle aynı sunucudur. Veri ve günlük klasörlerini içeren dizin, *wmqhagenel* grubuna verilen tam denetim izniyle paylaşılır.

## **Yordam**

1. *mar* setki alanı sunucusunda yerel yönetici olarak oturum açın, *mars\Administrator*.
2. Bir komut penceresi açın.
3. IBM MQ hizmetini yeniden başlatın.

Hizmetin altında çalıştığı kullanıcı kimliğinin, kendisi için yapılandırıdığınız ek güvenlik kimlik bilgilerini edinmesi için hizmeti yeniden başlatmanız gerekir.

Komutları yazın:

```
endmqsvc  
strmqsvc
```

Sistem yanıtları:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

Ve:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' started successfully.
```

4. Kuyruk yöneticisini yaratın.

```
crtmqm -a wmq\wmqha -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\sun\wmq\data -ld \\sun\wmq\logs  
QMGR
```

"*wmq\wmqha*" genel grubunun tam etki alanı adını belirterek *wmqha* alternatif güvenlik grubunun *wmqetki* alanını belirtmeniz gerekir.

\\sun\wmq paylaşımının UNC (Universal Naming Convention; Evrensel Adlandırma Kuralı) adını hecelemeli ve eşlenmiş bir sürücü başvurusu kullanmamalısınız.

Sistemin yapması gereken:

```
IBM MQ queue manager created.  
Directory '\\sun\wmq\data\QMGR' created.  
The queue manager is associated with installation '1'  
Creating or replacing default objects for queue manager 'QMGR'  
Default objects statistics : 74 created. 0 replaced.  
Completing setup.  
Setup completed.
```

## Sonraki adım

Bir iletiyi kuyruğa koyup alarak kuyruk yöneticisini sınavın.

1. Kuyruk yöneticisini başlatın.

```
strmqm QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. Bir test kuyruğu yaratın.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

Sistemin yapması gereken:

5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.

```
1 : define qlocal(QTEST)
AMQ8006: IBM MQ queue created.
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

3. **amqspu**törnek programını kullanarak bir sınama iletisi koyun.

```
echo 'A test message' | amqspu QTEST QMGR
```

Sistemin yapması gereken:

```
Sample AMQSPUT0 start
target queue is QTEST
Sample AMQSPUT0 end
```

4. **amqsge**törnek programını kullanarak sınama iletisini alın.

```
amqsge QTEST QMGR
```

Sistemin yapması gereken:

```
Sample AMQSGET0 start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. Kuyruk yöneticisini durdurun.

```
endmqm -i QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended.
```

6. Kuyruk yöneticisini silin.

```
dltmqm QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' deleted.
```

7. Oluşturduğunuz dizinleri silin.

**İpucu:** Komut isteminin her bir dosyayı ya da dizini silmesini önlemek için komutlara /Q seçeneğini ekleyin.

```
del /F /S C:\wmq\*. *  
rmdir /S C:\wmq
```

## İlgili görevler

**Windows** [Windows üzerinde Active Directory ve DNS etki alanı oluşturma](#)

**Windows** [IBM MQ ' in Windows etki alanındaki bir sunucuya ya da iş istasyonuna kurulması](#)

**Windows** [Windows üzerinde kuyruk yöneticisi verileri ve günlük dosyaları için paylaşılan dizin oluşturma](#)

**Windows** [Windows etki alanı denetleyicilerinde çok eşgörümlü kuyruk yöneticisi yaratılması](#)

Bir örnek, etki alanı denetleyicilerinde Windows üzerinde çok eşgörümlü bir kuyruk yöneticisinin nasıl ayarlanacağını gösterir. Kurulum, üretim ölçeği olmak yerine, ilgili kavramları gösterir. Örnek, Windows Server 2008 'e dayalıdır. Adımlar, Windows Server 'in diğer sürümlerinde farklı olabilir.

Yapılandırma, mini etki alanı ya da "domainlet" kavramını kullanır; [Windows 2000, Windows Server 2003 ve Windows Server 2008 küme düğümlerini etki alanı denetleyicileri olarak başlıklı konuya bakın](#). Var olan bir etki alanına çok eşgörümlü kuyruk yöneticileri eklemek için bkz. [“Windows üzerinde etki alanı iş istasyonlarında ya da sunucularda çok eşgörümlü kuyruk yöneticisi yaratılması” sayfa 499](#).

Örnek yapılandırma üç sunucudan oluşur:

### **sun**

İlk etki alanı denetleyicisi olarak kullanılan bir Windows Server 2008 sunucusu. *sun*, *earthve mars* çeren *wmq.example.com* etki alanını tanımlar. Çok eşgörümlü kuyruk yöneticisinin QMGRadlı bir eşgörümlü içerir.

### **earth**

İkinci etki alanı denetleyicisi IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. QMGRadlı çok eşgörümlü kuyruk yöneticisinin ikinci eşgörümlü içerir.

### **mars**

Dosya sunucusu olarak kullanılan bir Windows Server 2008.

Örnekteki italik adları, seçtiğiniz adlarla değiştirin.

## Başlamadan önce

1. Windows üzerinde, kuyruk yöneticisi verilerini ve günlük dosyalarını saklamayı planladığınız dosya sistemini doğrulamanız gerekmez. [Paylaşılan dosya sistemi davranışının doğrulanması](#) denetim yordamı AIX and Linux için geçerlidir. Windows üzerinde, denetimler her zaman başarılı olur.
2. İlk etki alanı denetleyicisini oluşturmak için [“Windows üzerinde Active Directory ve DNS etki alanı oluşturma” sayfa 503](#) içindeki adımları gerçekleştirin.
3. İkinci bir etki alanı denetleyicisi eklemek, her iki etki alanı denetleyicisine IBM MQ for Windows kurmak ve kuruluşları doğrulamak için [“Örnek bir etki alanına ikinci bir Windows etki alanı denetleyicisi eklenmesi” sayfa 518](#) içindeki adımları gerçekleştirin.
4. İki etki alanı denetleyicisine IBM MQ kurmak için [“IBM MQ ' in örnek bir etki alanında Windows etki alanı denetleyicilerine kurulması” sayfa 520](#) başlıklı konudaki adımları gerçekleştirin.

## Bu görev hakkında

Aynı etki alanındaki bir dosya sunucusunda, kuyruk yöneticisi günlüğü ve veri dizinleri için bir paylaşım yaratın. Daha sonra, etki alanı denetleyicilerinden birinde dosya paylaşımını kullanan çok eşgörümlü bir kuyruk yöneticisinin ilk eşgörümlü yaratın. Diğer etki alanı denetleyicisinde diğer örneği oluşturun ve son olarak yapılandırmayı doğrulayın. Dosya paylaşımını bir etki alanı denetleyicisinde yaratabilirsiniz.

Örnekte, *sun* birinci etki alanı denetleyicisi, *earth* ikinci ve *mars* dosya sunucusudur.

## Yordam

1. Kuyruk yöneticisi verilerini ve günlük dosyalarını içerecek dizinleri oluşturun.

a) *mars*' da şu komutu yazın:

```
md c:\wmq\data , c:\wmq\logs
```

2. Kuyruk yöneticisi verilerini ve günlük dosyalarını içerecek dizinleri paylaşın.

*mqmetki* alanı yerel grubuna ve kuyruk yöneticisini oluşturmak için kullandığınız kullanıcı kimliğine tam denetim erişimi vermeniz gerekir. Bu örnekte, Domain Administrators üyesi olan kullanıcı kimlikleri kuyruk yöneticileri yaratma yetkisine sahip olur.

Dosya paylaşımı, etki alanı denetleyicileriyle aynı etki alanında bulunan bir sunucuda olmalıdır. Örnekte, *mars* sunucusu etki alanı denetleyicileriyle aynı etki alanındadır.

a) Windows Explorer 'da **c: \wmq > Properties**(Özellikler) ögesini sağ tıklayın.

b) **Güvenlik** sekmesini tıklayın ve **Gelişmiş > Düzenle ...**seçeneğini tıklayın.

c) **Bu nesnenin iyesinden edinilebilir izinleri ekle**onay kutusunun işaretini kaldırın. Windows Security (Güvenlik) penceresinde **Copy** (Kopyala) seçeneğini tıklayın.

d) **İzin girişleri** listesinden Kullanıcılar için satırları seçin ve **Kaldır** ' ı tıklayın. SYSTEM, Administrators ve CREATOR OWNER satırlarını **İzin girişleri**listesinde bırakın.

e) **Ekle ...**düğmesini tıklayın, ve *mqmetki* alanı yerel grubunun adını yazın. **Adları Denetle** düğmesini tıklayın.

f) Windows Güvenlik penceresine yanıt olarak Domain Administrator ' nin adını ve parolasını yazın ve **Tamam > Tamam**düğmesini tıklayın.

g) *wmq* için İzin Girdisi penceresinde, **İzinler**listesinden **Tam Denetim** seçeneğini belirleyin.

h) **Tamam > Uygula > Tamam > Tamam > Tamam** düğmesini tıklayın.

i) Domain Administratorsekleme için **e - h** arasındaki adımları yineleyin.

j) Windows Explorer 'da **c: \wmq > Paylaş ...**ögesini sağ tıklayın.

k) **Gelişmiş Paylaşım ...** seçeneğini tıklayın. ve **Bu klasörü paylaş** onay kutusunu seçin. Paylaşım adını *wmq*olarak bırakın.

l) **İzinler > Ekle ...**, ve *mqm* ; Domain Administratorsetki alanı yerel grubunun adını yazın. **Adları Denetle**seçeneğini tıklayın.

m) Windows Güvenlik penceresine yanıt olarak Domain Administrator ' nin adını ve parolasını yazın ve **Tamam > Tamam**düğmesini tıklayın.

3. *QMGR* kuyruk yöneticisini ilk etki alanı denetleyicisinde ( *sun*) yaratın.

```
cirtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\mars\wmq\data -ld \\mars\wmq\logs QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager created.
Directory '\\mars\wmq\data\QMGR' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QMGR'.
Default objects statistics : 74 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

4. Yedek yönetim ortamına izin veren kuyruk yöneticisini *sun*üzerinde başlatın.



```
strmqm -x QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation 'Installation1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

5. *earth* üzerinde ikinci bir *QMGR* örneği oluşturun.

- a) Prefix (Önek) ve InstallationName değiştirgelerinin değerlerinin *earth* için doğru olup olmadığını denetleyin.

*sun*' da **dspmqrinf** komutunu çalıştırın:

```
dspmqrinf QMGR
```

Sistemin yapması gereken:

```
QueueManager:  
Name=QMGR  
Directory=QMGR  
Prefix=C:\ProgramData\IBM\MQ  
DataPath=\\mars\wmq\data\QMGR  
InstallationName=Installation1
```

- b) **QueueManager** kütüphanesinin makine tarafından okunabilir biçimini panoya kopyalayın.

*sun* üzerinde, -o command değiştirgesiyle **dspmqrinf** komutunu yeniden çalıştırın.

```
dspmqrinf -o command QMGR
```

Sistemin yapması gereken:

```
addmqinf -s QueueManager -v Name=QMGR  
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\MQ"  
-v DataPath=\\mars\wmq\data\QMGR
```

- c) *earth* üzerinde, *earth* üzerinde kuyruk yöneticisinin bir eşgörmümünü yaratmak için **addmqinf** komutunu panodan çalıştırın.

Gerekliyse, Prefix (Önek) ya da InstallationName değiştirgelerindeki farklılıkları karşılamak için komutu ayarlayın.

```
addmqinf -s QueueManager -v Name= QMGR  
-v Directory= QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"  
-v DataPath=\\mars\wmq\data\QMGR
```

```
IBM MQ configuration information added.
```

6. *earth* üzerinde kuyruk yöneticisinin yedek yönetim ortamını başlatın.

```
strmqm -x QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation 'Installation1'.  
A standby instance of queue manager 'QMGR' has been started. The active  
instance is running elsewhere.
```

## Sonuçlar

Kuyruk yöneticisinin *sun* 'dan *earth* 'a geçtiğini doğrulayın:

1. *sun* 'da şu komutu çalıştırın:

```
endmqm -i -r -s QMGR
```

*sun* üzerindeki sistem yanıtı:

```
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ended, permitting switchover to  
a standby instance.
```

2. *earth* üzerinde şu komutu arka arkaya yazın:

```
dspmq
```

Sistem yanıtları:

```
QMNAME(QMGR) STATUS(Running as standby)  
QMNAME(QMGR) STATUS(Running as standby)  
QMNAME(QMGR) STATUS(Running)
```

## Sonraki adım

Çok eşgörünümlü bir kuyruk yöneticisini örnek programları kullanarak doğrulamak için bkz. [“Windows üzerinde çok eşgörünümlü kuyruk yöneticisinin doğrulanması”](#) sayfa 522.

### İlgili görevler

[“Örnek bir etki alanına ikinci bir Windows etki alanı denetleyicisi eklenmesi”](#) sayfa 518

[“IBM MQ ' in örnek bir etki alanında Windows etki alanı denetleyicilerine kurulması”](#) sayfa 520

### İlgili bilgiler

[Etki alanı denetleyicileri olarak Windows 2000, Windows Server 2003 ve Windows Server 2008 küme düğümleri](#)

**Windows** Örnek bir etki alanına ikinci bir Windows etki alanı denetleyicisi eklenmesi

Etki alanı denetleyicilerinde ve dosya sunucularında çok eşgörünümlü kuyruk yöneticilerinin çalıştırılacağı bir Windows etki alanı oluşturmak için *wmq.example.com* etki alanına ikinci bir etki alanı denetleyicisi ekleyin.

Örnek yapılandırma üç sunucudan oluşur:

## **sun**

İlk etki alanı denetleyicisi olarak kullanılan bir Windows Server 2008 sunucusu. *sun*, *earth* ve *mars* için *wmq.example.com* etki alanını tanımlar. Çok eşgörümlü kuyruk yöneticisinin *QMGRadl* bir eşgörümünü içerir.

## **earth**

İkinci etki alanı denetleyicisi IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. *QMGRadl* çok eşgörümlü kuyruk yöneticisinin ikinci eşgörümünü içerir.

## **mars**

Dosya sunucusu olarak kullanılan bir Windows Server 2008.

Örnekteki italik adları, seçtiğiniz adlarla değiştirin.

## **Başlamadan önce**

1. *wmq.example.com* etki alanı için bir etki alanı denetleyicisi ( *sun* ) oluşturmak üzere “Windows üzerinde Active Directory ve DNS etki alanı oluşturma” sayfa 503 içindeki adımları gerçekleştirin. İtalik adları yapılandırmanıza uyacak şekilde değiştirin.
2. Windows Server 2008 'i varsayılan çalışma grubundaki ( WORKGROUP ) bir sunucuya kurun. Örneğin, sunucu *earth* olarak adlandırılır.

## **Bu görev hakkında**

Bu görevde, *wmq.example.com* etki alanında ikinci bir etki alanı denetleyicisi olarak *earth* adlı bir Windows Server 2008 yapılandırırsınız.

Bu görev, kuyruk yöneticisi verilerine ve günlük dosyalarına erişimi gösteren ilgili görevler kümesinden biridir. Görevler, seçtiğiniz bir dizinde saklanan verileri ve günlük dosyalarını okuma ve yazma yetkisi olan bir kuyruk yöneticisinin nasıl yaratılacağını gösterir. Görevle birlikte, “Windows etki alanları ve çok eşgörümlü kuyruk yöneticileri” sayfa 498.

## **Yordam**

1. Etki alanı denetleyicisini ( *sun.wmq.example.com* ) DNS sunucusu olarak *earth* ' e ekleyin.
  - a) *earth* ' da *earth\Administrator* olarak oturum açın ve **Start**(Başlat) seçeneğini tıklayın.
  - b) **Ağ > Özellikler > Ağ bağlantılarını yönet** seçeneğini sağ tıklayın.
  - c) Ağ bağdaştırıcısını sağ tıklayın ve **Özellikler** seçeneğini belirleyin.

Sistem, bağlantının kullandığı öğeleri listeleyen Yerel Ağ Bağlantısı Özellikleri penceresi ile yanıt verir.
  - d) Local Area Connection Properties (Yerel Ağ Bağlantısı Özellikleri) penceresindeki öğeler listesinden **Internet Protocol Version 4** ya da **Internet Protocol IBM WebSphere MQ 6** (Internet İletişim Kuralı) seçeneğini belirleyin. **Özellikler > Gelişmiş ...** seçeneğini tıklayın. ve **DNS** sekmesini tıklayın.
  - e) DNS sunucusu adreslerinin altında **Ekle ...** düğmesini tıklayın.
  - f) DNS sunucusu olan etki alanı denetleyicisinin IP adresini yazın ve **Ekle** düğmesini tıklayın.
  - g) **Bu DNS soneklerini ekle > Ekle ...** düğmesini tıklayın.
  - h) *wmq.example.com* yazın ve **Ekle** düğmesini tıklayın.
    - i) **Bu bağlantı için DNS soneki** alanına *wmq.example.com* yazın.
    - j) **Bu bağlantının adresini DNS ' de kaydet** ve **Bu bağlantının sonekini DNS kaydında kullan** seçeneğini belirleyin. **Tamam > Tamam > Kapat** düğmesini tıklayın.
  - k) Bir komut penceresi açın ve TCP/IP ayarlarını gözden geçirmek için **ipconfig /all** komutunu yazın.
2. *sun* etki alanı denetleyicisinde yerel ya da Workgroup yöneticisi olarak oturum açın.

Sunucu önceden bir etki alanı denetleyicisi olarak yapılandırıldıysa, etki alanı yöneticisi olarak oturum açmanız gerekir.

3. Active Directory Etki Alanı Hizmetleri sihirbazını çalıştırın.

a) **Başlat > Çalıştır ...** düğmesini tıklatın. `dcpromo` yazın ve **Tamam'** i tıklatın.

Active Directory ikili dosyaları önceden kurulu değilse, Windows dosyaları otomatik olarak kurar.

4. `earth'` i `wmq.example.com` etki alanındaki ikinci etki alanı denetleyicisi olarak yapılandırın.

a) Sihirbazın ilk penceresinde, **Use advanced mode installation** (Gelişmiş kip kuruluşunu kullan) onay kutusunu temizleyin. **Next > Next** (İleri) seçeneğini tıklatın ve **Create a domain controller to a existing domain > Next** (Var olan etki alanına etki alanı denetleyicisi ekle) seçeneğini tıklatın.

b) `wmq` yazın **Bu ormandaki herhangi bir etki alanının adını yazın ...** alan. **Diğer kimlik bilgileri** radyo düğmesi tıklatıldığında **Ayarla ...** düğmesini tıklatın. Etki alanı yöneticisinin adını ve parolasını yazın ve **Tamam > İleri > İleri > İleridüğmesini** tıklatın.

c) Ek Etki Alanı Denetleyicisi Seçenekleri penceresinde, seçilen **DNS sunucusu** ve **Genel katalog** seçeneklerini kabul edin; **İleri > İleridüğmesini** tıklatın.

d) Directory Services Restore Mode Administrator Password (Dizin Hizmetleri Geri Yükleme Kipi Yönetici Parolası) altında bir **Password** (Parola) ve **Confirm password** (Parolayı onayla) yazın ve **Next > Next** (İleri) düğmesini tıklatın.

e) **Ağ Kimlik Bilgileri** istendiğinde etki alanı yöneticisinin parolasını yazın. Son sihirbaz penceresinde **Reboot on tamamlanınca yeniden yükle** seçeneğini belirleyin.

f) Bir süre sonra, DNS yetkilendirmesiyle ilgili bir **DCPromo** hatası ile bir pencere açılabilir; **Tamam'** i tıklatın. Sunucu yeniden önyüklenir.

## Sonuçlar

`earth` yeniden yüklendiğinde, Etki Alanı Yöneticisi olarak oturum açın. `wmq.example.com` etki alanının `earth'` a eşlendiğini denetleyin.

## Sonraki adım

IBM MQ kuruluşuna devam edin; bkz. [“IBM MQ ' in örnek bir etki alanında Windows etki alanı denetleyicilerine kurulması” sayfa 520.](#)

### İlgili görevler

**Windows** [IBM MQ ' in örnek bir etki alanında Windows etki alanı denetleyicilerine kurulması](#)  
“Windows üzerinde Active Directory ve DNS etki alanı oluşturma” sayfa 503

**Windows** [IBM MQ ' in örnek bir etki alanında Windows etki alanı denetleyicilerine kurulması](#)  
`wmq.example.com` etki alanındaki her iki etki alanı denetleyicisine IBM MQ kuruluşunu kurun ve yapılandırın.

Örnek yapılandırma üç sunucudan oluşur:

### **sun**

İlk etki alanı denetleyicisi olarak kullanılan bir Windows Server 2008 sunucusu. `sun`, `earth` ve `mars` çeren `wmq.example.com` etki alanını tanımlar. Çok eşgörünümlü kuyruk yöneticisinin `QMGR` adlı bir eşgörünümünü içerir.

### **earth**

İkinci etki alanı denetleyicisi IBM MQ sunucusu olarak kullanılan bir Windows Server 2008. `QMGR` adlı çok eşgörünümlü kuyruk yöneticisinin ikinci eşgörünümünü içerir.

### **mars**

Dosya sunucusu olarak kullanılan bir Windows Server 2008.

Örnekteki italik adları, seçtiğiniz adlarla değiştirin.

## Başlamadan önce

1. *wmq.example.com* etki alanı için bir etki alanı denetleyicisi ( *sun*) oluşturmak üzere “[Windows üzerinde Active Directory ve DNS etki alanı oluşturma](#)” sayfa 503 içindeki adımları gerçekleştirin. İtaliye adları yapılandırmanıza uyacak şekilde değiştirin.
2. *wmq.example.com* etki alanı için ikinci bir etki alanı denetleyicisi ( *earth*) oluşturmak üzere “[Örnek bir etki alanına ikinci bir Windows etki alanı denetleyicisi eklenmesi](#)” sayfa 518 içindeki adımları gerçekleştirin. İtaliye adları yapılandırmanıza uyacak şekilde değiştirin.
3. IBM MQ üzerinde çalıştırabileceğiniz diğer Windows sürümleri için bkz. [Windows sistemlerinde donanım ve yazılım gereksinimleri](#) .

## Bu görev hakkında

*wmq.example.com* etki alanındaki her iki etki alanı denetleyicisine IBM MQ kuruluşunu kurun ve yapılandırın.

## Yordam

1. IBM MQ ürününü *sun* ve *earth* üzerine kurun.

Daha fazla bilgi için bkz. [Installing IBM MQ server on Windows](#) .

  - a) *sun* ve *earth* sistemlerinde *wmq\Administrator* etki alanı yöneticisi olarak oturum açın.
  - b) IBM MQ for Windows kuruluş ortamında **Setup** komutunu çalıştırın.

IBM MQ Launchpad uygulaması başlar.
  - c) Önkoşul olan yazılımın kurulu olup olmadığını denetlemek için **Software Requirements** (Yazılım Gereksinimleri) seçeneğini tıklayın.
  - d) **Network Configuration > No**(Ağ Yapılandırması) seçeneğini tıklayın.

Bu kuruluş için bir etki alanı kullanıcı kimliği yapılandırabilir ya da yapılandırmayabilirsiniz. Yaratılan kullanıcı kimliği, etki alanı yerel kullanıcı kimliğidir.
  - e) **IBM MQ Installation**(Kuruluş) seçeneğini tıklayın, bir kuruluş dili seçin ve IBM MQ Installer düğmesini tıklayın.
  - f) Lisans sözleşmesini onaylayın ve varsayılan yapılandırmayı kabul etmek için **Next > Next > Install** (İleri) düğmesini tıklayın. Kuruluşun tamamlanmasını bekleyin ve **Finish**(Son) düğmesini tıklayın.

Kuruluşun adını değiştirmek, farklı bileşenler kurmak, kuyruk yöneticisi verileri ve günlükleri için farklı bir dizin yapılandırmak ya da farklı bir dizine kurmak istiyorsanız, **Tipik** yerine **Özel** öğesini tıklayın.

IBM MQ kurulur ve kuruluş programı Prepare IBM MQ Wizard' ı başlatır.

IBM MQ for Windows kuruluşu, bir etki alanı yerel grubunu *mqm* ve bir etki alanı grubunu *Domain mqm* yapılandırır. Bu, *Domain mqm* 'i *mqm* ' un bir üyesi yapar. Aynı etki alanındaki sonraki etki alanı denetleyicileri *mqm* ve *Domain mqm* gruplarını paylaşır.
2. *earth* ve *sun* sistemlerinde Prepare IBM MQ Wizard komutunu çalıştırın.

Daha fazla bilgi için bkz. [IBM MQ ürününü Prepare IBM MQ Wizard ile yapılandırma](#).

  - a) IBM MQ kuruluş programı Prepare IBM MQ Wizard ' ı otomatik olarak çalıştırır.

Sihirbazı el ile başlatmak için, **Başlat > Tüm programlar > IBM MQ** klasöründe Prepare IBM MQ Wizard kısayolunu bulun. Çoklu kuruluş yapılandırmasında IBM MQ kuruluşuna karşılık gelen kısayolu seçin.
  - b) **Next** (İleri) düğmesini tıklayın ve **No** (Hayır) seçeneğini tıklayın ve soruya yanıt olarak "Ağda bir Windows 2000 ya da daha sonraki bir etki alanı denetleyicisi olup olmadığını belirleyin" sorusunu tıklayın.<sup>1</sup>.
  - c) Sihirbazın son sayfasında, gerek duyduğunuz onay kutularını seçin ya da temizleyin ve **Son** düğmesini tıklayın.

Prepare IBM MQ Wizard , ilk etki alanı denetleyicisinde bir etki alanı yerel kullanıcısı MUSR\_MQADMIN ve ikinci etki alanı denetleyicisinde başka bir etki alanı yerel kullanıcısı MUSR\_MQADMIN1 yaratır. Sihirbaz, her bir denetleyicide IBM MQ hizmetini, hizmette oturum açan kullanıcı olarak MUSR\_MQADMIN ya da MUSR\_MQADMIN1 ile oluşturur.

3. Kuyruk yöneticisi yaratma izni olan bir kullanıcı tanımlayın.

Kullanıcının yerel olarak oturum açma ve etki alanı yerel mqm grubunun bir üyesi olma hakkı olmalıdır. Etki alanı denetleyicilerinde, etki alanı kullanıcılarının yerel olarak oturum açma hakkı yoktur, ancak denetimcilerin vardır. Varsayılan olarak, hiçbir kullanıcı bu özniteliklerin ikisine de sahip değildir. Bu görevde, etki alanı yöneticilerini etki alanı yerel mqm grubuna ekleyin.

- Server Manager > Roller > Active Directory Domain Services > wmq.example.com > Kullanıcılar'** i açın.
- Etki Alanı Yöneticileri > Gruba ekle ...** seçeneğini sağ tıklayın. ve mqm yazın; **Adları denetle > Tamam > Tamam** düğmesini tıklayın.

## Sonuçlar

1. Prepare IBM MQ Wizard ' un MUSR\_MQADMIN etki alanı kullanıcıını oluşturduğunu kontrol edin:

- Server Manager > Roller > Active Directory Domain Services > wmq.example.com > Kullanıcılar'** i açın.
- Sağ tıklatma **MUSR\_MQADMIN > Özellikler ... > Üyesi** ve bunun bir Domain users ve mqm üyesi olduğunu görün.

2. MUSR\_MQADMIN ' in hizmet olarak çalıştırma hakkına sahip olup olmadığını denetleyin:

- Tıklat **Başlat > Çalıştır ...** düğmesini tıklayın, **secpol.msc** komutunu yazın ve **OK**(Tamam) düğmesini tıklayın.
- Güvenlik Ayarları > Yerel İlkeler > Kullanıcı Hakları Atamaları'** nı açın. İlkeler listesinde **Hizmet olarak oturum aç > Özellikler** ve bkz. MUSR\_MQADMIN , hizmet olarak oturum açma hakkına sahip olarak listelenir. **Tamam'** i tıklayın.

## Sonraki adım

- Kuruluş ve yapılandırmanın doğru çalıştığını doğrulamak için “Yerel mqm grubu tarafından yetkilendirilen veri ve günlük dosyalarının okunması ve yazılması” sayfa 530 görevini yapın.
- Etki alanı denetleyicilerinde çok eşgörünümlü bir kuyruk yöneticisi yapılandırma görevini tamamlamak için “Windows etki alanı denetleyicilerinde çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 515 görevine geri dönün.

## İlgili görevler

**Windows** Örnek bir etki alanına ikinci bir Windows etki alanı denetleyicisi eklenmesi

## İlgili başvurular

Bir IBM MQ Windows Hizmeti için gerekli kullanıcı hakları

**Windows** Windows üzerinde çok eşgörünümlü kuyruk yöneticisinin doğrulanması

Çok eşgörünümlü bir kuyruk yöneticisi yapılandırma doğrulamak için **amqsgbac**, **amqspbac** ve **amqsmbac** örnek programlarını kullanın. Bu konuda, Windows Server 2003 'te çok eşgörünümlü bir kuyruk yöneticisi yapılandırmasını doğrulamak için örnek bir yapılandırma sağlanır.

Yüksek kullanılabilirlikli örnek programlar otomatik istemci yeniden bağlantısını kullanır. Bağlı kuyruk yöneticisi başarısız olduğunda, istemci aynı kuyruk yöneticisi grubundaki bir kuyruk yöneticisine yeniden bağlanmayı dener. Örneklerin ( High Availability sample programs) açıklaması, yalınlık için tek bir

<sup>1</sup> Etki alanına ilişkin kuruluşu yapılandırabilirsiniz. Etki alanı denetleyicisindeki tüm kullanıcılar ve gruplar etki alanı kapsamına sahip olduğu için, bu bir fark yaratmaz. IBM MQ ürününü etki alanında değilmiş gibi kurmak daha kolaydır.

eşgörünüm kuyruk yöneticisi kullanılarak istemcinin yeniden bağlanmasını gösterir. Çok eşgörünümlü kuyruk yöneticisi yapılanışını doğrulamak için, çok eşgörünümlü kuyruk yöneticileriyle aynı örnekleri kullanabilirsiniz.

Bu örnek, “Windows etki alanı denetleyicilerinde çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 515’inde açıklanan çok eşgörünümlü yapılandırmayı kullanır. Çok eşgörünümlü kuyruk yöneticisinin yedek yönetim ortamına geçtiğini doğrulamak için yapılandırmayı kullanın. Kuyruk yöneticisini **endmqm** komutuyla durdurun ve -s, yedek sisteme geçiş, seçeneğini kullanın. İstemci programları yeni kuyruk yöneticisi yönetim ortamına yeniden bağlanır ve kısa bir gecikmeden sonra yeni yönetim ortamıyla çalışmaya devam eder.

İstemci, Windows 7 Service Pack 1 'i çalıştıran 400 MB VMware görüntüsüne kurulum. Güvenlik nedeniyle, çok eşgörünümlü kuyruk yöneticisini çalıştıran etki alanı sunucularıyla aynı VMware anasistem ağına bağlanır. Yapılandırmayı basitleştirmek için istemci bağlantı tablosunu içeren /MQHA klasörünü paylaşıyor.

### IBM MQ Explorer komutunu kullanarak hata durumunda yedek sisteme geçişin doğrulanması

Hata durumunda yedek sisteme geçişi doğrulamak için örnek uygulamaları kullanmadan önce her bir sunucuda IBM MQ Explorer komutunu çalıştırın. **Uzak Kuyruk Yöneticisi Ekle > Doğrudan çok eşgörünümlü bir kuyruk yöneticisine bağlan** sihirbazını kullanarak her iki kuyruk yöneticisi eşgörünümünü de her gezgine ekleyin. Yedek veritabanına izin veren her iki yönetim ortamının da çalıştığından emin olun. VMware görüntüsünü etkin yönetim ortamıyla birlikte çalıştıran pencereyi kapatın, sunucuyu sanal olarak kapatın ya da etkin yönetim ortamını durdurun; böylece yedek yönetim ortamına geçiş ve yeniden bağlanabilir istemcilerin yeniden bağlanması sağlanır.



**Uyarı:** Sunucuyu kapatırsanız, bunun MQHA klasörünü barındıran sunucu olmadığından emin olun!

**Not: Yedek yönetim ortamına geçmeye izin ver** seçeneği **Kuyruk Yöneticisini Durdur** iletişim penceresinde kullanılamaz. Kuyruk yöneticisi tek bir eşgörünüm kuyruk yöneticisi olarak çalıştığından seçenek eksik. Bu işlemi **Yedek yönetim ortamına izin ver** seçeneği olmadan başlatmış olmanız gerekir. Kuyruk yöneticisini durdurma isteğiniz reddedilirse, **Ayrıntılar** penceresine bakın; çalışmakta olan bir yedek eşgörünüm yoktur.

### Örnek programları kullanarak hata durumunda yedek sisteme geçişin doğrulanması

#### Etkin örneği çalıştırmak için bir sunucu seçin

MQHA dizinini ya da dosya sistemini barındırmak için sunuculardan birini seçmiş olabilirsiniz. Etkin sunucuyu çalıştıran VMware penceresini kapatarak hata durumunda yedek sisteme geçişi test etmeyi planlıyorsanız, bunun MQHA !

#### Etkin kuyruk yöneticisi eşgörünümünü çalıştıran sunucuda

1. *ipaddr1* ve *ipaddr2* komutlarını değiştirin ve N:\hasample.tst. içinde aşağıdaki komutları kaydedin

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(' ipaddr1 (1414), ipaddr2 (1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

**Not: MCAUSER** parametresi boş bırakılarak, istemci kullanıcı kimliği sunucuya gönderilir. İstemci kullanıcı kimliği, sunucular üzerinde doğru izinlere sahip olmalıdır. Diğer bir seçenek, SVRCONN kanalındaki **MCAUSER** parametresini, sunucuda yapılandırdığınız kullanıcı kimliğine ayarlamaktır.

2. N:\ yoluyla bir komut istemi açın ve komutu çalıştırın:

```
runmqsc -m QM1 < hasample.tst
```

3. **runmqsc** komutunun çıkışını inceleyerek, dinleyicinin çalıştığını ve kuyruk yöneticisi denetimine sahip olduğunu doğrulayın.

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

Ya da TCP/IP dinleyicisinin çalıştığı ve Control = Queue Manager' in olduğu IBM MQ Explorer kullanılarak.

### İstemcide

1. Sunucudaki C:\MQHA paylaşılan dizinini istemcideki N:\ ile eşleyin.
2. N:\yoluyla bir komut istemi açın. MQCHLLIB ortam değişkenini, sunucudaki istemci kanal tanımlama çizelgesini (CCDT) gösterecek şekilde ayarlayın:

```
SET MQCHLLIB=N:\data\QM1\@ipcc
```

3. Komut isteminde şu komutları yazın:

```
start amqsgnac TARGET QM1
start amqsmnac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

**Not:** Sorularınız varsa, neden kodunun konsola yazdırılması için uygulamaları bir komut isteminde başlatın ya da N:\data\QM1\errors klasöründeki AMQERR01.LOG dosyasına bakın.

### Etkin kuyruk yöneticisi eşgörünümünü çalıştıran sunucuda

1. Aşağıdakilerden birini yapın:
  - Etkin sunucu örneğiyle VMware görüntüsünü çalıştıran pencereyi kapatın.
  - IBM MQ Explorer komutunu kullanarak etkin kuyruk yöneticisi yönetim ortamını durdurun, yedek yönetim ortamına geçmeye ve yeniden bağlanabilir istemcilere yeniden bağlanma talimatı verin.
2. Üç istemci sonunda bağlantının kesilip yeniden bağlanacağını saptar. Bu yapılandırma, sunucu penceresini kapatırsanız, üç bağlantının da yeniden kurulması yaklaşık yedi dakika sürer. Bazı bağlantılar diğerlerinden önce yeniden kurulur.

### Sonuçlar

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmnac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```



```
N:\>amqsgnac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

### **Windows** Windows üzerinde paylaşılan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğinin sağlanması

Bu konuda, genel bir alternatif güvenlik grubu kullanarak, kuyruk yöneticisi verileri ve günlük dosyaları için paylaşılan bir yerin güvenliğini nasıl sağlayabileceğiniz açıklanmaktadır. Konumu, farklı sunucularda çalışan bir kuyruk yöneticisinin farklı eşgörünümleri arasında paylaşabilirsiniz.

Genellikle, kuyruk yöneticisi verileri ve günlük dosyaları için paylaşılan bir konum ayarlamayın. IBM MQ for Windows' u kurduğunuzda, kuruluş programı o sunucuda yaratılan kuyruk yöneticileri için seçtiğiniz bir ana dizini yaratır. Yerel mqm grubu ile dizinlerin güvenliğini sağlar ve IBM MQ hizmeti için dizinlere erişmek üzere bir kullanıcı kimliği yapılandırır.

Bir güvenlik grubuyla paylaşılan bir klasörü koruduğunuzda, klasöre erişmesine izin verilen bir kullanıcının grubun kimlik bilgileri olmalıdır. Uzak dosya sunucusundaki bir klasörün, marsadlı bir sunucudaki yerel mqm grubuyla güvenli kılındığını varsayın. Kuyruk yöneticisi işlemlerini çalıştıran kullanıcının marsüzerinde yerel mqm grubunun bir üyesi olmasını sağlayın. Kullanıcı, uzak dosya sunucusundaki klasörün kimlik bilgileriyle eşleşen kimlik bilgilerine sahiptir. Kuyruk yöneticisi, bu kimlik bilgilerini kullanarak verilerine erişebilir ve klasördeki dosyaları günlüğe kaydeder. Kuyruk yöneticisi işlemlerini farklı bir sunucuda çalıştıran kullanıcı, eşleşen kimlik bilgileri olmayan farklı bir yerel mqm grubunun üyesi. Kuyruk yöneticisi mars için farklı bir sunucuda çalıştığında, marsüzerinde çalıştığında yarattığı verilere ve günlük dosyalarına erişemez. Kullanıcıyı bir etki alanı kullanıcısı yapsanız bile, marsüzerindeki yerel mqm grubundan kimlik bilgilerini edinmesi gerektiğinden ve bunu farklı bir sunucudan yapamayacağından farklı kimlik bilgilerine sahiptir.

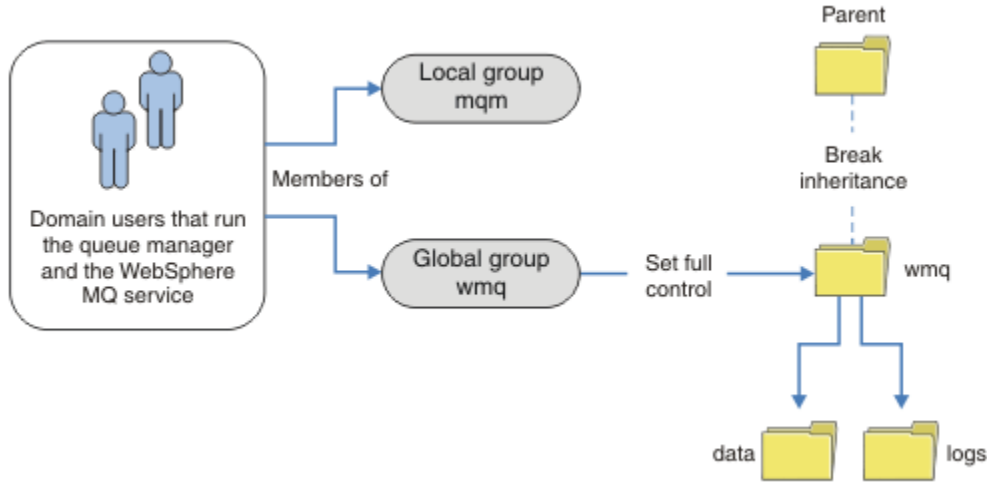
Kuyruk yöneticisine genel bir alternatif güvenlik grubu sağlanması sorunu çözer; bkz. [Şekil 73 sayfa 526](#). Genel bir grupta uzak bir klasörü güvenli hale alın. marsüzerinde yaratırken genel grubun adını kuyruk yöneticisine geçirin. **crtmqm** komutunda -a [ ı ] parametresini kullanarak, genel grup adını alternatif güvenlik grubu olarak geçirin. Kuyruk yöneticisini farklı bir sunucuda çalışacak şekilde aktarıyorsanız, güvenlik grubunun adı da bu grupta birlikte aktarılır. Ad, qm. ini dosyasındaki **AccessMode** kısmına SecurityGroup olarak aktarılır; örneğin:

```
AccessMode:
SecurityGroup=wmq\wmq
```

qm. ini içindeki **AccessMode** kısmı, RemoveMQMAccess ürününü de içerir; örneğin:

```
AccessMode:
RemoveMQMAccess=true/false
```

Bu öznitelik truedeğeriyle belirtilirse ve bir erişim grubu da verildiyse, yerel mqm grubuna kuyruk yöneticisi veri dosyalarına erişim izni verilmez.

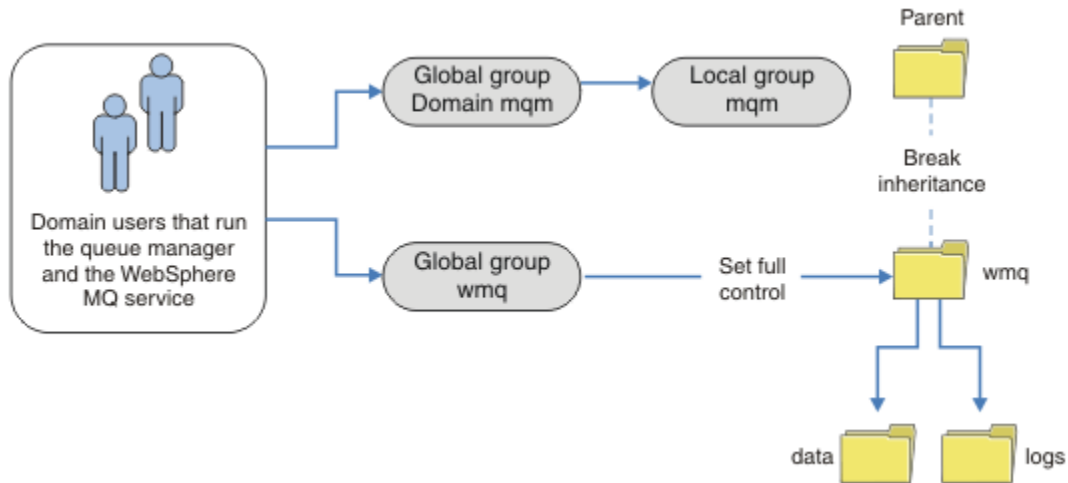


Şekil 73. Diğer bir genel güvenlik grubu kullanılarak kuyruk yöneticisi verilerinin ve günlüklerinin güvenliğinin sağlanması (1)

Kuyruk yöneticisi işlemlerinin çalıştırılacağı kullanıcı kimliğinin genel güvenlik grubunun eşleşen kimlik bilgilerine sahip olması için, kullanıcı kimliğinin de genel kapsamı olmalıdır. Yerel bir grubu ya da birincil kullanıcıyı genel bir grubun üyesi yapamazsınız. Şekil 73 sayfa 526’inde, kuyruk yöneticisi işlemlerini çalıştıran kullanıcılar etki alanı kullanıcıları olarak gösterilir.

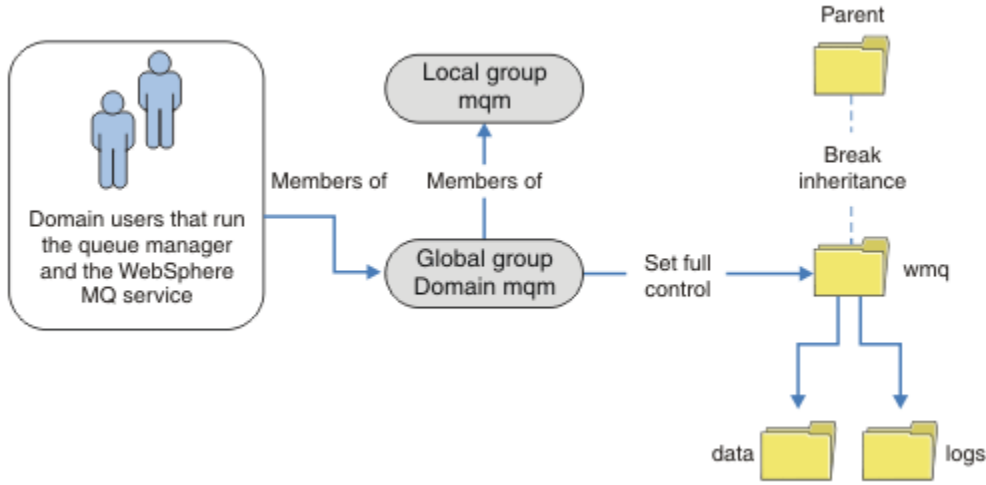
Birçok IBM MQ sunucusunu dağıtıyorsanız, Şekil 73 sayfa 526 içindeki kullanıcıların gruplanması uygun değildir. Her IBM MQ sunucusu için yerel gruplara kullanıcı ekleme işlemini yinelemeniz gerekir. Bunun yerine, etki alanı denetleyicisinde bir Domain mqm genel grubu oluşturun ve Domain mqm grubunun IBM MQ üyelerini çalıştıran kullanıcıları oluşturun; bkz. Şekil 74 sayfa 526. IBM MQ ürününü bir etki alanı kuruluşu olarak kurduğunuzda, Prepare IBM MQ Wizard Domain mqm grubunu otomatik olarak yerel mqm grubunun üyesi yapar. Aynı kullanıcılar hem Domain mqm hem de wmq genel gruplarında yer almakta.

**İpucu:** Aynı kullanıcılar IBM MQ 'ı farklı sunucularda çalıştırabilir, ancak tek bir sunucuda IBM MQ 'yi hizmet olarak çalıştırmak ve etkileşimli olarak çalıştırmak için farklı kullanıcılara sahip olmanız gerekir. Ayrıca, bir sunucudaki her kuruluş için farklı kullanıcılara sahip olmanız gerekir. Bu nedenle genellikle Domain mqm birçok kullanıcı içerir.



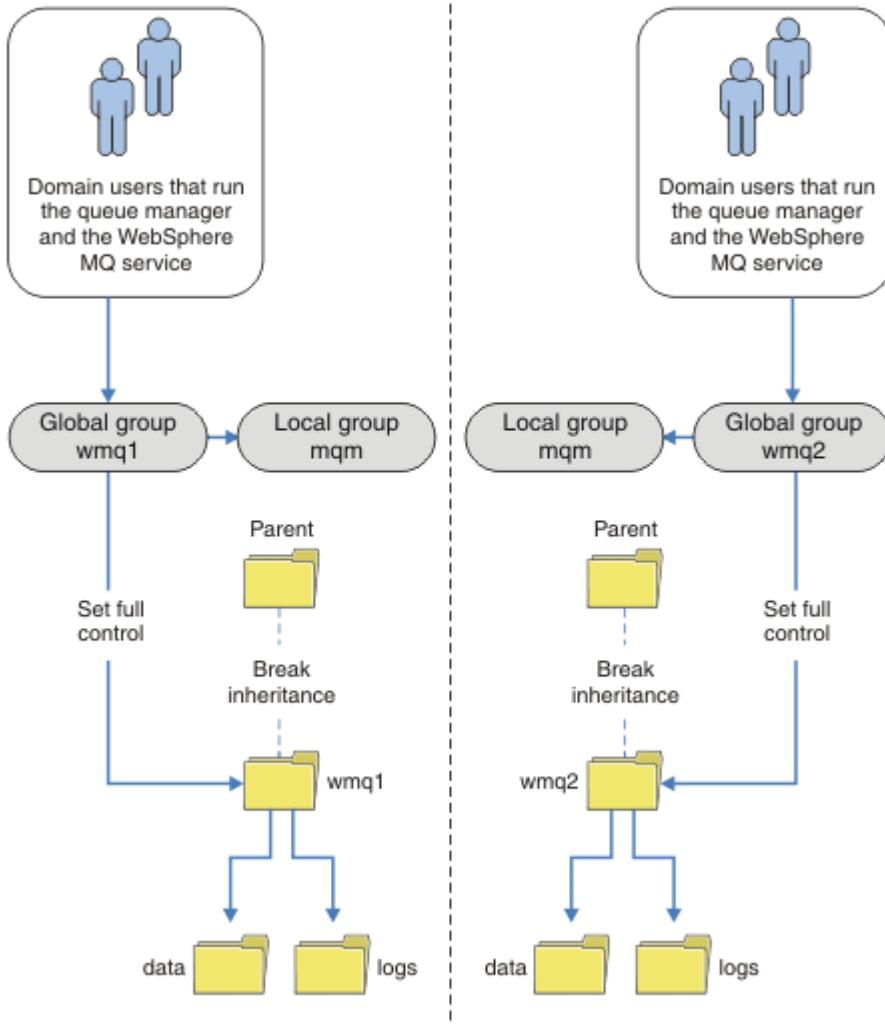
Şekil 74. Diğer bir genel güvenlik grubu kullanılarak kuyruk yöneticisi verilerinin ve günlüklerinin güvenliğinin sağlanması (2)

Şekil 74 sayfa 526 içindeki kuruluş, olduğu gibi gereksiz yere karmaşıktır. Düzenleme, aynı üyelere sahip iki küresel gruba sahiptir. Kuruluşu basitleştirebilir ve yalnızca bir genel grup tanımlayabilirsiniz; bkz. Şekil 75 sayfa 527.



Şekil 75. Diğer bir genel güvenlik grubu kullanılarak kuyruk yöneticisi verilerinin ve günlüklerinin güvenliğinin sağlanması (3)

Diğer bir seçenek olarak, farklı klasörlere erişebilmek için farklı kuyruk yöneticilerine sahip daha iyi bir erişim denetimine gereksinim duyabilirsiniz; bkz. Şekil 76 sayfa 528. Şekil 76 sayfa 528 içinde, farklı kuyruk yöneticisi günlüğü ve veri dosyalarının güvenliğini sağlamak için ayrı genel gruplar halinde iki etki alanı kullanıcı grubu tanımlanır. Farklı IBM MQ sunucularında olması gereken iki farklı yerel mqm grubu gösterilir. Bu örnekte, kuyruk yöneticileri iki kümeye ayrılır ve farklı kullanıcılar iki kümeye ayrılır. İki küme test ve üretim kuyruğu yöneticileri olabilir. Diğer güvenlik grupları wmq1 ve wmq2 olarak adlandırılır. Genel grupları wmq1 ve wmq2, test ya da üretim bölümünde olup olmadıklarına göre doğru kuyruk yöneticilerine el ile eklemelisiniz. İki kullanıcı grubu olduğundan, yapılandırma, IBM MQ kuruluşunun Domain mqm yerel mqm grubuna Şekil 75 sayfa 527 içinde olduğu gibi yayılmasından yararlanamaz.



Şekil 76. Diğer bir genel güvenlik birincil kullanıcısı kullanılarak kuyruk yöneticisi verilerinin ve günlüklerinin güvenliğinin sağlanması (4)

İki bölümü bölümlere ayırmanın alternatif bir yolu onları iki pencere etki alanına yerleştirmektir. Bu durumda, Şekil 75 sayfa 527’inde gösterilen daha basit modeli kullanmaya geri dönebilirsiniz.

**Windows** Windows üzerinde paylaşılmayan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğini sağlayın

Bu konuda, hem yerel mqm grubunu, hem de diğer bir güvenlik grubunu kullanarak kuyruk yöneticisi verileri ve günlük dosyaları için alternatif bir konumu nasıl koruyabileceğiniz açıklanmaktadır.

Genellikle, kuyruk yöneticisi verileri ve günlük dosyaları için alternatif bir konum ayarlamayın. IBM MQ for Windows' u kurduğunuzda, kuruluş programı, yaratılan kuyruk yöneticileri için seçtiğiniz bir ana dizini yaratır. Yerel mqm grubu ile dizinlerin güvenliğini sağlar ve IBM MQ hizmeti için dizinlere erişmek üzere bir kullanıcı kimliği yapılandırır.

İki örnek, IBM MQ için erişim denetiminin nasıl yapılandırılacağını gösterir. Bu örnekler, verileri ve kuruluş tarafından yaratılan veri ve günlük yollarında olmayan dizinlerdeki günlüklerle bir kuyruk yöneticisinin nasıl yaratılacağını gösterir. İlk örnekte ( “Yerel mqm grubu tarafından yetkilendirilen veri ve günlük dosyalarının okunması ve yazılması” sayfa 530), yerel mqm grubu tarafından yetkilendirilerek kuyruk ve günlük dizinlerine erişime izin verirsiniz. İkinci örnek olan “Alternatif bir yerel güvenlik grubu tarafından yetkilendirilen veri ve günlük dosyalarının okunması ve yazılması” sayfa 533, dizinlere erişimin alternatif bir güvenlik grubu tarafından yetkilendirildiğini gösterir. Dizinlere yalnızca bir sunucuda çalışan bir kuyruk yöneticisi tarafından erişildiğinde, verilerin ve günlük dosyalarının diğer güvenlik grubuyla güvenli

kılınması, farklı yerel gruplarla ya da asıl adlarla farklı kuyruk yöneticilerinin güvenliğini sağlama seçeneği sunar. Dizinlere, çok eşgözümlü kuyruk yöneticisi gibi farklı sunucularda çalışan bir kuyruk yöneticisi tarafından erişildiğinde, verilerin ve günlük dosyalarının diğer güvenlik grubuyla güvenliğinin sağlanması tek seçenektir; bkz. [“Windows üzerinde paylaşılan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğinin sağlanması” sayfa 525.](#)

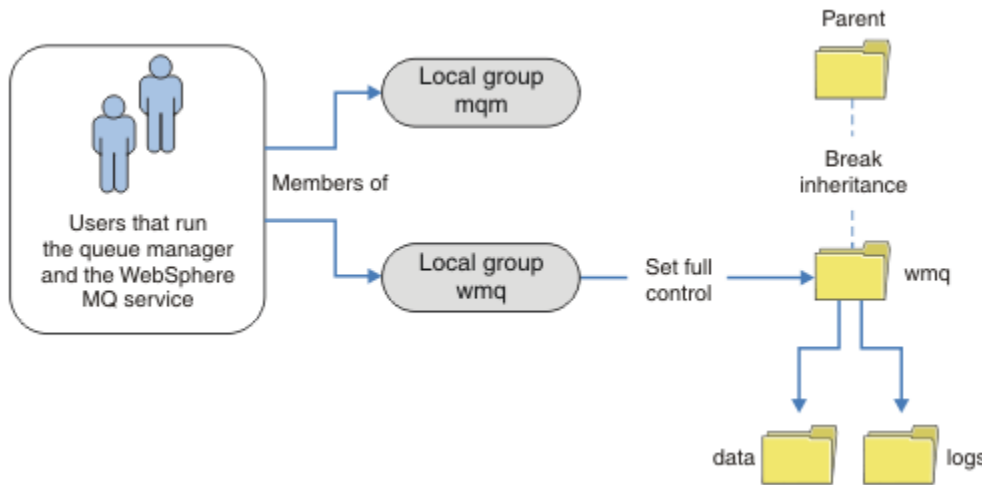
Kuyruk yöneticisi verilerinin ve günlük dosyalarının güvenlik izinlerinin yapılandırılması, Windows üzerinde sık kullanılan bir görev değildir. IBM MQ for Windows' u kurduğunuzda, kuyruk yöneticisi verileri ve günlükleri için izinleri belirtebilir ya da varsayılan izinleri kabul edebilirsiniz. Kuruluş programı, bu izinleri yerel mqm grubuyla otomatik olarak güvenli bir şekilde kullanarak tam denetim izni verir. Kuruluş işlemi, kuyruk yöneticilerini çalıştıran kullanıcı kimliğinin yerel mqm grubunun bir üyesi olduğundan emin olur. Erişim gereksinimlerinizi karşılamak için izinlerdeki diğer erişim izinlerini değiştirebilirsiniz.

Veri ve günlük dosyaları dizinini yeni konumlara taşırsanız, yeni konumların güvenliğini yapılandırmanız gerekir. Bir kuyruk yöneticisini yedekleyip farklı bir bilgisayara geri yüklerseniz ya da kuyruk yöneticisini çok eşgözümlü bir kuyruk yöneticisi olarak değiştirirseniz, izinlerin yerini değiştirebilirsiniz. Kuyruk yöneticisi verilerini ve günlük dizinlerini yeni yerlerinde güvenceye almanın iki yolu vardır. Yerel mqm grubuna erişimi sınırlayarak izinleri koruyabilir ya da istediğiniz herhangi bir güvenlik grubuna erişimi kısıtlayabilirsiniz.

Yerel mqm grubunu kullanarak izinlerin güvenliğini sağlamak için en az sayıda adım gerekir. Yerel mqm grubunun tam denetimine izin vermek için veri ve günlük dizinlerine ilişkin izinleri ayarlayın. Tipik bir yaklaşım, üst öğeden kalıtımı kaldırarak var olan izin kümesini kopyalamaktır. Daha sonra diğer birincil kullanıcı izinlerini kaldırabilir ya da kısıtlayabilirsiniz.

Kuyruk yöneticisini, Hazırla IBM MQ sihirbazı tarafından ayarlanan hizmet için farklı bir kullanıcı kimliği altında çalıştırırsanız, o kullanıcı kimliği yerel mqm grubunun bir üyesi olmalıdır. [“Yerel mqm grubu tarafından yetkilendirilen veri ve günlük dosyalarının okunması ve yazılması” sayfa 530](#) görevi, adımları boyunca size yol gösterir.

Diğer bir güvenlik grubunu kullanarak kuyruk yöneticisi verilerini ve günlük dosyalarını da güvenli hale alabilirsiniz. Kuyruk yöneticisi verilerinin ve günlük dosyalarının diğer güvenlik grubuyla güvenli kılınması işleminde, [Şekil 77 sayfa 529](#)' e gönderme yapan birkaç adım vardır. Yerel grup (wmq), alternatif bir güvenlik grubu örneğidir.



Şekil 77. Alternatif bir yerel güvenlik grubu kullanarak kuyruk yöneticisi verilerinin ve günlüklerinin güvenliğini sağlama, wmq

1. Kuyruk yöneticisi verileri ve günlükleri, ortak bir izin ya da ortak bir üst izin için ayrı izinler yaratın.
2. Dizinler ya da üst izin için var olan devralınan izinler kümesini kopyalayın ve bunları gereksinimlerinize göre değiştirin.

3. Diğer gruba ( wmq) dizinler için tam denetim izni vererek, kuyruk yöneticisini ve günlükleri içerecek dizinlerin güvenliğini sağlayın.
4. Kuyruk yöneticisi işlemlerini çalıştıran tüm kullanıcı kimliklerine diğer güvenlik grubunun ya da birincil kullanıcının kimlik bilgilerini verin:
  - a. Bir kullanıcıyı diğer güvenlik birincil kullanıcısı olarak tanımlarsanız, kullanıcı kuyruk yöneticisinin altında çalışacağı kullanıcı ile aynı olmalıdır. Kullanıcı, yerel mqm grubunun bir üyesi olmalıdır.
  - b. Alternatif güvenlik grubu olarak yerel bir grup tanımlarsanız, kuyruk yöneticisinin altında çalışacağı kullanıcıyı diğer gruba ekleyin. Kullanıcı, yerel mqm grubunun bir üyesi de olmalıdır.
  - c. Alternatif güvenlik grubu olarak bir genel grup tanımlarsanız, bkz. [“Windows üzerinde paylaşılan kuyruk yöneticisi verilerinin ve günlük dizinlerinin ve dosyalarının güvenliğinin sağlanması” sayfa 525.](#)
5. -a deęiřtirgesiyle, **crtmqm** komutunda diğer güvenlik grubunu ya da birincil kullanıcıyı belirterek kuyruk yöneticisini yaratın.

**Windows** Yerel mqm grubu tarafından yetkilendirilen veri ve günlük dosyalarının okunması ve yazılması  
Bu görev, seçtiğiniz herhangi bir dizinde saklanan veri ve günlük dosyalarıyla bir kuyruk yöneticisinin nasıl yaratılacağını gösterir. Dosyalara erişim, yerel mqm grubu tarafından korunmaktadır. Dizin paylaşılmıyor.

## Başlamadan önce

1. Birincil kuruluş olarak IBM MQ for Windows ürününü kurun.
2. Prepare IBM MQ Wizardkomutunu çalıştırın.

Daha fazla bilgi için bkz. [IBM MQ ürününü Prepare IBM MQ Wizardile yapılandırma.](#)

Bu görev için, kuruluşu yerel bir kullanıcı kimliği ya da bir etki alanı kullanıcı kimliği ile çalışacak şekilde yapılandırın. Sonuç olarak, [“Windows etki alanları ve çok eşgörünümlü kuyruk yöneticileri” sayfa 498](#) içindeki tüm görevleri tamamlamak için kuruluşun bir etki alanı için yapılandırılması gerekir.

3. Görevin ilk kısmını gerçekleştirmek için Yönetici yetkisiyle oturum açın.

## Bu görev hakkında

Bu görev, kuyruk yöneticisi verilerine ve günlük dosyalarına erişimi gösteren ilgili görevler kümesinden biridir. Görevler, seçtiğiniz bir dizinde saklanan verileri ve günlük dosyalarını okuma ve yazma yetkisi olan bir kuyruk yöneticisinin nasıl yaratılacağını gösterir. Görevle birlikte, [“Windows etki alanları ve çok eşgörünümlü kuyruk yöneticileri” sayfa 498.](#)

Windowsüzerinde, seçtiğiniz herhangi bir dizinde bir IBM MQ for Windows için varsayılan veri ve günlük yolları oluşturabilirsiniz. Kuruluş ve yapılanış sihirbazı otomatik olarak yerel mqm grubuna ve kuyruk yöneticisi işlemlerini çalıştıran kullanıcı kimliğine dizinlere erişim verir. Kuyruk yöneticisi verileri ve günlük dosyaları için farklı izinler belirten bir kuyruk yöneticisi yaratırsanız, izinler için tam denetim izni yapılandırmanız gerekir.

Bu örnekte, yerel mqm grubuna c : \wmq dizini için izin vererek kuyruk yöneticisine verileri ve günlük dosyaları üzerinde tam denetim verirsiniz.

**crtmqm** komutu, iş istasyonu IBM MQ hizmetini kullanmaya başladığında otomatik olarak başlayan bir kuyruk yöneticisi yaratır.

Görev açıklayıcı; deęiřtirebileceğiniz belirli deęerleri kullanır. Deęiřtirebileceğiniz deęerler italiktir. Görevin sonunda, yaptığınız tüm deęiřiklikleri kaldırmak için yönergeleri izleyin.

## Yordam

1. Bir komut istemi açın.
2. Komutu yazın:

```
md c:\wmq\data, c:\wmq\logs
```

3. Yerel mqm grubu okuma ve yazma erişimine izin vermek için dizinlere ilişkin izinleri ayarlayın.

```
cacls c:\wmq/T /E /G mqm:F
```

Sistemin yapması gereken:

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

4. İsteğe bağlı: Yerel mqm grubunun üyesi olan bir kullanıcı kimliğine geçin.

Yönetici olarak devam edebilirsiniz, ancak gerçekçi bir üretim yapılandırması için daha sınırlı haklara sahip bir kullanıcı kimliğiyle devam edin. Kullanıcı kimliği en azından yerel mqm grubunun bir üyesi olmalıdır.

IBM MQ kuruluşu bir etki alanının parçası olarak yapılandırıldıysa, kullanıcı kimliğini Domain mqm grubunun üyesi yapın. "Hazırla IBM MQ" sihirbazı Domain mqm genel grubunu yerel mqm grubunun üyesi yapar; bu nedenle, kullanıcı kimliğini doğrudan yerel mqm grubunun üyesi yapmanıza gerek yoktur.

5. Kuyruk yöneticisini yaratın.

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager created.
Directory 'c:\wmq\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
Completing setup.
Setup completed.
```

6. Kuyruk yöneticisi tarafından yaratılan dizinlerin c:\wmq dizininde olup olmadığını denetleyin.

```
dir c:\wmq/D /B /S
```

7. Dosyaların yerel mqm grubu için okuma ve yazma ya da tam denetim iznine sahip olup olmadığını denetleyin.

```
cacls c:\wmq\*.*
```

## Sonraki adım

Bir iletiyi kuyruğa koyup alarak kuyruk yöneticisini sınavın.

1. Kuyruk yöneticisini başlatın.

```
strmqm QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' starting.
```

The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
IBM MQ queue manager 'QMGR' started using V7.1.0.0.

2. Bir test kuyruğu yaratın.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

Sistemin yapması gereken:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: IBM MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. **amqsput**örnek programını kullanarak bir sına mesajı koyun.

```
echo 'A test message' | amqsput QTEST QMGR
```

Sistemin yapması gereken:

```
Sample AMQSPUT0 start  
target queue is QTEST  
Sample AMQSPUT0 end
```

4. **amqsget**örnek programını kullanarak sına mesajını alın.

```
amqsget QTEST QMGR
```

Sistemin yapması gereken:

```
Sample AMQSGET0 start  
message A test message  
Wait 15 seconds ...  
no more messages  
Sample AMQSGET0 end
```

5. Kuyruk yöneticisini durdurun.

```
endmqm -i QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ended.
```

6. Kuyruk yöneticisini silin.



```
dltmqm QMGR
```

Sistemin yapması gereken:

IBM MQ queue manager 'QMGR' deleted.

7. Oluşturduğunuz dizinleri silin.

**İpucu:** Komut isteminin her bir dosyayı ya da dizini silmesini önlemek için komutlara /Q seçeneğini ekleyin.

```
del /F /S C:\wmq\*. *  
rmdir /S C:\wmq
```

### İlgili kavramlar

[“Windows etki alanları ve çok eşgörünümlü kuyruk yöneticileri” sayfa 498](#)

Windows üzerinde çok eşgörünümlü bir kuyruk yöneticisi, verilerinin ve günlüklerinin paylaşılmasını gerektirir. Paylaşım, farklı sunucularda ya da iş istasyonlarında çalışan kuyruk yöneticisinin tüm eşgörünümleri için erişilebilir olmalıdır. Kuyruk yöneticilerini yapılandırın ve bir Windows etki alanının parçası olarak paylaşın. Kuyruk yöneticisi bir etki alanı iş istasyonunda ya da sunucusunda ya da etki alanı denetleyicisinde çalışabilir.

### İlgili görevler

**Windows** [Alternatif bir yerel güvenlik grubu tarafından yetkilendirilen veri ve günlük dosyalarının okunması ve yazılması](#)

Bu görev, **crtmqm** komutunda -a işaretinin nasıl kullanılacağını gösterir. Bu işaret, kuyruk yöneticisine günlük ve veri dosyalarına erişim vermek için alternatif bir yerel güvenlik grubu sağlar.

[“Alternatif bir genel güvenlik grubu tarafından yetkilendirilen paylaşılan verileri ve günlük dosyalarını okuma ve yazma” sayfa 512](#)

[“Windows üzerinde etki alanı iş istasyonlarında ya da sunucularda çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 499](#)

**Windows** [Alternatif bir yerel güvenlik grubu tarafından yetkilendirilen veri ve günlük dosyalarının okunması ve yazılması](#)

Bu görev, **crtmqm** komutunda -a işaretinin nasıl kullanılacağını gösterir. Bu işaret, kuyruk yöneticisine günlük ve veri dosyalarına erişim vermek için alternatif bir yerel güvenlik grubu sağlar.

### Başlamadan önce

1. Birincil kuruluş olarak IBM MQ for Windows ürününü kurun.
2. Prepare IBM MQ Wizardkomutunu çalıştırın.

Daha fazla bilgi için bkz. [IBM MQ ürününü Prepare IBM MQ Wizardile yapılandırma](#).

Bu görev için, kuruluşu yerel bir kullanıcı kimliği ya da bir etki alanı kullanıcı kimliği ile çalışacak şekilde yapılandırın. Sonuç olarak, [“Windows etki alanları ve çok eşgörünümlü kuyruk yöneticileri” sayfa 498](#) içindeki tüm görevleri tamamlamak için kuruluşun bir etki alanı için yapılandırılması gerekir.

3. Görevin ilk kısmını gerçekleştirmek için Yönetici yetkisiyle oturum açın.

### Bu görev hakkında

Bu görev, kuyruk yöneticisi verilerine ve günlük dosyalarına erişimi gösteren ilgili görevler kümesinden biridir. Görevler, seçtiğiniz bir dizinde saklanan verileri ve günlük dosyalarını okuma ve yazma yetkisi olan bir kuyruk yöneticisinin nasıl yaratılacağını gösterir. Görevle birlikte, [“Windows etki alanları ve çok eşgörünümlü kuyruk yöneticileri” sayfa 498](#).

Windows üzerinde, seçtiğiniz herhangi bir dizinde bir IBM MQ for Windows için varsayılan veri ve günlük yolları oluşturabilirsiniz. Kuruluş ve yapılandırma sihirbazı otomatik olarak yerel mqm grubuna ve kuyruk yöneticisi işlemlerini çalıştıran kullanıcı kimliğine dizinlere erişim verir. Kuyruk yöneticisi verileri ve günlük dosyaları için farklı dizinler belirten bir kuyruk yöneticisi yaratırsanız, dizinler için tam denetim izni yapılandırmanız gerekir.

Bu örnekte, kuyruk yöneticisine dizinler üzerinde tam denetim yetkisi olan alternatif bir güvenlik yerel grubu sağlarsınız. Diğer güvenlik grubu, kuyruk yöneticisine dizindeki dosyaları yönetme izni verir. Alternatif güvenlik grubunun birincil amacı, alternatif bir güvenlik genel grubuna yetki vermektir. Çok eşgörünümlü bir kuyruk yöneticisi ayarlamak için diğer bir güvenlik genel grubunu kullanın. Bu örnekte, yerel bir grubun konfigürasyonunu, bir etki alanına IBM MQ kurmadan diğer bir güvenlik grubu kullanımını tanımanız için tanımladınız. Yerel bir grubu alternatif bir güvenlik grubu olarak yapılandırmak alışılmadık bir durumdur.

**crtmqm** komutu, iş istasyonu IBM MQ hizmetini kullanmaya başladığında otomatik olarak başlayan bir kuyruk yöneticisi yaratır.

Görev açıklayıcı; değiştirebileceğiniz belirli değerleri kullanır. Değiştirebileceğiniz değerler italiktir. Görevin sonunda, yaptığınız tüm değişiklikleri kaldırmak için yönergeleri izleyin.

## Yordam

### 1. Alternatif bir güvenlik grubu ayarlayın.

Alternatif güvenlik grubu genellikle bir etki alanı grubudur. Örnekte, yerel bir diğer güvenlik grubunu kullanan bir kuyruk yöneticisi yaratırsınız. Yerel bir diğer güvenlik grubuyla, görevi bir etki alanının parçası olmayan bir IBM MQ kuruluşuyla yapabilirsiniz.

- a) Yerel Kullanıcılar ve Gruplar penceresini açmak için **lusrmgr.msc** komutunu çalıştırın.
- b) **Gruplar > Yeni Grup ...** öğesini sağ tıklayın.
- c) **Grup adı** alanına *altrmqm* yazın ve **Oluştur > Kapat** seçeneğini tıklayın.
- d) IBM MQ hizmetini çalıştıran kullanıcı kimliğini tanımlayın.
  - i) **Başlat > Çalıştır ...** düğmesini tıklayın, *services.msc* yazın ve **Tamam** düğmesini tıklayın.
  - ii) Hizmetler listesinde IBM MQ hizmetini tıklayın ve Oturum Açma sekmesini tıklayın.
  - iii) Kullanıcı kimliğini hatırlayın ve Hizmet Gezgini 'ni kapatın.
- e) IBM MQ hizmetini çalıştıran kullanıcı kimliğini *altrmqm* grubuna ekleyin. Ayrıca, bir kuyruk yöneticisi yaratmak için oturum açtığınız kullanıcı kimliğini ekleyin ve etkileşimli olarak çalıştırın.

Windows , kuyruk yöneticisi işlemlerini çalıştıran kullanıcı kimliğinin yetkisini denetleyerek, veri ve günlük dizinlerine erişmek için kuyruk yöneticisinin yetkisini denetler. Kullanıcı kimliği, doğrudan ya da dolaylı olarak dizinlere yetki veren *altrmqm* grubunun bir üyesi olmalıdır.

IBM MQ ürününü bir etki alanının parçası olarak kurduysanız ve [“Windows üzerinde etki alanı iş istasyonlarında ya da sunucularda çok eşgörünümlü kuyruk yöneticisi yaratılması” sayfa 499](#) içindeki görevleri gerçekleştirecekseniz, [“Windows üzerinde Active Directory ve DNS etki alanı oluşturma” sayfa 503](#) içinde oluşturulan etki alanı kullanıcı kimlikleri *wmquser1* ve *wmquser2* olur.

Kuyruk yöneticisini bir etki alanının parçası olarak kurmadıysanız, IBM MQ hizmetini çalıştıran varsayılan yerel kullanıcı kimliği *MUSR\_MQADMIN* olur. Görevleri Yönetici yetkisi olmadan gerçekleştirmek istiyorsanız, yerel mqm grubunun üyesi olan bir kullanıcı yaratın.

*altrmqm* e *wmquser1* ve *wmquser2* eklemek için aşağıdaki adımları izleyin. Yapılandırmanız farklıysa, adlarınızı kullanıcı kimlikleriyle ve grupta değiştirin.

- i) Grup listesinde **altrmqm > Özellikler > Ekle ...** seçeneğini sağ tıklayın.
- ii) Select Users, Computers ya da Groups (Kullanıcı, Bilgisayar ya da Grup Seç) penceresinde *wmquser1 ; wmquser2* yazın ve **Check Names** (Adları Denetle) seçeneğini tıklayın.
- iii) Windows Güvenlik penceresinde bir etki alanı yöneticisinin adını ve parolasını yazın ve **Tamam > Tamam > Uygula > Tamam** düğmesini tıklayın.

2. Bir komut istemi açın.
3. IBM MQ hizmetini yeniden başlatın.

Hizmetin altında çalıştığı kullanıcı kimliğinin, kendisi için yapılandığınız ek güvenlik kimlik bilgilerini edinmesi için hizmeti yeniden başlatmanız gerekir.

Komutları yazın:

```
endmqsvc  
strmqsvc
```

Sistem yanıtları:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

Ve:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' started successfully.
```

4. Komutu yazın:

```
md c:\wmq\data, c:\wmq\logs
```

5. *user* yerel kullanıcısının okuma ve yazma erişimine izin vermek için dizinlerdeki izinleri ayarlayın.

```
cacls c:\wmq/T /E /G a\mqm:F
```

Sistemin yapması gereken:

```
processed dir: c:\wmq  
processed dir: c:\wmq\data  
processed dir: c:\wmq\logs
```

6. İsteğe bağlı: Yerel mqm grubunun üyesi olan bir kullanıcı kimliğine geçin.

Yönetici olarak devam edebilirsiniz, ancak gerçekçi bir üretim yapılandırması için daha sınırlı haklara sahip bir kullanıcı kimliğiyle devam edin. Kullanıcı kimliği en azından yerel mqm grubunun bir üyesi olmalıdır.

IBM MQ kuruluşu bir etki alanının parçası olarak yapılandırıldıysa, kullanıcı kimliğini Domain mqm grubunun üyesi yapın. "Hazırla IBM MQ" sihirbazı Domain mqm genel grubunu yerel mqm grubunun üyesi yapar; bu nedenle, kullanıcı kimliğini doğrudan yerel mqm grubunun üyesi yapmanıza gerek yoktur.

7. Kuyruk yöneticisini yaratın.

```
crtmqm -a a\mqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager created.  
Directory 'c:\wmq1\data\QMGR' created.  
The queue manager is associated with installation '1'  
Creating or replacing default objects for queue manager 'QMGR'  
Default objects statistics : 74 created. 0 replaced.  
Completing setup.  
Setup completed.
```

8. Kuyruk yöneticisi tarafından yaratılan dizinlerin `c:\wmq` dizininde olup olmadığını denetleyin.

```
dir c:\wmq/D /B /S
```

9. Dosyaların yerel mqm grubu için okuma ve yazma ya da tam denetim iznine sahip olup olmadığını denetleyin.

```
cacls c:\wmq\*.*
```

## Sonraki adım

Bir iletiyi kuyruğa koyup alarak kuyruk yöneticisini sınavın.

1. Kuyruk yöneticisini başlatın.

```
strmqm QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. Bir test kuyruğu yaratın.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

Sistemin yapması gereken:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: IBM MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. **amqspu**törnek programını kullanarak bir sına ma iletisi koyun.

```
echo 'A test message' | amqspu QTEST QMGR
```

Sistemin yapması gereken:

```
Sample AMQSPUT0 start  
target queue is QTEST  
Sample AMQSPUT0 end
```

4. **amqsge**törnek programını kullanarak sına ma iletisini alın.

```
amqsge QTEST QMGR
```

Sistemin yapması gereken:

```
Sample AMQSGET0 start  
message A test message  
Wait 15 seconds ...  
no more messages  
Sample AMQSGET0 end
```

5. Kuyruk yöneticisini durdurun.

```
endmqm -i QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ended.
```

6. Kuyruk yöneticisini silin.

```
dltmqm QMGR
```

Sistemin yapması gereken:

```
IBM MQ queue manager 'QMGR' deleted.
```

7. Oluşturduğunuz dizinleri silin.

**İpucu:** Komut isteminin her bir dosyayı ya da dizini silmesini önlemek için komutlara /Q seçeneğini ekleyin.

```
del /F /S C:\wmq\*.*  
rmdir /S C:\wmq
```

## İlgili görevler

**Windows** Yerel mqm grubu tarafından yetkilendirilen veri ve günlük dosyalarının okunması ve yazılması

Bu görev, seçtiğiniz herhangi bir dizinde saklanan veri ve günlük dosyalarıyla bir kuyruk yöneticisinin nasıl yaratılacağını gösterir. Dosyalara erişim, yerel mqm grubu tarafından korunmaktadır. Dizin paylaşılmıyor.

**Linux** Linux üzerinde çok eşgörünümlü kuyruk yöneticisi yaratılması

Linux üzerinde çok eşgörünümlü bir kuyruk yöneticisinin nasıl ayarlanacağını gösteren bir örnek. Kuruluş, ilgili kavramları göstermek için küçüktür. Örnek, Linux Red Hat Enterprise 5 'e dayalıdır. Adımlar diğer UNIX platformlarında farklılık gösterir.

## Bu görev hakkında

Bu örnek, Windows 7 Service Pack 1 çalıştıran 3 GB RAM 'e sahip 2 GHz 'lik bir dizüstü bilgisayarda ayarlanır. İki VMware sanal makinesi, Server1 ve Server2, 640 MB görüntülerde Linux Red Hat Enterprise 5 'i çalıştırın. Server1 , ağ dosya sistemini (NFS), kuyruk yöneticisi günlüklerini ve bir HA örneğini barındırır. NFS sunucusunun kuyruk yöneticisi yönetim ortamlarından birini barındırmak için de olağan bir uygulama değildir; bu, örneği basitleştirmek içindir. Server2 , Server1 'in kuyruk yöneticisi günlüklerini bir yedek yönetim ortamıyla takar. WebSphere MQ MQI istemcisi, Windows 7 Service Pack 1 'i çalıştıran ve örnek yüksek kullanılabilirlikli uygulamaları çalıştıran ek 400 MB VMware görüntüsüne kurulur. Tüm sanal makineler, güvenlik nedenleriyle yalnızca VMware anasistem ağının bir parçası olarak yapılandırılır.

**Not:** NFS sunucusuna yalnızca kuyruk yöneticisi verilerini koymalısınız. NFS' de sistemi güvenli kılmak için bağlama komutuyla birlikte aşağıdaki üç seçeneği kullanın:

- **noexec**

Bu seçeneği kullanarak, uzak bir kullanıcının sistemde istenmeyen kodu çalıştırmasını önleyen NFS' de ikili dosyaların çalıştırılmasını engelleyebilirsiniz.

- **nosuid**

Bu seçeneği kullanarak, uzak bir kullanıcının daha yüksek ayrıcalıklar elde etmesini engelleyen set-user-identifier ve set-group-identifier bitlerinin kullanılmasını engelleyebilirsiniz.

- **nodev**

Bu seçeneği kullanarak, uzak bir kullanıcının bir chroot hapishanesinden çıkmasını engelleyen karakter ve özel aygıtların kullanılmasını ya da tanımlanmasını engelleyebilirsiniz.

## Yordam

1. Kök kullanıcı olarak oturum açın.
2. [Installing IBM MQ -overview](#) (Kuruluş-genel bakış) başlıklı konuyu okuyun ve IBM MQdosyasını kurmak için uygun bağlantıyı izleyin, mqm kullanıcı ve grubunu yarattın ve /var/mqmadresini tanımlayın.
3. Dosya sisteminin çok eşgözümlü kuyruk yöneticilerini destekleyip desteklemediğini denetlemek için [Paylaşılan dosya sistemi davranışının doğrulanması](#) görevini tamamlayın.
4. Server1için aşağıdaki adımı tamamlayın:
  - a. Paylaşılacak ortak bir klasörde ( /MQHA) günlük ve veri dizinleri oluşturun. Örneğin:
    - i) **mkdir** /MQHA
    - ii) **mkdir** /MQHA/logs
    - iii) **mkdir** /MQHA/qmgrs
5. Server2için aşağıdaki adımı tamamlayın:
  - a. Paylaşılan dosya sistemini bağlamak için /MQHAKlasörünü oluşturun. Yolu Server1' deki gibi tutun. Örneğin:
    - i) **mkdir** /MQHA
6. MQHA dizinlerinin kullanıcı ve grup mqm 'ye ait olduğundan ve erişim izinlerinin kullanıcı ve grup için rwx olarak ayarlandığından emin olun. Örneğin, **ls -al** , d1wx1wx1-x mqm mqm 4096 Nov 27 14:38 MQDATA değerini görüntüler.
  - a. **chown -R** mqm:mqm /MQHA
  - b. **chmod -R** ug+rwx /MQHA
7. Şu komutu girerek kuyruk yöneticisini yarattın: **crtmqm -ld /MQHA/logs -md /MQHA/qmgrs QM1**
8. Ekle<sup>2</sup>/MQHA \*(rw, sync, no\_wdelay, fsid=0) - /etc/exports
9. Server1için aşağıdaki adımları tamamlayın:
  - a. NFS yardımcı programını başlatın: /etc/init.d/ **nfs** start
  - b. Kuyruk yöneticisi yapılandırma ayrıntılarını Server1: ' den kopyalayın:

```
dspmqlnf -o command QM1
```

ve sonucu panoya kopyalayın:

```
addmqinf -s QueueManager  
-v Name=QM1  
-v Directory=QM1
```

<sup>2</sup> '\*' , bu bir bağlama/MQHA ' ya okuma/yazma için erişebilecek tüm makinelere izin verir. Bir üretim makinesine erişimi kısıtlayın.

```
-v Prefix=/var/mqm  
-v DataPath=/MQHA/qmgrs/QM1
```

10. Server2 için aşağıdaki adımları tamamlayın:

- Şu komutu girerek dışa aktarılan dosya sistemini /MQHA takın: **mount -t nfs4 -o hard,intr Server1:/ /MQHA**
- Kuyruk yöneticisi yapılanış komutunu Server2:

```
addmqinf -s QueueManager  
-v Name=QM1  
-v Directory=QM1  
-v Prefix=/var/mqm  
-v DataPath=/MQHA/qmgrs/QM1
```

11. Kuyruk yöneticisi yönetim ortamlarını sırasıyla şu-x deęiřtirgesiyle başlatın: **setmqm -x QM1**.

Kuyruk yöneticisi yönetim ortamlarını başlatmak için kullanılan komut, **addmqinf** komutuyla aynı IBM MQ kuruluşundan verilmelidir. Kuyruk yöneticisini farklı bir kuruluşta başlatmak ve durdurmak için, önce **setmqm** komutunu kullanarak kuyruk yöneticisiyle ilişkili kuruluşu ayarlamanız gerekir. Daha fazla bilgi için bkz. [setmqm](#).

**Linux** *Linux üzerinde çok eşgörunümlü kuyruk yöneticisinin doęrulanması*

Çok eşgörunümlü bir kuyruk yöneticisi yapılanışını doęrulamak için **amqsgbac**, **amqspbac** ve **amqsmbac** örnek programlarını kullanın. Bu konuda, Linux Red Hat Enterprise 5 üzerinde çok eşgörunümlü bir kuyruk yöneticisi yapılandırmasını doęrulamak için örnek bir yapılandırma sağlanmaktadır.

Yüksek kullanılabilirlikli örnek programlar otomatik istemci yeniden bağlantısını kullanır. Bağlı kuyruk yöneticisi başarısız olduęunda, istemci aynı kuyruk yöneticisi grubundaki bir kuyruk yöneticisine yeniden bağlanmayı dener. Örneklerin ( High Availability sample programs) açıklaması, yalınlık için tek bir eşgörunüm kuyruk yöneticisi kullanılarak istemcinin yeniden bağlanmasını gösterir. Çok eşgörunümlü kuyruk yöneticisi yapılanışını doęrulamak için, çok eşgörunümlü kuyruk yöneticileriyle aynı örnekleri kullanabilirsiniz.

Örnek, "Linux üzerinde çok eşgörunümlü kuyruk yöneticisi yaratılması" sayfa 537 içinde açıklanan çok eşgörunümlü yapılandırmayı kullanır. Çok eşgörunümlü kuyruk yöneticisinin yedek yönetim ortamına geçtiğini doęrulamak için yapılandırmayı kullanın. Kuyruk yöneticisini **endmqm** komutuyla durdurun ve -s, yedek sisteme geçiş, seçeneğini kullanın. İstemci programları yeni kuyruk yöneticisi yönetim ortamına yeniden bağlanır ve kısa bir gecikmeden sonra yeni yönetim ortamıyla çalışmaya devam eder.

Örnekte, istemci Windows 7 Service Pack 1 sisteminde çalışır. Sistem, çok eşgörunümlü kuyruk yöneticisini çalıştıran iki VMware Linux sunucusunu barındırıyor.

### **IBM MQ Explorer komutunu kullanarak hata durumunda yedek sisteme geçişin doęrulanması**

Hata durumunda yedek sisteme geçiři doęrulamak için örnek uygulamaları kullanmadan önce her bir sunucuda IBM MQ Explorer komutunu çalıştırın. **Uzak Kuyruk Yöneticisi Ekle > Doęrudan çok eşgörunümlü bir kuyruk yöneticisine bağlan** sihirbazını kullanarak her iki kuyruk yöneticisi eşgörunümünü de her gezgine ekleyin. Yedek veritabanına izin veren her iki yönetim ortamının da çalıştığından emin olun. VMware görüntüsünü etkin yönetim ortamıyla birlikte çalıştıran pencereyi kapatın, sunucuyu sanal olarak kapatın ya da etkin yönetim ortamını durdurarak yedek yönetim ortamına geçmeye izin verin.

**Not:** Sunucuyu kapatırsanız, bunun /MQHA ! ürününü barındıran sunucu olmadığından emin olun!

**Not: Yedek yönetim ortamına geçmeye izin ver** seçeneęi **Kuyruk Yöneticisini Durdur** iletiřim penceresinde kullanılamayabilir. Kuyruk yöneticisi tek bir eşgörunüm kuyruk yöneticisi olarak çalıştığından seçenek eksik. Bu iřlemi **Yedek yönetim ortamına izin ver** seçeneęi olmadan başlatmış olmanız gerekir. Kuyruk yöneticisini durdurma isteęinin reddedilirse, **Ayrıntılar** penceresine bakın; bunun nedeni, çalışmakta olan bir yedek yönetim ortamı olmaması olabilir.

## Örnek programları kullanarak hata durumunda yedek sisteme geçişin doğrulanması

### Etkin örneği çalıştırmak için bir sunucu seçin

MQHA dizinini ya da dosya sistemini barındırmak için sunuculardan birini seçmiş olabilirsiniz. Etkin sunucuyu çalıştıran VMware penceresini kapatarak hata durumunda yedek sisteme geçişi test etmeyi planlıyorsanız, bunun MQHA !

### Etkin kuyruk yöneticisi eşgörünümünü çalıştıran sunucuda

**Not:** SVRCONN kanalının MCAUSER ayarı mqmolarak çalıştırılması, örnekteki yapılandırma adımlarının sayısını azaltmak için kolaylık sağlar. Başka bir kullanıcı kimliği seçilirse ve sisteminiz örnekte kullanılandan farklı şekilde ayarlanırsa, erişim izni sorunlarıyla karşılaşabilirsiniz. mqm ' i gösterilen bir sistemde MCAUSER olarak kullanmayın; bu, büyük olasılıkla güvenliği büyük ölçüde tehlikeye atacaktır.

1. *ipaddr1* ve *ipaddr2* komutlarını değiştirin ve /MQHA/hasamples.tst. içinde aşağıdaki komutları kaydedin

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER('mqm') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNNAME('ipaddr1 (1414), ipaddr2
(1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
START LISTENER(LISTENER.TCP)
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

2. /MQHA yoluyla bir uçbirim penceresi açın ve komutu çalıştırın:

```
runmqsc -m QM1 < hasamples.tst
```

3. **runmqsc** komutunun çıkışını inceleyerek, dinleyicinin çalıştığını ve kuyruk yöneticisi denetimine sahip olduğunu doğrulayın.

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

Ya da TCP/IP dinleyicisinin çalıştığı ve Control = Queue Manager' in olduğu IBM MQ Explorer kullanılarak.

### İstemcide

1. AMQCLCHL.TAB istemci bağlantısı çizelgesini sunucudaki /MQHA/qmgrs/QM1.000/@ipcc dizininden istemcideki C:\ çizelgesine kopyalayın.
2. C:\ yoluyla bir komut istemi açın ve MQCHLLIB ortam değişkenini istemci kanal tanımlama çizelgesini (CCDT) gösterecek şekilde ayarlayın

```
SET MQCHLLIB=C:\
```

3. Komut isteminde şu komutları yazın:

```
start amqsgnac TARGET QM1
start amqsmnac -s SOURCE -t TARGET -m QM1
start amqspnac SOURCE QM1
```

### Etkin kuyruk yöneticisi eşgörünümünü çalıştıran sunucuda

1. Aşağıdakilerden birini yapın:
  - Etkin sunucu örneğiyle VMware görüntüsünü çalıştıran pencereyi kapatın.



- IBM MQ Explorer komutunu kullanarak, etkin kuyruk yöneticisi yönetim ortamını durdurarak yedek yönetim ortamına geçmeye ve yeniden bağlanabilir istemcilere yeniden bağlanma talimatı verin.
2. Üç istemci sonunda bağlantının kesilip yeniden bağlanacağını saptar. Bu yapılandırma, sunucu penceresini kapatırsanız, üç bağlantının da yeniden kurulması yaklaşık yedi dakika sürer. Bazı bağlantılar diğerlerinden önce yeniden kurulur.

## Sonuçlar

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsgnac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

### **Multi** Çok eşgörümlü kuyruk yöneticisinin silinmesi

Multiplatforms üzerinde, çok eşgörümlü bir kuyruk yöneticisini tamamen silmek için, **dltmqm** komutunu kullanarak kuyruk yöneticisini silebilir ve daha sonra, **rmvqmif** ya da **dltmqm** komutlarını kullanarak diğer sunuculardan yönetim ortamlarını kaldırabilirsiniz.

Diğer sunucularda tanımlı eşgörümleri olan bir kuyruk yöneticisini, o kuyruk yöneticisinin tanımlı olduğu herhangi bir sunucuda silmek için **dltmqm** komutunu çalıştırın. **dltmqm** komutunu, üzerinde yarattığınız sunucuda çalıştırmanıza gerek yoktur. Daha sonra, kuyruk yöneticisi tanımlaması olan diğer tüm sunucularda **rmvqmif** ya da **dltmqm** komutunu çalıştırın.

Bir kuyruk yöneticisini ancak durdurulduğunda silebilirsiniz. Sildiğiniz sırada hiçbir eşgörümlü çalışmıyor ve kuyruk yöneticisi, açıkça konuşursak, tek bir kuyruk yöneticisi ya da çok eşgörümlü bir kuyruk yöneticisi değil; yalnızca, kuyruk yöneticisi verilerine ve uzak bir paylaşımda günlük dosyalarına sahip bir kuyruk yöneticisidir. Bir kuyruk yöneticisini sildiğinizde, kuyruk yöneticisi verileri ve günlükleri silinir

ve kuyruk yöneticisi kısmı, **dltmqm** komutunu çalıştırdığınız sunucudaki `mqc.ini` dosyasından kaldırılır. Kuyruk yöneticisini sildiğinizde, kuyruk yöneticisi verilerini ve günlüklerini içeren ağ paylaşımına erişiminiz olması gerekir.

Kuyruk yöneticisinin eşgörünümlerini önceden yarattığınız diğer sunucularda, bu sunuculardaki `mqc.ini` dosyalarında da girişler vardır. Sırayla her sunucuyu ziyaret etmelisiniz ve **rmvmqinf** Kuyruk yöneticisi kısmı adlı komutunu çalıştırarak kuyruk yöneticisi kütanesini kaldırmanız gerekir.

**Linux** **AIX** AIX and Linux sistemlerinde, ağ depolamasına ortak bir `mqc.ini` dosyası yerleştirdiyseniz ve her bir sunucuda `AMQ_MQS_INI_LOCATION` ortam değişkenini ayarlayarak tüm sunuculardan bu dosyaya başvurduysanız, güncellenecek tek bir `mqc.ini` dosyası olduğundan kuyruk yöneticisini sunucularından yalnızca birinden silmeniz gerekir.

## Örnek

### İlk sunucu

**dltmqm** QM1

### Eşgörünümlerin tanımlandığı diğer sunucular

**rmvmqinf** QM1 , ya da

**dltmqm** QM1

## **Multi Çok eşgörünümlü kuyruk yöneticisinin başlatılması ve durdurulması**

Multiplatforms üzerinde yapılandırılan bir kuyruk yöneticisinin tek bir eşgörünüm ya da çok eşgörünümlü bir kuyruk yöneticisi olarak başlatılması ve durdurulması.

Bir sunucu çiftinde çok eşgörünümlü bir kuyruk yöneticisi tanımladığınızda, kuyruk yöneticisini tek bir eşgörünüm kuyruk yöneticisi ya da çok eşgörünümlü bir kuyruk yöneticisi olarak herhangi bir sunucuda çalıştırabilirsiniz.

Çok eşgörünümlü bir kuyruk yöneticisini çalıştırmak için, **strmqm -x QM1** komutunu kullanarak sunuculardan birinde kuyruk yöneticisini başlatın; `-x` seçeneği, yönetim ortamının yedek sisteme geçmesine izin verir. *Etkin eşgörünüm* olur. Aynı **strmqm -x QM1** komutunu kullanarak diğer sunucuda yedek yönetim ortamını başlatın; `-x` seçeneği, yönetim ortamının yedek veritabanı olarak başlamasına izin verir.

Kuyruk yöneticisi şimdi tüm istekleri işleyen tek bir etkin yönetim ortamıyla ve etkin yönetim ortamı başarısız olursa devralmaya hazır bir yedek yönetim ortamıyla çalışıyor. Etkin eşgörünümüne, kuyruk yöneticisi verileri ve günlükleri için dışlayıcı erişim verilir. Yedek veritabanı, kuyruk yöneticisi verileri ve günlükleri için dışlayıcı erişim verilmesini bekler. Yedek veritabanına dışlayıcı erişim verildiğinde, bu etkin yönetim ortamı olur.

Etkin yönetim ortamında **endmqm -s** komutunu çalıştırarak, denetimi yedek yönetim ortamına el ile de geçebilirsiniz. **endmqm -s** komutu, yedek veritabanını kapatmadan etkin yönetim ortamını kapatır. Kuyruk yöneticisi verileri ve günlüklerindeki dışlayıcı erişim kilidi serbest bırakılır ve yedek veritabanı devralır.

Tek bir yönetim ortamı kuyruk yöneticisi olarak farklı sunucularda birden çok yönetim ortamıyla yapılandırılmış bir kuyruk yöneticisini başlatabilir ve durdurabilirsiniz. **strmqm** komutunda `-x` seçeneğini kullanmadan kuyruk yöneticisini başlatacaksanız, diğer makinelerde yapılandırılan kuyruk yöneticisi yönetim ortamlarının yedek yönetim ortamı olarak başlatılması engellenir. Başka bir yönetim ortamı başlatma girişiminde bulunursanız, kuyruk yöneticisi yönetim ortamının yedek veritabanı olarak çalışmasına izin verilmediğini belirten bir yanıt alırsınız.

`-s` seçeneği olmadan **endmqm** komutunu kullanarak çok eşgörünümlü bir kuyruk yöneticisinin etkin eşgörünümünü durdurursanız, etkin ve yedek yönetim ortamlarının her ikisi de durur. Yedek yönetim ortamını `-x` seçeneğiyle **endmqm** komutunu kullanarak durdurursanız, yedek veritabanı olmayı durdurur ve etkin yönetim ortamı çalışmaya devam eder. Yedek veritabanında `-x` seçeneği olmadan **endmqm** komutu verilemez.

Aynı anda yalnızca iki kuyruk yöneticisi yönetim ortamı çalıştırılabilir; biri etkin yönetim ortamı, diğeri yedek yönetim ortamı. Aynı anda iki yönetim ortamı başlatacaksanız, IBM MQ hangi yönetim ortamının

etkin yönetim ortamı olacağı üzerinde denetime sahip değildir; bu, ağ dosya sistemi tarafından belirlenir. Kuyruk yöneticisi verilerine dışlayıcı erişim elde eden ilk yönetim ortamı etkin yönetim ortamı olur.

**Not:** Başarısız olan bir kuyruk yöneticisini yeniden başlatmadan önce, uygulamalarınızın kuyruk yöneticisinin o yönetim ortamıyla bağlantısını kesmeniz gerekir. Bunu yapmazsanız, kuyruk yöneticisi doğru olarak yeniden başlamayabilir.

### Multi **Paylaşılan kütük sistemi**

Multipatforms üzerinde, çok eşgörünümlü bir kuyruk yöneticisi, kuyruk yöneticisi yönetim ortamlarını yönetmek için ağa bağlı bir dosya sistemi kullanır.

Çok eşgörünümlü bir kuyruk yöneticisi, dosya sistemi kilitlerinin ve paylaşılan kuyruk yöneticisi verilerinin ve günlüklerinin birleşimini kullanarak hata durumunda yedek sisteme geçişi otomatikleştirir. Bir kuyruk yöneticisinin yalnızca bir eşgörünümü, paylaşılan kuyruk yöneticisi verilerine ve günlüklere özel erişime sahip olabilir. Erişim sağlandığında etkin eşgörünüm olur. Dışlayıcı erişimin alınmasında başarılı olmayan diğer yönetim ortamı, kuyruk yöneticisi verileri ve günlükleri kullanılabilir oluncaya kadar yedek yönetim ortamı olarak bekler.

Ağ bağlantılı dosya sistemi, etkin kuyruk yöneticisi yönetim ortamı için tuttuğu kilitlerin serbest bırakılmasından sorumludur. Etkin yönetim ortamı bir şekilde başarısız olursa, ağ bağlantılı dosya sistemi etkin yönetim ortamı için tuttuğu kilitleri serbest bırakır. Dışlayıcı kilit serbest bırakılır bırakılmaz, kilidi bekleyen bir yedek kuyruk yöneticisi kilidi elde etmeyi dener. Başarılı olursa, etkin yönetim ortamı olur ve kuyruk yöneticisi verilerine ve paylaşılan dosya sistemindeki günlüklere özel erişimi olur. Daha sonra çalışmaya devam eder.

İlgili [Planning file system support](#) (Dosya sistemi desteği planlaması) başlığı altında, dosya sisteminizin çok eşgörünümlü kuyruk yöneticilerini destekleyip desteklemediği açıklanır.

Çok eşgörünümlü bir kuyruk yöneticisi sizi dosya sistemindeki bir hataya karşı korumaz. Verilerinizi korumanın birkaç yolu vardır.

- Yedek disk dizileri (RAID) gibi güvenilir depolamaya yatırım yapın ve bunları ağ esnekliğine sahip bir ağa bağlı dosya sistemine ekleyin.
- IBM MQ doğrusal günlüklerini alternatif ortama yedekle ve birincil günlük ortamınız arızalanırsa, diğer ortamdaki günlükleri kullanarak kurtarın. Bu işlemi denetlemek için bir yedek kuyruk yöneticisini kullanabilirsiniz.

### Multi **Birden çok kuyruk yöneticisi örneği**

Çok eşgörünümlü bir kuyruk yöneticisi, hatadan sonra kuyruk yöneticisi kullanılabilirliğini geri yüklemek için yedek kuyruk yöneticisi yönetim ortamını kullandığı için esnekler.

Kuyruk yöneticisi eşgörünümünün eşlenmesi, kuyruk yöneticisi işlemlerinin kullanılabilirliğini artırmanın çok etkili bir yoludur. Basit bir kullanılabilirlik modeli kullanarak, yalnızca örnek olarak: Bir kuyruk yöneticisinin bir eşgörünümünün güvenilirliği %99 ise (bir yıl içinde kümülatif kapalı kalma süresi 3.65 gündür), kuyruk yöneticisinin başka bir eşgörünümünün eklenmesi kullanılabilirliği 99.99% (bir yıl içinde, yaklaşık bir saatlik kümülatif kapalı kalma süresi) olarak artırır.

Bu, size kullanılabilirlik hakkında pratik sayısal tahminler veremeyecek kadar basit bir modeldir. Kullanılabilirliği gerçekçi bir şekilde modellemek için, arızalar (MTBF) ile ortalama onarım süresi (MTTR) arasındaki ortalama süre ve arızalar ile onarım süreleri arasındaki zamanın olasılık dağılımına ilişkin istatistikleri toplamanız gerekir.

Çok eşgörünümlü kuyruk yöneticisi terimi, kuyruk yöneticisi verilerini ve günlüklerini paylaşan kuyruk yöneticisinin etkin ve yedek yönetim ortamlarının birleşimini ifade eder. Çok eşgörünümlü kuyruk yöneticileri, kuyruk yöneticisinin bir yönetim ortamının bir sunucuda etkin olması ve kuyruk yöneticisinin başka bir yönetim ortamının başka bir sunucuda yedekte beklemede olması, etkin yönetim ortamının başarısız olması durumunda otomatik olarak devralmaya hazır olması nedeniyle kuyruk yöneticisi işlemlerinin başarısız olmasına karşı sizi korur.

## Yedek sisteme geçiş ya da yedek sisteme geçiş

Bir yedek kuyruk yöneticisi yönetim ortamı, istek üzerine (yedek sisteme geçiş) ya da etkin yönetim ortamı başarısız olduğunda (yedek sisteme geçiş) etkin yönetim ortamından devralır.

- *Switchover* , etkin kuyruk yöneticisi yönetim ortamına yayınlanmakta olan **endmqm** -s komutuna yanıt olarak bir yedek yönetim ortamı başlatıldığında gerçekleşir. Kuyruk yöneticisinin nasıl aniden durdurulduğunu denetlemek için **endmqm** parametreleri -c, -i ya da -p belirtebilirsiniz.

**Not:** Yedek kuyruk yöneticisi yönetim ortamı önceden başlatıldıysa yedek sisteme geçiş gerçekleşir. **endmqm** -s komutu etkin kuyruk yöneticisi kilidini serbest bırakır ve yedek kuyruk yöneticisi yönetim ortamı başlatmaz.

- Yönetim ortamı beklenmedik bir şekilde durduğu için ( **endmqm** komutu verilmeden) etkin yönetim ortamı tarafından tutulan kuyruk yöneticisi verilerindeki kilit serbest bırakıldığında *yedek sisteme geçiş* oluşur.

Yedek yönetim ortamı etkin yönetim ortamı olarak devraldığında, kuyruk yöneticisi hata günlüğüne bir ileti yazar.

Yeniden bağlanabilir istemciler, bir kuyruk yöneticisi başarısız olduğunda ya da geçiş yaptığında otomatik olarak yeniden bağlanır. İstemcinin yeniden bağlanmasını istemek için **endmqm** komutuna -r işaretini eklemeniz gerekmez. Otomatik istemci yeniden bağlantısı IBM MQ classes for Javatarafından desteklenmez.

Yedeği devreye sokma işlemi gerçekleşmiş ve yedek yönetim ortamı etkin duruma gelmiş olsa da, başarısız olan bir yönetim ortamını yeniden başlatamadığınızı fark ederseniz, yerel olarak başarısız olan yönetim ortamına bağlı uygulamaların başarısız olan yönetim ortamıyla bağlantısının kesilip kesilmediğini denetleyin.

Başarısız olan yönetim ortamının yeniden başlatılabilmesi için, yerel olarak bağlı uygulamaların başarısız olan bir kuyruk yöneticisi yönetim ortamını sona erdirmesi ya da bağlantısını kesmesi gerekir. Yönetim ortamının yeniden başlatılmasını önlemek için, başarısız olan bir yönetim ortamı eylemine yönelik bir bağlantıyı tutan, paylaşılan bağ tanımlarını (varsayılan ayar) kullanan yerel olarak bağlı uygulamalar.

Yerel olarak bağlı uygulamalar sona erdirilemiyorsa ya da yerel kuyruk yöneticisi yönetim ortamı başarısız olduğunda bunların bağlantısının kesildiğinden emin olun, yalıtılmış bağ tanımlarını kullanmayı düşünün. Yalıtılmış bağ tanımlarını kullanan yerel olarak bağlı uygulamalar, bağlantıyı kesmeseler de, yerel kuyruk yöneticisi yönetim ortamının yeniden başlatılmasını engellemez.

## Kanal ve istemci yeniden bağlantısı

Kanal ve istemci yeniden bağlantısı, yedek kuyruk yöneticisi yönetim ortamı etkin olduktan sonra ileti işleminin geri yüklenmesinin önemli bir parçasıdır.

Farklı ağ adreslerine sahip sunuculara çok eşgörunümlü kuyruk yöneticisi yönetim ortamları kurulur. IBM MQ kanallarını ve istemcilerini tüm kuyruk yöneticisi yönetim ortamları için bağlantı bilgileriyle yapılandırmanız gerekir. Bir yedek veritabanı devraldığında, istemciler ve kanallar yeni ağ adresindeki yeni etkin kuyruk yöneticisi yönetim ortamına otomatik olarak yeniden bağlanır. Otomatik istemci yeniden bağlantısı IBM MQ classes for Javatarafından desteklenmez.

Tasarım, HA-CMP gibi yüksek kullanılabilirlikli ortamlardan farklıdır. HA-CMP, küme için sanal bir IP adresi sağlar ve adresi etkin sunucuya aktarmanızı sağlar. IBM MQ yeniden bağlantısı, IP adreslerini değiştirmez ya da yeniden yönlendirmez. Kanal tanımlarında ve istemci bağlantılarında tanımladığınız ağ adreslerini kullanarak yeniden bağlanarak çalışır. Bir denetimci olarak, kanal tanımlamalarında ve istemci bağlantılarında çok eşgörunümlü kuyruk yöneticisinin tüm yönetim ortamlarına ilişkin ağ adreslerini tanımlamanız gerekir. Ağ adreslerini çok eşgörunümlü bir kuyruk yöneticisine yapılandırmanın en iyi yolu bağlantıya bağlıdır:

### Kuyruk yöneticisi kanalları

Kanalların CONNAME özneliği, bağlantı adlarının virgülle ayrılmış bir listesidir; örneğin, CONNAME ('127.0.0.1(1234), 192.0.2.0(4321)'). Bağlantılar, bağlantı başarıyla kuruluncaya kadar bağlantı listesinde belirlenen sırayla denenecektir. Bağlantı başarılı olmazsa, kanal yeniden bağlanmayı dener.

## Küme kanalları

Genellikle, çok eşgörünümlü kuyruk yöneticilerinin bir kümede çalışmasını sağlamak için ek yapılandırma gerekmez.

Bir kuyruk yöneticisi bir havuz kuyruk yöneticisine bağlanırsa, havuz kuyruk yöneticisinin ağ adresini keşfeder. Kuyruk yöneticisindeki CLUSRCVR kanalının CONNAME ögesine başvurur. TCPIP ' de kuyruk yöneticisi, atlarsanız CONNAME değerini otomatik olarak ayarlar ya da boş olarak yapılandırır. Yedek bir yönetim ortamı devraldığına, bu yönetim ortamının IP adresi önceki etkin yönetim ortamının IP adresini CONNAMEolarak değiştirir.

Gerekliyse, CONNAME ögesini kuyruk yöneticisi yönetim ortamlarının ağ adresleri listesiyle el ile yapılandırabilirsiniz.

## İstemci bağlantıları

İstemci bağlantıları, diğer bağlantıları seçmek için bağlantı listelerini ya da kuyruk yöneticisi gruplarını kullanabilir.

Yedek sisteme geçiş gerçekleştiğinde, yeniden bağlantı biraz zaman alır. Yedek kuyruk yöneticisinin başlatmayı tamamlaması gerekir. Başarısız olan kuyruk yöneticisine bağlanan istemcilerin bağlantı hatasını saptaması ve yeni bir istemci bağlantısı başlatması gerekir. Yeni etkin duruma gelen yedek kuyruk yöneticisini yeni bir istemci bağlantısı seçerse, istemci aynı kuyruk yöneticisine yeniden bağlanır.

İstemci yeniden bağlanma sırasında bir MQI çağrısının ortasında, çağrı tamamlanmadan önce uzun bir bekleme süresine izin verilmelidir.

Hata bir ileti kanalında toplu aktarma sırasında oluşursa, toplu iş geriye işlenir ve yeniden başlatılır.

Geçiş işlemi, yedek sisteme geçişten daha hızlıdır ve yalnızca kuyruk yöneticisinin bir eşgörünümünü durdurup diğerini başlatması kadar sürer. Yeniden yürütmek için yalnızca birkaç günlük kaydı olan bir kuyruk yöneticisi için, en iyi geçiş birkaç saniye sürebilir. Hata durumunda yedek sisteme geçişin ne kadar süreceğini tahmin etmek için, hatanın saptanması için gereken süreyi eklemeniz gerekir. En iyi ihtimalle algılama 10 saniye sürer ve ağa ve dosya sistemine bağlı olarak birkaç dakika sürebilir.

## Multi Uygulama kurtarma

Uygulama kurtarma, hata durumunda yedek sisteme geçiş işleminden sonra uygulama işlemlerinin otomatik olarak devam etmesini sağlar. Hata durumunda yedek sisteme geçiş işleminden sonra uygulama kurtarma dikkatli tasarım gerektirir. Bazı uygulamaların hata durumunda yedek sisteme geçişin gerçekleştiğinin farkında olması gerekir.

Uygulama kurtarmanın amacı, uygulamanın kısa bir gecikmeyle işlemeye devam etmektir. Yeni işlemeye devam etmeden önce, uygulamanın hata sırasında işlediği iş birimini geri çekmesi ve yeniden göndermesi gerekir.

Uygulama kurtarma sorunu, IBM MQ MQI client ile kuyruk yöneticisi arasında paylaşılan ve kuyruk yöneticisinde saklanan bağlamı kaybediyor. IBM MQ MQI client bağlamın çoğunu geri yükler, ancak bağlamın güvenilir bir şekilde geri yüklenemeyen bazı kısımları vardır. Aşağıdaki kısımlarda, uygulama kurtarmanın bazı özellikleri ve bunların çok eşgörünümlü bir kuyruk yöneticisine bağlı uygulamaların kurtarılmasını nasıl etkilediği açıklanmaktadır.

## İşlemsel ileti sistemi

Hata durumunda yedek sisteme geçiş, iletileri teslim etme perspektifinden IBM MQ ileti sisteminin kalıcı özelliklerini değiştirmez. İletiler kalıcıysa ve iş birimleri içinde doğru şekilde yönetiliyorsa, hata durumunda yedek sisteme geçiş sırasında iletiler kaybolmaz.

Hareket işleme perspektifinden hareketler, yedeği devreye sokmadan sonra geriletiliyor ya da kesinleştiriliyor.

Kesinleştirilmemiş hareketler geriye işlenir. Yedeği devreye sokma işleminden sonra, yeniden bağlanabilir bir uygulama, işlemin başarısız olduğunu gösteren bir MQRC\_BACKED\_OUT neden kodu alır. Daha sonra, hareketin yeniden başlatılması gerekir.

Kesinleştirilmiş hareketler, iki aşamalı kesinleştirmenin ikinci aşamasına ya da MQCMIT' i başlatan tek aşamalı (yalnızca ileti) hareketlere ulaşılan hareketlerdir.

Kuyruk yöneticisi hareket eşgüdümçüye ve MQCMIT , başarısızlıktan önce iki aşamalı kesinleştirmesinin ikinci aşamasına başladıysa, hareket başarıyla tamamlanır. Tamamlanma, kuyruk yöneticisinin denetimi altındadır ve kuyruk yöneticisi yeniden çalışırken devam eder. Yeniden bağlanabilir bir uygulamada MQCMIT çağrısı normal şekilde tamamlanır.

Yalnızca iletileri içeren tek aşamalı kesinleştirmede, kesinleştirmeyi başlatmış olan bir hareket, yeniden çalıştıktan sonra kuyruk yöneticisinin denetimi altında olağan şekilde tamamlanır. Yeniden bağlanabilir bir uygulamada MQCMIT normal şekilde tamamlanır.

Yeniden bağlanabilir istemciler, hareket eşgüdümçüsü olarak kuyruk yöneticisinin denetimi altında tek fazlı hareketleri kullanabilir. Genişletilmiş hareket istemcisi yeniden bağlantıyı desteklemiyor. Hareket işleme istemcisi bağlandığında yeniden bağlanma istenirse, bağlantı başarılı olur, ancak yeniden bağlanma yeteneği olmaz. Bağlantı yeniden bağlanamıyormuş gibi davranır.

## Uygulamayı yeniden başlatma ya da sürdürme

Hata durumunda yedek sisteme geçiş, bir uygulamayı kesintiye uğratar. Bir hatadan sonra uygulama baştan yeniden başlatılabilir ya da kesintiye takiben işlemeye devam edebilir. İkincisi, *otomatik istemci yeniden bağlantısı* olarak adlandırılır. Otomatik istemci yeniden bağlantısı IBM MQ classes for Javatarafından desteklenmez.

Bir IBM MQ MQI client uygulamasıyla, istemciyi otomatik olarak yeniden bağlamak için bir bağlantı seçeneği ayarlayabilirsiniz. Seçenekler şunlardır: MQCNO\_RECONNECT ya da MQCNO\_RECONNECT\_Q\_MGR. Herhangi bir seçenek ayarlanmazsa, istemci otomatik olarak yeniden bağlanmayı denemez ve kuyruk yöneticisi hatası istemciye MQRC\_CONNECTION\_BROKEN değerini döndürür. İstemciyi, yeni bir MQCONN ya da MQCONNX çağrısı yayınlayarak yeni bir bağlantı başlatmayı deneyecek şekilde tasarlayabilir.

Sunucu programlarının yeniden başlatılması gerekir; kuyruk yöneticisi ya da sunucu başarısız olduğunda, kuyruk yöneticisi tarafından işlendikleri noktada bunlar otomatik olarak yeniden bağlanamaz. Çok eşgörünümlü bir kuyruk yöneticisi yönetim ortamı başarısız olduğunda, IBM MQ sunucu programları genellikle yedek kuyruk yöneticisi yönetim ortamında yeniden başlatılmaz.

Bir IBM MQ sunucu programının yedekleme sunucusunda yeniden başlatılmasını iki şekilde otomatikleştirebilirsiniz:

1. Sunucu uygulamanızı bir kuyruk yöneticisi hizmeti olarak paketleyin. Yedek kuyruk yöneticisi yeniden başlatıldığında yeniden başlatılır.
2. Örneğin, bir yedek kuyruk yöneticisi eşgörünümü tarafından başlatılırken yazılan hata durumunda yedek sisteme geçiş günlük iletiliyle tetiklenen kendi hata durumunda yedek sisteme geçiş mantığınızı yazın. Bundan sonra, kuyruk yöneticisine bağlantı yaratmak için, uygulama yönetim ortamının başlatıldıktan sonra MQCONN ya da MQCONNX ögesini çağırması gerekir.

## Hata durumunda yedek sisteme geçiş algılanıyor

Bazı uygulamaların hata durumunda yedek sisteme geçişi bilmeleri gerekir, bazılarının bilmeleri gerekmez. Bu iki örneği göz önünde bulundurun.

1. Bir ileti alışverişi kanalı üzerinden ileti alan ya da alan bir ileti alışverişi uygulaması, normalde kanalın diğer ucundaki kuyruk yöneticisinin çalışmasını gerektirmez; kanalın diğer ucundaki kuyruk yöneticisi yedek bir yönetim ortamında yeniden başlarsa bundan etkilenme olasılığı düşüktür.
2. IBM MQ MQI client uygulaması, bir kuyruktaki kalıcı ileti girişini işler ve kalıcı ileti yanıtlarını tek bir iş biriminin parçası olarak başka bir kuyruğa koyar: İş birimini yeniden başlatarak eşitleme noktası içinde bir MQRC\_BACKED\_OUT neden kodu MQPUT, MQGET ya da MQCMIT işlerse, hiçbir ileti kaybolmaz. Ayrıca, uygulamanın bir bağlantı hatasıyla başa çıkabilmek için özel bir işlem yapması gerekmez.

Ancak, ikinci örnekte uygulamanın MQGET seçeneğini ( MQGMO\_MSG\_UNDER\_CURSOR) kullanarak işlenecek iletiyi seçmek için kuyruğa göz atmakta olduğunu varsayın. Yeniden bağlanma, göz atma imlecini ilk durumuna getirir ve MQGET çağrısı doğru iletiyi döndürmez. Bu örnekte, uygulamanın hata durumunda

yedek sisteme geçişin gerçekleştiğinin farkında olması gerekir. Ayrıca, imleç altındaki ileti için başka bir MQGET yayınlamadan önce, uygulamanın göz atma imlecini geri yüklemesi gerekir.

Göz atma imlecinin kaybedilmesi, yeniden bağlantı sonrasında uygulama bağlamının nasıl değiştiğinin bir örneğidir. Diğer vakalar "Otomatik olarak yeniden bağlanan bir istemcinin kurtarılması" sayfa 547'inde belgelenmiştir.

Hata durumunda yedek sisteme geçişi izleyen IBM MQ MQI client uygulamaları için üç alternatif tasarım kalıbınız vardır. Yalnızca birinin hata durumunda yedek sisteme geçişi algılamasına gerek yoktur.

### **Yeniden bağlantı yok**

Bu örnekte, bağlantı kesildiğinde uygulama yürürlükteki bağlantıda tüm işlemleri durdurur. Uygulamanın işlemeye devam edebilmesi için kuyruk yöneticisiyle yeni bir bağlantı kurması gerekir. Uygulama, yeni bağlantı üzerinde işlemeye devam etmek için gerekli olan durum bilgilerini aktarmaktan tamamen sorumludur. Bağlantılarını kaybettikten sonra bir kuyruk yöneticisiyle yeniden bağlantı kuran var olan istemci uygulamaları bu şekilde yazılır.

İstemci, bağlantı kesildikten sonra sonraki MQI çağrısından MQRC\_CONNECTION\_BROKEN ya da MQRC\_Q\_MGR\_NOT\_AVAILABLE gibi bir neden kodu alır. Uygulama, kuyruk tanıtıcıları gibi tüm IBM MQ durum bilgilerini atmalı ve yeni bir bağlantı kurmak için yeni bir MQCONN ya da MQCONNX çağrısı yayınlamalı ve işlemesi gereken IBM MQ nesnelere yeniden açmalıdır.

Varsayılan MQI davranışı, kuyruk yöneticisiyle bağlantı kesildikten sonra kuyruk yöneticisi bağlantı tanıtıcılarının kullanılamaz duruma gelmesidir. Varsayılan değer, hata durumunda yedek sisteme geçiş işleminden sonra uygulamanın yeniden bağlanmasını önlemek için MQCONNX üzerinde MQCNO\_RECONNECT\_DISABLED seçeneğinin ayarlanmasıyla eşdeğerdir.

### **Hata durumunda yedek sisteme geçiş toleransı**

Uygulamayı, yedeği devreye sokmadan etkilenmeyecek şekilde yazın. Bazen dikkatli hata işleme, hata durumunda yedek sisteme geçiş ile başa çıkmak için yeterlidir.

### **Yeniden bağlantıya duyarlı**

Bir MQCBT\_EVENT\_HANDLER olay işleyicisini kuyruk yöneticisine kaydettirin. İstemci sunucuya yeniden bağlanmayı denemeye başladığında olay işleyici MQRC\_RECONNECTING ile ve başarılı bir yeniden bağlantıdan sonra MQRC\_RECONNECTED ile gönderilir. Daha sonra, istemci uygulamasının işlemeye devam edebilmesi için öngörülebilir bir durumu yeniden oluşturmak üzere bir yordamı çalıştırabilirsiniz.

## **Otomatik olarak yeniden bağlanan bir istemcinin kurtarılması**

Hata durumunda yedek sisteme geçiş beklenmeyen bir olaydır ve otomatik olarak yeniden bağlanan bir istemcinin tasarlandığı gibi çalışması için yeniden bağlantının sonuçları tahmin edilebilir olmalıdır.

Beklenmeyen bir hatayı öngörülebilir ve güvenilir bir kurtarmaya dönüştürmenin ana unsuru, işlemlerin kullanılmaktır.

In the previous section, an example, "2" sayfa 546, was given of an IBM MQ MQI client using a local transaction to coordinate MQGET and MQPUT. İstemci bir MQRC\_BACKED\_OUT hatasına yanıt olarak bir MQCMIT ya da MQBACK çağrısı yayınlar ve geriletilen işlemi yeniden sunar. Kuyruk yöneticisi hatası hareketin geriletilmesine neden olur ve istemci uygulamasının davranışı hareket olmamasını ve ileti olmamasını sağlar.

Geri çağırma için, geri çağırma tüketici değiştirgesi durumu: MQCS\_SUSPENDED\_USER\_ACTION ise, kullanan uygulamanın sürdürülmesi gerekebileceğini unutmayın.

Her program durumu bir işlemin parçası olarak yönetilmez ve bu nedenle yeniden bağlantının sonuçları anlaşılması zorlaşır. Hata durumunda yedek kuyruk yöneticisine geçişte hayatta kalacak şekilde istemci uygulamanızı tasarlamak için yeniden bağlantının IBM MQ MQI client durumunu nasıl değiştirdiğini bilmeniz gerekir.

Uygulamanızı özel bir hata durumunda yedek sisteme geçiş kodu olmadan tasarlamaya karar verebilir ve yeniden bağlantı hatalarını diğer hatalarla aynı mantıkla işlemeye karar verebilirsiniz. Diğer bir seçenek olarak, yeniden bağlantının özel hata işleme gerektirdiğini fark edebilir ve hata durumunda yedek sisteme



geçiş işlemek için bir yordamı çalıştırmak üzere bir olay işleyicisi IBM MQ ' e kaydettirmeyi seçebilirsiniz. Yordam yeniden bağlanma işlemini kendisi gerçekleştirebilir ya da ana program iş parçacığına, işleme devam ettiğinde kurtarma işlemi gerçekleştirmesi gerektiğini belirten bir işaret ayarlayabilir.

IBM MQ MQI client ortamı, hata durumunda yedek sisteme geçişin kendisini bilir ve yeniden bağlanmayı takiben, bazı durum bilgilerini istemcide saklayarak ve IBM MQ durumunu geri yüklemek için istemci uygulaması adına ek MQI çağrılarını vererek bağlamı geri yükler. Örneğin, hata noktasında açık olan nesnelerin tanıtıcıları geri yüklenir ve geçici dinamik kuyruklar aynı adla açılır. Ama kaçınılmaz değişiklikler var ve bu değişikliklerle başa çıkmak için tasarınıza ihtiyacınız var. Değişiklikler beş tür olarak sınıflandırılabilir:

1. Yeni ya da daha önce tanılanmamış hatalar, uygulama programı tarafından tutarlı bir bağlam durumu geri yükleninceye kadar MQI çağrılarında döndürülür.

Yeni bir hata alma örneği, yeniden bağlantıdan önce bağlamı kaydettikten sonra bağlamı geçirmeye çalışırken MQRC\_CONTEXT\_NOT\_AVAILABLE dönüş kodudur. Güvenlik bağlamı yetkisiz bir istemci programına geçirilmediğinden, yeniden bağlantıdan sonra bağlam geri yüklenemez. Bunu yapmak için, kötü amaçlı bir uygulama programının güvenlik bağlamını edinmesi gerekir.

Genellikle, uygulamalar genel ve öngörülebilir hataları dikkatle tasarlanmış bir şekilde işler ve sık rastlanmayan hataları genel bir hata işleyicisine iletir. Hata işleyici IBM MQ bağlantısını kesip yeniden bağlanabilir ya da programı tümüyle durdurabilir. Sürekliliği artırmak için bazı hatalarla farklı bir şekilde başa çıkmanız gerekebilir.

2. Kalıcı olmayan iletiler kaybolabilir.
3. Hareketler geriye işlenir (zamanuyumsuz tüketicileri de askıya alabilir, önceki metne bakın).
4. Bir eşitleme noktası dışında kullanılan MQGET ya da MQPUT çağrılarını, olası bir ileti kaybıyla kesintiye uğrayabilir.
5. Bir MQI çağrısındaki uzun bir bekleme nedeniyle zamanlama hataları ortaya çıktı.

Kayıp bağlamla ilgili bazı ayrıntılar aşağıdaki bölümde listelenmiştir.

- Kalıcı olmayan iletiler, NPMCLASS (HIGH) seçeneğiyle bir kuyruğa konmadıkça atılır ve kuyruk yöneticisi hatası, sona erdirme sırasında kalıcı olmayan iletileri saklama seçeneğini kesmez.
- Bir bağlantı kesildiğinde kalıcı olmayan abonelik kaybedilir. Yeniden bağlantı kurulduğunda, yeniden kurulur. Sürekli abonelik kullanmayı düşünün.
- Get-wait interval yeniden hesaplanır; sınırı aşırsa MQRC\_NO\_MSG\_AVAILABLE değerini döndürür. Benzer şekilde, abonelik süre bitimi de aynı genel süre sonu süresini vermek üzere yeniden hesaplanır.
- Göz atma imlecinin bir kuyruktaki konumu kaybolur; genellikle ilk iletiden önce yeniden oluşturulur.
  - MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR ya da MQGMO\_MSG\_UNDER\_CURSOR belirten MQGET çağrılarını, neden kodu MQRC\_NO\_MSG\_AVAILABLE ile başarısız olur.
  - Göz atma için kilitlenen iletilerin kilidi açılır.
  - İşle kapsamı olan işaretli iletilere göz at işareti kaldırılmaz ve yeniden göz atılabilir.
  - Çoğu durumda işaretli iletilere birlikte göz atma işareti kaldırılmaz.
- Güvenlik bağlamı kaybedildi. MQPMO\_PASS\_ALL\_CONTEXT ile ileti koymak gibi kaydedilmiş ileti bağlamını kullanma girişimleri MQRC\_CONTEXT\_NOT\_AVAILABLE ile başarısız olur.
- İleti belirteçleri kayboldu. MQGET ileti simgesi kullanarak MQRC\_NO\_MSG\_AVAILABLE neden kodunu döndürür.

**Not:** *MsgId* ve *CorrelId*, iletinin bir parçası olarak, hata durumunda yedek sisteme geçiş sırasında iletiyle birlikte korunur ve böylece MQGET *MsgId* ya da *CorrelId* iş beklendiği gibi çalışır.

- Kesinleştirilmemiş bir harekette uyumluluk noktası altında bir kuyruğa konan iletiler artık kullanılamaz.
- İletilerin mantıksal bir sırayla ya da bir ileti grubunda işlenmesi, yeniden bağlantıdan sonra MQRC\_RECONNECT\_INCOMPATIBLE dönüş koduyla sonuçlanır.
- Bir MQI çağrısı, istemcilerin genellikle bugün aldığı daha genel MQRC\_CONNECTION\_BROKEN yerine MQRC\_RECONNECT\_FAILED döndürebilir.



- IBM MQ MQI client iletinin kuyruk yöneticisine başarıyla teslim edilip edilmediğini bilmiyorsa, MQPUT tutarlılık noktası dışında arama sırasında yeniden bağlanma MQRC\_CALL\_INTERRUPTED değerini döndürür. MQCMIT sırasında yeniden bağlantı benzer şekilde davranır.
- MQRC\_CALL\_INTERRUPTED döndürülür-başarılı bir yeniden bağlanma işleminden sonra- IBM MQ MQI client , kuyruk yöneticisinden başarıyı ya da başarısızlığı belirtmek için herhangi bir yanıt almazsa
  - Eşitleme noktası dışında bir MQPUT çağrısı kullanılarak kalıcı bir iletinin teslim edilmesi.
  - Eşitleme noktası dışında bir MQPUT1 çağrısı kullanılarak, kalıcı bir iletinin ya da varsayılan kalıcılığı olan bir iletinin teslim edilmesi.
  - MQCMIT çağrısı kullanılarak bir hareketin kesinleştirilmesi. Yanıt yalnızca başarılı bir yeniden bağlantıdan sonra döndürülür.
- Kanallar yeni örnek olarak yeniden başlatılır (farklı kanallar da olabilir) ve kanal çıkış durumu korunmaz.
- Geçici dinamik kuyruklar, geçici dinamik kuyrukları açık olan yeniden bağlanabilir istemcileri kurtarma işleminin bir parçası olarak geri yüklenir. Geçici bir dinamik kuyruktaki hiçbir ileti geri yüklenmez, ancak kuyruğun açık olduğu ya da kuyruğun adını hatırladığı uygulamalar işlemeye devam edebilir.

Kuyruk, onu yaratan uygulamadan başka bir uygulama tarafından kullanılıyorsa, bir sonraki başvuruda bulunulacak kadar hızlı bir şekilde geri yüklenmeyebilir. Örneğin, bir istemci yanıt kuyruğu olarak geçici bir dinamik kuyruk oluşturursa ve bir yanıt ileti bir kanal tarafından kuyruğa yerleştirilecekse, kuyruk zamanında kurtarılamayabilir. Bu durumda, kanal genellikle yanıt iletiyi gitmeyen ileti kuyruğuna yerleştirir.

Yeniden bağlanabilir bir istemci uygulaması ada göre geçici bir dinamik kuyruk açarsa (başka bir uygulama tarafından yaratıldığından), yeniden bağlantı oluştuğunda IBM MQ MQI client geçici dinamik kuyruğu yeniden oluşturamaz; bu kuyruğun yaratılacağı modeli yoktur. MQI 'da, modele göre geçici dinamik kuyruğu yalnızca bir uygulama açabilir. Geçici dinamik kuyruğu kullanmak isteyen diğer uygulamalar MQPUT1'ya da sunucu bağ tanımlarını kullanmalı ya da başarısız olursa yeniden bağlantıyı yeniden deneyebilmelidir.

Yalnızca kalıcı olmayan iletiler geçici bir dinamik kuyruğa konabilir ve bu iletiler yedek sisteme geçiş sırasında kaybolur; bu kayıp, yeniden bağlantı sırasında MQPUT1 kullanılarak geçici bir dinamik kuyruğa konan iletiler için geçerlidir. MQPUT1 sırasında hata durumunda yedek sisteme geçiş gerçekleşirse, MQPUT1 başarılı olsa da ileti konmamış olabilir. Bu sorunun geçici bir çözümü, kalıcı dinamik kuyrukların kullanılmasıdır. Herhangi bir sunucu bağ tanımları uygulaması, yeniden bağlanabilir olmadığı için geçici dinamik kuyruğu ada göre açabilir.

## Multi

### **Veri kurtarma ve yüksek düzeyde kullanılabilirlik**

Çok eşgörünümlü kuyruk yöneticilerini kullanan yüksek kullanılabilirlikli çözümler, bir depolama hatasından sonra verileri kurtarmak için bir mekanizma içermelidir.

Çok eşgörünümlü bir kuyruk yöneticisi, kuyruk yöneticisi işlemlerinin kullanılabilirliğini artırır, ancak kuyruk yöneticisinin iletileri ve diğer bilgileri saklamak için kullandığı dosya sistemi gibi diğer bileşenlerin kullanılabilirliğini artırmaz.

Verileri yüksek düzeyde kullanılabilir hale getirmenin bir yolu, ağa bağlı esnek veri depolamasını kullanmaktır. Ağa bağlı bir dosya sistemi ve esnek veri depolaması kullanarak kendi çözümünüzü oluşturabilir ya da bütünleştirilmiş bir çözüm satın alabilirsiniz. Esnekliği olağanüstü durumdan kurtarma ile birleştirmek istiyorsanız, on ya da yüzlerce kilometre üzerinde disk kopyalamasına izin veren zamanuyumsuz disk eşlemesi kullanılabilir.

Ortamı en iyi şekilde kullanmak için farklı IBM MQ dizinlerinin depolama ortamıyla eşlenme şeklini yapılandırabilirsiniz. *Çok eşgörünümlü* kuyruk yöneticileri için, iki IBM MQ dizin ve dosya tipi arasında önemli bir fark vardır.

### **Bir kuyruk yöneticisinin eşgörünümleri arasında paylaşılması gereken dizinler.**

Bir kuyruk yöneticisinin farklı yönetim ortamları arasında paylaşılması gereken bilgiler iki dizinde bulunur: qmqs ve logs dizinleri. Dizinler, paylaşılan bir ağ dosya sisteminde olmalıdır. İletiler oluşturulup silindikçe veriler sürekli değiştiği için sürekli yüksek kullanılabilirlik ve mükemmel performans sağlayan bir depolama ortamı kullanmanız önerilir.

### **Bir kuyruk yöneticisinin eşgörünümleri arasında paylaşılması olmayan dizinler ve dosyalar.**

Diğer bazı dizinlerin, bir kuyruk yöneticisinin farklı yönetim ortamları arasında paylaşılması gerekmez ve ikizlenmiş dosya sistemi dışında hızlı bir şekilde geri yüklenir.

- IBM MQ yürütülebilir dosyaları ve araçlar dizini. Yeniden kurarak ya da yedeklenmiş bir dosya arşivinden yedekleyerek ve geri yükleyerek değiştirin.
- Kuruluş için bir bütün olarak değiştirilen yapılandırma bilgileri. Yapılandırma bilgileri, AIX, Linux, and Windows sistemlerinde `mqsc.ini` dosyası ya da **MQSC** yapılandırma komut dosyaları gibi kendi yapılandırma yönetiminizin bir parçası gibi IBM MQ tarafından yönetilir. Bir dosya arşivini kullanarak yedekleyin ve geri yükleyin.
- İzlemeler, hata günlükleri ve FFDC dosyaları gibi kuruluş genelinde çıktı. Dosyalar, varsayılan veri dizinindeki `errors` ve `trace` alt dizinlerinde saklanır. AIX and Linux sistemlerinde varsayılan veri dizini şudur: `/var/mqm`. Windows üzerinde varsayılan veri dizini IBM MQ kuruluş dizinidir.

Doğrusal günlük kaydını kullanarak çok eşgörünümlü bir kuyruk yöneticisinin olağan ortam yedeklerini almak için bir yedek kuyruk yöneticisini de kullanabilirsiniz. Yedekleme kuyruğu yöneticisi, ikizlenmiş bir dosya sistemi kadar hızlı kurtarma sağlamaz ve son yedeklemeden bu yana yapılan değişiklikleri kurtarmaz. Yedek kuyruk yöneticisi mekanizması, yerleştirilmiş bir depolama hatasından sonra bir kuyruk yöneticisini kurtarmaktansa, iş yeri dışında olağanüstü durumdan kurtarma senaryolarında kullanılmak için daha uygundur.

### **IBM MQ Kullanılabilirlik çözümlerini birleştirme**

Uygulamalar, kullanılabilirliği artırmak için diğer IBM MQ yeteneklerini kullanır. Çok eşgörünümlü kuyruk yöneticileri, diğer yüksek düzeyde kullanılabilirlik yeteneklerini tamamlar.

### **IBM MQ Kümeler kuyruk kullanılabilirliğini artırır**

Bir küme kuyruğuna ilişkin birden çok tanımlama yaratarak, kümedeki her bir yöneticide en çok bir kuyruk yaratarak kuyruk kullanılabilirliğini artırabilirsiniz.

Kümenin bir üyesinin başarısız olduğunu ve küme kuyruğuna yeni bir ileti gönderildiğini varsayın. *iletisinin* başarısız olan kuyruk yöneticisine gitmesi gerekmiyorsa, ileti, kümede kuyruk tanımlaması olan başka bir çalışan kuyruk yöneticisine gönderilir.

Kümeler kullanılabilirliği büyük ölçüde artırsa da, iletilerin gecikmesine neden olan iki ilgili hata senaryosu vardır. Çok eşgörünümlü kuyruk yöneticileriyle bir küme oluşturulması, bir iletinin geciktirilmesi olasılığını azaltır.

#### **Ayrılmış iletiler**

Kümedeki bir kuyruk yöneticisi başarısız olursa, kümedeki diğer kuyruk yöneticilerine yönlenebilecek başka bir ileti başarısız olan kuyruk yöneticisine yöneltilmez. Gönderilen iletiler, başarısız olan kuyruk yöneticisi yeniden başlatılıncaya kadar ayrılmış olur.

#### **Yakınlıklar**

Benzerlik, aksi halde iki ayrı hesaplama arasında paylaşılan bilgileri tanımlamak için kullanılan terimdir. Örneğin, bir sunucuya istek iletisi gönderen bir uygulama ile yanıtı işlemeyi bekleyen aynı uygulama arasında benzerlik vardır. Başka bir örnek, önceki iletilere bağlı olarak her iletinin işlenmesi olan bir ileti sırası olabilir.

Kümelenmiş kuyruklara ileti gönderirseniz, yakınlıkları göz önünde bulundurmanız gerekir. Aynı kuyruk yöneticisine ardışık iletiler göndermeniz mi gerekiyor, yoksa her ileti kümenin herhangi bir üyesine gidebilir mi?

Kümedeki aynı kuyruk yöneticisine ileti göndermeniz gerekirse ve ileti başarısız olursa, iletiler, başarısız olan küme kuyruk yöneticisi yeniden çalışıncaya kadar gönderenin iletim kuyruğunda bekler.

Küme çok eşgörünümlü kuyruk yöneticileriyle yapılandırıldıysa, başarısız olan kuyruk yöneticisinin yeniden başlatılmasını bekleme süresi, yedek veritabanı devralırken bir dakika ya da daha kısa bir süre ile sınırlıdır. Yedek veritabanı çalışırken, ayrılmış iletiler işlenmeye devam eder, yeni etkinleştirilen kuyruk yöneticisi eşgörünümlü giden kanallar başlatılır ve iletim kuyruklarında bekleyen iletiler akmaya başlar.

Bir kümeyi, başarısız olan bir kuyruk yöneticisi tarafından geciktirilen iletilerin üstesinden gelecek şekilde yapılandırmanın olası bir yolu, kümedeki her bir sunucuya iki farklı kuyruk yöneticisi konuşlandırmak ve birinin etkin, diğerinin farklı kuyruk yöneticilerinin yedek eşgörünümlü olmasını ayarlamaktır. Bu bir etkin yedekleme yapılandırmasıdır ve kümenin kullanılabilirliğini artırır.

Daha az yönetim ve daha fazla ölçeklenebilirlik avantajlarının yanı sıra kümeler, çok eşgörünümlü kuyruk yöneticilerini tamamlamak için ek kullanılabilirlik öğeleri sağlamaya devam eder. Kümeler, bir kuyruk yöneticisinin hem etkin hem de bekleme eşgörünümlerini etkileyen diğer hata tiplerine karşı koruma sağlar.

### **Kesintisiz hizmet**

Küme kesintisiz bir hizmet sağlar. Küme tarafından alınan yeni iletiler, işlenmek üzere etkin kuyruk yöneticilerine gönderilir. Beklemedeki kuyruk yöneticisinin arızayı algılaması ve başlatmasını tamamlaması, kanallarının yeniden bağlanması ve başarısız olan ileti gruplarının yeniden sunulması için zaman aldığından, çok eşgörünümlü bir kuyruk yöneticisinin kesintisiz bir hizmet sağlamasına güvenmeyin.

### **Yerelleştirilmiş kesinti**


Etkin, beklemedeki ve dosya sistemi sunucularının kabul edilebilir performans sağlamak için milisaniye hızlarında etkileşim kurmaları gerekirken, bu sunuculardan ne kadar uzakta olabileceğine ilişkin pratik sınırlamalar vardır.

Kümelenmiş kuyruk yöneticileri, etkileşim hızlarının birkaç saniye olmasını gerektirir ve coğrafi olarak dünyanın herhangi bir yerine dağıtılabilir.

### **İşletim hatası**

Kullanılabilirliği artırmak için iki farklı mekanizma kullanarak, insan hatası gibi bir operasyonel hatanın kullanılabilirlik çabalarınızı tehlikeye atması olasılığını azaltabilirsiniz.

## **Kuyruk paylaşım grupları, ileti işleme kullanılabilirliğini artırır**

 Yalnızca z/OS üzerinde sağlanan kuyruk paylaşım grupları, bir kuyruk yöneticisi grubunun bir kuyruk hizmeti paylaşmasına izin verir. Bir kuyruk yöneticisi başarısız olursa, diğer kuyruk yöneticileri kuyruktaki tüm iletileri işlemeye devam eder. Çok eşgörünümlü kuyruk yöneticileri z/OS platformunda desteklenmez ve kuyruk paylaşım gruplarını yalnızca daha geniş bir ileti sistemi mimarisinin bir parçası olarak tamamlar.

## **IBM MQ Müşteriler, uygulama kullanılabilirliğini artırır**

IBM MQ MQI client programları, kuyruk yöneticisi kullanılabilirliğine, bağlantı ağırlıklandırmalarına ve benzerliklere dayalı olarak bir kuyruk yöneticisi grubundaki farklı kuyruk yöneticilerine bağlanabilir. Bir uygulamayı kuyruk yöneticisinin çalıştığı makineden farklı bir makinede çalıştırarak, bağlı olduğu kuyruk yöneticisi eşgörünümlü başarısız olursa uygulamayı yeniden bağlamanın bir yolu olduğu sürece çözümün genel kullanılabilirliğini artırabilirsiniz.

Kuyruk yöneticisi grupları, durdurulan bir kuyruk yöneticisinden istemciyi ayıklayarak istemci kullanılabilirliğini artırmak ve IP yayıcısı gibi kuyruk yöneticilerinden oluşan bir gruptaki istemci bağlantılarını yük dengelemek için kullanılır. İstemci uygulamasının, belirli bir kuyruğa bağımlılık gibi, başarısız olan kuyruk yöneticisiyle herhangi bir yakınlığı olmamalıdır ya da işlemeyi sürdüremez.

Otomatik istemci yeniden bağlantısı ve çok eşgörünümlü kuyruk yöneticileri, bazı benzerlik sorunlarını çözerek istemci kullanılabilirliğini artırır. Otomatik istemci yeniden bağlantısı IBM MQ classes for Javatafından desteklenmez.

Bir istemciyi aynı kuyruk yöneticisine yeniden bağlanmaya zorlamak için MQCNO seçeneğini MQCNO\_RECONNECT\_Q\_MGRayarlayabilirsiniz:

1. Önceden bağlı olan tek yönetim ortamı kuyruk yöneticisi çalışmıyorsa, kuyruk yöneticisi yeniden çalışmaya kadar bağlantı yeniden denir.
2. Kuyruk yöneticisi çok eşgörünümlü bir kuyruk yöneticisi olarak yapılandırıldıysa, istemci etkin olan yönetim ortamına yeniden bağlanır.

Aynı kuyruk yöneticisine otomatik olarak yeniden bağlanarak, kuyruk yöneticisinin istemci adına tuttuğu açık kuyruklar ve abone olduğu konu gibi durum bilgilerinin çoğu geri yüklenir. İstemci, bir isteğe yanıt almak için dinamik bir yanıt kuyruğu açtıysa, yanıt kuyruğuna yönelik bağlantı da geri yüklenir.

Linux

MQ Adv.

## RDQM yüksek kullanılabilirlik düzeyi

RDQM (eşlenmiş veri kuyruğu yöneticisi), x86-64 platformları için Red Hat Enterprise Linux üzerinde kullanılabilen yüksek kullanılabilirlikli bir çözümdür.

RDQM yapılandırması, her biri kuyruk yöneticisinin eşgörünümüne sahip bir yüksek kullanılabilirlik (HA) grubunda yapılandırılan üç sunucudan oluşur. Bir yönetim ortamı, verilerini diğer iki yönetim ortamına zamanuyumlu olarak eşleyen, çalışmakta olan kuyruk yöneticisidir. Bu kuyruk yöneticisini çalıştıran sunucu başarısız olursa, kuyruk yöneticisinin başka bir eşgörünümü başlatılır ve üzerinde çalışılmakta olan veriler vardır. Kuyruk yöneticisinin üç eşgörünümü isteğe bağlı olarak bir kayan IP adresini paylaşabilir, bu nedenle istemcilerin yalnızca tek bir IP adresiyle yapılandırılması gerekir. HA grubu ağ sorunları nedeniyle bölünmüş olsa da, kuyruk yöneticisinin aynı anda yalnızca bir yönetim ortamı çalıştırılabilir. Kuyruk yöneticisini çalıştıran sunucu 'birincil' olarak bilinir, diğer iki sunucunun her biri 'ikincil' olarak bilinir.

Üç düğüm, bölünmüş beyin durumunun ortaya çıkma olasılığını büyük ölçüde azaltmak için kullanılır. İki düğümlü bir Yüksek Kullanılabilirlik sisteminde bölünmüş beyin, iki düğüm arasındaki bağlantı bozulduğunda ortaya çıkabilir. Bağlantı olmadan, her iki düğüm de kuyruk yöneticisini aynı anda çalıştırarak farklı veriler biriktirebilir. Bağlantı geri yüklendiğinde, verilerin iki farklı sürümü (bir 'uçtan beyin') vardır ve hangi veri kümesinin tutulup hangisinin atılacağına karar vermek için el ile müdahale gerekir.

RDQM, bölünmüş beyin durumunu önlemek için çekirdek içeren üç düğümlü bir sistem kullanır. Diğer düğümlerden en az biriyle iletişim kurabilen düğümler bir çekirdek oluşturur. Kuyruk yöneticileri yalnızca çekirdek içeren bir düğümde çalışabilir. Kuyruk yöneticisi, en az bir başka düğüme bağlı olmayan bir düğümde çalışamaz; bu nedenle, aynı anda iki düğümde hiçbir zaman çalışamaz:

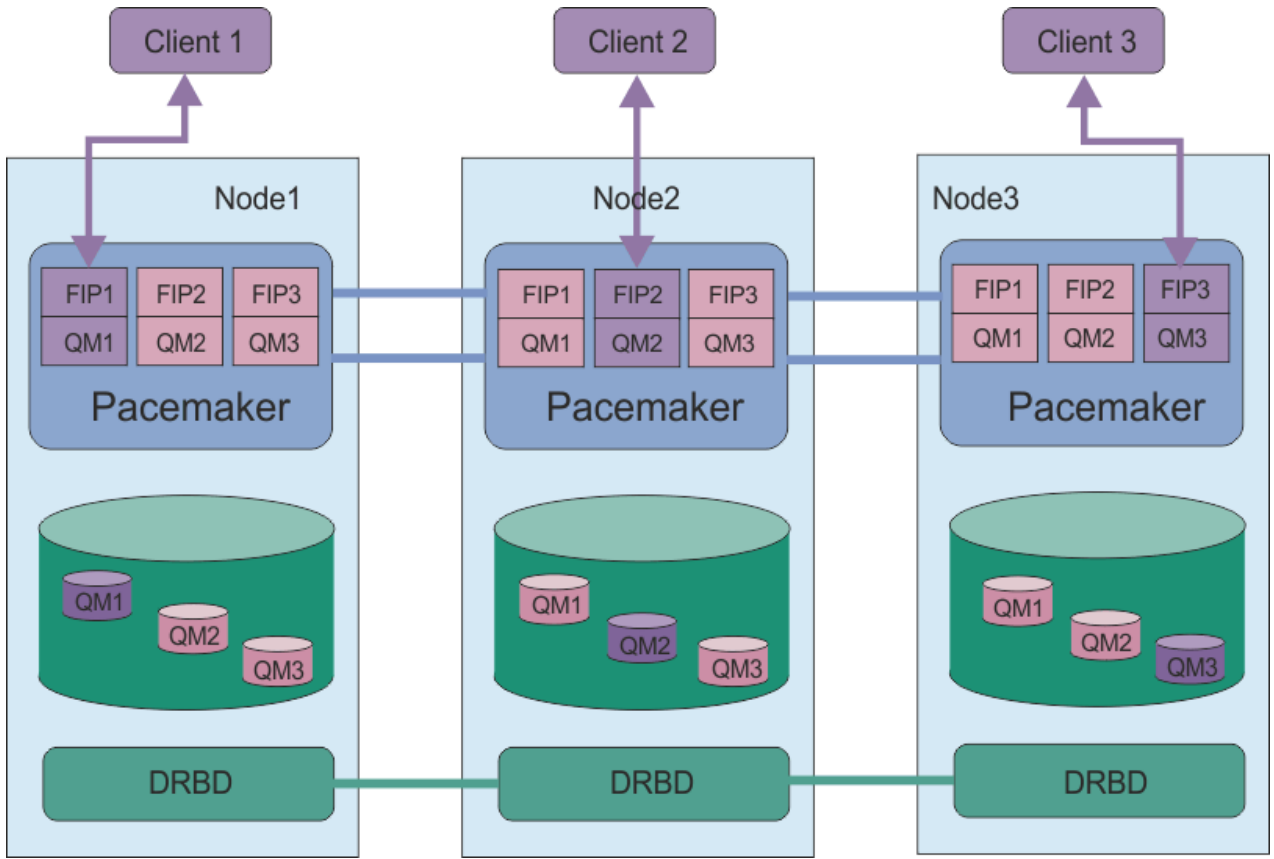
- Tek bir düğüm başarısız olursa, kuyruk yöneticisi diğer iki düğümden birinde çalışabilir. İki düğüm başarısız olursa, düğüm yeterli çekirdeğe sahip olmadığı için kuyruk yöneticisi geri kalan düğümde çalışamaz (geri kalan düğüm diğer iki düğümün başarısız olup olmadığını ya da hala çalışıp çalışmadığını ve bağlantırlığını kaybettiğini anlayamaz).
- Tek bir düğüm bağlantırlığı kaybederse, düğüm yeterli çekirdeğe sahip olmadığı için kuyruk yöneticisi bu düğümde çalışamaz. Kuyruk yöneticisi, yeterli kaynağı olan geri kalan iki düğümden birinde çalışabilir. Tüm düğümler bağlantırlığı kaybederse, düğümlerin hiçbirinde yeterli çekirdek olmadığından kuyruk yöneticisi hiçbir düğümde çalışamaz.

**Not:** IBM MQ Console , eşlenmiş veri kuyruğu yöneticilerini desteklemez. IBM MQ Explorer ' i eşlenmiş veri kuyruğu yöneticileriyle kullanabilirsiniz, ancak bu işlem RDQM özelliklerine özgü bilgileri görüntüleyemez.

Üç düğümün grup yapılandırması Pacemaker tarafından işlenir. Üç düğüm arasındaki eşleme DRBD tarafından işlenir. (DRBD ile ilgili bilgi için bkz. <https://clusterlabs.org/pacemaker/> . Pacemaker ve <https://docs.linbit.com/docs/users-guide-9.0/> .)

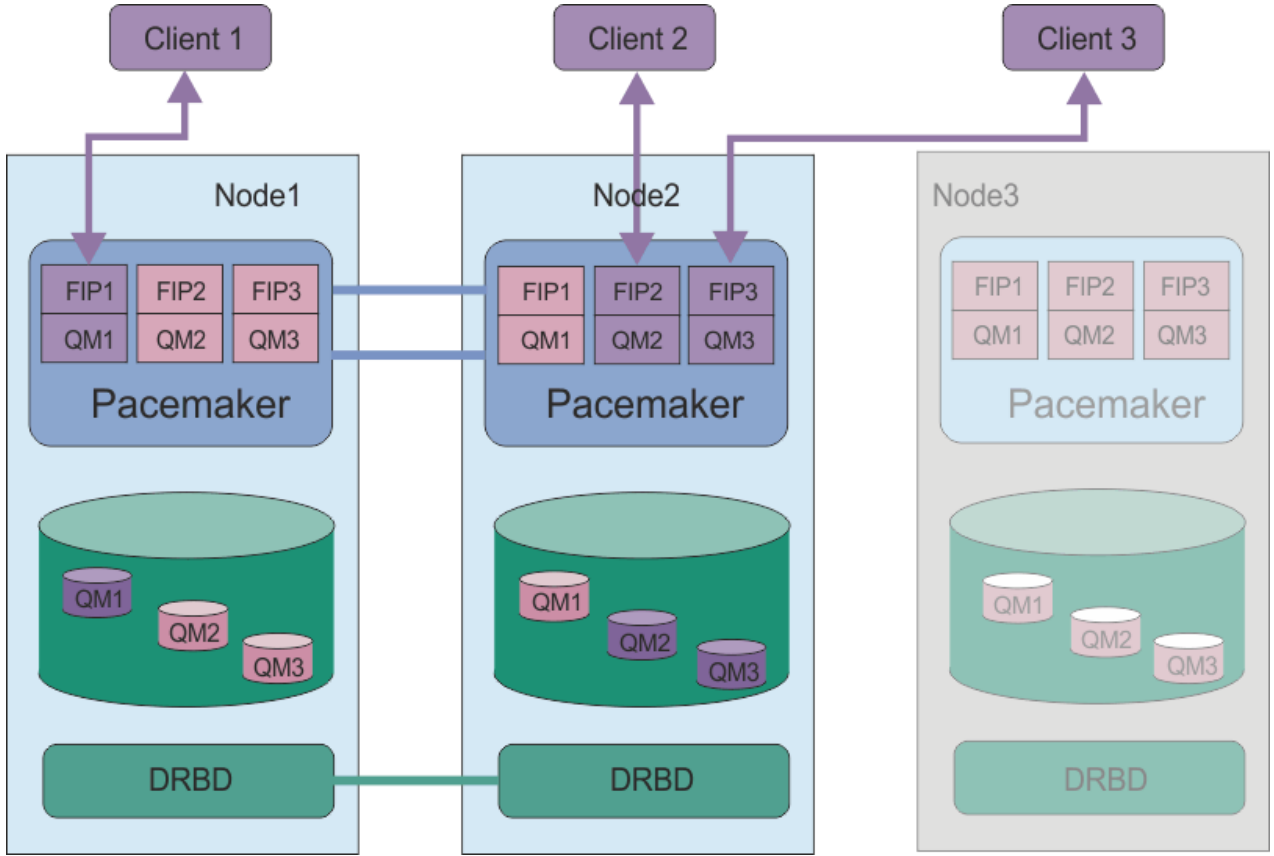
“Kuyruk yöneticisi verileri yedekleniyor” sayfa 656’inde açıklanan işlemi kullanarak eşlenen veri kuyruğu yöneticilerinizi yedekleyebilirsiniz. Kuyruk yöneticisinin durdurulması ve yedeklenmesi, RDQM yapılandırması tarafından yapılan düğüm izlemesini etkilemez.

Aşağıdaki şekil, HA grubundaki üç düğümün her birinde çalışan bir RDQM ile tipik bir konuşlandırmayı göstermektedir.



Şekil 78. Üç adet RDIMM içeren HA grubu örneği

Sonraki şekilde Node3 başarısız oldu, Pacemaker bağlantıları kayboldu ve kuyruk yöneticisi QM3 Node2 üzerinde çalışıyor.



Şekil 79. node3 başarısız olduktan sonraki örnek

**Not:** Kuyruk yöneticileri başka bir düğüme geçtiğinde, yedeği devreye sokma durumlarını korurlar. Çalışmakta olan kuyruk yöneticileri başlatılır, durdurulan kuyruk yöneticileri durdurulmuş olarak kalır.

### İlgili görevler

- [RDQM ' i kurma \(eşlenen veri kuyruğu yöneticileri\)](#)
- [RDQM için bakım düzeyi güncellemeleri uyguluyor](#)
- [Eşlenmiş veri kuyruğu yöneticileri geçiriliyor](#)
- [RDQM yapılandırmalarında sorun giderme](#)

### Linux RDQM HA çözümü için gereksinimler

RDQM yüksek kullanılabilirlik (HA) grubunu yapılandırmadan önce bir dizi gereksinimi karşılamamız gerekir.

### Sistem gereksinimleri

RDQM HA grubunu yapılandırmadan önce, HA grubunun bir parçası olacak üç sunucunun her birinde bir yapılandırmayı tamamlamamız gerekir.

- Her düğüm için drbdpooladlı bir birim grubu gerekir. Eşlenen her veri kuyruğu yöneticisi için depolama alanı, bu birim grubundan kuyruk yöneticisi başına ayrı bir mantıksal birim olarak ayrılır. En iyi performans için bu birim grubu, iç disk sürücülerine (tercihen SSD ' ler) karşılık gelen bir ya da daha fazla fiziksel birimden oluşmalıdır. RDQM HA çözümünü kurmadan önce ya da kurduktan sonra drbdpool oluşturabilirsiniz, ancak gerçekten herhangi bir RDIMM oluşturmadan önce drbdpool yaratmanız gerekir. **vgs** komutunu kullanarak birim grubu yapılandırmanızı denetleyin. Çıktı aşağıdakine benzer olmalıdır:

```
VG      #PV #LV #SN Attr   VSize  VFree
drbdpool 1   9   0 wz--n- <16.00g <7.00g
rhe1    1   2   0 wz--n- <15.00g  0
```

Özellikle, özniteliklerin altıncı sütununda (yani, wz - - nc) c karakteri bulunmadığını kontrol edin. c , kümelemenin etkinleştirildiğini ve etkinleştirildiyse, birim grubunu silmeniz ve kümelemeden yeniden oluşturmanız gerektiğini gösterir.

- d1rbdpool birim grubunu oluşturduktan sonra başka bir işlem yapmayın. IBM MQ , d1rbdpool içinde oluşturulan mantıksal birimleri ve bunların nasıl ve nereye bağlandığını yönetir.
- Her düğüm, RDQM desteğini yapılandırmak için kullanılan en çok üç arabirim gerektirir:
  - Pacemaker ' in HA grubunu izlemesi için birincil arabirim.
  - Pacemaker ' in HA grubunu izlemesi için alternatif bir arabirim.
  - Eşleme arabirimi olarak bilinen zamanuyumlu veri eşlemesi için bir arabirim. Bu, HA grubunda çalışan tüm eşlenmiş veri kuyruğu yöneticilerinin beklenen iş yükü göz önüne alındığında eşleme gereksinimlerini desteklemek için yeterli bant genişliğine sahip olmalıdır.

HA grubunu, her üç arabirim için de aynı IP adresinin kullanılabilmesi, her arabirim için ayrı bir IP adresinin kullanılabilmesi ya da eşleme arabirimi için birincil ve alternatif ve ayrı bir IP adresi olarak aynı IP adresinin kullanılabilmesi şeklinde yapılandırabilirsiniz.

Maksimum hata toleransı için, bu arabirimlerin bağımsız Ağ Arabirimi Kartları (NIC) olması gerekir.

- DRBD, RFC 1123 tarafından değiştirilen RFC 952 tarafından tanımlandığı şekilde, HA grubundaki her düğümün geçerli bir Internet anasistem adına ( uname -ntarafından döndürülen değer) sahip olmasını gerektirir.
- HA grubundaki düğümler arasında bir güvenlik duvarı varsa, güvenlik duvarı bir kapı aralığındaki düğümler arasında trafiğe izin vermelidir. RHEL ' de standart güvenlik duvarını çalıştırıyorsanız, gerekli kapıları açan bir örnek komut dosyası ( /opt/mqm/samp/rdqm/firewalld/configure.sh) sağlanır. Komut dosyasını rootolarak çalıştırmanız gerekir. Başka bir güvenlik duvarı kullanıyorsanız, hangi kapıların açılması gerektiğini görmek için /usr/lib/firewalld/services/rdqm\* hizmet tanımlarını inceleyin. Komut dosyası, DRBD, Pacemaker ve IBM MQ için aşağıdaki kalıcı firewallD hizmet kurallarını ekler:
  - MQ\_INSTALLATION\_PATH/samp/rdqm/firewalld/services/rdqm-drbd.xml 7000-7100 TCP bağlantı noktalarına izin verir.
  - MQ\_INSTALLATION\_PATH/samp/rdqm/firewalld/services/rdqm-pacemaker.xml 5404-5407 numaralı UDP kapılarına izin verir
  - MQ\_INSTALLATION\_PATH/samp/rdqm/firewalld/services/rdqm-mq.xml , TCP kapısı 1414 'e izin verir (farklı bir kapı gerekiyorsa komut dosyasını düzenlemeniz gerekir)
- Sistem, Zorlama kipinde SELinux kullanıyorsa, aşağıdaki komutu çalıştırmanız gerekebilir:

```
semanage permissive -a d1rbd_t
```

**V9.3.2** > **V9.3.0.1** drbd-selinux paketini kurduysanız, **semanage** programını çalıştırmanıza gerek yoktur. Bu paketi her düğümde kurmuş olmanız ya da her düğümde **semanage** komutunu çalıştırmanız gerekir.

## Ağ gereksinimleri

Aynı veri merkezinde RDQM HA grubundaki üç düğümü bulmanız önerilir.

Farklı veri merkezlerinde düğümleri bulmayı seçerseniz, aşağıdaki sınırlamaları dikkate alın:

- Veri merkezleri arasında artan gecikme süresiyle performans hızla azalır. IBM , 5 ms 'ye kadar bir gecikme süresi destekleyecek olsa da, uygulama performansınızı 1-2 ms 'den fazla gecikme süresine dayanamayabilirsiniz.
- Eşleme bağlantısı üzerinden gönderilen veriler, IBM MQ AMS kullanılarak gerçekleştirilebilecek verilerin ötesinde herhangi bir ek şifrelemeye tabi değildir.

İsteğe bağlı olarak, istemcinin HA grubunda hangi düğümde çalıştığından bağımsız olarak, eşlenmiş bir veri kuyruğu yöneticisi (RDQM) için aynı IP adresini kullanmasını sağlamak üzere bir kayan IP adresi yapılandırabilirsiniz. Kayan adres, RDQM için birincil düğümdeki adlandırılmış bir fiziksel arabirime

bağlanır. RDQM üzerinden başarısız olursa ve farklı bir düğüm birincil düğüm olursa, kayan IP, yeni birincil düğümde aynı ada sahip bir arabirime bağlanır. Üç düğümdeki fiziksel arabirimlerin tümü aynı ada sahip olmalı ve kayan IP adresiyle aynı alt ağa ait olmalıdır.

## Kümeyi yapılandırmaya ilişkin kullanıcı gereksinimleri

RDQM HA grubunu rootkullanıcısı olarak yapılandırabilirsiniz. rootolarak yapılandırmak istemiyorsanız, bunun yerine mqm grubunda bir kullanıcı olarak yapılandırabilirsiniz. mqm grubundaki bir kullanıcının RDQM kümesini yapılandırması için aşağıdaki gereksinimleri karşılamamız gerekir:

- mqm kullanıcısı, RDQM HA grubunu oluşturan üç sunucunun her birinde komut çalıştırmak için sudo kullanabilmelidir.
- mqm kullanıcısı, RDQM HA grubunu oluşturan üç sunucunun her birinde komut çalıştırmak için parola olmadan SSH kullanabiliyorsa, kullanıcının yalnızca bir sunucuda komut çalıştırması gerekir.
- mqm kullanıcısının üç sunucuda da aynı UID ' ye sahip olması gerekir.
- mqm grubu, üç sunucuda da aynı GID ' e sahip olmalıdır.

mqm kullanıcısının root yetkisiyle aşağıdaki komutları çalıştırabilmesi için sudo ' yı yapılandırmanız gerekir:

```
/opt/mqm/bin/crtmqm  
/opt/mqm/bin/dltmqm  
/opt/mqm/bin/rdqmadm  
/opt/mqm/bin/rdqmstatus
```

## Kuyruk yöneticileriyle çalışmaya ilişkin kullanıcı gereksinimleri

Eşlenen veri kuyruğu yöneticilerini (RQM ' ler) yaratmak, silmek ya da yapılandırmak için hem mqm hem de haclient gruplarına ait bir kullanıcı kimliği kullanılmalıdır ( haclient grubu, Pacemakerkuruluşu sırasında yaratılır).

### Linux Parolasız SSH ayarlanıyor

Parolasız SSH ' yi, HA grubundaki bir düğümde yalnızca yapılandırma komutlarını yayınlamanız gerekeceği şekilde ayarlayabilirsiniz. (parolasız SSH ' nin ayarlanması isteğe bağlıdır; diğer bir seçenek olarak, komutları her düğüme el ile kopyalayabilirsiniz.)

## Bu görev hakkında

Parolasız SSH ' yi ayarlamak için her düğümde mqm kimliğini yapılandırmanız ve sonra o kullanıcı için her düğümde bir anahtar oluşturmanız gerekir. Daha sonra anahtarları diğer düğümlere dağıtın ve her bir düğümü bilinen anasistemler listesine eklemek için bağlantıyı deneyin. Son olarak mqm kimliğini kilitletiniz.

**Not:** Yönergeler, ayrı birincil, alternatif ve eşleme arabirimleriyle bir HA grubu tanımladığınızı varsayar ve bu nedenle birincil ve diğer arabirimler üzerinden parolasız SSH erişimi tanımlarsınız. Bir sistemi tek bir IP adresiyle yapılandırmayı planlıyorsanız, o tek arabirim üzerinden parolasız SSH erişimi tanımlarsınız.

RDQM, ssh komutunun etkileşim olmadan, yani parola istemeden çalışmasını gerektirir.

## Yordam

1. Üç düğümün her birinde, mqm kullanıcısını ayarlamak ve bir SSH anahtarı oluşturmak için aşağıdaki adımları tamamlayın:

- a) mqm ana dizinini /home/mqmolarak değiştirin:

```
usermod -d /home/mqm mqm
```

- b) /home/mqm dizinini oluşturun:

```
mkdir -p /home/mqm
```

- c) mqm parolasını ekleyin:



```
passwd mqm
```

d) Etkileşimli kabuğu mqmolarak çalıştırın:

```
su mqm
```

e) mqm kimlik doğrulama anahtarını oluşturun:

```
ssh-keygen -t rsa -f /home/mqm/.ssh/id_rsa -N ''
```

2. Üç düğümün her birinde, düğümün anahtarını diğer iki düğüme eklemek ve her bir düğümün birincil ve (kullanılıyorsa) diğer adreslerine ilişkin bağlantıları sınamak için aşağıdaki adımları tamamlayın:

a) Anahtarı uzak düğümlere ekle

```
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_primary_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_alternate_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_primary_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_alternate_address
```

b) Uzak düğümler için parolasız ssh değerini denetleyin ve bilinen anasistemleri güncelleyin:

```
ssh remote_node1_primary_address uname -n
ssh remote_node1_alternate_address uname -n
ssh remote_node2_primary_address uname -n
ssh remote_node2_alternate_address uname -n
```

Her bağlantı için, devam etmek istediğinizi doğrulamanız istenir. Her birinin bilinen anasistemleri güncellemesini onaylayın. HA grubunu parolasız SSH kullanarak yapılandırmayı denemedenden önce bunu tamamlamanız gerekir.

c) Etkileşimli kabuktan şu şekilde çıkın: mqm:

```
exit
```

3. Kök olarak her düğümde mqm parolasını kaldırmak ve tanıtıcıyı kilitlemek için aşağıdaki adımları tamamlayın:

a) mqm parolasını kaldırın:

```
passwd -d mqm
```

b) Kilit mqm:

```
passwd -l mqm
```

4. Kök olarak her düğümde, mqm kullanıcısı için sudo erişimi ayarlamak üzere aşağıdaki adımları tamamlayın:

a) **visudo** komutunu kullanarak sudoers dosyasını düzenleyin:

```
visudo
```

b) "**##** Allows people in group wheel to run all commands" satırını arayın ve aşağıdaki metni satırın altına ekleyin:

```
##mqm ALL=(ALL) ALL
```

c) "**##** Same thing without a password" satırını arayın ve aşağıdaki metni satırın altına ekleyin:

```
%mqm ALL=(ALL) NOPASSWD: ALL
```

## Linux **Pacemaker kümesinin tanımlanması (HA grubu)**

HA grubu bir Pacemaker kümesidir. Pacemaker kümesini, /var/mqm/rdqm.ini kütüğünü düzenleyerek ve **rdqmadm** komutunu çalıştırarak tanımlayın.

## Bu görev hakkında

Pacemaker hakkında bilgi için bkz. <https://clusterlabs.org/pacemaker/> . mqm kullanıcısı sudo ' i kullanabiliyorsa, Pacemaker kümesini mqm grubunda bir kullanıcı olarak yaratabilirsiniz. Kullanıcı, parola olmadan her sunucuya SSH ' yi de yapabiliyorsa, Pacemaker kümesini yaratmak için rdqm . ini kütüğünü düzenlemeniz ve **rdqmadm** sunucusunu sunuculardan birinde çalıştırmanız gerekir. Aksi takdirde, dosyayı oluşturmanız ve komutu düğüm olacak her sunucuda root olarak çalıştırmanız gerekir.

rdqm . ini dosyası, Pacemaker kümesindeki düğümler için RDQM tarafından kullanılan IP adreslerini verir. RHEL 8 ve RHEL 9 kuruluşları için, **uname -n** komutunun döndürdüğü şekilde anasistem adı olması gereken her düğümün adını vermeniz gerekir. RHEL 7 kuruluşları için düğüm adı belirtimi isteğe bağlıdır.

Bir RDQM HA grubu, bir, iki ya da üç IP adresini kullanacak şekilde yapılandırılabilir:

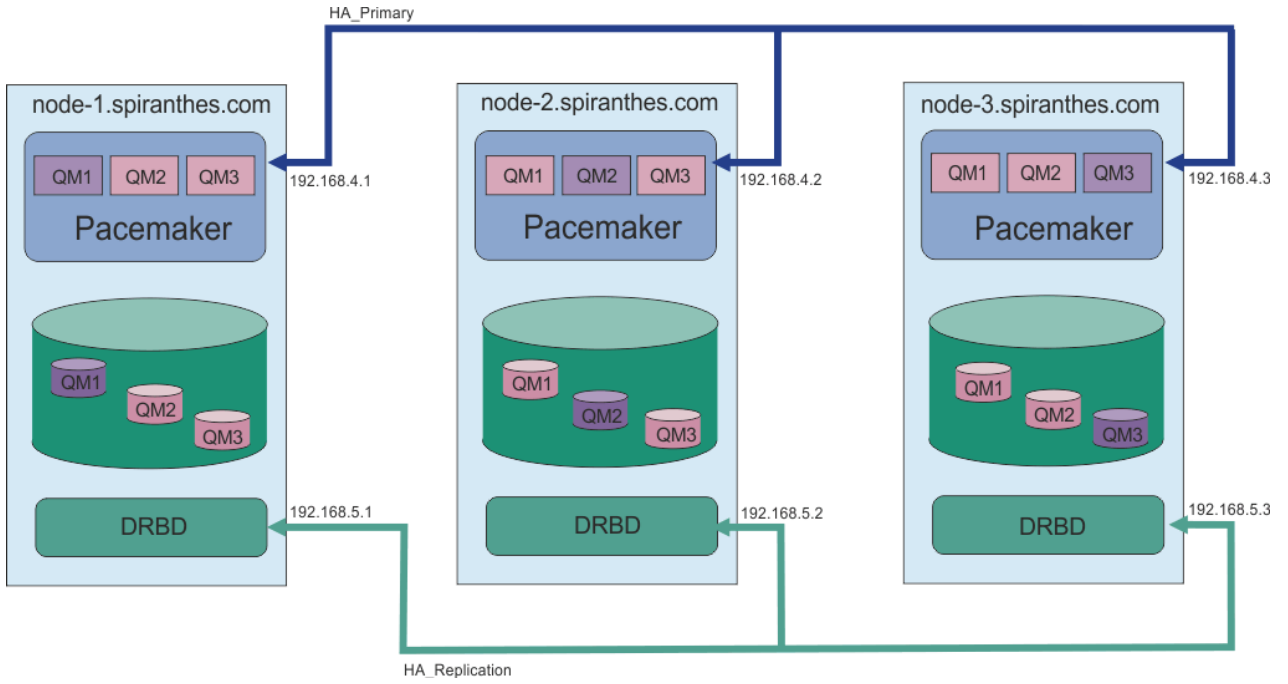
- Bir IP adresi: Heartbeats ve replikasyon aynı bağlantıyı paylaşıyor
- İki IP adresi: Heartbeats ve replikasyon ayrı bağlantılar kullanır
- Üç IP adresi: Eşleme için bir bağlantı ve sağlıklı işletim bildirimleri için iki ayrı bağlantı

Bu seçenekler, RDQM için farklı konuşlandırma kalıplarını desteklemek üzere sağlanır. Kullanılan ortama dayalı olarak RDQM çözümünün esnekliğini en üst düzeye çıkarmak için farklı seçenekler kullanılabilir. İki ya da üç IP adresi kullanan yapılandırmalar öncelikle, düğümler arasındaki bağlantı için yedekliliği yapılandırmak üzere fiziksel ağ bağlantılarının ve eşleme trafiğinin hangi fiziksel ağ üzerinden kullanılacağı üzerinde ayrıntılı denetimin gerekli olduğu devreye alımlar için tasarlanmıştır. Alternatif olarak, yüksek düzeyde kullanılabilir ve esnek bağlantı, örneğin bağlantı toplama kullanılarak ağ katmanında uygulanabilir. Bağlantı toplamada, tek tek fiziksel bağlantıların arızalanması durumunda çalışmaya devam edebilen tek bir mantıksal bağlantı sağlamak için birden çok fiziksel ağ bağlantısı kullanılır. RDQM, ağ bağlantılığının sanallaştırıldığı ve/veya ağ katmanında esnek bağlantılığın uygulandığı bir ortamda devreye alındıysa, hem sağlıklı işletim bildirimleri hem de eşleme için tek bir IP adresi kullanılması genellikle tercih edilir.

Aşağıdaki örnek ğnasıl şekilde şekilde den den iniz üzerinden çünkü den çünkü iki IP adresi kullanılmıştır. rdqm . ini dosyanız her düğüm için bir HA\_Primary ve HA\_Replication alanına sahiptir, ancak HA\_Alternate alanı yoktur:

```
Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Primary=192.168.4.1
  HA_Replication=192.168.5.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Primary=192.168.4.2
  HA_Replication=192.168.5.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Primary=192.168.4.3
  HA_Replication=192.168.5.3
```

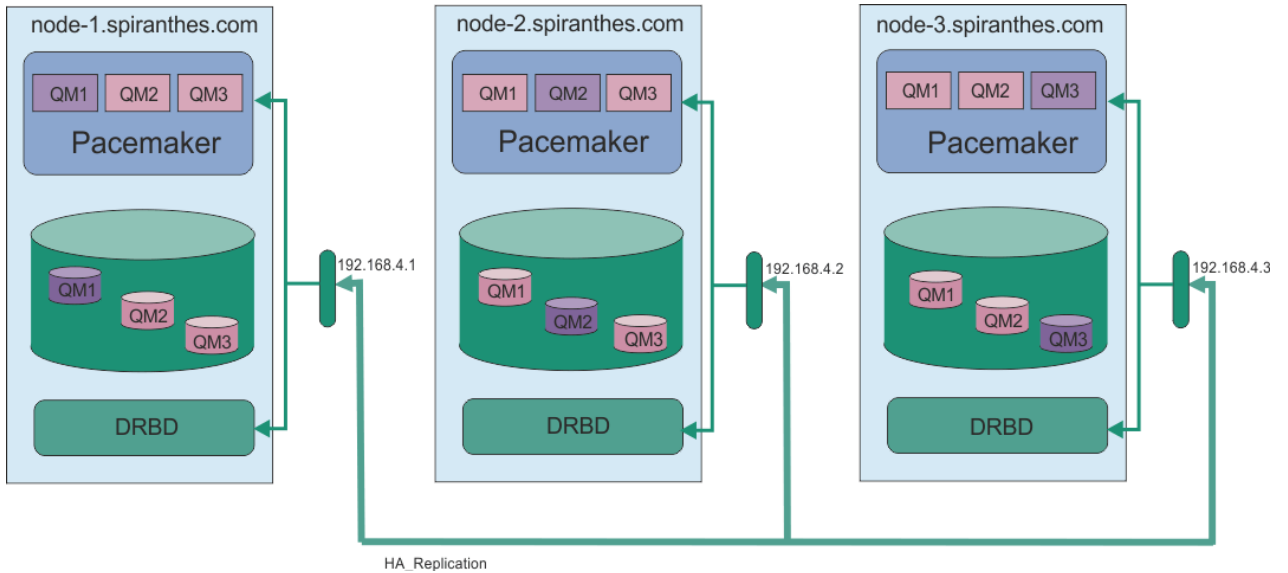
Aşağıdaki çizge bu yapılandırmayı göstermektedir:



Aşağıdaki örnek dosyada, izleme için HA\_Replication arabirimini kullanan örnek bir Pacemaker kümesi için yapılandırma gösterilmektedir (Örneğin, bu, kavram kanıtı konuşlandırması için kullanılabilir). Bu durumda yalnızca HA\_Replication arabirimini belirtirsiniz:

```
Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Replication=192.168.4.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Replication=192.168.4.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Replication=192.168.4.3
```

Aşağıdaki çizge bu yapılandırmayı göstermektedir:



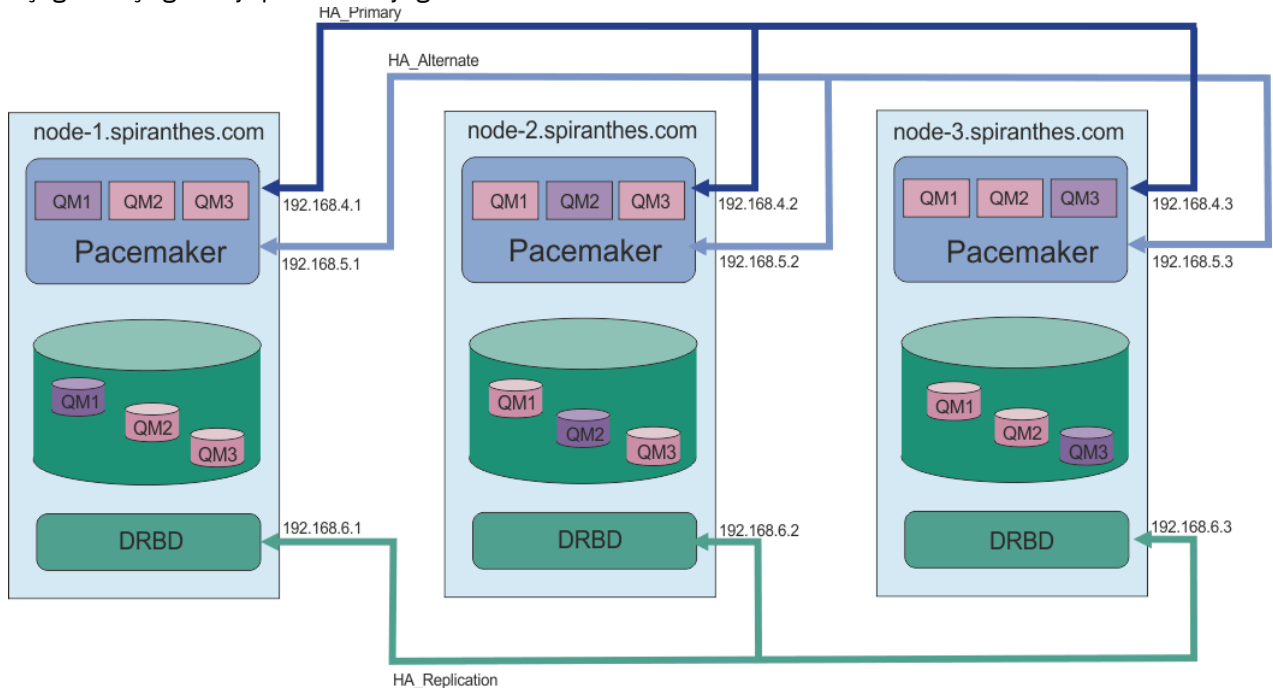
Aşağıdaki örnek dosyada, her arabirim için ayrı bir IP adresi kullanan örnek bir Pacemaker kümesi için yapılandırma gösterilmektedir:

```

Node:
Name=rdqm-node-1.spiranthes.com
HA_Primary=192.168.4.1
HA_Alternate=192.168.5.1
HA_Replication=192.168.6.1
Node:
Name=rdqm-node-2.spiranthes.com
HA_Primary=192.168.4.2
HA_Alternate=192.168.5.2
HA_Replication=192.168.6.2
Node:
Name=rdqm-node-3.spiranthes.com
HA_Primary=192.168.4.3
HA_Alternate=192.168.5.3
HA_Replication=192.168.6.3

```

Aşağıdaki çizge bu yapılandırmayı göstermektedir:



Düğümleri belirttiğiniz sıra, yapılandırmanızdaki tüm `rdqm.ini` dosyalarında aynı olmalıdır. Üç düğümünüzün, hangisi Node1, hangisi Node2 vb. ' dir, ortak bir görünümü olmalıdır.

## Yordam

- Pacemaker kümesini kullanıcı olarak tanımlamak için `root`:
  - Üç sunucudan birindeki `/var/mqm/rdqm.ini` dosyasını düzendosyasının bilirim, bin seçeneğini indaki seçeneğini DEN takıtek?
  - Dosyayı, Pacemaker kümesindeki düğüm olacak diğer iki sunucuya kopyalayın.
  - Aşağıdaki komutu üç sunucunun her birinde `root` olarak çalıştırın:

```
rdqmadm -c
```

- Pacemaker kümesini her düğümdeki `mqm` grubunda bir kullanıcı olarak tanımlamak için:
  - `mqm` kullanıcısının komutları çalıştırmak için **sudo** komutunu kullanabildiğinden emin olun.
  - Üç sunucudan birindeki `/var/mqm/rdqm.ini` kütüğünü düzenleyerek, kütüğün Pacemaker kümesini tanımlamasını sağlar.
  - `/var/mqm/rdqm.ini` dosyasını, Pacemaker kümesindeki düğüm olacak diğer iki sunucuya kopyalayın.

d) Her sunucuda aşağıdaki komutu çalıştırın:

```
rdqmadm -c
```

- Pacemaker kümesini mqm grubunda bir düğümden kullanıcı olarak tanımlamak için:
  - a) mqm kullanıcısının komutları çalıştırmak için **sudo** komutunu kullanabildiğinden ve parola olmadan SSH kullanarak her bir sunucuya bağlanabildiğinden emin olun.
  - b) Üç sunucudan birindeki /var/mqm/rdqm.ini kütüğünü düzenleyerek, kütüğün Pacemaker kümesini tanımlamasını sağlar.
  - c) Aşağıdaki komutu çalıştırın:

```
rdqmadm -c
```

## İlgili başvurular

[rdqmadm \(eşlenmiş veri kuyruğu yöneticisi kümesini yönet\)](#)

Linux

*Pacemaker kümesinin silinmesi (HA grubu)*

HA grubu bir Pacemaker kümedir. **rdqmadm** komutunu -u seçeneğiyle çalıştırarak bir Pacemaker küme yapılandırmasını silebilirsiniz.

## Bu görev hakkında

Eşlenmiş veri kuyruğu yöneticileri düğümlerin herhangi birinde bulunmaya devam ediyorsa, Pacemaker küme yapılandırmasını silemezsiniz.

## Yordam

- Pacemaker küme yapılandırmasını silmek için düğümlerden herhangi birinde aşağıdaki komutu girin:

```
rdqmadm -u
```

## İlgili başvurular

[rdqmadm \(eşlenmiş veri kuyruğu yöneticisi kümesini yönet\)](#)

Linux

*HA RDQM yaratılması*

Yüksek kullanılabilirlikli bir eşlenmiş veri kuyruğu yöneticisi (RDQM) yaratmak için **crtmqm** komutunu kullanırsınız.

## Bu görev hakkında

mqm kullanıcısı sudo kullanabiliyorsa, mqm grubundaki bir kullanıcı olarak yüksek kullanılabilirlikli eşlenmiş bir veri kuyruğu yöneticisi (RDQM) oluşturabilirsiniz. Kullanıcı, parola olmadan her bir düğüme SSH 'yi de yapabiliyorsa, RDQM' yi üç düğümden de yaratmak için tek bir düğümden RDQM yaratma komutunu çalıştırmanız gerekir. Aksi takdirde, bir RDQM oluşturmak için root olmanız ve üç düğümden de komut çalıştırmanız gerekir.

**Not:** Bir HA grubunda 129 kuyruk yöneticisinin mutlak sınırı vardır. Bundan daha fazla yaratma girişiminde bulunursanız, girişim başarısız olur. Uygulamada, bir HA grubuna 50 'den fazla kuyruk yöneticisi eklenmesi zamanaşımı sorunlarıyla karşılaşabilir.

Aşağıdaki noktalar, kuyruk yöneticisi dosya sisteminin boyutlandırılmasına ilişkin bazı kılavuzlar sağlar:

1. Bir RDQM kuyruk yöneticisi yaratılırken, kuyruk yöneticisi verilerini ve günlüklerini saklamak için bir dosya sistemi ayrılır. Kuyruk yöneticisinin devam eden etkinliği günlüklerine kaydedebilmesi ve uygulama iletilerini kuyruklara saklayabilmesi için bu dosya sisteminin uygun şekilde boyutlandırılması önemlidir. Dosya sisteminin boyutlandırırken en yoğun ileti sistemi gereksinimlerini, gelecekteki iş yükü artışını ve iletilerin kuyruklarda birikmesine neden olabilecek uygulama kesintilerini göz önünde bulundurun. Kuyruk yöneticisi kurtarma günlüğünün boyutunun hesaplanmasına ilişkin yönergeler için bkz. [“Günlük dosya sistemimi ne kadar büyük yapmalıyım?” sayfa 638](#). Uygulama iletileri için

depolama gereksinimleri hesaplanırken, iletilerin boyutu ve sayısı ve bunların MQMD üstbilgisi ve sahip oldukları ileti özellikleri dikkate alınmalı.

2. RDQM kuyruk yöneticisi dosya sistemleri dinamik olarak yeniden boyutlandırılmıyor. Bu gerekiyorsa, RDQM kuyruk yöneticisini daha büyük bir dosya sistemiyle yedeklemeniz ve geri yüklemeniz gerekir, bkz. “HA RDQM kuyruk yöneticisi için dosya sistemini yeniden boyutlandırma” sayfa 566.
3. MAXDEPTH ve MAXFSIZE gibi yerel kuyruk özniteliklerini kullanarak diskteki tek tek kuyrukların boyutunu sınırlayabilirsiniz. Bkz. [IBM MQ kuyruk dosyalarının değiştirilmesi](#).
4. Devam eden disk kullanımınızı izlemeniz ve dosya sistemi kullanımı kritik duruma gelmeden önce disk kullanımı artarsa uygun şekilde yanıt vermeniz gerekir. Dosya sistemi kullanımı, platform/işletim sistemi yetenekleri kullanılarak ya da [Sistem konularında yayınlanan ölçümler](#) başlıklı konuda açıklanan IBM MQ sistem konularına abone olunarak izlenebilir.

## Yordam

- mqm grubunda kullanıcı olarak bir RDQM yaratmak için:
  - a) mqm kullanıcısının komutları çalıştırmak için **sudo** komutunu kullanabildiğinden ve parola olmadan SSH kullanarak her bir sunucuya bağlanabildiğinden emin olun.
  - b) Aşağıdaki komutu girin:

```
crtmqm -sx [-fs FilesystemSize] qmname
```

Burada *qmname* , eşlenen veri kuyruğu yöneticisinin adıdır. İsteğe bağlı olarak, kuyruk yöneticisi için dosya sistemi boyutunu (yani, drbdpool birim grubunda oluşturulan mantıksal birimin boyutunu) belirtebilirsiniz.

Komut, mqm kullanıcısı olarak kümedeki diğer düğümlere bağlanmak için SSH ' yi kullanmayı dener. Bağlantı başarılı olursa, düğümlerde kuyruk yöneticisinin ikincil yönetim ortamları yaratılır. Ters durumda, ikincil yönetim ortamlarını yaratmanız ve daha sonra, **crtmqm -sx** komutunu çalıştırmanız gerekir (kullanıcı root için açıklandığı gibi).

- rootkullanıcısı olarak bir RDQM yaratmak için:
  - a) RDQM ' nin ikincil eşgörünümlerini barındıracak düğümlerin her birinde aşağıdaki komutu girin:

```
crtmqm -sxs [-fs FilesystemSize] qmname
```

Burada *qmname* , eşlenen veri kuyruğu yöneticisinin adıdır. İsteğe bağlı olarak, kuyruk yöneticisi için dosya sistemi boyutunu (yani, drbdpool birim grubunda oluşturulan mantıksal birimin boyutunu) belirtebilirsiniz. HA grubundaki üç düğümün hepsinde RDQM için aynı dosya sistemi boyutunu belirtmeniz gerekir. Boyut, GB olarak belirtilen sayısal bir değerdir. MB cinsinden bir değeri, M karakterinin izlediği değeri girerek belirtebilirsiniz.

Komut, RDQM ' nin ikincil bir eşgörünümünü yaratır.

- b) Geri kalan düğümde şu komutu girin:

```
crtmqm -sx [-fs FilesystemSize] qmname
```

Burada *qmname* , eşlenen veri kuyruğu yöneticisinin adıdır. İsteğe bağlı olarak, kuyruk yöneticisi için dosya sistemi boyutunu belirtebilirsiniz. Boyut, GB olarak belirtilen sayısal bir değerdir. MB cinsinden bir değeri, M karakterinin izlediği değeri girerek belirtebilirsiniz.

Komut, kuyruk yöneticisinin ikincil eşgörünümünün diğer iki düğümde olup olmadığını belirler. İkincil ikilemler varsa, komut birincil kuyruk yöneticisini yaratır ve başlatır. İkincil düğümler yoksa, düğümlerin her birinde **crtmqm -sxs** komutunu çalıştırmanız istenir.

DataPath (**-md**) ve LogPath (**-ld**) bağımsız değişkenlerinin yanı sıra, standart Linux kuyruk yöneticisi oluşturmak için geçerli olan tüm bağımsız değişkenler, eşlenmiş birincil veri kuyruğu yöneticisi için de geçerlidir.

**Not:** Bir RDQM oluşturduğunuzda, eşleme bağlantısı için 7000 'in üzerindeki bir sonraki boş kapı numarası ayrılır. Seçilen kapının başka bir uygulama tarafından kullanıldığı saptanırsa, **crtmqm** komutu

AMQ6543 hatasıyla başarısız olur ve bu kapı bir dışlama listesine eklenir. Kuyruk yöneticisinin ikincil yönetim ortamlarını silmeli ve **crtmqm** komutunu yeniden çalıştırmalısınız.

## İlgili başvurular

[crtmqm](#)

**Linux** **HA RDQM ' i silme**

Yüksek kullanılabilirlikli eşlenmiş bir veri kuyruğu yöneticisini (RDQM) silmek için **dltmqm** komutunu kullanırsınız.

## Bu görev hakkında

RDQM 'in birincil düğümünde RDQM' yi silmek için komutu çalıştırmanız gerekir. Önce RDQM sona erdirilmelidir. Kullanıcı gerekli sudo ayrıcalıklarına sahipse, komutu mqm kullanıcısı olarak çalıştırabilirsiniz. Ters durumda, komutu kök olarak çalıştırmanız gerekir. Birincil kuyruk yöneticisiyle ilişkili kaynaklar silindikten sonra, komut diğer düğümlere bağlanmak için ssh kullanarak ikincil kuyruk yöneticilerini silmeyi dener. Bu silme başarısız olursa, işlemi tamamlamak için diğer düğümlerde dltmqm komutunu el ile çalıştırmanız gerekir. İkincil bir düğümde, birincil kuyruk yöneticisi önceden silinmediyse komut başarısız olur.

## Yordam

- Bir RDQM ' yi silmek için şu komutu girin:

```
dltmqm RDQM_name
```

## İlgili başvurular

[dltmqm](#)

**Linux** **MQ Adv.** *Bir kuyruk yöneticisinin HA RDQM kuyruk yöneticisi olacak şekilde geçirilmesi*

Var olan bir kuyruk yöneticisini, kalıcı verilerini yedekleyerek ve daha sonra verileri aynı ada sahip yeni yaratılan bir RDQM kuyruk yöneticisine geri yükleyerek, yüksek kullanılabilirlikli (HA) eşlenmiş bir veri kuyruğu yöneticisi (RDQM) olacak şekilde geçirebilirsiniz.

## Bu görev hakkında

HA Replicated Data Queue Manager (HA Eşlenmiş Veri Kuyruğu Yöneticileri) için özel olarak ayrılmış bir mantıksal birim (dosya sistemi) ve disk eşleme ve HA denetimi yapılandırması gerekir. Bu bileşenler yalnızca yeni bir kuyruk yöneticisi yaratıldığında yapılandırılır. Var olan bir kuyruk yöneticisi, kalıcı verilerini yedekleyerek RDQM ' yi kullanmak üzere geçirilebilir, daha sonra verileri aynı ada sahip yeni yaratılan bir RDQM kuyruk yöneticisine geri yükleyebilir. Bu yordam, yedekleme yaratıldığı sırada kuyruk yöneticisi yapılanışını, durumunu ve kalıcı iletileri korur.

**Not:** You can only migrate a queue manager from a version of IBM MQ that is the same as or lower than the version where RDQM is installed. İşletim sistemi ve mimari de aynı olmalıdır. Ters durumda, hedef altyapınızda yeni bir kuyruk yöneticisi yaratmanız gerekir; bkz. [Kuyruk yöneticisinin farklı bir işlem sistemine taşınması](#).

Bir kuyruk yöneticisini geçirmeden önce aşağıdaki koşulları karşılamamız gerekir:

- Yüksek kullanılabilirlik gereksinimlerinizi değerlendirin ve bkz. [“RDQM yüksek kullanılabilirlik düzeyi” sayfa 552](#).
- Kuyruk yöneticisine bağlanan uygulamaları ve kuyruk yöneticilerini gözden geçirin. Bağlantıları kuyruk yöneticisinin çalıştığı RDQM düğümüne yönlendirmek için gereken değişiklikleri göz önünde bulundurun. Örneğin, RDQM yüksek kullanılabilirliğini yapılandırırsanız, kayan bir IP adresi kullanmayı düşünebilirsiniz, bkz. [“Kayan IP adresi oluşturma ve silme” sayfa 570](#).
- Seçtiğiniz yapılandırma için var olan RDQM düğümlerini yetkilendirin ya da tanımlayın. RDQM ' ye ilişkin sistem gereksinimleri hakkında bilgi için bkz. [“RDQM HA çözümü için gereksinimler” sayfa 554](#).

- Her düğümde RDQM özelliğini içeren IBM MQ Advanced ürününü kurun.
- RDQM HA grubu yapılandırmasını yapılandırma, bkz. [“Pacemaker kümesinin tanımlanması \(HA grubu\)” sayfa 557](#).
- İsteğe bağlı olarak, daha sonra silinebilecek bir test kuyruğu yöneticisini kullanarak RDQM yapılandırmasını doğrulayın. Kuyruk yöneticisini geçirmeden önce sorunları saptamak ve çözmek için yapılandırmanın sınanması önerilir.
- Kuyruk yöneticisi için güvenlik yapılandırmasını gözden geçirin ve ardından her bir RDQM düğümünde gerekli yerel kullanıcıları ve grupları eşin.
- API çıkışlarının, kanal çıkışlarının ya da veri dönüştürme çıkışlarının kullanılıp kullanılmadığını belirlemek için kuyruk yöneticisini ve kanal yapılandırmasını gözden geçirin. Her bir RDQM düğümünde gerekli çıkışları kurun.
- Tanımlanmış kuyruk yöneticisi hizmetlerini gözden geçirin, ardından her bir RDQM düğümünde gerekli süreçleri kurun ve yapılandırın.

## Yordam

### 1. Var olan kuyruk yöneticisini yedekle:

- a) Bir bekleme sona erdirme komutu `endmqm -wya` da hemen sona erdirme komutu `endmqm -ivererek`, var olan kuyruk yöneticisini durdurun. Bu adım, yedekteki verilerin tutarlı olmasını sağlamak için önemlidir.
- b) IBM MQ yapılandırma kütüğünü (`mqc.ini`) görüntüleyerek kuyruk yöneticisi verileri dizininin yerini saptayın. Linux üzerinde bu dosya, `/var/mqm` dizininde bulunur. `mqc.ini` hakkında daha fazla bilgi için bkz. [“IBM MQ yapılandırma dosyası, mqc.ini” sayfa 85](#).  
Dosyada kuyruk yöneticisine ilişkin `QueueManager` kısmı bulun. Bölüm `DataPath` adlı bir anahtar içeriyorsa, değeri kuyruk yöneticisi veri dizinidir. Anahtar yoksa, kuyruk yöneticisi veri dizini `Prefix` ve `Directory` anahtarlarının değerleri kullanılarak belirlenebilir. Kuyruk yöneticisi veri dizini, `prefix/qmgrs/directory` biçiminde bu değerlerin bir birleşimidir. `QueueManager` kısmı hakkında daha fazla bilgi için bkz. [“mqc.ini dosyasının QueueManager kısmı” sayfa 95](#).
- c) Kuyruk yöneticisi veri dizininin bir yedeğini oluşturun. Linux' da bunu **tar** komutunu kullanarak yapabilirsiniz. Örneğin, bir kuyruk yöneticisine ilişkin veri dizinini yedeklemek için aşağıdaki komutu kullanabilirsiniz. Komutun tek nokta (nokta) olan son deęiřtirgesine dikkat edin:

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- d) IBM MQ Kuyruk yöneticisi yapılandırma kütüğünü `qm.ini` görüntüleyerek kuyruk yöneticisi günlük dizininin yerini saptayın. Bu dosya, kuyruk yöneticisi veri dizininde bulunur. Dosya hakkında daha fazla bilgi için bkz. [“Kuyruk yöneticisi yapılandırma dosyaları, qm.ini” sayfa 97](#).  
Kuyruk yöneticisi günlük dizini, `Log` kısmı içindeki `LogPath` anahtarının değeri olarak tanımlanır. Kıta hakkında bilgi için bkz. [“qm.ini dosyasının günlük kısmı” sayfa 129](#).
- e) Kuyruk yöneticisi günlük dizininin bir yedeğini yaratın. Linux üzerinde, `tar` komutunu kullanarak bunu yapabilirsiniz. Örneğin, bir kuyruk yöneticisine ilişkin günlük dizinini yedeklemek için aşağıdaki komutu kullanabilirsiniz. Komutun tek nokta (nokta) olan son deęiřtirgesine dikkat edin:

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- f) Kuyruk yöneticisi veri dizininde bulunmuyorsa, kuyruk yöneticisi tarafından kullanılan sertifika havuzlarının yedeğini oluşturun. Hem anahtar veritabanı kütüğünün, hem de parola saklama kütüğünün yedeklendiğini doğrulayın. Kuyruk yöneticisi anahtar havuzu hakkında bilgi için bkz. [SSL/TLS anahtar havuzu](#) ve [Kuyruk yöneticisi için anahtar havuzunun bulunması](#). Kuyruk yöneticisi AMS Message Channel Agent (MCA) müdahalesi kullanacak şekilde yapılandırıldıysa, AMS anahtar deposunun bulunmasıyla ilgili bilgi için [Message Channel Agent \(MCA\) interception](#) başlıklı konuya bakın.
- g) Var olan kuyruk yöneticisi artık gerekli olmadığından silinebilir. Ancak, mümkünse, hedef sistemde başarıyla geri yüklendikten sonra varolan kuyruk yöneticisini silmeniz gerekir. Silme işleminin



ertelenmesi, geçiş işlemi başarıyla tamamlanmazsa kuyruk yöneticisinin yeniden başlatılabilmesini sağlar.

**Not:** Var olan kuyruk yöneticisinin silinmesini ertelerseniz, yeniden başlatmayın. Geçiş sırasında yapılanışında ya da durumunda yapılacak başka değişiklikler kaybedildiği için kuyruk yöneticisinin sona erdirilmeye devam etmesi önemlidir.

## 2. Birincil RDQM düğümünü hazırla:

- Yedeklediğiniz kuyruk yöneticisiyle aynı ada sahip yeni bir RDQM kuyruk yöneticisi yaratın. **crtmqm** tarafından RDQM kuyruk yöneticisi için ayrılan dosya sisteminin, var olan kuyruk yöneticisine ilişkin verileri, birincil günlükleri ve ikincil günlükleri ve gelecekteki genişletme için ek alan içerecek kadar büyük olduğundan emin olun. Bir RDQM kuyruk yöneticisinin nasıl yaratılacağı hakkında bilgi için bkz. “HA RDQM yaratılması” sayfa 561.
- Kuyruk yöneticisi için birincil RDQM düğümünü belirleyin. Birincil düğümün nasıl belirleneceğine ilişkin bilgi için bkz. [rdqmstatus \(RDQM durumunun görüntülenmesi\)](#).
- Birincil RDQM düğümünde RDQM kuyruk yöneticisi başlatıldıysa, `endmqm -w` ya da `endmqm -i` komutunu kullanarak durdurun.
- Birincil RDQM düğümünde, RDQM kuyruk yöneticisine ilişkin veri ve günlük dizinlerinin konumunu belirleyin ( 1b ve 1dadımlarında açıklanan yöntemleri kullanın).
- Birincil RDQM düğümünde, RDQM kuyruk yöneticisi verilerinin ve günlük dizinlerinin içeriğini silin, ancak dizinleri silmeyin.

## 3. Birincil RDQM düğümünde kuyruk yöneticisini geri yükleyin:

- Kuyruk yöneticisi verileri ve günlük dizinlerinin yedeklerini birincil RDQM düğümüne ve kuyruk yöneticisi tarafından kullanılan sertifika havuzlarının ayrı yedeklerine kopyalayın.
- Yeni RDQM kuyruk yöneticisi için kuyruk yöneticisi veri dizininin yedeğini boş veri dizinine geri yükleyip dosya sahipliğinin ve izinlerinin korunduğundan emin olun. Yedekleme, 1c adımıdaki örnek `tar` komutu kullanılarak yaratıldıysa, kök kullanıcı tarafından geri yüklemek için aşağıdaki komut kullanılabilir:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- Yeni RDQM kuyruk yöneticisi için kuyruk yöneticisi günlük dizininin yedeğini boş günlük dizinine geri yükleyip dosya sahipliğinin ve izinlerinin korunduğundan emin olun. Yedekleme, 1e adımıdaki örnek `tar` komutu kullanılarak yaratıldıysa, kök kullanıcı tarafından geri yüklemek için aşağıdaki komut kullanılabilir:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- RDQM kuyruk yöneticisine ilişkin veri dizininde geri yüklenen kuyruk yöneticisi yapılandırma dosyasını ( `qm.ini`) düzenleyin. Log kısmı içindeki `LogPath` anahtarının değerini, RDQM kuyruk yöneticisine ilişkin günlük dizinini belirtecek şekilde güncelleyin.

Yapılandırma dosyasında tanımlanan diğer dosya yollarını gözden geçirin ve gerekirse bunları güncelleyin. Örneğin, aşağıdaki yolları güncellemeniz gerekebilir:

- Tanımlama ileti hizmetleri tarafından oluşturulan hata günlüğü dosyalarının yolu.
- Kuyruk yöneticisinin gerektirdiği çıkışların yolu.
- Kuyruk yöneticisi bir XA hareket eşgüdümçüye, anahtar yükleme dosyalarının yolu.

- Kuyruk yöneticisi AMS Message Channel Agent (MCA) müdahalesi kullanacak şekilde yapılandırıldıysa, AMS anahtar deposunu yeni RDQM kuruluşuna kopyalayın ve yapılandırmayı gözden geçirin ve güncelleyin. Anahtar deposu her RDQM düğümünde kullanılabilir olmalıdır; bu nedenle, kuyruk yöneticisi için eşlenen dosya sisteminde bulunmuyorsa, bunun yerine her bir düğümüne kopyalanmalıdır. Daha fazla bilgi için bkz. [Message Channel Agent \(MCA\) interception](#).
- Kuyruk yöneticisinin **dspmq** komutu tarafından görüntülendiğini ve durumunun sona erdirildi olarak bildirildiğini doğrulayın. Aşağıdaki örnek, bir RDQM HA kuyruk yöneticisine ilişkin örnek çıktıyı göstermektedir:

```
$ dspmq -o status -o ha
QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

- g) Kuyruk yöneticisine ilişkin durumu görüntülemek için **rdqmstatus** komutunu kullanarak, geri yüklenen kuyruk yöneticisi verilerinin ikincil RDQM düğümlerine eşlendiğini doğrulayın. HA durumu her düğümde Normal olarak bildirilmelidir. Aşağıdaki örnek, bir RDQM HA kuyruk yöneticisine ilişkin örnek çıktısını göstermektedir:

```
$ rdqmstatus -m QM1
Node:                               mqhavam10-adm
Queue manager status:               Ended normally
Queue manager file system:          50MB used, 0.2GB allocated [42%]
HA role:                             Primary
HA status:                           Normal
HA control:                           Disabled
HA current location:                 This node
HA preferred location:                This node
HA floating IP interface:             None
HA floating IP address:               None

Node:                               mqhavam11-adm
HA status:                            Normal

Node:                               mqhavam12-adm
HA status:                            Normal
```

- h) Birincil RDQM düğümünde kuyruk yöneticisini başlatın.
- i) Kuyruk yöneticisine bağlanın ve kuyruk yöneticisi sertifika havuzunun yeni konumunu belirtmek için SSLKEYR kuyruk yöneticisi özneliğinin değerini güncelleyin. Varsayılan olarak, bu özneliğin değeri *queue\_manager\_data\_directory/ssl/key* olarak ayarlanır. Sertifika havuzu, her RDQM düğümünde aynı konumda bulunmalıdır. Havuz, kuyruk yöneticisine ilişkin eşlenmiş dosya sisteminde bulunmuyorsa, bunun yerine her düğüme kopyalanması gerekir.
- j) Kuyruk yöneticisi için IBM MQ nesne tanımlamalarını gözden geçirin ve aşağıdaki nesnelere de içinde olmak üzere, değiştirilen ağ ayarlarına, IBM MQ kuruluş dizinine ya da kuyruk yöneticisi veri dizinine başvuran nesne özneliklerinin değerini güncelleyin:
- Dinleyiciler tarafından kullanılan yerel IP adresleri (IPADDR özneliği).
  - Kanallar tarafından kullanılan yerel IP adresleri (LOCLADDR özneliği).
  - Küme alıcı kanalları için tanımlanan yerel IP adresleri (CONNNAME özneliği).
  - İletişim bilgileri nesnelere için tanımlanan yerel IP adresleri (GRPADDR özneliği).
  - Süreç ve hizmet nesnesi tanımlamaları için tanımlanan sistem yolları.
- k) Değişikliklerin etkili olmasını sağlamak için kuyruk yöneticisini durdurun ve yeniden başlatın.
- l) Uzak kuyruk yöneticileri için 3j adımını ve geçirilen kuyruk yöneticisine (aşağıdakiler de içinde olmak üzere) bağlanan uygulamalara ilişkin eşdeğer ayarları yineleyin:
- Kanal bağlantısı adları (CONNNAME özneliği).
  - Kuyruk yöneticisinden gelen bağlantıları IP adresine ya da anasistem adına göre kısıtlayan kanal kimlik doğrulama kuralları.
  - İstemci kanal tanımlama çizelgeleri (CCDT), etki alanı adı ayarları (DNS), ağ yöneltme ya da eşdeğer bağlantı bilgileri.
- m) Gerekli yapılandırmanın başarıyla oluşturulduğundan emin olmak için kuyruk yöneticisinin her bir RDQM düğümüne yönetilen bir hata durumunda yedek sisteme geçiş işlemi gerçekleştirin, bkz. ["RDQM için Tercih Edilen Konumun Ayarlanması"](#) sayfa 569.

#### *HA RDQM kuyruk yöneticisi için dosya sistemini yeniden boyutlandırma*

Var olan bir yüksek kullanılabilirlikli (HA) eşlenmiş veri kuyruğu yöneticisine (RDQM) ilişkin dosya sistemini yeniden boyutlandırmak için, kalıcı verilerini yedekleyin ve yeni yaratılan RDQM kuyruk yöneticisine geri yükleyin; bu kuyruk yöneticisi aynı adı, ancak dosya sistemi farklı boyutta.

## Bu görev hakkında

HA eşlenmiş veri kuyruğu yöneticileri için özel olarak ayrılmış bir mantıksal birim (dosya sistemi) ve disk eşleme ve HA denetimi yapılandırması gerekir. Bu bileşenler yalnızca yeni bir kuyruk yöneticisi yaratıldığında yapılandırılır. Dosya sistemi yaratıldıktan sonra yeniden boyutlandırılmaz; her düğümde aynı büyüklüğe sahip olmalıdır. Var olan bir eşlenmiş veri kuyruğu yöneticisine (RDQM) ilişkin dosya sistemini yeniden boyutlandırmak için, kalıcı verilerini yedekleyebilir ve daha sonra, aynı ada sahip, ancak farklı bir dosya sistemine sahip yeni yaratılan bir RDQM kuyruk yöneticisine geri yükleyebilirsiniz. Bu yordam, yedekleme yaratıldığı sırada kuyruk yöneticisi yapılanışını, durumunu ve kalıcı iletileri korur.

## Yordam

1. Birincil RDQM düğümünde var olan RDQM kuyruk yöneticisini yedekleyin:

- Kuyruk yöneticisi için birincil RDQM düğümünü belirleyin. Birincil düğümün nasıl belirleneceğine ilişkin bilgi için bkz. [rdqmstatus \(RDQM durumunun görüntülenmesi\)](#).
- Birincil RDQM düğümünde RDQM kuyruk yöneticisi başlatıldıysa, **endmqm -w** ya da **endmqm -i** komutunu kullanarak durdurun.
- IBM MQ yapılanış kütüğünü ( `mqs.ini`) görüntüleyerek kuyruk yöneticisi verileri dizininin yerini saptayın. Linux'üzünde bu dosya, `/var/mqm` dizininde bulunur. `mqs.ini` hakkında daha fazla bilgi için bkz. ["IBM MQ yapılandırma dosyası, mqs.ini" sayfa 85](#).

Dosyada kuyruk yöneticisine ilişkin `QueueManager` kısmı bulun. Kuyruk yöneticisi veri dizini, `DataPath` adlı anahtarın değeridir. `QueueManager` kısmı hakkında daha fazla bilgi için bkz. ["mqs.ini dosyasının QueueManager kısmı" sayfa 95](#).

- Kuyruk yöneticisi veri dizininin bir yedeğini oluşturun. Linux' da bunu **tar** komutunu kullanarak yapabilirsiniz. Örneğin, bir kuyruk yöneticisine ilişkin veri dizinini yedeklemek için aşağıdaki komutu kullanabilirsiniz. Komutun tek bir nokta (.) karakteri olan son deęiştirgesine dikkat edin:

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- IBM MQ Kuyruk yöneticisi yapılanış kütüğünü `qm.ini` görüntüleyerek kuyruk yöneticisi günlük dizininin yerini saptayın. Bu dosya, kuyruk yöneticisi veri dizininde bulunur. Dosya hakkında daha fazla bilgi için bkz. ["Kuyruk yöneticisi yapılanış dosyaları, qm.ini" sayfa 97](#).

Kuyruk yöneticisi günlük dizini, Günlük kısmı içindeki `LogPath` anahtarının değeri olarak tanımlanır. Kıta hakkında bilgi için bkz. ["qm.ini dosyasının günlük kısmı" sayfa 129](#).

- Kuyruk yöneticisi günlük dizininin bir yedeğini yaratın. Linux' da bunu **tar** komutunu kullanarak yapabilirsiniz. Örneğin, bir kuyruk yöneticisine ilişkin günlük dizinini yedeklemek için aşağıdaki komutu kullanabilirsiniz. Komutun tek bir nokta (.) karakteri olan son deęiştirgesine dikkat edin:

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- Var olan RDQM kuyruk yöneticisini silin.

2. Kuyruk yöneticisini gereken büyüklükte bir dosya sistemi ile geri yükleyin:

- Yedeklediğiniz kuyruk yöneticisiyle aynı ada sahip yeni bir RDQM kuyruk yöneticisi yaratın. **crtmqm** tarafından RDQM kuyruk yöneticisi için ayrılan dosya sisteminin gerekli boyut olduğundan ve var olan kuyruk yöneticisine ilişkin verileri, birincil günlükleri ve ikincil günlükleri içerecek kadar büyük olduğundan ve gelecekteki genişletme için ek alan bulunduğundan emin olun. Bir RDQM kuyruk yöneticisinin nasıl yaratılacağı hakkında bilgi için bkz. ["HA RDQM yaratılması" sayfa 561](#).
- Kuyruk yöneticisi için birincil RDQM düğümünü belirleyin. Birincil düğümün nasıl belirleneceğine ilişkin bilgi için bkz. [rdqmstatus \(RDQM durumunun görüntülenmesi\)](#).
- Birincil RDQM düğümünde RDQM kuyruk yöneticisi başlatıldıysa, **endmqm -w** ya da **endmqm -i** komutunu kullanarak durdurun.
- Birincil RDQM düğümünde, RDQM kuyruk yöneticisine ilişkin veri ve günlük dizinlerinin yeni konumunu belirleyin ( `1c` ve `1ead` imlarında açıklanan yöntemleri kullanın).

- e) Birincil RDQM düğümünde, RDQM kuyruk yöneticisi verilerinin ve günlük dizinlerinin içeriğini silin, ancak dizinleri silmeyin.
- f) Birincil RDQM düğümünde, kuyruk yöneticisi veri dizininin yedeğini yeni RDQM kuyruk yöneticisi için boş veri dizinine geri yükleyip dosya sahipliğinin ve izinlerinin korunduğundan emin olun. Yedekleme, 1d adımıdaki örnek **tar** komutu kullanılarak yaratıldıysa, kök kullanıcı tarafından geri yüklemek için aşağıdaki komut kullanılabilir:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- g) Birincil RDQM düğümünde, kuyruk yöneticisi günlük dizininin yedeğini yeni RDQM kuyruk yöneticisi için boş günlük dizinine geri yükleyip dosya sahipliğinin ve izinlerinin korunduğundan emin olun. Yedekleme, 1f adımıdaki örnek **tar** komutu kullanılarak yaratıldıysa, kök kullanıcı tarafından geri yüklemek için aşağıdaki komut kullanılabilir:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- h) Birincil RDQM düğümünde, yeni RDQM kuyruk yöneticisinin veri dizininde geri yüklenen kuyruk yöneticisi yapılandırma dosyasını ( `qm.ini` ) düzenleyin. Adım 2d' de belirlediğiniz yeni RDQM kuyruk yöneticisine ilişkin günlük dizinini belirtmek için Log kısmına ilişkin LogPath anahtarının değerini güncelleyin. Yapılandırma dosyasında tanımlanan diğer dosya yollarını gözden geçirin ve gerekirse bunları güncelleyin. Örneğin, aşağıdaki yolları güncellemeniz gerekebilir:
- Tanılama ileti hizmetleri tarafından oluşturulan hata günlüğü dosyalarının yolu.
  - Kuyruk yöneticisinin gerektirdiği çıkışların yolu.
  - Kuyruk yöneticisi bir XA hareket eşgüdümçüye, anahtar yükleme dosyalarının yolu.
- i) Kuyruk yöneticisinin **dspmqr** komutu tarafından görüntülendiğini ve durumunun sona erdirildi olarak bildirildiğini doğrulayın. Aşağıdaki örnek, bir RDQM HA kuyruk yöneticisine ilişkin örnek çıktıyı göstermektedir:

```
$ dspmqr -o status -o ha
QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

- j) Kuyruk yöneticisine ilişkin durumu görüntülemek için **rdqmstatus** komutunu kullanarak, geri yüklenen kuyruk yöneticisi verilerinin ikincil RDQM düğümlerine eşlendiğini doğrulayın. HA durumu her düğümde Normal olarak bildirilmelidir. Aşağıdaki örnek, bir RDQM HA kuyruk yöneticisine ilişkin örnek çıktıyı göstermektedir:

```
$ rdqmstatus -m QM1
Node: mqhavam10-adm
Queue manager status:      Ended normally
Queue manager file system: 50MB used, 0.2GB
allocated [42%]
HA role:                   Primary
HA status:                 Normal
HA control:                Disabled
HA current location:       This node
HA preferred location:     This node
HA floating IP interface:  None
HA floating IP address:    None
Node:                      mqhavam11-adm
HA status:                 Normal
Node:                      mqhavam12-adm
HA status:                 Normal
```

- k) Birincil RDQM düğümünde kuyruk yöneticisini başlatın.
- l) Gerekli yapılandırmanın başarıyla oluşturulduğundan emin olmak için kuyruk yöneticisinin her bir RDQM düğümüne yönetilen bir hata durumunda yedek sisteme geçiş işlemi gerçekleştirin, bkz. [“RDQM için Tercih Edilen Konumun Ayarlanması” sayfa 569.](#)

#### *Kalıcı uygulama durumu saklanıyor*

Diğer kuyruk yöneticisi verileriyle birlikte uygulamalarla ilgili kalıcı durum bilgilerini saklayabilirsiniz.

Her IBM MQ kuyruk yöneticisinin, hem kuyruk verilerini hem de kurtarma günlüğünü içeren, kalıcı durumu için özel olarak ayrılmış bir dosya sistemi vardır. Bir RDQM yapılandırmasında dosya sistemi,

Linux sistemleri (düğümler) arasında eşlenen bir mantıksal birim tarafından desteklenir. Dosya sistemi, uygulamalarınıza ilişkin kalıcı durum bilgilerini saklamak için kullanabileceğiniz bir `userdata` dizini içerir. Bu nedenle, eşlenmiş bir veri kuyruğu yöneticisi RDQM yapılandırmanızdaki başka bir düğümden çalışmak üzere hareket ettiğinde, kuyruk yöneticisi bağlamının yanı sıra uygulama bağlamını da kullanabilirsiniz. Bkz. [Directory content on Unix and Linux Systems](#).

Uygulama durumunu `userdata` dizininde saklamayı seçerseniz, bu yere yazılan verilerin kuyruk yöneticisine ayrılan kullanılabilir disk alanını kullanabileceğini bilmeniz gerekir. Kuyruk yöneticisinin kuyruk verilerini, günlüklerini ve diğer kalıcı durum bilgilerini yazması için yeterli disk yeri kaldığından emin olmanız gerekir.

`userdata` dizini `mqm` kullanıcı ve grup iyeliğine sahiptir ve kullanıcıların IBM MQ yönetici grubunda (`mqm`) olmalarına gerek kalmadan bu dizine erişebilmeleri için herkes tarafından okunabilir. `userdata` dizininin izinlerini değiştiremezsiniz, ancak gerek duyduğunuz sahiplik ve izinlerle içerik oluşturabilirsiniz.

Bir RDQM kuyruk yöneticisi hata durumunda yedek sisteme geçiş sırasında kuyruk yöneticisi sona erdirilir ve dosya sistemi yürürlükteki RDQM düğümünde kaldırılır. Daha sonra dosya sistemi sisteme bağlanır ve kuyruk yöneticisi RDQM yapılandırmasındaki başka bir düğümden yeniden başlatılır. Bir işlemin dosyalarından biri için açık bir tanıtıcısı varsa, dosya sistemi kaldırılamaz. Bir kuyruk yöneticisinin hata durumunda yedek sisteme geçiş işleminin tamamlanabilmesini sağlamak için, kuyruk yöneticisi dosya sistemi bağlantısını kaldırılamazsa, açık dosya tanıtıcısı olan işlemlere bir SIGTERM sinyali gönderilir ve ardından açık tutamaçlar serbest bırakılmazsa, SIGKILL işareti gönderilir. Uygulamalarınız, SIGTERM 'ye doğru yanıt verecek şekilde tasarlanmalıdır. Uygulamalar ya da işlemler kuyruk yöneticisi hizmeti olarak yapılandırıldıysa, yönetilen hata durumunda yedek sisteme geçiş sırasında, dosya sistemi kaldırılmadan önce kuyruk yöneticisi kapatılırken sona erdirilebilirler. Bir uygulama ya da işlem kuyruk yöneticisi hizmeti olarak yapılandırılmadıysa ya da çekirdek kaybı gibi yönetilmeyen bir hata durumunda yedek sisteme geçiş gerçekleşirse, dosya sistemini serbest bırakmak için sinyaller gönderilebilir.

## Linux RDQM için Tercih Edilen Konumun Ayarlanması

Eşlenmiş bir veri kuyruğu yöneticisi (RDQM) için Tercih Edilen Konum, bu düğüm kullanılabiliriyorsa RDQM 'in çalışacağı düğümü tanımlar.

### Bu görev hakkında

Tercih Edilen Konum, HA grubu olağan durumdayken (tüm düğümler ve bağlantılar) Pacemaker 'in kuyruk yöneticisini çalıştırması gereken düğümün adıdır. Tercih Edilen Konum, kuyruk yöneticisi yaratıldığında birincil düğümün adıyla ilk kullanıma hazırlanır. Üç düğümden herhangi birinde Tercih Edilen Konumu ayarlamak için komutları çalıştırabilirsiniz. Hem `mqm` hem de `haclient` gruplarına ait bir kullanıcı olmanız gerekir.

### Yordam

- Yerel ya da belirtilen düğümü, adı belirtilen kuyruk yöneticisi için Tercih Edilen Konum olarak atamak için şu komutu girin:

```
rdqmadm -p -m qmname [ -n nodename[,nodename ]
```

Burada `qmname` , tercih edilen konumu belirtmekte olduğunuz RDQM 'nin adı ve *düğüm adı* isteğe bağlı olarak tercih edilen düğümün adıdır.

HA grubu olağan durumdaysa ve Tercih Edilen Konum yürürlükteki birincil düğüm değilse, kuyruk yöneticisi durdurulur ve yeni Tercih Edilen Konum 'da yeniden başlatılır. Tercih edilen yer için ikinci bir tercih atamak üzere iki düğüm adının virgülle ayrılmış bir listesini belirtebilirsiniz.

- Kuyruk yöneticisinin geri yüklendiğinde bir düğüme otomatik olarak dönmemesi için Tercih Edilen Konumu temizlemek üzere şu komutu girin:

```
rdqmadm -p -m qmname -d
```

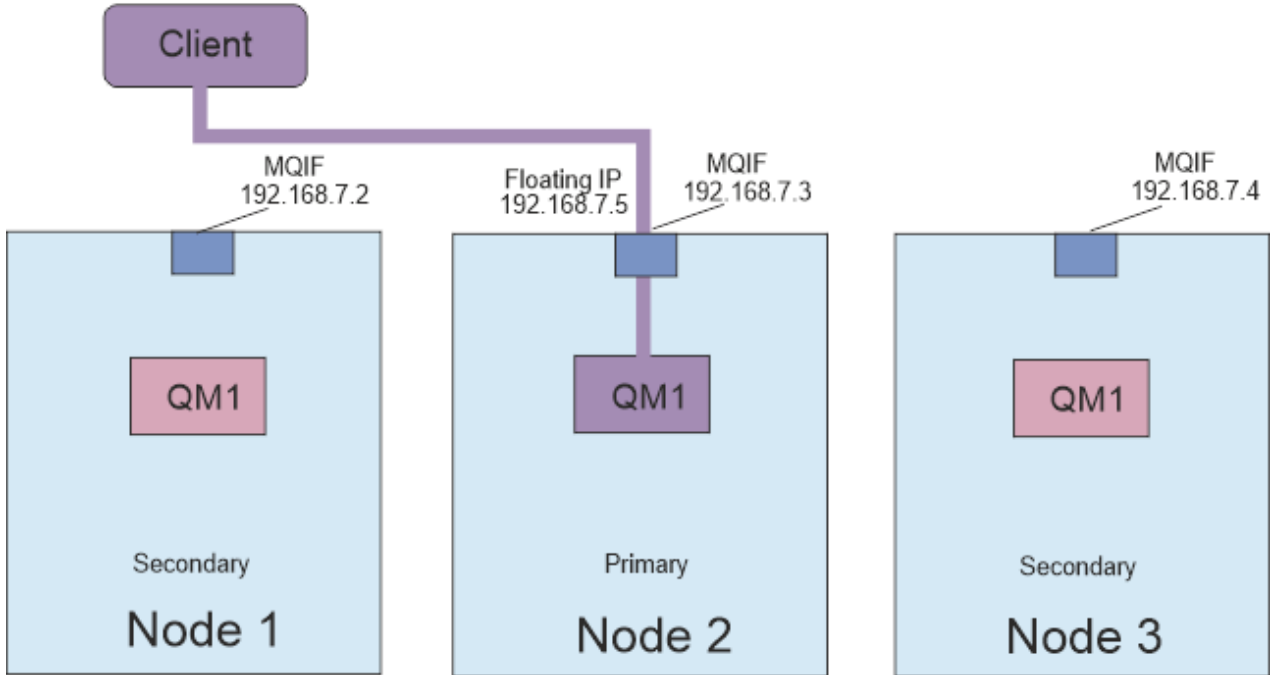
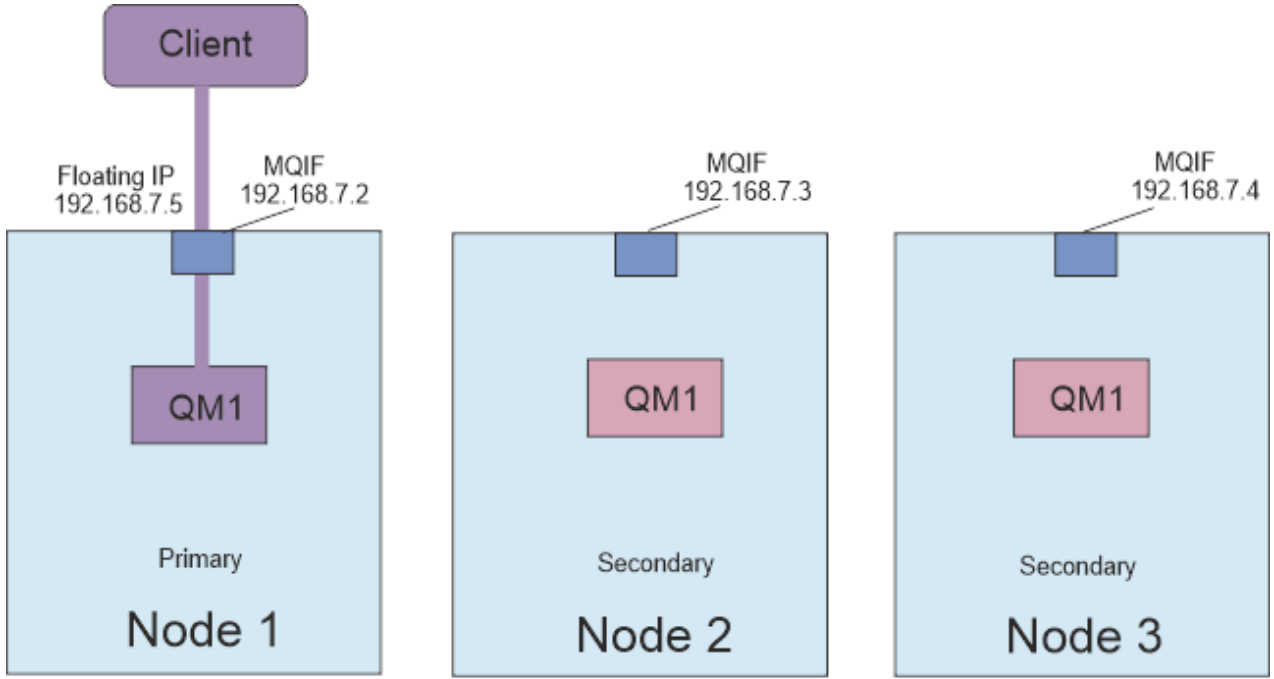
### İlgili başvurular

[rdqmadm \(eşlenmiş veri kuyruğu yöneticisi kümesini yönet\)](#)

Kayan IP adresi, istemcinin, üzerinde alıřtıđı HA grubundaki dğmden bađımsız olarak, eřlenmiř bir veri kuyruđu yneticisi (RDQM) iin aynı IP adresini kullanmasını sađlar. (Kayan IP adresi kullanımı isteđe bađlıdır.)

### **Bu grev hakkında**

**rdqmint** komutunu kullanarak kayan bir IP adresi oluřturabilir ya da silebilirsiniz. Kayan adres, RDQM iin birincil dğmdeki adlandırılmıř bir fiziksel arabirime bađlanır. RDQM zerinden bařarsız olursa ve farklı bir dğm birincil dğm olursa, kayan IP, yeni birincil dğmde aynı ada sahip bir arabirime bađlanır.  dğmdeki fiziksel arabirimler, kayan IP adresiyle aynı alt ađa ait olmalıdır. Ařađdaki řekil, kayan IP adresinin kullanımını gsterir.



Şekil 80. Kayan IP adresi

**rdqmint** komutunu çalıştırmak için hem **mqm** hem de **haclient** gruplarında bir kullanıcı olmanız gerekir. RDQM için birincil düğümde ya da ikincil düğümlerden birinde kayan IP adresi yaratabilir ya da silebilirsiniz.

**Not:** Birden çok RDIMM için aynı kayan IP adresini kullanamazsınız, her RDQM için kayan IP adresi benzersiz olmalıdır.

### Yordam

- Bir RDQM için kayan IP adresi yaratmak üzere şu komutu girin:

```
rdqmint -m qmname -a -f ipv4address -l interfacename
```

Burada:

### **qmname**

Kayan IP adresini yaratmakta olduğunuz RDQM ' nin adı.

### **ipv4address**

ipv4 biçiminde kayan IP adresi.

Kayan IP adresi, herhangi bir HA düğümünde önceden tanımlanmamış geçerli bir IPv4 adresi olmalıdır ve yerel arabirim için tanımlanan statik IP adresleriyle aynı alt ağa ait olmalıdır.

### **iç işlev kümeadi**

Bağlanacak birincil düğümdeki fiziksel arabirimin adı.

Örneğin:

```
rdqmint -m QM1 -a -f 192.168.7.5 -l MQIF
```

- Var olan bir kayan IP adresini silmek için aşağıdaki komutu girin:

```
rdqmint -m qmname -d
```

## **İlgili başvurular**

[rdqmint \(RDQM için kayan IP adresi ekleme ya da silme\)](#)

**Linux**

## **HA RDQM ' nin başlatılması, durdurulması ve durumunun görüntülenmesi**

Eşlenmiş bir veri kuyruğu yöneticisinin (RDQM) geçerli durumunu başlatmak, durdurmak ve görüntülemek için standart IBM MQ denetim komutlarının değişkenlerini kullanabilirsiniz.

## **Bu görev hakkında**

Eşlenmiş bir veri kuyruğu yöneticisinin (RDQM) geçerli durumunu hem mqm hem de haclient gruplarına ait bir kullanıcı olarak başlatan, durduran ve görüntüleyen komutları çalıştırmanız gerekir.

Bu kuyruk yöneticisine ilişkin birincil düğümde bir kuyruk yöneticisini başlatmak ve durdurmak için komutları çalıştırmanız gerekir.

## **Yordam**

- RDQM 'yi başlatmak için RDQM' in birincil düğümünde şu komutu girin:

```
strmqm qmname
```

Burada *qmname* , başlatmak istediğiniz RDQM ' nin adıdır.

RDQM başlatıldı ve Pacemaker RDQM ' yi yönetmeye başladı. Başka bir strmqm seçeneği belirtmek istiyorsanız, -ns seçeneğini strmqm ile belirtmeniz gerekir.

- Bir RDQM 'yi durdurmak için RDQM' in birincil düğümünde şu komutu girin:

```
endmqm qmname
```

Burada *qmname* , durdurmak istediğiniz RDQM ' nin adıdır.

Pacemaker RDQM ' yi yönetmeyi durdurduktan sonra RDQM sona erdirilir. Bir RDQM durdurulurken diğer tüm **endmqm** parametreleri kullanılabilir.

- Bir RDQM ' nin durumunu görüntülemek için şu komutu girin:

```
dspmq
```

Çıkış olan durum bilgileri, komutu RDQM ' in birincil ya da ikincil düğümünde çalıştırıp çalıştırmadığınıza bağlıdır. Birincil düğümde çalıştırılırsa, **dspmq** tarafından döndürülen normal



durum iletilerinden biri görüntülenir. Komutu ikincil bir düğümde çalıştırırsanız, `running elsewhere` durumu görüntülenir. Örneğin, **dspm** RDQM7 düğümünde çalıştırılırsa, aşağıdaki bilgiler döndürülebilir:

```
QMNAME (RDQM8)          STATUS(Running elsewhere)
QMNAME (RDQM9)          STATUS(Running elsewhere)
QMNAME (RDQM7)          STATUS(Running)
```

Birincil düğüm kullanılmıyorsa ya da **dspm**, root olmayan bir kullanıcı ya da `haclient` grubunun bir üyesi tarafından çalıştırılıyorsa, `Unavailable` durumu bildirilir. Örneğin:

```
QMNAME (RDQM8)          STATUS(Unavailable)
QMNAME (RDQM9)          STATUS(Unavailable)
QMNAME (RDQM7)          STATUS(Unavailable)
```

Bir düğüm tarafından bilinen kuyruk yöneticilerinin listesini ve bunların RDIMM olup olmadığını görüntülemek için **dspm -o ha** (ya da **dspm -o HA**) komutunu girebilirsiniz; örneğin:

```
dspm -o ha

QMNAME (RDQM8)          HA (Replicated)
QMNAME (RDQM9)          HA (Replicated)
QMNAME (RDQM7)          HA (Replicated)
QMNAME (QM7)            HA ()
```

## İlgili başvurular

[dspm \(kuyruk yöneticilerini görüntüle\)](#)

[endmqm \(kuyruk yöneticisini sona erdir\)](#)

[strmqm \(kuyruk yöneticisini başlat\)](#)

## V 9.3.0 Başarısız kaynak işlemleri

Bir RDQM yüksek kullanılabilirlik yapılandırmasının Pacemaker bileşeni, HA grubundaki düğümlerden birinde bir kaynakla ilgili bir sorunla karşılaştığında başarısız olan kaynak işlemleri ortaya çıkar.

RDQM HA çözümü, kaynakları izlemek ve yönetmek için Pacemaker 'ı kullanır (bkz. “RDQM yüksek kullanılabilirlik düzeyi” sayfa 552). Pacemaker düğümdeki bir kaynakta işlem gerçekleştirirken bir hata saptarsa, bu bilgileri başarısız bir kaynak işlemi kullanarak kaydeder. Bazı başarısız kaynak işlemleri kaynağın çalışmasını engelliyor ve Pacemaker 'ın kaynağı yeniden başlatabilmesi için önce kaynağın temizlenmesi gerekiyor.

**rdqmstatus -m** komutunu kullanarak, bir kuyruk yöneticisinin bir ya da daha çok düğümde başlatılmasını önleyen başarısız kaynak işlemleri olup olmadığını görebilirsiniz.

Daha sonra, bir kuyruk yöneticisiyle ilişkili başarısız olan kaynak işlemlerinin ayrıntılarını görüntülemek için **rdqmstatus -m qmname -a** komutunu kullanabilirsiniz. Bu başarısız olan kaynak işlemlerini temizlemek için **rdqmclean** komutunu kullanarak bu işlemi izleyin ve kısıtlanmış kaynakları serbest bırakın. (İlk olarak, başarısız kaynak işlemine neden olan sorunları çözmek için de işlem yapmanız gerekir.)

Aşağıdaki kaynaklar bir RDQM HA yapılandırmasında Pacemaker tarafından denetlenir ve başarısız kaynak işlemlerinin konuları olabilir:

- Kuyruk yöneticisi
- Kayan IP
- RDQM denetimi
- Dosya Sistemi
- DR çoğaltma (DRBD)
- HA eşleme (DRBD)

Her kaynak tipi aşağıdaki hata tiplerine tabi olabilir:

## Yumuşak

Geçici arızalar geçicidir ve Pacemaker , kaynağı zamanaşımına uğrayıncaya ya da durduruluncaya kadar kurtarmaya çalışmaya devam eder.

## Zor

Bir sabit hata, yönetim müdahalesi gerektirir. Sabit hatalar, kaynağın belirli bir düğümde çalışmasını engeller.

## Onarılmaz

Onulmaz bir hata, yönetim müdahalesi gerektirir. Onarılmaz hatalar, kaynağın herhangi bir düğümde çalışmasını engeller.

Başarısız kaynak kuyruğu işlemleri de içinde olmak üzere durum örnekleri için bkz. [“RDQM ve HA grubu durumunun görüntülenmesi” sayfa 574](#) .

Belirtilen bir kuyruk yöneticisiyle ilişkili tüm başarısız kaynak işlemlerini ya da RDQM HA yapılandırmasındaki tüm başarısız kaynak işlemlerini temizlemek için **rdqmclean** komutunu kullanabilirsiniz.

**Not:** Başarısız olan bazı kaynak işlemleri, kuyruk yöneticisinin bir düğümde engellenmesiyle sonuçlanmaz. Örneğin, beklenmeyen bir kuyruk yöneticisi sona erdikten sonra, Pacemaker çalışmadığı saptanan düğümde kuyruk yöneticisini yeniden başlatmayı dener. Başlatma başarılı olursa, kuyruk yöneticisinin düğümde çalışması engellenmez. Bu durumda başarısız olan kaynak işlemini bilmenin tek yolu **rdqmstatus -m qmname -a**komutunu çalıştırmaktır.

## İlgili görevler

[“RDQM ve HA grubu durumunun görüntülenmesi” sayfa 574](#)

HA grubunun ve tek tek eşlenmiş veri kuyruğu yöneticilerinin (RDIMM ' ler) durumunu görüntüleyebilirsiniz.

## İlgili başvurular

[rdqmclean](#)

[rdqmstatus](#)

Linux

## **RDQM ve HA grubu durumunun görüntülenmesi**

HA grubunun ve tek tek eşlenmiş veri kuyruğu yöneticilerinin (RDIMM ' ler) durumunu görüntüleyebilirsiniz.

## Bu görev hakkında

Tek tek RDIMM ' lerin ve HA grubunun durumunu bir bütün olarak görüntülemek için **rdqmstatus** komutunu kullanabilirsiniz.

V 9.3.0

Bir düğümün özet durumu, RDQM ' nin dayandığı DRBD çekirdek modülüyle ilgili bilgileri de görüntüler. RDQM ' yi yükselttiğinizde, sistemde çalışan RHEL çekirdeği sürümü için DRBD çekirdek modülünün doğru sürümünün kurulu olduğundan emin olmanız önemlidir. Durum, işletim sistemi çekirdeğinin sürümünü, DRBD modülünün oluşturulduğu çekirdek sürümünü, DRBD sürümünü ve DRBD çekirdek modülü yüklendi durumunu görüntüler.

**rdqmstatus** komutunu çalıştırmak için `mqm` ve `haclient` gruplarında bir kullanıcı olmanız gerekir. Komutu üç düğümden herhangi birinde çalıştırabilirsiniz.

## Yordam

- Bir düğümün ve HA yapılandırmasının parçası olan RDIMM ' lerin özet durumunu görüntülemek için:

```
rdqmstatus
```

Komutu üzerinde çalıştırdığınız düğümün tanıtıcısı, bu düğümle ilişkin çekirdek ve DRBD ayrıntıları ve HA yapılandırmasındaki RDIMM ' lerin durumu görüntülenir; örneğin:

```
Node: mqhvm07.exampleco.com
OS kernel version: 3.10.0-1160.15.2
DRBD OS kernel version: 3.10.0-1160
DRBD version: 9.1.1
DRBD kernel module status: Loaded

Queue manager name: RDQM8
Queue manager status: Running elsewhere
HA current location: mqhvm08.exampleco.com
HA preferred location: mqhvm08.exampleco.com
HA blocked location: None

Queue manager name: RDQM9
Queue manager status: Running elsewhere
HA current location: mqhvm09.exampleco.com
HA preferred location: mqhvm09.exampleco.com
HA blocked location: None

Queue manager name: RDQM7
Queue manager status: Running
HA current location: This node
HA preferred location: This node
HA blocked location: None
```

**V 9.3.0**

DRBD çekirdek modülü durumu aşağıdaki değerlerden biridir:

#### **Yüklendi**

DRBD modülünün yüklendiğini gösterir.

#### **Kısmen yüklendi**

DRBD modülü yüklendiğinde ortaya çıkabilir, ancak bir uyuşmazlık nedeniyle düzgün çalışmayabilir.

#### **Yüklenmedi**

DRBD modülü yüklenmedi. Bu, henüz bir RDQM kuyruk yöneticisi yaratılmadığında, yeni kurulan bir yapılandırmada görüntülenebilir.

#### **Kurulu değil**

DRBD modülünden birinin kurulu olmadığını gösterir. ya da IBM MQ , DRBD modülünün işletim sistemi çekirdeği sürümünü belirleyemedi.

#### **Önceden kurulan sürüm yüklenmeye devam ediyor**

Bu durum, var olan DRBD modülü çalışırken (yani bir RDQM kuyruk yöneticisi çalışırken) yeni bir DRBD modülü kurulursa ortaya çıkabilir. Yeni kurulan modül, gerçekte çalışmakta olan modül değil, durumda bildirilir.

- HA grubundaki üç düğümün durumunu görüntülemek için şu komutu girin:

```
rdqmstatus -n
```

Her düğümün çevrimiçi ya da çevrimdışı durumu bildirilir. Örneğin:

```
Node mqha04(mqhvm04.example.com) is online
Node mqha05(mqhvm05.example.com) is offline
Node mqha06(mqhvm06.example.com) is online
```

- HA grubundaki tüm düğümlerde belirli bir kuyruk yöneticisinin durumunu görüntülemek için şu komutu girin:

```
rdqmstatus -m qmname
```

Burada *qmname* , durumunu görüntülemek istediğiniz RDQM ' nin adıdır. Geçerli düğümdeki RDQM ' nin durumu görüntülenir ve bunu, geçerli düğümün perspektifinden diğer iki düğümün durumunun bir özeti izler.

**V 9.3.0**

- Başarısız olan kaynak işlemlerinin ayrıntıları da içinde olmak üzere, HA grubundaki tüm düğümlerde belirli bir kuyruk yöneticisinin durumunu görüntülemek için aşağıdaki komutu girin:

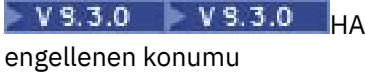
```
rdqmstatus -m qmname -a
```

Burada *qmname* , durumunu görüntülemek istediğiniz RDQM ' nin adıdır. Geçerli düğümdeki RDQM ' nin durumu görüntülenir ve bunu, geçerli düğümün perspektifinden diğer iki düğümün durumunun bir özeti izler. Bunu, RDQM ile ilişkili başarısız olan kaynak işlemleriyle ilgili ayrıntılar izler.

- Aşağıdaki tablo, bir RDQM için `rdqmstatus -m qmname` komutu tarafından döndürülebilecek geçerli düğümle ilgili bilgileri özetler.

Çizelge 32. Geçerli düğüm durumu		
Durum özneliği	Olası değerler	Görüntülendiğinde
Düğüm adı	<i>Düğüm Adı</i>	Her zaman görüntülenir
Kuyruk yöneticisi durumu	Çalışıyor Başka bir yerde koşmak Bitti Yok	Her zaman görüntülenir
CPU	<i>n.nn%</i>	Yalnızca geçerli düğümün birincil rolü varsa gösterilir (yani, bu düğümde RDQM çalışıyorsa)
Bellek	<i>nnn</i> MB kullanıldı, <i>y.y</i> GB ayrıldı	Yalnızca geçerli düğümün birincil rolü varsa gösterilir (yani, bu düğümde RDQM çalışıyorsa)
Kuyruk yöneticisi dosya sistemi	<i>nnn</i> MB kullanıldı, <i>y.y</i> GB ayrıldı [z%]	Yalnızca geçerli düğümün birincil rolü varsa gösterilir (yani, bu düğümde RDQM çalışıyorsa)
HA rolü	Birincil İkincil Bilinmeyen	Her zaman görüntülenir
HA durumu	Yedekteki tüm düğümler Bu düğüm yedekte bekliyor Yedek veritabanındaki uzak düğümler Karışık  <i>uzak düğümlerin durumu</i>	Yedekteki tüm düğümler Yedek veritabanında yürürlükteki düğüm Her iki uzak düğüm de yedekte Her uzak düğüm için farklı durum (tek tek durum için sonraki çizelgeye bakın)  Her iki uzak düğüm için aynı durum (tüm değerler için sonraki çizelgeye bakın)
HA denetimi	Etkin Geçersiz Kılındı Bilinmeyen	Her zaman görüntülenir. RDQM ' in Pacemaker denetimi altında olup olmadığını gösterir
HA tercih edilen konumu	Yok Bu düğüm Bilinmeyen <i>Düğüm Adı</i>	Her zaman görüntülenir

Çizelge 32. Geçerli düğüm durumu (devamı var)

Durum özneliği	Olası değerler	Görüntülediğinde
	<p>Yok-Kuyruk yöneticisinin herhangi bir düğümde çalışması engellenmedi</p> <p>Bu düğüm-Bir ya da daha fazla başarısız kaynak işlemi nedeniyle, kuyruk yöneticisinin yürürlükteki düğümde çalışması engellendi</p> <p><i>nodename</i> -Başarısız olan bir ya da daha çok kaynak işlemi nedeniyle, kuyruk yöneticisinin <i>nodename</i> üzerinde çalışması engellendi.</p> <p><i>nodename1, nodename2</i> -Bir ya da daha fazla başarısız kaynak işlemi nedeniyle, kuyruk yöneticisinin <i>nodename1</i> ve <i>nodename2</i> üzerinde çalışması engellendi.</p> <p>Tüm düğümler-Bir ya da daha fazla başarısız kaynak işlemi nedeniyle kuyruk yöneticisinin tüm düğümlerde çalışması engellendi</p>	Her zaman görüntülenir
HA kayan IP arabirimi	<i>Arabirim_adi</i>	Her zaman görüntülenir
HA kayan IP adresi	<i>IPV4_address</i>	Her zaman görüntülenir

Aşağıdaki çizelge, HA grubundaki diğer düğümler için `rdqmstatus -m qmname` komutu tarafından döndürülen bilgileri özetler.

Çizelge 33. Diğer düğüm durumu

Durum özneliği	Olası değerler	Görüntülediğinde
Düğüm adı	<i>nodename</i>	Her zaman görüntülenir
HA durumu	<p>Olağan</p> <p>Eşitleme devam ediyor</p> <p>Uzak kullanılamıyor</p> <p>Tutarsız</p> <p>Duraklatıldı</p> <p>Yedek veritabanında uzak düğüm</p> <p>Bilinmeyen</p>	<p>Düğümler birbiriyle uyumlu</p> <p>Uzak düğümle uyumlulaştırılıyor</p> <p>Uzak düğümle iletişim kurulamıyor</p> <p>Uzak düğümle uyumsuz ve eşitlenmiyor</p> <p>Eşleme duraklatıldı</p> <p>Yedek veritabanında uzak düğüm</p>
HA eşitlemesi devam ediyor	<i>n.n%</i>	Eşitleme devam ederken görüntülenir ve komut <code>root</code> olarak çalıştırılır
HA tahmini eşitleme süresi	<i>yyyy-aa-gg hh:mm:ss.nnn</i>	Eşitleme devam ederken görüntülenir
HA eşitlenmemiş veriler	<i>nKB</i>	Uzak düğüm kullanılmadığında ya da tutarsız olduğunda görüntülenir

Çizelge 33. Diğer düğüm durumu (devamı var)

Durum özneteliği	Olası değerler	Görüntülediğinde
V9.3.0 HA son eşitlemesi	yyyy-aa-gg hh:mm:ss.nnn	HA verileri zamanuyumlu olmadığına (ilk eşitlemeden sonra) görüntülenir. Verilerin en son eşitlendiği tarihi ve saati verir.

## Örnek

Birincil düğümdeki normal durum örneği:

```
Node: mqhavam07.exampleco.com
Queue manager status: Running
CPU: 0.00
Memory: 123MB
Queue manager file system: 606MB used, 1.0GB allocated [60%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: eth4
HA floating IP address: 192.0.2.4
```

```
Node: mqhavam08.exampleco.com
HA status: Normal
```

```
Node: mqhavam09.exampleco.com
HA status: Normal
```

İkincil bir düğümdeki normal durum örneği:

```
Node: mqhavam08.exampleco.com
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: mqhavam07.exampleco.com
HA preferred location: mqhavam07.exampleco.com
HA blocked location: None
HA floating IP interface: eth4
HA floating IP address: 192.0.2.4
```

```
Node: mqhavam07.exampleco.com
HA status: Normal
```

```
Node: mqhavam09.exampleco.com
HA status: Normal
```

Eşitleme devam ederken birincil düğümde durum örneği:

```
Node: mqhavam07.exampleco.com
Queue manager status: Running
CPU: 0.53
Memory: 124MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Synchronization in progress
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: eth4
HA floating IP address: 192.0.2.4
```

```
Node: mqhavam08.exampleco.com
HA status: Synchronization in progress
HA synchronization progress: 11.0%
```

```
HA estimated time to completion:      2017-09-06 14:55:05
Node:                                 mqhavam09.exampleco.com
HA status:                             Synchronization in progress
HA synchronization progress:          11.0%
HA estimated time to completion:      2017-09-06 14:55:06
```

### V 9.3.0 Eşitleme kaybedildiğinde birincil düğümdeki durum örneği:

```
Node:                                 mqhavam07.exampleco.com
Queue manager status:                 Running
CPU:                                  0.53
Memory:                               124MB
Queue manager file system:           51MB used, 1.0GB allocated [5%]
HA role:                              Primary
HA status:                            Mixed
HA control:                           Enabled
HA current location:                  This node
HA preferred location:                 This node
HA blocked location:                  None
HA floating IP interface:             eth4
HA floating IP address:                192.0.2.4

Node:                                 mqhavam08.exampleco.com
HA status:                             Normal

Node:                                 mqhavam09.exampleco.com
HA status:                             Inconsistent
HA out of sync data:                  15932KB
HA last in sync:                      2017-09-06 14:55:06
```

### Birden çok durumu gösteren birincil düğüm örneği:

```
Node:                                 mqhavam07.exampleco.com
Queue manager status:                 Running
CPU:                                  0.02
Memory:                               124MB
Queue manager file system:           51MB used, 1.0GB allocated [5%]
HA role:                              Primary
HA status:                            Mixed
HA control:                           Enabled
HA current location:                  This node
HA preferred location:                 This node
HA blocked location:                  None
HA floating IP interface:             eth4
HA floating IP address:                192.0.2.4

Node:                                 mqhavam08.exampleco.com
HA status:                             Normal

Node:                                 mqhavam09.exampleco.com
HA status:                             Inconsistent
```

### V 9.3.0 Başarısız olan kaynak işlemlerini gösteren birincil düğüm örneği:

```
Node:                                 mqhavam07.exampleco.com
Queue manager status:                 Running
CPU:                                  0.00%
Memory:                               123MB
Queue manager file system:           606MB used, 1.0GB allocated [60%]
HA role:                              Primary
HA status:                             Normal
HA control:                           Enabled
HA current location:                  This node
HA preferred location:                 mqhavam08.exampleco.com
HA blocked location:                  mqhavam08.exampleco.com
HA floating IP interface:             eth4
HA floating IP address:                192.0.2.4

Node:                                 mqhavam08.exampleco.com
HA status:                             Normal

Node:                                 mqhavam09.exampleco.com
HA status:                             Normal

Failed resource action:                Start
Resource type:                          Filesystem
```

```
Failure node: mqhavam08.exampleco.com
Failure time: 2017-09-06 12:00:00
Failure reason: Couldn't find directory [/var/mqm/vols/qmname] to use
as a mount point
Blocked location: mqhavam08.exampleco.com
```

Bu durum, Pacemaker 'ın mqhavam08.exampleco.com düğümünde dosya sistemini 12:00 'da başlatamadığını gösterir. Bu başarısız kaynak işlemi, kuyruk yöneticisinin mqhavam08.exampleco.com üzerinde çalışmasını engellediği anlamına gelir. Başarısız kaynak işlemine neden olan temel sorun çözüldükten sonra, başarısız olan işlemi temizlemek için **rdqmclean** komutunu çalıştırın; böylece Pacemaker işlemi yeniden deneyebilir (gerekirse).

**V 9.3.0** İşletim sistemi çekirdek sürümü (RHEL 7.9) ile DRBD çekirdek modülü (RHEL 7.8 için hedeflenmiş) arasındaki uyumsuzluğu gösteren bir özet durumu örneği. Durum, DRBD çekirdek modülünün yüklendiğini ve kuyruk yöneticisinin çalıştığını bildirir de, DRBD çekirdek modülünü bu durumda çalışan işletim sistemi çekirdeği için hedeflenen sürümle güncellemelisiniz.

```
Node: mqhavam07.exampleco.com
OS kernel version: 3.10.0-1160.15.2
DRBD OS kernel version: 3.10.0-1127
DRBD version: 9.1.1
DRBD kernel module status: Loaded

Queue manager name: RDQM7
Queue manager status: Running
HA current location: This node
HA preferred location: This node
HA blocked location: None
```

**V 9.3.0** İşletim sistemi çekirdek sürümü (RHEL 7.9) ile DRBD çekirdek modülü (RHEL 7.6 için hedeflenen) arasındaki uyumsuzluğu gösteren bir özet durumu örneği. Bu örnekte sürüm uyumsuzluğu daha ciddi ve DRBD çekirdek modülü başarıyla yüklenemedi. Sonuç olarak, kuyruk yöneticisi tercih edilen düğümünde başlayamaz ve HA durumu Unknown. Bu hatayı çözmek için DRBD çekirdek modülü, çalışan işletim sistemi çekirdeğine ilişkin sürüm hedefiyle güncellenmelidir.

```
Node: mqhavam57.exampleco.com
OS kernel version: 3.10.0-1160.15.2
DRBD OS kernel version: 3.10.0-957
DRBD version: 9.1.2+ptf.3
DRBD kernel module status: Partially loaded

Queue manager name: QM2
Queue manager status: Running elsewhere
HA status: Unknown
HA current location: mqhavam58.exampleco.com
HA preferred location: This node
HA blocked location: All nodes
```

## İlgili başvurular

**Linux** [rdqmstatus](#)

### **Yüksek kullanılabilirlikli yapılandırmalarda IP adreslerinin değiştirilmesi**

Yüksek kullanılabilirlik yapılandırmasındaki arabirimlerden herhangi birinin IP adreslerini değiştirirseniz, yüksek kullanılabilirlik işlemi artık kullanılamaz ve kuyruk yöneticisi adreslerin değiştirildiği düğümde çalışmaz.

rdqm.ini dosyasında HA işlemi için en çok üç IP adresi belirtirsiniz. Pacemaker izleme programı adreslerini önceden değiştirdiyseniz, yordamı izlemeden önce bunları geçici olarak özgün değerlerine geri yüklemeniz gerekir. Aksi takdirde HA RDQM kuyruk yöneticisi silinemez.

1. Her bir düğümdeki HA yapılandırmasını kaldırın. HA 'yı kaldırmak için kuyruk yöneticilerini yedekleyip sonra bunları silin, bkz. [“IBM MQ kuyruk yöneticisi verilerinin yedeklenmesi ve geri yüklenmesi” sayfa 655](#) ve [“HA RDQM 'i silme” sayfa 563](#)ve daha sonra HA grubunun kendisini kaldırın, bkz. [“Pacemaker kümesinin silinmesi \(HA grubu\)” sayfa 561](#).



2. HA yapılandırmasını yeni IP adresleriyle yeniden oluşturun, bkz. [“Pacemaker kümesinin tanımlanması \(HA grubu\)” sayfa 557](#).
3. HA kuyruk yöneticilerini yeniden yaratın ve yedeklemeyi geri yükleyin (bkz. [“HA RDQM yaratılması” sayfa 561](#) ve [“IBM MQ kuyruk yöneticisi verilerinin yedeklenmesi ve geri yüklenmesi” sayfa 655](#)).

Linux

## **Yüksek kullanılabilirlikli yapılandırmada hatalı bir düğümün değiştirilmesi**

HA grubundaki düğümlerden biri arızalanırsa, bunu değiştirebilirsiniz.

### **Bu görev hakkında**

Bir düğümü değiştirmek için atılacak adımlar senaryoya bağlıdır:

- Hatalı düğümü, aynı yapılandırmaya sahip bir düğümlerle değiştiriyorsanız, HA grubunu kesintiye uğratmadan düğümü değiştirebilirsiniz.
- Yeni düğümün farklı bir yapılandırması varsa, HA grubunu silmeniz ve yeniden oluşturmanız gerekir. Önce üzerinde çalıştıkları düğümden kuyruk yöneticilerini yedekleyebilir, daha sonra HA grubunu yeniden oluşturduktan sonra geri yükleyebilirsiniz.

### **Yordam**

- Yeni düğüm hatalı düğüme (aynı anasistem adı, aynı IP adresleri, vb.) benzeyecek şekilde yapılandırıldıysa, yeni düğümden aşağıdaki adımları tamamlayın:
  - a) Diğer düğümlerdeki dosyalarla eşleşen bir `rdqm.ini` dosyası oluşturun ve `rdqmadm -c` komutunu çalıştırın (bkz. [“Pacemaker kümesinin tanımlanması \(HA grubu\)” sayfa 557](#)).
  - b) Eşlenen her bir veri kuyruğu yöneticisini yeniden yaratmak için `crtmqm -sxs qmanager` komutunu çalıştırın (bkz. [“HA RDQM yaratılması” sayfa 561](#)).
- Yeni düğüm, hatalı düğümden farklı bir yapılandırmaya sahipse:
  - a) Gerekirse, kuyruk yöneticilerinizi yedekleyin (bkz. [“IBM MQ kuyruk yöneticisi verilerinin yedeklenmesi ve geri yüklenmesi” sayfa 655](#)).
  - b) `dltmqm` komutunu kullanarak, eşlenen veri kuyruğu yöneticilerini HA grubundaki diğer düğümlerden silin (bkz. [“HA RDQM 'i silme” sayfa 563](#)).
  - c) `rdqmadm -u` komutunu kullanarak Pacemaker kümesinin yapılandırmasını kaldırma (bkz. [“Pacemaker kümesinin silinmesi \(HA grubu\)” sayfa 561](#)).
  - d) `rdqmadm -c` komutunu kullanarak, yeni düğüme ilişkin bilgiler de içinde olmak üzere Pacemaker kümesini yeniden yapılandırın (bkz. [“Pacemaker kümesinin tanımlanması \(HA grubu\)” sayfa 557](#)).
  - e) Gerekliyse (diğer düğümlere SSH erişiminiz yoksa), diğer düğümlerde her bir eşlenmiş veri kuyruğu yöneticisini yeniden yaratmak için `crtmqm -sxs qmanager` komutunu çalıştırın (bkz. [“HA RDQM yaratılması” sayfa 561](#)).
  - f) Yeni düğümden kuyruk yöneticilerini oluşturmak için `crtmqm -sx qmanager` komutunu çalıştırın.
  - g) Gerekliyse, verileri ve yapılandırmayı kuyruk yöneticilerinize geri yükleyin (bkz. [“IBM MQ kuyruk yöneticisi verilerinin yedeklenmesi ve geri yüklenmesi” sayfa 655](#)).

Linux

MQ Adv.

## **RDQM olağanüstü durumdan kurtarma**

RDQM (eşlenmiş veri kuyruğu yöneticisi), Linux platformlarının bir alt kümesinde kullanılabilir ve bir olağanüstü durumdan kurtarma çözümü sağlayabilir.

Tam ayrıntılar için [Software Product Compatibility Reports](#) (Yazılım Ürünü Uyumluluk Raporları) başlıklı konuya bakın.

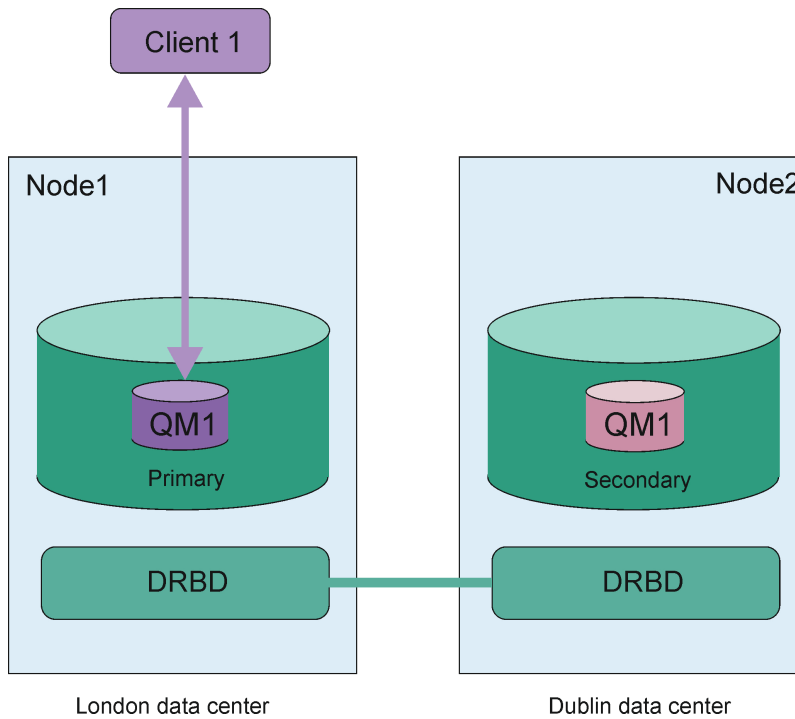
Bir sunucuda çalışan bir olağanüstü durumdan kurtarma kuyruk yöneticisinin birincil eşgörünümünü ve kurtarma düğümü olarak işlev gören başka bir sunucuda kuyruk yöneticisinin ikincil bir eşgörünümünü yaratabilirsiniz. Veriler, kuyruk yöneticisi eşgörünümleri arasında eşlenir. Birincil kuyruk yöneticinizi kaybederseniz, ikincil yönetim ortamını birincil yönetim ortamına el ile yerleştirip kuyruk yöneticisini

başlatılabilir ve işi aynı yerden sürdürebilirsiniz. Bir kuyruk yöneticisi ikincil roldeyken başlatılamaz. İki düğüm arasındaki verilerin eşlenmesi DRBD tarafından işlenir.

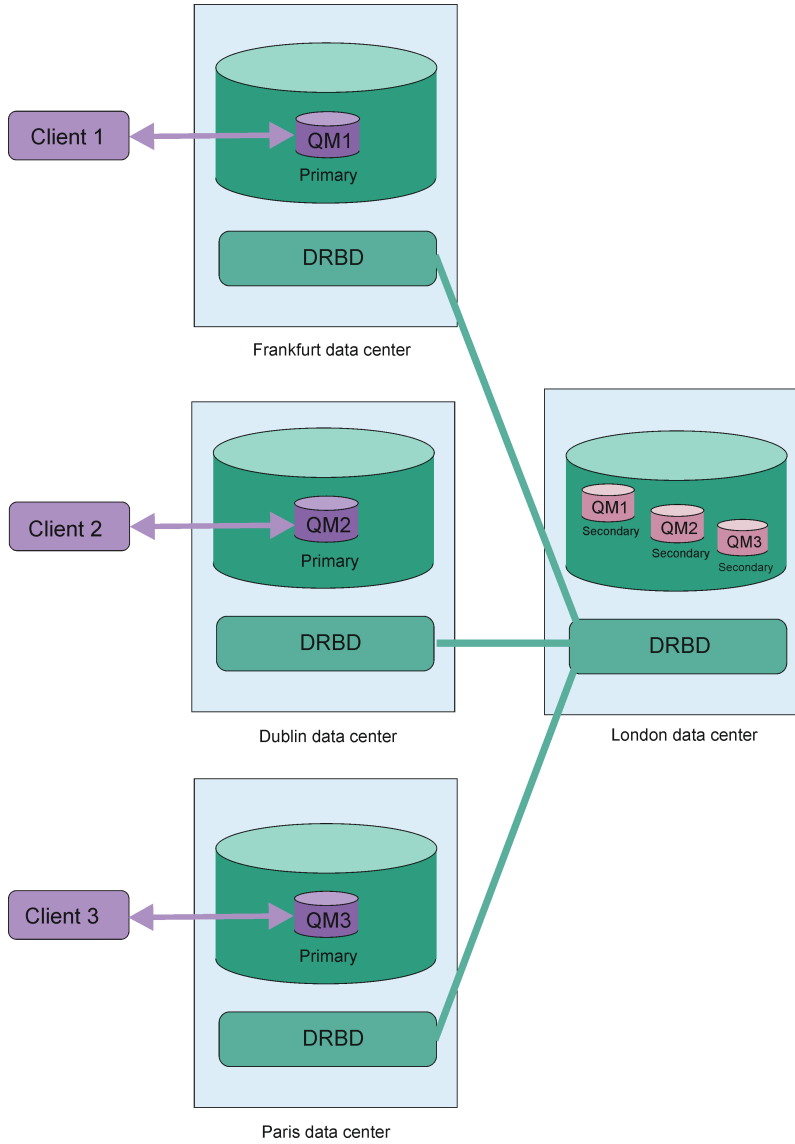
Birincil ve ikincil kuyruk yöneticileri arasında verilerin zamanuyumlu ve zamanuyumsuz olarak eşlenmesi arasında seçim yapabilirsiniz. Zamanuyumsuz seçeneği belirlerseniz, IBM MQ PUT ya da GET gibi işlemler tamamlanır ve olay ikincil kuyruk yöneticisine eşlenmeden önce uygulamaya geri döner. Zamanuyumsuz kopyalama, bir kurtarma durumu sonrasında bazı ileti sistemi verilerinin kaybolabileceği anlamına gelir. Ancak ikincil kuyruk yöneticisi tutarlı bir durumda olacak ve ileti akışının biraz daha önceki bir kısmında başlatılsa bile hemen çalışmaya başlayabilir.

Var olan bir kuyruk yöneticisine olağanüstü durumdan kurtarma ekleyemezsiniz, ancak var olan bir kuyruk yöneticisini RDQM kuyruk yöneticisi olacak şekilde geçirebilirsiniz (bkz. [“Bir kuyruk yöneticisinin DR RDQM kuyruk yöneticisi olacak şekilde geçirilmesi”](#) sayfa 589).

Farklı sunucularda çalışan birden çok RDQM kuyruk yöneticisi çiftiniz olabilir. Örneğin, tüm ikincil olağanüstü durumdan kurtarma kuyruğu yöneticileri aynı düğümde çalışırken, farklı düğümlerde çalışan birincil olağanüstü durumdan kurtarma kuyruğu yöneticileriniz olabilir. Bazı örnek yapılandırmalar aşağıdaki çizgelerde gösterilmiştir.



Şekil 81. Tek RDQM çifti



Şekil 82. Aynı düğümdeki ikincil kuyruk yöneticileri

## Eşleme, eşitleme ve anlık görüntüler

Bir olağanüstü durumdan kurtarma yapılandırmasındaki iki düğüm bağlıken, bir olağanüstü durumdan kurtarma kuyruk yöneticisine ilişkin kalıcı verilerde yapılan güncellemeler, kuyruk yöneticisinin birincil eşgörünümünden ikincil yönetim ortamına aktarılır. Bu, **eşleme** olarak bilinir.

İki düğüm arasındaki ağ bağlantısı kaybolursa, kuyruk yöneticisinin birincil eşgörünümüne ilişkin kalıcı verilerde yapılan değişiklikler izlenir. Ağ bağlantısı geri yüklendiğinde, ikincil eşgörünümü mümkün olan en kısa sürede hızlandırmak için farklı bir işlem kullanılır. Bu, **eşitleme** olarak bilinir.

Uyumlaştırma devam ederken, ikincil eşgörünümdeki veriler tutarsız durumda. İkincil kuyruk yöneticisi verilerinin durumunun **anlık görünümü** alınır. Uyumlaştırma sırasında ana düğümde ya da ağ bağlantısında bir hata oluşursa, ikincil eşgörünüm bu anlık görünüme döner ve kuyruk yöneticisi başlatılabilir. Ancak, özgün ağ hatasından bu yana gerçekleşen güncelleştirmelerden herhangi biri kaybolur.

## Bölümlenmiş veriler (bölünmüş beyin)

DR RDQM yapılandırmaları, kurtarma düğümünde ikincil eşgörünümü yükseltmek ve çalıştırmak için kuyruk yöneticisinin birincil eşgörünümünü kaybettiikten sonra kullanıcı işlemi gerektirir. Önceki birincil

kuyruk yöneticisinin durdurulduğundan emin olmak için ikincil eşgörünümü yükselten kişinin (ya da her neyse) sorumluluğundadır. Özgün birincil sistem çalışmaya devam ederse, iletiler işlenebilir ve olağan işlem geri yüklendiğinde, kuyruk yöneticisinin iki eşgörünümü verilerin farklı görünümüne sahip olur. Bu, bölümlenmiş veya bölünmüş beyin durumu olarak bilinir.

Aşağıdaki durumları göz önünde bulundurun:

- Birincil kuyruk yöneticisinin çalıştığı düğüm tamamen başarısız oluyor. İkincil yönetim ortamını birincil yönetim ortamı olacak şekilde yükseltebilirsiniz; çalışmadığı için özgün birincil yönetim ortamını durdurmak için işlem yapamazsınız. Özgün düğüm onarıldığında ya da değiştirildiğinde, o düğümdeki kuyruk yöneticisi başlangıçta ikincil olur ve kurtarma düğümündeki birincil kuyruk yöneticisiyle uyumlulaştırılır. Daha sonra, iki kuyruk yöneticisinin rolleri tersine çevrilir ve olağan işlem yeniden önerilir. Bu durumda olası tek veri kaybı, düğüm başarısız olmadan önce birincil verilerin ikincil düğüm eşlemesini tamamlamadığı verilerdir.
- Kuyruk yöneticisinin birincil ve ikincil örneklerini çalıştıran düğümler arasındaki eşleme bağlantısını etkileyen bir ağ hatası var. Bu durumda, ikincil değeri yükseltmeden önce özgün birincil birincil değeri durdurduğunuzdan emin olmanız gerekir. Özgün birincil veritabanında başka bir ağ bağlantısı varsa, aynı anda çalışan iki birincil yönetim ortamınız vardır ve bölümlenmiş veriler birikebilir. (Eşleme bağlantısı çalışıyorsa, birincil yönetim ortamı çalışmaya devam ediyorsa, ikincil bir kuyruk yöneticisini yükseltmezsiniz, komut başarısız olur.)
- Kuyruk yöneticisinin birincil yönetim ortamını çalıştıran düğümde tam bir ağ hatası var. İkincil yönetim ortamını yükseltmeden önce birincil yönetim ortamını durdurmaya dikkat etmelisiniz. Ağ geri yüklendiğinde önceki birincil yönetim ortamı çalışmaya devam ediyorsa, iki birincil yönetim ortamı olur ve yeniden bölümlenmiş veriler birikir.

Yönetilen bir hata durumunda yedek sisteme geçiş işlemi gerçekleştirdiğinizde, kuyruk yöneticisi eşgörünümleri için `partitioned` DR durumunu görmemeniz gerekir. Yönetilen bir yedek sisteme geçiş işlemi, birincil düğümdeki kuyruk yöneticisini sona erdirir ve veriler tam olarak eşlendikten sonra kurtarma düğümünde kuyruk yöneticisini başlatır. Kuyruk yöneticisi sona erdirildiği ve veriler, kurtarma düğümünde başlatılmadan önce düğümler arasında uyumlulaştırıldığı için bölümlenmiş durum beklenmiyor. Düğümler arasında bağlantı kaybı olduğunda kuyruk yöneticisi kurtarma düğümünde başlatılırsa, bağlantı kaybedildiğinde kuyruk yöneticisi ana düğümde etkinse, veri sapması olasılığı yüksektir. Bu senaryoda, kuyruk yöneticisi verileri eşitlenmediği için bağlantı geri yüklendiğinde bölümlenmiş bir durumun bildirilmesi beklenir. Bölümlenmiş bir durum oluşursa, iki veri kümesini incelemeniz ve hangi kümenin tutulması konusunda bilgilendirilmiş bir karar vermeniz gerekebilir. Bkz. [“DR RDQM ' de bölümlenmiş \(bölünmüş beyin\) sorununu çözme” sayfa 604.](#)

## Linux **RDQM DR çözümü için gereksinimler**

Bir RDQM olağanüstü durumdan kurtarma (DR) kuyruk yöneticisi çiftini yapılandırmadan önce bir dizi gereksinimi karşılamamız gerekir.

### Sistem gereksinimleri

RDQM DR ' yi yapılandırmadan önce, RDQM DR kuyruk yöneticilerini barındıracak her bir sunucuda bazı yapılandırmaları tamamlamanız gerekir.

- Her düğüm için `drbdpool` adlı bir birim grubu gerekir. Her olağanüstü durumdan kurtarma eşlenmiş veri kuyruğu yöneticisi (DR RDQM) için depolama, bu birim grubundan kuyruk yöneticisi başına iki ayrı mantıksal birim olarak ayrılır. (Her kuyruk yöneticisi, anlık görüntüye geri çevirme işlemi desteklemek için iki mantıksal birim gerektirir; bu nedenle her bir DR RDQM, oluşturduğunuzda belirttiğiniz depolama alanının yalnızca iki katının üzerinde ayrılır.) En iyi performans için bu birim grubu, iç disk sürücülerine (tercihen SSD ' ler) karşılık gelen bir ya da daha fazla fiziksel birimden oluşmalıdır.
- `drbdpool` birim grubunu oluşturduktan sonra başka bir işlem yapmayın. IBM MQ , `drbdpool` içinde oluşturulan mantıksal birimleri ve bunların nasıl ve nereye bağlandığını yönetir.
- Her düğüm, veri eşlemesi için kullanılan bir arabirim gerektirir. Bu, eşlenen tüm veri kuyruğu yöneticilerinin beklenen iş yüküne ilişkin eşleme gereksinimlerini desteklemek için yeterli bant genişliğine sahip olmalıdır.

Maksimum hata toleransı için, bu arabirim bağımsız bir Ağ Arabirim Kartları (NIC) olmalıdır.

- DRBD, RFC 1123 tarafından değiştirilen RFC 952 tarafından tanımlandığı şekilde, RDQM için kullanılan her düğümün geçerli bir Internet anasistem adına ( `uname -n` tarafından döndürülen değer) sahip olmasını gerektirir.
- DR RDQM için kullanılan düğümler arasında bir güvenlik duvarı varsa, güvenlik duvarı eşleme için kullanılan kapılardaki düğümler arasında trafiğe izin vermelidir. RHEL ' de standart güvenlik duvarını çalıştırıyorsanız, gerekli kapıları açan bir örnek komut dosyası ( `/opt/mqm/samp/rdqm/firewalld/configure.sh`) sağlanır. Komut dosyasını rootolarak çalıştırmanız gerekir. Başka bir güvenlik duvarı kullanıyorsanız, hangi kapıların açılması gerektiğini görmek için `/usr/lib/firewalld/services/rdqm*` hizmet tanımlarını inceleyin. Komut dosyası, DRBD ve IBM MQ için aşağıdaki kalıcı firewallD hizmet kurallarını ekler (HA kullanmıyorsanız, komut dosyasını Pacemaker kapılarını atlamak için düzenleyebilirsiniz):
  - `MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-drbd.xml` 7000-7100 TCP bağlantı noktalarına izin verir.
  - `MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-mq.xml` , TCP kapısı 1414 'e izin verir (farklı bir kapı gerekiyorsa komut dosyasını düzenlemeniz gerekir)
- Sistem SELinux 'u izin dışında bir kipte kullanıyorsa, aşağıdaki komutu çalıştırmanız gerekir:

```
semanage permissive -a drbd_t
```

## Ağ gereksinimleri

Farklı veri merkezlerinde olağanüstü durumdan kurtarma için kullanılan düğümleri bulmanız önerilir.

Aşağıdaki sınırlamaları dikkate almanız gerekir:

- Veri merkezleri arasında artan gecikme süresiyle performans hızla azalır. IBM , zamanuyumlu eşleme için en fazla 5 ms ve zamanuyumsuz kopyalama için 100 ms gecikme sürelerini destekler.
- Eşleme bağlantısı üzerinden gönderilen veriler, IBM MQ AMS kullanılarak gerçekleştirilebilecek verilerin ötesinde herhangi bir ek şifrelemeye tabi değildir.
- Olağanüstü durumdan kurtarma için bir RDQM kuyruk yöneticisinin yapılandırılması, iki RDQM düğümü arasında veri eşleme gereksinimi nedeniyle bir ek yüke neden olur. Zamanuyumlu eşleme, zamanuyumsuz kopyalamaktan daha fazla ek yük ile karşı karşıya gelir. Zamanuyumlu eşleme kullanıldığında, veriler her iki düğüme yazılncaya kadar disk G/Ç işlemleri engellenir. Zamanuyumsuz kopyalama kullanıldığında, işlemin devam edebilmesi için verilerin yalnızca birincil düğüme yazılması gerekir.

## Kuyruk yöneticileriyle çalışmaya ilişkin kullanıcı gereksinimleri

Eşlenmiş veri kuyruğu yöneticilerini (RDM) oluşturmak, silmek ya da yapılandırmak için kök kullanıcı olmanız ya da aşağıdaki komutlar için sudo yetkisi verilen `mqm` grubuna ait bir kullanıcı kimliğiniz olması gerekir:

- `crtmqm`
- `dltmqm`
- `rdqmdr`

`mqm` grubuna ait bir kullanıcı, aşağıdaki komutları kullanarak bir DR RDQM ' nin durumunu ve durumunu görüntüleyebilir:

- `dspm`
- `rdqmstatus`

`mqm` kullanıcısının her iki sunucuda da aynı UID 'ye sahip olması ve `mqm` grubunun her iki sunucuda da aynı GID' ye sahip olması gerekir.

Bir olağanüstü durumdan kurtarma yapılandırmasında birincil ya da ikincil olarak hareket etmek üzere eşlenmiş bir veri kuyruğu yöneticisi (RDQM) yaratmak için **crtmqm** komutunu kullanabilirsiniz.

## Bu görev hakkında

Kullanıcı sudo kullanabiliyorsa, mqm grubundaki bir kullanıcı olarak eşlenmiş bir veri kuyruğu yöneticisi (RDQM) oluşturabilirsiniz. Aksi takdirde RDQM ' yi kök olarak yaratmanız gerekir.

Bir düğümde birincil RDQM DR kuyruk yöneticisi yaratmalısınız. Daha sonra, başka bir düğümde aynı kuyruk yöneticisinin ikincil bir eşgörünümünü yaratmanız gerekir. Birincil ve ikincil eşgörünümler aynı ada sahip olmalı ve aynı miktarda saklama alanı ayrılmalıdır.

Aşağıdaki noktalar, kuyruk yöneticisi dosya sisteminin boyutlandırılmasına ilişkin bazı kılavuzlar sağlar:

1. Bir RDQM kuyruk yöneticisi yaratılırken, kuyruk yöneticisi verilerini ve günlüklerini saklamak için bir dosya sistemi ayrılır. Kuyruk yöneticisinin devam eden etkinliği günlüklerine kaydedebilmesi ve uygulama iletilerini kuyruklara saklayabilmesi için bu dosya sisteminin uygun şekilde boyutlandırılması önemlidir. Dosya sisteminin boyutlandırırken en yoğun ileti sistemi gereksinimlerini, gelecekteki iş yükü artışını ve iletilerin kuyruklarda birikmesine neden olabilecek uygulama kesintilerini göz önünde bulundurun. Kuyruk yöneticisi kurtarma günlüğünün boyutunun hesaplanmasına ilişkin yönergeler için bkz. [“Günlük dosya sistemimi ne kadar büyük yapmalıyım?” sayfa 638](#). Uygulama iletileri için depolama gereksinimleri hesaplanırken, iletilerin boyutu ve sayısı ve bunların MQMD üstbilgisi ve sahip oldukları ileti özellikleri dikkate alınmalı.
2. RDQM kuyruk yöneticisi dosya sistemleri dinamik olarak yeniden boyutlandırılmıyor. Bu gerekiyorsa, RDQM kuyruk yöneticisini daha büyük bir dosya sistemiyle yedeklemeniz ve geri yüklemeniz gerekir, bkz. [“HA RDQM kuyruk yöneticisi için dosya sisteminin yeniden boyutlandırma” sayfa 566](#).
3. MAXDEPTH ve MAXFSIZE gibi yerel kuyruk özniteliklerini kullanarak diskteki tek tek kuyrukların boyutunu sınırlayabilirsiniz. Bkz. [IBM MQ kuyruk dosyalarının değiştirilmesi](#).
4. Devam eden disk kullanımınızı izlemeniz ve dosya sistemi kullanımı kritik duruma gelmeden önce disk kullanımı artarsa uygun şekilde yanıt vermeniz gerekir. Dosya sistemi kullanımı, platform/işletim sistemi yetenekleri kullanılarak ya da [Sistem konularında yayınlanan ölçümler](#) başlıklı konuda açıklanan IBM MQ sistem konularına abone olunarak izlenebilir.

## Yordam

- Birincil DR RDQM yaratmak için:
  - a) Aşağıdaki komutu girin:

```
crtmqm -rr p [-rt (a | s)] -rl Local_IP -ri Recovery_IP -rn Recovery_Name -rp Port
[other_crtmqm_options] [-fs size] QMname
```

Burada:

### -rr p

Kuyruk yöneticisinin birincil eşgörünümünü yaratmakta olduğunuzu belirtir.

### -rt a | s

**-rt s** DR yapılandırmasının zamanuyumlu eşleme kullandığını belirtir, **-rt a** DR yapılandırmasının zamanuyumsuz kopyalama kullandığını belirtir. Zamanuyumsuz kopyalama varsayılan değerdir.

### -rl Yerel\_IP

Bu kuyruk yöneticisinin DR eşlemesinde kullanılacak yerel IP adresini belirler.

### -ri Recovery\_IP

Kuyruk yöneticisinin ikincil eşgörünümünü barındıran sunucuda eşleme için kullanılan arabirimin IP adresini belirler.

### **-rn Tanıma-Adı**

Kuyruk yöneticisinin ikincil eşgörünümünü barındıran sistemin adını belirtir. Ad, o sunucuda `uname -n` komutunu çalıştırırsanız döndürülen değerdir. Bu sunucuda belirtik olarak bir ikincil kuyruk yöneticisi yaratmalısınız.

### **-rp Kapı**

DR eşlemesi için kullanılacak kapıyı belirtir.

### **other\_crtmqm\_options**

İsteğe bağlı olarak aşağıdaki genel **crtmqm** seçeneklerinden birini ya da birkaçını belirtebilirsiniz:

- -z
- -q
- -c *Metin*
- -d *DefaultTransmissionKuyruk*
- -h *MaxHandles*
- -g *ApplicationGroup*
- -oa *kullanıcı|grup*
- -t *TrigInt*
- -u *DeadQ*
- -x *MaxUMsgs*
- -lp *LogPri*
- -ls *LogSec*
- -lc | -l
- -lla | -lln
- -lf *LogFileSize*
- -p *Kapı*

### **-fs boyut**

İsteğe bağlı olarak, kuyruk yöneticisi için yaratılacak dosya sisteminin boyutunu, yani drbdpool birim grubunda yaratılan mantıksal birimin boyutunu belirtir. Anlık görüntü işlemine geri dönmeyi desteklemek için bu boyuttaki başka bir mantıksal birim de oluşturulur, bu nedenle DR RDQM için toplam depolama alanı burada belirtilenden iki kat fazla olur.

*Boyut* , GB olarak belirtilen sayısal bir değerdir. MB cinsinden bir değeri, M karakterinin izlediği değeri girerek belirtebilirsiniz. Örneğin, 3 GB ' lik bir dosya sistemi boyutu belirtmek için 3 girin. 1024 MB dosya sistemi boyutunu belirtmek için 1024M girin. (Açıkça GB ' ye bir G soneki de ekleyebilirsiniz.)

### **QMname**

Eşlenen veri kuyruğu yöneticisinin adını belirtir. Bu ad büyük ve küçük harfe duyarlıdır.

Komut tamamlandıktan sonra, kuyruk yöneticisinin ikincil eşgörünümünü yaratmak için ikincil düğümde girmeniz gereken komutu verir. İkincil kuyruk yöneticisini yaratmak üzere ikincil düğümde çalıştırmanız gereken **crtmqm** komutunu almak için birincil düğümünüzdeki **rdqmdx** komutunu da kullanabilirsiniz, bkz. [“DR RDIMM ' lerinin birincil ve ikincil özelliklerinin yönetilmesi” sayfa 594.](#)

- İkincil bir DR RDQM yaratmak için:

a) RDQM ' nin ikincil eşgörünümünü barındıracak düğümde şu komutu girin:

```
crtmqm -rr s [-rt (a | s)] -rl Local_IP -ri Primary_IP -rn Primary_Name -rp Port
[other_crtmqm_options] [-fs size] QMname
```

Burada:

### **-rr s**

Kuyruk yöneticisinin ikincil eşgörünümünü yaratmakta olduğunuzu belirtir.

### -rt a | s

-rt s DR yapılandırmasının zamanuyumlu eşleme kullandığını belirtir, -rt a DR yapılandırmasının zamanuyumsuz kopyalama kullandığını belirtir.

### -rl Yerel\_IP

Bu kuyruk yöneticisinin DR eşlemesinde kullanılacak yerel IP adresini belirler.

### -ri İlk\_IP

Kuyruk yöneticisinin birincil örneğini barındıran sunucuda eşleme için kullanılan arabirimin IP adresini belirler.

### -rn Birincil\_Adi

Kuyruk yöneticisinin birincil eşgörünümünü barındıran sistemin adını belirtir. Ad, o sunucuda uname -n komutunu çalıştırırsanız döndürülen değerdir.

### -rp Kapı

DR eşlemesi için kullanılacak kapıyı belirtir.

### other\_crtmqm\_options

İsteğe bağlı olarak aşağıdaki genel crtmqm seçeneklerinden birini ya da birkaçını belirtebilirsiniz:

- -z

### -fs boyut

Kuyruk yöneticisi için yaratılacak dosya sisteminin boyutunu, yani drbdpool birim grubunda yaratılan mantıksal birimin boyutunu belirtir. Birincil kuyruk yöneticisini yaratırken varsayılan olmayan bir büyüklük belirttiyseniz, burada aynı değeri belirtmeniz gerekir.

Boyut , GB olarak belirtilen sayısal bir değerdir. MB cinsinden bir değeri, M karakterinin izlediği değeri girerek belirtebilirsiniz. Örneğin, 3 GB ' lik bir dosya sistemi boyutu belirtmek için 3 girin. 1024 MB dosya sistemi boyutunu belirtmek için 1024M girin. (Açıkça GB ' ye bir G soneki de ekleyebilirsiniz.)

### QMname

Eşlenen veri kuyruğu yöneticisinin adını belirtir. Bu, kuyruk yöneticisinin birincil eşgörünümü için belirttiğiniz adla aynı olmalıdır. Adın büyük ve küçük harfe duyarlı olduğunu unutmayın.

## Sonraki adım

Kuyruk yöneticinizin birincil ve ikincil yönetim ortamlarını yarattıktan sonra, her ikisinin de doğru olup olmadığını denetlemek için her iki düğümdeki durumu denetlemeniz gerekir. Her iki düğümde **rdqmstatus** komutunu kullanın. Düğümlerin, "[DR RDQM durumunu görüntüleme](#)" sayfa 596 içinde açıklandığı gibi normal durumu görüntülüyor olması gerekir. Bu durumu görüntülemiyorlarsa, ikincil eşgörünümü silin ve doğru bağımsız değişkenleri kullanmaya özen göstererek yeniden yaratın.

## İlgili başvurular

[crtmqm](#)

**Linux** DR RDQM ' in silinmesi

Olağanüstü durumdan kurtarma eşlenmiş bir veri kuyruğu yöneticisini (RDQM) silmek için **dltmqm** komutunu kullanırsınız.

## Bu görev hakkında

RDQM 'in birincil ve ikincil düğümlerinde RDQM' yi silmek için komutu çalıştırmanız gerekir. Önce RDQM sona erdirilmelidir. Kullanıcı gerekli sudo ayrıcalıklarına sahipse, komutu mqm kullanıcısı olarak çalıştırabilirsiniz. Ters durumda, komutu kök olarak çalıştırmanız gerekir.

## Yordam

- Bir DR RDQM ' yi silmek için şu komutu girin:

```
dltmqm RDQM_name
```



## İlgili başvurular

[dlmqm](#)

Linux

MQ Adv.

*Bir kuyruk yöneticisinin DR RDQM kuyruk yöneticisi olacak şekilde geçirilmesi*

Var olan bir kuyruk yöneticisini, kalıcı verilerini yedekleyerek ve daha sonra verileri aynı ada sahip yeni oluşturulan bir RDQM kuyruk yöneticisine geri yükleyerek olağanüstü durumdan kurtarma (DR) eşlenmiş veri kuyruğu yöneticisi (RDQM) olacak şekilde geçirebilirsiniz.

## Bu görev hakkında

DR Eşlenmiş Veri Kuyruğu Yöneticileri için özel olarak ayrılmış bir mantıksal birim (dosya sistemi) ve disk eşleme yapılandırması gerekir. Bu bileşenler yalnızca yeni bir kuyruk yöneticisi yaratıldığında yapılandırılır. Var olan bir kuyruk yöneticisi, kalıcı verilerini yedekleyerek RDQM 'yi kullanmak üzere geçirilebilir, daha sonra verileri aynı ada sahip yeni yaratılan bir RDQM kuyruk yöneticisine geri yükleyebilir. Bu yordam, yedekleme yaratıldığı sırada kuyruk yöneticisi yapılanışını, durumunu ve kalıcı iletileri korur.

**Not:** You can only migrate a queue manager from a version of IBM MQ that is the same as or lower than the version where RDQM is installed. İşletim sistemi ve mimari de aynı olmalıdır. Ters durumda, hedef altyapınızda yeni bir kuyruk yöneticisi yaratmanız gerekir; bkz. [Kuyruk yöneticisinin farklı bir işlem sistemine taşınması](#).

Bir kuyruk yöneticisini geçirmeden önce aşağıdaki koşulları karşılamamız gerekir:

- Olağanüstü durumdan kurtarma gereksinimlerinizi değerlendirin ve bkz. [“RDQM olağanüstü durumdan kurtarma” sayfa 581](#).
- Kuyruk yöneticisine bağlanan uygulamaları ve kuyruk yöneticilerini gözden geçirin. Bağlantıları kuyruk yöneticisinin çalıştığı RDQM düğümüne yönlendirmek için gereken değişiklikleri göz önünde bulundurun.
- Seçtiğiniz yapılandırma için var olan RDQM düğümlerini yetkilendirin ya da tanımlayın. RDQM 'ye ilişkin sistem gereksinimleri hakkında bilgi için bkz. [“RDQM DR çözümü için gereksinimler” sayfa 584](#).
- Her düğümde RDQM özelliğini içeren IBM MQ Advanced ürününü kurun.
- İsteğe bağlı olarak, daha sonra silinebilecek bir test kuyruğu yöneticisini kullanarak RDQM yapılandırmasını doğrulayın. Kuyruk yöneticisini geçirmeden önce sorunları saptamak ve çözmek için yapılanışın sınanması önerilir.
- Kuyruk yöneticisi için güvenlik yapılandırmasını gözden geçirin ve ardından her bir RDQM düğümünde gerekli yerel kullanıcıları ve grupları eşin.
- API çıkışlarının, kanal çıkışlarının ya da veri dönüştürme çıkışlarının kullanılıp kullanılmadığını belirlemek için kuyruk yöneticisini ve kanal yapılandırmasını gözden geçirin. Her bir RDQM düğümünde gerekli çıkışları kurun.
- Tanımlanmış kuyruk yöneticisi hizmetlerini gözden geçirin, ardından her bir RDQM düğümünde gerekli süreçleri kurun ve yapılandırın.

## Yordam

1. Var olan kuyruk yöneticisini yedekle:

- a) Bir bekleme sona erdirmeye komutu `endmqm -wya` da hemen sona erdirmeye komutu `endmqm -ivererek`, var olan kuyruk yöneticisini durdurun. Bu adım, yedekteki verilerin tutarlı olmasını sağlamak için önemlidir.
- b) IBM MQ yapılanış kütüğünü ( `mqs.ini`) görüntüleyerek kuyruk yöneticisi verileri dizininin yerini saptayın. Linux üzerinde bu dosya, `/var/mqm` dizininde bulunur. `mqs.ini` hakkında daha fazla bilgi için bkz. [“IBM MQ yapılandırma dosyası, mqs.ini” sayfa 85](#).

Dosyada kuyruk yöneticisine ilişkin `QueueManager` kısmı bulun. Bölüm `DataPath` adlı bir anahtar içeriyorsa, değeri kuyruk yöneticisi veri dizinidir. Anahtar yoksa, kuyruk yöneticisi veri dizini `Prefix` ve `Directory` anahtarlarının değerleri kullanılarak belirlenebilir. Kuyruk yöneticisi veri dizini, `prefix/qmgrs/directory` biçiminde bu değerlerin bir birleşimidir. `QueueManager` kısmı hakkında daha fazla bilgi için bkz. [“mqs.ini dosyasının QueueManager kısmı” sayfa 95](#).

- c) Kuyruk yöneticisi veri dizininin bir yedeğini oluşturun. Linux' da bunu **tar** komutunu kullanarak yapabilirsiniz. Örneğin, bir kuyruk yöneticisine ilişkin veri dizinini yedeklemek için aşağıdaki komutu kullanabilirsiniz. Komutun tek nokta (nokta) olan son deęiřtirgesine dikkat edin:

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- d) IBM MQ Kuyruk yöneticisi yapılanıř kütüğünü qm.inigörüntüleyerek kuyruk yöneticisi günlüğü dizininin yerini saptayın. Bu dosya, kuyruk yöneticisi veri dizininde bulunur. Dosya hakkında daha fazla bilgi için bkz. “Kuyruk yöneticisi yapılanıř dosyaları, qm.ini” sayfa 97.

Kuyruk yöneticisi günlük dizini, Log kısmı içindeki LogPath anahtarının deęeri olarak tanımlanır. Kıta hakkında bilgi için bkz. “qm.ini dosyasının günlük kısmı” sayfa 129.

- e) Kuyruk yöneticisi günlük dizininin bir yedeğini yaratın. Linuxüzerinde, tar komutunu kullanarak bunu yapabilirsiniz. Örneğin, bir kuyruk yöneticisine ilişkin günlük dizinini yedeklemek için aşağıdaki komutu kullanabilirsiniz. Komutun tek nokta (nokta) olan son deęiřtirgesine dikkat edin:

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- f) Kuyruk yöneticisi veri dizininde bulunmuyorsa, kuyruk yöneticisi tarafından kullanılan sertifika havuzlarının yedeğini oluşturun. Hem anahtar veritabanı kütüğünün, hem de parola saklama kütüğünün yedeklendiğini doęrulayın. Kuyruk yöneticisi anahtar havuzu hakkında bilgi için bkz. [SSL/TLS anahtar havuzu ve Kuyruk yöneticisi için anahtar havuzunun bulunması](#). Kuyruk yöneticisi AMS Message Channel Agent (MCA) müdahalesi kullanacak şekilde yapılandırıldıysa, AMS anahtar deposunun bulunmasıyla ilgili bilgi için [Message Channel Agent \(MCA\) interception](#) başlıklı konuya bakın.
- g) Var olan kuyruk yöneticisi artık gerekli olmadığından silinebilir. Ancak, mümkünse, hedef sistemde başarıyla geri yüklendikten sonra varolan kuyruk yöneticisini silmeniz gerekir. Silme işleminin ertelenmesi, geçiř işlemleri başarıyla tamamlanmazsa kuyruk yöneticisinin yeniden başlatılabilmesini saęlar.

**Not:** Var olan kuyruk yöneticisinin silinmesini ertelerseniz, yeniden başlatmayın. Geçiř sırasında yapılanıřında ya da durumunda yapılacak başka deęiřiklikler kaybedildięi için kuyruk yöneticisinin sona erdirilmeye devam etmesi önemlidir.

## 2. Birincil RDQM düęümünü hazırla:

- a) Yedeklediğiniz kuyruk yöneticisiyle aynı ada sahip yeni bir RDQM kuyruk yöneticisi yaratın. **crtmqm** tarafından RDQM kuyruk yöneticisi için ayrılan dosya sisteminin, var olan kuyruk yöneticisine ilişkin verileri, birincil günlükleri ve ikincil günlükleri ve gelecekteki genişletme için ek alan içerecek kadar büyük olduğundan emin olun. Bir RDQM kuyruk yöneticisinin nasıl yaratılacağı hakkında bilgi için bkz. “[Olaęanüstü durumdan kurtarma RDQM ' in yaratılması](#)” sayfa 586.
- b) Kuyruk yöneticisi için birincil RDQM düęümünü belirleyin. Birincil düęümün nasıl belirleneceęine ilişkin bilgi için bkz. [rdqmstatus](#) (RDQM durumunun görüntülenmesi).
- c) Birincil RDQM düęümünde RDQM kuyruk yöneticisi başlatıldıysa, **endmqm -w** ya da **endmqm -i** komutunu kullanarak durdurun.
- d) RDQM kuyruk yöneticisine ilişkin veri ve günlük dizinlerinin konumunu belirleyin ( 1b ve 1dadımlarında açıklanan yöntemleri kullanın).
- e) RDQM kuyruk yöneticisi verilerinin ve günlük dizinlerinin içeriğini silin, ancak dizinlerin içeriğini silmeyin.

## 3. Birincil RDQM düęümünde kuyruk yöneticisini geri yükleyin:

- a) Kuyruk yöneticisi verileri ve günlük dizinlerinin yedeklerini birincil RDQM düęümüne ve kuyruk yöneticisi tarafından kullanılan sertifika havuzlarının ayrı yedeklerine kopyalayın.
- b) Yeni RDQM kuyruk yöneticisi için kuyruk yöneticisi veri dizininin yedeğini boş veri dizinine geri yükleyip dosya sahiplięinin ve izinlerinin korunduğundan emin olun. Yedekleme, 1c adımıdaki örnek tar komutu kullanılarak yaratıldıysa, kök kullanıcı tarafından geri yüklemek için aşağıdaki komut kullanılabilir:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- c) Yeni RDQM kuyruk yöneticisi için kuyruk yöneticisi günlük dizininin yedeğini boş günlük dizinine geri yükleyip dosya sahipliğinin ve izinlerinin korunduğundan emin olun. Yedekleme, 1e adımındaki örnek tar komutu kullanılarak yaratıldıysa, kök kullanıcı tarafından geri yüklemek için aşağıdaki komut kullanılabilir:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- d) RDQM kuyruk yöneticisine ilişkin veri dizininde geri yüklenen kuyruk yöneticisi yapılandırma dosyasını (qm.ini) düzenleyin. Log kısmı içindeki LogPath anahtarının değerini, RDQM kuyruk yöneticisine ilişkin günlük dizinini belirtecek şekilde güncelleyin.

Yapılandırma dosyasında tanımlanan diğer dosya yollarını gözden geçirin ve gerekirse bunları güncelleyin. Örneğin, aşağıdaki yolları güncelleniz gerekebilir:

- Tanılama ileti hizmetleri tarafından oluşturulan hata günlüğü dosyalarının yolu.
- Kuyruk yöneticisinin gerektirdiği çıkışların yolu.
- Kuyruk yöneticisi bir XA hareket eşgüdümçüye, anahtar yükleme dosyalarının yolu.

- e) Kuyruk yöneticisi AMS Message Channel Agent (MCA) müdahalesi kullanacak şekilde yapılandırıldıysa, AMS anahtar deposunu yeni RDQM kuruluşuna kopyalayın ve yapılandırmayı gözden geçirin ve güncelleyin. Anahtar deposu her RDQM düğümünde kullanılabilir olmalıdır; bu nedenle, kuyruk yöneticisi için eşlenen dosya sisteminde bulunmuyorsa, bunun yerine her bir düğüme kopyalanmalıdır. Daha fazla bilgi için bkz. [Message Channel Agent \(MCA\) interception](#).

- f) Kuyruk yöneticisinin **dspm** komutu tarafından görüntülendiğini ve durumunun sona erdirildi olarak bildirildiğini doğrulayın. Aşağıdaki örnek, bir RDQM DR kuyruk yöneticisine ilişkin örnek çıktıyı göstermektedir:

```
$ dspm -o status -o dr
QMNAME(QM1) STATUS(Ended normally) DRROLE(Primary)
```

- g) Kuyruk yöneticisine ilişkin durumu görüntülemek için **rdqmstatus** komutunu kullanarak, geri yüklenen kuyruk yöneticisi verilerinin ikincil RDQM düğümlerine eşlendiğini doğrulayın. DR durumu, her düğümde Normal olarak bildirilmelidir. Aşağıdaki örnek, bir RDQM DR kuyruk yöneticisine ilişkin örnek çıktıyı göstermektedir:

```
$ rdqmstatus -m QM1
Queue manager status:           Ended normally
Queue manager file system:     51MB used, 1.0GB allocated [5%]
DR role:                        Primary
DR status:                       Normal
DR type:                         Synchronous
DR port:                          3000
DR local IP address:            192.168.20.1
DR remote IP address:          192.168.20.2
```

- h) Birincil RDQM düğümünde kuyruk yöneticisini başlatın.

- i) Kuyruk yöneticisine bağlanın ve kuyruk yöneticisi sertifika havuzunun yeni konumunu belirtmek için SSLKEYR kuyruk yöneticisi özneliğinin değerini güncelleyin. Varsayılan olarak, bu özneliğin değeri *queue\_manager\_data\_directory/ssl/key* olarak ayarlanır. Sertifika havuzu, her RDQM düğümünde aynı konumda bulunmalıdır. Havuz, kuyruk yöneticisine ilişkin eşlenmiş dosya sisteminde bulunmuyorsa, bunun yerine her düğüme kopyalanması gerekir.
- j) Kuyruk yöneticisi için IBM MQ nesne tanımlamalarını gözden geçirin ve aşağıdaki nesnelere de içinde olmak üzere, değiştirilen ağ ayarlarına, IBM MQ kuruluş dizinine ya da kuyruk yöneticisi veri dizinine başvuran nesne özneliklerinin değerini güncelleyin:

- Dinleyiciler tarafından kullanılan yerel IP adresleri (IPADDR özneliği).
- Kanallar tarafından kullanılan yerel IP adresleri (LOCLADDR özneliği).
- Küme alıcı kanalları için tanımlanan yerel IP adresleri (CONNNAME özneliği).
- İletişim bilgileri nesnelere için tanımlanan yerel IP adresleri (GRPADDR özneliği).
- Süreç ve hizmet nesnesi tanımlamaları için tanımlanan sistem yolları.

- k) Değişikliklerin etkili olmasını sağlamak için kuyruk yöneticisini durdurun ve yeniden başlatın.
- l) Uzak kuyruk yöneticileri için 3j adımını ve geçirilen kuyruk yöneticisine (aşağıdakiler de içinde olmak üzere) bağlanan uygulamalara ilişkin eşdeğer ayarları yineleyin:
- Kanal bağlantısı adları (CONNNAME özniteliği).
  - Kuyruk yöneticisinden gelen bağlantıları IP adresine ya da anasistem adına göre kısıtlayan kanal kimlik doğrulama kuralları.
  - İstemci kanal tanımlama çizelgeleri (CCDT), etki alanı adı ayarları (DNS), ağ yöneltme ya da eşdeğer bağlantı bilgileri.
- m) Gerekli yapılandırmanın başarıyla oluşturulduğundan emin olmak için kuyruk yöneticisinin her bir RDQM düğümüne yönetilen bir hata durumunda yedek sisteme geçiş işlemi gerçekleştirin, bkz. [“Kurtarma düğümüne geçiş” sayfa 601.](#)

#### *DR RDQM kuyruk yöneticisi için dosya sistemini yeniden boyutlandırma*

Var olan bir olağanüstü durumdan kurtarma (DR) eşlenmiş veri kuyruğu yöneticisine (RDQM) ilişkin dosya sistemini yeniden boyutlandırmak için, kalıcı verilerini yedekledikten sonra, verileri aynı ada sahip, ancak farklı bir dosya sistemine sahip yeni oluşturulan bir RDQM kuyruk yöneticisine geri yükleyin.

## Bu görev hakkında

DR eşlenmiş veri kuyruğu yöneticileri için özel olarak ayrılmış bir mantıksal birim (dosya sistemi) ve disk eşleme yapılandırması gerekir. Bu bileşenler yalnızca yeni bir kuyruk yöneticisi yaratıldığında yapılandırılır. Dosya sistemi yaratıldıktan sonra yeniden boyutlandırılmaz; her düğümde aynı büyüklüğe sahip olmalıdır. Var olan bir eşlenmiş veri kuyruğu yöneticisine (RDQM) ilişkin dosya sistemini yeniden boyutlandırmak için, kalıcı verilerini yedekleyebilir ve daha sonra, aynı ada sahip, ancak farklı bir dosya sistemine sahip yeni yaratılan bir RDQM kuyruk yöneticisine geri yükleyebilirsiniz. Bu yordam, yedekleme yaratıldığı sırada kuyruk yöneticisi yapılanışını, durumunu ve kalıcı iletileri korur.

## Yordam

1. Birincil RDQM düğümünde var olan RDQM kuyruk yöneticisini yedekleyin:
  - a) Kuyruk yöneticisi için birincil RDQM düğümünü belirleyin. Birincil düğümün nasıl belirleneceğine ilişkin bilgi için bkz. [rdqmstatus \(RDQM durumunun görüntülenmesi\)](#).
  - b) Birincil RDQM düğümünde RDQM kuyruk yöneticisi başlatıldıysa, **endmqm -w** ya da **endmqm -i** komutunu kullanarak durdurun.
  - c) IBM MQ yapılanış kütüğünü ( `mqs.ini` ) görüntüleyerek kuyruk yöneticisi verileri dizininin yerini saptayın. Linux'üzünde bu dosya, `/var/mqm` dizininde bulunur. `mqs.ini` hakkında daha fazla bilgi için bkz. [“IBM MQ yapılandırma dosyası, mqs.ini” sayfa 85.](#)

Dosyada kuyruk yöneticisine ilişkin `QueueManager` kısmı bulun. Kuyruk yöneticisi veri dizini, `DataPath` adlı anahtarın değeridir. `QueueManager` kısmı hakkında daha fazla bilgi için bkz. [“mqs.ini dosyasının QueueManager kısmı” sayfa 95.](#)

- d) Kuyruk yöneticisi veri dizininin bir yedeğini oluşturun. Linux' da bunu **tar** komutunu kullanarak yapabilirsiniz. Örneğin, bir kuyruk yöneticisine ilişkin veri dizinini yedeklemek için aşağıdaki komutu kullanabilirsiniz. Komutun tek bir nokta (.) karakteri olan son değiştirgesine dikkat edin:

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- e) IBM MQ Kuyruk yöneticisi yapılanış kütüğünü `qm.ini` görüntüleyerek kuyruk yöneticisi günlüğü dizininin yerini saptayın. Bu dosya, kuyruk yöneticisi veri dizininde bulunur. Dosya hakkında daha fazla bilgi için bkz. [“Kuyruk yöneticisi yapılanış dosyaları, qm.ini” sayfa 97.](#)

Kuyruk yöneticisi günlük dizini, Günlük kısmı içindeki `LogPath` anahtarının değeri olarak tanımlanır. Kıta hakkında bilgi için bkz. [“qm.ini dosyasının günlük kısmı” sayfa 129.](#)

- f) Kuyruk yöneticisi günlük dizininin bir yedeğini yaratın. Linux' da bunu **tar** komutunu kullanarak yapabilirsiniz. Örneğin, bir kuyruk yöneticisine ilişkin günlük dizini yedeklemek için aşağıdaki komutu kullanabilirsiniz. Komutun tek bir nokta (.) karakteri olan son deęiştirgesine dikkat edin:

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- g) Var olan RDQM kuyruk yöneticisini silin.

2. Kuyruk yöneticisini gereken büyüklükte bir dosya sistemi ile geri yükleyin:

- a) Yedeklediğiniz kuyruk yöneticisiyle aynı ada sahip yeni bir RDQM kuyruk yöneticisi yaratın. **crtmqm** tarafından RDQM kuyruk yöneticisi için ayrılan dosya sisteminin gerekli boyut olduğundan ve var olan kuyruk yöneticisine ilişkin verileri, birincil günlükleri ve ikincil günlükleri içerecek kadar büyük olduğundan ve gelecekteki genişletme için ek alan bulunduğundan emin olun. Bir RDQM kuyruk yöneticisinin nasıl yaratılacağı hakkında bilgi için bkz. "Olağanüstü durumdan kurtarma RDQM ' in yaratılması" sayfa 586.
- b) Kuyruk yöneticisi için birincil RDQM düğümünü belirleyin. Birincil düğümün nasıl belirleneceğine ilişkin bilgi için bkz. [rdqmstatus](#) (RDQM durumunun görüntülenmesi).
- c) Birincil RDQM düğümünde RDQM kuyruk yöneticisi başlatıldıysa, **endmqm -w** ya da **endmqm -i** komutunu kullanarak durdurun.
- d) Birincil RDQM düğümünde, RDQM kuyruk yöneticisine ilişkin veri ve günlük dizinlerinin yeni konumunu belirleyin ( 1c ve 1eadımlarında açıklanan yöntemleri kullanın).
- e) Birincil RDQM düğümünde, RDQM kuyruk yöneticisi verilerinin ve günlük dizinlerinin içeriğini silin, ancak dizinleri silmeyin.
- f) Birincil RDQM düğümünde, kuyruk yöneticisi veri dizininin yedeğini yeni RDQM kuyruk yöneticisi için boş veri dizinine geri yükleyip dosya sahipliğinin ve izinlerinin korunduğundan emin olun. Yedekleme, 1d adımıdaki örnek **tar** komutu kullanılarak yaratıldıysa, kök kullanıcı tarafından geri yüklemek için aşağıdaki komut kullanılabilir:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- g) Birincil RDQM düğümünde, kuyruk yöneticisi günlük dizininin yedeğini yeni RDQM kuyruk yöneticisi için boş günlük dizinine geri yükleyip dosya sahipliğinin ve izinlerinin korunduğundan emin olun. Yedekleme, 1f adımıdaki örnek **tar** komutu kullanılarak yaratıldıysa, kök kullanıcı tarafından geri yüklemek için aşağıdaki komut kullanılabilir:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- h) Birincil RDQM düğümünde, yeni RDQM kuyruk yöneticisinin veri dizininde geri yüklenen kuyruk yöneticisi yapılandırma dosyasını ( qm . ini) düzenleyin. Adım 2d' de belirlediğiniz yeni RDQM kuyruk yöneticisine ilişkin günlük dizinini belirtmek için Log kısmına ilişkin LogPath anahtarının değerini güncelleyin. Yapılandırma dosyasında tanımlanan diğer dosya yollarını gözden geçirin ve gerekirse bunları güncelleyin. Örneğin, aşağıdaki yolları güncellemeniz gerekebilir:

- Tanılama ileti hizmetleri tarafından oluşturulan hata günlüğü dosyalarının yolu.
- Kuyruk yöneticisinin gerektirdiği çıkışların yolu.
- Kuyruk yöneticisi bir XA hareket eşgüdümçüye, anahtar yükleme dosyalarının yolu.

- i) Kuyruk yöneticisinin **dspmq** komutu tarafından görüntülendiğini ve durumunun endedolarak bildirildiğini doğrulayın. Aşağıdaki örnek, bir RDQM DR kuyruk yöneticisine ilişkin örnek çıktıyı göstermektedir:

```
$ dspmq -o status -o dr
QMNAME(QM1) STATUS(Ended normally) DR(Primary)
```

- j) Kuyruk yöneticisinin durumunu görüntülemek için **rdqmstatus** komutunu kullanarak, geri yüklenen kuyruk yöneticisi verilerinin ikincil RDQM düğümüne eşlendiğini doğrulayın. DR durumu, her düğümde Normal olarak bildirilmelidir. Aşağıdaki örnek, birincil düğümdeki bir RDQM DR kuyruk yöneticisine ilişkin örnek çıktıyı göstermektedir:

```
$ rdqmstatus -m QM1
Queue manager status:      Running
CPU:                       0.00
Memory:                    123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role:                   Primary
DR status:                 Normal
DR type:                   Synchronous
DR port:                   3000
DR local IP address:       192.168.20.1
DR remote IP address:      192.168.20.2
```

Aşağıdaki örnek, kurtarma düğümündeki bir RDQM DR kuyruk yöneticisine ilişkin örnek çıktıyı göstermektedir:

```
Queue manager status:      Ended immediately
DR role:                   Secondary
DR status:                 Normal
DR port:                   3000
DR local IP address:       192.168.20.2
DR remote IP address:      192.168.20.1
```

- k) Birincil RDQM düğümünde kuyruk yöneticisini başlatın.
- l) Gerekli yapılandırmanın başarıyla oluşturulduğundan emin olmak için kuyruk yöneticisinin kurtarma düğümüne geçmesini sağlayın, bkz. [“Kurtarma düğümüne geçiş” sayfa 601](#).

#### *Kalıcı uygulama durumu saklanıyor*

Diğer kuyruk yöneticisi verileriyle birlikte uygulamalarla ilgili kalıcı durum bilgilerini saklayabilirsiniz.

Her IBM MQ kuyruk yöneticisinin, hem kuyruk verilerini hem de kurtarma günlüğünü içeren, kalıcı durumu için özel olarak ayrılmış bir dosya sistemi vardır. Bir RDQM yapılandırmasında dosya sistemi, Linux sistemleri (düğümler) arasında eşlenen bir mantıksal birim tarafından desteklenir. Dosya sistemi, uygulamalarınıza ilişkin kalıcı durum bilgilerini saklamak için kullanabileceğiniz bir `userdata` dizini içerir. Bu nedenle, eşlenmiş bir veri kuyruğu yöneticisi RDQM yapılandırmanızdaki başka bir düğümde çalışmak üzere hareket ettiğinde, kuyruk yöneticisi bağlamının yanı sıra uygulama bağlamını da kullanabilirsiniz. Bkz. [Directory content on Unix and Linux Systems](#).

Uygulama durumunu `userdata` dizininde saklamayı seçerseniz, bu yere yazılan verilerin kuyruk yöneticisine ayrılan kullanılabilir disk alanını kullanabileceğini bilmeniz gerekir. Kuyruk yöneticisinin kuyruk verilerini, günlüklerini ve diğer kalıcı durum bilgilerini yazması için yeterli disk yeri kaldığından emin olmanız gerekir.

`userdata` dizini `mqm` kullanıcı ve grup yetisiyle sahiptir ve kullanıcıların IBM MQ yönetici grubunda (`mqm`) olmalarına gerek kalmadan bu dizine erişebilmeleri için herkes tarafından okunabilir. `userdata` dizininin izinlerini değiştiremezsiniz, ancak gerek duyduğunuz sahiplik ve izinlerle içerik oluşturabilirsiniz.

Bir RDQM kuyruk yöneticisi hata durumunda yedek sisteme geçiş sırasında kuyruk yöneticisi sona erdirilir ve dosya sistemi yürürlükteki RDQM düğümünde kaldırılır. Daha sonra dosya sistemi sisteme bağlanır ve kuyruk yöneticisi RDQM yapılandırmasındaki başka bir düğümde yeniden başlatılır. Bir işlemin dosyalarından biri için açık bir tanıtıcısı varsa, dosya sistemi kaldırılmaz. Bir kuyruk yöneticisinin hata durumunda yedek sisteme geçiş işleminin tamamlanabilmesini sağlamak için, kuyruk yöneticisi dosya sistemi bağlantısı kaldırılmazsa, açık dosya tanıtıcısı olan işlemlere bir SIGTERM sinyali gönderilir ve ardından açık tutamaçlar serbest bırakılmazsa, SIGKILL işareti gönderilir. Uygulamalarınız, SIGTERM 'ye doğru yanıt verecek şekilde tasarlanmalıdır. Uygulamalar ya da işlemler kuyruk yöneticisi hizmeti olarak yapılandırıldıysa, yönetilen hata durumunda yedek sisteme geçiş sırasında, dosya sistemi kaldırılmadan önce kuyruk yöneticisi kapatılırken sona erdirilebilirler. Bir uygulama ya da işlem kuyruk yöneticisi hizmeti olarak yapılandırılmadıysa ya da çekirdek kaybı gibi yönetilmeyen bir hata durumunda yedek sisteme geçiş gerçekleşirse, dosya sistemini serbest bırakmak için sinyaller gönderilebilir.

#### **Linux DR RDIMM ' lerinin birincil ve ikincil özelliklerinin yönetilmesi**

İkincil bir olağanüstü durumdan kurtarma eşlenmiş veri kuyruğu yöneticisini (DR RDQM) birincil DR RDQM ' ye değiştirebilirsiniz. Birincil bir eşgörünümü ikincil bir örneğe de değiştirebilirsiniz.



## Bu görev hakkında

Bir RDQM 'nin ikincil eşgörünümünü birincil eşgörünüme değiştirmek için **rdqmdr** komutunu kullanın. Herhangi bir nedenle birincil eşgörünümünüzü kaybederseniz bu işlemi tamamlamanız gerekebilir. Bundan sonra kuyruk yöneticisini başlatabilir ve kurtarma düğümünde çalıştırmaya devam edebilirsiniz.

Bir RDQM 'nin birincil eşgörünümünü ikincil eşgörünümüne değiştirmek için **rdqmdr** komutunu da kullanabilirsiniz. Örneğin, sisteminizi yeniden yapılandırıyorsanız, bu işlemi tamamlamanız gerekebilir.

Birincil kuyruk yöneticisinde **rdqmdr** komutunu kullanarak, kurtarma düğümünüzdeki o kuyruk yöneticisinin ikincil bir eşgörünümünü yaratmak için gereken komutu tam olarak alabilirsiniz.

Kullanıcı sudo kullanabiliyorsa, **rdqmdr** komutunu mqm grubunda bir kullanıcı olarak kullanabilirsiniz. Ters durumda, kök olarak oturum açmanız gerekir.

## Yordam

- DR RDQM 'nin ikincil eşgörünümünü birincil eşgörünüme değiştirmek için şu komutu girin:

```
rdqmdr -m QMname -p
```

Kuyruk yöneticisinin birincil yönetim ortamı çalışmaya devam ediyorsa ve DR eşleme bağlantısı çalışmaya devam ediyorsa bu komut başarısız olur.

- Kuyruk yöneticisinin birincil yönetim ortamını ikincil yönetim ortamına değiştirmek için aşağıdaki komutu girin:

```
rdqmdr -m QMname -s
```

- Bir kuyruk yöneticisinin ikincil yönetim ortamını yapılandırmak için gereken **crtmqm** komutunu görüntülemek üzere birincil düğümünüze şu komutu girin:

```
rdqmdr -d -m QMname
```

RD RDQM 'nin ikincil eşgörünümünü yaratmak için ikincil düğümünüzdeki döndürülen **crtmqm** komutunu girebilirsiniz.

## Linux DR RDQM 'nin durumunun başlatılması, durdurulması ve görüntülenmesi

Olağanüstü durumdan kurtarma eşlenmiş veri kuyruğu yöneticisinin (DR RDQM) geçerli durumunu başlatmak, durdurmak ve görüntülemek için standart IBM MQ denetim komutlarının değişkenlerini kullanabilirsiniz.

## Bu görev hakkında

Eşlenmiş bir veri kuyruğu yöneticisinin (RDQM) geçerli durumunu mqm grubuna ait bir kullanıcı olarak başlatan, durduran ve görüntüleyen komutları çalıştırmanız gerekir.

Birincil düğümde bir kuyruk yöneticisini (yani, kuyruk yöneticisinin çalışmakta olduğu düğüm) başlatmak ve durdurmak için komutları çalıştırmanız gerekir.

## Yordam

- Bir DR RDQM 'yi başlatmak için RDQM' in birincil düğümünde şu komutu girin:

```
strmqm qmname
```

Burada *qmname* , başlatmak istediğiniz RDQM 'nin adıdır.

- Bir RDQM 'yi durdurmak için RDQM' in birincil düğümünde şu komutu girin:

```
endmqm qmname
```

Burada *qmname* , durdurmak istediğiniz RDQM 'nin adıdır.

- Bir RDQM ' nin durumunu görüntülemek için şu komutu girin:

```
dspmqr -m QMname
```

Çıkış olan durum bilgileri, komutu RDQM ' in birincil ya da ikincil düğümünde çalıştırıp çalıştırmadığınıza bağlıdır. Birincil düğümde çalıştırılırsa, **dspmqr** tarafından döndürülen normal durum iletilerinden biri görüntülenir. Komutu ikincil bir düğümde çalıştırırsanız, Ended **immediately** durumu görüntülenir. Örneğin, **dspmqr** RDQM7 düğümünde çalıştırılırsa, aşağıdaki bilgiler döndürülebilir:

QMNAME (DRQM8)	STATUS (Ended immediately)
QMNAME (DRQM7)	STATUS (Running)

Bir RDQM ' in olağanüstü durumdan kurtarma için yapılandırılıp yapılandırılmadığını ve şu anda birincil mi, yoksa ikincil eşgörünüm mi olduğunu belirlemek için dspmqr ile bağımsız değişkenleri kullanabilirsiniz:

```
dspmqr -m QMname -o (dr | DR)
```

Aşağıdaki yanıtlardan biri görüntülenir:

#### **DRROLE ()**

Kuyruk yöneticisinin olağanüstü durumdan kurtarma için yapılandırılmadığını gösterir.

#### **DRROLE (Primary)**

Kuyruk yöneticisinin DR birincil olarak yapılandırıldığını gösterir.

#### **DRROLE (Secondary)**

Kuyruk yöneticisinin DR ikincil olarak yapılandırıldığını gösterir.

### **İlgili başvurular**

[dismqr](#)

[sonmqm](#)

[strmqm](#)

### **Linux DR RDQM durumunu görüntüleme**

Bir düğümdeki tüm olağanüstü durumdan kurtarma eşlenmiş veri kuyruğu yöneticilerinin (DR RDM) durumunu ya da belirli bir DR RDQM ' ye ilişkin ayrıntılı bilgileri görüntüleyebilirsiniz.

### **Bu görev hakkında**

**rdqmstatus** komutu, tüm DR RDIMM ' lerinin ya da tek tek RDIMM ' lerin durumunu görüntülemek için kullanılır.

**V 9.3.0** Bir düğümün özet durumu, RDQM ' nin dayandığı DRBD çekirdek modülüyle ilgili bilgileri de görüntüler. RDQM ' yi yükselttiğinizde, sistemde çalışan RHEL çekirdeği sürümü için DRBD çekirdek modülünün doğru sürümünün kurulu olduğundan emin olmanız önemlidir. Durum, işletim sistemi çekirdeğinin sürümünü, DRBD modülünün oluşturulduğu çekirdek sürümünü, DRBD sürümünü ve DRBD çekirdek modülü yüklendi durumunu görüntüler.

**rdqmstatus** komutunu çalıştırmak için mqm grubunda bir kullanıcı olmanız gerekir. Komutu DR RDQM çiftinin herhangi bir düğümünde çalıştırabilirsiniz.

### **Yordam**

- Bir düğümdeki tüm DR RDIMM ' lerinin özet durumunu görüntülemek için o düğümde aşağıdaki komutu çalıştırın:

```
rdqmstatus
```

Düğümdeki DR RDIMM ' lerinin durumu görüntülenir; örneğin:



```

Node:                               mqhvm07.exampleco.com
OS kernel version:                  3.10.0-1160.15.2
DRBD OS kernel version:             3.10.0-1160
DRBD version:                       9.1.1
DRBD kernel module status:         Loaded

Queue manager name:                 DRQM8
Queue manager status:               Ended immediately
DR role:                            Secondary

Queue manager name:                 DRQM7
Queue manager status:               Running
DR role:                            Primary

```

**V9.3.0** DRBD çekirdek modülü durumu aşağıdaki değerlerden biridir:

### Yüklendi

DRBD modülünün yüklendiğini gösterir.

### Kısmen yüklendi

DRBD modülü yüklendiğinde ortaya çıkabilir, ancak bir uyumsuzluk nedeniyle düzgün çalışmayabilir.

### Yüklenmedi

DRBD modülü yüklenmedi. Bu, henüz bir RDQM kuyruk yöneticisi yaratılmadığında, yeni kurulan bir yapılandırma görüntülenebilir.

### Kurulu değil

DRBD modülünden birinin kurulu olmadığını gösterir. ya da IBM MQ , DRBD modülünün işletim sistemi çekirdeği sürümünü belirleyemedi.

### Önceden kurulan sürüm yüklenmeye devam ediyor

Bu durum, var olan DRBD modülü çalışırken (yani bir RDQM kuyruk yöneticisi çalışırken) yeni bir DRBD modülü kurulursa ortaya çıkabilir. Yeni kurulan modül, gerçekte çalışmakta olan modül değil, durumda bildirilir.

- Belirli bir RDQM ' nin durumunu görüntülemek için şu komutu girin:

```
rdqmstatus -m qmname
```

Aşağıdaki çizelge, döndürülen bilgileri özetler.

Çizelge 34. Durum öznitelikleri		
Durum Özniteliği	Olası Değerler	Görüntülendiğinde
Kuyruk yöneticisi durumu	state ( dspmqtarafından görüntülediği şekilde)	Her zaman görüntülenir
CPU	<i>n.nn%</i>	Yalnızca geçerli düğümdeki RDQM birincil role sahip olduğunda gösterilir
Bellek	<i>nnnMB</i>	Yalnızca geçerli düğümdeki RDQM birincil role sahip olduğunda gösterilir
Kuyruk yöneticisi dosya sistemi	<i>nnnMB</i> kullanıldı, <i>n.nGB</i> ayrıldı [ <i>n%</i> ]	Yalnızca geçerli düğümdeki RDQM birincil role sahip olduğunda gösterilir
DR rolü	Birincil İkincil Bilinmeyen	Her zaman görüntülenir
DR Durumu	Olağan	Olağan çalışma
	Eşitleme devam ediyor	Eşitleme devam ediyor

Çizelge 34. Durum öznitelikleri (devamı var)		
Durum Özniteliği	Olası Değerler	Görüntülediğinde
	Bölümlenmiş	DR eşleme ağı kullanılmıyorsa, kuyruk yöneticisi her iki düğümde de başlatıldı
	Uzak sistem kullanılmıyor	Diğer düğümle bağlantı kesildi
	Tutarsız	Bir eşitleme işlemi devam ediyordu, ancak kesintiye uğradı
	Anlık görüntüye geri dönüyor	Kullanıcı, kuyruk yöneticisi Tutarsız durumuna girdiğinde alınan anlık görünüme geri dönmeyi seçti.
	Uzak sistemin konfigürasyonu tanımlanmadı	RDQM 'nin birincil eşgörünümlü yapılandırıldı, ancak ikincil eşgörünüm yapılandırılmadı
	Anlaşma başarısız oldu	Düğümleden biri zamanuyumlu eşleme, diğeri zamanuyumsuz kopyalama olarak ayarlandı
DR tipi	Zamanuyumlu ya da zamanuyumsuz	Her zaman görüntülenir
DR kapısı	<i>kapı_no</i> (bu kuyruk yöneticisine ilişkin verileri eşlemek için kullanılan TCP/IP kapısı)	Her zaman görüntülenir
DR yerel IP adresi	Bu kuyruk yöneticisinin DR için eşlediği yerel IP adresi	Her zaman görüntülenir
DR uzak IP adresi	Bu kuyruk yöneticisinin DR için eşlediği uzak IP adresi	Her zaman görüntülenir
Olağanüstü durumdan kurtarma eşitlenmemiş verileri	<i>n</i> KB	Uzak düğüm kullanılmadığında ya da tutarsız olduğunda görüntülenir
DR eşitlemesi ilerleme durumu	<i>n</i> %	Eşitleme devam ederken görüntülenir
Olağanüstü durumdan kurtarma tahmini tamamlanma süresi	YYYY-MM-DD SS:MM:SS	Eşitleme devam ederken görüntülenir
Anlık görünüm yeniden sürümü ilerleme durumu	<i>n</i> %	DR durumu Reverting to snapshot olduğunda görüntülenir. Durum geri sayılıyor, bu nedenle %0 tamamlanmayı gösteriyor
► V 9.3.0 ► V 9.3.0 Son zamanuyumlu veri ayırma	YYYY-MM-DD SS:MM:SS	DR verileri eşitlenmemiş olduğunda (ilk eşitlemeden sonra) görüntülenir. Verilerin en son eşitlendiği tarihi ve saati verir.

## Örnek

Birincil düğümdeki normal durum örneği:

```
Queue manager status: Running
CPU: 0.00
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role: Primary
DR status: Normal
DR type: Synchronous
DR port: 3000
DR local IP address: 192.168.20.1
DR remote IP address: 192.168.20.2
```

İkincil bir düğümdeki normal durum örneği:

```
Queue manager status: Ended immediately
DR role: Secondary
DR status: Normal
DR port: 3000
DR local IP address: 192.168.20.2
DR remote IP address: 192.168.20.1
```

Eşitleme devam ederken birincil düğümde durum örneği:

```
Queue manager status: Running
CPU: 0.53
Memory: 124MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role: Primary
DR status: Synchronization in progress
DR type: Synchronous
DR port: 3000
DR local IP address: 192.168.20.1
DR remote IP address: 192.168.20.2
DR synchronization progress: 11.0%
DR estimated time to completion: 2017-09-06 14:55:05
```

Bölümlenmiş olduğunu gösteren birincil düğüm örneği:

```
Queue manager status: Running
CPU: 0.02
Memory: 124MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role: Primary
DR status: Partitioned
DR type: Synchronous
DR port: 3000
DR local IP address: 192.168.20.1
DR remote IP address: 192.168.20.2
```

### V 9.3.0

İkincil düğümle uyumsuz olduğunu gösteren birincil düğüm örneği:

```
Queue manager status: Running
CPU: 0.00
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role: Primary
DR status: Remote unavailable
DR type: Asynchronous
DR port: 3000
DR local IP address: 192.168.20.1
DR remote IP address: 192.168.20.2
DR out of sync data: 15932KB
DR last in sync: 2020-07-27 16:01:47
```

### V 9.3.0

İşletim sistemi çekirdek sürümü (RHEL 7.9) ile DRBD çekirdek modülü (RHEL 7.8 için hedeflenmiş) arasındaki uyumsuzluğu gösteren bir özet durumu örneği. Durum, DRBD çekirdek modülünün yüklendiğini ve beklenen kuyruk yöneticisinin çalıştığını bildirirse de, DRBD çekirdek modülünü bu durumda çalışan işletim sistemi çekirdeği için hedeflenen sürümle güncellemenizdir.

```
Node: mqhvm07.exampleco.com
OS kernel version: 3.10.0-1160.15.2
DRBD OS kernel version: 3.10.0-1127
DRBD version: 9.1.1
DRBD kernel module status: Loaded

Queue manager name: DRQM8
Queue manager status: Ended immediately
DR role: Secondary

Queue manager name: DRQM7
Queue manager status: Running
DR role: Primary
```

**V9.3.0** İşletim sistemi çekirdek sürümü (RHEL 7.9) ile DRBD çekirdek modülü (RHEL 7.6 için hedeflenen) arasındaki uyumsuzluğu gösteren bir özet durumu örneği. Bu örnekte sürüm uyumsuzluğu daha ciddi ve DRBD çekirdek modülü başarıyla yüklenemedi. QM3 bir DR kuyruk yöneticisidir ve Birincil eşgörünüm olması amaçlıdır, ancak DRBD çekirdek modülü tam olarak yüklenmediği için UnknownDR durumuyla İkincil olarak rapor eder. Bu hatayı çözmek için DRBD çekirdek modülü, çalışan işletim sistemi çekirdeğine ilişkin sürüm hedefiyle güncellenmelidir.

```
Node: mqhvm57.exampleco.com
OS kernel version: 3.10.0-1160.15.2
DRBD OS kernel version: 3.10.0-957
DRBD version: 9.1.2+ptf.3
DRBD kernel module status: Partially loaded

Queue manager name: QM3
Queue manager status: Status not available
DR role: Secondary
DR status: Unknown
```

## İlgili başvurular

**Linux** [rdqmstatus](#)

## **Linux** *Olağanüstü durumdan kurtarma ortamında çalışma*

Olağanüstü durumdan kurtarma yapılandırmasında ikincil kuyruk yöneticisine geçmek isteyebileceğiniz bazı durumlar vardır.

### Olağanüstü durum kurtarma

Ana sitede birincil kuyruk yöneticisinin tam olarak kaybolmasına rağmen, ikincil kuyruk yöneticisini kurtarma yerinde başlatabilirsiniz. Uygulamalar kurtarma yerindeki kuyruk yöneticisine yeniden bağlanır ve ikincil kuyruk yöneticisi uygulama iletilerini işler. Önceki yapılandırmaya geri dönmek için atılan adımlar, hatanın nedenine bağlıdır. Örneğin, ana düğümün geçici kaybına karşı tam kaybı.

Ana siteyi geçici olarak kaybetmenin ardından gerçekleştirilecek adımlar için bkz. [“Kurtarma düğümüne geçiş” sayfa 601](#). Kalıcı hatanın ardından gerçekleştirilecek adımlar için bkz. [“Olağanüstü durumdan kurtarma yapılandırmasında arızalı bir düğümün değiştirilmesi” sayfa 602](#).

### Olağanüstü durumdan kurtarma testi desteği

İkincil eşgörünüme geçici olarak geçiş yaparak ve uygulamaların başarıyla bağlanabildiğini denetleyerek olağanüstü durumdan kurtarma yapılandırmasını sınavabilirsiniz. Birincil düğümün geçici bir hatasının ardından geçiş yaparken de aynı yordamı izlemeniz gerekir, bkz. [“Kurtarma düğümüne geçiş” sayfa 601](#).

### Anlık görüntüye geri dönülüyor

Eşitleme devam ederken birincil düğümde bir hata yaşarsanız, eşitleme başlamadan hemen önce ikincil kuyruk yöneticisi verilerinin alınan anlık görüntüsüne geri dönebilirsiniz. Daha sonra ikincil, tutarlı bir duruma geri yüklenir ve birincil olarak çalıştırılabilir. Anlık görüntüye geri dönmek için, ikincil görüntüyü [“Kurtarma düğümüne geçiş” sayfa 601](#) içinde açıklandığı gibi birincil görünüme çevirin. Kuyruk yöneticisini başlatmadan önce, anlık görünüme geri döndürme işleminin tamamlandığını (**rdqmstatus** komutunu kullanarak) denetlemeniz gerekir.

Ana yerinizde bir olağanüstü durum oluşursa, kurtarma yerinize geçmek için adımlar atmanız gerekir.

## Bu görev hakkında

Ana yerdeki birincil kuyruk yöneticisinin kaybolmasını takiben, kurtarma yerindeki ikincil kuyruk yöneticisini birincil kuyruk yöneticisine çevirir ve yeniden başlatırsınız. Uygulamalar, kurtarma yerindeki kuyruk yöneticisine yeniden bağlanır ve kuyruk yöneticisi uygulama iletilerini işler. Bu yordamı, kurtarma düğümünüzü sınamak için de kullanabilirsiniz.

**Önemli:** Özgün ikincil yönetim ortamını yükseltmeden önce, bir kuyruk yöneticisinin birincil yönetim ortamının çalışmadığından ya da durdurulmadığından ve ikincil yönetim ortamı haline getirildiğinden emin olmanız gerekir. Tersi durumda, bölümlenmiş veriler birikebilir.

Kök olarak oturum açmanız ya da mqm grubuna ait bir kullanıcı olarak oturum açmanız ve gerekli sudo yapılandırmasına sahip olmanız gerekir.

## Yordam

1. İkincil kuyruk yöneticinizi sınamak için bu yordamı kullanıyorsanız (birincil yönetim ortamı çalışmaya devam ediyorsa), birincil yönetim ortamını durdurmalı ve ikincil yönetim ortamı olarak yeniden atamalısınız:

```
endmqm qmname  
rdqmdr -m qmname -s
```

2. Kurtarma düğümünde aşağıdaki komutu girerek ikincil kuyruk yöneticisini birincil sıraya alın:

```
rdqmdr -m qmname -p
```

3. Aşağıdaki komutu girerek kuyruk yöneticisini başlatın:

```
strmqm qmname
```

4. Uygulamalarınızın kurtarma kuyruğu yöneticisinde kuyruk yöneticisine yeniden bağlandığından emin olun. Kanallarınızı, birincil ve ikincil kuyruk yöneticilerinizi belirterek alternatif bağlantı adları listesiyle tanımlamanız koşuluyla, uygulamalarınız otomatik olarak yeni birincil kuyruk yöneticisine bağlanır.

## Sonraki adım

Başarısız olan düğüm geri yüklendiğinde, iki düğüm arasındaki bağlantının çalışması koşuluyla, kuyruk yöneticisi bu düğümde başlatılamıyor; ikincil kuyruk yöneticisi eşgörünümünü yükseldiğiniz kurtarma düğümünde çalışıyor. Olağan işleme dönmek için, kurtarma düğümünde kuyruk yöneticisini durdurmalı ve özgün düğümdeki kuyruk yöneticisini birincil role yükseltmelisiniz.

### İlgili başvurular

[strmqm](#)

[rdqmdr](#)

*Kurtarma RDQM kuyruk yöneticisi test ediliyor*

Bir RDQM olağanüstü durumdan kurtarma yapılandırmasındaki kuyruk yöneticisinin kurtarma eşgörünümünün, ana siteyi kesintiye uğratmadan doğru şekilde çalıştığını test edebilirsiniz.

## Bu görev hakkında

Kurtarma kuyruğu yöneticisini sınamak için, ana düğümler ile kurtarma düğümleri arasındaki arabirimi geçersiz kılın. İkincil kuyruk yöneticisini birincil sıraya yerleştirir ve bağımsız kuyruk yöneticisini sınavabilirsiniz. Sınama tamamlandıktan sonra arabirimi geri yükler ve sınama kuyruğu yöneticisini silersiniz. Daha sonra, kuyruk yöneticisini olağanüstü durumdan kurtarma yapılandırmasında ikincil kuyruk yöneticisi olarak yeniden yaratırsınız.

## Yordam

1. Ana düğüm ile kurtarma düğümü arasındaki ağ bağlantısını devre dışı bırakın.
2. Kurtarma düğümünde kuyruk yöneticisini birincil olarak belirleyin:

```
rdqmdr -m QMname -p
```

Burada *QMname* , kuyruk yöneticisinin adıdır.

3. Kuyruk yöneticisini başlat:

```
strmqm QMname
```

4. Uygulamaları kuyruk yöneticisine bağlayın ve beklendiği gibi çalıştıklarını test edin.
5. Kuyruk yöneticisini sona erdir:

```
endmqm QMname
```

6. Kuyruk yöneticisini sil:

```
dltmqm QMname
```

7. Ana ve kurtarma araçları arasındaki ağ bağlantısını geri yükleyin.
8. Ana düğümde, olağanüstü durumdan kurtarma işlemini ilk kez yapılandırırdığınızda kullandığınız **crtmqm** komutunu almak için aşağıdaki komutu çalıştırın.

```
rdqmdr -d -m QMname
```

9. İkincil kuyruk yöneticisini yeniden yaratmak için kurtarma düğümünde sonuçtaki **crtmqm** komutunu çalıştırın. Ana düğümdeki birincil kuyruk yöneticisi, verileri ikincil kuyruk yöneticisiyle uyumlulaştırarak güncelliğini sağlar.

### Linux

*Olağanüstü durumdan kurtarma yapılandırmasında arızalı bir düğümün değiştirilmesi*  
Olağanüstü durumdan kurtarma yapılandırmasındaki düğümlerden birini kaybederseniz, bu yordamı izleyerek düğümü değiştirebilir ve olağanüstü durumdan kurtarma yapılandırmasını geri yükleyebilirsiniz.

## Bu görev hakkında

Ana yerdeki düğüm onarılamayacak şekilde bir olağanüstü durum oluşursa, kuyruk yöneticisi kurtarma düğümünde çalışırken hatalı düğümü değiştirebilir ve ardından özgün olağanüstü durumdan kurtarma yapılandırmasını geri yükleyebilirsiniz. Yeni düğüm, hatalı düğümün kimliğini üstlenmelidir: Ad ve IP adresi aynı olmalıdır.

Kök olarak oturum açmanız ya da mqm grubuna ait bir kullanıcı olarak oturum açmanız ve gerekli sudo yapılandırmasına sahip olmanız gerekir.

## Yordam

Ana sitede kuyruk yöneticisinin kaybını takiben aşağıdaki adımları izleyin:

1. Kurtarma düğümünde, ikincil kuyruk yöneticisinin birincil rolü üstlenmesini sağlamak için aşağıdaki komutları çalıştırın:

```
rdqmdr -m QMname -p
```

Burada *QMname* , kuyruk yöneticisinin adıdır.

2. Olağanüstü durumdan kurtarma işlemini yeniden yapılandırmak için yedek birincil düğümde çalıştırmanız gereken komutu alın:

```
rdqmdr -m QMname -d
```

Bu komutun çıkışını kopyalayın.

3. Kuyruk yöneticisini başlatmak için aşağıdaki komutu çalıştırın:

```
strmqm QMname
```

4. Uygulamalarınızın kurtarma düğümündeki kuyruk yöneticisine yeniden bağlandığından emin olun. Kanallarınızı, birincil ve ikincil kuyruk yöneticilerinizi belirterek alternatif bağlantı adları listesiyle tanımlamanız koşuluyla, uygulamalarınız otomatik olarak yeni birincil kuyruk yöneticisine bağlanır.
5. Ana sitenizdeki arızalı düğümü değiştirin ve özgün düğümde olağanüstü durumdan kurtarma için kullandığınız adla ve IP adresiyle aynı olacak şekilde yapılandırın. Daha sonra adım 2 'de kopyaladığınız **strmqm** komutunu çalıştırarak olağanüstü durumdan kurtarma özelliğini yapılandırın. Artık kuyruk yöneticisinin ikincil bir yönetim ortamınız var ve birincil yönetim ortamı, verilerini ikincil yönetim ortamıyla uyumlulaştırıyor.
6. Yürürlükteki birincil eşgörünümü sona erdirin.
7. Uyumlulaştırma tamamlandıktan sonra, kurtarma düğümünde çalışan birincil yönetim ortamını ikincil yönetim ortamına bir kez daha geçirin:

```
rdqmdr -m QMname -s
```

8. Yeni birincil düğümde, kuyruk yöneticisinin ikincil eşgörünümünü birincil yönetim ortamına ayarlayın:

```
rdqmdr -m QMname -p
```

9. Yeni birincil düğümde kuyruk yöneticisini başlatın:

```
strmqm QMname
```

Yapılandırmayı, ana sitenizdeki hatadan önce olduğu gibi geri yüklediniz.

## İlgili başvurular

[strmqm](#)

[rdqmdr](#)

[sonmqm](#)

*DR RDQM ' de tutarsız bir sorunun çözülmesi*

Bir kuyruk yöneticisinin birincil ve ikincil eşgörünümleri arasında eşitleme başarısız olursa, inconsistent DR durumu bildirilir.

## Bu görev hakkında

Bir uyumlulaştırma işlemi sırasında birincil yönetim ortamıyla eşleme bağlantısı kesildiği için, kuyruk yöneticisinin ikincil yönetim ortamında tutarsız bir durum bildirildi. Bu durumu çözmek için harekete geçmeniz gerekebilir. Aşağıdaki olay dizisini göz önünde bulundurun:

1. DR ikincil kuyruk yöneticisiyle zamanuyumlu DR birincil kuyruk yöneticisi
2. Birincil ve ikincil arasında kaybolan eşleme bağlantısı
3. Birincil ve ikincil arasında geri yüklenen eşleme bağlantısı
4. DR ikincil kuyruk yöneticisinin DR birincil kuyruk yöneticisini yakaladığı bir yeniden eşzamanlama gerçekleşir. Bu süre içinde, her iki kuyruk yöneticisi için de *synchronization in progress* DR durumu bildirilir.
5. Eşleme yeniden eşzamanlama sırasında yeniden kaybolursa, DR ikincil durumundaki durum *Inconsistent* olarak bildirilir.

Birincil kuyruk yöneticisini barındıran düğüm çalışmaya devam ediyorsa ve eşleme bağlantısı geri yüklenebiliyorsa, yeniden eşzamanlama otomatik olarak gerçekleşir. Tutarsız durum, siz herhangi bir işlem yapmadan çözülür.

Birincil kuyruk yöneticisini barındıran düğüm artık çalışmıyorsa, ikincil kuyruk yöneticisinde anlık görünüme geri döndürme uygulayarak tutarsız durumu çözebilirsiniz. Bu işlem, verileri bilinen son iyi duruma geri çevirir.

## Yordam

Tutarsız bir durumu çözmek için:

1. Kurtarma düğümünde, ikincil yönetim ortamını birincil yönetim ortamında yapın:

```
rdqmdr -m qmname -p
```

Anlık görüntüye geri döndürme işlemi başlar.

2. Kurtarma düğümünde, anlık görüntüye geri döndürme işleminin ne zaman tamamlandığını görmek için kuyruk yöneticisinin durumunu denetleyin:

```
rdqmstatus -m qmname
```

3. Kuyruk yöneticisi durumu Normal olduğunda, kuyruk yöneticisini başlatın:

```
strmqm qmname
```

### DR RDQM ' de bölümlenmiş (bölünmüş beyin) sorununu çözme

Bir olağanüstü durumdan kurtarma çiftindeki her iki kuyruk yöneticisi de aynı anda birincil rolde çalışıyorsa, bölümlenmiş bir sorun oluşabilir.

## Bu görev hakkında

Özgün birincil yönetim ortamı ana düğümde çalışmaya devam ederken, kurtarma düğümünde bir kuyruk yöneticisinin ikincil yönetim ortamını yükseltirseniz, her biri kuyruk yöneticisi verilerine ilişkin kendi görünümü olan aynı kuyruk yöneticisinin iki sürümü çalışır durumda olur. Her düğümdeki kuyruk yöneticisine ilişkin DR durumu Partitioned olarak bildirilir.

İki kuyruk yöneticisinden hangisinin verilerin en doğru görünümüne sahip olduğuna karar vermeli ve diğerini atarken bu kümeyi korumalısınız. Bu işlemi tamamlamak için **rdqmdr** komutunu kullanın.

İki prosedür var. İlki, verilerin ana düğümde alınmasını, ikincisi ise verileri kurtarma düğümünden alıkoymayı açıklar.

## Yordam

- Ana düğümdeki kuyruk yöneticisinden verileri alıkoymak için:
  - a) Her iki kuyruk yöneticisi eşgörünümünün de durdurulduğunu doğrulayın.
  - b) Kurtarma düğümündeki kuyruk yöneticisinin ikincil olduğunu belirtin:

```
rdqmdr -m qmname -s
```

- c) Ana düğümdeki kuyruk yöneticisinin birincil olduğunu belirtin:

```
rdqmdr -m qmname -p
```

Eşitleme, kurtarma düğümüne kopyalanmakta olan ana düğümdeki kuyruk yöneticisinden alınan verilerle başlar.

- d) Eşitlemenin durumunu denetleyin:

```
rdqmstatus -m qmname
```



e) Eşitleme tamamlandığında, ana düğümde kuyruk yöneticisini başlatın:

```
stmqm qmname
```

• Verileri kurtarma düğümünde kuyruk yöneticisinden alıkoymak için:

a) Her iki kuyruk yöneticisi eşgörünümünün de durdurulduğunu doğrulayın.

b) Ana düğümdeki kuyruk yöneticisinin ikincil olduğunu belirtin:

```
rdqmdr -m qmname -s
```

c) Kurtarma düğümündeki kuyruk yöneticisinin birincil olduğunu belirtin:

```
rdqmdr -m qmname -p
```

Eşitleme, kurtarma düğümündeki kuyruk yöneticisinden alınan verilerin ana düğüme kopyalanmasıyla başlar.

d) Eşitlemenin durumunu denetleyin:

```
rdqmstatus -m qmname
```

e) Eşitleme tamamlandığında, kurtarma düğümünde kuyruk yöneticisini indirgeme:

```
rdqmdr -m qmname -s
```

f) Kuyruk yöneticisini ana düğümde yükselt ve başlat:

```
rdqmdr -m qmname -p  
stmqm qmname
```

### *Olağanüstü durumdan kurtarma yapılandırmalarında IP adreslerinin değiştirilmesi*

Olağanüstü durumdan kurtarma yapılandırmasındaki arabirimlerden birinin IP adreslerini değiştirirseniz, iki düğüm arasında eşleme artık mümkün değildir.

DR düğümlerinden herhangi biri için eşleme arabirimine ilişkin IP adreslerini değiştirmeniz gerekirse, aşağıdaki yordamı kullanmanız gerekir:

1. Birincil düğümde DR kuyruk yöneticilerini yedekleyip silin. Kurtarma düğümünde kuyruk yöneticilerini silin. Bkz. [“IBM MQ kuyruk yöneticisi verilerinin yedeklenmesi ve geri yüklenmesi” sayfa 655](#) ve [“DR RDQM ' in silinmesi” sayfa 588](#).
2. Yeni IP adreslerini belirterek DR kuyruk yöneticilerini yeniden yaratın ve yedekleri geri yükleyin, bkz. [“Olağanüstü durumdan kurtarma RDQM ' in yaratılması” sayfa 586](#) ve [“IBM MQ kuyruk yöneticisi verilerinin yedeklenmesi ve geri yüklenmesi” sayfa 655](#).

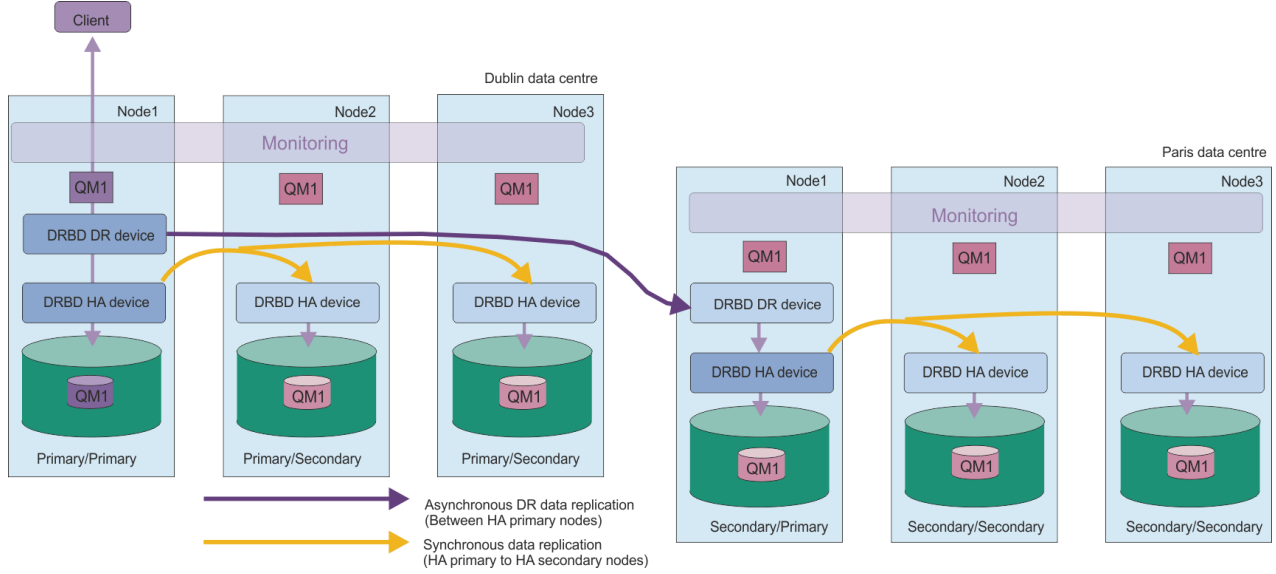
Linux

## **RDQM olağanüstü durumdan kurtarma ve yüksek düzeyde kullanılabilirlik**

Bir sitede yüksek kullanılabilirlik grubunda çalışan, ancak ilk grubu kullanılamaz duruma getiren bir olağanüstü durum oluşursa başka bir yerdeki başka bir yüksek kullanılabilirlik grubuna geçilebilen bir eşlenmiş veri kuyruğu yöneticisi (RDQM) yapılandırabilirsiniz. Bu, DR/HA RDQM olarak bilinir.

DR/HA RDQM, yüksek kullanılabilirlikli RDQM (bkz. [“RDQM yüksek kullanılabilirlik düzeyi” sayfa 552](#)) ve olağanüstü durum kurtarma RDQM ' nin özelliklerini birleştirir (bkz. [“RDQM olağanüstü durumdan kurtarma” sayfa 581](#)).

Aşağıdaki çizgede örnek bir DR/HA RDQM gösterilmektedir.



Ana yerdeki DR/HA RDIMM 'leri ile olağanüstü durumdan kurtarma yeri arasındaki eşleme her zaman zamanuyumsuzdur. Zamanuyumsuz kopyalamada, IBM MQ PUT ya da GET gibi işlemler tamamlanır ve olay ikincil kuyruk yöneticisine eşlenmeden önce uygulamaya geri döner.

Gerekirse, 'main' ve 'recovery' yerine iki etkin siteniz olabilir; bu nedenle, DR/HA RDIMM 'lerinizin bazıları bir yerde, bazıları normal çalışma sırasında çalışır. Bir olağanüstü durum ortaya çıkarsa ve bir yer kullanılamaz duruma gelirse, tüm DR/HA RDIMM 'leri aynı yerde aynı HA grubunda çalışır.

Her HA grubu, sıradan bir HA grubuyla aynı şekilde yapılandırılır. Her bir HA grubunda bir DR/HA RDQM için kayan IP adresleri tanımlayabilirsiniz. Kayan IP adresi, her HA grubu için aynı ya da farklı olabilir.

Var olan bir RDQM 'i DR/HA RDQM olacak şekilde yükseltemezsiniz, DR/HA RDQM yaratmanız gerekir. (Gerekliyse, var olan bir RDQM 'nin verilerini yedekleyebilir, silebilir, DR/HA RDQM olarak yeniden yaratabilir ve sonra verileri geri yükleyebilirsiniz, bkz. ["IBM MQ kuyruk yöneticisi verilerinin yedeklenmesi ve geri yüklenmesi"](#) sayfa 655.)

DR/HA RDIMM 'leri yapılandırmak için aşağıdaki ana adımları tamamlamanız gerekir:

1. 'main' sitesinde bir HA grubu yapılandırın.
2. 'Recovery' sitesinde bir HA grubu yapılandırın.
3. 'main' sitesindeki HA grubunun bir düğümünde birincil/birincil DR/HA RDQM yaratın.
4. 'Ana' sitedeki diğer iki düğümde birincil/ikincil DR/HA RDIMM 'leri oluşturun.
5. Bir uygulamanın 'ana' sitedeki HA grubunun herhangi bir düğümünde çalışırken DR/HA RDQM 'ye erişmesi için bir kayan IP adresi tanımlayın.
6. 'Recovery' yerindeki HA grubunun bir düğümünde ikincil/birincil DR/HA RDQM yaratın.
7. 'Recovery' yerindeki diğer iki düğümde ikincil/ikincil DR/HA RDIMM 'leri oluşturun.
8. Bir uygulamanın 'kurtarma' yerindeki HA grubunun herhangi bir düğümünde çalışırken DR/HA RDQM 'ye erişmesi için bir kayan IP adresi tanımlayın.

Bu adımların her birine ilişkin ayrıntılar aşağıdaki konularda verilmiştir.

### **Linux DR/HA RDQM çözümüne ilişkin gereksinimler**

DR/HA RDQM çözümüne ilişkin gereksinimler, HA RDQM çözümü ve DR RDQM çözümüne ilişkin gereksinimlerle aynıdır.

Yapılandırmanın HA bölümlerine ilişkin gereksinimlerin ayrıntıları için bkz. ["RDQM HA çözümü için gereksinimler"](#) sayfa 554.

Yapılandırmanın DR bölümüne ilişkin ayrıntılar için bkz. ["RDQM DR çözümü için gereksinimler"](#) sayfa 584.

Hem ana hem de kurtarma yerlerinde bir HA grubu oluşturmanız gerekir. Herhangi bir yerde var olan bir HA grubunuz varsa, bu HA grubunda DR/HA RDIMM 'leri oluşturabilirsiniz. (Var olan RDIMM 'ler önceki gibi çalışmaya devam eder.)

Yordam, RDQM yüksek kullanılabilirliği için anlatılan yordamla aynıdır, bkz. “Pacemaker kümesinin tanımlanması (HA grubu)” sayfa 557.

Yüksek kullanılabilirlikli bir grup tanımlarken, `rdqm.ini` dosyasındaki her bir düğüm tarafından izleme ve eşleme için kullanılan IP adreslerini belirtirsiniz. DR/HA RDIMM 'lerini desteklemek için bir HA grubu yaratırken, tanımladığınız HA grubu tarafından DR eşlemesi için kullanılan IP adreslerini ve DR çiftinin diğer HA grubundaki düğümler tarafından DR eşlemesi için kullanılan IP adreslerini de belirtebilirsiniz. (`rdqm.ini` dosyasında DR eşleme IP adreslerini belirtmezseniz, bunları bir DR/HA RDQM yaratırken komut satırında belirtebilirsiniz.)

Var olan bir HA grubunu yapılandırıyorsanız, var olan `rdqm.ini` dosyanıza DR eşleme IP adresleri ekleyebilirsiniz. `rdqm.ini` güncellendikten sonra `rdqmadm` komutunu yeniden çalıştırmanız gerekmez, ancak herhangi bir DR/HA RDIMM yaratmadan önce `rdqm.ini` 'yi güncellemeniz gerekir.

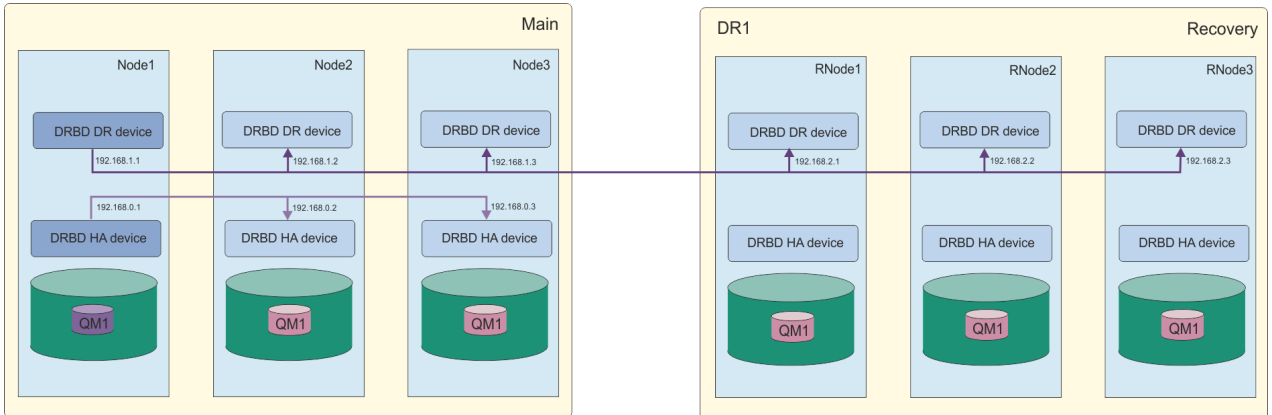
Tanımlamakta olduğunuz HA grubunda DR eşleme arabirimlerini belirtmek için Node stanzas içindeki `DR_Replication` özniteliğini kullanın; örneğin:

```
Node:
  Name=Node1
  HA_Replication=192.168.0.1
  DR_Replication=192.168.1.1
Node:
  Name=Node2
  HA_Replication=192.168.0.2
  DR_Replication=192.168.1.2
Node:
  Name=Node3
  HA_Replication=192.168.0.3
  DR_Replication=192.168.1.3
```

Uzak HA grubunun DR eşleme adreslerini belirtmek için `DRGroup` kısmı kullanın; örneğin:

```
DRGroup:
  Name=DR1
  DR_Replication=192.168.2.1
  DR_Replication=192.168.2.2
  DR_Replication=192.168.2.3
```

Aşağıdaki çizge bu yapılandırmayı göstermektedir:



Bir DR/HA RDQM yarattığınızda `rdqm.ini` dosyasındaki ya da komut satırındaki yerel HA grubundaki düğümler için DR eşleme IP adreslerini belirtmezseniz, her düğüm için tanımlanan `HA_Replication` arabirimleri DR eşlemesi için kullanılır. Uzak HA grubu DR eşleme adreslerini `rdqm.ini` dosyasında ya da `crtmqm` komut satırında belirtmeniz gerekir.

## Linux DR/HA RDIMM 'leri oluşturma

Bir DR/HA yapılandırmasında eşlenmiş bir veri kuyruğu yöneticisi (RDQM) oluşturmak için `crtmqm` komutunu kullanırsınız.

### Bu görev hakkında

Kullanıcı sudo kullanabiliyorsa, `mqm` grubunda bir kullanıcı olarak DR/HA RDQM oluşturabilirsiniz. Aksi takdirde RDQM 'yi kök olarak yaratmanız gerekir.

Bir dizi DR/HA RDM oluşturmanız gerekir:

- 'main' sitesindeki HA grubunda:
  - Kuyruk yöneticisinin normal koşullar altında çalışmasını istediğiniz düğümde birincil/birincil DR/HA RDQM 'yi yaratın.
  - HA grubundaki diğer iki düğümün her birinde bir birincil/ikincil DR/HA RDQM yaratın.
- 'Recovery' sitesindeki HA grubunda:
  - Kurtarma yerinde başarısız olursa kuyruk yöneticisinin çalışacağı düğümde ikincil/birincil DR/HA RDQM 'yi yaratın. 'Ana' sitede birincil/birincil kuyruk yöneticisini yaratırken komut çıkışı kullanabilirsiniz.
  - HA grubundaki diğer iki düğümün her birinde, ikincil/ikincil bir DR/HA RDQM yaratın.

Kuyruk yöneticisi yönetim ortamlarının tümü aynı ada sahip olmalı ve aynı miktarda saklama alanı ayrılmalıdır.

Aşağıdaki noktalar, kuyruk yöneticisi dosya sisteminin boyutlandırılmasına ilişkin bazı kılavuzlar sağlar:

1. Bir RDQM kuyruk yöneticisi yaratılırken, kuyruk yöneticisi verilerini ve günlüklerini saklamak için bir dosya sistemi ayrılır. Kuyruk yöneticisinin devam eden etkinliği günlüklerine kaydedebilmesi ve uygulama iletilerini kuyruklara saklayabilmesi için bu dosya sisteminin uygun şekilde boyutlandırılması önemlidir. Dosya sisteminin boyutlandırırken en yoğun ileti sistemi gereksinimlerini, gelecekteki iş yükü artışını ve iletilerin kuyruklarda birikmesine neden olabilecek uygulama kesintilerini göz önünde bulundurun. Kuyruk yöneticisi kurtarma günlüğünün boyutunun hesaplanmasına ilişkin yönergeler için bkz. [“Günlük dosya sistemimi ne kadar büyük yapmalıyım?” sayfa 638](#). Uygulama iletileri için depolama gereksinimleri hesaplanırken, iletilerin boyutu ve sayısı ve bunların MQMD üstbilgisi ve sahip oldukları ileti özellikleri dikkate alınmalı.
2. RDQM kuyruk yöneticisi dosya sistemleri dinamik olarak yeniden boyutlandırılmıyor. Bu gerekiyorsa, RDQM kuyruk yöneticisini daha büyük bir dosya sistemiyle yedeklemeniz ve geri yüklemeniz gerekir, bkz. [“HA RDQM kuyruk yöneticisi için dosya sisteminin yeniden boyutlandırma” sayfa 566](#).
3. MAXDEPTH ve MAXFSIZE gibi yerel kuyruk özniteliklerini kullanarak diskteki tek tek kuyrukların boyutunu sınırlayabilirsiniz. Bkz. [IBM MQ kuyruk dosyalarının değiştirilmesi](#).
4. Devam eden disk kullanımınızı izlemeniz ve dosya sistemi kullanımı kritik duruma gelmeden önce disk kullanımı artarsa uygun şekilde yanıt vermeniz gerekir. Dosya sistemi kullanımı, platform/işletim sistemi yetenekleri kullanılarak ya da [Sistem konularında yayınlanan ölçümlerbaşıllıklı](#) konuda açıklanan IBM MQ sistem konularına abone olunarak izlenebilir.

### Yordam

- Birincil/birincil DR/HA RDQM 'yi yaratmak için:
  - a) Aşağıdaki komutu girin:

```
crtmqm -sx -rr p
        [-r1 DRLocalIP1,DRLocalIP2,DRLocalIP3]
        (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -in GroupName)
        -rp DRPort
        [-z] [-q] [-c Text] [-d DefXmitQ] [-h MaxHandles]
        [-g ApplicationGroup] [-oa user|group]
        [-t TrigInt] [-u DeadQ] [-x MaxUMsgs]
        [-lp LogPri] [-ls LogSec]
```

```
[-lc | -ll | -lla | -lln] [-lf LogFileSize]
[-p Port] [-fs FilesystemSize] QMgrName
```

Burada:

**-sx**

İlk HA rolünün birincil olduğunu gösterir.

**-rr p**

İlk DR rolünün birincil olduğunu gösterir.

**-rl DRLocalIP1, DRLocalIP2, DRLocalIP3**

İsteğe bağlı olarak, yerel sitedeki üç düğümde (yani 'ana' sitede) DR arabirimlerinin IP adreslerini belirtin. Belirtilmezse, `rdqm.ini` dosyasında belirtilen IP adresleri kullanılır.

**-ri DRRemoteIP1, DRRemoteIP2, DRRemoteIP3**

Uzak sitedeki üç düğümdeki ('kurtarma' sitesi) DR arabirimlerinin IP adreslerini belirleyin. Bunu ya da `-rn` parametresini belirtmelisiniz.

**-rn GroupName**

Uzak HA grubu adını `rdqm.ini` kütüğünde belirtildiği şekilde belirtin. `-ri` ya da `-rn` belirtmeniz gerekir.

**-rp Kapı**

DR eşlemesi için kullanılacak kapıyı belirtir.

**other\_crtmqm\_options**

İsteğe bağlı olarak aşağıdaki genel **crtmqm** seçeneklerinden birini ya da birkaçını belirtebilirsiniz:

- z
- q
- c *Metin*
- d *DefaultTransmissionKuyruk*
- h *MaxHandles*
- g *ApplicationGroup*
- oa kullanıcı | grup
- t *TrigInt*
- u *DeadQ*
- x *MaxUMsgs*
- lp *LogPri*
- ls *LogSec*
- lc | -l
- lla | -lln
- lf *LogFileSize*
- p *Kapı*

**-fs boyut**

İsteğe bağlı olarak, kuyruk yöneticisi için yaratılacak dosya sisteminin boyutunu, yani `drbdpool` birim grubunda yaratılan mantıksal birimin boyutunu belirler. Anlık görüntü işlemine geri dönmeyi desteklemek için bu boyuttaki başka bir mantıksal birim de oluşturulur, bu nedenle DR RDQM için toplam depolama alanı burada belirtilenden iki kat fazla olur.

*Boyut*, GB olarak belirtilen sayısal bir değerdir. MB cinsinden bir değeri, M karakterinin izlediği değeri girerek belirtebilirsiniz. Örneğin, 3 GB 'lik bir dosya sistemi boyutu belirtmek için 3 girin. 1024 MB dosya sistemi boyutunu belirtmek için 1024M girin. (Açıkça GB 'ye bir G soneki de ekleyebilirsiniz.)

**QMname**

Eşlenen veri kuyruğu yöneticisinin adını belirtir. Bu ad büyük ve küçük harfe duyarlıdır.

Komut tamamlandıktan sonra, kuyruk yöneticisinin ikincil/birincil eşgörünümünü yaratmak için kurtarma yerine girebileceğiniz komutu verir.

- HA grubundaki diğer iki düğümde birincil/ikincil DR/HA RDQM yaratmak için:
  - a) Her düğümde aşağıdaki komutu girin:

```
crtmqm -sxs -rr p
          [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
          (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
          -rp DRPort
          [-fs FilesystemSize] QMgrName
```

Burada:

**-sxs**

İlk HA rolünün ikincil olduğunu gösterir.

**-rr p**

İlk DR rolünün birincil olduğunu gösterir.

**-rl DRLocalIP1, DRLocalIP2, DRLocalIP3**

İsteğe bağlı olarak, yerel sitedeki üç düğümde (yani 'ana' sitede) DR arabirimlerinin IP adreslerini belirtin. Belirtilmezse, rdqm.ini dosyasında belirtilen IP adresleri kullanılır.

**-ri DRRemoteIP1, DRRemoteIP2, DRRemoteIP3**

Uzak sitedeki üç düğümdeki ('kurtarma' sitesi) DR arabirimlerinin IP adreslerini belirleyin. Bunu ya da -rn parametresini belirtmelisiniz.

**-rn GroupName**

Uzak HA grubu adını rdqm.ini kütüğünde belirtildiği şekilde belirtin. -ri ya da -rn belirtmeniz gerekir.

**-rp Kapı**

DR eşlemesi için kullanılacak kapıyı belirtir.

**-fs boyut**

Kuyruk yöneticisi için yaratılacak dosya sisteminin boyutunu, yani drbdpool birim grubunda yaratılan mantıksal birimin boyutunu belirtir. Birincil/birincil RDQM 'yi yaratırken varsayılan olmayan bir boyut belirttiyseniz, burada aynı değeri belirtmeniz gerekir.

*Boyut*, GB olarak belirtilen sayısal bir değerdir. MB cinsinden bir değeri, M karakterinin izlediği değeri girerek belirtebilirsiniz. Örneğin, 3 GB 'lik bir dosya sistemi boyutu belirtmek için 3 girin. 1024 MB dosya sistemi boyutunu belirtmek için 1024M girin. (Açıkça GB 'ye bir G soneki de ekleyebilirsiniz.)

**QMname**

Birincil/ikincil RDQM 'nin adını belirtir. Bu, RDQM 'nin birincil/birincil eşgörünümü için belirttiğiniz adla aynı olmalıdır. Adın büyük ve küçük harfe duyarlı olduğunu unutmayın.

- Kurtarma yerinde başarısız olursa kuyruk yöneticisinin çalışacağı düğümde ikincil/birincil DR/HA RDQM yaratmak için:

- a) Ana sitede birincil/birincil DR/HA 'yı yaratırken komut çıkışını kullanın ya da aşağıdaki komutu girin:

```
crtmqm -sx -rr s
          [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
          (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
          -rp DRPort
          [-fs FilesystemSize] QMgrName
```

**-sx**

İlk HA rolünün birincil olduğunu gösterir.

**-rr s**

İlk DR rolünün ikincil olduğunu gösterir.

**-rl DRLocalIP1, DRLocalIP2, DRLocalIP3**

İsteğe bağlı olarak, yerel yerdeki üç düğümde ('kurtarma' sitesi) DR arabirimlerinin IP adreslerini belirtin. Belirtilmezse, rdqm.ini dosyasında belirtilen IP adresleri kullanılır.

**-ri DRRemoteIP1, DRRemoteIP2, DRRemoteIP3**

Uzak sitedeki üç düğümdeki ('ana' yer) DR arabirimlerinin IP adreslerini belirtin. Bunu ya da -rn parametresini belirtmelisiniz.

**-rn GroupName**

Uzak HA grubu adını rdqm.ini kütüğünde belirtildiği şekilde belirtin. -ri ya da -rn belirtmeniz gerekir.

**-rp Kapı**

DR eşlemesi için kullanılacak kapıyı belirtir.

**-fs boyut**

İsteğe bağlı olarak, kuyruk yöneticisi için yaratılacak dosya sisteminin boyutunu, yani drbdpool birim grubunda yaratılan mantıksal birimin boyutunu belirler. Anlık görüntü işlemine geri dönmeyi desteklemek için bu boyuttaki başka bir mantıksal birim de oluşturulur, bu nedenle DR RDQM için toplam depolama alanı burada belirtilenden iki kat fazla olur.

**QMname**

Eşlenen veri kuyruğu yöneticisinin adını belirtir. Bu ad büyük ve küçük harfe duyarlıdır.

- Kurtarma yerindeki diğer iki düğümde ikincil/ikincil HA/DR RDQM yaratmak için:

a) Her düğümde aşağıdaki komutu girin:

```
crtmqm -sxs -rr s  
          [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]  
          (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)  
          -rp DRPort  
          [-fs FilesystemSize] QMgrName
```

**-sxs**

İlk HA rolünün birincil olduğunu gösterir.

**-rr s**

İlk DR rolünün ikincil olduğunu gösterir.

**-rl DRLocalIP1, DRLocalIP2, DRLocalIP3**

İsteğe bağlı olarak, yerel yerdeki üç düğümde DR arabirimlerinin IP adreslerini belirtin. Belirtilmezse, rdqm.ini dosyasında belirtilen IP adresleri kullanılır.

**-ri DRRemoteIP1, DRRemoteIP2, DRRemoteIP3**

Uzak yerdeki üç düğümdeki DR arabirimlerinin IP adreslerini belirleyin. Bunu ya da -rn parametresini belirtmelisiniz.

**-rn GroupName**

Uzak HA grubu adını rdqm.ini kütüğünde belirtildiği şekilde belirtin. -ri ya da -rn belirtmeniz gerekir.

**-rp Kapı**

DR eşlemesi için kullanılacak kapıyı belirtir.

**-fs boyut**

İsteğe bağlı olarak, kuyruk yöneticisi için yaratılacak dosya sisteminin boyutunu, yani drbdpool birim grubunda yaratılan mantıksal birimin boyutunu belirler. Anlık görüntü işlemine geri dönmeyi desteklemek için bu boyuttaki başka bir mantıksal birim de oluşturulur, bu nedenle DR RDQM için toplam depolama alanı burada belirtilenden iki kat fazla olur.

**QMname**

Eşlenen veri kuyruğu yöneticisinin adını belirtir. Bu ad büyük ve küçük harfe duyarlıdır.

**Not:** Bir RDQM oluşturduğunuzda, HA eşleme bağlantısı için 7000 'in üzerindeki bir sonraki boş kapı numarası ayrılır. Seçilen kapının başka bir uygulama tarafından kullanıldığı saptanırsa, **crtmqm** komutu AMQ6543 hatasıyla başarısız olur ve bu kapı bir dışlama listesine eklenir. Kuyruk yöneticisinin ikincil yönetim ortamlarını silmeli ve **crtmqm** komutunu yeniden çalıştırmalısınız.

**Sonraki adım**

Tüm DR/HA RDIMM ' lerini yarattıktan sonra, tümünün doğru olup olmadığını denetlemek için birincil/ birincil ve ikincil/birincil eşgürünümlerin durumunu denetlemeniz gerekir. Düğümlerde **rdqmstatus**

komutunu kullanın. Düğümlerin, “DR/HA RDQM ve HA grubu durumunun görüntülenmesi” sayfa 614’inde açıklandığı gibi normal durumu görüntülüyor olması gerekir. Bu durum görüntülenmezse, ikincil/birincil örneği silin ve doğru bağımsız değişkenleri kullanmaya özen göstererek yeniden yaratın.

### İlgili görevler

“DR/HA RDIMM ' leri oluşturma” sayfa 608

Bir DR/HA yapılandırmasında eşlenmiş bir veri kuyruğu yöneticisi (RDQM) oluşturmak için **crtmqm** komutunu kullanırsınız.

### İlgili başvurular

[crtmqm](#)

**Linux** *DR/HA RDQM ' in silinmesi*

Bir DR/HA eşlenmiş veri kuyruğu yöneticisini (RDQM) silmek için **dltmqm** komutunu kullanırsınız.

## Bu görev hakkında

Hem birincil/birincil düğümde hem de ikincil/birincil düğümde RDQM ' yi silmek için komutu çalıştırmanız gerekir. Önce RDQM sona erdirilmelidir. Kullanıcı gerekli sudo ayrıcalıklarına sahipse, komutu mqm kullanıcısı olarak çalıştırabilirsiniz. Tersi durumda, komutu kök olarak çalıştırmanız gerekir.

## Yordam

- Bir DR/HA RDQM ' yi silmek için şu komutu girin:

```
dltmqm RDQM_name
```

### İlgili başvurular

[dltmqm](#)

**Linux** *Kayan IP adresi oluşturma*

Bir DR/HA RDQM yapılandırmasında HA gruplarınızın her biri için kayan IP adresleri oluşturabilirsiniz.

Kayan IP adresi, istemcinin hangi HA grubunda çalıştığından bağımsız olarak bir DR/HA RDQM için aynı IP adresini kullanmasını sağlar. İki HA grubunuzun uygulama bağlantılığı için özel/yalıtılmış ağları varsa, her iki grup için de aynı kayan IP adresi tanımlanabilir. Yine de bu kayan IP adresini iki kez tanımlamanız gerekir; ancak, HA gruplarınızın her birinde bir kez.

Kayan IP adreslerini HA RDQM ile aynı yöntemi kullanarak oluşturur ve silersiniz. Bkz. “Kayan IP adresi oluşturma ve silme” sayfa 570.

**Linux** *DR/HA RDQM ' nin başlatılması, durdurulması ve durumunun görüntülenmesi*

Bir DR/HA RDQM ' nin geçerli durumunu başlatmak, durdurmak ve görüntülemek için standart IBM MQ denetim komutlarının değişkenlerini kullanabilirsiniz.

## Bu görev hakkında

DR/HA RDQM ' nin geçerli durumunu hem mqm hem de haClient gruplarına ait bir kullanıcı olarak başlatan, durduran ve görüntüleyen komutları çalıştırmanız gerekir.

Bu kuyruk yöneticisine ilişkin birincil düğümde bir kuyruk yöneticisini başlatmak ve durdurmak için komutları çalıştırmanız gerekir.

## Yordam

- RDQM 'yi başlatmak için RDQM' in birincil düğümünde şu komutu girin:

```
startmqm qmname
```



Burada *qmname* , başlatmak istediğiniz DR/HA RDQM ' nin adıdır.

RDQM başlatıldı ve Pacemaker RDQM ' yi yönetmeye başladı. Başka bir *strmqm* seçeneği belirtmek istiyorsanız, *-ns* seçeneğini *strmqm* ile belirtmeniz gerekir.

- Bir RDQM 'yi durdurmak için DR/HA RDQM' in birincil düğümünde şu komutu girin:

```
endmqm qmname
```

Burada *qmname* , durdurmak istediğiniz RDQM ' nin adıdır.

Pacemaker RDQM ' yi yönetmeyi durdurduktan sonra RDQM sona erdirilir. Bir RDQM durdurulurken diğer tüm **endmqm** parametreleri kullanılabilir.

- Bir RDQM ' nin durumunu görüntülemek için şu komutu girin:

```
dspmqr -m QMname
```

Çıkış olan durum bilgileri, komutu RDQM ' in birincil ya da ikincil düğümünde çalıştırıp çalıştırmadığınıza bağlıdır. Birincil düğümde çalıştırılırsa, **dspmqr** tarafından döndürülen normal durum iletilerinden biri görüntülenir. Komutu ikincil bir düğümde çalıştırırsanız, **Ended immediately** durumu görüntülenir. Örneğin, **dspmqr** RDQM7 düğümünde çalıştırılırsa, aşağıdaki bilgiler döndürülebilir:

```
QMNAME(DRQM8)                STATUS(Ended immediately)
QMNAME(DRQM7)                 STATUS(Running)
```

Bir RDQM ' in olağanüstü durumdan kurtarma için yapılandırılıp yapılandırılmadığını ve şu anda birincil mi, yoksa ikincil eşgörünüm mi olduğunu belirlemek için *dspmqr* ile bağımsız değişkenleri kullanabilirsiniz:

```
dspmqr -m QMname -o (dr | DR)
```

Aşağıdaki yanıtlardan biri görüntülenir:

#### **DRROLE()**

Kuyruk yöneticisinin olağanüstü durumdan kurtarma için yapılandırılmadığını gösterir.

#### **DRROLE(Primary)**

Kuyruk yöneticisinin DR birincil olarak yapılandırıldığını gösterir.

#### **DRROLE(Secondary)**

Kuyruk yöneticisinin DR ikincil olarak yapılandırıldığını gösterir.

DR/HA RDIMM ' lerine ilişkin olağanüstü durumdan kurtarma ve yüksek kullanılabilirlik bilgilerini görüntülemek için **dspmqr -o all** komutunu kullanın. Örneğin, DR/HA RDQM ' in çalıştığı düğümde **dspmqr -o all** komutunu çalıştırırsanız, aşağıdaki durum bilgilerini görürsünüz:

```
QMNAME(TESTQM1)                STATUS(Running) HA(Replicated)
DRROLE(Primary)
```

### **İlgili başvurular**

[dspmqr \(kuyruk yöneticilerini görüntüle\)](#)

[endmqm \(kuyruk yöneticisini sona erdir\)](#)

[strmqm \(kuyruk yöneticisini başlat\)](#)

### **V9.3.0 DR/HA yapılandırmalarında başarısız kaynak işlemleri**

Bir RDQM yüksek kullanılabilirlik yapılandırmasının Pacemaker bileşeni, HA grubundaki düğümlerden birinde bir kaynakla ilgili bir sorunla karşılaştığında başarısız olan kaynak işlemleri ortaya çıkar.

Bir RDQM DR/HA yapılandırmasındaki HA yapılandırmalarından birinde başarısız kaynak işlemleri ortaya çıkabilir. Başarısız olan kaynak işlemlerini görüntülemek için **rdqmstatus** komutunu ve bunları temizlemek için **rdqmclean** komutunu kullanabilirsiniz (hatanın nedeni çözüldükten sonra). Süreç, DR bileşeni olmayan RDQM HA yapılandırmalarıyla aynıdır. Daha fazla ayrıntı için bkz. [“Başarısız kaynak işlemleri”](#) sayfa 573 .

## İlgili görevler

“DR/HA RDQM ve HA grubu durumunun görüntülenmesi” sayfa 614

DR/HA eşlenmiş veri kuyruğu yöneticilerinin (RDIMM 'ler) HA durumunu ve DR rolünü görüntüleyebilirsiniz.

“RDQM ve HA grubu durumunun görüntülenmesi” sayfa 574

HA grubunun ve tek tek eşlenmiş veri kuyruğu yöneticilerinin (RDIMM 'ler) durumunu görüntüleyebilirsiniz.

## İlgili başvurular

[rdqmclean](#)

[rdqmstatus](#)

## Linux DR/HA RDQM ve HA grubu durumunun görüntülenmesi

DR/HA eşlenmiş veri kuyruğu yöneticilerinin (RDIMM 'ler) HA durumunu ve DR rolünü görüntüleyebilirsiniz.

## Bu görev hakkında

Tek tek RDIMM 'lerin durumunu görüntülemek ya da HA grubu tarafından bilinen tüm RDIMM'lerin durumuna ilişkin bir genel bakış elde etmek için **rdqmstatus** komutunu kullanın.

**V 9.3.0** Bir düğümün özet durumu, RDQM 'nin dayandığı DRBD çekirdek modülüyle ilgili bilgileri de görüntüler. RDQM 'yi yükselttiğinizde, sistemde çalışan RHEL çekirdeği sürümü için DRBD çekirdek modülünün doğru sürümünün kurulu olduğundan emin olmanız önemlidir. Durum, işletim sistemi çekirdeğinin sürümünü, DRBD modülünün oluşturulduğu çekirdek sürümünü, DRBD sürümünü ve DRBD çekirdek modülü yüklendi durumunu görüntüler.

**Not:** Bir HA/DR yapılandırmasında, HA yapılandırması her zaman zamanuyumsuz kopyalamayı kullanırken, DR yapılandırmasının her zaman zamanuyumlu eşlemeyi kullandığını unutmayın. Bu değerler, birleşik bir HA/DR yapılandırmasında `rdqmstatus -m qmgr` komutunun çıkışında görüntülenmez.

**rdqmstatus** komutunu çalıştırmak için `mqm` ve `haclient` gruplarında bir kullanıcı olmanız gerekir. Komutu, HA gruplarından herhangi birindeki düğümlerde çalıştırabilirsiniz.

## Yordam

- Bir düğümün ve HA yapılandırmasının parçası olan RDIMM 'lerin özet durumunu görüntülemek için:

```
rdqmstatus
```

Komutu çalıştırdığınız düğümün tanıtıcısı ve HA yapılandırmasındaki RDIMM 'lerin durumu ve bunların yürürlükteki DR rolü görüntülenir; örneğin:

```
Node: main-alice
OS kernel version: 3.10.0-1160.15.2
DRBD OS kernel version: 3.10.0-1160
DRBD version: 9.1.1
DRBD kernel module status: Loaded

Queue manager name: RDQM1
Queue manager status: Running elsewhere
HA current location: main-charlie
HA preferred location: main-charlie
HA blocked location: None

Queue manager name: RDQM9
Queue manager status: Running elsewhere
HA current location: main-bob
HA preferred location: main-bob
HA blocked location: None
DR role: Primary

Queue manager name: RDQM7
Queue manager status: Running
HA current location: This node
```

HA preferred location:	This node
HA blocked location:	None
DR role:	Primary

Bu örnekte, RDQM7 ve RDQM8 hem DR/HA RDM, hem de RDQM1 olağanüstü durumdan kurtarma yerine geçebilecek şekilde yapılandırılmamış bir HA RDQM 'dir.

**V 9.3.0** DRBD çekirdek modülü durumu aşağıdaki değerlerden biridir:

#### Yüklendi

DRBD modülünün yüklendiğini gösterir.

#### Kısmen yüklendi

DRBD modülü yüklendiğinde ortaya çıkabilir, ancak bir uyumsuzluk nedeniyle düzgün çalışmayabilir.

#### Yüklenmedi

DRBD modülü yüklenmedi. Bu, henüz bir RDQM kuyruk yöneticisi yaratılmadığında, yeni kurulan bir yapılandırmada görüntülenebilir.

#### Kurulu değil

DRBD modülünden birinin kurulu olmadığını gösterir. ya da IBM MQ , DRBD modülünün işletim sistemi çekirdeği sürümünü belirleyemedi.

#### Önceden kurulan sürüm yüklenmeye devam ediyor

Bu durum, var olan DRBD modülü çalışırken (yani bir RDQM kuyruk yöneticisi çalışırken) yeni bir DRBD modülü kurulursa ortaya çıkabilir. Yeni kurulan modül, gerçekte çalışmakta olan modül değil, durumda bildirilir.

- HA grubundaki tüm düğümlerde belirli bir kuyruk yöneticisinin durumunu görüntülemek için şu komutu girin:

```
rdqmstatus -m qmname
```

Burada *qmname* , durumunu görüntülemek istediğiniz RDQM ' nin adıdır. Geçerli düğümdeki RDQM ' nin durumu görüntülenir ve bunu, geçerli düğümün perspektifinden diğer iki düğümün durumunun bir özeti izler.

**V 9.3.0**

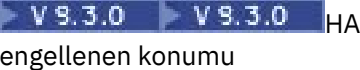
- Başarısız olan kaynak işlemlerinin ayrıntıları da içinde olmak üzere, HA grubundaki tüm düğümlerde belirli bir kuyruk yöneticisinin durumunu görüntülemek için aşağıdaki komutu girin:

```
rdqmstatus -m qmname -a
```

Burada *qmname* , durumunu görüntülemek istediğiniz RDQM ' nin adıdır. Geçerli düğümdeki RDQM ' nin durumu görüntülenir ve bunu, geçerli düğümün perspektifinden diğer iki düğümün durumunun bir özeti izler. Bunu, RDQM ile ilişkili başarısız olan kaynak işlemleriyle ilgili ayrıntılar izler.

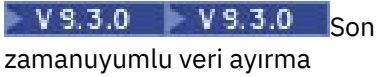
Aşağıdaki tablo, bir RDQM için `rdqmstatus -m qmname` komutu tarafından döndürülebilecek geçerli düğümle ilgili bilgileri özetler.

Çizelge 35. Geçerli düğüm durumu		
Durum özniteliği	Olası değerler	Görüntülendiğinde
Düğüm adı	<i>Düğüm Adı</i>	Her zaman görüntülenir
Kuyruk yöneticisi durumu	kuyruk yöneticisinin durumu ( <b>dspm</b> komutu için geçerli durumlardan biri)	Her zaman görüntülenir
CPU	<i>n.nn%</i>	Yalnızca bu düğümde RDQM çalışırken gösterilir
Bellek	Kullanılan <i>nnn</i> MB	Yalnızca bu düğümde RDQM çalışırken gösterilir

Çizelge 35. Geçerli düğüm durumu (devamı var)		
Durum özneliği	Olası değerler	Görüntülediğinde
Kuyruk yöneticisi dosya sistemi	<i>nnn</i> MB kullanıldı, <i>y.y</i> GB ayrıldı [ <i>z</i> %]	Yalnızca bu düğümde RDQM çalışırken gösterilir
HA rolü	Birincil İkincil Bilinmeyen	Her zaman görüntülenir
HA durumu	Yedekteki tüm düğümler Bu düğüm yedekte bekliyor Yedek veritabanındaki uzak düğümler Karışık	Yedekteki tüm düğümler Yedek veritabanında yürürlükteki düğüm Her iki uzak düğüm de yedekte Her uzak düğüm için farklı durum
HA denetimi	Etkin Geçersiz Kılındı Bilinmeyen	Her zaman görüntülenir. RDQM ' in Pacemaker denetimi altında olup olmadığını gösterir
HA tercih edilen konumu	Yok Bu düğüm Bilinmeyen <i>Düğüm Adı</i>	Her zaman görüntülenir
 HA engellenen konumu	Yok-Kuyruk yöneticisinin herhangi bir düğümde çalışması engellenmedi Bu düğüm-Bir ya da daha fazla başarısız kaynak işlemi nedeniyle, kuyruk yöneticisinin yürürlükteki düğümde çalışması engellendi <i>nodename</i> -Başarısız olan bir ya da daha çok kaynak işlemi nedeniyle, kuyruk yöneticisinin <i>nodename</i> üzerinde çalışması engellendi. <i>nodename1, nodename2</i> -Bir ya da daha fazla başarısız kaynak işlemi nedeniyle, kuyruk yöneticisinin <i>nodename1</i> ve <i>nodename2</i> üzerinde çalışması engellendi. Tüm düğümler-Bir ya da daha fazla başarısız kaynak işlemi nedeniyle kuyruk yöneticisinin tüm düğümlerde çalışması engellendi	Her zaman görüntülenir
HA kayan IP arabirimi	<i>Arabirim_adi</i>	Her zaman görüntülenir
HA kayan IP adresi	<i>IPV4_address</i>	Her zaman görüntülenir
DR rolü	Birincil İkincil İkincil beklemede Bilinmeyen	Her zaman görüntülenir

Çizelge 35. Geçerli düğüm durumu (devamı var)

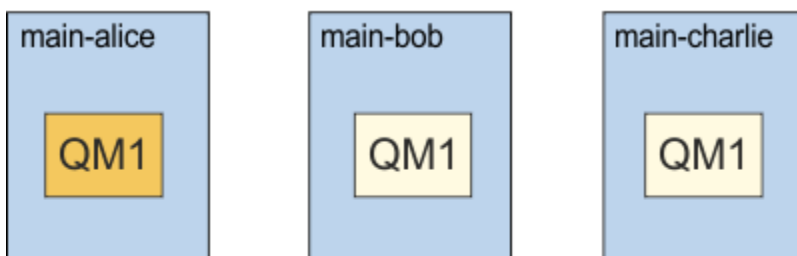
Durum özneliği	Olası değerler	Görüntülediğinde
DR durumu	Olağan Eşitleme devam ediyor Bölümlenmiş  Uzak sistem kullanılmıyor  Tutarsız  Anlık görüntüye geri dönülüyor  Uzak sistemin konfigürasyonu tanımlanmadı  Anlaşma başarısız oldu	Her şey yolunda. Eşitleme devam ediyor. Kullanıcı kuyruğu başlattı her düğümde yönetici DR eşleme ağı kullanılamaz. Diğer düğüme bağlantı kaybolmuş. Bir eşitleme devam ediyordu, ancak kesintiye uğradı. Kullanıcı, şu kullanıcıya geri dönmeyi seçti şu zaman alınan anlık görüntü Kuyruk yöneticisi Tutarsız durum. Birincil konfigürasyon tanımlandı Ama ikincinin yok. Birincil ve ikincil düğümler arasındaki ilk anlaşma başarısız oldu. Bunun nedeni uyumsuz eşleme tipleri ya da ikincil düğüm daha küçük bir dosya sistemi büyüklüğüyle yapılandırıldıysa olabilir.
DR durumu (HA ikincil düğümünde)	Bkz. <i>HA_Primary_Node</i>	DR durumu yalnızca HA birincil düğümünde bilindiği için HA ikincil düğümlerinde görüntülenir.
DR kapısı	Bu kuyruk yöneticisine ilişkin verileri eşlemek için kullanılan TCP/IP kapısı.	Her zaman görüntülenir.
DR yerel IP adresi	Bu kuyruk yöneticisinin DR eşlemesi için kullanacağı yerel IP adresi	Her zaman görüntülenir.
DR uzak IP adresi listesi	Bu kuyruk yöneticisinin DR eşlemesi için kullanacağı uzak IP adresleri. Üç IP adresinin virgülle ayrılmış listesi.	Her zaman görüntülenir.
DR yürürlükteki uzak IP adresi	Bu kuyruk yöneticisinin DR eşlemesi için bağlı olduğu yürürlükteki uzak IP.	Etkin DR bağlantısı olan bir HA birincil için.
DR yürürlükteki uzak IP adresi (HA ikincil düğümünde)	Bkz. <i>HA_Primary_Node</i>	DR bağlantısı yalnızca HA Birincil düğümünde olduğundan HA İkincil düğümünde görüntülenir
Olağanüstü durumdan kurtarma eşitlenmemiş verileri	xKB	Uzak düğüm kullanılmadığında ya da tutarsız olduğunda görüntülenir.
DR eşitlemesi ilerleme durumu	y%	Bir eşitleme işlemi devam ederken görüntülenir.

Çizelge 35. Geçerli düğüm durumu (devamı var)		
Durum özneteliği	Olası değerler	Görüntülediğinde
Olağanüstü durumdan kurtarma tahmini tamamlanma süresi	dd/MM/yyyy HH:mm:ss	Bir eşitleme işlemi devam ederken görüntülenir.
Anlık görünüm yeniden sürümü ilerleme durumu	y%	DR durumu "Anlık görüntüye geri dönülüyor" olduğunda görüntülenir
 Son zamanuyumlu veri ayırma	dd/MM/yyyy HH:mm:ss	DR verileri eşitlenmemiş olduğunda (ilk eşitlemeden sonra) görüntülenir. Verilerin en son eşitlendiği tarihi ve saati verir.

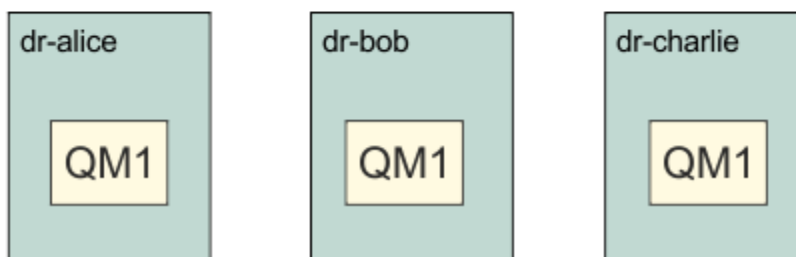
### Örnek

Bu örneklerde, aşağıdaki DR/HA yapılandırmasının çeşitli düğümlerinde çalıştırılan **rdqmstatus -m qm1** komutu gösterilmektedir:

#### main site



#### dr site



DR birincil ve HA birincil olan bir düğümdeki normal durum örneği:

```

Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: Normal
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: 192.168.2.1

```

```
Node:                main-bob
HA status:           Normal

Node:                main-charlie
HA status:           Normal
```

DR birincil ve HA ikincil olan bir düğümdeki normal durum örneği:

```
Node:                main-bob
Queue manager status: Running elsewhere
HA role:             Secondary
HA status:           Normal
HA control:          Enabled
HA current location: main-alice
HA preferred location: main-alice
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role:             Primary
DR status:           See main-alice
DR port:             3000
DR local IP address: 192.168.1.2
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: See main-alice

Node:                main-alice
HA status:           Normal

Node:                main-charlie
HA status:           Normal
```

DR ikincil ve HA birincil olan bir düğümdeki normal durum örneği:

```
Node:                dr-alice
Queue manager status: Ended immediately
HA role:             Primary
HA status:           Normal
HA control:          Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role:             Secondary
DR status:           Normal
DR port:             3000
DR local IP address: 192.168.2.1
DR remote IP address list: 192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address: 192.168.1.1

Node:                dr-bob
HA status:           Normal

Node:                dr-charlie
HA status:           Normal
```

DR ikincil ve HA ikincil olan bir düğümdeki normal durum örneği:

```
Node:                dr-bob
Queue manager status: Ended immediately
HA role:             Secondary
HA status:           Normal
HA control:          Enabled
HA current location: dr-alice
HA preferred location: dr-alice
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role:             Secondary
DR status:           See dr-alice
DR port:             3000
DR local IP address: 192.168.2.2
DR remote IP address list: 192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address: See dr-alice

Node:                dr-alice
HA status:           Normal
```

```
Node: dr-charlie
HA status: Normal
```

DR birincil ve HA birincil olan bir düğümde devam eden DR eşitlemesi örneği:

```
Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: Normal
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: 192.168.2.1
DR synchronization progress: 11.0%
DR estimated time to completion: 2018-09-06 14:55:05

Node: main-bob
HA status: Normal

Node: main-charlie
HA status: Normal
```

DR birincil ve HA birincil olan bir düğümde bölümlenmiş DR örneği:

```
Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: Partitioned
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: 192.168.2.1
DR out of sync data: 372KB

Node: main-bob
HA status: Normal

Node: main-charlie
HA status: Normal
```

**V9.3.0** DR birincil ve HA birincil olan bir düğümde zamanuyumsuz veri ayırma örneği:

```
Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
```



```

DR role: Primary
DR status: Remote unavailable
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: Unknown
DR out of sync data: 372KB
DR last in sync: 2020-02-02 20:22:02

Node: main-bob
HA status: Normal

Node: main-charlie
HA status: Normal

```

### V 9.3.0 DR birincil ve HA birincil olan bir düğümde eşitlenmemiş HA örneği:

```

Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: Normal
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: 192.168.2.1

Node: main-bob
HA status: Inconsistent
HA out of sync data: 15932KB
HA last in sync: 2020-02-02 20:22:02

Node: main-charlie
HA status: Normal

```

**V 9.3.0** İşletim sistemi çekirdek sürümü (RHEL 7.9) ile DRBD çekirdek modülü (RHEL 7.8 için hedeflenmiş) arasındaki uyumsuzluğu gösteren bir özet durumu örneği. Durum, DRBD çekirdek modülünün yüklendiğini ve kuyruk yöneticisinin çalıştığını bildirirse de, DRBD çekirdek modülünü bu durumda çalışan işletim sistemi çekirdeği için hedeflenen sürümle güncellemelisiniz.

```

Node: main-alice
OS kernel version: 3.10.0-1160.15.2
DRBD OS kernel version: 3.10.0-1127
DRBD version: 9.1.1
DRBD kernel module status: Loaded

Queue manager name: QM1
Queue manager status: Running
HA current location: This node
HA preferred location: This node
HA blocked location: None
DR role: Primary

```

**V 9.3.0** İşletim sistemi çekirdek sürümü (RHEL 7.9) ile DRBD çekirdek modülü (RHEL 7.6 için hedeflenen) arasındaki uyumsuzluğu gösteren bir özet durumu örneği. Bu örnekte sürüm uyumsuzluğu daha ciddi ve DRBD çekirdek modülü başarıyla yüklenemedi. QM1 bir HA/DR kuyruk yöneticisidir ve başka bir düğüme geçer, HA durumu bilinmiyor ve DR durumu bilinmiyor. Bu hatayı çözmek için DRBD çekirdek modülü, çalışan işletim sistemi çekirdeğine ilişkin sürüm hedefiyle güncellenmelidir.

```

Node: main-alice
OS kernel version: 3.10.0-1160.15.2
DRBD OS kernel version: 3.10.0-957
DRBD version: 9.1.2+ptf.3
DRBD kernel module status: Partially loaded

```

```
Queue manager name: QM1
Queue manager status: Running elsewhere
HA status: Unknown
HA current location: main-bob
HA preferred location: This node
HA blocked location: None
DR role: Primary
DR status: Unknown
```

## İlgili başvurular

Linux [rdqmstatus](#)

### Linux **DR/HA ortamında çalışma**

Bir DR/HA ortamında çalışırken, yüksek düzeyde kullanılabilirlik ve olağanüstü durumdan kurtarma için ayrı konular vardır.

Bir DR/HA RDQM ' in çalıştığı düğüm başarısız olursa, RDQM otomatik olarak bu HA grubundaki başka bir düğüme geçemez. Sitenin tamamı başarısız olursa, kurtarma yerindeki HA grubundaki tercih edilen düğümden RDQM ' yi ile başlatmanız gerekir. Burada sıradan bir DR RDQM ile ilgili olarak dikkat edilecek noktalar aynıdır, daha fazla bilgi için bkz. [“Olağanüstü durumdan kurtarma ortamında çalışma” sayfa 600](#) .

Düğümlerden biri tamamen arızalanırsa ve değiştirilmesi gerekirse, yönergeler için bkz. [“Olağanüstü durumdan kurtarma yapılandırmasında arızalı bir düğümün değiştirilmesi” sayfa 602](#) ve [“Yüksek kullanılabilirlikli yapılandırmada hatalı bir düğümün değiştirilmesi” sayfa 581](#) .

### Linux **DR/HA yapılandırmasındaki hatalı bir düğümün değiştirilmesi**

HA gruplarınızdan birindeki düğümlerden biri arızalanırsa, bunu değiştirebilirsiniz.

## Bu görev hakkında

Yordam, değiştirmekte olduğunuz düğümün DR yapılandırmasında birincil mi, yoksa ikincil mi olduğuna göre değişir. Her iki durumda da, yeni düğüm değiştirmekte olduğunuz düğümlerle aynı yapılandırmaya sahip olmalıdır; yani, aynı anasistem adına, aynı IP adreslerine sahip olmalıdır.

Ana ya da kurtarma yerinizde HA grubunu tamamen kaybetmiş ve tüm HA grubunu değiştirmeniz gereken bir durumla da karşılaşabilirsiniz.

## Yordam

- DR yapılandırmasında birincil olan bir yedek düğüm için, yeni düğümden aşağıdaki adımları tamamlayın:
  - a) Diğer düğümlerdeki dosyalarla eşleşen bir `rdqm.ini` dosyası oluşturun ve `rdqmadm -c` komutunu çalıştırın (bkz. [“Pacemaker kümesinin tanımlanması \(HA grubu\)” sayfa 557](#)).
  - b) Her bir DR/HA RDQM ' yi yeniden yaratmak için `crtmqm -sxs -rr p qmanager` komutunu çalıştırın (bkz. [“DR/HA RDIMM ' leri oluşturma” sayfa 608](#)).
- DR yapılandırmasında ikincil olan bir yedek düğüm için, yeni düğümden aşağıdaki adımları tamamlayın:
  - a) Diğer düğümlerdeki dosyalarla eşleşen bir `rdqm.ini` dosyası oluşturun ve `rdqmadm -c` komutunu çalıştırın (bkz. [“Pacemaker kümesinin tanımlanması \(HA grubu\)” sayfa 557](#)).
  - b) Her bir DR/HA RDQM ' yi yeniden yaratmak için `crtmqm -sx -rr s qmanager` komutunu çalıştırın (bkz. [“DR/HA RDIMM ' leri oluşturma” sayfa 608](#)).
- Tüm HA grubunu değiştirmek için aşağıdaki adımları tamamlayın:
  - a) DR birincil yerinde (ana yer) tüm HA grubunu kaybederseniz, DR/HA RQM ' lerinizi çalıştırmaya devam etmek için DR ikincil yerine yönetilen bir yedek yere geçiş gerçekleştirmek için adımları izlemeniz gerekir (bkz. [“Olağanüstü durumdan kurtarma ortamında çalışma” sayfa 600](#)). (Kurtarma yerinde bir HA grubunun tamamını kaybederseniz, DR/HA RDIMM ' leriniz müdahaleniz olmadan ana sitede çalışmaya devam eder.)
  - b) Üç yeni düğümünüzdeki HA grubunu yeniden oluşturun (bkz. [“DR/HA RDIMM ' leri için HA gruplarının yapılandırılması” sayfa 607](#)).

- c) Yeni HA grubunda DR/HA RDIMM ' lerinizi “DR/HA RDIMM ' leri oluşturma” sayfa 608’inde açıklandığı gibi yeniden yaratın.
- d) Gerekirse, kurtarma yerinizden ana yerinize kadar yönetilen bir hata durumunda yedek sisteme geçiş işlemi gerçekleştirin.

### Linux DR/HA RDQM çalıştı örneği

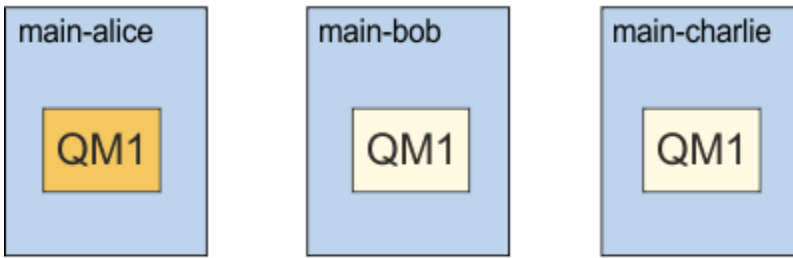
Bu örnek, DR/HA RDQM ' nın nasıl yaratılacağını ve silineceğini gösterir.

### DR/HA RDQM yaratılması

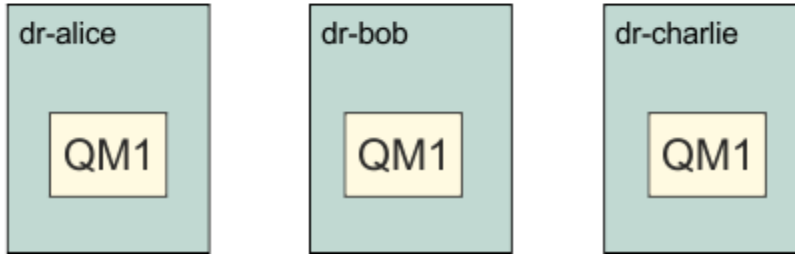
Örnek yapılandırma 'main' ve 'dr' adlı iki yer vardır. Her sitenin 'alice', 'bob' ve 'charlie' adında üç düğümü vardır. Düğümlerin tam adı site adı ve adından oluşmuştur, bu nedenle 'main-alice', 'dr-alice' vb.

Aşağıdaki adımlar, ana alice üzerinde çalışan QM1 adlı bir DR/HA RDQM yaratır. Ana-alice düğümü HA ve DR birincil düğümdür.

#### main site



#### dr site



rdqm.ini dosyasında yerel ve uzak DR IP adresleri belirtilirse, komut satırında herhangi bir IP adresi belirtilmesi gerekmez ve ana hat üzerinde aşağıdaki komut çalıştırılarak QM1 adlı bir DR/HA RDQM yaratılabilir:

```
crtmqm -sx -rr p -rn DR1 -rp 7001 QM1
```

rdqm.ini dosyasında yerel DR IP adresleri belirtilirse, komut satırında uzak DR IP adresleri belirtilebilir:

```
crtmqm -sx -rr p -ri 192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

rdqm.ini dosyasında DR IP adresi belirtilmezse, komut satırında hem uzak hem de yerel DR IP adresleri belirtilebilir:

```
crtmqm -sx -rr p -rl 192.168.1.1,192.168.1.2,192.168.1.3 -ri  
192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

QM1 yaratılmasına yanıt olarak çıkış aşağıdaki örnekte gösterildiği gibidir:

```
Creating replicated data queue manager configuration.  
Secondary queue manager created on 'main-bob'.  
Secondary queue manager created on 'main-charlie'.
```

```
IBM MQ queue manager created.
Directory '/var/mqm/vols/qm1/qmgr/qm1' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QM1'.
Default objects statistics : 83 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
Enabling replicated data queue manager.
Replicated data queue manager enabled.
Issue the following command on the remote HA group to create the DR/HA secondary queue manager:
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

Dr-alice üzerinde QM1 ' in DR ikincil örneğini yaratmak için komutu iletiden kopyalayın:

```
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

Aşağıdaki ileti dr-alice üzerinde çıkışa yazılır:

```
Creating replicated data queue manager configuration.
Secondary queue manager created on 'dr-bob'.
Secondary queue manager created on 'dr-charlie'.
IBM MQ secondary queue manager created.
Enabling replicated data queue manager.
```

## DR İkincil Durumunu Test Et

QM1'in olağanüstü durumdan kurtarma özelliklerini sınamak için, QM1 ' i DR ikincil eşgörünümü yapmak üzere main-alice 'de aşağıdaki komutu çalıştırın:

```
rdqmdr -m QM1 -s
Queue manager 'QM1' has been made the DR secondary on this node.
```

QM1 ' i o düğümdeki DR birincil örneği yapmak için dr-alice üzerinde aşağıdaki komutu çalıştırın:

```
rdqmdr -m QM1 -p
Queue manager 'QM1' has been made the DR primary on this node.
```

## DR/HA RDQM ' in silinmesi

QM1adlı DR/HA RDQM ' yi silmek için önce ana hat üzerinde kuyruk yöneticisini sona erdirin:

```
endmqm -w QM1
Replicated data queue manager disabled.
Waiting for queue manager 'QM1' to end.
IBM MQ queue manager 'QM1' ended.
```

Daha sonra QM1:

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'main-bob'.
Secondary queue manager deleted on 'main-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

Son olarak, dr-alice üzerinde QM1 ögesini silmeniz gerekir:

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'dr-bob'.
Secondary queue manager deleted on 'dr-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

## İlgili kavramlar

[“DR/HA ortamında çalışma” sayfa 622](#)

Bir DR/HA ortamında çalışırken, yüksek düzeyde kullanılabilirlik ve olağanüstü durumdan kurtarma için ayrı konular vardır.

### İlgili görevler

“DR/HA RDIMM ' leri oluşturma” sayfa 608

Bir DR/HA yapılandırmasında eşlenmiş bir veri kuyruğu yöneticisi (RDQM) oluşturmak için **crtmqm** komutunu kullanırsınız.

“DR/HA RDQM ' in silinmesi” sayfa 612

Bir DR/HA eşlenmiş veri kuyruğu yöneticisini (RDQM) silmek için **dlmqm** komutunu kullanırsınız.

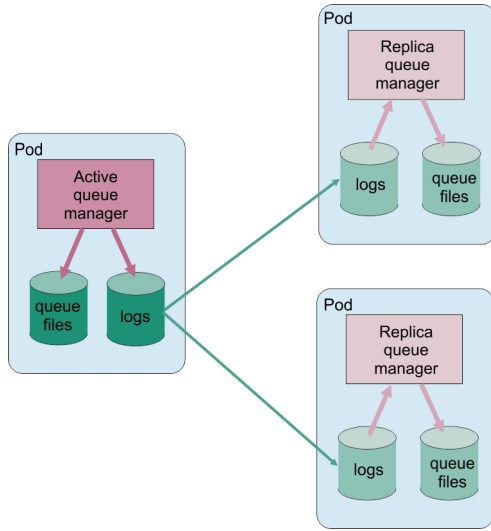
## CP4I MQ Adv. Yerel HA

Yerel HA, IBM MQ' in kapsayıcı devreye alımlarında bulunan yüksek kullanılabilirlikli bir çözümdür.

Yerel HA yapılandırması, her biri kuyruk yöneticisinin bir eşgörünümü olan üç düğümden (örneğin, üç Kubernetes bölmesi) oluşur. Bir yönetim ortamı, etkin kuyruk yöneticisi, iletileri işliyor ve günlüğüne yazıyor. Günlük yazıldığında, etkin kuyruk yöneticisi verileri 'eşlemeler' olarak bilinen diğer iki örneğe gönderir. Her eşleme kendi günlüğüne yazar, verileri onaylar ve eşlenen günlükten kendi kuyruk verilerini günceller. Etkin kuyruk yöneticisini çalıştıran düğüm başarısız olursa, kuyruk yöneticisinin eşleme eşgörünümlerinden biri etkin rolü devralır ve üzerinde çalışılmakta olan verileri içerir.

Ayrıntılı bir genel bakış için, bu belgenin Kapsayıcılar bölümündeki [Yerel HA](#) bölümüne bakın.

Aşağıdaki şekil, üç taşıyıcıda konuşlandırılan bir kuyruk yöneticisinin üç eşgörünümünü içeren tipik bir konuşlandırmayı göstermektedir.



Şekil 83. Yerel HA yapılandırması örneği

## CP4I MQ Adv. Native HA çözümünün oluşturulması

Yerel HA çözümü oluşturmak için önerilen yöntem, IBM MQ Operatorkullanılarak sağlanır. Diğer bir seçenek olarak, kendi taşıyıcılarınızı yaratabilir ve Yerel HA ' yı el ile yapılandırabilirsiniz.

**Not:** Bu bilgiler yalnızca kapsayıcı ortamlar için geçerlidir.

IBM MQ Operatorkullanılarak bir Native HA çözümü oluşturmak için genel bakış için [Native HA](#) ve ayrıntılı yönergeler için [Example: Configuring a Native HA queue manager](#) başlıklı konuya bakın.

Kendi kapsayıcılarınızı yaratmak ve Yerel HA ' yı el ile yapılandırmak için [Kendi kapsayıcılarınızı oluşturuyorsanız Yerel HA grubunun yaratılması](#) başlıklı konuya bakın.

## CP4I MQ Adv. Yerel HA kuyruk yöneticileri sona erdiriliyor

Kapsayıcılarda IBM MQ için, Yerel HA grubunun bir parçası olan bir etkin ya da eşleme kuyruk yöneticisini sona erdirmek için **endmqm** komutunu kullanabilirsiniz.

## Bu görev hakkında

**Not:** Bu bilgiler yalnızca kapsayıcı ortamlar için geçerlidir.

Yerel HA grubunun bir parçası olan bir kuyruk yöneticisinin durdurulmasına ilişkin yordam, bunun etkin bir eşgörünüm mü, yoksa bir eşleme eşgörünümü mü olduğuna bağlıdır. Her bir yönetim ortamı tipini sona erdirdiğinizde, yönetim ortamını sonlandırmanın Yerli HA grubunun yetersayısının yetersayısının bozulmamasını sağlamak için bir denetim yapılır. Çekirdek bozulacaksa, **endmqm** komutu başarısız olur.

Bir **endmqm** komutu verdiğinizde, gruptaki diğer eşgörünümler bunun gerçekleştiği konusunda uyarılır, böylece bağlantı kesildiğinde hata bildirmezler.

Etkin bir eşgörünüm, çok sayıda eşleme örneğinin sona ermesi ya da bağlantısının kesilmesi nedeniyle yetersayı kaybederse, etkin eşgörünüm, tamamen sona ermeden önce yapılandırılabilir bir süre bekler. Bu, uygulamaların yalnızca bağlantılarının kesilmesi yerine, belirli bir zaman döneminin işlenmesini zarif bir şekilde kapatmasına olanak tanır. Bu zaman aşımı değeri, `qm.ini` dosyasının `NativeHALocalInstance` kısmına ilişkin `QuorumConnectivityTimeout` özneliği tarafından belirtilebilir. Varsayılan değer 0 saniyedir.

## Yordam

- Bir kuyruk yöneticisinin etkin yönetim ortamını sona erdirmek için, etkin yönetim ortamının çalıştığı düğümde aşağıdaki komutu verin:

```
endmqm -s QMgrName
```

- İstemci uygulamalarının başka bir yönetim ortamına yeniden bağlanmasına yardımcı olmak için `-x` seçeneğini belirtin.
- Bu yönetim ortamı Yerel HA grubundaki etkin yönetim ortamı değilse, komut başarısız olur.
- Bu etkin yönetim ortamının sona erdirilmesi, grup yetersayısının başarısız olmasına neden olursa, komut başarısız olur. (Siz bu komutu çalıştırırken diğer yönetim ortamları sona erer ya da kullanılamaz duruma gelirse, yetersayı denetimi bunu saptayamayabilir, Yerli HA grubu sona erer ve yalnızca yeterli yönetim ortamı kullanılabilir olduğunda yeniden başlatılabilir.)

Etkin kuyruk yöneticisi sona erdiğinde, eşleme eşgörünümlerinden biri etkin rolü devralır. Hangi eşlemenin devralınacağını belirtemezsiniz; bu, grup içindeki kararlaştırmaya göre belirlenir ve hangilerinin en güncel işlem günlüklerine sahip olduğuna bağlıdır.

- Bir kuyruk yöneticisinin eşleme örneğini sona erdirmek için aşağıdaki komutu verin:

```
endmqm -x QMgrName
```

- Bu yönetim ortamı etkin yönetim ortamıyla oluşursa, komut başarısız olur.
- Bu eşleme örneğinin sona erdirilmesi grup yetersayısının başarısız olmasına neden olursa, komut başarısız olur. (Siz bu komutu çalıştırırken diğer yönetim ortamları sona erer ya da kullanılamaz duruma gelirse, yetersayı denetimi bunu saptayamayabilir, Yerli HA grubu sona erer ve yalnızca yeterli yönetim ortamı kullanılabilir olduğunda yeniden başlatılabilir.)

**Not:** Yerli HA yönetim ortamlarındaki **endmqm** komutuyla birlikte `-c`, `-i`, `-p` ya da `-w` anahtarlarını, hangi rolde olduklarından bağımsız olarak kullanabilirsiniz. Kuyruk yöneticisi eşgörünümü, grup yetersayısı üzerindeki etkiyi yoksayararak sona erer. Ancak, bilgiler gruptaki diğer örneklerle paylaşılır. Bu anahtarları etkin örnek için `-s` ile birlikte kullanabilirsiniz. Bu anahtarları, eşleme eşgörünümleri için `-x` anahtarıyla birlikte kullanamazsınız.

## İlgili başvurular

[endmqm \(kuyruk yöneticisini sona erdir\)](#)

## Günlüğe kaydetme: İletilerin kaybolmadığından emin olmak

IBM MQ , kuyruk yöneticisi tarafından denetlenen kalıcı verilerdeki tüm önemli değişiklikleri bir kurtarma günlüğüne kaydeder.

Bu, nesnelere oluşturulmasını ve silinmesini, kalıcı ileti güncellemelerini, işlem durumlarını, nesne özniteliklerinde yapılan değişiklikleri ve kanal etkinliklerini içerir. Günlük, ileti kuyruklarına ilişkin tüm güncellemeleri kurtarmak için gereken bilgileri içerir:

- Kuyruk yöneticisi değişikliklerinin kayıtlarını tutma
- Yeniden başlatma işlemi tarafından kullanılmak üzere kuyruk güncellemelerinin kayıtlarını tutma
- Donanım ya da yazılım hatasından sonra verileri geri yüklemenizi sağlama

Ancak IBM MQ , günlük dosyaları da içinde olmak üzere, dosyalarını barındıran disk sistemine de dayanır. Disk sisteminin kendisi güvenilmezse, günlük bilgileri de içinde olmak üzere bilgiler kaybolmaya devam edebilir.



**DİKKAT:** Kurtarma günlüklerini farklı bir işletim sistemine taşıyamazsınız.

## Günlüklerin nasıl görüldüğünü

Günlükler, birincil ve ikincil dosyalardan ve bir denetim dosyasından oluşur. Günlük dosyalarının sayısını ve boyutunu ve bunların dosya sisteminde nerede saklandığını tanımlarsınız.

IBM MQ günlüğü iki bileşenden oluşur:

1. Bir ya da daha çok günlük verisi dosyası.
2. Günlük denetim dosyası

Günlük verileri dosyası, günlük kapsamı olarak da bilinir.

Kaydedilmekte olan verileri içeren bir dizi günlük kapsamı vardır. Sayı ve boyutu ( "mqs.ini dosyasının LogDefaults kısmı" sayfa 92 içinde açıklandığı gibi) tanımlayabilir ya da üç birincil ve iki ikincil kapsamın sistem varsayılan değerini alabilirsiniz.

Üç birincil ve iki ikincil kapsamın her biri varsayılan olarak 16 MB ' ye ayarlanır.

Bir kuyruk yöneticisi yarattığınızda, önceden ayrılmış günlük kapsamı sayısı, ayrılmış *birincil* günlük kapsamı sayısıdır. Bir sayı belirtmezseniz, varsayılan değer kullanılır.

IBM MQ iki tip günlük kaydı kullanır:

- Dairesel
- Doğrusal

Doğrusal günlük kaydı ile kullanılan günlük kapsamı sayısı, ortam görüntüsü kaydetme sıklığına bağlı olarak çok büyük olabilir.

Ek bilgi için bkz. "Günlüğe kaydetme tipleri" sayfa 628 .

**ALW** IBM MQ for AIX or Linux sistemlerinde günlük yolunu değiştirmediyse, günlük kapsamları dizin altında yaratılır:

```
/var/mqm/log/QMgrName
```

**Windows** IBM MQ for Windows içinde, günlük yolunu değiştirmediyse, günlük kapsamları dizin altında yaratılır:

```
C:\ProgramData\IBM\MQ\log\QMgrName
```

IBM MQ bu birincil günlük kapsamlarıyla başlar, ancak birincil günlük alanı yeterli değilse *ikincil* günlük kapsamlarını ayırır. Bunu dinamik olarak yapar ve günlük alanı talebi azaldığında bunları kaldırır. Varsayılan olarak, en çok iki ikincil günlük kapsamı ayrılabilir. Bu varsayılan ayırmayı, "Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerini değiştirme" sayfa 83 içinde açıklandığı gibi değiştirebilirsiniz.

Günlük kapsamlarına S harfi ya da Rharfi eklenir. Etkin, etkin olmayan ve gereksiz kapsamların başına Söneki konurken, yeniden kullanım kapsamlarına Röneki eklenir.

Kuyruk yöneticinizi yedeklerken ya da geri yüklerken, günlük denetim dosyasıyla birlikte tüm etkin, etkin olmayan ve gereksiz kapsamları yedekleyin ve geri yükleyin.

**Not:** Yeniden kullanım kapsamlarını yedeklemeniz ve geri yüklemeniz gerekmez.

### **Günlük denetim dosyası**

Günlük denetim dosyası, günlük kapsamlarının durumunu (boyut ve konumu ve sonraki kullanılabilir kapsamın adı gibi) tanımlamak için gereken bilgileri içerir.

**Önemli:** Günlük denetim dosyası yalnızca iç kuyruk yöneticisi kullanımı içindir.

Kuyruk yöneticisi, kurtarma günlüğünün durumuyla ilişkili denetim verilerini günlük denetim dosyasında tutar ve günlük denetim dosyasının içeriğini değiştirmemelisiniz.

Günlük denetim dosyası günlük yolunda ve amqhlct1.lf olarak adlandırılır. Kuyruk yöneticinizi yedeklerken ya da geri yüklerken, günlük denetim dosyasının günlük kapsamlarıyla birlikte yedeklendiğinden ve geri yüklendiğinden emin olun.

### **Günlüğe kaydetme tipleri**

IBM MQ 'de kuyruk yöneticisi etkinliklerinin kayıtlarını tutmanın iki yolu vardır: çevrimsel günlük kaydı ve doğrusal günlük kaydı. Üçüncü bir günlük kaydı tipi (eşlendi), yalnızca Yerel HA yapılandırmaları tarafından kullanılır.

### **Döngüsel günlük kaydı**

Tüm istediğiniz kurtarmayı yeniden başlatmaksa, sistem durduğunda devam eden hareketleri geri almak için günlüğü kullanarak döngüsel günlük kaydını kullanın.

Döngüsel günlük kaydı, tüm yeniden başlatma verilerini bir günlük dosyaları halkasında tutar. Günlüğe kaydetme, halkadaki ilk dosyayı doldurur, sonra sonrakine geçer ve tüm dosyalar doluncaya kadar devam eder. Daha sonra ringdeki ilk dosyaya geri döner ve yeniden başlar. Bu, ürün kullanımında olduğu sürece devam eder ve günlük dosyalarınız hiç bitmez.

IBM MQ , kuyruk yöneticisi verilerinin kurtarılmasını sağlamak için artık gerekli olmadıkça, kuyruk yöneticisini yeniden başlatmak için gereken günlük girişlerini veri kaybı olmadan tutar. Günlük dosyalarını yeniden kullanmak üzere serbest bırakma mekanizması "[Tam kurtarma için denetim gösterimini kullanma](#)" sayfa 630içinde açıklanmıştır.

### **Doğrusal günlük kaydı**

Hem kurtarmayı yeniden başlatmak hem de ortam kurtarmayı (günlüğün içeriğini yeniden oynayarak kayıp ya da hasarlı verileri yeniden yaratmak) istiyorsanız, doğrusal günlük kaydını kullanın. Doğrusal günlük kaydı, günlük verilerini sürekli günlük dosyaları sırasında tutar.

Günlük dosyaları isteğe bağlı olarak şu olabilir:

- Yeniden kullanılır, ancak yalnızca yeniden başlatma kurtarma ya da ortam kurtarma için artık gerekli olmadığına.
- Daha uzun süreli depolama ve analiz için el ile arşivlendi.

Ortam görüntülerinin sıklığı, doğrusal günlük dosyalarının ne zaman yeniden kullanılabilirliğini belirler ve doğrusal günlük dosyaları için ne kadar disk alanı olması gerektiğini önemli bir faktördür.

Kuyruk yöneticisini, zamana ya da günlük kullanımına bağlı olarak düzenli ortam görüntülerini otomatik olarak alacak şekilde yapılandırabilir ya da ortam görüntülerini el ile zamanlayabilirsiniz.

Hangi ilkenin uygulanacağına ve disk alanı kullanımı üzerindeki etkilere denetimciniz karar verir. Yeniden başlatma kurtarması için gereken günlük dosyaları her zaman kullanılabilir olmalıdır; ancak, yalnızca ortam kurtarma için gereken günlük dosyaları daha uzun süreli depolamaya (örneğin, manyetik bant) arşivlenebilir.



Sistem yöneticiniz otomatik günlük yönetimi ve otomatik ortam görüntülerini etkinleştirirse, doğrusal günlük kaydı çok büyük bir dairesel günlüğe benzer bir şekilde davranır, ancak ortam kurtarma özelliği ortam hatasına karşı daha fazla yedeklilik sağlar.

IBM MQ 9.1.0' den, `migmqlog` komutunu kullanarak bir kuyruk yöneticisi için var olan bir günlük tipini, doğrusal tipten çevrimsel ya da çevrimsel tipten doğrusal tipine değiştirebilirsiniz.

## Eşlenmiş günlük kaydı

CP4I

Yerel HA yapılandırmasını yapılandırmak için eşlenmiş günlük kaydını kullanın. Yerel HA grubu yarattığınızda, farklı düğümlerde üç kuyruk yöneticisi yaratırsınız. Her kuyruk yöneticisi için benzersiz bir eşgörünüm adıyla birlikte eşlenmiş bir günlük kaydı tipi belirtirsiniz. Yerel HA yapılandırması, etkin bir eşgörünüm, günlük verilerini iki eşleme örneğine eşleyerek yüksek kullanılabilirlikli bir çözüm sağlar. Etkin eşgörünüm başarısız olursa, eşleme eşgörünümlerinden biri etkin rolü devralır. Günlük eşleme, varsa çok az verinin kaybolmasını sağlar. Daha fazla ayrıntı için bkz. [“Yerel HA” sayfa 625](#) . Eşlenmiş bir günlük, otomatik günlük yönetimi ve otomatik ortam görüntülerinin etkinleştirildiği doğrusal bir günlüğe eşdeğerdir.

## Etkin olmayan doğrusal günlük kapsamları

Multi

IBM MQ 9.1.0' den, arşivleme de içinde olmak üzere otomatik günlük yönetimi kullanıyorsanız, kaydedici etkin olmayan doğrusal günlük kapsamlarını izler.



**Uyarı:** Arşivleme olmadan otomatik günlük yönetimi kullanıyorsanız, bu işlem için yedek kuyruk yöneticisinin kullanılması desteklenmez.

ALW

Bir günlük kapsamı artık kurtarma için gerekli olmadığına ve gerekirse arşivlendiğinde, günlük kaydedici uygun bir noktada günlük kapsamını silecek ya da yeniden kullanacaktır.

Yeniden kullanılan bir günlük kapsamı, günlük sırasında bir sonraki olacak şekilde yeniden adlandırılır. AMQ7490 iletisi düzenli aralıklarla yazılır; bu, kaç kapsam yaratıldığını, silindiğini ya da yeniden kullanıldığını gösterir.

Kaydedici, yeniden kullanım için hazır tutması gereken kapsam sayısını ve bu kapsamların ne zaman silineceğini seçer.

## Etkin günlük

Doğrusal ve dairesel günlük kaydında *etkin* olduğu söylenir. Etkin günlük, ister çevrimsel ister doğrusal günlük kaydı kullanıyor olun, yeniden başlatma kurtarması tarafından başvurulabilecek günlük alanı miktarı üst sınırdır.

Etkin günlük kütüklerinin sayısı genellikle, yapılanış kütüklerinde tanımlanan birincil günlük kütüklerinin sayısından azdır. (Sayıyı tanımlamaya ilişkin bilgi için bkz. [“Günlüğün boyutunun hesaplanması” sayfa 633](#) .)

Etkin günlük alanının ortam kurtarma için gerekli alanı içermediğini ve doğrusal günlük kaydı ile kullanılan günlük dosyalarının sayısının, ileti akışınıza ve ortam görüntülerinin sıklığına bağlı olarak çok büyük olabileceğini unutmayın.

## Etkin olmayan günlük

Yeniden başlatma kurtarması için artık bir günlük dosyası gerekmediği zaman, *etkin değıldurumuna* gelir. Yeniden başlatma kurtarması ya da ortam kurtarma için gerekli olmayan günlük dosyaları, gereksiz günlük dosyaları olarak kabul edilebilir.

Otomatik günlük yönetimi kullanılırken, kuyruk yöneticisi bu gereksiz günlük dosyalarının işlenmesini denetler. El ile günlük yönetimini seçtiyseniz, artık işletiminize ilgi duymuyorsa, gereksiz günlük dosyalarını yönetmek (örneğin, silme ve arşivleme) yöneticinizin sorumluluğu olur.

Günlük dosyalarının atılmasına ilişkin ek bilgi için bkz. [“Günlükleri yönetme” sayfa 639](#).

## İkincil günlük dosyaları

İkincil günlük dosyaları doğrusal günlük kaydı için tanımlansa da, bunlar olağan işlemde kullanılmaz. Uzun ömürlü hareketler nedeniyle bir dosyanın yeniden başlatılması gerekebileceğinden, dosyanın etkin havuzdan serbest bırakılmaması durumunda bir durum ortaya çıkarsa, ikincil dosyalar biçimlendirilir ve etkin günlük dosyası havuzuna eklenir.

Kullanılabilir ikincil dosya sayısı kullanılırsa, günlük etkinliği gerektiren diğer işlemlerin çoğuna ilişkin istekler, uygulamaya döndürülen bir MQRC\_RESOURCE\_PROBLEM dönüş koduyla reddedilir ve zamanuyumsuz geriye işleme için uzun süren hareketler dikkate alınır.



**Uyarı:** Tüm günlük kaydı tipleri, donanım arızası olmadığı varsayılarak beklenmedik güç kaybıyla başa çıkabilir.

## Tam kurtarma için denetim gösterimini kullanma

Hem çevrimsel günlük kaydı hem de doğrusal günlük kaydı kuyruk yöneticileri yeniden başlatma kurtarmasını destekler. Kuyruk yöneticisinin önceki örneğinin yeniden başlatıldığında (örneğin bir güç kesintisi) ne kadar aniden sonlandığına bakılmaksızın, kuyruk yöneticisi, sonlandırma noktasında kalıcı durumunu doğru işlem durumuna geri yükler.

Yeniden başlatma kurtarması, disk bütünlüğünün korunmasına bağlıdır. Benzer şekilde, işletim sistemi, işletim sistemi sonlandırmasının ne kadar ani gerçekleşebileceğinden bağımsız olarak, disk bütünlüğünü sağlamalıdır.

Disk bütünlüğünün korunmaması durumunda, doğrusal günlük kaydı (ve ortam kurtarma) daha fazla yedeklilik ve kurtarılabirlik seçeneği sağlar. RAID gibi giderek yaygınlaşan teknolojiyle, disk bütünlüğü sorunlarıyla karşılaşılması giderek daha nadir hale geliyor ve birçok kuruluş döngüsel günlük kaydını yapılandırıyor ve yalnızca yeniden başlatma kurtarmasını kullanıyor.

IBM MQ , klasik bir Önden Yazma Günlük Kaydı kaynak yöneticisi olarak tasarlanmıştır. İleti kuyruklarında kalıcı güncellemeler iki aşamada gerçekleşir:

1. Güncellemeyi gösteren günlük kayıtları, kurtarma günlüğüne güvenilir bir şekilde yazılır
2. Kuyruk dosyası ya da arabellekler, sisteminiz için en verimli olacak şekilde güncellenir, ancak tutarlı olması gerekmez.

Bu nedenle günlük dosyaları, temel kuyruk arabelleği ve dosya durumundan daha güncel olabilir.

Bu durumun devam etmesine izin verildiyse, çökme sonrası kurtarma işleminin ardından kuyruk durumunu tutarlı hale getirmek için çok büyük bir günlük kaydı birimi gerekir.

IBM MQ , çökme sonrası kurtarma işleminin ardından gerekli olan günlük kaydı hacmini sınırlamak için checkpoints komutunu kullanır. Bir günlük dosyasının etkin olarak adlandırılıp adlandırılmadığını denetleyen anahtar olay bir checkpoint.

IBM MQ denetim noktası bir noktadır:

- Kurtarma günlüğü ve nesne dosyaları arasındaki tutarlılık.
- Günlükte, kuyruk yöneticisinin sona erdiği sırada kuyruğun doğru mantıksal durumuna geri yüklenmesi için sonraki günlük kayıtlarının ileriye doğru yeniden oynatılmasının garanti edildiği bir yeri tanımlar.

Bir denetim noktası sırasında IBM MQ , çökme sonrası kurtarma işleminin ardından kuyrukları tutarlı duruma getirmek için yeniden yürütülmesi gereken günlük kayıtlarının hacmini sınırlamak için kuyruk dosyalarında gerektiği şekilde daha eski güncellemeleri temizler.

En son tamamlanan denetim noktası, günlükte, çökme sonrası kurtarma sırasında yeniden yürütme işleminin gerçekleştirilmesi gereken bir noktayı işaretler. Bu nedenle, denetim noktasının sıklığı, kayıt

denetim noktalarının ek yükü ile bu denetim noktalarının belirttiği potansiyel kurtarma sürelerinin iyileştirilmesi arasında bir değiş tokuştur.

Günlüğe kaydedici, etkin günlüğü birincil günlük kapsamlarında tutmayı denediği için IBM MQ 9.1.0' den, günlüğe kaydedici denetim noktalarını daha sık zamanlar (bu nedenle bir sonraki, önceki tamamlanmadan önce zamanlanır). Bu mümkün değilse, bir [AMQ7466](#) hatası günlüğe kaydedilir.

En son tamamlama denetim noktasının başlangıcı günlüğündeki konum, bir günlük dosyasının etkin olup olmadığını belirlemede önemli faktörlerden biridir. Diğer anahtar katsayısı, geçerli etkin bir işlem tarafından yapılan ilk kalıcı güncellemeyle ilgili ilk günlük kaydının günlüğündeki konumdur.

İkinci ya da daha sonraki bir günlük dosyasına yeni bir denetim noktası kaydedilirse ve geçerli bir işlem ilk günlük dosyasındaki bir günlük kaydına başvurmazsa, ilk günlük dosyası devre dışı olur. Döngüsel günlük kaydı durumunda ilk günlük dosyası yeniden kullanılmaya hazırdır. Doğrusal günlük kaydı durumunda ilk günlük dosyası genellikle ortam kurtarma için gerekli olmaya devam eder.

Çevrimsel günlük kaydını ya da otomatik günlük yönetimini yapılandırırsanız, kuyruk yöneticisi etkin olmayan günlük dosyalarını yönetir. Doğrusal günlük kaydını el ile günlük yönetimiyle yapılandırırsanız, işlem gereksinimlerinize göre etkin olmayan dosyaları yönetmek için bir yönetim görevi olur.

IBM MQ , denetim noktalarını otomatik olarak oluşturur. Bunlar aşağıdaki zamanlarda alınır:

- Kuyruk yöneticisi başladığında
- Kapatma sırasında
- Günlük kaydı alanı azaldığında
- **Multi** Önceki denetim noktasının alınmasından bu yana 50.000 işlem günlüğe kaydedildikten sonra
- **z/OS** Önceki denetim noktasının alınmasından bu yana *number\_of\_operations* günlüğe kaydedildikten sonra, burada *number\_of\_operations* **LOGLOAD** özelliğinde ayarlanan işlem sayısıdır.

IBM MQ yeniden başlatıldığında, günlükteki en son denetim noktası kaydını bulur. Bu bilgiler, her denetim noktasının sonunda güncellenen denetim noktası dosyasında tutulur. Denetim noktasından bu yana gerçekleşen tüm işlemler ileriye doğru yürütüyor. Bu tekrarlama aşaması olarak bilinir.

Yeniden yürütme aşaması, kuyrukları sistem arızası ya da kapanmadan önce buldukları mantıksal duruma geri getirir. Yeniden yürütme aşamasında, sistem arızası ya da kapanma oluştuğunda devam eden hareketlerin bir listesi oluşturulur.

**Multi** Yeniden yürütme aşamasının ilerleyişini belirtmek için [AMQ7229](#) ve [AMQ7230](#) iletileri yayınlanır.

Hangi işlemlerin geri çekileceği ya da kesinleştirileceği bilmek için IBM MQ , hareket eden bir işlemle ilişkili her bir etkin günlük kaydına erişir. Bu, kurtarma aşaması olarak bilinir.

**Multi** Kurtarma aşamasının ilerleyişini belirtmek için [AMQ7231](#), [AMQ7232](#) ve [AMQ7234](#) iletileri yayınlanır.

Kurtarma aşamasında gereken tüm günlük kayıtlarına erişildikten sonra, her etkin hareket çözülür ve hareketle ilişkili her işlem geriletilecek ya da kesinleştirilecek. Bu, çözünürlük aşaması olarak bilinir.

**Multi** Çözüm aşamasının ilerleyişini belirtmek için [AMQ7233](#) iletileri yayınlanır.

**z/OS** z/OS işletim sistemi üzerinde, yeniden başlatma işlemleri çeşitli aşamalardan oluşur.

1. Kurtarma günlüğü aralığı, sayfa kümeleri için gerekli ortam kurtarmasına ve iş birimlerinin yedeklenmesi ve belirsiz iş birimlerine ilişkin kilitlerin alınması için gerekli en eski günlük kaydına dayalı olarak oluşturulur.
2. Günlük aralığı belirlendikten sonra, sayfa kümelerini en son duruma getirmek ve aynı zamanda belirsiz ya da devam eden iş birimleriyle ilgili iletileri kilitlemek için iletme günlüğü okuma gerçekleştirilir.

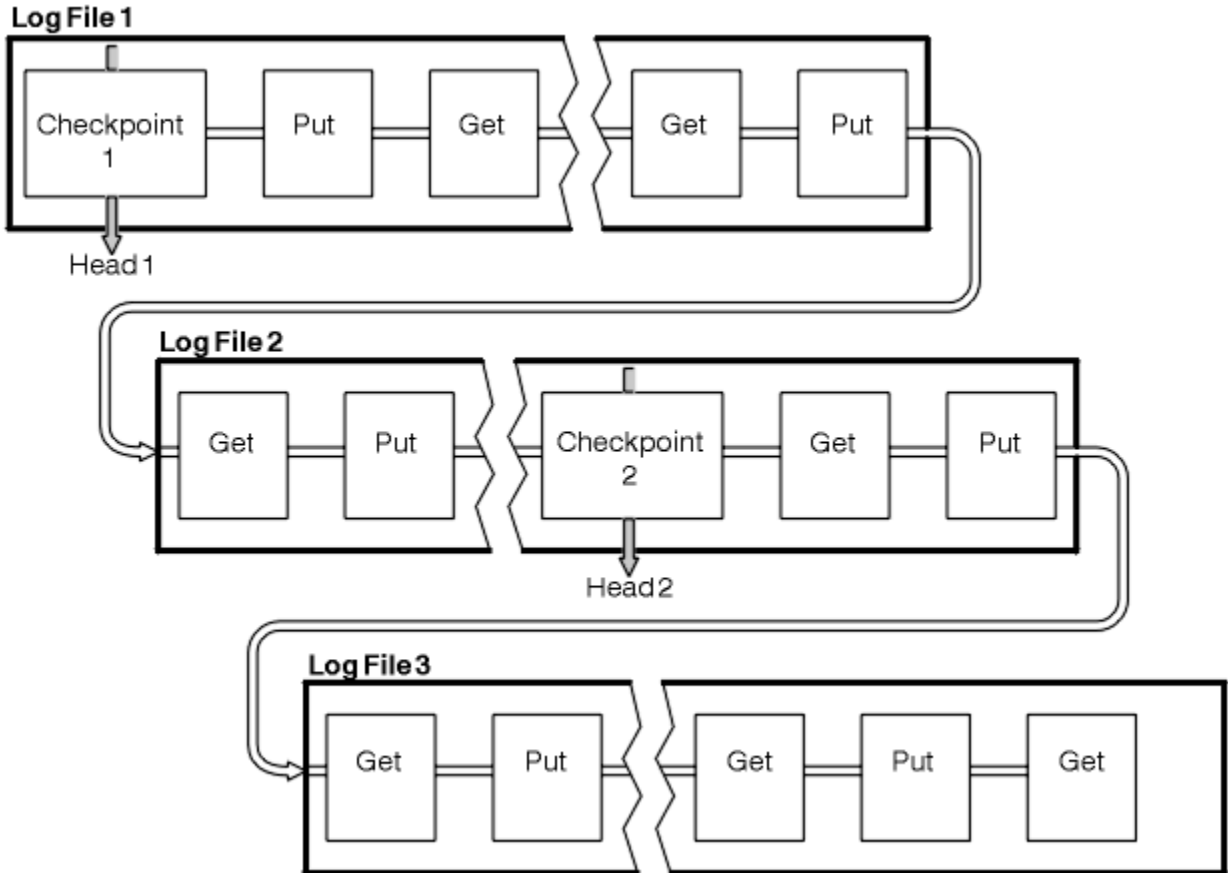
3. İleriye dönük günlük okuma işlemi tamamlandığında, hata sırasında devam eden ya da geriye doğru giden tüm iş birimlerini yedeklemek için günlükler geriye doğru okunur.

**z/OS** Görebileceğiniz iletilere bir örnek:

```
CSQR001I +MQOX RESTART INITIATED
CSQR003I +MQOX RESTART - PRIOR CHECKPOINT RBA=00000001E48C0A5E
CSQR004I +MQOX RESTART - UR COUNTS - 806
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN BACKOUT=0
CSQR030I +MQOX Forward recovery log range 815
from RBA=00000001E45FF7AD to RBA=00000001E48C1882
CSQR005I +MQOX RESTART - FORWARD RECOVERY COMPLETE - 816
IN COMMIT=0, INDOUBT=0
CSQR032I +MQOX Backward recovery log range 817
from RBA=00000001E48C1882 to RBA=00000001E48C1882
CSQR006I +MQOX RESTART - BACKWARD RECOVERY COMPLETE - 818
INFLIGHT=0, IN BACKOUT=0
CSQR002I +MQOX RESTART COMPLETED
```

**Not:** Okunacak çok miktarda günlük varsa, ilerlemeyi göstermek için belirli aralıklarla CSQR031I (ileriye işleyerek kurtarma) ve CSQR033I (geriye doğru kurtarma) iletileri yayınlanır.

Şekil 84 sayfa 632'inde, en son denetim noktası olan Denetim Noktası 2 'den önceki tüm kayıtlar artık IBM MQ tarafından gerekli değildir. Kuyruklar, denetim noktası bilgilerinden ve daha sonraki günlük girişlerinden kurtarılabilir. Çevrimsel günlük kaydı için, denetim noktasından önce serbest bırakılan dosyalar yeniden kullanılabilir. Doğrusal bir günlük için, serbest bırakılan günlük dosyalarına artık normal işlem için erişilmesine ve devre dışı kalmasına gerek yoktur. Örnekte, kuyruk başı işaretçisi en son denetim noktası olan Denetim Noktası 2 'ye taşınır ve daha sonra yeni kuyruk başı, Başlık 2 olur. Günlük Dosyası 1 artık yeniden kullanılabilir.



Şekil 84. Denetim işaretleme

## Uzun süreli hareketlerle işaretleme

Uzun süren bir hareketin günlük dosyalarının yeniden kullanımını nasıl etkilediği.

Şekil 85 sayfa 633 içinde, uzun süreli bir işlemin günlük dosyalarının yeniden kullanımını nasıl etkilediği gösterilmektedir. Örnekte, uzun süreli bir hareket, gösterilen ilk denetim noktasından sonra LR 1 olarak gösterilen günlüğe bir giriş yapmıştır. İşlem, üçüncü denetim noktasından sonraya kadar tamamlanmaz (LR 2 noktasında). LR 1 'den itibaren tüm günlük bilgileri, gerekirse, tamamlanmaya kadar bu işlemin kurtarılmasına izin vermek için korunur.

Uzun süren işlem tamamlandıktan sonra, LR 2 'de, günlüğün başı mantıksal olarak en son günlüğe kaydedilen denetim noktası olan Denetim Noktası 3 'e taşınır. Kontrol noktası 3, Başlık 2 'den önceki günlük kayıtlarını içeren dosyalar artık gerekli değildir. Dairesel günlük kaydı kullanıyorsanız, alan yeniden kullanılabilir.

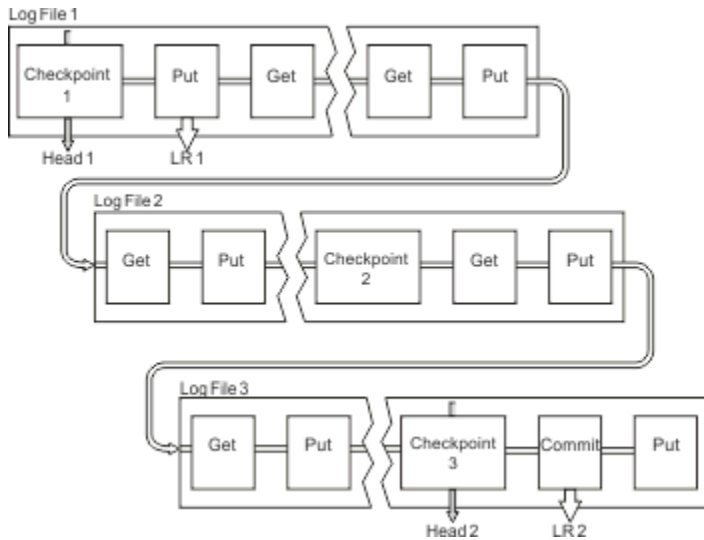
Birincil günlük dosyaları uzun süren işlem tamamlanmadan tamamen doluyorsa, günlüklerin dolmasını önlemek için ikincil günlük dosyaları kullanılabilir.

Tamamen kuyruk yöneticisinin denetimi altında olan etkinlikler (örneğin, denetim işaretleme), birincil günlük içinde etkinliği denemeyi ve saklamayı deneyecek şekilde zamanlanır.

Ancak, kuyruk yöneticisinin denetimi dışındaki davranışları (örneğin, hareketlerinizin birinin süresi) desteklemek için ikincil günlük alanı gerektiğinde, kuyruk yöneticisi, etkinliğin tamamlanmasına izin vermek için tanımlanmış herhangi bir ikincil günlük alanını kullanmayı dener.

Bu etkinlik, toplam günlük alanının %80 'i kullanımda olduğu zamana kadar tamamlanmazsa, kuyruk yöneticisi, bunun uygulama üzerinde bir etkisi olduğundan bağımsız olarak, günlük alanını geri almak için işlem başlatır.

Günlük başı taşındığında ve çevrimsel günlük kaydı kullandığınızda, birincil günlük dosyaları yeniden kullanılabilir duruma gelebilir ve günlük kaydedici, geçerli dosyayı doldurduktan sonra, kullanılabilir ilk birincil dosyayı yeniden kullanır. Doğrusal günlük kaydı kullanıyorsanız, günlük başı etkin havuzda aşağı doğru taşınır ve ilk dosya devre dışı olur. Yeni bir birincil dosya biçimlendirilir ve gelecekteki günlüğe kaydetme etkinliklerine hazır olarak havuzun en altına eklenir.



Şekil 85. Uzun süreli bir işlemle denetim

## Günlüğün boyutunun hesaplanması

Bir kuyruk yöneticisinin gereksinim duyduğu günlük boyutu tahmin ediliyor.

Kuyruk yöneticisinin çevrimsel ya da doğrusal günlük kaydı kullanıp kullanmayacağına karar verdikten sonra, kuyruk yöneticisinin gereksinim duyduğu Etkin günlük 'ün boyutunu tahmin yapmanız gerekir. Etkin günlüğün boyutu, aşağıdaki günlük yapılandırma parametrelerine göre belirlenir:

## LogFileSayfaları

Her bir birincil ve ikincil günlük dosyasının 4K sayfa birimi cinsinden boyutu

## LogPrimaryDosyaları

Önceden ayrılmış birincil günlük dosyalarının sayısı

## LogSecondaryDosyaları

Birincil günlük dosyaları dolduğunda kullanılmak üzere oluşturulabilecek ikincil günlük dosyalarının sayısı

## Notlar:

1. Kuyruk yöneticisinin her başlatılışında birincil ve ikincil günlük dosyalarının sayısını değiştirebilirsiniz, ancak ikincil günlüklerde yapacağınız değişikliğin etkisini hemen fark etmeyebilirsiniz.
2. Günlük dosyası boyutunu değiştiremezsiniz; kuyruk yöneticisini yaratmadan **önce** belirlemeniz gerekir.
3. Birincil günlük dosyalarının sayısı ve günlük dosyası büyüklüğü, kuyruk yöneticisi yaratıldığında önceden ayrılan günlük alanı miktarını belirler.
4. AIX and Linux sistemlerinde birincil ve ikincil günlük kütüklerinin toplam sayısı 511 'i ya da Windowssistemlerinde 255 'i geçemez; bu sayı, uzun süren hareketlerin varlığında, yeniden başlatma kurtarması için kuyruk yöneticisinin kullanabileceği günlük yeri üst sınırını belirler. Ortam kurtarma işlemi için kuyruk yöneticisinin gerek duyabileceği günlük alanı miktarı bu sınırı paylaşmıyor.
5. *çevrimsel* günlük kaydı kullanıldığında, kuyruk yöneticisi birincil ve ikincil günlük alanını yeniden kullanır. Kuyruk yöneticisi, bir günlük dosyası dolduğunda ve sıradaki bir sonraki birincil günlük dosyası kullanılmadığında, bir sınıra kadar ikincil bir günlük dosyası ayırır.

Ayırmanız gereken günlük sayısına ilişkin bilgi için bkz. "[Aktif günlüğümü ne kadar büyük yapmalıyım?](#)" sayfa 634 . Birincil günlük kapsamları sırayla kullanılır ve bu sıra değişmez.

Örneğin, üç birincil günlüğünüz 0, 1ve 2varsa, kullanım sırası 0,1,2 ve ardından 1,2,0, 2,0,1, geri 0,1,2 vb. gelir. Ayırdığınız ikincil günlükler gerektiği şekilde serpiştirilir.

6. Birincil günlük dosyaları, bir denetim noktası sırasında yeniden kullanılabilir duruma getirilebilir. Günlük alanı azaldığı için, kuyruk yöneticisi bir denetim noktası almadan önce hem birincil hem de ikincil günlük alanını dikkate alır.

Kuyruk yöneticisi, denetim noktalarını, günlük kullanımını birincil kapsamlarda tutacak şekilde zamanlamayı dener.

Ek bilgi için bkz. "[mq5.ini dosyasının LogDefaults kısmı](#)" sayfa 92 .

## **Aktif günlüğümü ne kadar büyük yapmalıyım?**

Bir kuyruk yöneticisinin gereksinim duyduğu etkin günlük boyutunu tahmin etme.

Etkin günlüğün büyüklüğü şunlarla sınırlıdır:

```
logsize = (primaryfiles + secondaryfiles) * logfilepages * 4096
```

Kuyruk yöneticisi diske saniye başına veri miktarı üst sınırını yazarken, günlük, çalışmakta olan en uzun hareketle başa çıkabilecek büyüklükte olmalıdır.

En uzun süre çalışan hareketiniz N saniye boyunca çalışıyorsa ve kuyruk yöneticisi tarafından diske yazılan veri miktarı üst sınırı günlükte saniyede B baytsa, günlüğünüz en az:

```
logsize >= 2 * (N+1) * B
```

Kuyruk yöneticisi, iş yükünün doruk noktasında çalışırken ya da ortam görüntülerini kaydederken, saniyede veri miktarı üst sınırını diske yazabilir.

Bir işlem, ilk günlük kaydını içeren günlük kapsamı etkin günlüğün içinde bulunmayacak kadar uzun süre çalışırsa, kuyruk yöneticisi en eski günlük kaydıyla başlayarak etkin hareketleri birer birer geri alır.

Kuyruk yöneticisinin, birincil ve ikincil dosya sayısı üst sınırı kullanılmadan önce eski günlük kapsamlarını devre dışı yapması ve kuyruk yöneticisinin başka bir günlük kapsamı ayırması gerekir.

Kuyruk yöneticisinin işlemleri geriye işlemlerine izin verilmeden önce, en uzun süren hareketin ne kadar süreyle çalışmasını istediğinize karar verin. En uzun süre çalışan hareketiniz, yavaş ağ trafiğini bekliyor olabilir ya da kötü tasarlanmış bir işlem olması durumunda, kullanıcı girişi bekliyor olabilir.

Aşağıdaki **runmqsc** komutunu vererek, en uzun süre çalışan hareketinizin ne kadar süreyle çalıştığını araştırabilirsiniz:

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

dspmqtın -a komutu verilirken, tüm durumlardaki XA ve XA olmayan komutların tümü gösterilir.

Bu komutun verilmesi, yürürlükteki tüm hareketleriniz için ilk günlük kaydının yazıldığı tarih ve saati listeler.



**Uyarı:** Günlük boyutunu hesaplamak için, uygulamanın ya da işlemin başlamasından bu yana geçen süre değil, ilk günlük kaydının yazılmasından bu yana geçen zamandır. En uzun çalışan hareketin uzunluğunu en yakın saniyeye yuvarlar. Bunun nedeni, kuyruk yöneticisindeki eniyilemeler.

Uygulama, örneğin, bir iletiyi gerçekten almadan önce bir süre bekleyen bir MQGET çağrısıyla başlıyorsa, ilk günlük kaydı uygulama başlatıldıktan uzun süre sonra yazılabilir.

Gözlenen tarih ve saat çıkışı üst sınırını gözden geçirerek

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

Başlangıçta, geçerli tarih ve saatten itibaren, en uzun süren hareketin ne kadar süreyle çalışacağını tahmin edebilirsiniz.

En uzun süre çalışan hareketleriniz en yüksek iş yükünde çalışırken bu **runmqsc** komutunu tekrar tekrar çalıştırdığınızdan emin olun; böylece en uzun süren hareketin uzunluğunu hafife almayın.

IBM MQ 8.0 içinde işletim sistemi araçlarını kullanın; örneğin, UNIX platformlarında **iostat**.

IBM MQ 9.0'den, aşağıdaki komutu vererek kuyruk yöneticisinin günlüğe yazmakta olduğu bayt/saniye baytlarını keşfedebilirsiniz:

```
amqsrua -m qmgr -c DISK -t Log
```

Yazılan mantıksal baytlar, kuyruk yöneticisinin günlüğe yazmakta olduğu saniyedeki baytları gösterir. Örneğin:

```
$ amqsrua -m mark -c DISK -t Log
Publication received PutDate:20160920 PutTime:15383157 Interval:4 minutes,39.579 seconds
Log - bytes in use 37748736
Log - bytes max 50331648
Log file system - bytes in use 316243968
Log file system - bytes max 5368709120
Log - physical bytes written 4334030848 15501948/sec
Log - logical bytes written 3567624710 12760669/sec
Log - write latency 411 uSec
```

Bu örnekte, günlüğe yazılan saniye başına mantıksal bayt sayısı 12760669/sec ya da yaklaşık 12 MiB / saniyedir.

Kullanılan

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

en uzun süre çalışan işlemin şu şekilde olduğunu gösterdi:

```
CONN(57E14F6820700069)
EXTCONN(414D51436D61726B2020202020202020)
TYPE(CONN)
APPLTAG(msginteg_r) UOWLOGDA(2016-09-20)
UOWLOGTI(16.44.14)
```



Güncel tarih ve saat 2016-09-20 16.44.19 olduğu için bu işlem 5 saniyedir devam ediyordu. Ancak, kuyruk yöneticisi bunları geri almadan önce 10 saniye boyunca çalışan hareketlere izin vermeniz gerekir. Günlük boyutunuz şu olmalıdır:

$$2 * (10 + 1) * 12 = 264 \text{ MiB}$$

Günlük dosyalarının sayısı, beklenen en büyük günlük boyutunu (önceki metinde hesaplanır) içerebilmelidir. Bu, aşağıda belirtilenler olacaktır:

Günlük kütüğü sayısı alt sınırı = (Gerekli günlük büyüklüğü)/(**LogFilePages** \* günlük kütüğü sayfa büyüklüğü (4096))

4096 olan varsayılan **LogFilePages** ve önceki metinde hesaplanan 264MiB günlük boyutu tahmini kullanılarak, günlük dosyası sayısı alt sınırı şu olmalıdır:

$$264\text{MiB} / (4096 \times 4096) = 16.5$$

Yani, 17 günlük dosyası.

Günlüğünüzü, beklenen iş yükünün birincil dosyalar içinde çalışacağı şekilde boyutlandırırsanız:

- İkincil dosyalar, ek günlük alanı gerektiğinde bazı beklenmedik durumlar sağlar.
- Döngüsel günlük kaydı her zaman, ikincil dosyaları ayırmaktan ve ayırmaktan çok daha hızlı olan önceden ayrılmış birincil dosyaları kullanır.
- Kuyruk yöneticisi, bir sonraki denetim noktasının ne zaman alınacağını hesaplamak için yalnızca birincil dosyalarda kalan alanı kullanır.

Bu nedenle, önceki örnekte, iş yükünün birincil günlük dosyaları içinde çalışması için aşağıdaki değerleri ayarlayın:

- **LogFilePages** = 4096
- **LogPrimaryFiles** = 17
- **LogSecondaryFiles** = 5

Aşağıdakileri unutmayın:

- Bu örnekte 5 ikincilleri, etkin günlük alanının yüzde 20 'sinden fazlasıdır.

IBM MQ 9.1.0' den kaydedici, iş yükünü birincil dosyalarda tek başına tutmayı dener. Bu nedenle, günlük kaydedici, birincil dosyaların bir bölümü tek başına dolu olduğunda denetim noktalarını zamanlar.

İkincil dosyaların olması, beklenmedik bir şekilde uzun süren bir işlem olması durumunda olası bir durumdur.

Toplam günlük alanının yüzde 80 'inden fazlası kullanımdayken, kuyruk yöneticisinin günlük alanı kullanımını azaltmak için işlem gerçekleştirdiğini unutmayın.

- Doğrusal ya da dairesel günlük kaydı kullandığınızdan bağımsız olarak aynı hesaplamayı gerçekleştirin.

Aktif günlük kavramı hem doğrusal günlüğe kaydetme hem de dairesel günlük kaydında aynı anlama geldiğinden, doğrusal ya da dairesel bir aktif günlüğün boyutunu hesaplayıp hesaplamadığınız fark etmez.

- Yalnızca ortam kurtarma için gereken günlük kapsamları etkin günlük içinde değildir ve bu nedenle birincil ve ikincil dosyaların sayısında sayılmaz.
- IBM MQ 9.1.0 **DISPLAY QMSTATUS LOG** ' un **LOGUTIL** alanından, yaklaşık olarak gerekli etkin günlüğün boyutunu hesaplamaya yardımcı olmak için kullanılabilir.

Bu alan, en uzun süren hareketlerinizin süresini ya da kuyruk yöneticisinin en yüksek verim oranını belirlemek için sürekli örnekleme olmadan gerekli günlük boyutunu makul bir şekilde tahmin etmeniz için tasarlanmıştır.



## LogFileSayfalarımı ne kadar büyük yapmalıyım?

Genellikle LogFileSayfalarınızı, birincil dosya sayısı üst sınırına ulaşmadan etkin günlüğünüz boyutunu kolayca artırabilecek kadar büyük yapın. Birkaç büyük günlük dosyası, birkaç büyük günlük dosyası, gerek duyarsanız, günlük boyutunu artırmanız için daha fazla esneklik sağladığından, birçok küçük günlük dosyasına tercih edilir.

Doğrusal günlük kaydı için, çok büyük günlük dosyaları performans değişkenini yapabilir. Çok büyük günlük dosyalarıyla yeni bir günlük dosyası oluşturmak ve biçimlendirmek ya da eski bir günlük dosyasını arşivlemek için daha büyük bir adım vardır. Otomatik günlük yönetimi ile yeni günlük dosyaları nadiren oluşturulduğundan, bu daha çok el ile ve arşiv günlüğü yönetimiyle ilgili bir sorundur.

### **Günlüğümü çok küçük yaparsam ne olur?**

Günlüğün minimum boyutunu tahmin ederken göz önünde bulundurmanız gereken noktalar.

Günlüğünüzü çok küçük yaparsanız:

- Uzun süreli hareketler geri çekilecek.
- Bir sonraki denetim noktası, önceki denetim noktası sona ermeden önce başlamak istiyor.

**Önemli:** Günlüğün boyutunu ne kadar yanlış tahmin etseniz de, veri bütünlüğü korunur.

Denetim noktalarına ilişkin açıklamalar için bkz. [“Tam kurtarma için denetim gösterimini kullanma” sayfa 630](#) . Etkin günlük kapsamlarında kalan günlük alanı miktarı kısa olursa, kuyruk yöneticisi denetim noktalarını daha sık zamanlar.

Bir kontrol noktası biraz zaman alır; anlık değildir. Denetim noktasına kaydedilmesi gereken veriler ne kadar çok olursa, denetim noktası o kadar uzun sürer. Günlük küçük denetim noktaları çakışabiliyorsa, önceki denetim noktası sona ermeden önce bir sonraki denetim noktası istenir. Bu durumda hata iletileri yazılır.

Uzun süre çalışan hareketler geri çekilirse ya da denetim noktaları çakıştıysa, kuyruk yöneticisi iş yükünü işlemeye devam eder. Kısa ömürlü işlemler olağan şekilde çalışmaya devam eder.

Ancak, kuyruk yöneticisi en iyi şekilde çalışmıyor ve başarım düşebilir. Kuyruk yöneticisini yeterli günlük alanıyla yeniden başlatmanız gerekir.

### **Günlüğümü çok büyük yaparsam ne olur?**

Günlüğün maksimum boyutunu tahmin ederken göz önünde bulundurmanız gereken noktalar.

Günlüğünüzü çok büyük yaparsanız:

- Acil durumda yeniden başlatma için geçen süreyi artırabilirsiniz, ancak bu pek mümkün değildir.
- Gereksiz disk alanı kullanıyorsunuz.
- Çok uzun süredir devam eden işlemler tolere edilir.

**Önemli:** Günlüğün boyutunu ne kadar yanlış tahmin etseniz de, veri bütünlüğü korunur.

Günlüğün maksimum boyutunu tahmin etmenize yardımcı olmak için günlük kullanım istatistiklerini kullanabilirsiniz. Ek bilgi için bkz. [“IMGLOGLN ve IMGINTVL ayarlarının nasıl yapılacağına karar verilmesi” sayfa 643 ve ALTER QMGR](#).

Kuyruk yöneticisinin yeniden başlatıldığında günlüğü nasıl okuyacağına ilişkin açıklamalar için bkz. [“Tam kurtarma için denetim gösterimini kullanma” sayfa 630](#) . Kuyruk yöneticisi günlüğü son denetim noktasından yeniden kaydeder ve kuyruk yöneticisi sona erdiğinde etkin olan tüm hareketleri çözer.

Bir hareketi çözmek için, kuyruk yöneticisi o işlemlerle ilişkili tüm günlük kayıtlarını geri okur. Bu günlük kayıtları, son denetim noktasının tarihinden önce olabilir.

Kuyruk yöneticisine çok büyük bir günlük ayırarak, kuyruk yöneticisine yeniden başlatma sırasında her günlük kaydını okuma izni verirsiniz; ancak, genellikle kuyruk yöneticisinin bunu yapması gerekmez. Potansiyel olarak, bunun gerçekleşmesi olası olmayan bir durumda, bu işlem uzun sürebilir.

Denetim, kuyruk yöneticisi sona ermeden önce beklenmedik bir şekilde durdurulduysa, bu, büyük bir günlüğü olan bir kuyruk yöneticisinin yeniden başlatma süresini önemli ölçüde artırır. Günlüğün boyutunun sınırlanması, acil durum yeniden başlatma süresini sınırlar.

Bu sorunları önlemek için aşağıdakilerden emin olmanız gerekir:

- İş yükünüz, aşırı büyük olmayan bir günlüğe rahatça sığabilir.
- Uzun süreli işlemlerden kaçınıyorsunuz.

### **Günlük dosya sistemini ne kadar büyük yapmalıyım?**

Bir kuyruk yöneticisinin gereksinim duyduğu günlük dosya sistemi boyutu tahmin ediliyor.

Günlük dosya sistemini yeterince büyük hale getirmeniz önemlidir, böylece kuyruk yöneticiniz günlüğünü yazmak için yeterli alana sahip olur. Kuyruk yöneticisi günlük dosya sistemini tamamen doldurursa, FFDC 'leri yazar, işlemleri geriye işler ve kuyruk yöneticisini aniden sonlandırabilir.

Günlüğünüz için ayırdığınız disk alanı en az etkin günlük kadar büyük olmalıdır. Tam olarak ne kadar büyük olduğuna bağlı:

- Günlük tipi seçiminiz (doğrusal ya da dairesel)
- Etkin günlüğün büyüklüğü (birincil dosyalar, ikincil dosyalar, günlük dosyası sayfaları)
- Günlük yönetimi seçiminiz (el ile, otomatik ya da arşiv)
- Hasarlı bir nesne durumunda acil durum planlarınız.

Çevrimsel bir günlük seçerseniz, günlük dosya sistemi

```
LogFilesystemSize >= (PrimaryFiles + SecondaryFiles + 1) * LogFileSize
```

Bu, kuyruk yöneticisinin tüm birincil ve ikincil dosyalara yazmasını sağlar. Kural dışı durumlarda, kuyruk yöneticisi ikincil sınırların ötesinde ek bir kapsam yazabilir. Önceki algoritma bunu dikkate alır.

Doğrusal bir günlük seçerseniz, günlük dosya sistemi etkin günlükten önemli ölçüde daha büyük olmalıdır.

El ile günlük yönetimini seçerseniz, kuyruk yöneticisi gerektiğinde yeni günlük kapsamlarına yazmaya devam eder ve bunları silmek (ve arşivlemek) sizin sorumluluğunuzdadır.

Günlük dosya sisteminin ne kadar büyük olması gerektiği büyük ölçüde gereksiz veya etkin olmayan kapsamları silme stratejinize bağlıdır.

Kapsamları etkinlik dışı duruma gelir gelmez (yeniden başlatma kurtarması için gerekli değil) arşivleyip silmeye karar verebilir ya da yalnızca gereksiz kapsamları arşivleyip silmeye karar verebilirsiniz (ortam ya da kurtarma işlemini yeniden başlatmak için gerekli değildir).

Yalnızca gereksiz kapsamları arşivleyip siliyorsanız ve zarar görmüş bir nesneniz varsa, **MEDIALOG** ileri gitmez, bu nedenle artık kapsamlar gereksiz hale gelmeyecektir. Sorunu çözene kadar, belki de nesneyi kurtararak kapsamları arşivlemeyi ve silmeyi durdurursunuzdur.

İş yükünü durdurmazsanız, sorunu çözmek için ne kadar zamanınız olması gereken günlük dosya sisteminin boyutuna bağlıdır. Bu nedenle, doğrusal günlük kaydı kullanılırken cömert bir günlük dosya sistemine sahip olmak en iyi uygulamadır.

Doğrusal bir günlük ve otomatik ya da arşiv günlüğü yönetimi seçerseniz, kuyruk yöneticisi günlük kapsamlarını yeniden kullanır.

Yeniden kullanılabilecek günlük kapsamlarına Rharfi eklenir. Gereksiz kapsamlar arşivlendiği için bir ortam görüntüsü kaydedildiğinde, kuyruk yöneticisi bu kapsamları yeniden kullanabilir.

Bu nedenle, yeniden kullanım kapsamları, ortam görüntüleri arasında günlüğe yazılan veri uzunluğundan daha az:

```
ReuseExtents <= LogDataLengthBetweenMediaImages
```

Ortam görüntülerini otomatik olarak kaydederken ve **IMGLOGLN**ayarlararken, **IMGLOGLN** sabit bir üst sınır olmadığı için LogDataLengthBetweenMediaImages en fazla iki kez **IMGLOGLN** olabilir.

Ortam görüntülerini el ile kaydederken ya da aralığa göre otomatik olarak kaydederken LogDataLengthBetweenMediaImages , iş yükünüze ve görüntü alma aralığına bağlıdır.

Etkin kapsamlara ve yeniden kullanım kapsamlarına ek olarak, etkin olmayan kapsamlar (yalnızca ortam kurtarma için gereklidir) ve gereksiz kapsamlar (yeniden başlatma ya da ortam kurtarma için gerekli değildir) vardır.

Otomatik ya da arşiv günlüğü yönetimi kullanılırken, kuyruk yöneticisi ortam kurtarma için gereken kapsamları yeniden kullanmaz. Bu nedenle, etkin olmayan kapsamların sayısı, ortam görüntülerini ne sıklıkta aldığınıza ve bunları el ile mi, yoksa otomatik olarak mı aldığınıza bağlıdır.

**IMGINTVL** ve **IMGLOGLN** , ortam görüntüleri arasında sabit bir alt sınır ya da üst sınır değil, hedeflerdir. Ancak, gereksinim duyabileceğiniz günlük kütük sistemi büyüklüğü üst sınırını tahmin ederken, otomatik ortam görüntülerinin ikiden fazla **IMGINTVL** ya da **IMGLOGLN** ayrı olarak kaydedilmesi olası değildir.

Günlük dosya sistemini otomatik ya da arşiv günlük yönetimi kullanarak boyutlandırırken, bir kuyruk ya da başka bir nesne zarar görürse neler olabileceğini de göz önünde bulundurmanız gerekir. Bu durumda, kuyruk yöneticisi zarar gören nesnenin ortam görüntüsünü alamaz ve **MEDIALOG** ileri gitmez.

İş yükünüz devam ederse, ortam kurtarma için gereken en eski kapsam hala gerekli olduğundan ve yeniden kullanılmadığından, etkin olmayan günlüğünüz engellenmeden büyür. İş yükünüz devam ederse, kuyruk yöneticisi işlemleri geriye işleme başlatmadan ve hatta aniden sona ermeden önce, günlük dosya sistemi sorunu düzeltmek için tamamen doluncaya kadar iş yükünüz devam eder.

Bu nedenle otomatik ve arşiv günlüğü yönetimi için:

```
LogFilesystemSize > (PrimaryFiles + SecondaryFiles +  
((TimeBetweenMediaImages *2) + TimeNeededToResolveDamagedObject) * ExtentsUsedPerHour))  
* LogFilePages
```

**Not:** Önceki algoritma, arşiv günlüğü yönetimi için ortam kurtarma için artık gerekli olmadığı anda her kapsam için **SET LOG ARCHIVED** ürününün çağrıldığı varsayılır.

## Günlükleri yönetme

Ürün, IBM MQ 9.1.0' den doğrusal günlüklerin otomatik günlük yönetimini ve otomatik ortam kurtarmasını destekler. Dairesel günlükler neredeyse kendi kendini yönetir, ancak bazen alan sorunlarını çözmek için müdahale gerekir.

**Not:** **IBM i** Otomatik ve arşiv günlüğü yönetimi IBM üzerinde geçerli değildir.

Çevrimsel günlük kaydında, kuyruk yöneticisi günlük dosyalarında boş yer açar. Bu etkinlik kullanıcı için belirgin değildir ve ayrılan alan hızlı bir şekilde yeniden kullanıldığından, genellikle kullanılan disk alanı miktarının azaldığını görmezsiniz.

IBM MQ 9.1.0 ' dan döngüsel günlük kaydı kullanırken ikincil dosyaları silebilirsiniz. Daha fazla bilgi için bkz. **RESET QMGR TYPE (REDUCELOG)** .

Doğrusal günlük kaydında, bir denetim noktası uzun süre alınmamışsa ya da uzun süre çalışan bir işlem uzun zaman önce bir günlük kaydı yazdıysa günlük doldurabilir. Kuyruk yöneticisi, ilk sorunu önlemek için denetim noktalarını yeterince sık almayı dener.

**Multi** Günlük doluysa, AMQ7463 iletisi yayınlanır. Ayrıca, uzun süreli bir hareket alanın serbest bırakılmasını engellediği için günlük doluysa, AMQ7465 iletisi yayınlanır.

Kuyruk yöneticisini yeniden başlatmak için, günlük kayıtlarından yalnızca son tam denetim noktasının başlangıcından bu yana yazılan kayıtlar ve etkin hareketler tarafından yazılan kayıtlar gerekir.

Zaman içinde, yazılan en eski günlük kayıtları kuyruk yöneticisini yeniden başlatmak için gereksiz hale gelir.

Uzun süreli bir hareket algılandığında, etkinlik o hareketi zamanuyumsuz olarak geri almak üzere zamanlanır. Beklenmeyen bir nedenle zamanuyumsuz geriye işleme başarısız olursa, bazı MQI çağrıları bu durumda MQRC\_RESOURCE\_PROBLEM döndürür.

Tüm uçuştaki hareketleri kesinleştirmek ya da geriye işlemek için yer ayrıldığı için **MQCMIT** ya da **MQBACK** başarısız olmamalıdır.

Bu şekilde geriye işlenen bir hareketi olan bir uygulama, aynı hareket altında eşitleme noktası belirten sonraki **MQPUT** ya da **MQGET** işlemlerini gerçekleştiremez.

Bu durumda bir iletiyi eşitleme noktası altına koyma ya da alma girişimi **MQRC\_BACKED\_OUT** değerini döndürür. Daha sonra uygulama, **MQRC\_BACKED\_OUT** ya da **MQBACK** değerini döndüren **MQCMIT** komutunu verebilir ve yeni bir hareket başlatabilir. Çok fazla günlük alanı tüketen hareket geriye işlendiğinde, günlük alanı serbest bırakılır ve kuyruk yöneticisi olağan şekilde çalışmaya devam eder.

### ***Bir disk dolunca ne olur?***

Bir kuyruk yöneticisi doğrusal günlük kaydı kullanacak şekilde yapılandırıldığında, kuyruk yöneticisi günlük kaydı bileşeni disk dolu koşuluna aşağıdaki şekillerde tepki verir.

Günlük dosyalarını içeren disk dolu olursa, aşağıdaki işlemleri yapın:

- Kuyruk yöneticisi bu koşulu yalnızca, gerekli büyüklükte yeni bir günlük dosyası yaratırken keşfeder; bu, gerektiğinde önceden de bunu yapar.
- İşletim sistemi, dosyayı gerekli boyuta genişletmek için istekten bir hata döndürdüğünde disk dolu durumunu keşfeder.
- Kuyruk yöneticisi, kuyruk yöneticisi hata günlüğüne AMQ6708 iletisini gönderir.
- Sistem genelinde hatalar dizinine First Failure Support Technology (FFST) kaydı yazılır. Bu kayıt, disk dolu koşulunun ayrıntılarını sağlar ve IBM Destek ile iletişim kurmanız gerekirse saklanmalıdır.

Günlük dosyaları, günlük kayıtları onlara yazılırken genişletilmek yerine, sabit boyutlarında oluşturulur. Bu, IBM MQ ' in yalnızca yeni bir dosya yaratırken disk alanı tükenebilir; günlüğe bir kayıt yazarken yer kalmaz. IBM MQ , var olan günlük dosyalarında ne kadar yer olduğunu her zaman bilir ve dosyalar içindeki alanı buna göre yönetir.

IBM MQ 9.1.0' den doğrusal günlük kaydını kullandığınızda, aşağıdakileri kullanma seçeneğiniz vardır:

- Günlük kapsamlarını otomatik olarak yönetme.

Yeni günlük özniteliklerine ilişkin ek bilgi için [DISPLAY QMSTATUS](#) başlıklı konuya bakın.

Ayrıca, aşağıdaki komutlara ya da bunların PCF eşdeğerlerine bakın:

- [QMGR ' YI SIFIRLAYIN](#)
- Dağıtılmış altyapılar için [SET LOG](#)

- Ortam görüntülerinin kullanımını denetleyen seçenekler.

Aşağıdakilere ilişkin ek bilgi için [ALTER QMGR](#) komutuna ve [ALTER KUYRUK](#) komutuna bakın:

- [IMGINTVL](#)
- [IMGLOGLN](#)
- [IMGRCOVO](#).
- [IMGRCOVQ](#).
- [GÖRÜNTÜLENMİŞ](#)

Döngüsel günlük kaydı bir kaynak sorunu döndürür.

Yeriniz tükenmeye devam ederse, kuyruk yöneticisi yapılanış kütüğündeki günlük yapılanışının doğru olup olmadığını denetleyin. Günlüğün kullanılabilir alanı aşmaması için birincil ya da ikincil günlük dosyalarının sayısını azaltabilirsiniz.

Var olan bir kuyruk yöneticisi için günlük dosyalarının boyutunu değiştiremezsiniz. Kuyruk yöneticisi, tüm günlük kapsamlarının aynı boyutta olmasını gerektirir.

### ***Günlük dosyalarını yönetme***

Günlük dosyalarınız için yeterli alan ayırın. Doğrusal günlük kaydı için, eski günlük dosyalarını artık gerekli olmadığına silebilirsiniz.

## Döngüsel günlüğe kaydetmeye özgü bilgiler

Döngüsel günlük kaydı kullanıyorsanız, sisteminizi yapılandırırken günlük dosyalarını tutmak için yeterli alan olduğundan emin olun (bkz. “mqsc.ini dosyasının LogDefaults kısmı” sayfa 92 ve “qm.ini dosyasının günlük kısmı” sayfa 129). Günlük tarafından kullanılan disk alanı miktarı, gerektiğinde yaratılacak ikincil dosyalar için alan da içinde olmak üzere, konfigürasyonu tanımlanmış büyüklüğü aşmaz.

## Doğrusal günlüğe kaydetmeye özgü bilgiler

Doğrusal bir günlük kullanıyorsanız, veriler günlüğe kaydedildikçe günlük dosyaları sürekli olarak eklenir ve kullanılan disk alanı miktarı zamanla artar. Günlüğe kaydedilmekte olan veri hızı yüksekse, disk alanı yeni günlük dosyaları tarafından hızla kullanılır.

Zamanla, doğrusal bir günlüğe ilişkin eski günlük dosyalarının kuyruk yöneticisini yeniden başlatması ya da zarar görmüş nesnelere için ortam kurtarma işlemi gerçekleştirmesi gerekmez. Aşağıdaki yöntemler hangi günlük dosyalarının gerekli olduğunu belirler:

### Günlüğe kaydedici olay iletileri

Önemli bir olay ortaya çıktığında, örneğin bir kayıt ortamı görüntüsü, kaydedici olay iletileri oluşturulur. Günlüğe kaydedici olay iletilerinin içeriği, kuyruk yöneticisinin yeniden başlatılması ve ortam kurtarması için hala gerekli olan günlük dosyalarını belirtir. Günlüğe kaydedici olay iletileriyle ilgili daha fazla bilgi için bkz. [Günlüğe kaydedici olayları](#)

### Kuyruk yöneticisi durumu

MQSC komutu, DISPLAY QMSTATUS ya da PCF komutu Inquire Queue Manager Status çalıştırıldığında, gerekli günlük dosyalarının ayrıntıları da içinde olmak üzere kuyruk yöneticisi bilgileri döndürülür. MQSC komutlarıyla ilgili ek bilgi için [Administering IBM MQ using MQSC commands](#)(MQSC komutlarını kullanarak yönetme) konusuna bakın ve PCF komutlarıyla ilgili bilgi için [Automating administration](#) tasksbaşlıklı konuya bakın.

### Kuyruk yöneticisi iletileri

Belirli aralıklarla, kuyruk yöneticisi günlük dosyalarından hangisinin gerekli olduğunu belirtmek için bir çift ileti yayınlar:

- AMQ7467I iletili, kuyruk yöneticisini yeniden başlatmak için gereken en eski günlük dosyasının adını verir. Kuyruk yöneticisi yeniden başlatılırken bu günlük dosyası ve daha yeni tüm günlük dosyaları kullanılabilir olmalıdır.
- AMQ7468I iletili, ortam kurtarma için gereken en eski günlük dosyasının adını verir.

"Daha eski" ve "daha yeni" günlük dosyalarını belirlemek için, dosya sistemi tarafından uygulanan değişiklik süreleri yerine günlük dosyası numarasını kullanın.

## Her iki günlük kaydı tipi için geçerli bilgiler

Yalnızca kuyruk yöneticisinin yeniden başlatılması için gereken günlük dosyalarının, etkin günlük dosyalarının çevrimiçi olması gerekir. Etkin olmayan günlük dosyaları, olağanüstü durumdan kurtarma için manyetik bant gibi bir arşiv ortamına kopyalanabilir ve günlük dizininden kaldırılabilir. Ortam kurtarma için gerekli olmayan etkin olmayan günlük dosyaları, gereksiz günlük dosyaları olarak kabul edilebilir. İşlemin artık ilgilenmediği gereksiz günlük dosyalarını silebilirsiniz.

Gereken herhangi bir günlük dosyası bulunamazsa, AMQ6767E işletmen iletili yayınlanır. Günlük dosyasını ve sonraki tüm günlük dosyalarını kuyruk yöneticisinin kullanımına açın ve işlemi yeniden deneyin.

## Günlük kapsamalarını otomatik olarak temizleme-yalnızca doğrusal günlük kaydı

Multi

IBM MQ 9.1.0 ' den, kurtarma için artık gerekli olmayan doğrusal günlük kapsamaları otomatik yönetimini kullanma seçeneğiniz vardır.

Otomatik yönetimi ayarlamak için qm.ini dosyasının Log (Günlük) kısmına **LogManagement** özneliğini ya da IBM MQ Explorer komutunu kullanarak kullanabilirsiniz. Ek bilgi için bkz. "[qm.ini dosyasının günlük kısmı](#)" sayfa 129 .

Günlüğün işlemeyle ilgili ayrıntılar için **DISPLAY QMSTATUS** içindeki LOG değiştirgesine ve günlükün kullanılmasına ilişkin aşağıdaki komutlara bakın:

- [QMGR ' YI SIFIRLAYIN](#)
- [Günlüğü AYARLA](#)

## Ortam resimlerini otomatik olarak alma-yalnızca doğrusal günlük kaydı

IBM MQ 9.1.0 ' den, kuyruk yöneticisinin ortam görüntülerini otomatik olarak yazıp yazmadığını denetleyen genel bir anahtar vardır; varsayılan değer, anahtarın ayarlanmamış olmasıdır.

Aşağıdaki kuyruk yöneticisi özneliklerini kullanarak otomatik ortam görüntülemenin gerçekleşip gerçekleşmeyeceğini ve işlemin sıklığını denetleyebilirsiniz:

### IMGSCHED

Kuyruk yöneticisinin ortam görüntülerini otomatik olarak yazıp yazmadığını belirler

### IMGINTVL

Ortam görüntülerini yazma sıklığı (dakika)

### IMGLOGLN

Bir nesnenin önceki ortam görüntüsünden bu yana yazılan günlük megabayt değeri.

İş yükünün çok ağır olduğu gün içinde kritik bir süreniz varsa ve otomatik ortam görüntülerinin alınmasıyla sistem verimi etkilenmediğinden emin olmak istiyorsanız, **IMGSCHED(MANUAL)** ayarını yaparak otomatik ortam görüntülemeyi geçici olarak devre dışı bırakmak isteyebilirsiniz.

İş yükü sırasında istediğiniz zaman **IMGSCHED ' e** geçiş yapabilirsiniz.



**Uyarı:** Ortam görüntüleri almıyorsanız **MEDIALOG** ileriye taşınmaz; bu nedenle, kapsamları arşivlemeniz ya da yeterli disk alanınızın olduğundan emin olmanız gerekir.

**IMGRCOVO** özneliğini kullanarak diğer kullanıcı tanımlı nesnelere için otomatik ve el ile ortam görüntülerini de denetleyebilirsiniz:

- Kimlik doğrulama bilgileri
- Kanal
- İstemci bağlantısı
- Dinleyici
- Ad listesi
- Süreç
- Diğer ad kuyruğu
- Yerel kuyruk
- Hizmet
- Konu

Nesne kataloğu ve kuyruk yöneticisi nesnesi gibi iç sistem nesnelere için, kuyruk yöneticisi ortam görüntülerini uygun şekilde otomatik olarak yazar.

Özneliklere ilişkin ek bilgi için [ALTER QMGR](#) başlıklı konuya bakın.

Ayrıca, yalnızca yerel ve kalıcı dinamik kuyruklar için otomatik ve el ile ortam görüntülerini etkinleştirebilir ya da geçersiz kılabilirsiniz. Bunu **IMGRCOVQ** kuyruk özneliğini kullanarak yaparsınız.

**IMGRCOVQ** özneliğine ilişkin ek bilgi için [ALTER KUYRUK](#) başlıklı konuya bakın.

### Notlar:

1. Ortam görüntüleri yalnızca doğrusal günlük kaydı kullanıyorsanız desteklenir. Otomatik ortam görüntülerini etkinleştirdiyseniz, ancak çevrimsel günlük kaydı kullanıyorsanız, bir hata iletisi yayınlanır ve kuyruk yöneticisinin otomatik ortam görüntüleri özneliği geçersiz kılınır.
2. Otomatik ortam görüntülerini etkinleştirdiyseniz, ancak bir sıklık (dakika ya da megabayt) belirtmediyseniz, bir hata iletisi yayınlanır ve otomatik ortam görüntüleri yazılmaz.
3. Bir ortam görüntüsünü, isterseniz **IMGSCHEM(AUTO)** ayarını tanımladığınızda `rcdmqimg` komutunu kullanarak el ile kaydedebilirsiniz.

Bu, kuruluşunuz için uygun bir zamanda (örneğin, sisteminiz sessiz olduğunda) ortam görüntülerini almanızı sağlar. Otomatik ortam görüntüleme, bu manuel ortam görüntülerini dikkate alır, çünkü el ile ortam görüntüsü alınması, bir sonraki otomatik ortam görüntüsünün alınmadan önce aralığı ve günlük uzunluğunu sıfırlar.

4. Kuyruk yöneticisi, IBM MQ 9.1.0' den kalıcı iletileri kalıcı olmayan iletilere değil, yalnızca ortam görüntülerine yazar. Bu, IBM MQ 9.1.0 ya da sonraki bir sürüme geçiş yaparken ortam görüntülerinin boyutunu azaltabilir

## IMGLOGLN ve IMGINTVL ayarlarının nasıl yapılacağına karar verilmesi

**V 9.3.4** Varsayılan olarak **IMGLOGLN** , Yerel HA kuyruk yöneticileri dışındaki kuyruk yöneticileri için `off` olarak ayarlanır. (Yerel HA kuyruk yöneticileri, **IMGLOGLN** değeri, kurtarma günlüklerinin yazılacağı birimdeki kullanılabilir alanın %25 'i olarak ayarlanarak oluşturulur.)

**V 9.3.4** Varsayılan olarak **IMGINTVL** 60 dakika olarak ayarlanır. **IMGINTVL** ile belirtilen aralık, kuyruk yöneticisinde yeni bir görüntünün kaydedilmesine degecek kadar çok yeni iş gerçekleştirildiğinde kabul edilir. Aksi takdirde yeni resimlerin çekimleri gecikir.

Yapılandırmanız için en iyi çözümü elde etmek üzere **IMGLOGLN** ve **IMGINTVL** değerlerini değiştirebilirsiniz. **IMGLOGLN** ve **IMGINTVL** ' i yeterince büyük yapın; böylece kuyruk yöneticisi, ortam görüntülerini kaydetmek için zamanının yalnızca bir kısmını harcamış olur, ancak aşağıdakileri yapacak kadar küçüktür:

- Hasarlı nesnelere makul bir süre içinde kurtarılabilir, ve
- Yer tükenmeden günlüğünüz diskinize sığacak kadar küçük.

**IMGLOGLN** ayarını tanımlarsanız, iyi bir uygulama, **IMGLOGLN** ' in kuyruklarınızdaki veri miktarının ve iş yükünüzdeki veri hızının birçok katını oluşturmasıdır. **IMGLOGLN** ' i ne kadar büyük yaparsanız, kuyruk yöneticiniz ortam görüntülerini kaydetmeye o kadar az zaman harcar.

Benzer şekilde, **IMGINTVL** değerini ayarlarsanız, kuyruk yöneticisinin bir ortam görüntüsünü kaydetmek için harcadığı sürenin çoğuna **IMGINTVL** kadar olması iyi bir uygulamadır. Bir ortam görüntüsünü el ile kaydederek kaydetmenin ne kadar süreceğini öğrenebilirsiniz.

**IMGLOGLN** ve **IMGINTVL** öğelerini çok büyük yaparsanız, son ortam görüntüsünden sonraki tüm kapsamların yeniden oynatılması gerektiğinden, hasarlı bir nesnenin kurtarılması çok uzun sürebilir.

**IMGLOGLN** ve **IMGINTVL** ' yi, zarar görmüş bir nesneyi kurtarmak için gereken süre üst sınırının sizin için kabul edilebilir olması için yeterince küçük yapın.

**IMGLOGLN** ve **IMGINTVL** çok büyük olması, ortam görüntülerinin çok nadiren kaydedildiği için günlüğün çok büyük olduğu anlamına gelir.



**Uyarı:** Günlük dosya sistemi tamamen dolursa iş yükünüz yedekleneceği için, bu büyüklükte bir günlüğün günlük dosya sisteminize rahatça uyduğundan emin olun.

Hem **IMGINTVL** hem de **IMGLOGLN** ayarını yapabilirsiniz. Bu, ağır iş yükü sırasında ( **IMGLOGLN** tarafından denetlenir) otomatik ortam görüntülerinin düzenli olarak alındığından emin olmak için yararlı olabilir, ancak iş yükü çok hafif olduğunda ( **IMGINTVL** tarafından denetlenir) zaman zaman da alınır.

**IMGINTVL** ve **IMGLOGLN** , otomatik ortam görüntülerinin alındığı aralık ve günlük verileri uzunluğuna ilişkin hedeflerdir.



Bu öznitelikler sabit bir üst sınır ya da alt sınır olarak görülmemelidir. Kuyruk yöneticisi, uygun bir zaman olduğunu algılayarsa, kuyruk yöneticisi otomatik ortam görüntüsünü daha erken zamanlamaya karar verebilir:

- Kuyruk boş olduğundan, ortam görüntüsünü almak performans açısından en verimli olandır, ve
- Bir süredir bir ortam görüntüsü kaydedilmedi

Bazı durumlarda, otomatik ortam görüntüleri arasındaki boşluk, **IMGINTVL** ve **IMGLOGLN** resimlerinden herhangi birinden ya da her ikisinden biraz daha uzun olabilir.

Kuyruklardaki veri miktarı **IMGLOGLN** yaklaşıyorsa, ortam görüntüleri arasındaki boşluk **IMGLOGLN** değerinden büyük olabilir. Ortam görüntülerinin kaydedilmesi neredeyse **IMGINTVL** kadar uzun sürerse, ortam görüntüleri arasındaki boşluk **IMGINTVL** değerinden büyük olabilir.

Kuyruk yöneticisi zamanının çoğunu medya görüntülerini kaydederek geçireceği için bu kötü bir uygulamadır.

Otomatik ortam görüntüsü kaydı kullanılırken, kuyruk yöneticisi her nesne ve kuyruk için ayrı ayrı bir ortam görüntüsü kaydeder; böylece, kuyruk yöneticisi her nesne için görüntüler arasındaki aralığı ve günlük uzunluğunu ayrı ayrı izler.

Zamanla, ortam görüntülerinin kaydedilmesi, tüm nesnelere için ortam görüntülerinin aynı anda kaydedilmesi yerine aşamalı olarak yayılır. Bu aşamalandırma, ortam görüntülerinin kaydedilmesinin performans üzerindeki etkisini yayar ve ortam görüntülerinin manuel kayıt üzerinden otomatik olarak kaydedilmesinin bir başka avantajıdır.

## Ortam görüntülerini el ile alma-yalnızca doğrusal günlük kaydı

Bir kuyruğun ortam görüntüsünün kaydedilmesi, o kuyruktaki tüm kalıcı iletilerin günlüğe yazılmasını içerir. Büyük hacimli ileti verileri içeren kuyruklar için bu, günlüğe büyük miktarda veri yazılmasını içerir ve bu işlem, gerçekleşirken sistemin performansını etkileyebilir.

Diğer nesnelerin ortam görüntüsü kullanıcı verileri içermediğinden, diğer nesnelerin ortam görüntülerinin kaydedilmesi nispeten hızlı olabilir.

Kuyrukların ortam görüntülerini kaydederken dikkatli bir şekilde göz önünde bulundurmanız gerekir, böylece işlem en yüksek iş yüküne engel olmaz.

Ortam kurtarma için gereken en eski günlük kapsamını güncellemek üzere tüm nesnelerin ortam görüntüsünü düzenli olarak kaydetmeniz gerekir.

Bir kuyruğun ortam görüntüsünü kaydetmek için iyi bir zaman, o noktada günlüğe ileti verileri yazılmadığı için kuyruğun boş olmasıdır. Tersine, kuyruk çok derin olduğunda ya da üzerinde çok büyük iletiler olduğunda kötü bir zaman olur.

Bir kuyruğun ortam görüntüsünü kaydetmek için iyi bir zaman, sisteminiz sessizken, en yoğun iş yükü sırasında kötü bir zaman gelir. Örneğin, iş yükünüz gece yarısı her zaman sessizse, her gece yarısı ortam görüntülerini kaydetmeye karar verebilirsiniz.

Her bir kuyruğunuzun kaydının aşamalandırılması, performans etkisini dağıtabilir ve böylece etkisini azaltabilir. Ortam görüntülerini en son kaydettiğiniz zamandan bu yana ne kadar uzun sürdüyseniz, ortam kurtarma için gereken günlük kapsamları sayısı arttıkça bu görüntülerin kaydedilmesi de o kadar önemli hale gelir.

**Not:** Ortam kurtarma işlemi gerçekleştirilirken, gereken tüm günlük kütükleri aynı anda günlük kütüğü dizininde bulunmalıdır. Gerekli tüm günlük dosyalarını tutmak üzere disk alanının tükenmesini önlemek için, kurtarmak isteyebileceğiniz nesnelerin olağan ortam görüntülerini aldığınızdan emin olun.

Örneğin, kuyruk yöneticinizdeki tüm nesnelere ortam görüntüsünü almak için aşağıdaki örneklerde gösterildiği gibi **rcdmqimg** komutunu çalıştırın:

**Windows** AçıkWindows

```
rcdmqimg -m QMNAME -t all *
```



```
rcdmqimg -m QMNAME -t all "*"
```

**rcdmqimg** ' un çalıştırılması, ortam günlüğü sıra numarasını (LSN) ileriye taşır. Günlük sıra numaralarıyla ilgili daha fazla ayrıntı için bkz. "[dmpmqlog komutunu kullanarak günlüğün içeriğinin dökümü](#)" sayfa 652. **rcdmqimg** otomatik olarak çalışmaz, bu nedenle el ile ya da oluşturduğunuz otomatik bir görevden çalıştırılmalıdır. Bu komutla ilgili daha fazla bilgi için bkz. [rcdmqimg](#) ve [dmpmqlog](#).

Günlük alanını yönetmek için **rcdmqimg** ile ortam görüntülerinin el ile kaydedilmesi gerekmez; kuyruk yöneticisi denetimli otomatik ortam görüntülemesiyle doğrusal günlük kaydını kullanmayı seçmiş olsanız.

**Not:** `rcdmqimg` komutu çalıştırılırken AMQ7467 ve AMQ7468 iletileri de yayınlanabilir.

## Kısmi ortam görüntüleri

IBM MQ iletilerinin yalnızca yakın gelecekte kullanılması beklenen veriler için kullanılması iyi bir uygulamadır; böylece, her ileti nispeten kısa bir süre boyunca kuyrukta olur.

Bunun tersine, bir veritabanı gibi uzun vadeli verileri depolamak için IBM MQ iletilerini kullanmak kötü bir uygulamadır.

Ayrıca, kuyruklarınızın nispeten sığ ve iletileri uzun zamandır kuyrukta olan derin kuyruklara sahip olmanızı sağlamak da iyi bir uygulamadır.

Bu yönergeleri izleyerek, kuyruk yöneticisinin ortam görüntülerinin otomatik olarak kaydedilmesini en iyi duruma getirmesini sağlayın.

Boş bir kuyruğun ortam görüntüsünün kaydedilmesi çok verimlidir (performans açısından), ancak üzerinde büyük miktarda veri bulunan bir kuyruğun ortam görüntüsünü almak çok verimsizdir, çünkü tüm bu verilerin ortam görüntüsünde günlüğe yazılması gerekir.

Üzerine yeni ileti konan sığ kuyruklar için, kuyruk yöneticisi daha fazla eniyileme yapabilir.

Kuyruktaki tüm iletiler en son geçmişe yerleştirildiyse, kuyruk yöneticisi ortam görüntüsünü tüm iletiler konmadan hemen önce bir zaman (*kurtarma noktası*) adına kaydedebilir ve böylece boş kuyruğun görüntüsünü kaydedebilir. Bu süreç, performans açısından çok düşük maliyetlidir.

Kurtarma noktasında kuyrukta bulunan tüm iletiler daha sonra alındıysa, artık kuyrukta olmadıkları için bu iletilerin ortam görüntüsüne kaydedilmesi gerekmez.

Buna *kısmi ortam görüntüsü* denir. Daha sonra, kuyruğun kurtarılması gerekmesi durumunda, son ortam görüntüsünden bu yana bu kuyrukla ilgili tüm günlük kayıtları yeniden yürütülecektir, bu nedenle en son konan tüm iletiler geri yüklenecektir.

Kurtarma noktasında kuyrukta birkaç ileti bulunsan bile (bu nedenle kısmi ortam görüntüsüne kaydedilmesi gerekir), bu küçük kısmi ortam görüntüsünün kaydedilmesi, tüm iletilerin tam ortam görüntüsünden daha verimlidir.

İletilerin kısa bir süre kuyruklarda kalmasını sağlamak, ortam görüntülerinin otomatik olarak kaydedilmesinin performansını artırabilir.

### *Gereksiz günlük dosyalarının belirlenmesi-yalnızca doğrusal günlük kayıt*

Çevrimsel günlük kayıt için, verileri hiçbir zaman günlük dizininden silmez. Doğrusal günlük dosyalarını yönetirken, hangi dosyaların silinebileceğinden ya da arşivlenebildiğinden emin olmak önemlidir. Bu bilgiler, bu kararı vermeniz için size yardımcı olacaktır.

"Daha eski" günlük dosyalarını belirlemek için dosya sisteminin değişiklik saatlerini kullanmayın. Yalnızca günlük dosyası numarasını kullanın. Kuyruk yöneticisinin günlük dosyalarını kullanımı, günlük dosyalarının gerekli olmadan önce önceden ayrılması ve biçimlendirilmesi de dahil olmak üzere karmaşık kuralları izler. Göreli yaşı belirlemek için bu saatleri kullanmaya çalışırsanız, değişiklik saatlerine sahip günlük dosyalarını görebilirsiniz.

Gereken en eski günlük dosyasını saptamak için kullanabileceğiniz üç yer vardır:

- DISPLAY QMSTATUS komutu
- Olay iletilerini günlüğe kaydedici ve son olarak
- Hata günlüğü iletileri

DISPLAY QMSTATUS komutunda, aşağıdaki işlemleri yapmak için gereken en eski günlük kapsamını belirlemek için:

- Kuyruk yöneticisini yeniden başlatın ve DISPLAY QMSTATUS RECLOGkomutunu verin.
- Ortam kurtarma işlemini gerçekleştirin ve DISPLAY QMSTATUS MEDIALOGkomutunu verin.
- Arşiv bildirimini adını saptayın, DISPLAY QMSTATUS ARCHLOGkomutunu verin.

**RESET QMGR TYPE (REDUCELOG)** komutunu çalıştırarak, döngüsel günlük kaydı kullanırken ikincil günlük kapsamı sayısını azaltabilirsiniz.

Genel olarak, daha düşük bir günlük dosyası numarası daha eski bir günlüğü belirtir. Çok yüksek bir günlük dosyası devri olmadıkça, 10 yıl boyunca günde 3000 günlük dosyası siparişi verilmedikçe, 9 999 999 999 numaralı kaydirmayı yapmanız gerekmez. Bu durumda, RECLOG değerinden küçük bir sayıyla herhangi bir günlük dosyasını arşivleyebilir ve RECLOG ve MEDIALOG değerlerinden daha küçük bir sayı içeren herhangi bir günlük dosyasını silebilirsiniz.



**Uyarı:** Günlük dosyası kaydı için, 9 999 999 'dan sonraki sayı sıfırdır.

#### *Günlük dosyası yeri*

Günlük dosyalarınız için bir konum seçerken, IBM MQ disk yeri yetersizliği nedeniyle yeni bir günlüğü biçimlendiremezse işlemin önemli ölçüde etkilendiğini unutmayın.

Çevrimsel bir günlük kullanıyorsanız, sürücüde en az yapılandırılan birincil günlük dosyaları için yeterli yer olduğundan emin olun. Ayrıca, günlüğün büyümesi gerekiyorsa, en az bir ikincil günlük kütüğü için de yer açın.

Doğrusal bir günlük kullanıyorsanız, çok daha fazla alana izin verin; veriler günlüğe kaydedildikçe günlük tarafından tüketilen alan sürekli olarak artar.

Günlük dosyalarını, kuyruk yöneticisi verilerinden ayrı bir disk sürücüsüne yerleştirmeniz gerekir.

Bu aygıttaki veri bütünlüğü çok önemlidir-yerleşik yedeklilik için izin vermeniz gerekir.

Günlük dosyalarını ikizlenmiş bir düzende birden çok disk sürücüsüne yerleştirmek de olanaklı olabilir. Bu, günlüğü içeren sürücünün arızalanmasını korur. İkizleme olmadan, IBM MQ sisteminizin son yedeğine geri dönmek zorunda kalabilirsiniz.

### **Coldstart: Günlük kapsamı eksikse ya da bozursa ne yapmalı?**

İşletmeniz yeniden başlatma kurtarması için gereken günlük kapsamlarından bazılarını ya da tümünü kaybederse, kuyruk yöneticisi kurtarma günlüğünü yeniden yürütemez ve yeniden başlatılamaz. Kurtarma günlüğü herhangi bir şekilde bozulduğunda, veri bütünlüğünü koruma pahasına kuyruk yöneticinizin yeniden başlatılmasını gerektiriyorsanız, bu işlem şiddetle önerilmemesine rağmen gerçekleştirilebilir. Bu işlem, bir kuyruk yöneticisini *coldstarting* (soğuk başlatma) olarak bilinir.

**Önemli:** Bir kuyruk yöneticisini Coldstarting, yalnızca istisnai durumlarda dikkate alınmalı ve bu sayfada açıklandığı gibi veri bütünlüğü risklerini taşımaktadır. IBM, bozuk veri dosyalarına yanıt olarak, coldstarting değil, bir kuyruk yöneticisini yeniden oluşturmanızı önerir.

Operasyonel nedenlerden ötürü bir coldstart gerekiyorsa, IBM destek temsilcinizle görüşün ve sorunun temel nedenini gözden geçirin. Coldbaşlatılan bir kuyruk yöneticisini, en kısa zamanda yeniden oluşturulan bir kuyruk yöneticisiyle değiştirmeniz gerekir.

### **Coldstart 'ın etkileri**

Coldstart 'da, kuyruk yöneticisi boş bir kurtarma günlüğü yaratır ve kuyruk dosyalarındaki ve var olan durumlarındaki diğer nesne dosyalarındaki verilere dayanır. Kuyruk dosyalarındaki veriler tutarsız olabileceğinden, iletiler kaybolabilir, çoğaltılabilir, bozulabilir ya da tutarsız olabilir.

Kuyruk yöneticisi, kurtarma günlüğündeki ve nesne dosyalarındaki diğer tüm kalıcı olarak saklanan nesnelerin yapılanışını saklar. Diğer iç durum verileri de kurtarma günlüğüne kaydedilir; bu nedenle, coldstart 'ta iç durum verileri ilk durumuna getirilir ve diğer tüm yapılandırma verileri yanlış olabilir.

Coldstart 'ın etkileri öngörülemez ve geniş aralıklidir, bu nedenle kesinlikle gerekmedikçe bir coldstart (soğuk başlatma) kullanmaktan kaçınmalısınız. Yeniden başlatma işleminden sonra, kuyruk ve nesne dosyalarındaki bilgiler tutarsız olabilir ve kuyruk yöneticisi yeniden başlamaz.

Kuyruk yöneticisi yeniden başlatılırsa, hangi ileti verilerine ya da yapılanışına güvenebileceğini ve güvenilemeyeceğini keşfetmenin basit bir yolu yoktur. Ayrıca, bir coldstart işleminden sonra kuyruklar zarar görebilir ve bu nedenle tamamen kullanılamaz duruma gelebilir.

Ayrıca, belirli bir kuyruktan alabiliyor ya da kuyruğa yerleşebiliyorsanız, kuyruktaki iletiler bozuk, eksik ya da çoğaltılmış olabilir. İşlemler ve kanallar şüphe içinde kalmış olabilir. Kuyruk yöneticiniz başarıyla başlasa ve kuyruklar sağlam görünse bile, coldstart 'ın öngörülemeyen etkileri çok daha sonraya kadar gerçekleşmeyebilir.

## Yeniden başlatmanız gerekirse ne yapmalısınız?

Bir coldstart işlemi gerçekleştirmek standart operasyonel uygulama olarak kabul edilmemelidir ve IBM bunu yapmana şiddetle karşı çıkmaktadır. Ancak, kesinlikle bir kuyruk yöneticisini yeniden başlatmanız gereken bir konumdaysanız, [IBM MQ Destek](#) ile iletişim kurun.

Bir kuyruk yöneticisini coldstarting işlemi, doğrusal bir kuyruk yöneticisi için dairesel olandan çok daha karmaşıktır. IBM MQ 9.1.3 içinde, coldstart işlemi çok basitleştirilmiştir ve artık günlük kapsamlarını kopyalamayı ya da yeniden adlandırmayı içermez.

IBM MQ 9.1.3' den, bir kuyruk yöneticisini yeniden başlatmak için **strmqm** komutuna geçirdiğiniz bir anahtar size verecek olan IBM Support (Destek) ile iletişim kurun.



**Uyarı:** IBM MQ 9.1.3 coldstart komutu, el ile coldstart ile aynı veri bütünlüğünü kaybetme risklerini taşımaya devam eder ve IBM bunu yapmanızı şiddetle engeller.

## Gelecekteki soğuk başlangıçlarını ortadan kaldırma: bir istek

strmqm komutu coldstart için bir anahtar gerektirir; çünkü IBM MQ , coldstart yapmanız gerekiyorsa IBM MQ Support ile iletişim kurmanızı ister; çünkü IBM MQ bu duruma nasıl girdiğinizi anlamaya isteklidir.

Açıkça görülüyor ki, Coldstart en iyi önlenmeyi sağlayan bir şey. IBM MQ , kuyruk yöneticinizi yeniden başlatmanıza gerek kalmayacağından emin olmak için büyük çaba harcadı ve IBM , ürünün coldstart komutunu çalıştırmayı azaltmak için yapabileceği başka bir şey olup olmadığını keşfetmek istiyor.

## coldstart önlemeye ilişkin önlemler

Kuyruk yöneticisi yaratılırken varsayılan günlüğe kaydetme yöntemi çevrimsel günlüğe kaydetmektir. Çevrimsel günlük kaydı ile, kuyruk yöneticisinin belirli bir büyüklükteki birincil ve ikincil günlük kapsamı sayısına izin verirsiniz. Tüm birincil ve ikincil günlük kapsamlarını içerecek kadar büyük günlük dosya sistemi oluşturun ve bunları denetlemeniz gerekmemelidir.

Alternatif olarak, çevrimsel yerine doğrusal günlük kaydını kullanabilirsiniz. Doğrusal günlük kaydı, olası olmayan durumlarda, kuyrukları ve diğer nesneleri kurtarma yeteneği sağlar. Ancak varsayılan olarak, doğrusal günlük kaydı, yeniden başlatma ya da ortam kurtarma için artık gerekli olmayan günlük kapsamlarını silmenizi gerektirir. Bu, el ile günlük yönetimi olarak adlandırılır.

Günlük kapsamlarını bu şekilde yönetirken, yanlışlıkla çok fazla günlük kapsamını silebilir ve bu nedenle coldstart işlemi sona erer. Bu riski azaltmak için otomatik günlük yönetimini kullanın, böylece kuyruk yöneticisi sizin adınıza günlük kapsamlarını yönetir.

En iyi yöntem, kurtarma günlüğünüzü yalnızca kurtarma günlüğünü içeren ayrı bir günlük dosya sistemine koymaktır. Kurtarma günlüğünüzü, kuyruk yöneticinizin geri kalanıyla aynı dosya sistemine koyarsanız, büyük kuyruk dosyaları nedeniyle dosya sisteminin yanlışlıkla doldurulduğunu da bulabilirsiniz. Kuyruk yöneticisine ilişkin günlük dizinini ayrı bir dosya sistemi yapın ya da **crtmqm** komutunda **-ld** komut satırını seçeneğini kullanarak farklı bir günlük dosya sistemi belirtin.

Kuyruk dosyalarını tutan dosya sistemi doluyorsa, bu kuyruklara yerleřtirilemeyebilir, ancak kuyruk yöneticisi alıřmaya devam eder. Kurtarma gnlğn ieren dosya sistemi doldurulursa, kuyruk yöneticisi aniden sona erer ve siz yer aıncaya kadar yeniden bařlamaz.

Yeniden bařlatma kurtarması iin gereken gnlk kapsamlarını silmemeye dikkat edin; tersi durumda, kendinizi Coldstart 'a gereksinim duyarsınız. Bazen, kurtarma gnlklerini ieren disk bařarısız olduėu iin yeniden bařlatmanız gerektiėini dřnebilirsiniz. En iyi uygulama, kurtarma gnlğn kopyalanmıř bir diske koymak ve bu nedenle bir disk kmesi riskini azaltmaktır.

İletilerinizin ve yapılandırmanızın yeni bir yeni kuyruk yöneticisine tařınması, daha nce bařlatılan bir kuyruk yöneticisiyle ilgili devam eden sorunlar olasılıėını ortadan kaldırır.

Uzun zaman nce soėuk bařlatılmıř olsalar ve bu arada durdurulmuř, yeniden bařlatılmıř ve yeni dzeye geirilmiř olsalar bile, daha nce bařlatılan kuyruk yöneticilerini not alın. IBM Desteėi ile iletiřim kurduėunuzda, kuyruk yöneticisinin daha nce bařlatılıp bařlatılmadıėı ve bařlatıldıysa, coldstart gereksinimine neyin neden olduėu hakkında mmkn olduėunca ok bilgi verin.

## Kurtarma iin gnlğn kullanılması

Hatalardan kurtulmanıza yardımcı olması iin gnlklerden alınan bilgileri kullanabilirsiniz.

Verilerinizin zarar grmesi iin birkaç yol vardır. IBM MQ , ařaėıdakileri gerekleřtirmenize yardımcı olur:

- Zarar grmř bir veri nesnesi
- Sistemdeki bir g kaybı
- İletiřim hatası

Bu blmde, gnlklerin bu sorunlardan kurtulmak iin nasıl kullanıldıėı ele alınmıřtır.

### ***G kaybından ya da iletiřim hatalarından kurtarma***

IBM MQ , hem iletiřim hatalarından hem de g kaybından kurtulabilir. Bazen, bir dosyanın istenmeyen silinmesi gibi diėer sorun tiplerinden de kurtarılabilir.

Bir iletiřim hatası durumunda, kalıcı iletiler alan bir uygulama tarafından kaldırılıncaya kadar kuyruklarda kalır. İleti iletiliyor ise, bařarılı bir řekilde iletilinceye kadar iletim kuyruėunda kalır. Bir iletiřim hatasından kurtulmak iin, genellikle bařarısız olan baėlantıyı kullanarak kanalları yeniden bařlatabilirsiniz.

Gc kaybederseniz, kuyruk yöneticisi yeniden bařlatıldıėında IBM MQ , hata sırasında kuyrukları kesinleřtirdi durumuna geri ykler. Bu, kalıcı iletilerin kaybolmamasını saėlar. Kalıcı olmayan iletiler atılır; IBM MQ aniden durduėunda hayatta kalmazlar.

### ***Hasarlı nesnelere kurtarılması***

IBM MQ nesnesinin, rneėin istenmeyen zararlar nedeniyle kullanılamaz duruma gelmesinin eřitli yolları vardır. Daha sonra, sisteminizin tamamını ya da bir kısmını kurtarmanız gerekir. Yapılması gereken iřlem, hasarın ne zaman saptandıėına, seilen gnlk ynteminin ortam kurtarmayı destekleyip desteklemediėine ve hangi nesnelere zarar grdėine baėlıdır.

## Ortam kurtarma

Nesnelere iliřkin ortam grntlerini, zarar grrse kurtarılabilmesi iin kaydedebilirsiniz. Bu zellik yalnızca doėrusal gnlk kaydı ya da eřlenmiř gnlk kaydı kullanan kuyruk yöneticilerine ve doėrusal gnlk kaydı iin yalnızca kurtarılabilir olarak tanımlanan nesnelere iin kullanılabilir. Nesne tiplerinin kurtarılabilir olduėunu tanımlamak iin **IMGRCOVO** ve **IMGRCOVQ** kuyruk yöneticisi zneliklerini kullanın, bkz. [ALTER QMGR](#). Kurtarılabilir olarak tanımlanmayan bir nesne zarar grrse, kurtarma seenekleri evrimsel gnlğe kaydetme iřlemiyle aynıdır.

Ortam kurtarma, nesnelere doėrusal bir gnlğe ya da eřlenmiř bir gnlğekaydedilen bilgilerden yeniden oluřturur. rneėin, bir nesne dosyası yanlışlıkla silinirse ya da bařka bir nedenle kullanılamaz duruma gelirse, ortam kurtarma iřlemi dosyayı yeniden oluřturabilir. Bir nesnenin ortam kurtarması iin gereken gnlkteki bilgilere *ortam grnts*denir.

Ortam görüntüsü, nesnenin kendisinin yeniden yaratılabileceği bir nesnenin görüntüsünü içeren günlük kayıtları dizisidir.

Bir nesneyi yeniden oluşturmak için gereken ilk günlük kaydı, nesnenin *ortam kurtarma kaydı* olarak bilinir; nesne için en son ortam görüntüsünün başlangıcını oluşturur. Her bir nesnenin ortam kurtarma kaydı, bir denetim noktası sırasında kaydedilen bilgi parçalarından biridir.

Bir nesne ortam görüntüsünden yeniden oluşturulduğunda, son görüntünün alınmasından bu yana nesne üzerinde gerçekleştirilen güncellemeleri açıklayan günlük kayıtlarını yeniden yürütmek de gerekir.

Örneğin, kalıcı bir ileti kuyruğa konmadan önce kuyruk nesnesinin resmini içeren bir yerel kuyruk düşünün. Nesnenin en son görüntüsünü yeniden oluşturmak için, görüntünün kendisini yeniden yürütmenin yanı sıra, iletinin kuyruğa konmasını kaydeden günlük girişlerini yeniden yürütmek gerekir.

Bir nesne yaratıldığında, yazılan günlük kayıtları nesneyi tamamen yeniden yaratmak için yeterli bilgi içerir. Bu kayıtlar, nesnenin ilk ortam görüntüsünü oluşturur. Daha sonra, her kapanma sırasında, kuyruk yöneticisi ortam görüntülerini otomatik olarak aşağıdaki gibi kaydeder:

- Yerel olmayan tüm süreç nesnelerinin ve kuyruklarının resimleri
- Boş yerel kuyrukların resimleri

Ortam görüntüleri, `rcdmqimg` içinde açıklanan `rcdmqimg` komutu kullanılarak el ile de kaydedilebilir. Bu komut, IBM MQ nesnesinin ortam görüntüsünü yazar.

**IMGSCHED(AUTO)** ayarlanırsa, kuyruk yöneticisi ortam görüntülerini otomatik olarak kaydeder. Ek bilgi için bkz. **ALTER QMGR**, **IMGINTVL** ve **INGLOGLN**.

Bir ortam görüntüsü yazıldığında, zarar görmüş nesnelere yeniden oluşturmak için yalnızca ortam görüntüsünü tutan günlükler ve şu andan sonra oluşturulan tüm günlükler gerekir. Ortam görüntülerinin oluşturulmasının yararı, kullanılabilir boş depolama miktarı ve günlük dosyalarının oluşturulma hızı gibi etkenlere bağlıdır.

## Ortam görüntülerinden kurtarma

Kuyruk yöneticisi başlatılırken, bazı nesnelere ortam görüntülerinden otomatik olarak kurtarılır. Kuyruk yöneticisi son kapandığında tamamlanmamış olan herhangi bir harekette yer aldıysa ve yeniden başlatma işlemi sırasında bozuk ya da hasarlı olduğu bulunursa, kuyruk otomatik olarak kurtarılır.

IBM MQ nesnesini yeniden oluşturmak için günlükteki kayıtları yeniden yürüden `rcrmqobj` komutunu kullanarak diğer nesnelere el ile kurtarmanız gerekir. Nesne, görüntünün kaydedildiği zaman ile yeniden yaratma komutunun verildiği zaman arasındaki tüm uygulanabilir günlük olaylarıyla birlikte, günlükte bulunan en son görüntüsünden yeniden oluşturulur. Bir IBM MQ nesnesi zarar görürse, yapılabilecek tek geçerli işlem nesneyi silmek ya da bu yöntemle yeniden yaratmak olur. Kalıcı olmayan iletiler bu şekilde kurtarılamaz.

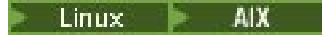
`rcrmqobj` komutuyla ilgili daha fazla ayrıntı için bkz. `rcrmqobj`.

Ortam kurtarma kaydını ve sonraki tüm günlük dosyalarını içeren günlük dosyası, bir nesne için ortam kurtarma girişimi sırasında günlük dosyası dizisinde bulunmalıdır. Gerekli bir dosya bulunamazsa, AMQ6767 işletmen iletisi yayınlanır ve ortam kurtarma işlemi başarısız olur. Yeniden oluşturmak istediğiniz nesnelere olağan ortam görüntülerini almazsanız, bir nesneyi yeniden oluşturmak için gereken tüm günlük dosyalarını tutmak için yeterli disk alanınız olmayabilir.

**V9.3.3** Yerel HA kuyruk yöneticileri, eşlenmiş günlük kaydını kullanır. Bu tür kuyruk yöneticileri, hasar saptandığında uygun nesnelere otomatik olarak kurtarmayı dener. Başlatıldıktan sonra, Native HA kuyruk yöneticileri, varsayılan olarak, nesne hasarı algılandığında zamanuyumsuz kurtarma işlemi otomatik olarak dener. Örneğin, nesne bir uygulama tarafından kullanılırsa ya da ortam kurtarma için gereken günlük kapsamı kullanılamıyorsa, kurtarma hemen gerçekleştirilemez. Bu durumlarda, zamanuyumsuz kurtarma işlemi belirli aralıklarla yeniden dener. Kurtarmayı önleyen sorun çözümlenirse, nesne sonraki yeniden denemede kurtarılacak ya da nesne, `rcrmqobj` komutu kullanılarak el ile kurtarılabilir.

## Hangi nesne dosyaları var?

Kuyruk yöneticisi, **runmqsc** içinde tanımlanan nesnelerin özniteliklerini diskteki dosyalarda saklar. Bu nesne dosyaları, kuyruk yöneticisinin veri dizini altındaki alt dizinlerdedir.

 Örneğin, AIX and Linux platformlarında kanallar `/var/mqm/qmgrs/qmgr/channel` içinde depolanır.

Bu nesne dosyalarındaki veriler, nesnelerin ortam görüntüsüdür. Bu nesne dosyaları silinirse ya da bozulursa, o dosyada saklanan nesne zarar görür. Doğrusal günlük kaydı kuyruk yöneticisi kullanılarak, zarar görmüş nesnelere `rcrmqobj` komutu kullanılarak günlükten kurtarılabilir. Eşlenmiş günlük kaydı (Native HA) kuyruk yöneticileri, algılandıklarında zarar görmüş nesnelere otomatik olarak kurtarmayı dener.

Çoğu nesne dosyası, yalnızca nesnenin özniteliklerini içerir, bu nedenle kanal dosyaları kanalların özniteliklerini içerir. Kural dışı durumlar şunlardır:

- Katalog  
Nesne kataloğu, tüm tiplerdeki tüm nesnelere kataloğa alır ve `qmanage/QMQMOBJCAT` içinde saklanır.
- Eşitleme Dosyaları  
Eşitleme dosyası, tüm kanallarla ilişkili iç durum verilerini içerir.
- Kuyruklar  
Kuyruk dosyaları, kuyruktaki iletilerin yanı sıra o kuyruğun özniteliklerini de içerir.

**runmqsc** ya da IBM MQ Explorer içinde gösterilen bir katalog ya da syncfile nesnesi olmadığını unutmayın.

Katalog ve kuyruk yöneticisi kaydedilebilir, ancak kurtarılamaz. Bu nesnelere zarar görürse, kuyruk yöneticisi öncelikli olarak sona erer ve bu nesnelere yeniden başlatıldığında otomatik olarak kurtarılır.

Sürekli abonelikler bir sistem kuyruğunda saklandığı için, kaydedilecek ya da kurtarılabilecek nesnelere abonelikler listelenmez. Sürekli abonelikleri kaydetmek ya da kurtarmak için `SYSTEM.DURABLE.SUBSCRIBER.QUEUE QUEUE`.

## Başlatma sırasında hasarlı nesnelerin kurtarılması

Kuyruk yöneticisi başlatma sırasında zarar görmüş bir nesne saparsa, bu işlem nesnenin tipine ve kuyruk yöneticisinin ortam kurtarmayı destekleyecek şekilde yapılandırılıp yapılandırılmamasına bağlıdır.

Kuyruk yöneticisi nesnesi zarar gördüyse, kuyruk yöneticisi nesneyi kurtarmadığı sürece başlatılamaz. Kuyruk yöneticisi doğrusal bir günlükle yapılandırıldıysa ve ortam kurtarmayı destekliyorsa, IBM MQ otomatik olarak, kuyruk yöneticisi nesnesini ortam görüntülerinden yeniden yaratmayı dener. Seçilen günlük yöntemi ortam kurtarmayı desteklemiyorsa, kuyruk yöneticisinin bir yedeğini geri yükleyebilir ya da kuyruk yöneticisini silebilirsiniz.

Kuyruk yöneticisi durduğunda herhangi bir hareket etkinse, kuyruk yöneticisini başarıyla başlatmak için, bu hareketlere konan ya da bu hareketlere konan kalıcı, kesinleştirilmemiş iletileri içeren yerel kuyruklar da gereklidir. Bu yerel kuyruklardan herhangi birinin zarar görmüş olduğu bulunursa ve kuyruk yöneticisi ortam kurtarmayı destekliyorsa, otomatik olarak bunları ortam görüntülerinden yeniden yaratmaya çalışır. Kuyruklardan herhangi biri kurtarılamazsa, IBM MQ başlatılamaz.

Ortam kurtarmayı desteklemeyen bir kuyruk yöneticisinde başlatma işlemi sırasında kesinleştirilmemiş iletileri içeren zarar görmüş yerel kuyruklar saptanırsa, kuyruklar zarar görmüş nesnelere işaretlenir ve bu kuyruklardaki kesinleştirilmemiş iletiler dikkate alınmaz. Bunun nedeni, böyle bir kuyruk yöneticisinde zarar görmüş nesnelerin ortam kurtarma işleminin gerçekleştirilememesi ve geriye kalan tek işlemin bunları silmek olması olabilir. Herhangi bir zarar olduğunu bildirmek için AMQ7472 iletileri yayınlandı.



## Hasarlı nesnelerin başka zamanlarda kurtarılması

Nesnelerin ortam kurtarma işlemi yalnızca başlatma sırasında otomatik olarak gerçekleştirilir (varsayılan olarak otomatik kurtarma kullanan Native HA kuyruk yöneticileri dışında). Diğer zamanlarda, nesne hasarı algılandığında, AMQ7472 işletmen iletisi yayınlanır ve nesneyi kullanan işlemlerin çoğu MQRC\_OBJECT\_ZARAR görmüş dönüş koduyla başarısız olur. Kuyruk yöneticisi başlatıldıktan sonra herhangi bir zamanda kuyruk yöneticisi nesnesi zarar görürse, kuyruk yöneticisi önleyici bir kapatma işlemi gerçekleştirir. Bir nesne zarar gördüğünde nesneyi silebilirsiniz ya da kuyruk yöneticisi doğrusal bir günlük kullanıyorsa, **rcrmqobj** komutunu kullanarak nesneyi ortam görüntüsünden kurtarmayı deneyin (daha fazla ayrıntı için bkz. [rcrmqobj](#)).

Bir kuyruk (ya da başka bir nesne) zarar görürse, **MEDIALOG** ileri gitmez. Bunun nedeni, **MEDIALOG** ' in ortam kurtarma için gerekli en eski kapsam olmasıdır. İş yükünüz devam ediyorsa, **CURRLOG** devam eder ve yeni kapsamlar yazılır. Yapılandırmanıza bağlı olarak ( **LogManagement** ayarınız da içinde olmak üzere), bu işlem günlük dosya sisteminizi doldurmaya başlayabilir. Günlük dosya sistemi tamamen doldurulursa, hareketler geriye işlenir ve kuyruk yöneticisi aniden sona erebilir. Bu nedenle, bir kuyruk zarar gördüğünde, kuyruk yöneticiniz sona ermeden önce işlem yapmak için yalnızca sınırlı bir süreniz olabilir. Ne kadar zamanınız olduğuna, iş yükünün kuyruk yöneticisinin yeni kapsamlar yazmasına neden olma hızına ve günlük dosya sisteminde boş alan miktarına bağlıdır.

El ile günlük yönetimi kullanıyorsanız, kurtarma işlemi yeniden başlatmak için gerekli olmayan kapsamı arşivleyip bunları, ortam kurtarma için hala gerekli olsalar da günlük dosya sisteminde siliyor olabilirsiniz. Bu, gerektiğinde arşivinizden geri yükleyebileceğiniz sürece kabul edilebilir. Bu ilke, bir kuyruk zarar gördüğünde günlük dosya sistemi dolmasına neden olmaz ve **MEDIALOG** ileriye doğru ilerlemeyi durdurur. Ancak, yalnızca yeniden başlatma ya da ortam kurtarma için gerekli olmayan kapsamı arşivler ve silerseniz, bir kuyruk zarar görürse günlük dosya sisteminizi doldurmaya başlar.

Otomatik ya da arşiv günlüğü yönetimi kullanıyorsanız, arşivlenmiş olsalar ve **SET LOG ARCHIVE** kullanarak kuyruk yöneticisine bildirmiş olsanız da, kuyruk yöneticisi ortam kurtarma işlemi için hala gerekli olan kapsamı yeniden kullanmayacaktır. Sonuç olarak, bir kuyruk zarar görürse, günlük dosya sistemi doldurmaya başlar.

Bir kuyruk zarar görürse, OBJECT HASARLI FFC ' ler yazılır ve **MEDIALOG** ilerlemeyi durdurur. Hasarlı nesne, **runmqsc** içinde durumunu görüntülediğinizde FFDC ' den ya da en eski **MEDIALOG** nesnesine sahip nesne olduğu için tanımlanabilir.

Günlük dosya sistemi doluysa ve günlük dosya sistemi olduğu için iş yükünün geri çekilmesinden endişe duyuyorsanız, nesne kurtarılıyor ya da iş yükünüzü durduruyor olabilir.

**V9.3.3** Yerel HA kuyruk yöneticilerinin (eşlenmiş günlük kaydını kullanan) zarar görmüş nesnelere otomatik kurtarma girişiminde bulunulması. Başlatıldıktan sonra, Native HA kuyruk yöneticileri, varsayılan olarak, nesne hasarı algılandığında zamanuyumsuz kurtarma işlemi otomatik olarak dener. Örneğin, nesne bir uygulama tarafından kullanılırsa ya da ortam kurtarma için gereken günlük kapsamı kullanılamıyorsa, kurtarma hemen gerçekleştirilemez. Bu durumlarda, zamanuyumsuz kurtarma işlemi belirli aralıklarla yeniden dener. Kurtarmayı önleyen sorun çözümlenirse, nesne sonraki yeniden denemede kurtarılacak ya da nesne, **rcrmqobj** komutu kullanılarak el ile kurtarılabilir.

## IBM MQ günlük dosyalarını koruma

Bir kuyruk yöneticisi çalışırken günlük dosyalarına dokunmayın, kurtarma işlemi olanaksız olabilir. Günlük dosyalarını istenmeyen değişikliklere karşı korumak için süper kullanıcı ya da mqm yetkisini kullanın.

Bir IBM MQ kuyruk yöneticisi çalışırken etkin günlük dosyalarını el ile kaldırmayın. Bir kullanıcı, bir kuyruk yöneticisinin yeniden başlatması gereken günlük dosyalarını yanlışlıkla silerse, IBM MQ **hata vermez** ve kalıcı iletiler de içinde olmak üzere *verileri işlemeye devam eder*. Kuyruk yöneticisi olağan bir şekilde kapanır, ancak yeniden başlatılamaz. Daha sonra mesajların kurtarılması imkansız hale gelir.

Etkin bir kuyruk yöneticisi tarafından kullanılan günlükleri kaldırma yetkisine sahip kullanıcılar, diğer önemli kuyruk yöneticisi kaynaklarını (kuyruk dosyaları, nesne kataloğu ve IBM MQ yürütülür dosyaları gibi) silme yetkisine de sahiptir. Bu nedenle, IBM MQ ' in kendisini koruyamayacağı bir şekilde, belki de deneyimsiz, çalışan ya da atıl bir kuyruk yöneticisine zarar verebilir.

Ayrıcalıklı kullanıcı ya da mqm yetkisi verilirken dikkatli olun.

## dmpmqlog komutunu kullanarak günlüğün içeriğinin dökümü

Kuyruk yöneticisi günlüğün içeriğinin dökümünü almak için dmpmqlog komutunu kullanma.

Kuyruk yöneticisi günlüğün içeriğinin dökümünü almak için dmpmqlog komutunu kullanın. Varsayılan olarak, tüm etkin günlük kayıtlarının dökümü dökümü verilir; yani, komutun dökümü günlüğün başından başlar (genellikle son tamamlanan denetim noktasının başlangıcı).

Günlük, genellikle yalnızca kuyruk yöneticisi çalışmadığında dökümü alabilir. Kuyruk yöneticisi sona erdirme sırasında bir denetim noktası aldığından, günlüğün etkin kısmı genellikle az sayıda günlük kaydı içerir. Ancak, dökümün başlangıç konumunu değiştirmek için aşağıdaki seçeneklerden birini kullanarak daha fazla günlük kaydı dökümü almak için dmpmqlog komutunu kullanabilirsiniz:

- Günlüğün *tabanından* döküm alma işlemini başlatın. Günlüğün temeli, günlüğün başını içeren günlük dosyasındaki ilk günlük kayıttır. Bu durumda dökümü alınan ek veri miktarı, günlük başkanının günlük dosyasında konumlandırıldığı yere bağlıdır. Günlük dosyasının başlangıca yakınsa, yalnızca az miktarda ek veri dökümü olur. Kafa, günlük dosyasının sonuna yakınsa, önemli ölçüde daha fazla veri dökümü olur.
- Dökümün başlangıç konumunu tek bir günlük kaydı olarak belirtin. Her günlük kaydı, benzersiz bir *günlük sıra numarası (LSN)* ile tanımlanır. Çevrimsel günlüğe kaydetme durumunda, bu başlangıç günlüğü kaydı günlüğün tabanından önce olamaz; bu kısıtlama doğrusal günlükler için geçerli değildir. Komutu çalıştırmadan önce etkin olmayan günlük dosyalarını yeniden çalıştırmamız gerekebilir. Başlangıç konumu olarak önceki dmpmqlog çıkışından alınan geçerli bir LSN belirtmelisiniz.

Örneğin, doğrusal günlük kaydı ile son dmpmqlog çıkışınızdan nextlsn belirtebilirsiniz. nextlsn , Log File Header içinde görünür ve yazılacak sonraki günlük kaydının LSN 'sini gösterir. Günlüğün son dökümü aldığından bu yana yazılan tüm günlük kayıtlarını biçimlendirmek için başlangıç konumu olarak bunu kullanın.

- **Yalnızca doğrusal günlükler için**, dmpmqlog günlük kayıtlarını belirli bir günlük dosyası kapsamına göre biçimlendirmeye başlamasını isteyebilirsiniz. Bu durumda dmpmqlog , bu günlük dosyasını ve ardışık her birini etkin günlük dosyalarıyla aynı dizinde bulmayı bekler. Bu seçenek çevrimsel günlükler için geçerli değildir; burada dmpmqlog , günlüğün tabanından önceki günlük kayıtlarına erişemez.

dmpmqlog komutunun çıkışı Log File Header ve biçimlenmiş günlük kayıtları serisidir. Kuyruk yöneticisi, verilerinde yapılan değişiklikleri kaydetmek için birkaç günlük kaydı kullanır.

Biçimlendirilen bilgilerin bazıları yalnızca dahili olarak kullanılır. Aşağıdaki liste en kullanışlı günlük kayıtlarını içerir:

### Günlük Dosyası Üstbilgisi

Her günlük tek bir günlük dosyası üstbilgisine sahiptir; bu, her zaman dmpmqlog komutuyla biçimlendirilen ilk şeydir. Aşağıdaki alanları içerir:

<i>günlük etkin</i>	Birincil günlük kapsamları sayısı.
<i>oturum açma etkin</i>	İkincil günlük kapsamları sayısı.
<i>günlük büyüklüğü</i>	Kapsam başına 4 KB sayfa sayısı.
<i>temel çizgiler</i>	Günlük kapsamı içindeki ilk LSN, günlüğün başını içerir.
<i>nextlsn</i>	Yazılacak sonraki günlük kaydının LSN 'si.
<i>başlıklar</i>	Günlüğün başındaki günlük kaydının LSN 'si.
<i>uyarlama</i>	Günlüğün kuyruk konumunu tanımlayan LSN.
<i>hflag1</i>	Günlüğün CIRCULAR mı, yoksa LOG RETAIN (linear) mi olduğunu belirtir.
<i>HeadExtentTanıtıcısı</i>	Günlüğün başını içeren günlük kapsamı.

### Günlük Kaydı Üstbilgisi

Günlük içindeki her günlük kaydının aşağıdaki bilgileri içeren sabit bir üstbilgisi vardır:



<i>LSN</i>	Günlük sıra numarası.
<i>LogRecdTip</i>	Günlük kaydının tipi.
<i>XTranid</i>	Bu günlük kaydıyla ilişkili işlem tanıtıcısı (varsa). MQI <i>TranType</i> , yalnızca IBM MQhareketini gösterir. XA 'nın <i>TranType</i> ' i diğer kaynak yöneticileriyle birlikte yer aldı. Aynı iş biriminde yer alan güncellemeler aynı <i>XTranid</i> .
<i>QueueName</i>	Bu günlük kaydıyla ilişkili kuyruk (varsa).
<i>Qid</i>	Kuyruğa ilişkin benzersiz iç tanıtıcı.
<i>PrevLSN</i>	Aynı işlem içinde (varsa) önceki günlük kaydının LSN 'si.

### **Kuyruk Yöneticisini Başlat**

Bu, kuyruk yöneticisinin başlattığını günlüğe kaydeder.

<i>StartDate</i>	Kuyruk yöneticisinin başlatıldığı tarih.
<i>StartTime</i>	Kuyruk yöneticisinin başlatıldığı saat.

### **Kuyruk Yöneticisini Durdur**

Bu, kuyruk yöneticisinin durdurduğunu günlüğe kaydeder.

<i>StopDate</i>	Kuyruk yöneticisinin durduğu tarih.
<i>StopTime</i>	Kuyruk yöneticisinin durdurulduğu saat.
<i>ForceFlag</i>	Kullanılan kapatma tipi.

### **Denetim Noktasını Başlat**

Bu, bir kuyruk yöneticisi denetim noktasının başlangıcını belirtir.

### **Bitiş Denetim Noktası**

Bu, kuyruk yöneticisi denetim noktasının sonunu gösterir.

<i>ChkPtLSN</i>	Bu denetim noktasını başlatan günlük kaydının LSN 'si.
-----------------	--

### **Koyma İletisi**

Bu, kuyruğa konan kalıcı bir iletiyi günlüğe kaydeder. İleti eşitleme noktası altına konduysa, günlük kaydı üstbilgisi boş olmayan bir *XTranid* içerir. Kaydın geri kalanı şunları içerir:

<i>MapIndex</i>	Kuyruktaki iletiye ilişkin tanıtıcı. Bu iletiyi kuyruktan almak için kullanılan karşılık gelen MQGET ile eşleşmek için kullanılabilir. Bu durumda, aynı <i>QueueName</i> ve <i>MapIndex</i> ögesini içeren sonraki bir <i>Get Message</i> günlük kaydı bulunabilir. Bu noktada, <i>MapIndex</i> tanıtıcısı sonraki bir koyma ileti için o kuyruğa yeniden kullanılabilir.
<i>Veriler</i>	Bu günlük kaydına ilişkin onaltılı dökümde çeşitli iç veriler bulunur; bunu İleti Tanımlayıcısı (eyecatcher MD) ve ileti verilerinin bir gösterimi izler.

### **Parça Koy**

Tek bir günlük kaydı için çok büyük olan kalıcı iletiler, tek bir *Put Message* kaydının izlediği birden çok *Put Part* günlük kaydı olarak günlüğe kaydedilir. *Put Part* kayıtları varsa, *PrevLSN* alanı *Put Part* kayıtlarını ve son *Put Message* kaydını birbirine zincirler.

<i>Veriler</i>	Önceki günlük kaydının bırakıldığı yerde ileti verilerini devam ettirir.
----------------	--

### **İleti Al**

Yalnızca kalıcı iletilerin alımlarını günlüğe kaydeder. İleti eşitleme noktası altında alındıysa, günlük kaydı üstbilgisi boş olmayan bir *XTranid* içerir. Kaydın geri kalanı şunları içerir:

<i>MapIndex</i>	Kuyruktan alınan iletiyi tanımlar. Aynı <i>QueueName</i> ve <i>MapIndex</i> içeren en son <i>Put Message</i> günlük kaydı, alınan iletiyi tanımlar.
<i>Kalite Önceliği</i>	Kuyruktan alınan iletinin önceliği.

### İşlemi Başlat

Yeni bir hareketin başlangıcını gösterir. MQI TranType , yalnızca IBM MQhareketini gösterir. TranType (Geçiş Tipi), diğer kaynak yöneticilerini içeren bir XA ' yı gösterir. Bu işlem tarafından yapılan tüm güncellemeler aynı *XTranid* olacaktır.

### İşlemi Hazırla

Kuyruk yöneticisinin, belirtilen *XTranid* ile ilişkili güncellemeleri kesinleştirmeye hazır olduğunu gösterir. Bu günlük kaydı, diğer kaynak yöneticilerini içeren iki aşamalı kesinleştirmenin bir parçası olarak yazılır.

### İşlemi Kesinleştir

Kuyruk yöneticisinin bir hareket tarafından yapılan tüm güncellemeleri kesinleştirdiğini gösterir.

### İşlemi Geri Döndür

Bu, kuyruk yöneticisinin bir hareketi geriye işlemeyi amaçladığını gösterir.

### İşlemi Sona Erdir

Bu, geriye işlenen bir hareketin sonunu gösterir.

### İşlem Tablosu

Bu kayıt, eşitleme noktası sırasında yazılır. Kalıcı güncellemeler yapan her işlemin durumunu kaydeder. Her işlem için aşağıdaki bilgiler kaydedilir:

<i>XTranid</i>	İşlem tanıtıcısı.
<i>FirstLSN</i>	İşlemlerle ilişkili ilk günlük kaydının LSN 'si.
<i>LastLSN</i>	İşlemlerle ilişkili son günlük kaydının LSN 'si.

### İşlem Katılımcıları

Bu günlük kaydı, kuyruk yöneticisinin XA Transaction Manager bileşeni tarafından yazılır. İşlemlere katılan dış kaynak yöneticilerini kaydeder. Her katılımcı için aşağıdaki öğeler kaydedilir:

<i>RMName</i>	Kaynak yöneticisinin adı.
<i>RMID</i>	Kaynak yöneticisi tanıtıcısı. Bu, kaynak yöneticisinin katılmakta olduğu genel hareketleri kaydeden sonraki <i>Transaction Prepared</i> günlük kayıtlarına da kaydedilir.
<i>SwitchFile</i>	Bu kaynak yöneticisine ilişkin anahtar yükleme dosyası.
<i>XAOpenString</i>	Bu kaynak yöneticisine ilişkin XA açık dizgisi.
<i>XACloseString</i>	Bu kaynak yöneticisine ilişkin XA kapatma dizgisi.

### Hareket Hazırlandı

Bu günlük kaydı, kuyruk yöneticisinin XA Transaction Manager bileşeni tarafından yazılır. Belirtilen genel hareketin başarıyla hazırlandığını gösterir. Katılan kaynak yöneticilerinin her birine kesinleştirme talimatı verilir. Hazırlanan her kaynak yöneticisinin *RMID* kaydı günlük kaydına kaydedilir. Kuyruk yöneticisinin kendisi de harekete katılıyorsa, sıfır *RMID* değerine sahip bir *Participant Entry* vardır.

### İşlem Unut

Bu günlük kaydı, kuyruk yöneticisinin XA Transaction Manager bileşeni tarafından yazılır. Kesinleştirme kararı her katılımcıya teslim edildiğinde *Transaction Prepared* günlük kaydını izler.

### Kuyruğu Temizle

Bu, kuyruktaki tüm iletilerin (örneğin, MQSC komutu CLEAR QUEUE kullanılarak) temizlendiğini günlüğe kaydeder.

### Kuyruk Öznitelikleri

Bu, bir kuyruğun özniteliklerinin kullanıma hazırlanmasını ya da değiştirilmesini günlüğe kaydeder.

### Nesne Oluştur

Bu, bir IBM MQ nesnesinin oluşturulmasını günlüğe kaydeder.

<i>ObjName</i>	Yaratılan nesnenin adı.
<i>UserId</i>	Yaratma işlemini gerçekleştiren kullanıcı kimliği.

### Nesneyi Sil

Bu, bir IBM MQ nesnesinin silinmesini günlüğe kaydeder.

<i>ObjName</i>	Silinen nesnenin adı.
----------------	-----------------------

## IBM MQ kuyruk yöneticisi verilerinin yedeklenmesi ve geri yüklenmesi

Kuyruk yöneticilerini, kuyruk yöneticilerini ve kuyruk yöneticisi verilerini yedekleyerek, yalnızca kuyruk yöneticisi yapılandırmasını yedekleyerek ve bir yedek kuyruk yöneticisini kullanarak, donanım hatalarından kaynaklanan olası bozulmaya karşı koruyabilirsiniz.

### Bu görev hakkında



**DİKKAT:** Bir kuyruk yöneticisini farklı bir işletim sistemine taşırsanız çok dikkatli olmanız gerekir. Ek bilgi için [Kuyruk yöneticisinin farklı bir işletim sistemine taşınması](#) başlıklı konuya bakın.

Düzenli aralıklarla, kuyruk yöneticilerini donanım arızalarının neden olduğu olası bozulmaya karşı korumak için önlemler alabilirsiniz. Bir kuyruk yöneticisini korumanın üç yolu vardır:

#### Kuyruk yöneticisi verilerini yedekle

Donanım başarısız olursa, bir kuyruk yöneticisi durmaya zorlanabilir. Donanım arızası nedeniyle herhangi bir kuyruk yöneticisi günlük verileri kaybolursa, kuyruk yöneticisi yeniden başlatılamaz. Kuyruk yöneticisi verilerini yedeklerse, kayıp kuyruk yöneticisi verilerinin bazılarını ya da tümünü kurtarabilirsiniz.

Genel olarak, kuyruk yöneticisi verilerini ne kadar sık yedeklerseniz, kurtarma günlüğünün bütünlüğünün kaybolmasına neden olan donanım arızası durumunda o kadar az veri kaybedersiniz.

Kuyruk yöneticisi verilerini yedeklemek için kuyruk yöneticisinin çalışmaması gerekir.

#### Yalnızca kuyruk yöneticisi yapılandırmasını yedekle

Donanım başarısız olursa, bir kuyruk yöneticisi durmaya zorlanabilir. Hem kuyruk yöneticisi yapılandırması, hem de günlük verileri donanım hatası nedeniyle kaybolursa, kuyruk yöneticisi yeniden başlatılamaz ya da günlükten kurtarılamaz. Kuyruk yöneticisi yapılandırmasını yedeklerse, kuyruk yöneticisini ve tüm nesnelerini saklanmış tanımlamalardan yeniden yaratabilirsiniz.

Kuyruk yöneticisi yapılandırmasını yedeklemek için kuyruk yöneticisinin çalışıyor olması gerekir.

#### Yedek kuyruk yöneticisi kullan

Donanım arızası önemli ise, kuyruk yöneticisi kurtarılamaz. Bu durumda, kurtarılamaz kuyruk yöneticisinin özel olarak ayrılmış bir yedek kuyruk yöneticisi varsa, yedek kuyruk yöneticisi kurtarılamaz kuyruk yöneticisi yerine etkinleştirilebilir. Düzenli olarak güncellenirse, yedek kuyruk yöneticisi günlüğü, kurtarılamayan kuyruk yöneticisinden alınan son tam günlüğü içeren günlük verilerini içerebilir.

Varolan kuyruk yöneticisi çalışırken yedek kuyruk yöneticisi güncellenebilir.

### Yordam

- Kuyruk yöneticisi verilerini yedeklemek ve geri yüklemek için aşağıdaki adımları izleyin:
  - [“Kuyruk yöneticisi verileri yedekleniyor” sayfa 656.](#)
  - [“Kuyruk yöneticisi verilerinin geri yüklenmesi” sayfa 657.](#)
- Kuyruk yöneticisi yapılandırmasını yedeklemek ve geri yüklemek için aşağıdaki adımları izleyin:
  - [“Kuyruk yöneticisi yapılandırması yedekleniyor” sayfa 657](#)

– “Kuyruk yöneticisi yapılanışı geri yükleniyor” sayfa 658

- Bir yedek kuyruk yöneticisi yaratmak, güncellemek ve başlatmak için bkz. “Yedek kuyruk yöneticisinin kullanılması” sayfa 659.

## Kuyruk yöneticisi verileri yedekleniyor

Kuyruk yöneticisi verilerinin yedeklenmesi, donanım hatalarından kaynaklanan olası veri kaybına karşı korunmanıza yardımcı olabilir.

### Başlamadan önce

Kuyruk yöneticisini yedeklemeye başlamadan önce, kuyruk yöneticisinin çalışmadığından emin olun. Çalışmakta olan bir kuyruk yöneticisinin yedeğini almaya çalışırsanız, dosyalar kopyalandığında devam eden güncellemeler nedeniyle yedekleme tutarlı olmayabilir. Olanaklıysa, **endmqm -w** komutunu (bekleme sona erdirme) çalıştırarak kuyruk yöneticinizi durdurun; bu işlem başarısız olursa **endmqm -i** komutunu kullanın (hemen sona erdirme).

### Bu görev hakkında

Bir kuyruk yöneticisi verilerinin yedek kopyasını almak için aşağıdaki görevleri tamamlayın:

#### Yordam

1. Yapılanış dosyalarındaki bilgileri kullanarak, kuyruk yöneticisinin verilerini ve günlük dosyalarını yerleştirdiği dizinleri arayın.

Daha fazla bilgi için bkz. “Çoklu platformlarda .ini dosyalarında IBM MQ yapılandırma bilgilerini değiştirme” sayfa 83.

**Not:** Dizinde görüntülenen adlar, IBM MQ kullanmakta olduğunuz platformla uyumlu olduklarından emin olmak için dönüştürülür. Ad dönüşümleriyle ilgili ek bilgi için [IBM MQ dosya adlarını anlamabaşlıklı konuya](#) bakın.


2. Tüm alt dizinler de içinde olmak üzere, kuyruk yöneticisinin tüm veri ve günlük dosyası dizinlerinin kopyalarını alın.

“Günlüklerin nasıl görüldüğünü” sayfa 627 başlıklı konuda açıklandığı gibi herhangi bir dosyayı, özellikle de günlük denetim dosyasını ve yapılandırma dosyalarını gözden kaçırmadığınızdan emin olun (bkz. “Başlatma ve yapılandırma dosyaları” sayfa 235). Bazı dizinler boş olabilir, ancak daha sonraki bir tarihte yedeklemeyi geri yüklemek için bu dizinlerin tümüne gereksinim duyarsınız.

Çevrimsel günlük kaydı için, tutarlı bir kuyruk yöneticisi verileri ve günlükleri kümesini geri yükleyebilmek için kuyruk yöneticisi verilerini ve günlük dosyası dizinlerini aynı anda yedekleyin.

Doğrusal günlük kaydı için, kuyruk yöneticisi verilerini ve günlük dosyası dizinlerini aynı anda yedekleyin. Günlük dosyalarının karşılık gelen tam sırası varsa, yalnızca kuyruk yöneticisi veri dosyaları geri yüklenebilir.

3. Dosyaların sahiplerini koruyun.

 IBM MQ for UNIX ve Linux sistemleri için bunu **tar** komutuyla yapabilirsiniz. (2 GB ' den büyük kuyrukunuz varsa, **tar** komutunu kullanamazsınız. Ek bilgi için [Büyük kuyrukların etkinleştirilmesi](#) başlıklı konuya bakın.

**Not:** IBM WebSphere MQ 7.5 ve sonrasına büyütme yaptığınızda, **qm .ini** dosyasının ve kayıt girişlerinin bir yedeğini aldığınızdan emin olun. Kuyruk yöneticisi bilgileri **qm .ini** dosyasında saklanır ve önceki bir IBM MQ sürümüne geri dönmek için kullanılabilir.

#### İlgili görevler

[Kuyruk yöneticisinin durdurulması](#)

[“Kuyruk yöneticisi yaratıldıktan sonra yapılanış kütüklerinin yedeklenmesi” sayfa 14](#)

[IBM MQ yapılandırma bilgileri, AIX, Linux, and Windows üzerindeki yapılandırma dosyalarında depolanır.](#) Bir kuyruk yöneticisi yarattıktan sonra, yapılanış dosyalarınızı yedekleyin. Daha sonra, sorunlara neden

olan başka bir kuyruk yöneticisi oluşturursanız, sorunun kaynağını kaldırdığınızda yedekleri yeniden yürürlüğe alabilirsiniz.

## Kuyruk yöneticisi verilerinin geri yüklenmesi

Bir kuyruk yöneticisi verilerinin yedeğini geri yüklemek için aşağıdaki adımları izleyin.

### Başlamadan önce

Yedeklemeyi başlatmadan önce kuyruk yöneticisinin çalışmadığından emin olun.

Bir kümedeki kuyruk yöneticisinin yedeğini geri yüklerken, daha fazla bilgi için bkz. “Küme kuyruk yöneticisinin kurtarılması” sayfa 359 ve [Clustering: Availability, multi-instance, and disaster recovery](#).

**Not:** Daha sonraki bir IBM MQ sürümüne yükseltirken, **.ini** dosyasının ve kayıt girişlerinin bir yedeğini aldığınızdan emin olun. Kuyruk yöneticisi bilgileri **.ini** dosyasında saklanır ve önceki bir IBM MQ sürümüne geri dönmek için kullanılabilir.

### Yordam

1. Yapılanış dosyalarındaki bilgileri kullanarak, kuyruk yöneticisinin verilerini ve günlük dosyalarını yerleştirdiği dizinleri bulun.
2. Yedeklenen verileri koyacağınız dizinleri boşaltın.
3. Yedeklenen kuyruk yöneticisi verilerini ve günlük dosyalarını doğru yerlere kopyalayın.  
Günlük dosyalarının yanı sıra bir günlük denetim dosyasına sahip olduğunuzdan emin olun.  
Çevrimsel günlük kaydı için, tutarlı bir kuyruk yöneticisi verileri ve günlükleri kümesini geri yükleyebilmek için kuyruk yöneticisi verilerini ve günlük dosyası dizinlerini aynı anda yedekleyin.  
Doğrusal günlük kaydı için, kuyruk yöneticisi verilerini ve günlük dosyası dizinlerini aynı anda yedekleyin. Günlük dosyalarının karşılık gelen tam sırası varsa, yalnızca kuyruk yöneticisi veri dosyaları geri yüklenebilir.
4. Yapılandırma bilgileri dosyalarını güncelleyin.  
IBM MQ ' in geri yüklenen verileri doğru yerlerde arayabilmesi için IBM MQ ve kuyruk yöneticisi yapılanış kütüklerinin tutarlı olup olmadığını denetleyin.
5. Tüm gerekli dizinlere sahip olduğunuzdan emin olmak için sonuçtaki dizin yapısını denetleyin.  
IBM MQ dizinlerine ve alt dizinlerine ilişkin ek bilgi için [Windows sistemlerinde dizin yapısı](#) ve [AIX and Linux sistemlerinde dizin içeriği](#) başlıklı konuya bakın.

### Sonuçlar

Veriler doğru bir şekilde yedeklenip geri yüklendiyse, kuyruk yöneticisi başlatılır.

Multi

## Kuyruk yöneticisi yapılanışı yedekleniyor

Kuyruk yöneticisi yapılanışının yedeklenmesi, hem kuyruk yöneticisi yapılanışı, hem de günlük verileri donanım hatası nedeniyle kaybolursa ve kuyruk yöneticisi yeniden başlatılamazsa ya da günlükten kurtarılamazsa, kuyruk yöneticisini tanımlamalarından yeniden oluşturmanıza yardımcı olabilir.

### Bu görev hakkında

ALW

AIX, Linux, and Windowssistemlerinde, bir IBM MQ kuyruk yöneticisinin yapılandırmasının dökümünü almak için **dmpmqcfg** komutunu kullanabilirsiniz.

IBM i

IBM üzerinde, bir kuyruk yöneticisine ilişkin yapılandırma nesnelерinin ve yetkilerinin dökümünü almak için Dump MQ Configuration (**DMPMQCFG**) komutunu kullanabilirsiniz.

## Yordam

1. Kuyruk yöneticisinin çalıştığından emin olun.
2. Altyapınıza bağlı olarak, kuyruk yöneticisi yapısını yedeklemek için aşağıdaki komutlardan birini kullanın:

- **ALW** AIX, Linux, and Windows üzerinde: (-f mqsc) MQSC 'nin varsayılan biçimlendirme seçeneğini ve tüm öznitelikleri (-a) kullanarak MQ Configuration komutunu yürütün. **dmpmqcfig**, tanımlamaları bir dosyada saklamak için standart çıkış yeniden yönlendirmesini kullanın. Örneğin:

```
dmpmqcfig -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
```

- **IBM i** IBM üzerinde: OUTPUT (\*MQSC) ve EXPATTR (\*ALL) varsayılan biçimlendirme seçeneğini kullanarak MQ Configuration komutunu (**DMPMQMCFG**) yürütün, tanımları fiziksel bir dosya üyesinde saklamak için TOFILE ve TOMBR kullanın. Örneğin:

```
DMPMQMCFG MQMNAME(MYQMGR) OUTPUT(*MQSC) EXPATTR(*ALL) TOFILE(QMQMSAMP/QMQSC)  
TOMBR(MYQMGRDEF)
```

## İlgili görevler

[“Kuyruk yöneticisi yapısını geri yükleniyor” sayfa 658](#)

Önce kuyruk yöneticisinin çalıştığından emin olarak ve altyapınız için uygun komutu çalıştırarak, kuyruk yöneticisinin yapısını bir yedekten geri yükleyebilirsiniz.

## İlgili başvurular

[dmpmqcfig \(döküm kuyruğu yöneticisi yapılandırması\)](#)

[Döküm MQ Yapısını \(DMPMQMCFG\)](#)

**Multi**

## Kuyruk yöneticisi yapısını geri yükleniyor

Önce kuyruk yöneticisinin çalıştığından emin olarak ve altyapınız için uygun komutu çalıştırarak, kuyruk yöneticisinin yapısını bir yedekten geri yükleyebilirsiniz.

## Bu görev hakkında

**ALW** AIX, Linux, and Windows işletim sistemlerinde, bir IBM MQ kuyruk yöneticisinin yapısını geri yüklemek için **runmqsc** komutunu kullanabilirsiniz.

**IBM i** IBM i' da, bir kuyruk yöneticisine ilişkin yapılandırma nesnelerini ve yetkileri geri yüklemek için **STRMQMMSQC** komutunu kullanabilirsiniz.

## Yordam

1. Kuyruk yöneticisinin çalıştığından emin olun.  
Veriler ve günlükler başka yollarla kurtarılamazsa, kuyruk yöneticisinin yeniden yaratılmış olabileceğini unutmayın.
2. Altyapınıza bağlı olarak, kuyruk yöneticisi yapısını geri yüklemek için aşağıdaki komutlardan birini kullanın:

- **ALW** AIX, Linux, and Windows üzerinde, kuyruk yöneticisine karşı **runmqsc** komutunu çalıştırın, Dump MQ Configuration (**dmpmqcfig**) komutu tarafından oluşturulan bir komut dosyasından tanımlamaları geri yüklemek için standart giriş yeniden yönlendirmesini kullanın (bkz. [“Kuyruk yöneticisi yapısını yedekleniyor” sayfa 657](#)). Örneğin:

```
runmqsc MYQMGR < /mq/backups/MYQMGR.mqsc
```

- **IBM i** IBM üzerinde: Kuyruk yöneticisi için **STRMQMMQSC** komutunu çalıştırın ve Dump MQ Configuration (**DMPMQMCFG**) komutu tarafından oluşturulan fiziksel dosya üyesinden tanımlamaları geri yüklemek için **SRCMBR** ve **SRCFILE** parametrelerini kullanın (bkz. [“Kuyruk yöneticisi yapısını yedekleniyor”](#) sayfa 657). Örneğin:

```
STRMQMMQSC MQMNAME(MYQMGR) SRCFILE(QMQMSAMP/QMQSC) SRCMBR(MYQMGR)
```

### İlgili görevler

[“Kuyruk yöneticisi yapısını yedekleniyor”](#) sayfa 657

Kuyruk yöneticisi yapısının yedeklenmesi, hem kuyruk yöneticisi yapısını, hem de günlük verileri donanım hatası nedeniyle kaybolursa ve kuyruk yöneticisi yeniden başlatılamazsa ya da günlükten kurtarılamazsa, kuyruk yöneticisini tanımlamalarından yeniden oluşturmanıza yardımcı olabilir.

### İlgili başvurular

[dmpmqcfg \(döküm kuyruğu yöneticisi yapılandırması\)](#)

[runmqsc \(MQSC komutlarını çalıştır\)](#)

[Döküm MQ Yapısını \(DMPMQMCFG\)](#)

[IBM MQ Komutlarını Başlat \(STRMQMMQSC\)](#)

## Yedek kuyruk yöneticisinin kullanılması

Var olan bir kuyruk yöneticisinin olağanüstü durumdan kurtarma amacıyla özel olarak ayrılmış bir yedek kuyruk yöneticisi olabilir.

### Bu görev hakkında

Yedek kuyruk yöneticisi, var olan kuyruk yöneticisinin etkin olmayan bir kopyasıdır. Önemli donanım arızası nedeniyle var olan kuyruk yöneticisi kurtarılamaz duruma gelirse, yedek kuyruk yöneticisi kurtarılamayan kuyruk yöneticisini değiştirmek için çevrimiçi duruma getirilebilir.

Yedek kuyruk yöneticisinin olağanüstü durumdan kurtarma için etkin bir yöntem olarak kaldığından emin olmak için, var olan kuyruk yöneticisi günlük dosyalarının düzenli olarak yedek kuyruk yöneticisine kopyalanması gerekir. Günlük dosyalarının kopyalanması için var olan kuyruk yöneticisinin durdurulması gerekmez; ancak, kuyruk yöneticisi günlük dosyasına yazmayı tamamladıysa, günlük dosyasını kopyalamanız gerekir; belirli bir günlük dosyasına artık yazılmamasını nasıl sağlayacağınıza ilişkin bilgi için bkz. [“Yedek kuyruk yöneticisinin güncellenmesi”](#) sayfa 660 .

**Not:** Var olan kuyruk yöneticisi günlüğü sürekli olarak güncellendiğinden, var olan kuyruk yöneticisi günlüğü ile yedek kuyruk yöneticisi günlüğüne kopyalanan günlük verileri arasında her zaman küçük bir tutarsızlık vardır. Yedek kuyruk yöneticisinde yapılan olağan güncellemeler, iki günlük arasındaki tutarsızlığı en aza indirir.

Bir yedek kuyruk yöneticisinin çevrimiçi duruma getirilmesi gerekiyorsa, etkinleştirilmeli ve başlatılmalıdır. Başlatılmadan önce bir yedek kuyruk yöneticisini etkinleştirme gereksinimi, yanlışlıkla başlatılan bir yedekleme kuyruk yöneticisine karşı koruma amaçlı bir önleyici önlemdir. Bir yedek kuyruk yöneticisi etkinleştirildikten sonra artık güncellenemez.

**Önemli:** Eski yedek kuyruk yöneticisi yeni etkin kuyruk yöneticisi olduktan sonra, nedeni ne olursa olsun, artık bir yedek kuyruk yöneticisi yoktur. Bu, zamanuyumsuz kopyalamanın etkili bir biçimidir ve yeni etkin kuyruk yöneticisinin, mantıksal olarak eski etkin kuyruk yöneticisinin biraz gerisinde olması beklenir. Bu nedenle, eski etkin kuyruk yöneticisi artık yeni etkin kuyruk yöneticisinin yedeği olarak işlev görmez.

### Yordam

- Yedek kuyruk yöneticisini kullanma hakkında bilgi için aşağıdaki konulara bakın:
  - [“Yedek kuyruk yöneticisi yaratılması”](#) sayfa 660
  - [“Yedek kuyruk yöneticisinin güncellenmesi”](#) sayfa 660
  - [“Yedekleme kuyruğu yöneticisinin başlatılması”](#) sayfa 661

## İlgili kavramlar

“Günlüğe kaydetme: İletilerin kaybolmadığından emin olmak” sayfa 626

IBM MQ , kuyruk yöneticisi tarafından denetlenen kalıcı verilerdeki tüm önemli değişiklikleri bir kurtarma günlüğüne kaydeder.

## Yedek kuyruk yöneticisi yaratılması

Var olan kuyruk yöneticisinin etkin olmayan bir kopyası olarak bir yedek kuyruk yöneticisi yaratırsınız.

## Bu görev hakkında

**Önemli:** Bir yedek kuyruk yöneticisini yalnızca doğrusal günlük kaydı kullanırken kullanabilirsiniz.

Yedek kuyruk yöneticisi aşağıdakileri gerektirir:

- Kuyruk yöneticisi adı, günlük kaydı tipi ve günlük dosyası büyüklüğü gibi, var olan kuyruk yöneticisiyle aynı özniteliklere sahip olmak için.
- Var olan kuyruk yöneticisiyle aynı platformda olmak.
- Var olan kuyruk yöneticisine eşit ya da daha yüksek bir kod düzeyinde olmak.

## Yordam

1. `crtmqm`denetim komutunu kullanarak, varolan kuyruk yöneticisi için bir yedek kuyruk yöneticisi yaratın.
2. “Kuyruk yöneticisi verileri yedekleniyor” sayfa 656 konusunda açıklandığı gibi, tüm alt dizinler de içinde olmak üzere, var olan tüm kuyruk yöneticisi verilerinin ve günlük dosyası dizinlerinin kopyalarını alın.
3. Var olan kuyruk yöneticisinden alınan kopyalarla, tüm alt dizinler de içinde olmak üzere, yedek kuyruk yöneticisinin verilerinin ve günlük dosyası dizinlerinin üzerine yazın.
4. Aşağıdaki örnekte gösterildiği gibi, yedek kuyruk yöneticisinde `strmqm` denetim komutunu çalıştırın:

```
strmqm -r BackupQMName
```

Bu komut, kuyruk yöneticisini IBM MQ içinde yedek kuyruk yöneticisi olarak işaretler ve yedek kuyruk yöneticisini var olan kuyruk yöneticisine adım adım getirmek için kopyalanan tüm günlük kapsamlarını yeniden yürütür.

## İlgili başvurular

[crtmqm \(kuyruk yöneticisi yarat\)](#)

[strmqm \(kuyruk yöneticisini başlat\)](#)

## Yedek kuyruk yöneticisinin güncellenmesi

Bir yedek kuyruk yöneticisinin olağanüstü durumdan kurtarma için etkin bir yöntem olarak kalmasını sağlamak için düzenli olarak güncellenmesi gerekir.

## Bu görev hakkında

Düzenli güncelleme, yedek kuyruk yöneticisi günlüğü ile yürürlükteki kuyruk yöneticisi günlüğü arasındaki uyumsuzluğu azaltır. Yedeklemeden önce kuyruk yöneticisini durdurmanıza gerek yoktur.



**Uyarı:** Bitişik olmayan bir günlük kümesini yedek kuyruk yöneticisi günlük dizinine kopyalarsanız, yalnızca ilk eksik günlüğün bulunduğu noktaya kadar olan günlükler yeniden yürütülür.

## Yordam

1. Yedeklenecek kuyruk yöneticisinde şu Script (MQSC) komutunu verin:

```
RESET QMGR TYPE(ADVANCELOG)
```

Bu, yürürlükteki günlüğe yazmayı durdurur ve kuyruk yöneticisi günlük kaydını sonraki günlük kapsamına ilerler. Bu, geçerli saate kadar kaydedilen tüm bilgilerin yedeklenmesini sağlar.



2. Yedeklenecek kuyruk yöneticisinde aşağıdaki Script (MQSC) komutunu vererek (yeni) yürürlükteki etkin günlük kapsamı numarasını alın:

```
DIS QMSTATUS CURRLOG
```

3. Güncellenen günlük kapsamı dosyalarını yürürlükteki kuyruk yöneticisi günlük dizininden yedek kuyruk yöneticisi günlük dizinine kopyalayın.

Son güncellemeden bu yana tüm günlük kapsamlarını ve “2” sayfa 661 içinde belirtilen geçerli kapsama kadar (ancak dahil değil) tüm günlük kapsamlarını kopyalayın. Yalnızca "S ..." ile başlayan günlük kapsamı dosyalarını kopyalayın.

4. Aşağıdaki örnekte gösterildiği gibi, yedek kuyruk yöneticisinde **strmqm** denetim komutunu çalıştırın:

```
strmqm -r BackupQMName
```

Kopyalanan tüm günlük kapsamlarını yeniden yürür ve yedek kuyruk yöneticisini kuyruk yöneticisiyle aynı adıma getirir. Yeniden yürütme tamamlandığında, yeniden başlatma kurtarma için gerekli tüm günlük kapsamlarını ve ortam kurtarma için gerekli tüm günlük kapsamlarını tanımlayan bir ileti alırsınız.

### İlgili başvurular

QMGR ' YI SIFIR

QMSTATUS DURUMUNU GÖRÜNTÜLE

strmqm (kuyruk yöneticisini başlat)

### Yedekleme kuyruğu yöneticisinin başlatılması

Bir yedek kuyruk yöneticisini kurtarılamayan bir kuyruk yöneticisiyle değiştirebilirsiniz.

### Bu görev hakkında

Bir kümedeki kuyruk yöneticisinin yedeğini geri yüklerken, daha fazla bilgi için bkz. “Küme kuyruk yöneticisinin kurtarılması” sayfa 359 ve Clustering: Availability, multi-instance, and disaster recovery .

Kurtarılamayan bir kuyruk yöneticisinin adanmış bir yedek kuyruk yöneticisi varsa, kurtarılamayan kuyruk yöneticisi yerine yedek kuyruk yöneticisini etkinleştirebilirsiniz.

Kurtarılamayan bir kuyruk yöneticisinin yerine bir yedek kuyruk yöneticisi konduğunda, kurtarılamayan kuyruk yöneticisindeki bazı kuyruk yöneticisi verileri kaybolabilir. Kaybolan veri miktarı, yedek kuyruk yöneticisinin en son ne zaman güncellendiğine bağlıdır. Son güncelleme ne kadar yakın olursa, kuyruk yöneticisi veri kaybı o kadar azalır.

**Not:** Kuyruk yöneticisi verileri ve günlük dosyaları farklı dizinlerde tutulsa da, dizinleri aynı anda yedekleyip geri yüklediğinizden emin olun. Kuyruk yöneticisi verileri ve günlük dosyaları farklı yaşlara sahipse, kuyruk yöneticisi geçerli bir durumda değildir ve büyük olasılıkla başlamayacaktır. Başlasa bile, verileriniz bozulmuş olabilir.

### Yordam

1. Aşağıdaki örnekte gösterildiği gibi yedek kuyruk yöneticisini etkinleştirmek için **strmqm** denetim komutunu çalıştırın:

```
strmqm -a BackupQMName
```

Yedek kuyruk yöneticisi etkinleştirildi. Artık etkin olduğu için, yedek kuyruk yöneticisi artık güncellenemiyor.

2. Aşağıdaki örnekte gösterildiği gibi yedek kuyruk yöneticisini başlatmak için **strmqm** denetim komutunu çalıştırın:

```
strmqm BackupQMName
```

IBM MQ , bunu yeniden başlatma kurtarması olarak kabul eder ve yedek kuyruk yöneticisindeki günlüğü kullanır. Yedek kuyruk yöneticisine yapılan son güncelleme sırasında, yeniden yürütme gerçekleştiği için yalnızca son kaydedilen denetim noktasındaki etkin hareketler geriye işlenir.

3. Tüm kanalları yeniden başlatın.

4. Tüm gerekli dizinlere sahip olduğunuzdan emin olmak için sonuçtaki dizin yapısını denetleyin.

IBM MQ dizinleri ve alt dizinleriyle ilgili ek bilgi için [Planning file system support](#) başlıklı konuya bakın.

5. Günlük dosyalarının yanı sıra bir günlük denetim dosyasına sahip olduğunuzdan emin olun. Ayrıca, IBM MQ ve kuyruk yöneticisi yapıları kütüklerinin tutarlı olup olmadığını denetleyin; böylece, IBM MQ geri yüklenen veriler için doğru yerlere bakabilir.

## Sonuçlar

Veriler yedeklenip doğru olarak geri yüklendiye, kuyruk yöneticisi şimdi başlatılır.

### İlgili görevler

“Durdurulan kanallar yeniden başlatılıyor” sayfa 228

Bir kanal DURDURULDU durumuna geçtiğinde, kanalı el ile yeniden başlatmanız gerekir.

### İlgili başvurular

[strmqm \(kuyruk yöneticisini başlat\)](#)

## Küme hatası kurtarmada yapılan değişiklikler ( z/OS dışındaki sunucularda)

Kuyruk yöneticisi, sorunlar çözülmüncye kadar sorunlara neden olan işlemleri yeniden çalıştırır. Beş gün sonra sorunlar çözülmüncye, kuyruk yöneticisi önbelleğin daha eski olmasını önlemek için kapanır.

Kuyruk yöneticisi, sorunlar çözülmüncye kadar sorunlara neden olan işlemleri yeniden çalıştırır. Beş gün sonra sorunlar çözülmüncye, kuyruk yöneticisi önbelleğin daha eski olmasını önlemek için kapanır. Önbellek daha eski hale geldikçe daha fazla soruna neden olur. Küme hatalarıyla ilgili bu davranış z/OS için geçerli değildir.

Küme yönetiminin her yönü, bir kuyruk yöneticisi için yerel havuz yöneticisi işlemi ( amqrmf) tarafından işlenir. Herhangi bir küme tanımlaması olmasa da, işlem tüm kuyruk yöneticileriyle çalışır.

Havuz yöneticisini durdurmak ve havuz yöneticisi olmadan devam etmektense IBM MQ, başarısız olan işlemleri yeniden çalıştırır. Kuyruk yöneticisi, havuz yöneticisiyle ilgili bir sorun saptarsa, iki işlem kursundan birini izler.

1. Hata, kuyruk yöneticisinin işlemini tehlikeye atmazsa, kuyruk yöneticisi hata günlüğüne bir ileti yazar. İşlem başarılı oluncaya kadar her 10 dakikada bir başarısız işlemi yeniden çalıştırır. Varsayılan olarak, hatayla başa çıkmak için beş gününüz vardır; başarısız olursa, kuyruk yöneticisi hata günlüğüne bir ileti yazar ve kapanır. Beş günlük kapatmayı erteleyebilirsiniz.
2. Hata kuyruk yöneticisinin işlemini tehlikeye atarsa, kuyruk yöneticisi hata günlüğüne bir ileti yazar ve hemen sona erer.

Kuyruk yöneticisinin işlemini tehlikeye alan bir hata, kuyruk yöneticisinin tanılamadığı bir hata ya da öngörülemez sonuçları olabilecek bir hata. Bu tip hatalar genellikle kuyruk yöneticisinin bir FFST dosyası yazması ile sonuçlanır. Kuyruk yöneticisinin işlemini tehlikeye atacak hatalar, IBM MQ içindeki bir hatadan ya da bir sistem yöneticisinden ya da bir programdan kaynaklanıyor olabilir; bu hata, IBM MQ işleminin sona erdirilmesi gibi beklenmeyen bir şey yapıyor olabilir.

Hata kurtarma davranışındaki değişikliğin amacı, kuyruk yöneticisinin giderek artan sayıda tutarsız küme tanımlamasıyla çalışmaya devam ettiği süreyi sınırlamaktır. Küme tanımlarındaki tutarsızlıkların sayısı arttıkça, anormal uygulama davranışı olasılığı da artar.

Kuyruk yöneticisini beş gün sonra kapatmanın varsayılan seçeneği, tutarsızlıkların sayısının sınırlanması ve sorunlar saptanıp çözülmüncye kadar kuyruk yöneticisinin kullanılabilir durumda tutulması arasında bir uzlaşmadır.

Kuyruk yöneticisini süresiz olarak kapatmadan önce, sorunu düzeltirken ya da planlanan bir kuyruk yöneticisinin sona ermesini beklerken bu süreyi uzatabilirsiniz. Beş günlük kalış, kuyruk yöneticisinin uzun

bir hafta sonu boyunca çalışmasını sağlar ve herhangi bir soruna yanıt vermeniz ya da kuyruk yöneticisini yeniden başlatmadan önce süreyi uzatmanız için size zaman verir.

## Düzeltilici işlemler

Küme hatası kurtarma sorunlarıyla ilgilenmek için bir işlem seçeneğiniz vardır. Birinci seçenek, sorunu izlemek ve düzeltmek, ikincisi ise sorunu izlemek ve düzeltmek.

1. AMQ9448 ve AMQ5008 hata iletileri için kuyruk yöneticisi hata günlüğünü izleyin ve sorunu düzeltin.

AMQ9448 , havuz yöneticisinin bir komutu çalıştırdıktan sonra hata döndürdüğünü gösterir.

Bu hata, komutu her 10 dakikada bir yeniden denemenin başlangıcını ve sona erdirmeyi ertelediğiniz sürece, kuyruk yöneticisini beş gün sonra durdurmanın başlangıcını gösterir.

AMQ5008 , bir IBM MQ işlemi eksik olduğu için kuyruk yöneticisinin durdurulduğunu gösterir.

AMQ5008 , havuz yöneticisinin beş gün sonra durduğu sonucunu verir. Havuz yöneticisi durursa, kuyruk yöneticisi durur.

2. AMQ9448 hata iletilerine ilişkin kuyruk yöneticisi hata günlüğünü izleyin ve sorunu düzeltmeye son verin.

SYSTEM.CLUSTER.COMMAND.QUEUE' den ileti almayı devre dışı bırakırsanız, havuz yöneticisi komutları çalıştırmayı durdurur ve herhangi bir işi işlemekten süresiz olarak devam eder. Ancak, havuz yöneticisinin kuyruklara tuttuğu tanıtıcıları serbest bırakılır. Havuz yöneticisi durmadığı için, kuyruk yöneticisi beş gün sonra durdurulmaz.

SYSTEM.CLUSTER.COMMAND.QUEUE' den ileti almayı devre dışı bırakmak için bir MQSC komutunu çalıştırın:

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(DISABLED)
```

SYSTEM.CLUSTER.COMMAND.QUEUE' den ileti almaya devam etmek için bir MQSC komutu çalıştırın:

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
```

## Özel önem

amqrmfa ögesinin IBM MQ içinde durdurulması, kuyruk yöneticisi hatası olarak kabul edildiğinden, kuyruk yöneticisinin durmasına neden olur. Kuyruk yöneticisi ayarlama değişikliğini (TolerateRepositoryFailure) ayarlamadan amqrmfa işlemini durdurmamalısınız.

## Örnek

```
TuningParameters:  
TolerateRepositoryFailure=TRUE
```

*Şekil 86. qm.ini içinde TolerateRepositoryFailure değerini TRUE olarak ayarlayın*

## İlgili kavramlar

“Kuyruk yöneticisi yapılandırma dosyaları, qm.ini” sayfa 97

Kuyruk yöneticisi yapılandırma dosyası (qm.ini), belirli bir kuyruk yöneticisiyle ilgili bilgileri içerir. Tek bir kuyruk yöneticisinin yapılandırmasını değiştirmek için kullanabileceğiniz öznitelikler, IBM MQ ile ilgili ayarları geçersiz kılar.

## JMS ve Jakarta Messaging kaynaklarının yapılandırılması

Bir JMS ya da Jakarta Messaging uygulamasının IBM MQ ' e bağlanmak için gerek duyduğu kaynakları ve iletileri göndermek ya da almak için gereken hedeflere erişmek üzere kullanabileceği yollardan biri, JNDI ad alanı adı verilen adlandırma ve dizin hizmeti içindeki bir konumdan yönetilen nesnelere almak için Java Naming and Directory Interface (JNDI) olanağını kullanmaktır. Bir JMS uygulamasının JNDI ad alanından denetlenen nesnelere alabilmesi için önce yönetilen nesnelere yaratmanız ve yapılandırmanız gerekir.

## Bu görev hakkında

**V 9.3.0** **V 9.3.0** **JM 3.0** IBM MQ 9.3.0' dan Jakarta Messaging 3.0 , yeni uygulamalar geliştirmek için desteklenir. IBM MQ 9.3.0 , var olan uygulamalar için JMS 2.0 ' e destek vermeye devam eder. Aynı uygulamada hem Jakarta Messaging 3.0 API hem de JMS 2.0 API ' nin kullanılması desteklenmez. Daha fazla bilgi için, bkz. [JMS/Jakarta İleti Sistemi için IBM MQ sınıflarının kullanılması](#).

Aşağıdaki araçlardan birini kullanarak IBM MQ içinde yönetilen nesnel oluşturabilir ve yapılandırabilirsiniz:

### IBM MQ JMS ve Jakarta Messaging yönetim araçları

IBM MQ JMS yönetim aracı, **JMSAdmin** ve Jakarta Messaging yönetim aracı **JMS30Admin**, LDAP ' ta, yerel dosya sisteminde ya da diğer konumlarda saklanan IBM MQ JMS ve Jakarta Messaging nesnelere oluşturmak ve yapılandırmak için kullanabileceğiniz komut satırı araçlarıdır. JMS ve Jakarta Messaging yönetim araçları, **runmqsc** ile benzer bir sözdizimi kullanır ve komut dosyası yazmayı da destekler.

Yönetim araçları, belirli özelliklerin değerlerini ayarlamak için bir yapılandırma dosyası kullanır. JMS kaynaklarını yapılandırmak için aracı kullanarak başlamadan önce sisteminize uyacak şekilde düzenleyebileceğiniz örnek bir yapılandırma dosyası sağlanır. Yapılandırma dosyasıyla ilgili daha fazla bilgi için bkz. "[JMSAdmin ve JMS30Admin araçlarının yapılandırılması](#)" sayfa 671.

### **JMS 2.0** IBM MQ Explorer

JMS 2.0 için, LDAP ' ta, yerel dosya sisteminde ya da diğer yerlerde saklanan JMS 2.0 nesne tanımlamalarını yaratmak ve denetlemek için IBM MQ Explorer komutunu kullanabilirsiniz.

**V 9.3.0** **V 9.3.0** **JM 3.0** Jakarta Messaging 3.0 için, IBM MQ Explorer kullanarak JNDI ' yi yönetemezsiniz. JNDI yönetimi, **JMSAdmin** ' ın **JMS30Admin** olan Jakarta Messaging 3.0 çeşitlemesi tarafından desteklenir.

WebSphere Application Server ' e konuşlandırılan IBM MQ JMS uygulamalarının, uygulama sunucusu JNDI havuzundan JMS nesnelere erişmesi gerekir. Bu nedenle, WebSphere Application Server ile IBM MQ arasında JMS ileti alışverişi kullanıyorsanız, WebSphere Application Server içinde IBM MQ içinde oluşturduğunuz nesnelere karşılık gelen nesnel oluşturmanız gerekir.

**V 9.3.0** **V 9.3.0** **JM 3.0** IBM MQ 9.3 , Jakarta Messaging 3.0 ürününü desteklese de, WebSphere Application Server şu anda eşdeğer bir desteğe sahip değildir. Bu nedenle, WebSphere Application Server içinde Java Message Service 2.0 kaynaklarını yapılandırabilirsiniz.

IBM MQ Explorer ve IBM MQ JMS yönetim aracı, WebSphere Application Server içinde saklanan IBM MQ JMS nesnelere yönetmek için kullanılamaz. Bunun yerine, aşağıdaki araçlardan birini kullanarak WebSphere Application Server içinde yönetilen nesnel oluşturabilir ve yapılandırabilirsiniz:

### WebSphere Application Server yönetim konsolu

WebSphere Application Server yönetim konsolu, WebSphere Application Server içindeki IBM MQ JMS nesnelere yönetmek için kullanabileceğiniz web tabanlı bir araçtır.

### WebSphere Application Server wsadmin komut dosyası oluşturma istemcisi

WebSphere Application Server wsadmin komut dosyası oluşturma istemcisi, WebSphere Application Server içindeki IBM MQ JMS nesnelere yönetmek için özel komutlar sağlar.

Bir IBM MQ kuyruk yöneticisinin kaynaklarına WebSphere Application Server içinde erişmek için JMS uygulamasını kullanmak istiyorsanız, IBM MQ classes for JMS ' in bir sürümünü içeren WebSphere Application Server içindeki IBM MQ ileti alışverişi sağlayıcısını kullanmanız gerekir. WebSphere Application Server ile verilen IBM MQ kaynak bağdaştırıcısı, IBM MQ ileti alışverişi sağlayıcısıyla JMS ileti alışverişi sağlayan tüm uygulamalar tarafından kullanılır. IBM MQ kaynak bağdaştırıcısı genellikle WebSphere Application Server düzeltme paketlerini uyguladığınızda otomatik olarak güncellenir, ancak kaynak bağdaştırıcısını el ile güncellediyse, bakımın doğru şekilde uygulandığından emin olmak için yapılandırmanızı el ile güncellemeniz gerekir.

### İlgili kavramlar

[JMS uygulaması için IBM MQ sınıflarında bağlantı üreticileri ve hedefler oluşturma ve yapılandırma](#)

## İlgili başvurular

[runmqsc \(MQSC komutlarını çalıştır\)](#)

# JNDI ad alanında bağlantı üreticilerini ve hedefleri yapılandırma

JMS ve Jakarta Messaging uygulamaları, adlandırma ve izin hizmetindeki yönetilen nesnelere Java Naming and Directory Interface (JNDI) aracılığıyla erişir. JMS ya da Jakarta Messaging tarafından yönetilen nesnelere, JNDI ad alanı olarak adlandırılan adlandırma ve izin hizmeti içindeki bir yerde saklanır. Bir JMS ya da Jakarta Messaging uygulaması, IBM MQ 'e bağlanmak için yönetilen nesnelere ve ileti göndermek ya da almak için erişim hedeflerini arayabilir.

## Bu görev hakkında

JMS ya da Jakarta Messaging uygulamaları, bağlantıları kullanarak adlandırma ve izin hizmetindeki JMS ya da Jakarta Messaging nesnelere adlarını arayabilir:

### İlk bağlam

İlk bağlam, JNDI ad alanının kökünü tanımlar. Adlandırma ve izin hizmetindeki her bir konum için, bir JMS ya da Jakarta Messaging uygulamasının, adlandırma ve izin hizmetinin bulunduğu yerde denetlenen nesnelere adlarını çözümleyebileceği bir başlangıç noktası vermek üzere bir başlangıç bağlamı belirtmeniz gerekir.

### Alt bağlamlar

Bir bağlamın bir ya da daha çok alt bağlamı olabilir. Alt bağlam, JNDI ad alanının bir alt bölümüdür ve bağlantı üreticileri ve hedefler gibi yönetilen nesnelere ve diğer alt bağlamları içerebilir. Alt bağlam, kendi hakkı olan bir nesne değildir; alt bağlamdaki nesnelere için adlandırma kuralının bir uzantısıdır.

Bir IBM MQ classes for JMS ya da IBM MQ classes for Jakarta Messaging uygulamasının bir JNDI ad alanından denetlenen nesnelere alabilmesi için önce yönetilen nesnelere yaratmanız gerekir. Aşağıdaki JMS ya da Jakarta Messaging nesne tiplerini oluşturabilir ve yapılandırabilirsiniz:

### Bağlantı üreticisi

JMS ya da Jakarta Messaging bağlantı üreticisi nesnesi, bağlantılar için standart yapılandırma özellikleri kümesini tanımlar. Bir JMS ya da Jakarta Messaging uygulaması, IBM MQ ile bağlantı oluşturmak için bir bağlantı üreticisi kullanır. İki ileti sistemi etki alanından birine özgü bir bağlantı üreticisi, noktadan noktaya ileti sistemi etki alanı ve yayınlama/abone olma ileti sistemi etki alanı yaratabilirsiniz.

Diğer bir seçenek olarak, JMS 1.1' den hem noktadan noktaya iletişim hem de yayınlama/abone olma ileti sistemi için kullanılacak etki alanından bağımsız bağlantı üreticileri yaratabilirsiniz. Daha fazla bilgi için bkz. [JMS ve Jakarta Messaging modeli](#).

### Hedef

JMS ya da Jakarta Messaging hedefi, istemcinin ürettiği iletilerin hedefini ve JMS uygulamasının kullandığı iletilerin kaynağını gösteren bir nesnedir. JMS ya da Jakarta Messaging uygulaması, iletilere üzerine koymak ve ileti almak için tek bir hedef nesne kullanabilir ya da uygulama ayrı hedef nesnelere kullanabilir. İki tip hedef nesne vardır:

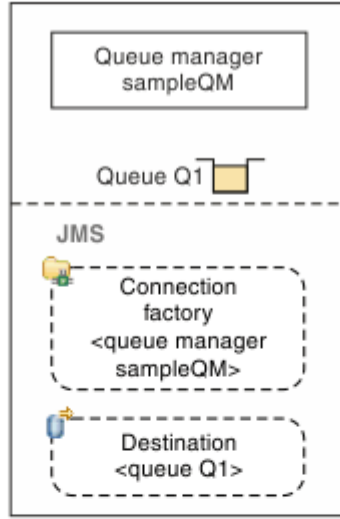
- Noktadan noktaya ileti sisteminde kullanılan JMS ya da Jakarta Messaging kuyruk hedefi
- Yayınlama/abone olma ileti sisteminde kullanılan JMS ya da Jakarta Messaging konu hedefi

**JMS 2.0** JMS 2.0 için, IBM MQ Explorer ya da IBM MQ JMS yönetim aracını **JMSAdmin** kullanarak bağlamlar ve yönetilen nesnelere oluşturabilirsiniz.

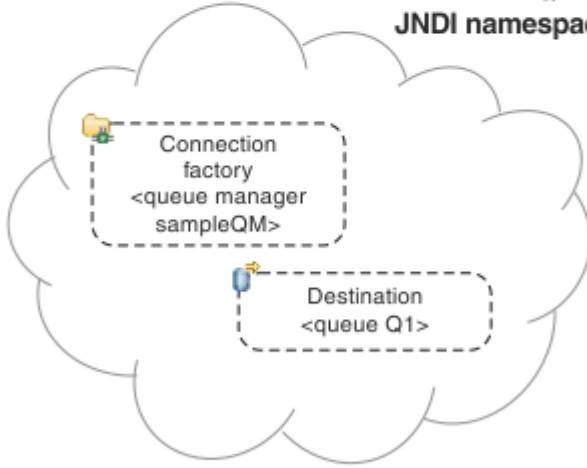
**Not:** **V 9.3.0** **V 9.3.0** **JM 3.0** Jakarta Messaging 3.0 için, IBM MQ Explorer kullanarak JNDI 'yi yönetemezsiniz. JNDI yönetimi, **JMSAdmin** 'in **JMS30Admin** olan Jakarta Messaging 3.0 çeşitlemesi tarafından desteklenir.

Aşağıdaki çizgede, IBM MQ JNDI ad alanında yaratılan JMS ya da Jakarta Messaging nesnelere bir örneği gösterilmektedir.

## IBM MQ



## IBM MQ JNDI namespace

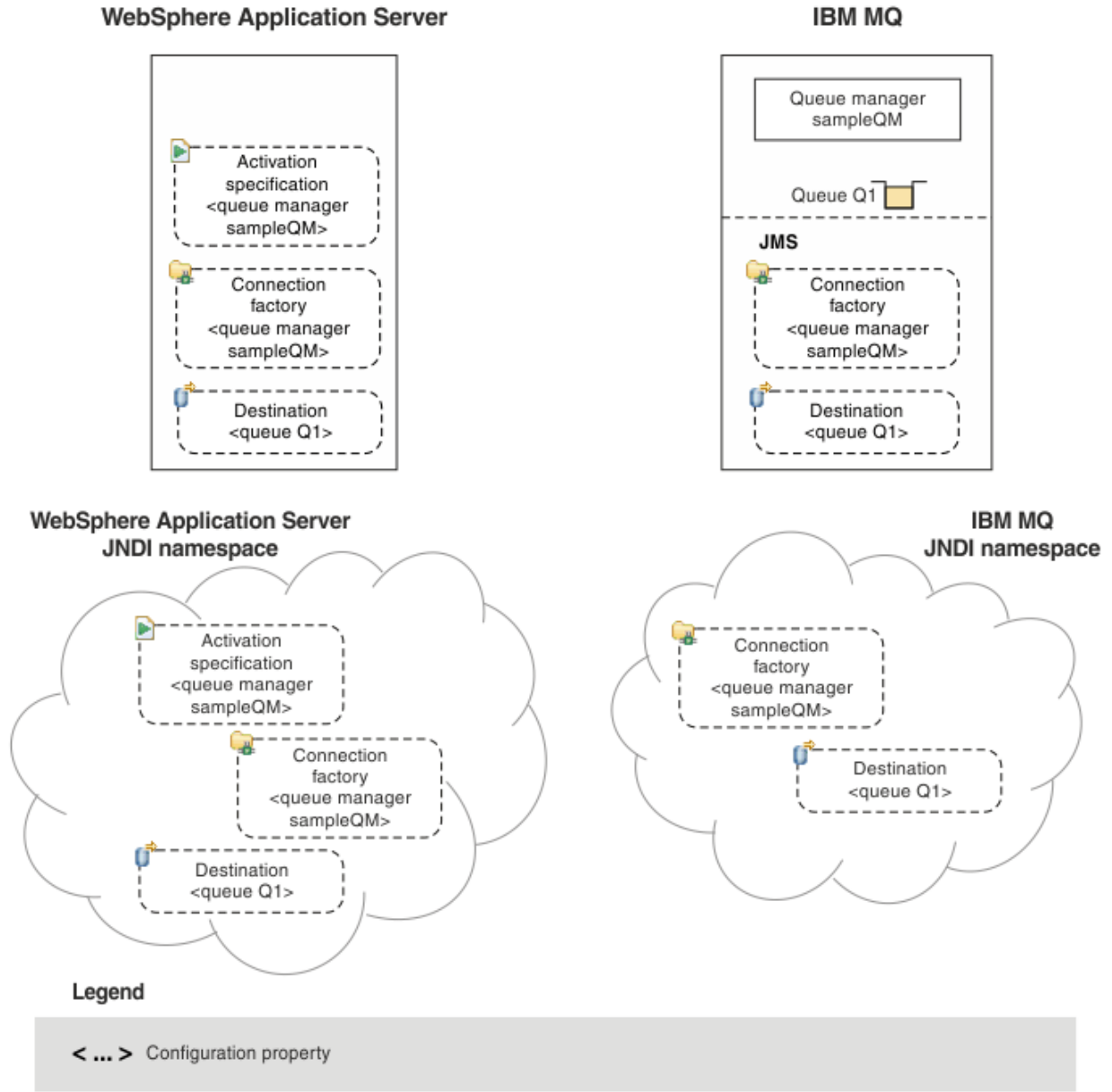


### Legend

< ... > Configuration property

Şekil 87. IBM MQ içinde oluşturulan JMS ya da Jakarta Messaging nesneleri

WebSphere Application Server ile IBM MQ arasında JMS ileti alışverişi kullanıyorsanız, IBM MQ ile iletişim kurmak için WebSphere Application Server içinde karşılık gelen nesneleri oluşturmanız gerekir. Bu nesnelere birini WebSphere Application Server içinde yarattığınızda, nesne aşağıdaki çizgede gösterildiği gibi WebSphere Application Server JNDI ad alanında saklanır.



Şekil 88. WebSphere Application Server’inde oluşturulan nesnelere ve IBM MQ’de karşılık gelen nesnelere

V 9.3.0 V 9.3.0 JM 3.0 IBM MQ 9.3, Jakarta Messaging 3.0 ürününü desteklese de, WebSphere Application Server şu anda eşdeğer bir desteğe sahip değildir. Bu nedenle, WebSphere Application Server’inde Java Message Service 2.0 kaynaklarını yapılandırabilirsiniz.

Uygulamanız ileti odaklı bir bean (MDB) kullanıyorsa, bağlantı üreticisi yalnızca giden iletiler için kullanılır ve gelen iletiler bir etkinleştirme belirtimi tarafından alınır. Etkinleştirme belirtimleri, Java EE Connector Architecture 1.5 (JCA 1.5) standardının bir parçasıdır. JCA 1.5, IBM MQ’gibi JMS sağlayıcılarını WebSphere Application Server’gibi Java EE uygulama sunucularıyla bütünleştirmek için standart bir yol sağlar. JMS etkinleştirme belirtimi, bir ya da daha fazla iletiyle yönlendirilen bean (MDB) ile ilişkilendirilebilir ve bu veritabanı yöneticilerinin bir hedefe ulaşan iletileri dinlemesi için gereken yapılandırmayı sağlar.

Gereksinim duyduğunuz JMS kaynaklarını oluşturmak ve yapılandırmak için WebSphere Application Server yönetim konsolunu ya da wsadmin komut dosyası oluşturma komutlarını kullanabilirsiniz.



## Yordam

- **JMS 2.0**  
JMS nesnelere IBM MQ Explorer kullanarak IBM MQ için yapılandırmak üzere bkz. [“JMS 2.0 nesnelere IBM MQ Explorer kullanarak yapılandırma”](#) sayfa 668.
- **JMS 2.0**  
JMS nesnelere IBM MQ için IBM MQ JMS yönetim aracını ( **JMSAdmin**) kullanarak yapılandırmak için bkz. [“Yönetim araçlarını kullanarak JMS ve Jakarta Messaging nesnelere yapılandırma”](#) sayfa 669.
- **V 9.3.0** **V 9.3.0** **JM 3.0**  
Jakarta Messaging nesnelere IBM MQ için IBM MQ Jakarta Messaging yönetim aracını ( **JMS30Admin**) kullanarak yapılandırmak için bkz. [“Yönetim araçlarını kullanarak JMS ve Jakarta Messaging nesnelere yapılandırma”](#) sayfa 669.
- **JMS 2.0**  
JMS nesnelere WebSphere Application Server için yapılandırmak üzere bkz. [“WebSphere Application Server içinde JMS 2.0 kaynaklarının yapılandırılması”](#) sayfa 679.

## Sonuçlar

Bir IBM MQ classes for JMS ya da IBM MQ classes for Jakarta Messaging uygulaması, JNDI ad alanından yönetilen nesnelere alabilir ve gerekirse, IBM JMS uzantılarını ya da IBM MQ JMS uzantılarını kullanarak bir ya da daha fazla özelliğini ayarlayabilir ya da değiştirebilir.

### İlgili görevler

JMS uygulamasında denetlenen nesnelere almak için JNDI ' yi kullanma

[IBM MQ classes for JMS uygulamasında bağlantı üreticileri ve hedefler oluşturma ve yapılandırma](#)

## **JMS 2.0** JMS 2.0 nesnelere IBM MQ Explorer kullanarak yapılandırma

JMS nesnelere IBM MQ nesnelereinden ve IBM MQ nesnelereini JMS nesnelereinden yaratmak ve diğer IBM MQ nesnelereini denetlemek ve izlemek için IBM MQ Explorer grafik kullanıcı arabirimini kullanın.

### Bu görev hakkında

**JMS 2.0** IBM MQ Explorer , yerel bilgisayarınızda ya da uzak bir sistemde barındırılan IBM MQ nesnelereini yönetebileceğiniz ve izleyebileceğiniz grafik kullanıcı arabirimidir. IBM MQ Explorer , Windows ve Linux for x86-64 üzerinde çalışır. z/OSda dahil olmak üzere desteklenen herhangi bir platformda çalışan kuyruk yöneticilerine uzaktan bağlanarak tüm ileti sistemi omurganızın görüntülenmesine, keşfedilmesine ve konsoldan değiştirilmesine olanak sağlar.

**Not:** **V 9.3.0** **V 9.3.0** **JM 3.0** Jakarta Messaging 3.0 için, IBM MQ Explorer kullanarak JNDI ' yi yönetemezsiniz. JNDI yönetimi, **JMSAdmin** ' ın **JMS30Admin** olan Jakarta Messaging 3.0 çeşitlemesi tarafından desteklenir.

IBM MQ Explorer içinde, tüm bağlantı üreticileri uygun bağlam ve alt bağlamlarda Bağlantı Üreticileri klasörlerinde saklanır.

IBM MQ Explorer ile, IBM MQ Explorer içinde var olan bir nesneden bağlamsal olarak ya da yeni nesne yaratma sihirbazından aşağıdaki görev tiplerini gerçekleştirebilirsiniz:

- Aşağıdaki IBM MQ nesnelereinden herhangi birinden bir JMS Connection Factory oluşturun:
  - İster yerel bilgisayarınızda, ister uzak bir sistemde olsun, IBM MQ kuyruk yöneticisi.
  - Bir IBM MQ kanalı.
  - Bir IBM MQ dinleyicisi.
- Bir IBM MQ kuyruk yöneticisini JMS Connection Factory kullanarak IBM MQ Explorer ' e ekleyin.
- IBM MQ kuyruğundan bir JMS kuyruğu oluşturun.



- JMS kuyruğundan bir IBM MQ kuyruğu oluşturun.
- Bir IBM MQ konusundan, IBM MQ nesnesi ya da dinamik bir konu olabilen bir JMS konusu oluşturun.
- JMS konusundan bir IBM MQ konusu oluşturun.

## Yordam

- Zaten çalışmıyorsa, IBM MQ Explorer' a başlayın.  
IBM MQ Explorer çalışıyorsa ve Hoş Geldiniz sayfasını görüntülüyorsa, IBM MQ nesnelerini yönetmeye başlamak için Hoş Geldiniz sayfasını kapatın.
- Henüz yapmadıysanız, JMS nesnelerinin adlandırma ve izin hizmetinde saklandığı JNDI ad alanının kökünü tanımlayan bir başlangıç bağımları yaratın.  
İlk bağımları IBM MQ Explorer' a eklediğinizde, JNDI ad alanında bağlantı üreticisi nesnelere, hedef nesnelere ve alt bağımları yaratabilirsiniz.  
İlk bağımları, JMS Administered Objects klasöründeki Navigator görünümünde görüntülenir. JNDI ad alanının tüm içeriği görüntülense de, IBM MQ Explorer içinde yalnızca burada saklanan IBM MQ classes for JMS nesnelerini düzenleyebilirsiniz. Daha fazla bilgi için [Başlangıç bağımları eklenmesibaşlıklı konuya](#) bakın.
- Gereksinim duyduğunuz alt bağımları ve JMS tarafından denetlenen nesnelere oluşturun ve yapılandırın.  
Daha fazla bilgi için [JMS yönetilen nesnelere oluşturma ve yapılandırma](#) başlıklı konuya bakın.
- IBM MQ' yi yapılandırın.  
Daha fazla bilgi için, bkz. [IBM MQ Explorer kullanarak IBM MQ ' nin yapılandırılması](#) .

## İlgili kavramlar

[IBM MQ Explorer ' a Giriş](#)

[IBM MQ classes for JMS uygulamasında bağlantı üreticileri ve hedefler oluşturma ve yapılandırma](#)

## Yönetim araçlarını kullanarak JMS ve Jakarta Messaging nesnelerini yapılandırma

IBM MQ , sekiz tip IBM MQ classes for JMS ya da IBM MQ classes for Jakarta Messaging nesnesinin özelliklerini tanımlamak ve bunları bir JNDI ad alanında saklamak için kullanabileceğiniz yönetim araçları sağlar. Uygulamalar daha sonra, ad alanından bu denetlenen nesnelere almak için JNDI ' yi kullanabilir.

## Bu görev hakkında

**JMS 2.0** JMS 2.0 için, JNDI yönetimi **JMSAdmin** aracı tarafından desteklenir.

**V 9.3.0** **V 9.3.0** **JM 3.0** Jakarta Messaging 3.0 için JNDI yönetimi, **JMS30Admin** olan **JMSAdmin** Jakarta Messaging 3.0 çeşitlemesi tarafından desteklenir.

Aşağıdaki tabloda, fiil kullanarak yaratabileceğiniz, yapılandırabileceğiniz ve işleyebileceğiniz sekiz tip yönetilen nesne gösterilmektedir. Anahtar Sözcük sütunu, Çizelge 36 sayfa 669 içinde gösterilen komutlarda *TYPE* yerine koyabileceğiniz dizgileri gösterir.

Çizelge 36. Yönetim aracı tarafından işlenen JMS ve Jakarta Messaging nesne tipleri		
Nesne Tipi	Anahtar Kelime	Açıklama
MQConnectionFactory	CF	JMS ConnectionFactory arabiriminin IBM MQ uygulaması. Bu, hem noktadan noktaya iletişim hem de yayınlama/abone olma etki alanlarında bağlantı oluşturmak için bir üretici nesnesini temsil eder.

Çizelge 36. Yönetim aracı tarafından işlenen JMS ve Jakarta Messaging nesne tipleri (devamı var)

Nesne Tipi	Anahtar Kelime	Açıklama
MQQueueConnectionFactory	QCF	JMS QueueConnectionFactory arabiriminin IBM MQ uygulaması. Bu, noktadan noktaya iletişim etki alanında bağlantı yaratmak için kullanılan bir üretici nesnesini gösterir.
MQTopicConnectionFactory	TCF	JMS TopicConnectionFactory arabiriminin IBM MQ uygulaması. Bu, yayınlama/abone olma etki alanında bağlantı yaratmak için kullanılan bir üretici nesnesini gösterir.
MQQueue	Q	JMS kuyruk arabiriminin IBM MQ uygulaması. Bu, noktadan noktaya iletişim etki alanındaki iletilere ilişkin bir hedefi gösterir.
MQTopic	T	JMS Konu arabiriminin IBM MQ uygulaması. Bu, yayınlama/abone olma etki alanındaki iletiler için bir hedefi temsil eder.
MQXAConnectionFactory <a href="#">"1" sayfa 670</a>	XACF	JMS XAConnectionFactory arabiriminin IBM MQ uygulaması. Bu, hem noktadan noktaya iletişim hem de yayınlama/abone olma etki alanlarında bağlantı yaratmak için kullanılan ve bağlantıların JMS sınıflarının XA sürümlerini kullandığı bir üretici nesnesini temsil eder.
MQXAQueueConnectionFactory <a href="#">"1" sayfa 670</a>	XAQCF	JMS XAQueueConnectionFactory arabiriminin IBM MQ uygulaması. Bu, JMS sınıflarının XA sürümlerini kullanan noktadan noktaya iletişim etki alanında bağlantı yaratmak için kullanılan bir üretici nesnesini gösterir.
MQXATopicConnectionFactory <a href="#">"1" sayfa 670</a>	XATCF	JMS XATopicConnectionFactory arabiriminin IBM MQ uygulaması. Bu, JMS sınıflarının XA sürümlerini kullanan yayınlama/abone olma etki alanında bağlantı yaratmak için kullanılan bir üretici nesnesini temsil eder.

**Not:**

1. Bu sınıflar, uygulama sunucusu satıcıları tarafından kullanılmak üzere sağlanır. Uygulama programcıları için doğrudan kullanışlı olma olasılıkları düşüktür.

Bu nesnelerin nasıl yapılandırılacağı hakkında daha fazla bilgi için bkz. ["JMS nesnelerini yapılandırma" sayfa 678.](#)

Bu aracı kullanmanız için gereken özellik tipleri ve değerleri, [IBM MQ classes for JMS nesnelerinin özellikleri](#) içinde listelenir.

Bu aracı, ["Alt bağlamların yapılandırılması" sayfa 675](#) içinde açıklandığı gibi JNDI içindeki dizin ad alanı alt bağlamlarını işlemek için de kullanabilirsiniz.

**JMS 2.0** JMS 2.0 ve daha önceki sürümler için, IBM MQ Explorer ile yönetilen IBM MQ classes for JMS nesneleri oluşturabilir ve yapılandırabilirsiniz.

**V 9.3.0** **V 9.3.0** **JM 3.0** Jakarta Messaging 3.0 için, IBM MQ Explorer kullanarak JNDI ' yi yönetemezsiniz. JNDI yönetimi, **JMSAdmin** ' in **JMS30Admin** olan Jakarta Messaging 3.0 çeşitlemesi tarafından desteklenir.

### İlgili kavramlar

IBM MQ classes for JMS uygulamasında bağlantı üreticileri ve hedefler oluşturma ve yapılandırma JMS uygulamasında denetlenen nesnelere almak için JNDI ' yi kullanma

## JMSAdmin ve JMS30Admin araçlarının yapılandırılması

IBM MQ JMS ve Jakarta Messaging yönetim araçları, belirli özelliklerin değerlerini ayarlamak için bir yapılandırma dosyası kullanır. Her bir durumda, sisteminize uyacak şekilde düzenleyebileceğiniz örnek bir yapılandırma dosyası sağlanır.

### Bu görev hakkında

**V 9.3.0** **V 9.3.0** **JM 3.0** IBM MQ 9.3.0 , Jakarta Messaging 3.0 için destek sağlar. JMS 2.0 tam olarak desteklenmeye devam etmektedir.

Yapılandırma dosyası, eşittir işaretiyle (=) ayrılmış bir anahtar-değer çifti kümesinden oluşan düz metin dosyasıdır. Yapılanış kütüğünde tanımlı olan üç özellik için değer ayarlayarak, denetim aracını yapılandırabilirsiniz. Aşağıdaki örnekte şu üç özellik gösterilmektedir:

```
#Set the service provider
INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
#Set the initial context
PROVIDER_URL=ldap://polaris/o=ibm_us,c=us
#Set the authentication type
SECURITY_AUTHENTICATION=none
```

Bu örnekte, satırın ilk kolonundaki bir HASH işareti (#) bir açıklamayı ya da kullanılmayan bir satırı gösterir.

Varsayılan yapılandırma dosyası olarak kullanılan örnek bir yapılandırma dosyası IBM MQ ile birlikte sağlanır. Örnek dosya **JMSAdmin.config** ( JMS 2.0 için) ya da **JMS30Admin.config** ( Jakarta Messaging 3.0 için) olarak adlandırılır. Bu dosya, **MQ\_JAVA\_INSTALL\_PATH/bin** dizininde bulunur. Örnek dosyayı düzenleyerek sisteminiz için gereken ayarları tanımlayabilir ya da kendi yapılanış dosyanızı yaratabilirsiniz.

Yönetim aracını başlattığınızda, kullanmak istediğiniz yapılanış kütüğünü -c f g komut satırı değiştirgesini kullanarak belirtebilirsiniz (açıklamalar için bkz. “[JMSAdmin ve JMS30Admin araçlarının başlatılması](#)” sayfa 673). Aracı çağırdığınızda bir yapılandırma dosyası adı belirtmezseniz, araç varsayılan yapılandırma dosyasını ( **JMSAdmin.config** ya da **JMS30Admin.config**) yüklemeyi dener. Önce geçerli dizinde, sonra **MQ\_JAVA\_INSTALL\_PATH/bin** dizininde bu dosyayı arar; burada **MQ\_JAVA\_INSTALL\_PATH** , IBM MQ classes for JMS ya da IBM MQ classes for Jakarta Messaging kurulumunuzun yoludur.

LDAP ortamında saklanan JMS ya da Jakarta Messaging nesnelere adları, LDAP adlandırma kurallarına uygun olmalıdır. Bu kurallardan biri, nesne ve bağlam adlarının cn= (ortak ad) ya da ou= (kuruluş birimi) gibi bir önek içermesi gereklidir. Yönetim aracı, bir önek olmadan nesne ve bağlam adlarına başvurmanızı sağlayarak LDAP hizmet sağlayıcılarının kullanımını basitleştirir. Bir önek belirtmezseniz, araç otomatik olarak belirttiğiniz ada bir varsayılan önek ekler. LDAP için bu cn=. Gerekirse, yapılandırma dosyasında **NAME\_PREFIX** özelliğini ayarlayarak varsayılan öneki değiştirebilirsiniz.

**Not:** Java nesnelere depolamak için LDAP sunucunuzu yapılandırmanız gerekebilir. Daha fazla bilgi için LDAP sunucunuza ilişkin belgelere bakın.

### Yordam

1. **INITIAL\_CONTEXT\_FACTORY** özelliğini yapılandırarak aracın kullandığı hizmet sağlayıcıyı tanımlayın. Bu özellik için desteklenen değerler şunlardır:
  - com.sun.jndi.ldap.LdapCtxFactory (LDAP için)

- com.sun.jndi.fscontext.ReffFSContextFactory (dosya sistemi bağlamı için)
- **z/OS** com.ibm.jndi.LDAPCtxFactory yalnızca z/OS üzerinde desteklenir ve bir LDAP sunucusuna erişim sağlar. Ancak, bu sınıf com.sun.jndi.Ldap.LdapCtxFactory ile uyumsuz; bu, bir InitialContextFactory kullanılarak yaratılan nesnelerin okunamadığı ya da değiştirilemediği anlamına geliyor.

Denetim aracını, JMSAdmin ya da JMS30Admin yapılandırma dosyasında tanımlanan üç değiştirgeyi kullanarak diğer JNDI bağlamlarına bağlanmak için de kullanabilirsiniz. Farklı bir InitialContext üreticisi kullanmak için:

- a) **INITIAL\_CONTEXT\_FACTORY** özelliğini gerekli sınıf adına ayarlayın.
- b) **USE\_INITIAL\_DIR\_CONTEXT**, **NAME\_PREFIX** ve **NAME\_READABILITY\_MARKER** özelliklerini kullanarak InitialContextFactory davranışını tanımlayın.

Bu özelliklere ilişkin ayarlar, örnek yapılandırma dosyası açıklamalarında açıklanır.

Desteklenen **INITIAL\_CONTEXT\_FACTORY** değerlerinden birini kullanıyorsanız, **USE\_INITIAL\_DIR\_CONTEXT**, **NAME\_PREFIX** ve **NAME\_READABILITY\_MARKER** özelliklerini tanımlamanıza gerek yoktur. Ancak, sistem varsayılanlarını geçersiz kılmak istiyorsanız, bu özelliklere değer verebilirsiniz. Örneğin, nesnelere bir LDAP ortamında saklandıysa, **NAME\_PREFIX** özelliğini gerekli öneke ayarlayarak aracın nesne ve bağlam adlarına eklediği varsayılan öneki değiştirebilirsiniz.

Üç InitialContextFactory özelliğinin birini ya da birkaçını atlıyorsanız, denetim aracı diğer özelliklerin değerlerine dayalı olarak uygun varsayılan değerler sağlar.

2. **PROVIDER\_URL** özelliğini yapılandırarak oturumun ilk bağlamının URL adresini tanımlayın.

Bu URL, araç tarafından gerçekleştirilen tüm JNDI işlemlerinin köküdür. Bu özelliğin iki biçimi desteklenir:

- ldap://hostname/contextname
- kütük: [ sürücü: ] /yol adı

LDAP URL 'nin biçimi LDAP sağlayıcınıza bağlı olarak değişebilir. Daha fazla bilgi için LDAP belgelerinize bakın.

3. JNDI 'nin **SECURITY\_AUTHENTICATION** özelliğini yapılandırarak güvenlik kimlik bilgilerini hizmet sağlayıcınıza iletip aktarmayacağını tanımlayın.

Bu özellik yalnızca bir LDAP hizmet sağlayıcısı kullanıldığında kullanılır ve üç değerden birini alabilir:

#### **none (anonim kimlik doğrulaması)**

Bu parametreyi none olarak ayarlarsanız, JNDI hizmet sağlayıcıya güvenlik kimlik bilgileri geçirmez ve *anonim kimlik doğrulaması* gerçekleştirilir.

#### **basit (basit kimlik doğrulama)**

Parametreyi basit olarak ayarlarsanız, güvenlik kimlik bilgileri JNDI aracılığıyla temel hizmet sağlayıcısına iletilir. Bu güvenlik kimlik bilgileri, kullanıcı ayırt edici adı (User DN) ve parola biçimindedir.

#### **CRAM-MD5 (CRAM-MD5 kimlik doğrulama mekanizması)**

Parametreyi CRAM-MD5 olarak ayarlarsanız, güvenlik kimlik bilgileri JNDI üzerinden temel hizmet sağlayıcısına iletilir. Bu güvenlik kimlik bilgileri, kullanıcı ayırt edici adı (User DN) ve parola biçimindedir.

**SECURITY\_AUTHENTICATION** özelliği için geçerli bir değer sağlamazsanız, özellik varsayılan olarak none değerine ayarlanır.

Güvenlik kimlik bilgileri gerekiyorsa, araç kullanıma hazırlandığında sizden bu kimlik bilgileri istenir. JMSAdmin yapılandırma dosyasında **PROVIDER\_USERDN** ve **PROVIDER\_PASSWORD** özelliklerini ayarlayarak bunu önleyebilirsiniz.

**Not:** Bu özellikleri kullanmazsanız, *parola da içinde olmak üzere* yazılan metin ekrana yansıtılır. Bunun güvenlik sonuçları olabilir.

Araç kimlik doğrulamasını kendisi yapmaz; kimlik doğrulama görevi LDAP sunucusuna devredilir. LDAP sunucusu denetimcisi, dizinin farklı bölümlerine erişim ayrıcalıkları ayarlamalı ve bu ayrıcalıkları

korunmalıdır. Daha fazla bilgi için LDAP belgelerinize bakın. Kimlik doğrulaması başarısız olursa, araç uygun bir hata iletisi görüntüler ve sonlandırılır.

Güvenlik ve JNDI ile ilgili daha ayrıntılı bilgiler Oracle' ın Java web sitesinde ( [Oracle Technology Network for Java Developers](#) ) yer alır.

## JMSAdmin ve JMS30Admin araçlarının başlatılması

IBM MQ JMS ve Jakarta Messaging yönetim araçlarının etkileşimli olarak ya da toplu işlem başlatmak için kullanabileceğiniz bir komut satırı arabirimi vardır.

### Bu görev hakkında

Etkileşimli kip, denetim komutlarını girebileceğiniz bir komut istemi sağlar. Toplu kipte, aracı başlatma komutu, yönetim komutu komut dosyası içeren bir dosyanın adını içerir.

### Yordam

Etkileşimli kip

- Aracı etkileşimli kipte başlatmak için aşağıdaki komutu girin:

```
> JMS 2.0
```

```
JMSAdmin [-t] [-v] [-cfg config_filename]
```

```
> JM 3.0
```

```
JMS30Admin [-t] [-v] [-cfg config_filename]
```

Burada:

**-t**

İzlemeyi etkinleştirir (varsayılan: izleme kapalı).

İzleme kütüğü şu yerde üretilir: "%MQ\_JAVA\_DATA\_PATH%\errors ( Windows ) ya da /var/mqm/trace ( AIX and Linux ). İzleme dosyasının adı şu biçimdedir:

```
mjms_PID.trc
```

Burada *PID* , JVM ' nin işlem tanıtıcısıdır.

**-v**

Ayrıntılı çıkış üretir (varsayılan değer terse çıkıştır).

**-cfg config\_filename**

Diğer bir yapılandırma dosyasını adlandırıyor. Bu parametre atlanırsa, varsayılan yapılandırma dosyası JMSAdmin.config ( JMS 2.0 için ) ya da JMS30Admin.config ( Jakarta Messaging 3.0 için ) kullanılır. Yapılandırma dosyasıyla ilgili daha fazla bilgi için bkz. "[JMSAdmin ve JMS30Admin araçlarının yapılandırılması](#)" sayfa 671.

Aracın yönetim komutlarını kabul etmeye hazır olduğunu gösteren bir komut istemi görüntülenir. Bu bilgi istemi başlangıçta şu şekilde görünür:

```
InitCtx>
```

Yürürlükteki bağlamın (yani, tüm adlandırma ve izin işlemlerinin başvuruda bulunduğu JNDI bağlamının) **PROVIDER\_URL** yapılanış değiştirilmesinde tanımlanan ilk bağlam olduğunu gösterir. Bu parametreyle ilgili daha fazla bilgi için bkz. "[JMSAdmin ve JMS30Admin araçlarının yapılandırılması](#)" sayfa 671.

Dizin ad alanını geçtikçe, bilgi istemi bunu yansıtacak şekilde değişir, böylece bilgi istemi her zaman geçerli bağlamı görüntüler.

Toplu iş kipi

- Aracı toplu kipte başlatmak için şu komutu girin:

JMS 2.0

```
JMSAdmin test.scp
```

JM 3.0

```
JMS30Admin test.scp
```

Burada *test.scp* , yönetim komutlarını içeren bir komut dosyasıdır. Daha fazla bilgi için bkz [“JMSAdmin ve JMS30Admin ile yönetim komutlarının kullanılması” sayfa 674](#). Dosyadaki son komut END komutu olmalıdır.

## JMSAdmin ve JMS30Admin ile yönetim komutlarının kullanılması

IBM MQ JMS ve Jakarta Messaging yönetim araçları, bir yönetim filinden ve ilgili değiştirgelerinden oluşan komutları kabul eder.

### Bu görev hakkında

Aşağıdaki çizelge, denetim araçlarıyla komut girerken kullanabileceğiniz denetim komutlarını listeler.

Çizelge 37. Yönetim fiilleri		
Komut	Kısa Biçim	Açıklama
Çeviri	Alt	Denetlenen bir nesnenin özelliklerinden en az birinin değiştirilmesi
Tanımla	DEF	Yönetilen bir nesne oluşturun ve saklayın ya da bir alt bağlam oluşturun
GÖRÜNTÜLE	DIS	Bir ya da daha fazla saklanan yönetilen nesnenin özelliklerini ya da yürürlükteki bağlamın içeriğini görüntüler
SİL	Sil	Bir ya da daha çok denetlenen nesneyi ad alanından kaldırın ya da boş bir alt bağlamı kaldırın
Değiştir	CHG	Kullanıcının dizin ad alanını ilk bağlamın altında herhangi bir yerde geçmesine izin vererek yürürlükteki bağlamı değiştirin (güvenlik açıklığı bekleniyor)
kopyala	CP	Depolanan bir yönetilen nesnenin bir kopyasını oluşturun ve bunu alternatif bir ad altında depolayın
Taşı	MV	Denetlenen bir nesnenin saklandığı adı değiştir
BİTİŞ		Yönetim aracını kapat

### Yordam

- Yönetim aracı başlatılmadıysa, [“JMSAdmin ve JMS30Admin araçlarının başlatılması” sayfa 673](#) başlıklı konuda açıklandığı gibi başlatın.

Aracın yönetim komutlarını kabul etmeye hazır olduğunu gösteren komut istemi görüntülenir. Bu bilgi istemi başlangıçta şu şekilde görünür:

InitCtx>

Yürürlükteki bağlamı değiştirmek için, “[Alt bağlamların yapılandırılması](#)” sayfa 675’inde açıkladığı gibi CHANGE komutunu kullanın.

- Komutları aşağıdaki biçimde girin:

**verb** [param]\*

Burada **verb** , Çizelge 37 sayfa 674’inde listelenen yönetim fiillerinden biridir. Tüm geçerli komutlar, komutun başında standart ya da kısa biçiminde görünen bir fiil içerir. Fiil adları büyük ve küçük harfe duyarlı değildir.

- Bir komutu sonlandırmak için, birden çok komutu birlikte girmek istemiyorsanız Enter tuşuna basın; bu durumda, Enter tuşuna basmadan önce artı işaretini (+) doğrudan yazın.

Genellikle, komutları sonlandırmak için Enter tuşuna basmanız gerekir. Ancak, Enter tuşuna basmadan önce artı işaretini (+) yazarak bunu geçersiz kılabilirsiniz. Bu, aşağıdaki örnekte gösterildiği gibi çok satırlı komutlar girmenizi sağlar:

```
DEFINE Q(BookingsInputQueue) +
QMGR(QM.POLARIS.TEST) +
QUEUE(BOOKINGS.INPUT.QUEUE) +
PORT(1415) +
CCSID(437)
```

- Yönetim aracını kapatmak için **END** fiilini kullanın.  
Bu komut herhangi bir parametre alamaz.

## Alt bağlamların yapılandırılması

Dizin ad alanı alt bağlamlarını yapılandırmak için **CHANGE**, **DEFINE**, **DISPLAY** ve **DELETE** fiillerini kullanabilirsiniz.

## Bu görev hakkında

Bu fiillerin kullanımı aşağıdaki tabloda açıklanmıştır.

Çizelge 38. Alt bağlamları işlemek için kullanılan komutların sözdizimi ve tanımı	
Komut sözdizimi	Açıklama
DEFINE CTX (ctxName)	ctxNameadlı yürürlükteki bağlamın alt bağlamını yaratma girişiminde bulunur. Bir güvenlik ihlali varsa, alt bağlam zaten varsa ya da belirtilen ad geçerliyse başarısız olur.
CTX GÖRÜNTÜLE	Yürürlükteki bağlamın içeriğini görüntüler. Yönetilen nesnelere a, alt bağlamlarına [D] ile ek açıklama konmuştur. Her nesnenin Java tipi de görüntülenir.
DELETE CTX (ctxName)	ctxNameadlı yürürlükteki bağlamın alt bağlamını silme girişiminde bulunur. Bağlam bulunamazsa, boş değilse ya da bir güvenlik ihlali varsa başarısız olur.

Çizelge 38. Alt bağlamları işlemek için kullanılan komutların sözdizimi ve tanımı (devamı var)

Komut sözdizimi	Açıklama
CHANGE CTX (ctxName)	<p>Yürürlükteki bağlamı değiştirir; böylece, şimdi ctxNameadlı alt bağlama gönderme yapar. ctxName özel değerlerinden biri sağlanabilir:</p> <p><b>= UP (Yukarı)</b> yürürlükteki bağlamın üst ögesine gider</p> <p><b>= INIT</b> doğrudan ilk bağlama gider</p> <p>Belirtilen bağlam yoksa ya da bir güvenlik ihlali varsa başarısız olur.</p>

LDAP ortamında saklanan JMS ya da Jakarta Messaging nesnelere adları, LDAP adlandırma kurallarına uygun olmalıdır. Bu kurallardan biri, nesne ve bağlam adlarının cn= (ortak ad) ya da ou= (kuruluş birimi) gibi bir önek içermesi gereklidir. Yönetim aracı, bir önek olmadan nesne ve bağlam adlarına başvurmanızı sağlayarak LDAP hizmet sağlayıcılarının kullanımını basitleştirir. Bir önek belirtmezseniz, araç otomatik olarak belirttiğiniz ada bir varsayılan önek ekler. LDAP için bu cn=. Gerekirse, yapılandırma dosyasında **NAME\_PREFIX** özelliğini ayarlayarak varsayılan öneki değiştirebilirsiniz. Daha fazla bilgi için bkz [“JMSAdmin ve JMS30Admin araçlarının yapılandırılması” sayfa 671.](#)

**Not:** Java nesnelere depolamak için LDAP sunucunuzu yapılandırmanız gerekebilir. Daha fazla bilgi için LDAP sunucunuza ilişkin belgelere bakın.

## JMS nesnelere oluşturma

JMS ya da Jakarta Messaging bağlantı üreticisi ve hedef nesnelere yaratmak ve bunları bir JNDI ad alanında saklamak için DEFINE komutunu kullanın. Nesnelere bir LDAP ortamında saklamak için, nesnelere belirli kurallara uygun adlar vermeniz gerekir. Denetim aracı, nesne adlarına varsayılan bir önek ekleyerek LDAP adlandırma kurallarına uymanıza yardımcı olabilir.

## Bu görev hakkında

DEFINE komutu, belirttiğiniz tip, ad ve özelliklerle denetlenen bir nesne yaratır. Yeni nesne yürürlükteki bağlamda saklanır.

LDAP ortamında saklanan JMS ya da Jakarta Messaging nesnelere adları, LDAP adlandırma kurallarına uygun olmalıdır. Bu kurallardan biri, nesne ve bağlam adlarının cn= (ortak ad) ya da ou= (kuruluş birimi) gibi bir önek içermesi gereklidir. Yönetim aracı, bir önek olmadan nesne ve bağlam adlarına başvurmanızı sağlayarak LDAP hizmet sağlayıcılarının kullanımını basitleştirir. Bir önek belirtmezseniz, araç otomatik olarak belirttiğiniz ada bir varsayılan önek ekler. LDAP için bu cn=. Gerekirse, yapılandırma dosyasında **NAME\_PREFIX** özelliğini ayarlayarak varsayılan öneki değiştirebilirsiniz. Daha fazla bilgi için bkz [“JMSAdmin ve JMS30Admin araçlarının yapılandırılması” sayfa 671.](#)

**Not:** Java nesnelere depolamak için LDAP sunucunuzu yapılandırmanız gerekebilir. Daha fazla bilgi için LDAP sunucunuza ilişkin belgelere bakın.

## Yordam

1. Yönetim aracı başlatılmadıysa, [“JMSAdmin ve JMS30Admin araçlarının başlatılması” sayfa 673](#) başlıklı konuda açıklandığı gibi başlatın.  
Aracın yönetim komutlarını kabul etmeye hazır olduğunu gösteren komut istemi görüntülenir.
2. Komut isteminin, yeni nesneyi yaratmak istediğiniz bağlamı gösterdiğinden emin olun.  
Yönetim aracını başlattığınızda, bilgi istemi başlangıçta şu şekilde görünür:

```
InitCtx>
```



Yürürlükteki bağlamı değiştirmek için, “[Alt bağlamların yapılandırılması](#)” sayfa 675’inde açıklandığı gibi CHANGE komutunu kullanın.

3. Bir bağlantı üreticisi, kuyruk hedefi ya da konu hedefi yaratmak için aşağıdaki komut sözdizimini kullanın:

```
DEFINE TYPE (name) [property]*
```

Başka bir deyişle, DEFINE komutunu, ardından bir *TYPE* (name) yönetilen nesne başvurusu, ardından sıfır ya da daha fazla *özellik* yazın (bkz. [IBM MQ classes for JMS nesnelerinin özellikleri](#)).

4. Bir bağlantı üreticisi, kuyruk hedefi ya da konu hedefi yaratmak için aşağıdaki komut sözdizimini kullanın:

```
DEFINE TYPE (name) [property]*
```

5. Yeni yaratılan nesneyi görüntülemek için DISPLAY komutunu aşağıdaki komut sözdizimiyle kullanın:

```
DISPLAY TYPE (name)
```

## Örnek

Aşağıdaki örnekte, DEFINE komutu kullanılarak ilk bağlamda yaratılan testQueue adlı bir kuyruk gösterilmektedir. Bu nesne LDAP ortamında saklandığı için, testQueue nesne adı bir önekle girilirse de, araç LDAP adlandırma kuralına uygunluğu sağlamak için otomatik olarak bir nesne ekler. DISPLAY Q(testQueue) komutunun gönderilmesi de bu önekin eklenmesine neden olur.

```
V9.3.0 V9.3.0 JM 3.0
InitCtx> DEFINE Q(testQueue)
InitCtx> DISPLAY CTX
Contents of InitCtx
a cn=testQueue          com.ibm.mq.jakarta.jms.MQQueue
1 Object(s)
0 Context(s)
1 Binding(s), 1 Administered
```

```
JMS 2.0
InitCtx> DEFINE Q(testQueue)
InitCtx> DISPLAY CTX
Contents of InitCtx
a cn=testQueue          com.ibm.mq.jms.MQQueue
1 Object(s)
0 Context(s)
1 Binding(s), 1 Administered
```

## **JMS nesnesi yaratılırken örnek hata koşulları**

Bir nesne oluşturduğunuzda bir dizi genel hata koşulu ortaya çıkabilir.

Aşağıda aşağıdaki hata koşullarına ilişkin örnekler verilmiştir:

### **CipherSpec , CipherSuite ile eşlendi**

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SSLCIPHERSUITE(RC4_MD5_US)
WARNING: Converting CipherSpec RC4_MD5_US to
CipherSuite SSL_RSA_WITH_RC4_128_MD5
```

## Nesne için geçersiz özellik

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PRIORITY(4)
Unable to create a valid object, please check the parameters supplied
Invalid property for a QCF: PRI
```

## Özellik değeri için geçersiz tip

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) CCSID(english)
Unable to create a valid object, please check the parameters supplied
Invalid value for CCS property: English
```

## Özellik çakışması-istemci/bintanımları

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) HOSTNAME(polaris.hursley.ibm.com)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: Client-bindings attribute clash
```

## Özellik çakışması-Başlatma işleminden çık

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SECEXITINIT(initStr)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: ExitInit string supplied
without Exit string
```

## Özellik değeri geçerli aralığın dışında

```
InitCtx/cn=Trash> DEFINE Q(testQ) PRIORITY(12)
Unable to create a valid object, please check the parameters supplied
Invalid value for PRI property: 12
```

## Bilinmeyen özellik

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PIZZA(ham and mushroom)
Unable to create a valid object, please check the parameters supplied
Unknown property: PIZZA
```

Aşağıda, bir JMS uygulamasından JNDI yönetimli nesnelere aranırken Windows ' da ortaya çıkabilecek hata koşulları örnekleri verilmiştir.

1. WebSphere JNDI sağlayıcısını ( com.ibm.websphere.naming.WsnInitialContextFactory) kullanıyorsanız, alt bağlamlarda tanımlanan denetlenen nesnelere erişmek için bir eğik çizgi (/) kullanmalısınız; örneğin, jms/MyQueueAd. Ters eğik çizgi (\) kullanırsanız, InvalidNamekural dışı durumu yayınlanır.
2. Oracle JNDI sağlayıcısını ( com.sun.jndi.fscontext.RefFSContextFactory) kullanıyorsanız, alt bağlamlarda tanımlanan yönetilen nesnelere erişmek için bir ters eğik çizgi (\) kullanmanız gerekir; örneğin, ctx1\\fred. Eğik çizgi (/) kullanırsanız, NameNotFoundException yayınlanır.

## JMS nesnelere yapılandırma

Dizin ad alanında denetlenen nesnelere işlemek için ALTER, DEFINE, DISPLAY, DELETE, COPYve MOVE fiillerini kullanabilirsiniz.

## Bu görev hakkında

Çizelge 39 sayfa 679 içinde bu fiillerin kullanımını özetlenmektedir. *TYPE* yerine, “Yönetim araçlarını kullanarak JMS ve Jakarta Messaging nesnelere yapılandırma” sayfa 669 içinde açıklandığı gibi, gerekli yönetilen nesneyi gösteren anahtar sözcüğü kullanın.

Çizelge 39. Denetlenen nesnelere işlemek için kullanılan komutların sözdizimi ve açıklaması	
Komut sözdizimi	Açıklama
ALTER TYPE (ad) [ özellik ] *	Denetlenen nesnenin özelliklerini, sağlananlarla güncelleme girişiminde bulunur. Bir güvenlik ihlali varsa, belirtilen nesne bulunamazsa ya da sağlanan yeni özellikler geçersizse başarısız olur.
DEFINE TYPE (ad) [ özellik ] *	Sağlanan özelliklerle TYPE tipinde yönetilen bir nesne yaratmayı dener ve bunu geçerli bağlamda name adı altında saklar. Bir güvenlik ihlali varsa, belirtilen ad geçerli değilse ya da o adı taşıyan bir nesne varsa ya da sağlanan özellikler geçersizse başarısız olur.
DISPLAY TYPE (TIP) (ad)	Geçerli bağlamda name adı altında bağlı TYPE tipinde yönetilen nesnenin özelliklerini görüntüler. Nesne yoksa ya da bir güvenlik ihlali varsa başarısız olur.
DELETE TYPE (ad)	nameadını taşıyan TYPE adlı yönetilen nesneyi yürürlükteki bağlamdan kaldırmayı dener. Nesne yoksa ya da bir güvenlik ihlali varsa başarısız olur.
COPY TYPE (nameA) TYPE (nameB)	TYPE tipinde yönetilen nesnenin bir kopyasını oluşturur; nameA adıyla nameB adlı kopyayı adlandırır. Bu durum, yürürlükteki bağlamın kapsamı içinde oluşur. Kopyalanacak nesne yoksa, nameB adlı bir nesne varsa ya da bir güvenlik ihlali varsa başarısız olur.
TYPE (nameA) TYPE (nameB)	nameA adını taşıyan TYPE türündeki yönetilen nesneyi nameB adresine taşır (yeniden adlandırır). Bu durum, yürürlükteki bağlamın kapsamı içinde oluşur. Taşınacak nesne yoksa, nameB adlı bir nesne varsa ya da bir güvenlik ihlali varsa başarısız olur.

## JMS 2.0 WebSphere Application Server içinde JMS 2.0 kaynaklarının yapılandırılması

WebSphere Application Server içinde JMS 2.0 kaynaklarını yapılandırmak için yönetim konsolunu ya da wsadmin komutlarını kullanabilirsiniz.

### Başlamadan önce

**V 9.3.0** **V 9.3.0** **JM 3.0** IBM MQ 9.3, Jakarta Messaging 3.0 ürününü desteklese de, WebSphere Application Server şu anda eşdeğer bir desteğe sahip değildir. Bu nedenle, WebSphere Application Server içinde Java Message Service 2.0 kaynaklarını yapılandırabilirsiniz.

### Bu görev hakkında

Java Message Service 2.0 uygulamaları genellikle uygulamanın JMS sağlayıcısına ve eriştiği hedeflere nasıl bağlandığını açıklayan, dışarıdan yapılandırılmış nesnelere dayanır. JMS uygulamaları yürütüm sırasında aşağıdaki nesne tiplerine erişmek için Java Naming Directory Interface (JNDI) ögesini kullanır:

- Etkinleştirme belirtileri ( Java EE uygulama sunucuları tarafından kullanılır)
- Birleşik bağlantı fabrikaları ( JMS 1.1 ve sonrası ile, etki alanından bağımsız (birleşik) bağlantı fabrikaları, etki alanına özgü kuyruk bağlantı fabrikalarına ve konu bağlantı fabrikalarına tercih edilir)
- Konu bağlantı üreticileri ( JMS 1.0 uygulamaları tarafından kullanılır)
- Kuyruk bağlantısı üreticileri ( JMS 1.0 uygulamaları tarafından kullanılır)

- Kuyruklar
- Konular

WebSphere Application Serverindeki IBM MQ ileti alışverişi sağlayıcısı aracılığıyla Java Message Service (JMS) ileti sistemi uygulamaları, IBM MQ sisteminizi JMS ileti sistemi kaynaklarının dış sağlayıcısı olarak kullanabilir. Bu yaklaşımı etkinleştirmek için, WebSphere Application Server içindeki IBM MQ ileti alışverişi sağlayıcısını, IBM MQ ağındaki herhangi bir kuyruk yöneticisine bağlanmak üzere JMS kaynaklarını tanımlayacak şekilde yapılandırın.

Uygulamalar (örneğin, kuyruk bağlantısı üreticileri) için IBM MQ kaynaklarını yapılandırmak ve JMS hedefleriyle ilişkili iletileri ve abonelikleri yönetmek için WebSphere Application Server komutunu kullanabilirsiniz. IBM MQ aracılığıyla güvenliği yönetebilirsiniz.

### İlgili görevler

[IBM MQ ve WebSphere Application Server ' yi birlikte kullanma](#)

### WebSphere Application Server Konular

[IBM MQ ileti alışverişi sağlayıcısı kullanılarak birlikte işlem](#)

[IBM MQ ileti sistemi sağlayıcısıyla ileti alışverişini yönetme](#)

[Yönetim konsolu pano adlarının komut adlarıyla ve IBM MQ adlarıyla eşlenmesi](#)

## JMS 2.0 Yönetim konsolunu kullanarak JMS 2.0 kaynaklarının yapılandırılması

IBM MQ JMS sağlayıcısına ilişkin etkinleştirme belirteçlerini, bağlantı üreticilerini ve hedefleri yapılandırmak için WebSphere Application Server yönetim konsolunu kullanabilirsiniz.

### Bu görev hakkında

Aşağıdaki kaynaklardan herhangi birini oluşturmak, görüntülemek ya da değiştirmek için WebSphere Application Server yönetim konsolunu kullanabilirsiniz:

- Etkinleştirme belirteçleri
- Etki alanından bağımsız bağlantı üreticileri (JMS 1.1 ya da üstü)
- Kuyruk bağlantısı üreticileri
- Konu bağlantı üreticileri
- Kuyruklar
- Konular

Aşağıdaki adımlarda, JMS kaynaklarını IBM MQ ileti sistemi sağlayıcısıyla kullanmak üzere yapılandırmak için yönetim konsolunu kullanma yöntemlerine ilişkin bir genel bakış sağlanır. Her adım, daha fazla bilgi için başvurabileceğiniz WebSphere Application Server ürün belgelerindeki konunun adını içerir. IBM Documentation içinde bu konulara ilişkin bağlantılar için *İlgili bağlantılar* konusuna bakın.

Karma sürümlü bir WebSphere Application Server hücresinde, IBM MQ kaynaklarını tüm sürümlerin düğümlerinde denetleyebilirsiniz. Ancak, bazı özellikler tüm sürümlerde kullanılamaz. Bu durumda, denetim konsolunda yalnızca o düğümün özellikleri görüntülenir.

### Yordam

IBM MQ ileti sistemi sağlayıcısıyla kullanmak üzere bir etkinleştirme belirtimi oluşturmak ya da yapılandırmak için:

- Bir etkinleştirme belirtimi oluşturmak için Create IBM MQ JMS kaynak sihirbazını kullanın. Etkinleştirme belirtimine ilişkin tüm ayrıntıları belirtmek için sihirbazı kullanabilir ya da bir istemci kanal tanımlama çizelgesi (CCDT) kullanarak IBM MQ için bağlantı ayrıntılarını belirtmeyi seçebilirsiniz. Sihirbazı kullanarak bağlantı ayrıntılarını belirtirken, anasistem ve kapı bilgilerini ayrı olarak girmeyi ya da çok eşgörünümü bir kuyruk yöneticisi kullanıyorsanız, anasistem ve kapı bilgilerini bağlantı

adı listesi biçiminde girmeyi seçebilirsiniz. Daha fazla bilgi için *IBM MQ ileti alışverişi sağlayıcısı için etkinleştirme belirtimi oluşturmabaşlıklı* konuya bakın.

- Bir etkinleştirme belirtiminin yapılandırma özelliklerini görüntülemek ya da değiştirmek için yönetim konsolu IBM MQ ileti sistemi sağlayıcısı bağlantı üreticisi ayarları panosunu kullanın.

Bu yapılandırma özellikleri, ilişkili kuyruklar ve konularla bağlantıların nasıl yaratılacağını denetler. Daha fazla bilgi için *IBM MQ ileti sistemi sağlayıcısı için etkinleştirme belirtimini yapılandırmabaşlıklı* konuya bakın.

IBM MQ ileti sistemi sağlayıcısıyla kullanılmak üzere birleşik bağlantı üreticisi, kuyruk bağlantısı üreticisi ya da konu bağlantısı üreticisi yaratmak ya da yapılandırmak için:

- Bir bağlantı üreticisi yaratmak için, önce yaratmak istediğiniz bağlantı üreticisi tipini seçin, ardından ayrıntıları belirtmek için IBM MQ JMS kaynak yaratma sihirbazını kullanın.
  - JMS uygulamanızın yalnızca noktadan noktaya iletişim ileti sistemi kullanması amaçlanıyorsa, noktadan noktaya iletişim ileti sistemi için özel olarak noktadan noktaya iletişim için bağlantı yaratmak üzere kullanılacak noktadan noktaya iletişim ileti sistemi etki alanı için etki alanına özgü bir bağlantı üreticisi yaratın.
  - JMS uygulamanızın yalnızca yayınlama/abone olma ileti sistemini kullanması amaçlanıyorsa, özellikle yayınlama/abone olma ileti sistemi için bağlantı yaratmak üzere kullanılacak, yayınlama/abone olma ileti sistemi etki alanı için etki alanına özgü bir bağlantı üreticisi yaratın.
  - JMS 1.1 ya da daha sonraki düzeyler için, hem noktadan noktaya ileti sistemi hem de yayınlama/abone olma ileti sistemi için kullanılacak etki alanından bağımsız bir bağlantı üreticisi yaratın; böylece, uygulamanız aynı hareket altında hem noktadan noktaya iletişim hem de yayınlama/abone olma işlerini gerçekleştirebilir.

Bağlantı üreticisine ilişkin tüm ayrıntıları belirtmek için sihirbazı kullanmayı seçebilir ya da bir istemci kanal tanımlama çizelgesi (CCDT) kullanarak IBM MQ için bağlantı ayrıntılarını belirtmeyi seçebilirsiniz. Sihirbazı kullanarak bağlantı ayrıntılarını belirtirken, anasistem ve kapı bilgilerini ayrı olarak girmeyi ya da çok eşgörünümlü bir kuyruk yöneticisi kullanıyorsanız, anasistem ve kapı bilgilerini bağlantı adı listesi biçiminde girmeyi seçebilirsiniz. Daha fazla bilgi için *IBM MQ ileti alışverişi sağlayıcısı için bağlantı üreticisi oluşturmabaşlıklı* konuya bakın.

Bir bağlantı üreticisinin yapılandırma özelliklerini görüntülemek ya da değiştirmek için:

- Yapılandırmak istediğiniz bağlantı üreticisi tipine ilişkin denetim konsolu bağlantı üreticisi ayarları panosunu kullanın.

Yapılandırma özellikleri, ilişkili kuyruklar ve konularla bağlantıların nasıl yaratıldığını denetler. Daha fazla bilgi için bkz. *IBM MQ ileti alışverişi sağlayıcısı için kaynak grubu üreticisi yapılandırılması* ya da *IBM MQ ileti alışverişi sağlayıcısı için kuyruk derlemi üreticisi yapılandırılması* ya da *IBM MQ İleti alışverişi sağlayıcısı için konu derlemi üreticisi yapılandırılması*.

IBM MQ ileti alışverişi sağlayıcısıyla noktadan noktaya ileti alışverişi için bir JMS kuyruk hedefi yapılandırmak üzere:

- Aşağıdaki özellik tiplerini tanımlamak için yönetim konsolu IBM MQ ileti sistemi sağlayıcısı kuyruk ayarları panosunu kullanın:
  - Yönetim ve IBM MQ kuyruk özellikleri de içinde olmak üzere genel özellikler.
  - Kuyruğu barındıran kuyruk yöneticisine nasıl bağlanılacağını belirleyen bağlantı özellikleri.
  - IBM MQ ileti sistemi sağlayıcısı hedeflerine yapılan bağlantıların davranışını denetleyen gelişmiş özellikler.
  - Kuyruk hedefine ilişkin herhangi bir özel özellik.

Daha fazla bilgi için *IBM MQ ileti alışverişi sağlayıcısı için kuyruk yapılandırmabaşlıklı* konuya bakın.

IBM MQ ileti alışverişi sağlayıcısıyla yayınlama/abone olma ileti sistemi için bir JMS konu hedefi yaratmak ya da yapılandırmak için:

- Aşağıdaki özellik tiplerini tanımlamak için IBM MQ ileti alışverişi sağlayıcısı konu ayarları panosunu kullanın:
  - Yönetim ve IBM MQ konu özellikleri de içinde olmak üzere genel özellikler.

- IBM MQ ileti sistemi sağlayıcısı hedeflerine yapılan bağlantıların davranışını denetleyen gelişmiş özellikler.
- Kuyruk hedefine ilişkin herhangi bir özel özellik.

Daha fazla bilgi için *IBM MQ ileti sistemi sağlayıcısı için bir konuyu yapılandırmabaşlıklı* konuya bakın.

### İlgili kavramlar

[“Çok eşgörünümlü kuyruk yöneticileri” sayfa 494](#)

Çok eşgörünümlü kuyruk yöneticileri, farklı sunucularda yapılandırılan aynı kuyruk yöneticisinin eşgörünümleridir. Kuyruk yöneticisinin bir yönetim ortamı etkin yönetim ortamı olarak tanımlandı ve başka bir yönetim ortamı yedek yönetim ortamı olarak tanımlandı. Etkin yönetim ortamı başarısız olursa, çok eşgörünümlü kuyruk yöneticisi yedek sunucuda otomatik olarak yeniden başlatılır.

### İlgili görevler

[“İkili biçim CCDT ' nin yapılandırılması” sayfa 42](#)

İstemci kanal tanımlama çizelgesi (CCDT), kuyruk yöneticisine bağlanmak için istemci uygulamaları tarafından kullanılan kanal tanımlarını ve kimlik doğrulama bilgilerini belirler. Multiplatforms üzerinde, kuyruk yöneticisi yaratıldığında varsayılan ayarları içeren ikili bir CCDT otomatik olarak yaratılır. İkili CCDT ' yi güncellemek için **runmqsc** komutunu kullanın.

[“Yayınlama/abone olma ileti sisteminin yapılandırılması” sayfa 421](#)

Kuyruğa alınan yayınlama/abone olma durumunu başlatabilir, durdurabilir ve görüntüleyebilirsiniz. Ayrıca, akışları ekleyebilir ve kaldırabilir ve bir aracı sıradüzenine kuyruk yöneticileri ekleyebilir ve silebilirsiniz.

### WebSphere Application Server Konular

[IBM MQ ileti alışverişi sağlayıcısı etkinleştirme belirtileri](#)

[IBM MQ ileti alışverişi sağlayıcısı için etkinleştirme belirtimi oluşturma](#)

[IBM MQ ileti alışverişi sağlayıcısı için etkinleştirme belirtiminin yapılandırılması](#)

[IBM MQ ileti alışverişi sağlayıcısı için bağlantı üreticisi yaratılması](#)

[IBM MQ ileti sistemi sağlayıcısı için birleşik bağlantı üreticisinin yapılandırılması](#)

[IBM MQ ileti alışverişi sağlayıcısı için kuyruk bağlantısı üreticisinin yapılandırılması](#)

[IBM MQ ileti alışverişi sağlayıcısı için konu bağlantısı üreticisinin yapılandırılması](#)

[IBM MQ ileti alışverişi sağlayıcısı için kuyruk yapılandırılması](#)

[IBM MQ ileti alışverişi sağlayıcısı için konu yapılandırılması](#)

## **JMS 2.0 wsadmin komut dosyası oluşturma komutlarını kullanarak JMS 2.0 kaynaklarının yapılandırılması**

JMS etkinleştirme belirtileri, bağlantı üreticileri, kuyruklar ve konularla ilgili bilgileri oluşturmak, değiştirmek, silmek ya da göstermek için WebSphere Application Server wsadmin komut dosyası oluşturma komutlarını kullanabilirsiniz. IBM MQ kaynak bağdaştırıcısına ilişkin ayarları görüntüleyebilir ve yönetebilirsiniz.

### Bu görev hakkında

Aşağıdaki adımlarda, JMS kaynaklarını IBM MQ ileti sistemi sağlayıcısıyla kullanmak üzere yapılandırmak için WebSphere Application Server wsadmin komutlarını kullanma yöntemlerine ilişkin bir genel bakış sağlanır. Bu komutların nasıl kullanılacağına ilişkin ek bilgi için WebSphere Application Server ürün belgelerine ilişkin bağlantılar için *İlgili bağlantılar* konusuna bakın.

Bir komutu çalıştırmak için, wsadmin komut dosyası oluşturma istemcisinin AdminTask nesnesini kullanın.

Yeni bir nesne yaratmak ya da değişiklik yapmak için bir komut kullandıktan sonra, ana yapılandırmada yaptığınız değişiklikleri kaydedin. Örneğin, aşağıdaki komutu kullanın:

```
AdminConfig.save()
```

Kullanılabilir IBM MQ ileti sistemi sağlayıcısı yönetim komutlarının ve her komutun kısa bir açıklamasını görmek için, wsadmin komut isteminde şu komutu girin:

```
print AdminTask.help('WMQAdminCommands')
```

Verili bir komutla ilgili genel bakış yardımını görmek için, wsadmin komut isteminde aşağıdaki komutu girin:

```
print AdminTask.help('command_name')
```

## Yordam

Bir komutun verildiği kapsamda tanımlanan tüm IBM MQ ileti sistemi sağlayıcı kaynaklarını listelemek için aşağıdaki komutları kullanın.

- Etkinleştirme belirtilerini listelemek için **listWMQActivationSpecs** komutunu kullanın.
- Bağlantı üreticilerini listelemek için **listWMQConnectionFactory** komutunu kullanın.
- Kuyruk tipi hedeflerini listelemek için **listWMQQueues** komutunu kullanın.
- Konu tipi hedeflerini listelemek için **listWMQTopics** komutunu kullanın.

Belirli bir kapsamda IBM MQ ileti alışverişi sağlayıcısı için bir JMS kaynağı yaratmak üzere aşağıdaki komutları kullanın.

- Bir etkinleştirme belirtimi oluşturmak için **createWMQActivationSpec** komutunu kullanın.  
Bağlantı kurmak için kullanılacak tüm değişirgeleri belirterek bir etkinleştirme belirtimi yaratabilir ya da bağlantı kurulacak kuyruk yöneticisini bulmak için bir istemci kanal tanımlama çizelgesi (CCDT) kullanması için etkinleştirme belirtimini yaratabilirsiniz.
- Bir bağlantı üreticisi yaratmak için, yaratmak istediğiniz bağlantı üreticisinin tipini belirtmek üzere **-type** değişirgesini kullanarak **createWMQConnectionFactory** komutunu kullanın:
  - JMS uygulamanızın yalnızca noktadan noktaya iletişim ileti sistemi kullanması amaçlanıyorsa, noktadan noktaya iletişim ileti sistemi için özel olarak noktadan noktaya iletişim için bağlantı yaratmak üzere kullanılabilir noktadan noktaya iletişim ileti sistemi etki alanı için etki alanına özgü bir bağlantı üreticisi yaratın.
  - JMS uygulamanızın yalnızca yayınlama/abone olma ileti sistemini kullanması amaçlanıyorsa, özellikle yayınlama/abone olma ileti sistemi için bağlantı yaratmak üzere kullanılabilir, yayınlama/abone olma ileti sistemi etki alanı için etki alanına özgü bir bağlantı üreticisi yaratın.
  - JMS 1.1 ya da daha sonraki düzeyler için, hem noktadan noktaya ileti sistemi hem de yayınlama/abone olma ileti sistemi için kullanılabilir etki alanından bağımsız bir bağlantı üreticisi yaratın; böylece, uygulamanız aynı hareket altında hem noktadan noktaya iletişim hem de yayınlama/abone olma işlerini gerçekleştirebilir.

Varsayılan tip, etki alanından bağımsız bağlantı üreticidir.

- Bir kuyruk tipi hedefi yaratmak için **createWMQQueue** komutunu kullanın.
- Konu tipi hedefi yaratmak için **createWMQTopic** komutunu kullanın.

Belirli bir kapsamda IBM MQ ileti alışverişi sağlayıcısına ilişkin bir JMS kaynağını değiştirmek için aşağıdaki komutları kullanın.

- Bir etkinleştirme belirtimini değiştirmek için **modifyWMQActivationSpec** komutunu kullanın.  
Bir etkinleştirme belirtiminin tipini değiştiremezsiniz. Örneğin, tüm yapılandırma bilgilerini el ile girdiğiniz ve CCDT kullanacak şekilde değiştirdiğiniz bir etkinleştirme belirtimi oluşturamazsınız.
- Bir bağlantı üreticisini değiştirmek için **modifyWMQConnectionFactory** komutunu kullanın.
- Bir kuyruk tipi hedefini değiştirmek için **modifyWMQQueue** komutunu kullanın.
- Bir konu tipi hedefini değiştirmek için **modifyWMQTopic** komutunu kullanın.

Belirli bir kapsamda IBM MQ ileti sistemi sağlayıcısına ilişkin bir JMS kaynağını silmek için aşağıdaki komutları kullanın.

- Bir etkinleştirme belirtimini silmek için **deleteWMQActivationSpec** komutunu kullanın.
- Bir bağlantı üreticisini silmek için **deleteWMQConnectionFactory** komutunu kullanın.
- Bir kuyruk tipi hedefini silmek için **deleteWMQQueue** komutunu kullanın.
- Bir konu tipi hedefini silmek için **deleteWMQTopic** komutunu kullanın.

Belirli bir IBM MQ ileti alışverişi sağlayıcısı kaynağıyla ilgili bilgileri görüntülemek için aşağıdaki komutları kullanın.

- Belirli bir etkinleştirme belirtimiyle ilişkili tüm parametreleri ve değerlerini görüntülemek için **showWMQActivationSpec** komutunu kullanın.
- Belirli bir bağlantı üreticisiyle ilişkili tüm parametreleri ve değerlerini görüntülemek için **showWMQConnectionFactory** komutunu kullanın.
- Belirli bir kuyruk tipi hedefiyle ilişkili tüm parametreleri ve değerlerini görüntülemek için **showWMQQueue** komutunu kullanın.
- Bir konu tipi hedefiyle ilişkili tüm parametreleri ve değerlerini görüntülemek için **deleteWMQTopic** komutunu kullanın.

IBM MQ kaynak bağdaştırıcısına ya da IBM MQ ileti alışverişi sağlayıcısına ilişkin ayarları yönetmek için aşağıdaki komutları kullanın.

- Belirli bir kapsamda kurulu IBM MQ kaynak bağdaştırıcısının ayarlarını yönetmek için **manageWMQ** komutunu kullanın.
- **manageWMQ** komutuyla ayarlanabilecek tüm parametreleri ve değerlerini görüntülemek için **showWMQ** komutunu kullanın. Bu ayarlar, IBM MQ kaynak bağdaştırıcısı ya da IBM MQ ileti alışverişi sağlayıcısıyla ilgilidir. **showWMQ** komutu, IBM MQ kaynak bağdaştırıcısında ayarlanan özel özellikleri de gösterir.

### İlgili kavramlar

[“Çok eşgörünümlü kuyruk yöneticileri” sayfa 494](#)

Çok eşgörünümlü kuyruk yöneticileri, farklı sunucularda yapılandırılan aynı kuyruk yöneticisinin eşgörünümleridir. Kuyruk yöneticisinin bir yönetim ortamı etkin yönetim ortamı olarak tanımlandı ve başka bir yönetim ortamı yedek yönetim ortamı olarak tanımlandı. Etkin yönetim ortamı başarısız olursa, çok eşgörünümlü kuyruk yöneticisi yedek sunucuda otomatik olarak yeniden başlatılır.

### İlgili görevler

[“İkili biçim CCDT ' nin yapılandırılması” sayfa 42](#)

İstemci kanal tanımlama çizelgesi (CCDT), kuyruk yöneticisine bağlanmak için istemci uygulamaları tarafından kullanılan kanal tanımlarını ve kimlik doğrulama bilgilerini belirler. Multiplatforms üzerinde, kuyruk yöneticisi yaratıldığında varsayılan ayarları içeren ikili bir CCDT otomatik olarak yaratılır. İkili CCDT ' yi güncellemek için **runmqsc** komutunu kullanın.

[“Yayınlama/abone olma ileti sisteminin yapılandırılması” sayfa 421](#)

Kuyruğa alınan yayınlama/abone olma durumunu başlatabilir, durdurabilir ve görüntüleyebilirsiniz. Ayrıca, akışları ekleyebilir ve kaldırabilir ve bir aracı sıradüzenine kuyruk yöneticileri ekleyebilir ve silebilirsiniz.

### WebSphere Application Server Konular

[\*\*createWMQActivationSpec\*\* DELETE ...](#)

[\*\*createWMQConnectionFactory\*\* DELETE ...](#)

[\*\*createWMQQueue\*\* DELETE ...](#)

[\*\*createWMQTopic\*\* DELETE ...](#)

[\*\*deleteWMQActivationSpec\*\* DELETE ...](#)

[\*\*deleteWMQConnectionFactory\*\* DELETE ...](#)

[\*\*deleteWMQQueue\*\* DELETE ...](#)

[\*\*deleteWMQTopic\*\* DELETE ...](#)

[\*\*listWMQActivationSpecs\*\* DELETE ...](#)

[\*\*listWMQConnectionFactories\*\* DELETE ...](#)

[\*\*listWMQQueues\*\* DELETE ...](#)

[\*\*listWMQTopics\*\* DELETE ...](#)



[modifyWMQActivationSpec DELETE ...](#)  
[modifyWMQConnectionFactory DELETE ...](#)  
[modifyWMQQueue DELETE ...](#)  
[modifyWMQTopic DELETE ...](#)  
[showWMQActivationSpec DELETE ...](#)  
[showWMQConnectionFactory DELETE ...](#)  
[showWMQQueue DELETE ...](#)  
[showWMQTopic DELETE ...](#)  
[showWMQ DELETE ...](#)  
[manageWMQ DELETE ...](#)

## **JMS 2.0** JMS 2.0 paylaşılan aboneliklerinin kullanılması

WebSphere Application Server traditional 9.0 içinde, IBM MQ 9.0 ile JMS 2.0 paylaşılan aboneliklerini yapılandırabilir ve kullanabilirsiniz.

### **Bu görev hakkında**

JMS 2.0 belirtimi, tek bir aboneliğin bir ya da daha fazla tüketici tarafından açılmasını sağlayan paylaşılan abonelik kavramını tanıttı. Mesajlar tüm bu tüketiciler arasında paylaşılır. Aynı kuyruk yöneticisine bağlandıkları sürece bu tüketicilerin bulunduğu yerde herhangi bir kısıtlama yoktur.

Paylaşılan Abonelikler, artık paylaşılmayan abonelikler olarak anılanlarla aynı anlambilimle dayanıklı ya da kalıcı olmayan olabilir.

Bir tüketicinin hangi aboneliğin kullanılacağını belirleyebilmesi için bir abonelik adı sağlaması gerekir. Bu, paylaşılmayan sürekli aboneliklere benzer, ancak paylaşılan aboneliğin gerekli olduğu tüm durumlarda abonelik adı gereklidir. Ancak, kalıcı bir paylaşılan-subscription; sağlanabilir, ancak zorunlu değildir. clientID, kalıcı bir paylaşılan-subscription'da;

Paylaşılan abonelikler bir yükleme dengeleme mekanizması olarak düşünülebilir, ancak IBM MQ ve JMS 2.0 belirtimlerinde iletilerin tüketiciler arasında nasıl dağıtıldığına ilişkin herhangi bir taahhüt yoktur.

WebSphere Application Server traditional 9.0 içinde bir IBM MQ 9.0 kaynak bağdaştırıcısı önceden kurulmuştur.

Aşağıdaki adımlarda, WebSphere Application Server traditional yönetim konsolunu kullanarak paylaşılan bir sürekli ya da paylaşılan bir kalıcı olmayan aboneliği kullanmak için bir etkinleştirme belirtiminin nasıl yapılandırılacağı gösterilir.

### **Yordam**

Önce JNDI içinde nesnelere oluşturun.

1. JNDI içinde olağan bir konu hedefi oluşturun (bkz. [“Yönetim konsolunu kullanarak JMS 2.0 kaynaklarının yapılandırılması” sayfa 680](#)).
2. Etkinleştirme belirtimini oluşturun (bkz. [“Yönetim konsolunu kullanarak JMS 2.0 kaynaklarının yapılandırılması” sayfa 680](#)).

Etkinleştirme belirtimini tam olarak gereksinim duyduğunuz özelliklerle oluşturabilirsiniz. Sürekli abonelik kullanmak istiyorsanız, yaratma işleminde bu aboneliği seçebilir ve bir ad belirtebilirsiniz. Kalıcı olmayan bir abonelik kullanmak istiyorsanız, bu noktada bir ad belirtmezsiniz. Bunun yerine, abonelik adı için özel bir özellik oluşturmanız gerekir.

Gerekli özel özelliklerle oluşturduğunuz etkinleştirme belirtimini güncelleyin. Belirtmeniz gereken iki özel özellik vardır:

- Her durumda, bu etkinleştirme belirtiminin paylaşılan bir abonelik kullanması gerektiğini belirtmek için özel bir özellik oluşturmanız gerekir.
- Abonelik kalıcı olmayan olarak yaratıldıysa, abonelik adı özelliğinin özel bir özellik olarak ayarlanması gerekir.

Aşağıdaki tablo, her özel özellik için belirtebileceğiniz geçerli değeri göstermektedir:

Özellik Adı	Tip	Geçerli değerler
sharedSubscription	Dizgi	doğru, yanlış
subscriptionName	Dizgi	Sıfır uzunluklu olmayan java Dizesi

3. **Etkinleştirme belirtimi derlemi** formunda görüntülenen listeden etkinleştirme belirtimini seçin. Etkinleştirme belirtimine ilişkin ayrıntılar, **IBM MQ ileti alışverişi sağlayıcısı etkinleştirme belirtimi ayarları** formunda görüntülenir.
4. **IBM MQ ileti sistemi sağlayıcısı etkinleştirme belirtimi ayarları** formunda **Özel özellikler** i tıklatın. **Özel özellikler** formu görüntülenir.
5. Kalıcı olmayan bir abonelik kullanıyorsanız, subscriptionName özel özelliğini oluşturun.

Etkinleştirme belirtiminin **Özel özellikler** panosunda **Yenidüğmesini** tıklatın ve aşağıdaki ayrıntıları girin:

#### Ad

Bu durumda subscriptionNameolan özel özelliğin adı.

#### Değer

Özel özelliğin değeri. JNDI adlarını **Değer** alanında kullanabilirsiniz; örneğin, WASSharedSubOne.

#### Tip

Özel özelliğin tipi. Listedeki özel özellik tipini seçin; bu durumda java.Lang.Stringolmalıdır.

6. Hem paylaşılan sürekli hem de paylaşılan kalıcı olmayan abonelik için sharedSubscription özel özelliğini oluşturun.

Etkinleştirme belirtiminin **Özel özellikler** panosunda **Yenidüğmesini** tıklatın ve aşağıdaki ayrıntıları girin:

#### Ad

Bu durumda sharedSubscriptionolan özel özelliğin adı.

#### Değer

Özel özelliğin değeri. Etkinleştirme belirtiminin paylaşılan bir abonelik kullandığını belirtmek için değeri trueolarak ayarlayın. Daha sonra bu etkinleştirme belirtimi için paylaşılan abonelik kullanmayı durdurmak istiyorsanız, bu özel özelliğin değerini falseolarak ayarlayarak bunu yapabilirsiniz.

#### Tip

Özel özelliğin tipi. Listedeki özel özellik tipini seçin; bu durumda java.Lang.Stringolmalıdır.

7. Özellikler ayarlandığında uygulama sunucusunu yeniden başlatın.

Daha sonra, etkinleştirme belirtileri için ileti odaklı Bean 'ler, iletiler geldiğinde yönlendirilir, ancak yalnızca MDB 'ler gönderilen iletileri paylaşır.

### İlgili kavramlar

[Klonlanan ve paylaşılan abonelikler](#)

[Abonelik dayanıklılığı](#)

### İlgili görevler

[Kaynak bağdaştırıcısının gelen iletişim için yapılandırılması](#)

**WebSphere Application Server traditional 9.0 ile ilgili bilgiler**

[IBM MQ ileti alışverişi sağlayıcısı için konu yapılandırılması](#)

[IBM MQ ileti alışverişi sağlayıcısı etkinleştirme belirtileri](#)

[IBM MQ ileti alışverişi sağlayıcısı için etkinleştirme belirtimi oluşturma](#)

[IBM MQ ileti alışverişi sağlayıcısı için etkinleştirme belirtiminin yapılandırılması](#)

## JMS 2.0 JMS 2.0 ConnectionFactory ve Hedef Arama özelliklerinin kullanılması

WebSphere Application Server traditional 9.0 içinde, bir etkinleştirme belirtiminin ConnectionFactoryLookup ve DestinationLookup özellikleri, diğer etkinleştirme belirtimi özellikleri tercihinde kullanılacak yönetilen bir nesnenin JNDI adıyla birlikte sağlanabilir.

### Bu görev hakkında

JMS 2.0 belirtimi, iletiyle yönlendirilen Bean 'leri (MDB) sürücüyü yönlendirmek için kullanılan etkinleştirme belirtimine ilişkin iki ek özelliği belirtir. Daha önce, her bir satıcı, bir ileti sistemi sistemine bağlanmak ve iletilerin hangi hedeften gönderileceğini tanımlamak için gerekli ayrıntıları sağlamak üzere etkinleştirme belirtimine ilişkin özel özellikleri belirtmesi gerekiyordu.

Şimdi standart connectionFactoryLookup ve destinationLookup özellikleri, arama ve kullanım için ilgili nesnenin JNDI adını vermek için kullanılabilir. WebSphere Application Server traditional 9.0 içinde bir IBM MQ 9.0 kaynak bağdaştırıcısı önceden kurulur.

Aşağıdaki adımlarda, WebSphere Application Server traditional denetim konsolunu kullanarak bu iki özelliğin nasıl özelleştirileceği ve kullanılacağı gösterilir.

### Yordam

Önce JNDI içinde nesnelere oluşturun.

1. JNDI içinde ConnectionFactory 'yi normal olarak oluşturun (bkz. [“Yönetim konsolunu kullanarak JMS 2.0 kaynaklarının yapılandırılması” sayfa 680](#)).
2. Hedefi JNDI içinde normal olarak oluşturun (bkz. [“Yönetim konsolunu kullanarak JMS 2.0 kaynaklarının yapılandırılması” sayfa 680](#)).
3. Gerekli olan değerleri kullanarak etkinleştirme belirtimini oluşturun (bkz. [“Yönetim konsolunu kullanarak JMS 2.0 kaynaklarının yapılandırılması” sayfa 680](#)).

Etkinleştirme belirtimini tam olarak gereksinim duyduğunuz özelliklerle oluşturabilirsiniz. Ancak, aşağıdaki hususları göz önünde bulundurmanız gerekir:

- IBM MQ kaynak bağdaştırıcısının Java EE bağlantı üreticisi ve hedef arama özelliklerini kullanmasını istiyorsanız, etkinleştirme belirtimini oluşturduğunuzda kullanılan özellikler daha az ilgilidir (bkz. [ActivationSpec ConnectionFactoryLookup ve DestinationLookup özellikleri](#)).
- Ancak, bağlantı üreticisinde ya da hedefte tanımlı olmayan herhangi bir özellik, etkinleştirme belirtiminde belirtilmeye devam etmelidir. Bu nedenle, bağlantı tüketicisi özelliklerini ve ek özellikleri ve bir bağlantı gerçekten yaratıldığında kullanılan kimlik doğrulama bilgilerini tanımlamanız gerekir.
- Bağlantı üreticisinde tanımlanan özelliklerden ClientID özelliği özel işlemeye sahiptir. Bunun nedeni, ortak bir senaryonun birden çok etkinleştirme belirtimini içeren tek bir bağlantı üreticisi kullanmış olmasıdır. Bu, yönetimi basitleştirir, ancak JMS belirtimi benzersiz istemci tanıtıcıları gerektirir; bu nedenle, etkinleştirme belirtiminin ConnectionFactory' de ayarlanan herhangi bir değeri geçersiz kılma yeteneği olması gerekir. Etkinleştirme belirtiminde bir ClientID ayarlanmazsa, bağlantı üreticisinin herhangi bir değeri kullanılır.

“4” sayfa 687. adımda açıklandığı gibi WebSphere Application Server yönetim konsolunu kullanarak iki yeni özel özellik oluşturduğunuz etkinleştirme belirtimini güncelleyin ya da “5” sayfa 688. adımda açıklandığı gibi ek açıklamaları kullanın.

4. WebSphere Application Server yönetim konsolunda etkinleştirme belirtimini güncelleyin.

Bu iki özelliğin, etkinleştirme belirtiminin özel özellikler panosunda ayarlanması gerekir. Bu özellikler, ana etkinleştirme belirtimi panolarında ya da Etkinleştirme Belirtimi oluşturma sihirbazında yoktur.

- a) **Etkinleştirme belirtimi derlemi** formunda görüntülenen listeden etkinleştirme belirtimini seçin.

Etkinleştirme belirtimine ilişkin ayrıntılar, **IBM MQ ileti alışverişi sağlayıcısı etkinleştirme belirtimi ayarları** formunda görüntülenir.

- b) **IBM MQ ileti sistemi sağlayıcısı etkinleştirme belirtimi ayarları** formunda **Özel özellikler**' i tıklatın.

**Özel özellikler** formu görüntülenir.

- c) **Özel özellikler** formunda, her ikisi de java.lang.Stringtipinde iki yeni özel özellik yaratın.

Her bir durumda, **Yeni** ' i tıklatın ve özel özellik için aşağıdaki ayrıntıları girin:

#### **Ad**

Özel özelliğin adı; connectionFactoryLookup ya da destinationLookup.

#### **Değer**

Özel özelliğin değeri. **Değer** alanında JNDI adlarını kullanabilirsiniz; örneğin, QuoteCF ve QuoteQ.

#### **Tip**

Özel özelliğin tipi. Listedeki özel özellik tipini seçin; bu durumda java.lang.Stringolmalıdır.

Konuşlandırılan MDB şimdi bağlantı üreticisini ve hedefi yaratmak için bu değerleri kullanacak. MDB devreye alınırken, JNDI değer yapılandırmasını ayarlama gereksinimi yoktur.

5. Etkinleştirme belirtimi yerine ek açıklamalar kullanın.

Değerleri belirtmek için MDB kodundaki ek açıklamalar da kullanılabilir. Örneğin, JNDI adlar QuoteCF ve QuoteQkullanılarak kod şöyle görünür:

```
@MessageDriven(activationConfig = {
    @ActivationConfigProperty(propertyName = "destinationType" , propertyValue =
"javax.jms.Topic" ),
    @ActivationConfigProperty(propertyName = "destinationLookup" , propertyValue =
"QuoteQ" ),
    @ActivationConfigProperty(propertyName = "connectionFactoryLookup" , propertyValue
= "QuoteCF" )}, mappedName = "LookupMDB" )
@Transactional(TransactionalAttributeType.REQUIRED)
@TransactionManagement(TransactionManagementType.CONTAINER)
publicclass LookupMDB implements MessageListener {
```

### **İlgili görevler**

[Kaynak bağdaştırıcısının gelen iletişim için yapılandırılması](#)

### **WebSphere Application Server traditional 9.0 ile ilgili bilgiler**

[IBM MQ ileti sistemi sağlayıcısı için birleşik bağlantı üreticisinin yapılandırılması](#)

[IBM MQ ileti alışverişi sağlayıcısı için konu yapılandırılması](#)

[IBM MQ ileti alışverişi sağlayıcısı etkinleştirme belirtileri](#)

[IBM MQ ileti alışverişi sağlayıcısı için etkinleştirme belirtimi oluşturma](#)

[IBM MQ ileti alışverişi sağlayıcısı için etkinleştirme belirtiminin yapılandırılması](#)

[IBM MQ Messaging Provider JMS kaynakları için özel özelliklerin yapılandırılması](#)

## **WebSphere Application Server ' in en son kaynak bağdaştırıcısı bakım düzeyini kullanacak şekilde yapılandırılması**

WebSphere Application Server düzeltme paketlerini uyguladığınızda IBM MQ kaynak bağdaştırıcısının otomatik olarak kullanılabilir en son bakım düzeyine güncellenmesini sağlamak için, ortamınızdaki tüm sunucuları, her düğümün kuruluşuna uyguladığınız WebSphere Application Server düzeltme paketinde bulunan kaynak bağdaştırıcısının en son sürümünü kullanacak şekilde yapılandırabilirsiniz.

### **Başlamadan önce**

**Önemli:**

- **JM 3.0** WebSphere Application Server traditional şu anda Jakarta EE' yi desteklemez. Bkz. [IBM MQ kaynak bağıdaştırıcısı destek bildirimini](#).
- WebSphere Application Server 8.5 ya da daha önceki bir sürümü herhangi bir platformda kullanıyorsanız, IBM MQ 8.0 ya da sonraki kaynak bağıdaştırıcısını uygulama sunucusuna kurmayın. IBM MQ 8.0 ya da daha sonraki bir kaynak bağıdaştırıcısı yalnızca JMS 2.0' ı destekleyen bir uygulama sunucusuna konuşlandırılabilir. Ancak, WebSphere Application Server 8.5 ya da önceki sürümler yalnızca JMS 1.1' yi destekler.

## Bu görev hakkında

Aşağıdaki koşullardan herhangi biri yapılandırmanız için geçerse ve ortamınızdaki tüm sunucuları IBM MQ kaynak bağıdaştırıcısının en son sürümünü kullanacak şekilde yapılandırmak istiyorsanız bu görevi kullanın:

- Ortamınızdaki herhangi bir uygulama sunucusunun JVM günlükleri, WebSphere Application Server 7.0.0 Fix Pack 1 ya da üstü uygulandıktan sonra aşağıdaki IBM MQ kaynak bağıdaştırıcısı sürüm bilgilerini gösterir:  
WMSG1703I:RAR Uygulama Sürümü 7.0.0.0-k700-L080820
- Ortamınızdaki herhangi bir uygulama sunucusunun JVM günlükleri aşağıdaki girişi içerir:  
WMSG1625E: Saptamadı  
Belirtilen yoldaki IBM MQ ileti alışverişi sağlayıcısı kodu < null>
- Bir ya da daha çok düğüm, şu anda geçerli WebSphere Application Server bakım düzeyinde bulunan kaynak bağıdaştırıcısının en son sürümünün yerini alan IBM MQ kaynak bağıdaştırıcısının belirli bir bakım düzeyini kullanacak şekilde el ile güncellendi.

Örneklerin başvurduğu *profile\_root* dizini, WebSphere Application Server profilinin ana dizinidir; örneğin, C:\Program Files\IBM\WebSphere\AppServer1.

Ortamınızdaki tüm hücreler ve tek sunucu kuruluşları için aşağıdaki adımları gerçekleştirdiğinizde, yeni bir WebSphere Application Server düzeltme paketi uygulandığında sunucularınız otomatik olarak IBM MQ kaynak bağıdaştırıcısına bakım alır.

## Yordam

1. Uygulama sunucusunu başlatın. Profil bir ağ devreye alma yapılandırmasının parçasıysa, devreye alma yöneticisini ve tüm düğüm araçlarını başlatın. Profil bir yönetim aracı içeriyorsa, yönetim aracısını başlatın.
2. IBM MQ kaynak bağıdaştırıcısının bakım düzeyini denetleyin.
  - a) Bir komut istemi penceresi açın ve *profile\_root\bin* dizinine geçin.  
Örneğin, cd C:\Program Files\IBM\WebSphere\AppServer1\bingirin.
  - b) wsadmin.bat -lang jythonkomutunu girerek wsadmin aracını başlatın, ardından bunu yapmanız istenirse kullanıcı adınızı ve parolanızı girin.
  - c) Aşağıdaki komutu yazın ve daha sonra, Enter tuşuna iki kez basın:

```
wmqInfoMBeansUnsplit = AdminControl.queryNames("WebSphere:type=WmqInfo,*")
wmqInfoMBeansSplit = AdminUtilities.convertToList(wmqInfoMBeansUnsplit)
for wmqInfoMBean in wmqInfoMBeansSplit: print wmqInfoMBean; print AdminControl.invoke(wmqInfoMBean, 'getInfo', '')
```

Bu komutu Jacl 'de de çalıştırabilirsiniz. Bunun nasıl gerçekleştirileceğine ilişkin daha fazla bilgi için WebSphere Application Server ürün belgelerinde *Sunucuların kullanılabilir en son IBM MQ kaynak bağıdaştırıcısı bakım düzeyini kullanmasının sağlanması* başlıklı konuya bakın.

- d) Komutun görüntülenen çıkışında WMSG1703I iletisini bulun ve kaynak bağıdaştırıcısı düzeyini denetleyin.

Örneğin, WebSphere Application Server 7.0.1 Fix Pack 5 için ileti şöyle olmalıdır:

```
WMSG1703I: RAR somutlaması sürümü 7.0.1.3-k701-103-100812
```

Bu ileti, sürümün 7.0.1.3-k701-103-100812olduğunu gösterir; bu, bu düzeltme paketi için doğru kaynak bağdaştırıcısı düzeyidir. Ancak, bunun yerine aşağıdaki ileti görüntülenirse, kaynak bağdaştırıcısını WebSphere Application Server 7.0.1 Fix Pack 5 için doğru bakım düzeyine ayarlamamız gerekir.

WMSG1703I: RAR somutlaması sürümü 7.0.0.0-k700-L080820

3. Aşağıdaki Jython komut dosyasını `convertWMQRA.py` adlı bir dosyaya kopyalayın ve profil kök dizinine (örneğin, `C:\Program Files\IBM\WebSphere\AppServer1\bin`) kaydedin.

```
ras = AdminUtilities.convertToList(AdminConfig.list('J2CResourceAdapter'))

for ra in ras :
    desc = AdminConfig.showAttribute(ra, "description")
    if (desc == "WAS 7.0 Built In MQ Resource Adapter") or (desc == "WAS 7.0.0.1 Built In MQ Resource Adapter"):
        print "Updating archivePath and classpath of " + ra
        AdminConfig.modify(ra, [['archivePath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar]])
        AdminConfig.unsetAttributes(ra, ['classpath'])
        AdminConfig.modify(ra, [['classpath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar]])
        AdminConfig.save()
    #end if
#end for
```

**İpucu:** Dosyayı kaydederken, dosyanın metin dosyası yerine python dosyası olarak kaydedildiğinden emin olun.

4. Az önce oluşturduğunuz Jython komut dosyasını çalıştırmak için WebSphere Application Server `wsadmin` aracını kullanın.

Bir komut istemi açın ve WebSphere Application Server (örneğin, `C:\Program Files\IBM\WebSphere\AppServer1\bin`) dizinine ilişkin ana dizindeki `\bin` dizinine gidin ve aşağıdaki komutu yazın ve Return tuşuna basın:

```
wsadmin -lang jython -f convertWMQRA.py
```

Bunu yapmanız istenirse, kullanıcı adınızı ve parolanızı girin.

**Not:** Komut dosyasını bir ağ devreye alma yapılandırmasının parçası olan bir profil için çalıştırırsanız, komut dosyası o yapılandırmada güncellenmesi gereken tüm profilleri günceller. Önceden var olan yapılandırma dosyası tutarsızlıklarınız varsa, tam yeniden eşzamanlama gerekebilir.

5. Bir ağ devreye alma yapılandırmasında çalışıyorsanız, düğüm araçlarının tam olarak yeniden eşitlendiğinden emin olun. Daha fazla bilgi için, `wsadmin` komut dosyası oluşturma aracını kullanarak düğümlerin eşitlenmesi ya da düğümlerin eklenmesi, yönetilmesi ve kaldırılması başlıklı konuya bakın.
6. Profildeki tüm sunucuları durdurun. Profil bir ağ devreye alma yapılandırmasının parçasıysa, yapılandırmadaki küme üyelerini de durdurun, yapılandırmadaki tüm düğüm araçlarını durdurun ve konuşlandırma yöneticisini durdurun. Profil bir yönetim aracı içeriyorsa, yönetim aracısını durdurun.
7. `profile_root/bin` dizininden **osgiCfgInit** komutunu çalıştırın.  
`osgiCfgInit` komutu, OSGi yürütme ortamı tarafından kullanılan sınıf önbelleğini ilk durumuna getirir. Profil bir ağ devreye alma yapılandırmasının parçasıysa, yapılandırmanın parçası olan her profilin `profile_root/bin` dizininden **osgiCfgInit** komutunu çalıştırın.
8. Profildeki tüm sunucuları yeniden başlatın. Profil bir ağ devreye alma yapılandırmasının parçasıysa, yapılandırmadaki küme üyelerini de yeniden başlatın, yapılandırmadaki tüm düğüm araçlarını yeniden başlatın ve devreye alma yöneticisini yeniden başlatın. Profil bir yönetim aracı içeriyorsa, yönetim aracısını yeniden başlatın.
9. Kaynak bağdaştırıcısının doğru düzeyde olup olmadığını denetlemek için 2. adımı yineleyin.

## Sonraki adım

Bu konuda açıklanan adımları gerçekleştirdikten sonra sorunlarla karşılaşmaya devam ederseniz ve WebSphere Application Server yönetim konsolundaki JMS Sağlayıcı Ayarları panosunda IBM MQ kaynak bağdaştırıcısını ortamınızdaki düğümlerde güncellemek için **Kaynak bağdaştırıcısını güncelle** düğmesini kullandıysanız, [APAR PM10308](#) içinde açıklanan sorunla karşılaşmış olmanız olasıdır.

## İlgili kavramlar

IBM MQ kaynak bağdaştırıcısının kullanılması

### WebSphere Application Server 8.5.5 ile ilgili bilgiler

Sunucuların kullanılabilir en son IBM MQ kaynak bağdaştırıcısı bakım düzeyini kullandığından emin olunması

wsadmin komut dosyası oluşturma aracı kullanılarak düğümlerin eşitlenmesi

Düğüm ekleme, yönetme ve kaldırma

JMS Sağlayıcı ayarları

## JMS PROVIDERVERSION özelliğinin yapılandırılması

IBM MQ ileti alışverişi sağlayıcısının üç çalışma kipi vardır: Normal kip, kısıtlamalar içeren normal kip ve geçiş kipi. JMS **PROVIDERVERSION** özelliğini, bir JMS uygulamasının yayınlama ve abone olma için hangi kipleri kullanacağını seçecek şekilde ayarlayabilirsiniz.

### Bu görev hakkında

IBM MQ ileti alışverişi sağlayıcısı işletim kipinin seçilmesi, öncelikle PROVIDERVERSION bağlantı üreticisi özelliği ayarlanarak denetlenebilir. Bir kip belirtilmediyse, işlem kipi otomatik olarak seçilebilir.

**PROVIDERVERSION** özelliği, üç IBM MQ ileti alışverişi sağlayıcısı işletim kipi arasında fark yaratır:

#### IBM MQ ileti alışverişi sağlayıcısı normal kip

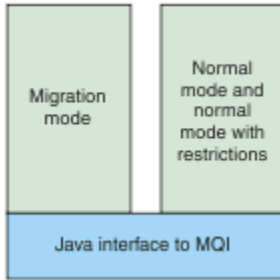
Normal kip, JMS' i gerçekleştirmek için IBM MQ kuyruk yöneticisinin tüm özelliklerini kullanır. Bu kip, JMS 2.0 API ve işlevselliğini kullanmak için eniyelenmiştir.

#### Kısıtlamalar içeren IBM MQ ileti alışverişi sağlayıcısı normal kip

Kısıtlamalarla normal kip JMS 2.0 API 'sini kullanır, ancak yeni özellikleri (paylaşılan abonelikler, gecikmeli teslim ve zamanuyumsuz gönderme) kullanmaz.

#### IBM MQ ileti alışverişi sağlayıcısı geçiş kipi

Geçiş kipiyle bir IBM MQ 8.0 ya da daha sonraki bir kuyruk yöneticisine bağlanabilirsiniz, ancak önden okuma ve akış gibi IBM WebSphere MQ 7.0 ya da daha sonraki bir kuyruk yöneticisinin özellikleri kullanılmaz.



Şekil 89. İleti alışverişi sağlayıcısı kipleri

## Yordam

Belirli bir bağlantı üreticisi için **PROVIDERVERSION** özelliğini yapılandırmak üzere:

- IBM MQ Explorer kullanarak **PROVIDERVERSION** özelliğini yapılandırmak için [Kuyruk yöneticilerini ve nesnelerini yapılandırmabaşlıklı konuya](#) bakın.
- **PROVIDERVERSION** özelliğini JMS yönetim aracını kullanarak yapılandırmak için [Kuyruk yöneticilerini ve nesnelerini yapılandırmabaşlıklı konuya](#) bakın.
- IBM JMS uzantılarını ya da IBM MQ JMS uzantılarını kullanarak bir JMS uygulamasında **PROVIDERVERSION** özelliğini yapılandırmak için bkz. [IBM MQ classes for JMS uygulamasında bağlantı üreticilerini ve hedefleri oluşturma ve yapılandırma](#).



JVM ' deki tüm bağlantı üreticileri için bağlantı üreticisi sağlayıcı kipi ayarlarını geçersiz kılmak için:

- Bağlantı üreticisi sağlayıcı kipi ayarlarını geçersiz kılmak için `com.ibm.msg.client.wmq.overrideProviderVersion` özelliğini kullanın  
Kullanmakta olduğunuz bağlantı üreticisini değiştiremezseniz, bağlantı üreticisindeki herhangi bir ayarı geçersiz kılmak için `com.ibm.msg.client.wmq.overrideProviderVersion` özelliğini kullanabilirsiniz. Bu geçersiz kılma, JVM ' deki tüm bağlantı üreticileri için geçerlidir, ancak gerçek bağlantı üreticisi nesnelere değiştirilmez.

### İlgili kavramlar

[JMS sağlayıcı sürümü sorun giderme](#)

### İlgili başvurular

[PROVIDERVERSION](#)

[Bağlantı üreticisi özellikleri](#)

[IBM MQ classes for JMS nesnelerinin özellikleri arasındaki bağımlılıklar](#)

## IBM MQ ileti alışverişi sağlayıcısı işletim kipleri

Bağlantı üreticisine ilişkin PROVIDERVERSION özelliğini uygun değere ayarlayarak, bir JMS uygulamasının yayınlamak ve abone olmak için hangi IBM MQ ileti alışverişi sağlayıcısı kipini kullanacağını seçebilirsiniz. Bazı durumlarda, PROVIDERVERSION özelliği belirtilmemiş olarak ayarlanır; bu durumda, JMS istemcisi hangi işlem kipinin kullanılacağını belirlemek için bir algoritma kullanır.

### PROVIDERVERSION özellik değerleri

Bağlantı üreticisi **PROVIDERVERSION** özelliğini aşağıdaki değerlerden herhangi birine ayarlayabilirsiniz:

#### 8 -normal kip

JMS uygulaması normal kipi kullanır. Bu kip, JMS' i gerçekleştirmek için bir IBM MQ kuyruk yöneticisinin tüm özelliklerini kullanır.

#### 7 -kısıtlamalar içeren olağan kip

JMS uygulaması, kısıtlamalarla normal kipi kullanır. Bu kip JMS 2.0 API 'sini kullanır, ancak paylaşılan abonelikler, gecikmeli teslim ya da zamanuyumsuz gönderme gibi yeni özellikleri kullanmaz.

#### 6 -geçiş kipi

JMS uygulaması geçiş kipini kullanır. Geçiş kipinde IBM MQ classes for JMS , IBM WebSphere MQ 6.0 ile birlikte verilen özelliklere ve algoritmalara benzer özellikleri ve algoritmaları kullanır.

#### belirlenmedi (varsayılan değer)

JMS istemcisi, hangi işlem kipinin kullanıldığını saptamak için bir algoritma kullanır.

**PROVIDERVERSION** özelliği için belirttiğiniz değer bir dizgi olmalıdır. 8, 7 ya da 6 seçeneklerinden birini belirtiyorsanız, bunu aşağıdaki biçimlerden herhangi birinde yapabilirsiniz:

- V.R.M.F
- V.R.M
- V.R
- V

Burada V, R, M ve F sıfırdan büyük ya da sıfıra eşit tamsayı değerleridir. Ek R, M ve F değerleri isteğe bağlıdır ve ayrıntılı denetim gerektiğinde kullanmanız için kullanılabilir. Örneğin, **PROVIDERVERSION** 7 düzeyini kullanmak istiyorsanız, **PROVIDERVERSION** = 7, 7.0, 7.0.0 ya da 7.0.0.0 değerini ayarlayabilirsiniz.

### Bağlantı üreticisi nesnesi tipleri

Aşağıdaki bağlantı üreticisi nesnesi tipleri için **PROVIDERVERSION** özelliğini ayarlayabilirsiniz:

- MQConnectionFactory
- MQQueueConnectionÜreticisi



- MQTopicConnectionÜreticisi
- MQXAConnectionFactory
- MQXAQueueConnectionÜreticisi
- MQXAQueueConnectionÜreticisi
- MQXAQueueConnectionÜreticisi
- MQXATopicConnectionÜreticisi

Bu farklı bağlantı üreticisi tipleri hakkında daha fazla bilgi için bkz. [“Yönetim araçlarını kullanarak JMS ve Jakarta Messaging nesnelerini yapılandırma”](#) sayfa 669.

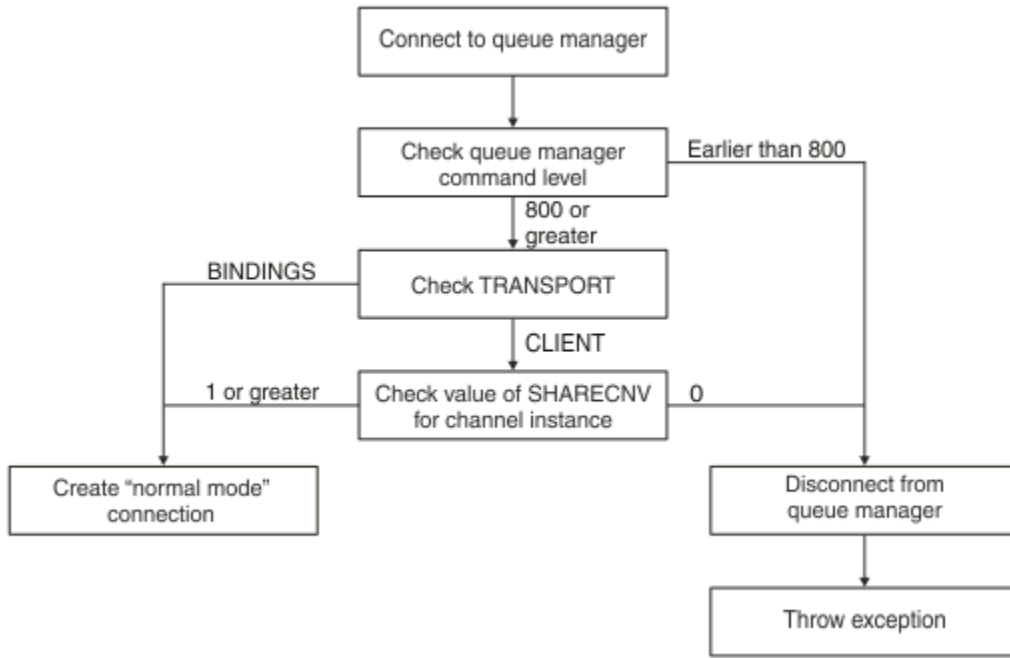
### İlgili kavramlar

IBM MQ ileti alışverişi sağlayıcısı

### **PROVIDERVERSION Normal kip**

Normal kip, JMS' i gerçekleştirmek için IBM MQ kuyruk yöneticisinin tüm özelliklerini kullanır. Bu kip, JMS 2.0 API ve işlevselliğini kullanmak için eniyilenmiştir.

Aşağıdaki akış grafiği, JMS istemcisinin normal kip bağlantısının yaratılıp yaratılmayacağını belirlemek için yaptığı denetimleri gösterir.



Şekil 90. PROVIDERVERSION normal kipi

Bağlantı üreticisi ayarlarında belirtilen kuyruk yöneticisi 800 ya da daha yüksek bir komut düzeyine sahipse ve bağlantı üreticisinin **TRANSPORT** özelliği BINDINGS olarak ayarlandıysa, başka özellikler denetlenmeden olağan kip bağlantısı yaratılır.

Bağlantı üreticisi ayarlarında belirtilen kuyruk yöneticisi 800 ya da daha yüksek bir komut düzeyine sahipse ve **TRANSPORT** özelliği CLIENT olarak ayarlandıysa, sunucu bağlantısı kanalındaki **SHARECNV** özelliği de denetlenir. IBM MQ ileti alışverişi sağlayıcısı normal kipi paylaşım etkileşimleri özelliğini kullandığından bu denetim gereklidir. Bu nedenle, olağan kipte bağlantı girişiminin başarılı olması için, paylaşılabilen etkileşimlerin sayısını denetleyen **SHARECNV** özelliğinin 1 ya da daha büyük bir değeri olmalıdır.

Akış grafiğinde gösterilen tüm denetimler başarılı olursa, kuyruk yöneticisine normal kipte bir bağlantı oluşturulur ve tüm JMS 2.0 API ve özellikleri (zamanuyumsuz gönderme, gecikmeli teslim ve paylaşılan abonelik) kullanılabilir.

Aşağıdaki nedenlerden birinden ötürü olağan kip bağlantısı yaratma girişimi başarısız oldu:

- Bağlantı üreticisi ayarlarında belirtilen kuyruk yöneticisi, 800 'den önceki bir komut düzeyine sahip. Bu durumda, `createConnection` yöntemi bir kural dışı durumla başarısız olur `JMSFMQ0003`.
- Sunucu bağlantı kanalındaki **SHARECNV** özelliği 0 olarak ayarlanır. Bu özelliğin değeri 1 ya da daha büyük değilse, `createConnection` yöntemi bir kural dışı durumla başarısız olur `JMSCC5007`.

### İlgili başvurular

IBM MQ classes for JMS nesnelерinin özellikleri arasındaki bağımlılıklar

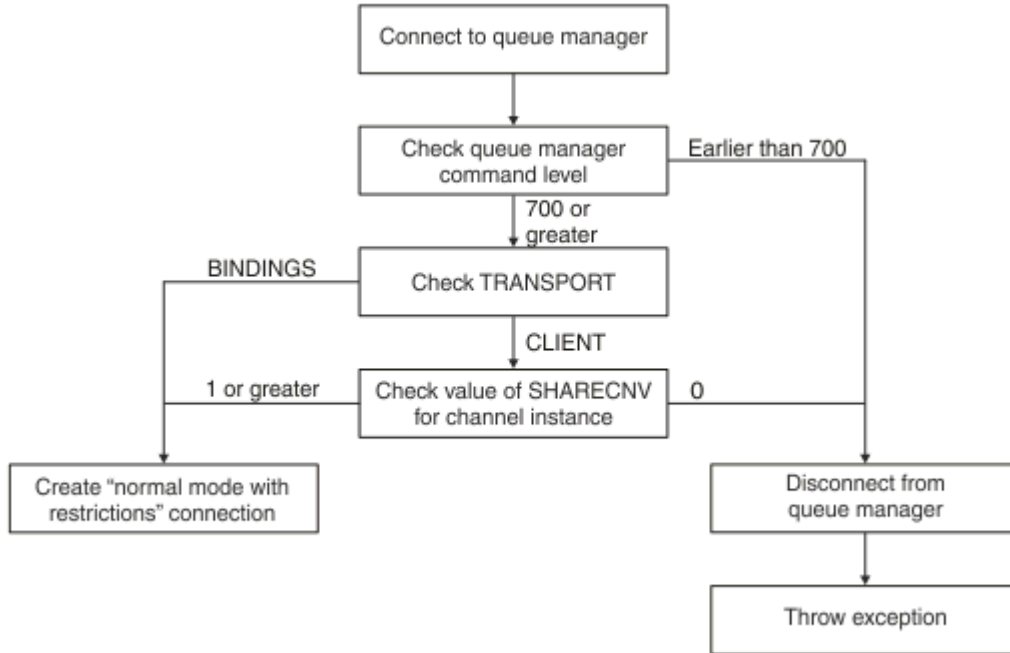
DEFINE CHANNEL (SHARECNV özelliği)

Aktarım

### **PROVIDERVERSION kısıtlamalar içeren normal kip**

Kısıtlamalarla normal kip JMS 2.0 API 'sini kullanır, ancak paylaşılan abonelikler, gecikmiş teslim ya da zamanuyumsuz gönderme gibi yeni IBM MQ 8.0 ya da sonraki özellikleri kullanmaz.

Aşağıdaki akış grafiği, kısıtlamalar bağlantısı olan normal bir kipin yaratılıp yaratılamayacağını belirlemek için JMS istemcisinin yaptığı denetimleri gösterir.



Şekil 91. PROVIDERVERSION kısıtlamalar içeren normal kip

Bağlantı üreticisi ayarlarında belirtilen kuyruk yöneticisi 700 ya da daha yüksek bir komut düzeyine sahipse ve bağlantı üreticisinin **TRANSPORT** özelliği BINDINGS olarak ayarlıysa, başka özellikler denetlenmeden olağan kipte bir bağlantı yaratılır.

Bağlantı üreticisi ayarlarında belirtilen kuyruk yöneticisi 700 ya da daha yüksek bir komut düzeyine sahipse ve **TRANSPORT** özelliği CLIENT olarak ayarlandıysa, sunucu bağlantısı kanalındaki **SHARECNV** özelliği de denetlenir. Kısıtlamalar içeren IBM MQ ileti alışverişi sağlayıcısı normal kipi paylaşım etkileşimleri özelliğini kullandığından bu denetim gereklidir. Bu nedenle, bağlantı kısıtlamaları olan olağan bir kipin başarılı olması için, paylaşılabilen etkileşimlerin sayısını denetleyen **SHARECNV** özelliğinin 1 ya da daha büyük bir değeri olmalıdır.

Akış grafiğinde gösterilen tüm denetimler başarılı olursa, kuyruk yöneticisine kısıtlamalar içeren normal bir kip yaratılır ve JMS 2.0 API 'sini kullanabilirsiniz, ancak zamanuyumsuz gönderme, gecikmeli teslim ya da paylaşılan abonelik özelliklerini kullanamazsınız.

Aşağıdaki nedenlerden birinden ötürü, kısıtlama bağlantısıyla olağan kip yaratma girişimi başarısız oldu:

- Bağlantı üreticisi ayarlarında belirtilen kuyruk yöneticisinin komut düzeyi 700 'den önce. Bu durumda, `createConnection` yöntemi JMSFCC5008kural dışı durumuyla başarısız olur.
- Sunucu bağlantı kanalındaki **SHARECNV** özelliği 0olarak ayarlanır. Bu özelliğin değeri 1 ya da daha büyük değilse, `createConnection` yöntemi bir kural dışı durumla başarısız olur JM5CC5007.

### İlgili başvurular

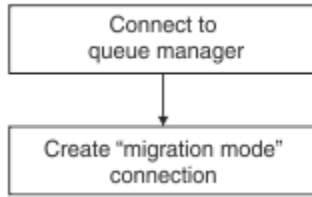
[IBM MQ classes for JMS nesnelerinin özellikleri arasındaki bağımlılıklar](#)

[DEFINE CHANNEL \(SHARECNV özelliği\)](#)

[Aktarım](#)

### **PROVIDERVERSION** geçiş kipi

Geçiş kipi için IBM MQ classes for JMS , kuyruğa yollanmış yayınlama/abone olma, istemci tarafında uygulanan seçim, çoklu yol olmayan kanallar ve dinleyicileri uygulamak için kullanılan yoklama gibi IBM WebSphere MQ 6.0ile sağlanan özelliklere ve algoritmalara benzer özellikleri ve algoritmaları kullanır.



Şekil 92. **PROVIDERVERSION** geçiş kipi

WebSphere Message Broker 6.0 ya da WebSphere Message Broker 6.1 ' a IBM MQ Enterprise Transport 6.0sürümünü kullanarak bağlanmak istiyorsanız, geçiş kipini kullanmanız gerekir.

Geçiş kipini kullanarak bir IBM MQ 8.0 kuyruk yöneticisine bağlanabilirsiniz, ancak IBM MQ classes for JMS kuyruk yöneticisinin yeni özelliklerinin hiçbiri kullanılmaz; örneğin, önden okuma ya da akış. Dağıtık bir altyapıda ( **z/OS** ya da IBM MQ for z/OS 8.0 ya da daha sonraki bir kuyruk yöneticisinde ) IBM MQ 8.0 ya da sonraki bir kuyruk yöneticisine bağlanan bir IBM MQ 8.0 ya da sonraki bir istemciniz varsa, ileti seçimi istemci sisteminde değil, kuyruk yöneticisi tarafından gerçekleştirilir.

IBM MQ ileti alışverişi sağlayıcısı geçiş kipi belirtilirse ve IBM MQ classes for JMS , JMS 2.0 API 'sinden herhangi birini kullanmayı denerse, API yöntemi çağırısı JM5CC5007kural dışı durumuyla başarısız olur.

### İlgili başvurular

[IBM MQ classes for JMS nesnelerinin özellikleri arasındaki bağımlılıklar](#)

[Aktarım](#)

### **PROVIDERVERSION** belirlenmedi

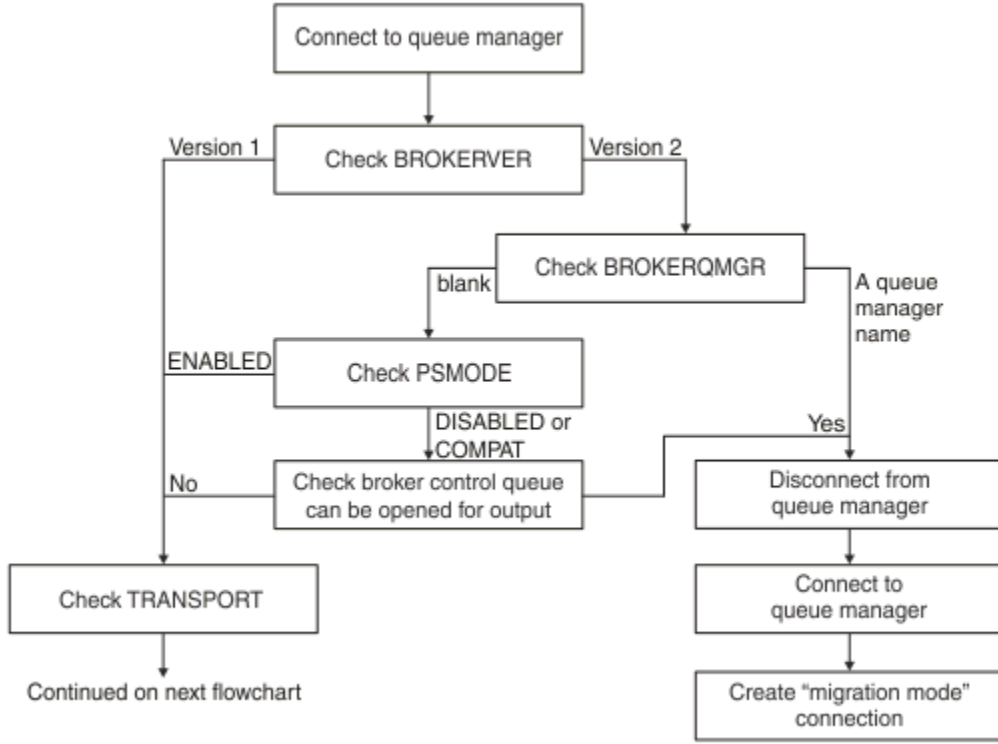
Bir bağlantı üreticisinin **PROVIDERVERSION** özelliği belirtilmediğinde, JMS istemcisi kuyruk yöneticisine bağlanmak için hangi işlem kipinin kullanıldığını saptamak için bir algoritma kullanır. JNDI ad alanında önceki bir IBM MQ classes for JMS sürümüyle yaratılan bir bağlantı üreticisi, bağlantı üreticisi yeni IBM MQ classes for JMSsürümüyle kullanıldığında belirlenmemiş değeri alır.

**PROVIDERVERSION** özelliği belirtilmezse, `createConnection` yöntemi çağrıldığında algoritma kullanılır. Algoritma, IBM MQ ileti alışverişi sağlayıcısı normal kipinin mi, kısıtlamalar içeren olağan kipinin mi, yoksa IBM MQ ileti alışverişi sağlayıcısı geçiş kipinin mi gerekli olduğunu saptamak için bağlantı üreticisi özelliklerinin sayısını denetler. Olağan kip her zaman önce denir, daha sonra sınırlamalarla olağan kip. Bu tip bağlantı kurulamazsa, JMS istemcisi kuyruk yöneticisiyle bağlantısını keser ve geçiş kipi bağlantısını kurmak için kuyruk yöneticisiyle yeniden bağlantı kurar.

### **BROKERVER, BROKERQMGR, PSMODEve BROKERCONQ** özelliklerinin denetlenmesi

Özellik değerlerinin denetlenmesi, Şekil 1çinde gösterildiği gibi **BROKERVER** özelliğiyle başlar.

**BROKERVER** özelliği V1olarak ayarlanırsa, Şekil 2' de gösterildiği gibi, **TRANSPORT** özelliği daha sonra denetlenir. Ancak **BROKERVER** özelliği V2olarak ayarlanırsa, Şekil 1 ' de gösterilen ek denetim **TRANSPORT** özelliği denetlenmeden önce yapılır.



Şekil 93. PROVIDERVERSION belirtilmedi

**BROKERVER** özelliği V2olarak ayarlanırsa, normal kip bağlantısının kurulabilmesi için **BROKERQMGR** özelliğinin blank olması gerekir. Buna ek olarak, kuyruk yöneticisindeki **PSMODE** özneliği ENABLED olarak ayarlanmalı ya da **BROKERCONQ** özelliği tarafından belirtilen aracı denetim kuyruğu çıkışı için açılmamalıdır.

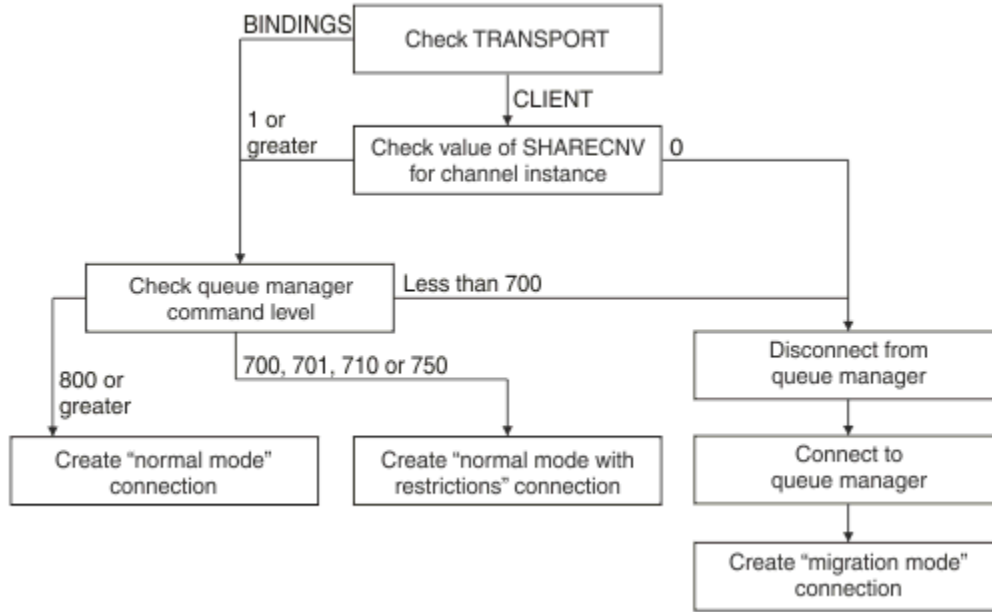
Özellik değerleri normal kip bağlantısı için gerekli olarak ayarlanırsa, Şekil 2' de gösterildiği gibi **TRANSPORT** özelliğine gider.

Özellik değerleri normal kip bağlantısı için gerekli olarak ayarlanmazsa, JMS istemcisi kuyruk yöneticisiyle bağlantısını keser ve yeniden bağlantı kurar ve bir geçiş kipi bağlantısı yaratır. Bu durum aşağıdaki durumlarda oluşur:

- **BROKERQMGR** özelliği blank ise ve kuyruk yöneticisindeki **PSMODE** özneliği COMPAT ya da DISABLED olarak ayarlandıysa ve **BROKERCONQ** özelliği tarafından belirtilen aracı denetim kuyruğu çıkışı için açılabilir (yani, çıkış başarılı olduğunda MQOPEN).
- **BROKERQMGR** özelliği bir kuyruk adı belirtiyorsa.

## TRANSPORT özelliğinin ve komut düzeyinin denetlenmesi

Şekil 2 , kuyruk yöneticisinin **TRANSPORT** özelliği ve komut düzeyi için yapılan denetimleri gösterir.



Şekil 94. PROVIDERVERSION belirtilmedi (devam)

Aşağıdaki durumlardan birinde olağan kip bağlantısı yaratılır:

- Bağlantı üreticisinin **TRANSPORT** özelliği BINDINGS olarak ayarlandı ve kuyruk yöneticisinin komut düzeyi 800 ya da daha yüksek.
- **TRANSPORT** özelliği CLIENT olarak ayarlanır, sunucu bağlantısı kanalındaki **SHARECNV** özelliği 1 ya da daha büyük bir değere sahiptir ve kuyruk yöneticisi 800 ya da daha yüksek bir komut düzeyine sahiptir.

Kuyruk yöneticisinin komut düzeyi 750 ise, kuyruk yöneticisiyle bağlantı kısıtlaması olan olağan bir kip yaratılır.

**TRANSPORT** özelliği CLIENT olarak ayarlıysa ve sunucu bağlantı kanalındaki **SHARECNV** özelliği 0 değerine sahipse, geçiş kipi bağlantısı da yaratılır.

### İlgili başvurular

[IBM MQ classes for JMS nesnelere özellikleri arasındaki bağımlılıklar](#)

[ALTER QMGR \(PSMODE özneliği\)](#)

[BROKERCONQ](#)

[BROKERQMGR](#)

[ARALIKÇI](#)

[DEFINE CHANNEL \(SHARECNV özelliği\)](#)

[Aktarım](#)

## WebSphere Application Server içinde sağlayıcı sürümü bilgilerini yapılandırma

WebSphere Application Server içinde sağlayıcı sürümü bilgilerini yapılandırmak için yönetim konsolunu ya da wsadmin komutlarını kullanabilirsiniz.

### Yordam

WebSphere Application Server içinde bir IBM MQ bağlantı üreticisi ya da etkinleştirme belirtimi nesnesine ilişkin sağlayıcı sürümü bilgilerini yapılandırmak için, WebSphere Application Server ürün belgelerinde daha fazla bilgi için *İlgili bilgiler* konusuna bakın.

### **WebSphere Application Server 8.5.5 ile ilgili bilgiler**

[IBM MQ ileti alışverişi sağlayıcısı bağlantı üreticisi ayarları](#)

**createWMQConnectionFactory** DELETE ...

[IBM MQ ileti alışverişi sağlayıcısı etkinleştirme belirtimi ayarları](#)

**createWMQActivationSpec** DELETE ...

### **WebSphere Application Server 8.0.0 ile ilgili bilgiler**

[IBM MQ ileti alışverişi sağlayıcısı bağlantı üreticisi ayarları](#)

**createWMQConnectionFactory** DELETE ...

[IBM MQ etkinleştirme belirtimi ayarları](#)

**createWMQActivationSpec** DELETE ...

### **WebSphere Application Server 7.0.0 ile ilgili bilgiler**

[IBM MQ ileti alışverişi sağlayıcısı bağlantı üreticisi ayarları](#)

**createWMQConnectionFactory** DELETE ...

[IBM MQ etkinleştirme belirtimi ayarları](#)

**createWMQActivationSpec** DELETE ...

## **Sürekli aboneliklerin WebSphere Application Server kaldırılması**

IBM MQ ileti alışverişi sağlayıcısını WebSphere Application Server 7.0 ve WebSphere Application Server 8.0 ile kullandığınızda, etkinleştirme belirtimlerine bağlı ileti odaklı Bean uygulamaları tarafından oluşturulan sürekli abonelikler kaldırılmaz. Sürekli abonelikler, IBM MQ Explorer ya da IBM MQ komut satırı yardımcı programı kullanılarak kaldırılabilir.

### **Bu görev hakkında**

Uygulamanın IBM MQ olanağına bağlanmak için IBM MQ ileti alışverişi sağlayıcısı normal kipini kullanan bir WebSphere Application Server 7.0 ya da WebSphere Application Server 8.0 yönetim ortamı içinde çalışması koşuluyla, sürekli aboneliği kaldıran bir Bean uygulaması, dinleyici kapısını ya da etkinleştirme belirtimini kullanacak şekilde yapılandırılabilir.

İletiyi yönlendirilen Bean uygulaması bir dinleyici kapısına bağlıysa, IBM MQ ileti alışverişi sağlayıcısı uygulama ilk başlatıldığında uygulamaya ilişkin kalıcı aboneliği yaratır. İletiyi yönlendirilen Bean uygulaması bir uygulama sunucusundan kaldırıldığında ve uygulama sunucusu yeniden başlatıldığında sürekli abonelik kaldırılır.

Bir etkinleştirme belirtimine bağlı, iletiyle yönlendirilen bir bean uygulaması biraz farklı bir şekilde çalışır. Uygulama ilk kez başlatıldığında uygulama için sürekli abonelik yaratılır. Ancak, uygulama kaldırıldığında ve uygulama sunucusu yeniden başlatıldığında sürekli abonelik kaldırılmaz.

Bu, artık bir WebSphere Application Server sisteminde kurulu olmayan uygulamalar için IBM MQ Yayınla/ Abone Ol altyapısında bir dizi sürekli aboneliğe neden olabilir. Bu abonelikler "artık abonelikler" olarak bilinir ve Yayınlama/Abone Olma altyapısı çalışırken kuyruk yöneticisinde sorunlara yol açabilir.

Bir konuda bir ileti yayınlandığında, IBM MQ Yayınla/Abone ol altyapısı, bu konuda kayıtlı olan her sürekli abonelik için iletinin bir kopyasını oluşturur ve bunu bir iç kuyruğa yerleştirir. Bu sürekli aboneliği kullanan uygulamalar, iletiyi bu iç kuyruktan alır ve kullanır.

Bu sürekli aboneliği kullanan ileti odaklı bean uygulaması artık kurulu değilse, uygulamaya ilişkin yayınlanan iletilerin kopyaları oluşturulmaya devam eder. Ancak, bu iletiler hiçbir zaman işlenmez; bu, iç kuyrukta hiçbir zaman kaldırılamayacak çok sayıda ileti kalabileceğini gösterir.

### **Başlamadan önce**

IBM MQ Yayınlama/Abone Olma motoruna kayıtlı aboneliklerle ilişkilendirilmiş bir Abonelik Adı olacaktır.

Etkinleştirme belirtilerine bağlı ileti odaklı Bean 'ler için WebSphere Application Server IBM MQ ileti sistemi sağlayıcısı tarafından sürekli abonelikler, aşağıdaki biçimde bir Abonelik Adı içerir:

```
JMS:queue manager name:client identifier:subscription name
```

Burada:

### **Kuyruk yöneticisi adı**

Bu, Yayınlama/Abone Olma altyapısının çalıştığı IBM MQ kuyruk yöneticisinin adıdır.

### **İstemci Tanıtıcısı**

Bu, iletiyle yönlendirilen bean 'in bağlandığı etkinleştirme belirtiminin İstemci Tanıtıcısı özelliğinin değeridir.

### **Abonelik adı**

Bu, iletiyle yönlendirilen Bean uygulamasının kullanmak üzere yapılandırıldığı etkinleştirme belirtimine ilişkin etkinleştirme belirtimi özelliği Abonelik adının değeridir.

Örneğin, testQMkuyruk yöneticisine bağlanmak için ayarlanmış bir etkinleştirme belirtimine sahip olduğumuzu varsayalım. Etkinleştirme belirtiminde şu özellikler ayarlı:

- İstemci tanıtıcısı = testClienttanıtıcısı
- Abonelik adı = durableSubscription1

Sürekli aboneliği alan bir ileti odaklı bean bu etkinleştirme belirtimine bağlıysa, şu Abonelik adına sahip kuyruk yöneticisindeki IBM MQ yayınlama/abone olma altyapısında testQM bir abonelik yaratılır:

- JMS:testQM:testClientID:durableSubscription1

Belirli bir kuyruk yöneticisi için IBM MQ yayınlama/abone olma motoruna kaydedilen abonelikler aşağıdaki iki yoldan biriyle görüntülenebilir:

- İlk seçenek MQ Explorer 'ı kullanmaktır. MQ Explorer yayınlama/abone olma işi için kullanılan bir kuyruk yöneticisine bağlandığında, yayınlama/abone olma motoruna kayıtlı olan abonelerin listesi, gezinme bölmesinde IBM WebSphere MQ ->queue manager name-> Subscriptions girdisi tıklanarak görüntülenebilir.
- Bir yayınlama/abone olma motoruna kayıtlı abonelikleri görüntülemenin diğer yolu, IBM MQ komut satırı yardımcı programını **runmqsc** kullanmak ve **display sub**komutunu çalıştırmaktır. Bunu yapmak için bir komut istemi için, *WebSphere MQ* \bin dizinine geçin ve **runmqsc**komutunu başlatmak için aşağıdaki komutu girin:

```
- runmqsc queue manager name
```

**runmqsc** yardımcı programı başlatıldığında, **runmqsc** ' in bağlı olduğu kuyruk yöneticisinde çalışan yayınlama/abone olma işlemcisiyle kayıtlı olan tüm sürekli abonelikleri listelemek için aşağıdaki komutu girin:

```
- display sub(*) durable
```

Yayınlama/abone olma işlemcileri ile kayıtlı sürekli aboneliklerin etkin olup olmadığını denetlemek için:

1. Yayınlama/Abone Olma motoruna kaydedilen sürekli aboneliklerin listesini oluşturun.

2. Her sürekli abonelik için:

- Sürekli abone için abonelik adına bakın ve *istemci tanıtıcısı* ve *abonelik adı* değerini not edin.
- Bu Yayınlama/Abone Olma motoruna bağlanan WebSphere Application Server sistemlerine bakın. *İstemci tanıtıcısı* değeriyle ve *abonelik adı*yla eşleşen abonelik adı özelliğiyle eşleşen İstemci Tanıtıcısı özelliğine sahip tanımlanmış etkinleştirme belirtileri olup olmadığını görün.
- IBM MQ abonelik adında *istemci tanıtıcısı* ve *abonelik adı* alanlarıyla eşleşen İstemci Tanıtıcısı ve abonelik adı özelliklerine sahip etkinleştirme belirtileri bulunamazsa, bu sürekli aboneliği kullanan etkinleştirme belirtileri yoktur. Sürekli abonelik silinebilir.

- Sürekli abonelik adıyla eşleşen tanımlanmış bir etkinleştirme belirtimi varsa, yapılması gereken son denetim, bu etkinleştirme belirtimini kullanan bir ileti odaklı bean uygulaması olup olmadığını görmektir. Bunu yapmak için:
  - Aradığınız sürekli aboneliği çıkarmış olan etkinleştirme belirtimine ilişkin JNDI adını not edin.
  - Kurulan her ileti odaklı bean uygulaması için WebSphere Application Server yönetim konsolunda Yapılandırma bölmesini açın.
  - Yapılandırma bölmesinde İleti Odaklı Bean dinleyici bağ tanımları bağlantısını tıklatın.
  - İletiyi yönlendirilen Bean uygulamasıyla ilgili bilgileri içeren bir tablo görüntülenir. Bağlayıcılar kolonunda etkinleştirme belirtimi radyo düğmesi seçildiyse ve Hedef Kaynak JNDI ad alanı, sürekli aboneliği çıkarmış olan etkinleştirme belirtimine ilişkin JNDI adını içeriyorsa, abonelik kullanılmaya devam eder ve silinemez.
  - Etkinleştirme belirtimini kullanan ileti odaklı Bean uygulamaları bulunamazsa, kalıcı abonelik silinebilir.

## Yordam

"Artık" sürekli abonelik tanımlandıktan sonra, IBM MQ Explorer ya da IBM MQ komut satırı yardımcı programı **runmqsc** kullanılarak silinebilir.

IBM MQ Explorer kullanarak "artık" sürekli aboneliği silmek için:

1. Abonelik için girdiyi vurgulayın
2. Girdiyi sağ tıklatın ve **Sil ...** seçeneğini belirleyin. menüden. Bir onay penceresi görüntülenir.
3. Doğrulama penceresinde görüntülenen abonelik adının doğru olup olmadığını denetleyin ve **Evet** ' i tıklatın.

IBM MQ Explorer , Yayınlama/Abone Olma motorundan aboneliği siler ve bununla ilişkili tüm iç kaynakları temizler (sürekli aboneliğin kaydedildiği konu için yayınlanan işlenmemiş iletiler gibi).

IBM MQ komut satırı yardımcı programını **runmqsc** kullanarak bir "artık" sürekli aboneliği silmek için **delete sub** komutu çalıştırılmalıdır:

1. Komut istemi oturumu aç
2. *IBM MQ* \bin dizinine gidin
3. **runmqsc** ' i başlatmak için aşağıdaki komutu girin:

```
runmqsc queue manager name
```

4. **runmqsc** yardımcı programı başlatıldığında şunu girin:

```
delete sub(Subscription name)
```

Burada *Abonelik adı* , sürekli aboneliğin abonelik adıdır ve şu formu alır:

- *JMS:queue manager name:client identifier:subscription name*

## yapılandırmaManaged File Transfer

Kuruluşta sonra Managed File Transfer özelliklerini yapılandırabilirsiniz.

Managed File Transfer yapılandırmanızın esnekliğini artırmak için IBM MQ yüksek düzeyde kullanılabilirlik çözümlerinden yararlanabilirsiniz. Aracılarınız eşlenmiş veri kuyruğu yöneticilerini (RDIMM ' ler) kullanıyorsa, bunları kayan IP adresi özelliğini kullanacak şekilde yapılandırmanız gerekir. Bu, araçların şu anda çalışmakta olan üç RDQM eşgörünümünden hangisiyle iletişim kurmak için aynı IP adresini kullandığı ve hata durumunda yedek sisteme geçiş sırasında otomatik olarak yeniden bağlandığı anlamına gelir (bkz. RDQM yüksek kullanılabilirlik ve Kayan IP adresi oluşturma ve silme). Çok eşgörünümlü kuyruk yöneticisi çözümünü kullanırsanız, uygulamalar her bir eşgörünümle iletişim kurmak için farklı bir IP adresi kullanır;



bu adres, hata durumunda yedek sisteme geçiş sırasında istemci yeniden bağlanması ile işlenir (bkz. [Çok eşgörünümlü kuyruk yöneticileri ve Kanal ve istemci yeniden bağlantısı](#)).

### İlgili kavramlar

[Managed File Transfer kullanımı için ipuçları](#)

### İlgili görevler

[MFT kaynaklarının izlenmesi](#)

[MFT ürününü kullanıcı çıkışlarıyla özelleştirme](#)

[MQMFTCredentials.xml ' in yapılandırılması](#)

[güvenlikManaged File Transfer](#)

[MFT ile çalıştırılacak programları belirtme](#)

[Sorun gidermeManaged File Transfer](#)

[YönetmeManaged File Transfer](#)

### İlgili başvurular

[MFT komutlar](#)

[MFTagent.properties dosyası](#)

[MFT kurtarma ve yeniden başlatma](#)

## Çoklu platformlarda MFT yapılandırma seçenekleri

Managed File Transfer içinde, kurulumunuzla ilgili temel bilgileri içeren ve işlem için gerekli olan bir özellikler dosyaları kümesi sağlanır. Bu özellikler dosyaları, ürünü kurarken tanımladığınız yapılandırma dizininde bulunur.

Birden çok yapılandırma seçeneği kümeniz olabilir, her yapılandırma seçeneği kümesi bir dizi dizin ve özellik dosyası içerir. Bu özellikler dosyalarında tanımlanan değerler, komut satırında belirttik olarak farklı bir değer belirtmediğiniz sürece, tüm Managed File Transfer komutları için varsayılan parametre olarak kullanılır.

Kullanmakta olduğunuz varsayılan yapılandırma seçenekleri kümesini değiştirmek için **fteChangeDefaultConfigurationOptions** komutunu kullanabilirsiniz. Tek bir komut için kullandığınız yapılandırma seçenekleri kümesini değiştirmek üzere herhangi bir Managed File Transfer komutuyla **-p** parametresini kullanabilirsiniz.

Bir yapılandırma seçeneği kümesinin adı, eşgüdüm kuyruğu yöneticisinin adıdır ve bu seçeneğin değiştirilmemesi önerilir. Ancak, bir yapılandırma seçeneği kümesinin adını değiştirebilirsiniz, ancak **config** ve **logs** dizinlerinin adını değiştirmeniz gerekir. Aşağıdaki örneklerde, yapılandırma seçenekleri kümesinin adı **coordination\_qmgr\_name** olarak gösterilir.

### Yapılandırma seçenekleri dizin yapısı

Ürünü yapılandırdığınızda, yapılandırma dizininde aşağıdaki yapıda dizinler ve özellikler dosyaları oluşturulur. Bu dizinleri ve özellikler dosyalarını şu komutlarla da değiştirebilirsiniz: **fteSetupCoordination**, **fteSetupCommands**, **fteChangeDefaultConfiguration** ve **fteCreateAgent**.

```
MQ_DATA_PATH/mqft/  
  config/  
    coordination_qmgr_name/  
      coordination.properties  
      command.properties  
      agents/  
        agent_name/  
          agent.properties  
          exits  
      loggers/  
        logger_name  
          logger.properties  
  installations/  
    installation_name/  
      installation.properties
```

`coordination_qmgr_name` dizini, bir yapılandırma seçenekleri dizinidir. Yapılandırma dizininde birden çok yapılandırma seçeneği dizini olabilir. `agent_name` dizini bir aracı dizinidir. Bu dizin, `agent.properties` dosyasını içermenin yanı sıra, kullanıcı çıkışı yordamları ve **`fteCreateBridgeAgent`** ve **`fteCreateCDAgent`** komutları tarafından oluşturulan çeşitli XML dosyaları için varsayılan konum olan `exits` dizinini içerir. Bir yapılandırma seçenekleri kümesinin `agents` dizininde birden çok aracı dizini olabilir.

## Özellikler dosyaları

### **installation.properties**

`installation.properties` dosyası, varsayılan yapılandırma seçenekleri kümesinin adını belirtir. Bu giriş, kullanılacak yapılandırmayı içeren yapılandırılmış bir dizin ve özellik dosyaları kümesini Managed File Transfer ' e işaret eder. Genellikle bir yapılandırma seçenekleri kümesinin adı, ilişkili eşgüdüm kuyruğu yöneticisinin adıdır. `installation.properties` dosyasıyla ilgili daha fazla bilgi için bkz. [MFT installation.properties dosyası](#).

### **coordination.properties**

`coordination.properties` dosyası, eşgüdüm kuyruk yöneticisine yönelik bağlantı ayrıntılarını belirtir. Birden çok Managed File Transfer kuruluşu aynı koordinasyon kuyruk yöneticisini paylaşabileceğinden, paylaşılan bir sürücüdeki ortak bir `coordination.properties` dosyasına sembolik bir bağlantı kullanabilirsiniz. `coordination.properties` dosyası hakkında daha fazla bilgi için bkz. [MFT coordination.properties dosyası](#).

### **command.properties**

MFT `command.properties` dosyası, komutları yayınladığınızda bağlanılacak komut kuyruğu yöneticisini ve Managed File Transfer ' un o kuyruk yöneticisiyle iletişim kurması için gereken bilgileri belirtir. `command.properties` dosyası hakkında daha fazla bilgi için bkz. [MFT command.properties dosyası](#).

### **agent.properties**

Her Managed File Transfer Agent , bir aracının kuyruk yöneticisine bağlanmak için kullandığı bilgileri içermesi gereken kendi özellikler dosyasına ( `agent.properties` ) sahiptir. `agent.properties` dosyası, aracının davranışını değiştiren özellikleri de içerebilir. `agent.properties` dosyası hakkında daha fazla bilgi için bkz. [MFT agent.properties dosyası](#).

### **logger.properties**

`logger.properties` dosyası, kaydedicilere ilişkin yapılandırma özelliklerini belirtir. `logger.properties` dosyasıyla ilgili daha fazla bilgi için bkz. [MFT kaydedici yapılandırma özellikleri](#).

## Özellikler dosyaları ve kod sayfaları

Javasınırlaması nedeniyle, tüm Managed File Transfer özellik dosyalarının içeriği İngilizce olarak kalmalıdır. ABD dışı bir İngilizce sistemdeki özellik dosyalarını düzenlerseniz, Unicode kaçış dizilerini kullanmanız gerekir.

### **İlgili başvurular**

[MFT için SSL/TLS özellikleri](#)

[MFT için Java sistem özellikleri](#)

[fteChangeDefaultConfigurationSeçenekleri](#)

[fteSetupKomutları: MFT command.properties dosyasını oluşturun](#)

[fteSetupKoordinasyonu](#)

[fteCreateAracı](#)

## **MFT yapılandırma seçenekleri z/OS**

z/OS üzerindeki Managed File Transfer yapılandırma seçenekleri, dağıtılmış platformlara ilişkin seçenekle aynıdır.

Çoklu platformlar üzerindeki yapılandırma seçenekleri hakkında daha fazla bilgi için bkz. "[Çoklu platformlarda MFT yapılandırma seçenekleri](#)" sayfa 701.

z/OS' da, yapılandırma konumu `BFG_DATA` ortam değişkeniyle tanımlanır. `BFG_DATA` tarafından başvuru z/OS UNIX System Services dizini altında bir yapılandırma yoksa, bir MFT komutu PDSE

kitaplığı veri kümesinin BFGCUSTM JCL komut dosyası, yapılandırmayı oluşturmak için gerekli işleri oluşturur. Bu oluşturulan işleri çalıştırdığınızda yapılandırma yaratılır. Yapılandırma oluşturma, erişilebilir olan var olan bir dizine başvuran BFG\_DATA ögesine dayanır.

Ayrıca, hem Multiplatforms hem de z/OS' de kullanılabilen aynı **fte** komutlarını kullanarak bir yapılandırma oluşturabilir ve koruyabilirsiniz. **fte** komutlarının bir listesi için bkz. [MFT komutları](#) .

### İlgili kavramlar

“Çoklu platformlarda MFT yapılandırma seçenekleri” sayfa 701

Managed File Transfer içinde, kurulumunuzla ilgili temel bilgileri içeren ve işlem için gerekli olan bir özellikler dosyaları kümesi sağlanır. Bu özellikler dosyaları, ürünü kurarken tanımladığınız yapılandırma dizininde bulunur.


“Bir Aracı Oluşturma” sayfa 719

Aracıya özgü PDSE 'yi (örneğin, *user.MFT.AGENT1*. Varsa, PDSE ' yi önceki bir aracı ya da kaydedici yapılandırmasından kopyalayın. Bu ilk yapılandırmanız ise, MFT ile birlikte sağlanan PDSE ' yi kopyalayın.

“Eşgüdüm kuyruğu yöneticisinin tanımlanması” sayfa 717

Managed File Transfer , eşgüdüm kuyruğu yöneticisi olarak işlev gören bir kuyruk yöneticisinin yaratılmasını gerektirir.

### İlgili görevler

 z/OS üzerinde MQMFTCredentials.xml ' in yapılandırılması

“z/OS üzerinde var olan bir MFT Agent ya da Logger komut veri kümesinin güncellenmesi” sayfa 720  
Managed File Transfer komut şablonu veri kümesinden oluşturulan bir Managed File Transfer komutu PDSE kitaplığı veri kümesini güncelleyebilirsiniz.

Linux

Windows

## Redistributable Managed File Transfer components



### ürünü karşıdan yükleme ve yapılandırma

Redistributable Managed File Transfer package , var olan bir IBM MQ altyapısına bağlanmak için yapılandırabileceğiniz Redistributable Managed File Transfer Agentolanağını sağlar ve kullanıcıların IBM MQürünü kurmalarına gerek kalmadan dosyaları aktarmalarını sağlar. IBM MQ 9.3.0' den yeniden dağıtılabilir paket, Redistributable Managed File Transfer Loggerürünü de içerir.

### Başlamadan önce

Redistributable Managed File Transfer Agent ve Redistributable Managed File Transfer Loggeriçin yeniden dağıtılabilir lisans koşulları hakkında bilgi için bkz. [IBM MQ Yeniden Dağıtılabilir Bileşenler](#).

Redistributable Managed File Transfer package bileşenleri, aşağıdaki kural dışı durumlarla birlikte Managed File Transfer işlevlerini sağlar:

- Redistributable Managed File Transfer Agentiçin, eşgüdüm, komut ve aracı kuyruk yöneticilerine bağ tanımlama kipi bağlantısı desteklenmez, istemci kipi bağlantısını kullanmanız gerekir. Komut yayınlarken, IBM MQ'un bir parçası olarak kurulan Managed File Transfer ' i kullanırken isteğe bağlı parametreleri sağlamanız gerekir: kuyruk yöneticisi anasistemi, kapısı, adı ve kanal adı.
-  Redistributable Managed File Transfer Logger yalnızca istemci kipinde yalnızca koordinasyon kuyruğu yöneticisine bağlanarak FILE tipi kaydedicileri destekler. Bir veritabanı kaydedici için eşgüdüm kuyruğu yöneticisine istemci kipi bağlantısı desteklenmiyor. Bağ tanımlama kipi bağlantısı gerekiyorsa, standart bir IBM MQkuruluşu kullanmanız gerekir.
-  IBM MQ 9.3.0' den **fteCreateCDAgent.cmd** komutu eklenmez. Kullanılabilir komutların tam listesi için bkz. [Kurulu MFT komut kümeleri](#).
- Managed File Transfer Connect:Direct desteklenmez.
- IBM MQ Explorer dahil değildir.

**Windows** Redistributable Managed File Transfer Agent'ününü kullanabilmek için sisteminize Microsoft'tarafından sağlanan Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 kitaplıklarını kurmanız gerekir. Bkz. [Desteklenen en son Visual C++ yüklemeleri](#).

**V 9.3.0** IBM MQ 9.3.0' den Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 kitaplıkları da Redistributable Managed File Transfer Loggeri için gereklidir.

**Not:** Advanced Message Security , Redistributable Managed File Transfer package ile desteklenmez.

## Bu görev hakkında

İsteğe bağlı olarak Redistributable Managed File Transfer package dosyasını karşıdan yükleyebilir ve kullanıcıların Managed File Transfer işlevselliğini elde etmek için IBM MQ ürününü kurmalarına gerek kalmadan yerel ortamlarıyla var olan IBM MQ altyapısı arasında dosya aktarmalarına izin vermek üzere var olan bir IBM MQ altyapısına bağlanmak için Redistributable Managed File Transfer Agent ' yi yapılandırabilirsiniz.

**V 9.3.0** IBM MQ 9.3.0' den Redistributable Managed File Transfer package , istemci kipinde eşgüdüm kuyruğu yöneticisine bağlanmak için bir dosya kaydedici ayarlamasını sağlayan Redistributable Managed File Transfer Loggerdosyasını da içerir.

## Yordam

1. [IBM MQ yeniden dağıtılabılır Managed File Transfer Agent paketini](#) Fix Centraladresinden yükleyin.

a) İşletim sisteminize ilişkin paketi seçin.

Arşiv ya da .zip dosyası adları, dosya içeriğini ve eşdeğer bakım düzeylerini açıklar. Dosya adları aşağıdaki biçimdedir:

- **Windows** V.R.M.F-IBM-MQFA-Redist-Win64
- **Linux** V.R.M.F-IBM-MQFA-Redist-LinuxX64
- **Linux** V.R.M.F-IBM-MQFA-Redist-LinuxS390X
- **Linux** V.R.M.F-IBM-MQFA-Redist-LinuxPPC64LE

Burada V.R.M.F , 9.2.0.0 ya da 9.2.1.0 gibi sürüm numarasıdır.

b) Paketi çıkarmak istediğiniz dizini tanımlayın, örneğin:

- **Windows** C:\MFTZ
- **Linux** /home/MFTZ

2. Karşıdan yüklenen paketin içeriğini çıkarın:

- **Windows** Windows üzerinde, çıkarmak için Windows Explorer araçlarını kullanın.
- **Linux** Linux üzerinde, aşağıdaki gibi ayıklayın ve çıkarın:

```
gunzip V.R.M.F-IBM-MQFA-Redist-LinuxX64.tar.gz
```

ve sonra

```
tar xvf V.R.M.F-IBM-MQFA-Redist-LinuxX64.tar
```

Burada V.R.M.F , 9.3.0.0 ya da 9.3.1.0 gibi sürüm numarasıdır.

Aşağıdaki dizinler oluşturulur:

- **Linux** **Windows** bin: Gerekli tüm MFT komutlarını içerir

- **Windows** bin64: Windows 64 bit işletim sistemi desteği için gerekli kitaplıkları içerir
- **Linux** **Windows** java: IBM JRE ve IBM MQ kitaplıklarını içerir
- **Linux** **Windows** licenses: Lisans dosyalarını içerir
- **V 9.3.0** **Windows** META-INF: Kod imzalama bilgilerine sahip dosyaları içerir
- **Linux** **Windows** mqft: Ant desteği ve çekirdek MFT işlev desteği için gerekli ant ve lib dizinlerini içerir
- **Linux** **Windows** swtag: Lisans yöneticileri tarafından makinedeki kuruluşları tanımlamak için gereken swidtag dosyasını içerir

## Sonraki adım

Bir Managed File Transfer Agent yapılındırmaya hazırsınız. Sonraki adımlar için bkz. [“Redistributable Managed File Transfer Agent için ilk yapılandırmanın oluşturulması”](#) sayfa 705.

**V 9.3.0** IBM MQ 9.3.0' den bir Managed File Transfer Loggerda yapılandırabilirsiniz. Kaydediciyi yapılandırmaya ilişkin sonraki adımlar için bkz. [“Redistributable Managed File Transfer Logger için ilk yapılandırmanın oluşturulması”](#) sayfa 707.

## İlgili başvurular

[Redistributable Managed File Transfer components yapılandırılırken olası hatalar](#)

## **Linux** **Windows** Redistributable Managed File Transfer Agent için ilk yapılandırmanın oluşturulması

Var olan bir IBM MQ yapılandırmasına bağlanmak için bir Managed File Transfer Agent yapılandırabilirsiniz.

## Başlamadan önce

Redistributable Managed File Transfer Agent paketinin içeriğini karşıdan yüklediğinizden ve çıkardığınızdan emin olun. Daha fazla bilgi için bkz [“Redistributable Managed File Transfer components ürününü karşıdan yükleme ve yapılandırma”](#) sayfa 703.

## Bu görev hakkında

Önce Redistributable Managed File Transfer Agent ' in gereksinim duyduğu ortamı yaratırsınız. Daha sonra, aracıyı başlatmadan ve doğrulamadan önce, IBM MQ sunucusunda çalışan kuyruk yöneticisiyle bağlantı ayarlayabilir ve bir aracı ve aracı kuyruk yöneticisini yapılandırabilirsiniz.

**V 9.3.0** IBM MQ 9.3.0' den oluşturduğunuz ortam, Redistributable Managed File Transfer Logger ile paylaşılır. Daha fazla bilgi için bkz [“Redistributable Managed File Transfer Logger için ilk yapılandırmanın oluşturulması”](#) sayfa 707.

## Yordam

1. Redistributable Managed File Transfer Agent için ortam oluşturun.

**fteCreateEnvironment** komutunu çalıştırdığınızda, MFT araçlarına ilişkin yapılandırma bilgilerini içeren MFT veri dizini oluşturulur. Karşıdan yüklenen Redistributable Managed File Transfer Agent bileşenini açtığınızda yaratılan bin dizininde bulunduğunuzdan emin olun. Aşağıdaki komutu çalıştırın:

- **Windows**

```
fteCreateEnvironment.cmd -d datapath location
```

```
./fteCreateEnvironment -d datapath location
```

Bu komut aşağıdaki isteğe bağlı parametreleri alır:

#### -d

Bu değiştirge, MFT yapılandırmasının yaratıldığı, saklandığı ve korunduğu veri yolunun yerini belirtir. **fteCreateEnvironment** komutunu veri konumunu belirtmeden çalıştırırsanız, mftdata dizini, Redistributable Managed File Transfer Agent ' in çıkarıldığı konumda oluşturulur.

**Not:** Yeniden dağıtılabılır aracı bir Windows hizmeti olarak çalışacaksa, hizmetin çalışması için sistem ortamında **BFG\_DATA** ortam değişkeninin ayarlanması gerekir.

#### -n kuruluş adı

Bu parametre, bir IBM MQ kuruluşunun adını ya da benzersiz bir adı belirtmek için kullanılır.

Bu parametreyi kullanmak isteyebileceğiniz durumlara örnek olarak şunlar verilebilir:

- Aracılardan yalnızca istemci kipinde kuyruk yöneticisine bağlanacak şekilde yapılandırıldığı var olan yapılandırmayla yeniden dağıtılabılır paketi kullanarak yeni bir işlevi ya da özelliği hızlı bir şekilde sınamak istiyorsanız. (Bu parametrenin, bağ tanımlama kipinde bir kuyruk yöneticisine bağlanmak için yapılandırılmış hiçbir aracı için geçerli olmadığını unutmayın.)
- Standart bir Managed File Transfer kuruluşundan bir Redistributable Managed File Transfer Agent paketine geçiş yapıyorsanız ve standart kuruluş tarafından oluşturulan yapılandırmayla aynı yapılandırmayı kullanmak istiyorsanız. Bu, standart Managed File Transfer ' in kurulduğu, ancak başka bir makinede çalışan bir aracı kuyruk yöneticisine bağlandığı durumdur.

Varsayılan kuruluş adı değişkeni: **BFG\_INSTALLATION\_NAME**.

**fteCreateEnvironment** komutuyla ilgili daha fazla bilgi için bkz. [fteCreateEnvironment \(set up environment for Redistributable Managed File Transfer Agent\)](#).

**BFG\_DATA** ortam değişkenini veri yolu yeriyle de ayarlayabilirsiniz:

```
BFG_DATA=Datapath location
```

Bir aracı ya da diğer komutları oluşturmadan, başlatmadan ve durdurmadan önce, **BFG\_DATA** değişkeninin doğru veri yolu konumuna ayarlandığından emin olmanız gerekir.

## 2. IBM MQ bağlantılığını ayarlayın.

a) **fteSetupCoordination** komutunu kullanarak koordinasyon kuyruk yöneticisini ayarlayın.

**fteSetupCoordination** komutu, koordinasyon kuyruğu yöneticileri ve daha fazla yapılandırma için gerekli olan izinler için gerekli olan kümeye yaratır. Redistributable Managed File Transfer Agent istemci kipinde çalışır; bu nedenle, bağ tanımlama kipi desteklenmediğinden, bir hatayı önlemek için bu komutla ek parametreler sağlamanız gerekir.

```
fteSetupCoordination -coordinationQMGr PRMFTDEM02
                    -coordinationQMGrHost 9.121.59.233 -coordinationQMGrPort 3002
                    -coordinationQMGrChannel SYSTEM.DEF.SVRCONN
```

**fteSetupCoordination** komutunu kullanmaya ilişkin daha fazla ayrıntı ve adımlar için bkz. [fteSetupKoordinasyon](#). Eşgüdüm kuyruk yöneticisinin nasıl yapılandırılacağı hakkında bilgi için bkz. ["MFT için koordinasyon kuyruk yöneticisini yapılandırma" sayfa 746](#).

b) Komut kuyruğu yöneticisini yaratın ve ayarlayın:

```
fteSetupCommands -p PRMFTDEM02 -connectionQMGrHost 9.121.59.233
                  -connectionQMGrPort 3002 -connectionQMGrChannel SYSTEM.DEF.SVRCONN
                  -connectionQMGr PRMFTDEM02 -f
```

**fteSetupCommands** komutunu kullanmaya ilişkin daha fazla ayrıntı ve adımlar için bkz. [fteSetupKomutları: MFT command.properties dosyasını oluşturma](#).

## 3. Bir uç nokta için MFT Agent tanımlaması yaratın.

```
fteCreateAgent -p PRMFTDEM02 -agentQMgriHost 9.121.59.233
-agentQMgriPort 3002 -agentQMgriChannel SYSTEM.DEF.SVRCONN
-agentName AGENT.TRI.BANK -agentQMgri PRMFTDEM02 -f
```

Bir aracı ve aracı kuyruk yöneticisini yapılandırmak için **fteCreateAgent** komutunu kullanma hakkında daha fazla bilgi için bkz. [fteCreateAgent](#).

**Not:** Aracı kuyruk yöneticisinde aracı nesnelere tanımlamak için komut çıktısının bir parçası olarak görüntülenen MQSC komutlarını kullanmanız gerekir; tersi durumda, “4” sayfa 707 numaralı adımdaki yönergeler çalışmaz.

Her aracı için “2” sayfa 706 ve “3” sayfa 706 adımlarında, aracı kuyruk yöneticisinde kuyruk ve konu tanımlamaları yaratırsınız.

4. Aracıyı başlatın ve dosyaları aktarmaya hazırsınız.

```
fteStartAgent -p PRMFTDEM02 AGENT.TRI.BANK
```

Aşağıdaki komutu çalıştırarak aracının durumunu doğrulayabilirsiniz:

```
fteListAgents
```

**fteListAgents** komutunu kullanma hakkında daha fazla ayrıntı için bkz. [fteListAgents](#).

## Sonraki adım

**V 9.3.0** Redistributable Managed File Transfer Logger ürününü yapılandırmak istiyorsanız, “Redistributable Managed File Transfer Logger için ilk yapılandırmanın oluşturulması” sayfa 707’deki adımları tamamlayın.

### İlgili kavramlar

“yapılandırmaManaged File Transfer” sayfa 700

Kuruluştan sonra Managed File Transfer özelliklerini yapılandırabilirsiniz.

“Çoklu platformlarda MFT yapılandırma seçenekleri” sayfa 701

Managed File Transfer içinde, kurulumunuzla ilgili temel bilgileri içeren ve işlem için gerekli olan bir özellikler dosyaları kümesi sağlanır. Bu özellikler dosyaları, ürünü kurarken tanımladığınız yapılandırma dizininde bulunur.

### İlgili başvurular

**fteCreateTransfer:** yeni bir dosya aktarımı başlatır

## **V 9.3.0** **Linux** **Windows** Redistributable Managed File Transfer Logger için ilk yapılandırmanın oluşturulması

İstemci kipinde bir eşgüdüm kuyruğu yöneticisine bağlanmak için Managed File Transfer Logger FILE tipini yapılandırabilirsiniz.

## Başlamadan önce

Redistributable Managed File Transfer Agent paketinin içeriğini karşıdan yüklediğinizden ve çıkardığınızdan emin olun. IBM MQ 9.3.0’den bu paket, Redistributable Managed File Transfer Logger ürününü de içerir. Daha fazla bilgi için bkz. “Redistributable Managed File Transfer components ürününü karşıdan yükleme ve yapılandırma” sayfa 703.

## Bu görev hakkında

Redistributable Managed File Transfer Agent ve Redistributable Managed File Transfer Logger aynı ortamı paylaşır. Bu ortam oluşturulduktan ve IBM MQ bağlantı ayarlandıktan sonra günlük kaydediciyi oluşturabilir ve başlatabilirsiniz.



## Yordam

1. Redistributable Managed File Transfer Agent ve Redistributable Managed File Transfer Logger için paylaşılan ortamın Adım “1” sayfa 705 içinde açıklandığı şekilde oluşturulduğundan ve IBM MQ bağlantılılığının “Redistributable Managed File Transfer Agent için ilk yapılandırmanın oluşturulması” sayfa 705 Adım “2” sayfa 706 ' da açıklandığı şekilde ayarlandığından emin olun.

2. **fteCreateLogger** komutunu kullanarak bir dosya kaydedici oluşturun.

Örneğin:

```
fteCreateLogger FILELOGGER -loggerType FILE -loggerQMGr PRMFTDEMO2  
-loggerQMGrHost 9.121.59.233 -loggerQMGrPort 3003 -loggerQMGrChannel SYSTEM.DEF.SVRCONN  
-fileSize 20MB -fileCount 10 -fileLoggerMode CIRCULAR
```

**fteCreateLogger** komutunu kullanma hakkında daha fazla bilgi için bkz. [fteCreateLogger](#).

3. **fteStartLogger** komutunu kullanarak kaydediciyi başlatın.

**fteStartLogger** komutuyla ilgili daha fazla bilgi için bkz. [fteStartLogger](#).

## İlgili kavramlar

[“yapılandırmaManaged File Transfer” sayfa 700](#)

Kuruluştan sonra Managed File Transfer özelliklerini yapılandırabilirsiniz.

[“Çoklu platformlarda MFT yapılandırma seçenekleri” sayfa 701](#)

Managed File Transfer içinde, kurulumunuzla ilgili temel bilgileri içeren ve işlem için gerekli olan bir özellikler dosyaları kümesi sağlanır. Bu özellikler dosyaları, ürünü kurarken tanımladığınız yapılandırma dizininde bulunur.

## yükseltmeRedistributable Managed File Transfer components

Yeni bir Redistributable Managed File Transfer packagekarşından yükleyerek Redistributable Managed File Transfer components ürününü yükseltebilirsiniz.

## Başlamadan önce

Redistributable Managed File Transfer Agent ve Redistributable Managed File Transfer Loggeriçin yeniden dağıtılabılır lisans koşulları hakkında bilgi için bkz. [IBM MQ Yeniden Dağıtılabılır Bileşenler](#).

**Not:** Advanced Message Security , Redistributable Managed File Transfer packageile desteklenmez.

## Bu görev hakkında

Redistributable Managed File Transfer componentsürününü önceden kurduysanız, yeni bir yeniden dağıtılabılır paketi karşıdan yükleyerek ve içeriği aynı konuma çıkararak bunları büyütebilirsiniz.

## Yordam

1. İşletim sisteminize ilişkin [IBM MQ yeniden dağıtılabılır Managed File Transfer Agent paketini Fix Centraladresinden](#) yükleyin.
2. Tüm Managed File Transfer araçları ve kaydediciyi durdurun ve çalışan Managed File Transfer komutlarının tamamlanmasını bekleyin.
3. Karşıdan yüklediğiniz yeni yeniden dağıtılabılır paketin içeriğini, Redistributable Managed File Transfer components ürününün kurulu olduğu dizinle aynı dizine açarak, var olan Redistributable Managed File Transfer components kuruluşunuz için dosyaları güncelleyin.

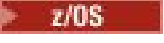


Belirli bir Managed File Transfer Agent ya da Managed File Transfer Logger için Managed File Transfer komut şablonu veri kümesinden belirli bir eşgüdüm için bir PDSE veri kümesi yaratabilirsiniz.

## Bu görev hakkında

Aşağıdaki adımları izleyin:

### Yordam

1. MFT komut şablonu PDSE kitaplık veri kümesi SCSQFCMD ' nin bir kopyasını alın.  
SCSQFCMD yeni bir kitaplığa kopyalanmalıdır; örneğin, *prefix.agent*. JCL. Aşağıdaki yerine koyma değerleriyle SCSQFCMD (BFGCOPY) üyesinin güncellenmiş bir sürümünü kullanabilirsiniz:
  - `++ supplied-library ++` yerine, SCSQFCMD PDSE ' nin tam olarak nitelenmiş adını koyun.
  -  `++ service-library ++` değiştirmesini, yeni MFT komutu PDSE kitaplığı veri kümesinin tam olarak nitelenmiş adıyla değiştirin. `++ service-library ++`, oluşturulan aracı ya da kaydedici hizmeti için çıkış verileri kümeidir.
2. Yeni MFT komutu PDSE kitaplığı veri kümesi için, aracı ya da kaydediciye ilişkin komutları özelleştirmek üzere JCL komut dosyası olan BFGCUSTM üyesini düzenleyin. Her değişken, gerekli değeriyle değiştirmeniz gereken `++ değişken adı ++` biçiminde belirtilir. Çeşitli JCL değişkenlerinin açıklaması için bkz. “z/OS JCL değişkenleri” sayfa 721. BFGSTDIN DD deyimi değişkenleri üç kategoride tanımlar: Değişkenler, Özellikler ve Ortam. Deyimin biçimi şöyledir:

```
[Variables]
variable1=value1
variable2=value2
....
variableN=valueN
[Properties]
property1=property value1
property2=property value2
...
propertyN=property valueN
[Environment]
custom_variable1=value1
custom_variable2=value2
....
custom_variableN=valueN
```

Değişkenler, her komut için gerekli olan ayar ve ortam değişkenleri kümesini tanımlar.

Özellikler, MFT yapılandırma özellikleri için geçersiz kılmaları tanımlar. Ortamınız için aracıyı ya da kaydediciyi özelleştirmek için gerektiği şekilde aracı ve kaydedici özellikleri ekleyebilirsiniz. Tüm özelliklerin bir listesi için bkz. “Yapılandırma özellikleri dosyaları” sayfa 731. Bu olanak, z/OS UNIX System Services dosyaları olarak tutulan MFT yapılandırma özellikleri dosyalarına erişmek zorunda kalmanın kaydedilmesi için sağlar.

Ortam, ek olarak gerekli olan özel ortam değişkenlerini tanımlar.

3. Yeni MFT komutu PDSE kitaplığı veri kümesi için BFGCUSTM işini gönderin. Bu iş, aracı ya da kaydedici için uygun yeni PDSE üyeleri olarak JCL komutları kümesini oluşturur. Komutların tam listesi için bkz. “z/OS aracı ve kaydedici komutu JCL komut dosyaları” sayfa 724.

İş BFGCUSTM, DD deyimi içeren JCL ' yi içeren kitaplığı DISP=OLD ile günceller. İşin yürütülmesine izin vermek için gönderimden sonra düzenleyiciden çıkmanız gerekir.

JCL komut dosyasının başarıyla çalıştırıp çalışmadığını denetlemek için çıkış iş günlüğünü inceleyin. Herhangi bir hata varsa, bunları düzeltin ve BFGCUSTM işini yeniden gönderin.

BFGCUSTM JCL komut dosyası, dosyaları adım içinde tutmak için z/OS UNIX System Services MFT yapılandırma özellikleri dosyalarını da gerektiği şekilde günceller. CoordinationQMgr özelliği tarafından tanımlanan yapılandırma yoksa, uyarı iletileri çıkışa yazılır ve yapılandırma özellikleri

dosyalarını oluşturmak için oluşturulan BFGCFR ve BFGCMCR işlerini çalıştırmanız gerekir. Bir aracı için BFGAGCR ve kaydedici düzenlemesi için BFGLGCRS çalıştırmanız gerekir. Belirtilen yapılandırma zaten varsa, yapılandırma BFTCUSTM JCL komut dosyasında tanımlanan özelliklerle güncellenir.

### İlgili kavramlar

“MFT yapılandırma seçenekleri z/OS” sayfa 702

z/OS üzerindeki Managed File Transfer yapılandırma seçenekleri, dağıtılmış platformlara ilişkin seçenekle aynıdır.

### İlgili görevler

“z/OS üzerinde var olan bir MFT Agent ya da Logger komut veri kümesinin güncellenmesi” sayfa 720

Managed File Transfer komut şablonu veri kümesinden oluşturulan bir Managed File Transfer komutu PDSE kitaplığı veri kümesini güncelleyebilirsiniz.

z/OS

## yapılandırmaManaged File Transfer for z/OS

Managed File Transfer for z/OS , bileşenin doğru şekilde çalışmasını sağlamak için özelleştirme gerektirir.

### Bu görev hakkında

Aşağıdakileri yapmanız gerekir:

1. Yapılandırma verilerini belirtmek için bir PDSE üyesinin düzenlenmesi
2. Eşgüdüm kuyruğu yöneticisini tanımlayın.
3. Komut kuyruğu yöneticisini tanımlayın
4. Bir ya da daha çok aracı yapılandır
5. İsteğe bağlı olarak: Db2 içinde verileri depolamak için bir kaydedici görevi yapılandırın

Gerçekleştirmeniz gereken görevlerin sırası aşağıdaki konularda ayrıntılı olarak açıklanmıştır.

### İlgili kavramlar

“MFT yapılandırmasını gözden geçirme” sayfa 710

Başlamadan önce sisteminizin yapılandırmasını gözden geçirmeniz gerekir.

### İlgili görevler

kurmaIBM MQ Advanced for z/OS

z/OS

## MFT yapılandırmasını gözden geçirme

Başlamadan önce sisteminizin yapılandırmasını gözden geçirmeniz gerekir.

Managed File Transfer (MFT), tanımlı her MFT yapılandırması için bir ya da daha çok kuyruk yöneticisinin aşağıdaki rollerde hareket etmesi gerekir:

- Eşgüdümdeki bir konuda yayınlanan yapılandırmadaki her bir aracının durumuna ilişkin bilgileri tutan bir koordinasyon kuyruk yöneticisi.
- MFT komutları için IBM MQ ağına giriş noktası görevi yapan bir ya da daha çok komut ya da bağlantı kuyruğu yöneticisi.
- Bir MFT aracı ile IBM MQ ağı arasında iletişim sağlayan bir ya da daha fazla aracı kuyruk yöneticisi.

Yukarıdaki rollerin her biri ayrı bir kuyruk yöneticisi tarafından gerçekleştirilebilir ya da rolleri birleştirebilirsiniz; böylece, en basit yapılandırmada tüm roller tek bir kuyruk yöneticisi tarafından gerçekleştirilir.

Var olan bir MFT ortamına z/OS kuyruk yöneticisi ekliyorsanız, z/OS kuyruk yöneticisi ile yapılandırmadaki diğer kuyruk yöneticileri arasında bağlantı tanımlamanız gerekir. Bunu el ile tanımlanan iletim kuyruklarıyla ya da kümeleme kullanımıyla yapabilirsiniz.

Her bir MFT aracı tek bir kuyruk yöneticisiyle iletişim kurar. Birden çok aracı aynı kuyruk yöneticisiyle iletişim kurarsa, aracı kuyruk yöneticisinde her aracı için birden çok kuyruk tanımlanır:

- SYSTEM.FTE.COMMAND.aracı\_adi
- SYSTEM.FTE.DATA.aracı\_adi
- SYSTEM.FTE.REPLY.aracı\_adi
- SYSTEM.FTE.STATE.aracı\_adi
- SYSTEM.FTE.EVENT.aracı\_adi
- SYSTEM.FTE.AUTHAGT1.aracı\_adi
- SYSTEM.FTE.AUTHTRN1.aracı\_adi
- SYSTEM.FTE.AUTHOPS1.aracı\_adi
- SYSTEM.FTE.AUTHSCH1.aracı\_adi
- SYSTEM.FTE.AUTHMON1.aracı\_adi
- SYSTEM.FTE.AUTHADM1.aracı\_adi

SYSTEM . FTE . COMMAND . \*gibi bir profili kullandığınız ya da her bir aracı için belirli profiller tanımlayabileceğiniz soysal güvenlik profilleri tanımlayabileceğinizi unutmayın.

### İlgili kavramlar

“MFT for z/OS ürününü yapılandırmaya başlamadan önce” sayfa 711

Managed File Transfer (MFT) yapılandırması, z/OS UNIX System Services (z/OS UNIX) ve PDSE veri kümelerindeki dosyaları kullanır.

### İlgili başvurular

[MFT sistem kuyrukları ve sistem konusu](#)

## ► z/OS MFT for z/OS ürününü yapılandırmaya başlamadan önce

Managed File Transfer (MFT) yapılandırması, z/OS UNIX System Services (z/OS UNIX) ve PDSE veri kümelerindeki dosyaları kullanır.

Yapılandırmanın ve işlemin çoğu bir PDSE ' den JCL kullanılarak yapılır ve z/OS UNIX ortamında çalışmaya alışmanız gerekir.

OMVS 'ye ISPF ' den erişebilir ya da Telnet tipi oturumunu iş istasyonunuzdaki komutları kullanarak (örneğin, Telnet Putty ya da SSH) kullanabilirsiniz.

ISPF 'deki OMVS' yi kullanıyorsanız, standart ISPF düzenleyicisini kullanabilir ve **oedit** ve **obrowse** komutlarına göz atabilirsiniz.

Aşağıdaki z/OS UNIX komutlarını bilmeniz gerekir

Çizelge 40. Ortak z/OS UNIX komutları	
Komut	İşlev
chmod xxx yolu	Dosya erişim izinlerini değiştirin.
df -k yolu	Dosya sisteminde ne kadar boş alan olduğunu bildirir. -k , KB cinsinden boş alanı bildirir.
du -kt yolu	Yol altındaki dizinlerin boyutlarını raporlar. KB cinsinden bildirilen boyut.
find path -name xxx	Yol dizininde xxxx adlı dosyayı arayın. xxx büyük ve küçük harfe duyarlıdır ve *zzz gibi olabilir.
ls -ltrd dizini	Dizindeki dosyalar yerine, belirtilen dizinle ilgili bilgileri listeler.
ls -ltr yolu	Yoldaki dosyalarla ilgili bilgileri listeler.
obrowse dosya adı	Dosya adına göz atın.

Çizelge 40. Ortak z/OS UNIX komutları (devamı var)

Komut	İşlev
oedit dosya adı	OMVS içindeki bir dosyayı düzenleyin.

Aşağıdaki tablodaki öğeleri gözden geçirin ve tabloyu işletmeniz için uygun girdilerle tamamlayın. BFGCUSTM üyesini düzenlerken bu değerlere gereksinim duyarsınız.

Çizelge 41. Üye BFGCUSTM için gerekli parametreler

Ad	Örnek veriler	Yorumlar
ADMIN_JOB1		İş kartı. Tüm işler aynı JCL kartıyla oluşturulur.
armELEMENT	ARM kullanılıyorsa, bu aracı ya da kaydedici için ARM ilkesinde belirtilen ARM ELEMENT değerini kullanın. ARM kullanılmıyorsa, bu parametreyi boşluk olarak ayarlayın; örneğin, armELEMENT=	
armELEMENTYPE	ARM kullanılıyorsa, ARM ilkesinde belirtilen ARM ELEMENTYPE değerini kullanın. Örneğin, bir aracı için armELEMENTYPE= SYSBFGAG ya da günlüğe kaydedici için armELEMENTYPE= SYSBFLG. ARM kullanılmıyorsa, bu parametreyi boşluk olarak ayarlayın; örneğin, armELEMENTYPE=	
BFG_VERİLERİ		Gerektiği şekilde tamamla
BFG_GRUPLARI_ADI	MQM	
BFG_JAVA_HOME	/java/java71_bit64_GA/J7.1_64/	
BFG_JVM_PROPERTIES		Gerektiği şekilde tamamla
BFG_PROD	/mqm/V9R2M0/mqft	IBM MQ for z/OS UNIX System Services Components dizini altındaki mqft dizininin tam yolu.
BFG_WTO	EVET	Sistem günlüğünde bir MFT iletisi almak için.
CLEAN_AGENT_PROPS	-trs	Bu parametre, BFGAGCL üyesi çalıştırıldığında bir aracıyı temizlemek için kullanılacak seçenekleri belirler. Bu parametreye ilişkin geçerli değerler hakkında daha fazla bilgi için bkz. <a href="#">fteCleanAgent: bir MFT Agent 'ın temizlenmesi.</a>
coordinationQMgr	MQPV	Zorunlu yapılandırma
CREDENTIAL_YOLU		Geçişte kullanılır
Db2_HLQ	SYS2.Db2.V10	

Çizelge 41. Üye BFGCUSTM için gerekli parametreler (devamı var)		
Ad	Örnek veriler	Yorumlar
DB_PROPS_YOLU		Geçişte kullanılır
FTE_YAPILANISI		Geçişte kullanılır
JOBCARD1		Bu, uzun süren görevler, araçlar ve kaydediciler için iş kartıdır.
KİTAPLIK	SCEN.FTE.JCL	MFT PDSE ' nin adı. Her aracı ya da kaydedici görevi için bir kopya gerekir.
MQ_HLQ	IBM MQ veri kümeleri için üst düzey niteleyici. Örnek: MQM.V920	
MQ_LANG	E	
MQ_PATH	/mqm/V9R2M0	IBM MQ for z/OS UNIX System Services Componentskuruluşunun tam izin yolu.
AD	AGENT1	
Çıkış_Sınıfı	*	
Yol	bin:/usr/bin:/usr/sbin	
productId	GELİŞEN	Bu parametre, Managed File Transfer kullanımı kaydedilecek ürün tipini ayarlamak için kullanılır. Bu parametreye ilişkin geçerli değerler hakkında bilgi için bkz. <a href="#">fteSetProductId: set z/OS SCRT recording product id.</a>
QMGR	MQPV	
HİZMETİ_TIPI	AGENT ya da LOGGER	
TMPDIR	/tmp	Geçici dosyalar için erişilebilir z/OS UNIX yolunu okuyun ve yazın.

Ayrıca, aşağıdaki değişkenleri gözden geçirmeniz ve gerektiğinde değerleri sağlamanız gerekir:

- coordinationQMGrAnasistem =
- coordinationQMGrKapısı =
- coordinationQMGrKanal =
- connectionQMGr=
- connectionQMGrAnasistem =
- connectionQMGrKapı =
- connectionQMGrKanal =

Bu özellikler AGENT ya da LOGGER için ortaktır.

**Not:** İstemci bağlantısı için anasistem, kapı ve kanal gereklidir, ancak yerel makinedeki bağ tanımları bağlantısı için boş bırakılmalıdır.

#### İlgili kavramlar

“Denetlenecek öğeler” sayfa 714

Yeterli disk alanınızın, veri saklamak için bir dizinin ve gerekli dosyaların var olduğundan emin olun.

“Üye BFGCUSTM düzenleniyor” sayfa 716

İşçi çalıştırmadan önce, BFGCUSTM üyesini düzenlemeniz ve işletmenizin kullandığı parametrelere ilişkin değerleri girmeniz gerekir.

## Denetlenecek öğeler

Yeterli disk alanınızın, veri saklamak için bir dizinin ve gerekli dosyaların var olduğundan emin olun.

### Yeterli disk alanınız olup olmadığını denetleyin

Konfigürasyona özgü dosyaları saklayacağınız dosya sisteminde yeterli disk alanınız olup olmadığını denetleyin.

Bir aracı izlemesi etkinleştirilirse, varsayılan olarak 100 MB disk alanı kullanabilir.

Yapılandırma dosyalarının kendisi küçüktür, yalnızca birkaç KB boyutunda.

İki aracı ve bir kaydedici kullanmayı planlıyorsanız, en az 300 MB gerekir. **df -k** pathkomutunu kullanabilirsiniz; burada path , kuruluşa özgü dosyaların konumudur. Bu, kullanılabilir ve toplam alanı KB olarak verir.

300 MB 307.200 KB olduğundan en az 310.000 KB ' ye izin vermelisiniz

### Managed File Transfer verilerini saklamak için izin yaratılması ve denetlenmesi

Managed File Transfer (MFT) verilerini saklamak için bir izin gerekir.

**df -k /vardosya** sisteminde yeterli alanınız olup olmadığını denetleyin. Bu dosya sisteminde en az 310.000 KB kullanılabilir olmalıdır.

Bu dosya sisteminin oluşturmadıysanız, **mkdir** komutunu kullanın; örneğin **mkdir /var/mft**.

**ls -ltrd /var/mft** komutunu kullanarak, kullanıcıların bu izin üzerindeki izinlerini görüntüleyin.

Sahip ya da grup doğru değilse, **chown owner:group /var/mft** komutunu kullanın.

Gruba ilişkin izinler doğru değilse, sahip ve gruba okuma, yazma ve yürütme izinleri vermek için aşağıdaki komutu kullanın. Aşağıdaki komutun tüm kullanıcılara okuma ve yürütme izinleri de verdiğini unutmayın **chmod 775 /var/mft**.

### Dosyaların var olup olmadığını ve bunlara erişiminiz olup olmadığını denetleyin.

Uyarılama sırasında kullanacağınız dosyalar için **ls -ltr** komutunu kullanın. Örneğin:

```
ls -ltrd /java/java71_bit64_GA/J7.1_64/bin
```

verir

```
drwxr-xr-x 4 SYSTASK TSouser 8192 Nov 15 2013 /java/java71_bit64_GA/J7.1_64/bin
```

Burada **drwxr-xr-x** ,

**d**

Bu bir izin.

**rwX**

**SYSTASK** iyesinin dizine okuma, yazma ve yürütme erişimi vardır.

**r-x**

**TSouser** grubundaki kişiler dizindeki dosyaları okuyabilir ve yürütebilir.

**r-x**

Evrensel erişim (yani, herkes dizindeki dosyaları okuyabilir ya da yürütebilir).

Belirtilen dosyaları denetle:

Çizelge 42. Kullanıcıların belirli dosyalara erişmesi gerekiyor

Yol	Yapılandırmayı yapan kullanıcılar için erişim gerekli
BFG_JAVA_HOME	Oku ve yürüt
/tmp	Okuma ve yazma
BFG_PROD	<
BFG_VERİLERİ	Yaz
MQ_PATH	<

### İlgili kavramlar

“MFT for z/OS ürününü yapılandırmaya başlamadan önce” sayfa 711

Managed File Transfer (MFT) yapılandırması, z/OS UNIX System Services (z/OS UNIX) ve PDSE veri kümelerindeki dosyaları kullanır.

“z/OS yapılandırmaları için ortak MFT” sayfa 715

Farklı Managed File Transfer yapılandırmalarına genel bakış

### z/OS z/OS yapılandırmaları için ortak MFT

Farklı Managed File Transfer yapılandırmalarına genel bakış

Managed File Transfer , veri aktarmak için bir kuyruk yöneticisine bağlı araçları kullanır.

MFT birden çok kuyruk yöneticisi kullanabilir:

- Verileri aktarmak için bir ya da daha çok kuyruk yöneticisi.
- İstekleri veren bir komut kuyruğu yöneticisi. Örneğin, bu kuyruk yöneticisine aktarma başlatma isteği gönderilir ve ilişkili komutlar MFT araçlarına yönlendirilir.
- İşi yöneten bir koordinasyon kuyruk yöneticisi.

Üç ortak Managed File Transfer (MFT) yapılandırması vardır:

1. Yerel bağlantıları kullanan bir ya da daha çok aracıya sahip tek bir kuyruk yöneticisi. Bu, bir veri kümesinin içeriğini IBM MQ kuyruklarına koymak için kullanılabilir.
2. İstemci bağ tanımlarını kullanan dağıtılmış bir makinede MFT istemcisi olan tek bir kuyruk yöneticisi.
3. Kanallar tarafından bağlanan iki kuyruk yöneticisi ve her makinede bir ya da daha fazla aracı. Bu araçlar istemci ya da yerel bağ tanımları olabilir.

Aşağıdaki noktaları göz önünde bulundurun:

1. MFT, bazı kabuk komut dosyaları ve MFT ' yi yapılandırmak ve çalıştırmak için JCL ile Java içinde yazılır.
2. Db2 durumu ve etkinliği günlüğe kaydedilebilir ve bu, Db2 tablolarında saklanabilir.
3. MFT ' yi yapılandıran kişi z/OS UNIX System Services (z/OS UNIX) ile ilgili bilgi sahibi olmalıdır.

Örneğin:

- /u/userID/myfile.txt2 gibi adlara sahip dosyaların bulunduğu dizin yapısı
- z/OS UNIX komutları, örneğin:

**cd** (dizini değiştir)

**ls** (Liste)

**chmod** (dosya izinlerini değiştirme)

**chown** (dosya iyeliğini ya da dosyaya ya da dizine erişebilen grupları değiştirme)

4. MFT ' yi yapılandırabilmek ve çalıştırabilmek için z/OS UNIX içinde aşağıdaki ürünler gereklidir:

- Java; örneğin, /java/java71\_bit64\_GA/J7.1\_64/
- IBM MQ V920, örneğin /mqm/V9R2M0.

- Db2 JDBC kitaplıkları; örneğin, durum ve geçmiş için Db2 kullanmak istiyorsanız; /db2/db2v12/jdbc/lib

Bir koordinasyon kuyruk yöneticisine ihtiyacınız var. Ancak, araçları çalıştırmak, komutları işlemek ve eşgüdüm için aynı kuyruk yöneticisini kullanabilirsiniz. Birden çok kuyruk yöneticisi kullanıyorsanız, eşgüdümcü olarak işlev göreceği birini seçmeniz gerekir.

### IBM MQ bağlantılığını denetleyin

Varolan bir MFT eşgüdümcü kuyruk yöneticiniz varsa, yapılanışı gerçekleştirdiğiniz kuyruk yöneticisi ile eşgüdümlü ve komut kuyruğu yöneticileri arasında bağlantılılık gerekir.

## z/OS JCL kitaplığı yaratmak için SCSQFCMD ' yi kopyala

Her aracı ve kaydedici için bir JCL kitaplığı oluşturmanız gerekir. JCL, aracıyı ya da kaydediciyi oluşturmak ve çalıştırmak için kullanılan yapılandırma ve işleri içerir.

Her aracı ve kaydedici için, BFGCOPY üyesini düzenleyip çalıştırarak IBM tarafından sağlanan SCSQFCMD kitaplığının bir kopyasını yaratın.

Bu kitaplık, aracı ya da kaydedici için yapılandırmayı tanımlamak için kullanılır ve özelleştirmeden sonra, gerekli Managed File Transfer yapılandırmasını ve aracısını ya da kaydediciyi oluşturmak için kullanılabilir işleri içerir.

Bu işlemin bir parçası olarak üye BFGCUSTM oluşturursunuz.

**Not:** z/OS UNIX komutlarını biliyorsanız, z/OS ürününü diğer platformlarda kullandığınız komutlarla yapılandırabilirsiniz.

### İlgili kavramlar

[“z/OS yapılandırmaları için ortak MFT” sayfa 715](#)

[Farklı Managed File Transfer yapılandırmalarına genel bakış](#)

[“Üye BFGCUSTM düzenleniyor” sayfa 716](#)

İşleri çalıştırmadan önce, BFGCUSTM üyesini düzenlemeniz ve işletmenizin kullandığı parametrelere ilişkin değerleri girmeniz gerekir.

## z/OS Üye BFGCUSTM düzenleniyor

İşleri çalıştırmadan önce, BFGCUSTM üyesini düzenlemeniz ve işletmenizin kullandığı parametrelere ilişkin değerleri girmeniz gerekir.

Belirli değerler gerektiren parametrelerin bir listesi için [Üye BFGCUSTM için gerekli parametreler](#) başlıklı konuya bakın.

Ayrıca, aşağıdaki değişkenleri gözden geçirmeniz ve gerektiğinde değerleri sağlamanız gerekir:

- coordinationQMGrAnasistem =
- coordinationQMGrKapısı =
- coordinationQMGrKanal =
- connectionQMGr=
- connectionQMGrAnasistem =
- connectionQMGrKapı =
- connectionQMGrKanal =

Bu özellikler AGENT ya da LOGGER için ortaktır.

**Not:** İstemci bağlantısı için anasistem, kapı ve kanal gereklidir, ancak yerel makinedeki bağlantı tanımları bağlantısı için boş bırakılmalıdır.



Bu, Managed File Transfer ortamınızdaki ilk kuyruk yöneticisiyse ve koordinasyon, komutlar ve çalışan araçlar için aynı kuyruk yöneticisini kullanmak istiyorsanız, değerleri yerel kuyruk yöneticisi adına ayarlayın.

```
coordinationQMGr=MQPV  
connectionQMGr=MQPV
```

Burada MQPV , yerel kuyruk yöneticisi adınızı belirtir.

PDSE ' yi güncelleyen işi sunun ve belirtilen yol altında bir dizin yapısı yaratır.

Bu işin özel kullanım gerektirdiğini, bu nedenle iş çalıştırılırken PSDE ' yi kullanmayı durdurmanız gerektiğini unutmayın.

**İpucu:** BFGCUSTM işini gönderdiğinizde, iş tüm JCL dosyalarının yerini alır. Değiştirdiğiniz her üyeyi yeniden adlandırmanız gerekir.

### İlgili kavramlar

“MFT for z/OS ürününü yapılandırmaya başlamadan önce” sayfa 711

Managed File Transfer (MFT) yapılandırması, z/OS UNIX System Services (z/OS UNIX) ve PDSE veri kümelerindeki dosyaları kullanır.

“Bir Aracı Oluşturma” sayfa 719

Aracıya özgü PDSE 'yi (örneğin, *user.MFT.AGENT1*). Varsa, PDSE ' yi önceki bir aracı ya da kaydedici yapılandırmasından kopyalayın. Bu ilk yapılandırmanız ise, MFT ile birlikte sağlanan PDSE ' yi kopyalayın.

## z/OS Eşgüdüm kuyruğu yöneticisinin tanımlanması

Managed File Transfer , eşgüdüm kuyruğu yöneticisi olarak işlev gören bir kuyruk yöneticisinin yaratılmasını gerektirir.

Seçtiğiniz yapılanışa bağlı olarak, bu kuyruk yöneticisi yerel MVS sisteminde ya da başka bir makinede bulunur. Önceki durumda, bağlantı bağlantıları bağ tanımlama bağlantılarıdır ve ikinci durumda, bunlar istemci bağlantılarıdır.

Yapılandırma adınızı başarıyla çalıştırdıktan sonra PDSE ' de yapılandırılmış üyeler vardır.

Üye BFGCFR, eşgüdüm kuyruğu yöneticisini ve bu işi tanımlar:

1. Managed File Transfer (MFT) dizininde bir dizin yapısı oluşturur ve yapılandırma dosyaları oluşturur.
2. IBM MQ kaynaklarını tanımlamak için CSQUTIL yürütür.

Eşgüdüm kuyruğu yöneticisi uzak bir makadaysa, bu iş adımı başarısız olur.

Üye BCFCFR z/OS UNIX System Services içinde dosyalar yaratır ve MQ tanımlamaları yaratır. Bu iş:

1. Bir MFT konusu oluşturur,
2. Bir MFT kuyruğu oluşturur
3. *NAMELIST (SYSTEM.QPUBSUB.QUEUE.NAMELIST) AD (SYSTEM.BROKER.DEFAULT.STREAM, SYSTEM.BROKER.ADMIN.STREAM, SYSTEM.FTE)*
4. *ALTER QMGR PSMODE (ENABLED)* işlemini gerçekleştirir.

*DISPLAY NAMELIST (SYSTEM.QPUBSUB.QUEUE.NAMELIST)* komutu, değiştirme işleminden önce verilir. NAMLIST varsayılan değer değilse, ad listenizi SYSTEM.FTE TZE

BCFCFR üyesini kendi önekinizle (örneğin, CCPCFR) yeniden adlandırın; bu dosyayı yeniden uyarlamaz bu dosyanın yerini alır.

Kimlik bilgileri dosyanızın adını ekleyerek bu yeniden adlandırılan üyeyi düzenleyin. Örneğin:

```
%BFGCMD CMD=fteSetupCoordination +  
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
```

İş i kaydedin ve gönderin. İş i yeniden sunmanız gerekiyorsa, *-f* seçeneğini eklemeniz gerektiğini unutmayın.

Bu iş çalıştığı nda, yarattığı IBM MQ kaynaklarını listeler. Bu kaynakları korumalısınız.

```
DEFINE TOPIC('SYSTEM.FTE') TOPICSTR('SYSTEM.FTE') REPLACE
ALTER TOPIC('SYSTEM.FTE') NPMGDLV(ALLAVAIL) PMSGDLV(ALLAVAIL)
DEFINE QLOCAL(SYSTEM.FTE) LIKE(SYSTEM.BROKER.DEFAULT.STREAM) REPLACE
ALTER QLOCAL(SYSTEM.FTE) DESCR('Stream for MFT Pub/Sub interface')
* Altering namelist: SYSTEM.QPUBSUB.QUEUE.NAMELIST
* Value prior to alteration:
DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)
ALTER NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST) +
NAMES(SYSTEM.BROKER.DEFAULT.STREAM+
,SYSTEM.BROKER.ADMIN.STREAM,SYSTEM.FTE)
* Altering PSMODE. Value prior to alteration:
DISPLAY QMGR PSMODE
ALTER QMGR PSMODE(ENABLED)
```

## İlgili görevler

[“Komut kuyruğu yöneticisinin tanımlanması” sayfa 718](#)

Eşgüdüm ve komut kuyruğu yöneticileriyle aynı kuyruk yöneticisini kullanabilir ya da yeni bir komut kuyruğu yöneticisi yaratabilirsiniz.

## Komut kuyruğu yöneticisinin tanımlanması

Eşgüdüm ve komut kuyruğu yöneticileriyle aynı kuyruk yöneticisini kullanabilir ya da yeni bir komut kuyruğu yöneticisi yaratabilirsiniz.

## Bu görev hakkında

Bir komut kuyruğu yöneticiniz olmalıdır; ancak, eşgüdüm ve komut kuyruğu yöneticileri için aynı kuyruk yöneticisini kullanabilirsiniz. Ters durumda, yeni bir komut kuyruğu yöneticisi yaratmanız gerekir. Bu, koordinasyon kuyruk yöneticisiyle aynı makinede olabilir, ancak olması gerekmez.

## Yordam

1. BFGCMCR üyesini kendi önekinizle yeniden adlandırın; örneğin, CCPCMCR.  
Bu dosyayı yeniden özelleştirdiğiniz için BFGCMCR ' yi yeniden adlandırmanız gerekir.
2. Kimlik bilgileri dosyanızın adını ekleyerek yeniden adlandırılan üyeyi düzenleyin.  
Örneğin:

```
%BFGCMD CMD=fteSetupCommands +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
```

3. İş i kaydedin ve gönderin.  
İş i yeniden sunmanız gerekiyorsa, *-f* seçeneğini eklemeniz gerektiğini unutmayın.  
Bu kuyruk yöneticisi, **ftePingAgent** gibi komutlar için kullanılır.
4. Bu üyeyi gözden geçirin, gönderin ve çıkışı gözden geçirin.

## Sonraki adım

Bir aracı nasıl oluşturacağınıza ilişkin bilgi için bkz. [“Bir Aracı Oluşturma” sayfa 719](#) .

## İlgili kavramlar

[“Eşgüdüm kuyruğu yöneticisinin tanımlanması” sayfa 717](#)

Managed File Transfer , eşgüdüm kuyruğu yöneticisi olarak işlev gören bir kuyruk yöneticisinin yaratılmasını gerektirir.

## İlgili görevler

[MQMFTCredentials.xml ' in yapılandırılması](#)

## İlgili başvurular

[MFT kimlik bilgileri dosya biçimi](#)

## Bir Aracı Oluşturma

Aracıya özgü PDSE 'yi (örneğin, *user.MFT.AGENT1*. Varsa, PDSE ' yi önceki bir aracı ya da kaydedici yapılandırmasından kopyalayın. Bu ilk yapılandırmanız ise, MFT ile birlikte sağlanan PDSE ' yi kopyalayın.

BFGCUSTM üyesini gözden geçirin ve farklı bir kimlik bilgileri dosyası kullanmanız gerekirse bir kimlik bilgileri dosyası oluşturun.

İçeriğin çoğu, [“Üye BFGCUSTM düzenleniyor” sayfa 716](#) içinde ayrıntılı olarak açıklanan özelleştirmeden aynı kalır.

Şunları değiştirmeniz gerekir:

- // SYSEXEC DD DSN=SCEN.FTE.JCL.AGENT1
- Aracı PDSE ile eşleşecek KİTAPLIK
- SERVICE\_TYPE=AGENT
- Aracının adı olacak AD (PDSE ile eşleşen) JOBCARD
- BFG\_JVM\_PROPERTIES = "-Xmx1024M" değerini değiştirin

İşin veri kümesine özel erişim gerektirdiğini anımsayarak bu işi gönderin.

Aracının tüm işleri *BFGAG\** biçiminde adlara sahip

*BFGAGCR* adlı üyeyi yeniden adlandırın. Bu iş, Managed File Transfer dizinindeki dosyaları günceller ve yerel kuyruk yöneticisinde aracıya özgü kuyruklar yaratmak için CSQUTIL kullanır. Kimlik bilgileri dosyanızın adını belirtin; örneğin, *-credentialsFile // 'SCEN.FTE.JCL.VB(CREDOLD)*. Adı belirtmezseniz, aracıyı başlatma işi bir kimlik bilgileri dosyası kullanmaz.

İşlemin başarıyla çalıştığından emin olmak için çıkışı denetleyin.

**İpucu:** *agent.properties* dosyasının yol adını işin çıkışından aracıya ilişkin PDSE içindeki bir üyeye kopyalayın.

Örneğin, */u/userid/fte/wmqmft/mqft/config/MQPA/agents/AGENT1/agent.properties* dosyasını üye AGENT ' e kopyalayın.

Bu, özellikler dosyasını görüntülemeniz ve */u/userid/fte/wmqmft/mqft/logs/MQPA/agents/AGENT1/logssatırını* eklemeniz gerekiyorsa kullanışlıdır.

Bu, izleme dosyalarının saklandığı yerdir.

## İlgili kavramlar

[“Eşgüdüm kuyruğu yöneticisinin tanımlanması” sayfa 717](#)

Managed File Transfer , eşgüdüm kuyruğu yöneticisi olarak işlev gören bir kuyruk yöneticisinin yaratılmasını gerektirir.

[“Aracıyı kullanma” sayfa 719](#)

Aracının doğru çalıştığından emin olmak için çeşitli komutları nasıl kullandığınızı.

## İlgili görevler

[“Komut kuyruğu yöneticisinin tanımlanması” sayfa 718](#)

Eşgüdüm ve komut kuyruğu yöneticileriyle aynı kuyruk yöneticisini kullanabilir ya da yeni bir komut kuyruğu yöneticisi yaratabilirsiniz.

## Aracıyı kullanma

Aracının doğru çalıştığından emin olmak için çeşitli komutları nasıl kullandığınızı.

## Aracıyı başlat

BFGAGST üyesini yeniden adlandırın, üyeyi gözden geçirin ve işi gönderin.

Bu işe yararsa, BFGAG0059Iiletisini alırsınız: Aracı başarıyla başlatıldı.

## Etkin araçları görüntüle

BFGAGLI üyesini yeniden adlandırın, üyeyi gözden geçirin ve eşgüdümçü kuyruk yöneticisini kullanan işi gönderin.

Bağlanırlık sorunlarını çözmeniz gerekir

## Aracının çalıştığını denetlemek için aracıya ping komutu gönder

BFGAGPI üyesini yeniden adlandırın, üyeyi gözden geçirin ve komut kuyruğu yöneticisini kullanan işi gönderin.

Bağlanırlık sorunlarını çözmeniz gerekir

## Test aktarımını gerçekleştireceğini

Daha fazla bilgi için bkz. [“Doğrulama aktarımı gerçekleştiriliyor” sayfa 726](#) .

## Aracıyı durdur

BFGAGSP üyesini yeniden adlandırın, üyeyi gözden geçirin ve işi gönderin.

BFGAGST üyesini kullanarak aracıyı yeniden başlatın.

### İlgili kavramlar

[“Bir Aracı Oluşturma” sayfa 719](#)

Aracıya özgü PDSE 'yi (örneğin, *user.MFT.AGENT1*. Varsa, PDSE ' yi önceki bir aracı ya da kaydedici yapılandırmasından kopyalayın. Bu ilk yapılandırmanız ise, MFT ile birlikte sağlanan PDSE ' yi kopyalayın.

## z/OS üzerinde var olan bir MFT Agent ya da Logger komut veri kümesinin güncellenmesi

Managed File Transfer komut şablonu veri kümesinden oluşturulan bir Managed File Transfer komutu PDSE kitaplığı veri kümesini güncelleyebilirsiniz.

## Yordam

1. BFGCUSTM JCL komut dosyası üyesini düzenleyin ve BFGSTDIN DD deyimindeki değişkenleri ve özellikleri güncelleyin.

Önceden tanımlanmış bir özelliği kaldırmak istiyorsanız, girdiyi kaldırmak yerine değerini boş olarak ayarlayın. BFGCUSTM JCL komut dosyası çalıştırıldığında, belirtilen özellikler, gerçek aracı ve kaydedici z/OS UNIX System Services özellik dosyalarına güncelleme olarak uygulanır; bir özelliğin boş bir değere ayarlanması, özelliğin kaldırılacağını gösterir.

2. BFGCUSTM işini gönderin. Bu iş, aracı ya da kaydedici için uygun JCL komutları kümesini yeniden oluşturur. Komutların tam listesi için bkz. [“z/OS aracı ve kaydedici komutu JCL komut dosyaları” sayfa 724](#). JCL komut dosyasının başarıyla çalıştırıp çalışmadığını denetlemek için çıkış iş günlüğünü inceleyin. Herhangi bir hata varsa, bunları düzeltin ve BFGCUSTM işini yeniden gönderin.

## Sonuçlar

Oluşturulan JCL komut dosyalarını değiştirebilir ve kendi mantığınızı ekleyebilirsiniz. Ancak, özel mantığın üzerine yazabileceğiniz için BFGCUSTM ' yi yeniden çalıştırırken dikkatli olun.

### İlgili kavramlar

[“MFT yapılandırma seçenekleri z/OS” sayfa 702](#)

z/OS üzerindeki Managed File Transfer yapılandırma seçenekleri, dağıtılmış platformlara ilişkin seçenekle aynıdır.

### İlgili görevler

“MFT Agent ya da Logger komut veri kümesi yaratılması” sayfa 709

Belirli bir Managed File Transfer Agent ya da Managed File Transfer Logger için Managed File Transfer komut şablonu veri kümesinden belirli bir eşgüdüm için bir PDSE veri kümesi yaratabilirsiniz.

### **z/OS** z/OS JCL değişkenleri

BFGCUSTM komut dosyasında yerine koyma değerlerini, JCL değişkenlerini ve yapılandırma özelliklerini kullanabilirsiniz.

Aşağıdaki çizelgede, bir MFT komutu PDSE kitaplığı veri kümesindeki BFGCUSTM JCL komut dosyasına ilişkin yerine koyma değerleri listelenmektedir. BFGCUSTM işini göndermeden önce bu yerine koyma değerlerini uygun değerlerle değiştirmeniz gerekir.

Çizelge 43. BFGCUSTM JCL komut dosyası için yerine koyma değerleri	
Yerine koyma değişkeni	Değer
++ kitaplık ++	İçeren MFT komutu PDSE kitaplığının veri kümesi adı.
++ bfg_java_home ++	Java kuruluşunuzun konumu.
++ mq_path ++	IBM MQ for z/OS UNIX System Services Components dizininin yolu. Örneğin, /mqm/V9R2M0. Bu, MFT kuruluşunun tam yolunu vermek için kullanılır; örneğin, /mqm/V9R2M0/mqft.

Aşağıdaki çizelgede, bir MFT komutu PDSE kitaplığı veri kümesindeki ([ Değişkenler] bölümünde) BFGCUSTM JCL komut dosyasına ilişkin BFGSTDIN DD deyimine ilişkin ortam değişkenleri açıklanmaktadır. BFGCUSTM işini göndermeden önce, belirtilen tüm değişkenleri yerine koyma değerleriyle (yani, iki artı işareti içine alınmış değerler, ++ ) uygun değerlerle değiştirmeniz gerekir.

Çizelge 44. Ortam değişkenleri	
Ortam değişkeni	Değer
KİTAPLIK	İçeren MFT komutu PDSE kitaplığının veri kümesi adı.
TMPDIR	Geçici dosyalar için z/OS UNIX System Services dizini.
BFG_PROD	IBM MQ for z/OS UNIX System Services Components dizini altındaki mqft dizininin tam yolu; örneğin: /mqm/V9R2M0/mqft.
BFG_VERİLERİ	DATA_DIR yolu olan Managed File Transfer for z/OS veri dizininin konumu.
BFG_JAVA_HOME	Java kuruluşunuzun konumu.
BFG_JVM_PROPERTIES	İsteğe Bağlı. BFG_JVM_PROPERTIES ortam değişkeni için bir değer belirler. Bu özellikler Java sanal makinesine iletilir.

Çizelge 44. Ortam değişkenleri (devamı var)

Ortam değişkeni	Değer
BFG_GRUPLARI_ADı	<p>mqm dosya grubu genellikle MFT yapılandırma verileri dosyaları ve komutlarıyla ilişkilendirilir. Sonuç olarak, mqm grubunda üye olan tüm kullanıcılar MFT yapılandırmasına erişebilir ve bu yapılandırmada değişiklik yapabilir. Daha fazla bilgi için, bkz. <a href="#">IBM MQ içinde MFT için dosya sistemi izinleri</a>.</p> <p>Bir z/OS sistemi için, dosya grubu bir z/OS UNIX System Services (z/OS UNIX) dosya sistemi varlığı ve mqm dosya grubunun tanımlanmış olması gerekmez. BFG_GROUP_NAME ortam değişkenini kullanarak MFT yapılandırma verileri dosyaları için bir z/OS UNIX dosya sistemi grubunu ilişkilendirebilirsiniz. Örneğin, z/OS UNIX kabuk komut isteminde şunu kullanın:</p> <pre>export BFG_GROUP_NAME=FTEGB</pre> <p>geçerli z/OS UNIX oturumu için sonradan oluşturulan yapılandırma dosyalarıyla ilişkilendirilecek FTEGB grubunu tanımlar.</p> <p>BFG_GROUP_NAME değerini boş bir değere ayarlayabilir ya da kaldırabilirsiniz.</p> <p><b>Not:</b> BFGCUSTM ilk kez çalıştırılırken, MFT yapılandırması birden çok kullanıcı kimliği tarafından kullanılacaksa, BFG_GROUP_NAME değerinin, tüm gerekli kullanıcı kimlikleri tarafından erişilebilir bir gruba ayarlanması önemlidir. BFGCUSTM yeniden çalıştırılırsa, BFG_GROUP_NAME değiştirilmemelidir (tersi durumda, BFG_DATA tarafından başvuru dizindeki tüm dosyalar ve dizinler için z/OS UNIX grup dosyası izinleri de yeni BFG_GROUP_NAME ayarını yansıtacak şekilde değiştirilmelidir).</p>
BFG_WTO	<p>z/OS günlük kaydı, BFG_WTO için YES, ON ya da TRUE değeri belirlendiğinde etkinleştirilir. Bu, aracı olay günlüğüne yazılan iletilerin, JCL ' den bir aracı çalıştırdığınızda otomasyon ürünleri için daha kolay erişim sağlayan z/OS işletmen günlüğü olanağına yazılıp yazılmayacağını denetler. Yönelme kodu, Programcı Bilgileri (11) ve tanımlayıcı kodu Bilgi 'dir (12).</p>
HIZMETI_TIPI	<p>MFT komut kitaplığının bir aracı için mi, yoksa kaydedici için mi olduğunu belirtir. Geçerli değerler şunlardır: AGENT ya da LOGGER.</p>
AD	<p>SERVICE_TYPE değerine ilişkin aracının ya da kaydedicinin adı.</p>
QMGR	<p>SERVICE_TYPE değeri için aracı ya da kaydedici ile ilişkili yerel kuyruk yöneticisinin adı.</p>

Çizelge 44. Ortam değişkenleri (devamı var)	
Ortam değişkeni	Değer
ÇIKIŞ_SINIFI	SYSOUT veri kümeleri için dış sınıf. Varsayılan değer olarak, iş deyiminden MSGCLASS parametresiyle aynı çıkış sınıfını isteyen * kullanılır.
MQ_PATH	IBM MQ for z/OS UNIX Components dizininin yolu.
MQ_HLQ	IBM MQ veri kümeleri için üst düzey niteleyici.
MQ_LANG	Gerekli olan dil.
DB2_HLQ	İsteğe Bağlı. Db2 veri kümeleri için üst düzey niteleyici.
JOBCARD1	JCL komut işi için üstbilgi satırı 1.
JOBCARD2	JCL komut işi için üstbilgi satırı 2.
JOBCARD3	JCL komut işi için üstbilgi satırı 3.
ADMIN_JOB1	Bir yönetici işi için üstbilgi satırı 1.
ADMIN_JOB2	Bir yönetici işi için üstbilgi satırı 2.
ADMIN_JOB3	Bir yönetici işi için üstbilgi satırı 3.
FTE_YAPILANISI	Geçiş için var olan MFT yapılandırması. Geçiş gerekmiyorsa boş bir değere ayarlayın.
CREDENTIAL_YOLU	Geçiş için kimlik bilgileri dosyasının yolu; örneğin, /u/user1/agent3. Yalnızca geçiş komutları <b>BFGAGMG</b> ve <b>BFGLGMG</b> JCL komut dosyaları için gereklidir. Geçiş gerekmiyorsa boş bir değere ayarlayın. Ayrıca şunu da unutmayın:
DB_PROPS_YOLU	Geçiş için veritabanı kaydedici özellikleri dosyasını belirtir. Bu seçenek, özellikler dosyası şu varsayılan adı ve yolu kullanmıyorsa gereklidir: config_directory/coordination_qmgr/databaselogger.properties. Geçiş gerekmiyorsa boş bir değere ayarlayın.

Aşağıdaki çizelgede, MFT komut PDSE kitaplığı veri kümesindeki BFGCUSTM JCL komut dosyası için BFGSTDIN DD deyimi için zorunlu MFT yapılandırma özellikleri açıklanmaktadır. BFGCUSTM işini göndermeden önce, belirtilen özellikleri yerine koyma değerleriyle (yani, iki artı işareti içine alınmış değerler, + +) boşluk olmayan uygun bir değerle değiştirmeniz gerekir. Bu özellikler, MFT yapılandırma özellikleri için geçersiz kılmaları tanımlar. Ortamınıza ilişkin araçları ya da kaydedicileri özelleştirmek için aracı ve kaydedici özellikleri ekleyebilirsiniz. Tüm özelliklerin bir listesi için bkz. [“Yapılandırma özellikleri dosyaları”](#) sayfa 731.

Çizelge 45. BFGSTDIN DD deyimi için zorunlu yapılandırma özellikleri	
Özellik	Değer
coordinationQMGr	Aracı ya da kaydedicinin ilişkilendirildiği yapılanışa ilişkin eşgüdüm kuyruğu yöneticisinin adı.
coordinationQMGrAnasistemi	İsteğe Bağlı. Eşgüdüm kuyruk yöneticisinin çalıştığı sistemin anasistem adı. Bu özelliğe ilişkin değeri boş bırakırsanız, bağ tanımlama kipi bağlantısı olduğu varsayılır.

Çizelge 45. BFGSTDIN DD deyimi için zorunlu yapılandırma özellikleri (devamı var)	
Özellik	Değer
coordinationQMgrKapısı	İsteğe Bağlı. Koordinasyon kuyruk yöneticisinin dinlediği kapı numarası. Bu parametre, coordinationQMgrHost özelliği için boş olmayan bir değer de belirttiğinizde kullanılır.
coordinationQMgrKanalı	İsteğe Bağlı. Koordinasyon kuyruk yöneticisine bağlanmak için kullanılacak kanal. Bu parametre, coordinationQMgrHost özelliği için boş olmayan bir değer de belirttiğinizde kullanılır.
connectionQMgr	Aracının ya da kaydedicinin ilişkilendirildiği yapılanışa ilişkin komut kuyruğu yöneticisinin adı.
connectionQMgrAnasistem	İsteğe Bağlı. Komut kuyruğu yöneticisinin çalıştığı sistemin anasistem adı. Bu özelliğe ilişkin değeri boş bırakırsanız, bağ tanımlama kipi bağlantısı olduğu varsayılır.
connectionQMgrKapısı	İsteğe Bağlı. Komut kuyruğu yöneticisinin dinlediği kapı numarası. Bu parametre yalnızca connectionQMgrAnasistem özelliği için boş olmayan bir değer belirttiğinizde kullanılır.
connectionQMgrKanalı	İsteğe Bağlı. Komut kuyruğu yöneticisine bağlanmak için kullanılacak kanal. Bu parametre yalnızca connectionQMgrAnasistem özelliği için boş olmayan bir değer belirttiğinizde kullanılır.

### ► z/OS z/OS aracı ve kaydedici komutu JCL komut dosyaları

Bir MFT komutu PDSE kitaplığı veri kümesinde bulunan JCL komutları kümesi.

Çizelge 46. MFT komutu PDSE kitaplığı veri kümesinde bulunan JCL komutları	
Üye	Tanımlama ya da fte komut satırı komutu
BFGKOPYASı	Bu kitaplığın bir kopyasını yaratma işi
BFGCUSTM	Bu kitaplığı aracı ya da kaydedici için özelleştirme işi
BFGZCFRC	<a href="#">fteSetupKoordinasyonu</a>
BFGZCMCR	<a href="#">fteSetupKomutları: MFT command.properties dosyasını oluşturun</a>
BFGZAGCR	<a href="#">fteCreateAgent</a> . Yalnızca SERVICE_TYPE değişkenini AGENT olarak ayarladığınızda oluşturulur.
BFGLGCRS	<a href="#">fteCreateKaydedici</a> . Yalnızca SERVICE_TYPE değişkenini LOGGER olarak ayarladığınızda oluşturulur.
BFGZAGST	<a href="#">fteStartAgent</a> . Yalnızca SERVICE_TYPE değişkenini AGENT olarak ayarladığınızda oluşturulur.
BFGAGSTP	<b>fteStartAgent</b> yordamı. Yalnızca SERVICE_TYPE değişkenini AGENT olarak ayarladığınızda oluşturulur.



Çizelge 46. MFT komutu PDSE kitaplığı veri kümesinde bulunan JCL komutları (devamı var)

Üye	Tanımlama ya da fte komut satırı komutu
BFGZAGPI	<u>ftePingAgent</u> (Aracı). Yalnızca SERVICE_TYPE değişkenini AGENT olarak ayarladığınızda oluşturulur.
BFGZAGSP	<u>fteStopAgent</u> . Yalnızca SERVICE_TYPE değişkenini AGENT olarak ayarladığınızda oluşturulur.
BFGZLGST	<u>fteStartKaydedici</u> . Yalnızca SERVICE_TYPE değişkenini LOGGER olarak ayarladığınızda oluşturulur.
BFGLGSTP	<b>fteStartLogger</b> yordamı. Yalnızca SERVICE_TYPE değişkenini LOGGER olarak ayarladığınızda oluşturulur.
BFGZLGP	<u>fteStopKaydedici</u> . Yalnızca SERVICE_TYPE değişkenini LOGGER olarak ayarladığınızda oluşturulur.
BFGZAGSH	<u>fteShowAgentDetails</u> . Yalnızca SERVICE_TYPE değişkenini AGENT olarak ayarladığınızda oluşturulur.
BFGZLGSH	<u>fteShowLoggerDetails</u> . Yalnızca SERVICE_TYPE değişkenini LOGGER olarak ayarladığınızda oluşturulur.
BFGZCFDF	<u>fteChangeDefaultConfigurationSeçenekler</u>
BFGZAGCL	<u>fteCleanAgent</u> . Yalnızca SERVICE_TYPE değişkenini AGENT olarak ayarladığınızda oluşturulur.
BFGZAGDE	<u>fteDeleteAgent</u> . Yalnızca SERVICE_TYPE değişkenini AGENT olarak ayarladığınızda oluşturulur.
BFGZLGDE	<u>fteDeleteKaydedici</u> . Yalnızca SERVICE_TYPE değişkenini LOGGER olarak ayarladığınızda oluşturulur.
BFGZPRSH	<u>fteDisplaySürüm</u>
BFGZAGLI	<u>fteListAracıları</u> . Yalnızca SERVICE_TYPE değişkenini AGENT olarak ayarladığınızda oluşturulur.
BFGZMNLİ	<u>fteListİzleme Programları</u>
BFGZSTLI	<u>fteListScheduledTransfers</u>
BFGZTMLI	<u>fteListŞablonları</u>
BFGXCROB	<b>fteObfuscate</b> örnek
BFGZRAS	<b>fteRAS</b>
BFGZAGTC	<u>fteSetAgentTraceLevel</u> (Aracı İzleme Düzeyi). Yalnızca SERVICE_TYPE değişkenini AGENT olarak ayarladığınızda oluşturulur.

Çizelge 46. MFT komutu PDSE kitaplığı veri kümesinde bulunan JCL komutları (devamı var)	
Üye	Tanımlama ya da fte komut satırı komutu
BFGZLGTC	fteSetLoggerTraceLevel(Düzey). Yalnızca SERVICE_TYPE değişkenini LOGGER olarak ayarladığınızda oluşturulur.
BFGXPRAN	<b>fteAnt</b> örnek
BFGXTRCA	<b>fteCancelTransfer</b> örnek
BFGXMNCR	<b>fteCreateMonitor</b> örnek
BFGXTMCR	<b>fteCreateTemplate</b> örnek
BFGXTRCR	<b>fteCreateTransfer</b> örnek
BFGXMNDE	<b>fteDeleteMonitor</b> örnek
BFGXSTDE	<b>fteDeleteScheduledTransfer</b> örnek
BFGXTMDE	<b>fteDeleteTemplate</b> örnek

#### Notlar:

- JCL, MQSC yaratan ya da silme komut dosyalarına başvuran komutlar için, sizden bir komut dosyası çalıştırmanızı ister, ancak komut dosyası zaten iş tarafından çalıştırıldı.
- BFGZRAS, BGCUSTOM işi çalıştırıldığında BFGRAS üyesini oluşturur.

#### z/OS Doğrulama aktarımı gerçekleştiriliyor

Ürünün doğru çalışıp çalışmadığını denetlemek için bir aktarımı nasıl gerçekleştireceğinizi.

BFGTRCRS üyesini yeniden adlandırın ve düzenleyin.

1. %BFGCMD CMD=fteCreateTransfer -h ürününden önce bir /\* ekleyin
2. Üyedeki diğer yorumları kaldırın.
3. -sa ve -da için geçerli aracı adını belirtin
4. JCL ' yi Sakla
5. JCL ' yi Gönder

Bu JCL, komut kuyruğu yöneticisine bağlanır.

#### z/OS Günlüğe kaydetme görevinin yapılandırılması

Günlük kaydı görevinin koordinasyon kuyruk yöneticisiyle aynı görüntüde çalıştırılması gerekir. Db2' da oturum açabilirsiniz.

#### Günlüğe kaydetme görevi yaratılması

Günlüğe kaydediciyi özel PDSE yapmak için PDSE ' yi kopyalayın. Örneğin, user . MFT . LOGGER.

Farklı bir kimlik bilgileri dosyası kullanmanız gerekirse, bir kimlik bilgileri dosyası oluşturun. Bkz. [z/OS üzerinde MQMFTCredentials.xml ' in yapılandırılması](#).

[BFGCUSTOM](#) üyesini gözden geçirin. İçeriğin çoğunun önceki uyarlamadan aynı kalacağını unutmayın.

Ancak, aşağıdakileri yapmanız gerekir:

- // SYSEXEC DD DSN=SCEN.FTE.JCL....
- LIBRARY değerini aracı PDSE ile eşleşecek şekilde değiştir
- QMGR ' yi koordinasyon kuyruğu yöneticisinin adına çevirin

- SERVICE\_TYPE=LOGGER yap
- NAME değerini kaydedicinin adı olacak şekilde değiştirin (PDSE ile eşleşiyor)
- JOBCARD ' yi inceleyin ve iş adını, adın araçların iş adlarından farklı olacak şekilde değiştirin.
- BFG\_JVM\_PROPERTIES = "-Xmx1024M"

Db2 kaydedicisini kullanıyorsanız, Db2 sorunlarını belirlemenize yardımcı olacak Db2 izlemelerini yakalayabilmeniz için bir dosya oluşturmanız yararlı olur.

Dosyanın adı, JDBC izleme özellikleri dosyasının içeriği olduğu JVM özelliklerinde belirtilir:

```
db2.jcc.traceDirectory=/u/johndoe/fte
db2.jcc.traceFile=jccTrace1
db2.jcc.traceFileAppend=false
# turn on all traces
# db2.jcc.traceLevel=-1
# turn off all traces
db2.jcc.traceLevel=0
```

İki JVM özelliği kurdu-,,

```
BFG_JVM_PROPERTIES=-Ddb2.jcc.propertiesFile=/u/.../sql.properties
-Ddb2.jcc.ssid=DBCA
```

Burada /u/.../sql.properties , Db2 izleme özellikleri dosyanızın adı ve *DBCA* , Db2 altsisteminizin adıdır.

İşin veri kümesine özel erişim gerektirdiğini belirterek bu işi sunun. Aracıya ilişkin işlerin hepsinin *BFGLG\** gibi adları vardır.

## Dosyalara kaydetme

Db2' da günlüğe kaydetme hakkında daha fazla bilgi için bkz. [“Db2 ' da oturum açarken bir günlük kaydı görevi oluşturma” sayfa 728](#)

Üye *BFGLGCRS ' yi* yeniden adlandırın. Bu iş, Managed File Transfer (MFT) dizinindeki dosyaları günceller ve yerel kuyruk yöneticisinde aracıya özgü kuyruklar yaratmak için *CSQUTIL ' i* kullanır.

Özgün dosyada, komutun sözdizimini listeleyen *%BFGCMD CMD=fteCreateLogger -h* komutu vardır.

Günlüğe kaydedici görevi yaratmak için, *%BFGCMD CMD=fteCreateLogger -h* deyimini deyimimin önüne */\** koyarak açıklama satırı yapın ve birinci kolonun boş olduğundan emin olun.

İkinci komuttan açıklamaları kaldırın ve deyimleri yapılandırın. Örneğin:

```
%BFGCMD CMD=fteCreateLogger +
-p MQPH +
-loggerMgr MQPH +
-loggerType FILE +
-fileLoggerMode circular +
-fileSize 5MB +
-fileCount 5 +
-p MQPH +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
LOGGER
```

Başarıyla işlendiğini görmek için çıkışı denetleyin.

**İpucu:** *logger.properties* dosyasının yol adını işin çıkışından aracının PDSE ' deki bir üyeye kopyalayın.

Örneğin, *APATH* üyesine kopyalama

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties
```

Özellikler dosyasını görüntülemeniz gerekiyorsa bu kullanışlıdır.

Dizini bu dosyaya ekle:

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/
```

Dosyaya kaydediyorsanız, günlük dosyaları bu dizinde saklanır; örneğin, LOGGER0-20140522123654897.log.

İzleme dosyaları günlük alt dizininde bulunur; örneğin,

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

Artık günlüğe kaydetme görevini başlatabilirsiniz.

## Db2 ' da oturum açarken bir günlük kaydı görevi oluşturma

Üye BFGLGCRS ' yi yeniden adlandırın.

Bu iş, MFT dizinindeki dosyaları günceller ve yerel kuyruk yöneticisinde aracıya özgü kuyruklar yaratmak için CSQUTIL ' i kullanır.

Bilmeniz gereken:

Çizelge 47. Db2 değişkenler	
Db2 ad	Örnek
-dbName databaseName	Bunu Db2 altsistemimize ilişkin DSNL004I iletisindeki yer değerinden alabilirsiniz.
-dbDriver filePath	Örnek: /db2/db2v10/jdbc/classes/db2jcc.jar
-dbLib filePath	Örnek: /db2/db2v10/jdbc/lib/libdb2jcc2zos_64.so

Dosyayı düzenleyin. Özgün dosyada, komutun sözdizimini listeleyen %BFGCMD CMD=fteCreateLogger -h komutu vardır.

İkinci komuttan açıklamaları kaldırın ve deyimleri yapılandırın. Örnek:

```
%BFGCMD CMD=fteCreateLogger +
-p MQPH +
-loggerQMgr MQPH +
-loggerType DATABASE +
-dbType DB2 +
-databaseName DSNDBCP +
-dbDriver /db2/db2v10/jdbc/classes/db2jcc.jar +
-dbLib /db2/db2v10/jdbc/lib/ +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
LOGGER
```

Günlüğe kaydedici görevi yaratmak için, %BFGCMD CMD=fteCreateLogger -h deyimini deyimimin önüne /\* koyarak açıklama satırı yapın ve birinci kolonun boş olduğundan emin olun.

İşi sunun ve başarıyla işlendiğini görmek için çıkışı denetleyin.

**İpucu:** logger.properties dosyasının yol adını işin çıkışından araçların PDSE ' deki bir üyeye kopyalayın.

Örneğin, APATHüyesine kopyala ama ama:

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties into member USS
```

Özellikler dosyasını görüntülemeniz gerekiyorsa bu kullanışlıdır.

İzleme dosyaları günlük altdizininde bulunur; örneğin:

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

## Db2 çizelgeleri yaratılması

Db2 çizelgelerini yaratmanız gerekir. Tanımlar z/OS UNIX System Services kütütarafından `gfi` ile gerekir `gfi` ile gerekir şlarından. `mqft/sql/fteolog_tables_zos.sql`.

PDSE ' de Db2 adlı bir üye oluşturun. Bu üyeyi düzenleyin ve komut satırında COPY komutunu kullanın. z/OS UNIX System Services tanımlama dosyasından kopyalayın.

Yere özgü gereksinimler büyük ölçüde değişebildiği için, bu dosya yalnızca çizelgelerin temel yapılarını ve bunların bulunacağı çizelge alanını belirtir.

Çizelge alanı, SQL komut kütüğü tarafından, sayfa büyüklüğü olabilecek en büyük çizelge satırlarını tutmaya yetecek olan bir arabellek havuzu kullanılarak yaratılmasını sağlamak için belirtilir. LOB konumları gibi özniteliklerin belirtilmediğini unutmayın.

Veritabanı denetimciniz, performansla ilgili bu öznitelikleri tanımlamak için bu dosyanın bir kopyasını değiştirmek isteyebilir.

Bu dosya ayrıca varsayılan şema adı FTELOG, varsayılan tablo alanı adı FTELOGTSve veritabanı adı FTELOGDBolarak da varsayılır. Dosyanın başındaki açıklamalarda açıklanan işlemi izleyerek, varolan bir veritabanı ve yerel adlandırma kurallarıyla eşleşmesi için gerekirse o ünüzü landırmak üzere bu adları değiştirebilirsiniz.

**Önemli:** Dosyada açıklamalar olduğundan ve **DSNTINAD** gibi toplu iş programları yorum kabul etmediğinden, komutları çalıştırmak için **SPUFI** gibi çevrimiçi olanakları kullanın.

Ek bilgi için [SPUFI](#) kullanarak SQL ' i yürütme başlıklı konuya bakın. Buna ek olarak, SCSQPROC içindeki CSQ45STB ' de, Db2 SELECT komutlarını yapmak için uyarlayabileceğiniz örnek JCL vardır.

## Günlüğe kaydedici görevini başlatma

Üyeyi yeniden adlandırın, gözden geçirin ve gönderin BFGDB0023İletisini almalısınız: Kaydedici başlatma etkinliklerini tamamladı ve şu anda çalışıyor.

## Günlüğe kaydedici işlemleri

Kaydedici durumunu görüntülemek için, üyeyi yeniden adlandır, gözden geçir ve gönder BFGLGSH

Kaydediciyi durdurmak için, üyeyi yeniden adlandırmak, gözden geçirmek ve göndermek için BFGLGSP.

## z/OS üzerinde MFT için ortam değişkenleri

Komutları doğrudan z/OS UNIX System Services (z/OS UNIX) ortamından ya da kendi JCL komut dosyalarınızdan çalıştırıyorsanız, özelleştirme ve yapılandırmadan sonra, Managed File Transfertarafından sağlanan yapılandırma ve yönetim komut dosyalarını çalıştırmadan önce bir dizi ortam değişkeni ayarlamanız gerekir. Bu değişkenleri her kullanıcı için ve komut dosyalarının çağrılacağı her ortamda ayarlamanız gerekir.

Diğer ürünlerle çakışmayı önlemek için ana dizininizde bir `.wmqftec` komut dosyası oluşturmayı seçebilirsiniz. Daha sonra `.wmqftec` komut dosyası, Managed File Transfer komut dosyalarının her biri tarafından çağrılır ve Managed File Transferiçin özel ortam ayarları sağlamak üzere bu komut dosyasını kullanabilirsiniz.

JCL ' den araçlar çalıştırılırken işletmen günlüğüne ileti göndermek üzere ayarlayabileceğiniz isteğe bağlı bir BFG\_WTO ortam değişkeni de vardır.

Çizelge 48. Gereklı z/OS ortam deęiřkenleri

Ortam deęiřkeni	Deęer
BFG_JAVA_HOME	Java kuruluřununun konumu. Desteklenen Java dūzeyleri hakkında daha fazla bilgi iin bkz. <a href="#">IBM MQiin Sistem Gereksinimleri</a> .
BFG_VERİLERİ	Managed File Transfer for z/OSiin veri dizininin konumu. Bu, <i>DATA_DIR</i> yolu.
STEPLIB	Ařaęıdaki IBM MQ veri kūmelerini iermelidir: <ul style="list-style-type: none"><li>• SCSQAUTH</li><li>• SCSQANLE</li><li>• SCSQLOAD</li></ul> Veritabanı kaydedici bileřeni bir z/OS sisteminde alıřtırmak istiyorsanız, STEPLIB ařaęıdaki Db2 veri kūmelerini de gōsterilen sırada iermelidir: <ul style="list-style-type: none"><li>• SDSNEXIT</li><li>• SDSNLOAD2</li><li>• SDSNLOAD (YŪKLE)</li></ul>

Ařaęıda, Managed File Transferiin ortam deęiřkenlerini doęru řekilde yapılandırılan bir `.profile` orneęi verilmiřtir:

```
STEPLIB=MQM.V920.SCSQAUTH:MQM.V920.SCSQANLE:MQM.V920.SCSQLOAD
PATH=/u/fteuser/bin:/u/fteuser/J7.0/bin:/bin:/usr/bin:/u/fteuser/extras/bin:/bin:$PATH
BFG_JAVA_HOME=/u/fteuser/J7.0
BFG_DATA=/u/fteuser/DATA_DIR
export PATH STEPLIB BFG_JAVA_HOME BFG_DATA
```



**Uyarı:** z/OS UNIX ortamından **fte\*** komutları aęrılırken LIBPATH ortam deęiřkeni artık gerekli deęildir ve var olan herhangi bir `.wmqfterc` komut dosyasından kaldırılmalıdır

İsteęe baęlı olarak, ařaęıdaki ortam deęiřkenlerini de ayarlayabilirsiniz:

Çizelge 49. İsteğe bağlı z/OS ortam değişkeni

Ortam değişkeni	Değer
BFG_WTO	<p>Aşağıdaki değerlerden biri BFG_WTO 'yu etkinleştirecek:</p> <ul style="list-style-type: none"><li>• EVET</li><li>• AÇIK</li><li>• DOĞRU</li></ul> <p>Aşağıdaki değerlerden biri BFG_WTO 'yu devre dışı bırakacak. Bu değerler büyük ve küçük harfe duyarlı değildir.</p> <ul style="list-style-type: none"><li>• BOŞ DEĞERLİ</li><li>• HAYIR</li><li>• KAPALI</li><li>• YANLIŞ</li></ul> <p>z/OS günlük kaydını etkinleştirir. Varsayılan olarak bu ortam değişkeni devre dışıdır.</p> <p>Aracı olay günlüğüne yazılan iletiler, JCL 'den bir aracı çalıştırdığınızda otomasyon ürünleri için daha kolay erişim sağlayan z/OS işletmen günlüğü olanağına da yazılır. Yönelme kodu, Programcı Bilgileri (11) ve tanımlayıcı kodu Bilgi 'dir (12).</p>
BFG_GRUPLARI_ADı	<p>mqm dosya grubu genellikle Managed File Transfer yapılandırma verileri dosyaları ve komutlarıyla ilişkilendirilir. Sonuç olarak, mqm grubunun üyesi olan tüm kullanıcılar Managed File Transfer yapılandırmasına erişebilir ve bu yapılandırmada değişiklik yapabilir. Daha fazla bilgi için, bkz. <a href="#">IBM MQ içinde MFT için dosya sistemi izinleri</a>.</p> <p>Bir z/OS sistemi için, dosya grubu bir z/OS UNIX dosya sistemi varlığı ve mqm dosya grubunun tanımlanmış olması gerekmez. BFG_GROUP_NAME ortam değişkenini kullanarak Managed File Transfer yapılandırma verileri dosyaları için alternatif, var olan bir z/OS UNIX dosya sistemi grubu tanımlayabilirsiniz. Örneğin, z/OS UNIX kabuk komut isteminde:</p> <pre>export BFG_GROUP_NAME=FTEGB</pre> <p>geçerli z/OS UNIX oturumu için sonradan oluşturulan yapılandırma dosyalarıyla ilişkilendirilecek grup FTEGB 'yi tanımlar.</p> <p>BFG_GROUP_NAME değerini boş bir değere ayarlayabilir ya da kaldırabilirsiniz.</p>

## ► z/OS Yapılandırma özellikleri dosyaları

Managed File Transferinde kullanılan özelliklerin özeti.

- [MFT coordination.properties dosyası](#)
- [MFT command.properties dosyası](#)
- [MFT agent.properties dosyası](#)
- [Günlük kaydedici yapılandırma özellikleri dosyası](#)

## z/OS MFT ' in z/OS Automatic Restart Manager (ARM) için yapılandırılması

Managed File Transfer , ARM etkin bir uygulamadır.

### Başlamadan önce

ARM ' yi etkinleştirme ve sisteminize ilişkin ARM ilkelerini tanımlama hakkında daha fazla bilgi için [z/OS Automatic Restart Manager \(ARM\)](#) başlıklı konuya bakın.

Bir Db2 veritabanını otomatik olarak yeniden başlatmak ve yeniden bağlanmak için MFT DB Logger yeteneğini kullanmak istiyorsanız, desteklenen tek yeniden başlatma yöneticisi ARM 'dir.

### Bu görev hakkında

ARM kullanılarak, aracı/kaydedici özellikleri armELEMTYPE ve armELEMENT ayarlanarak, araçlar ve kaydediciler yeniden başlatma için yapılandırılabilir. armELEMTYPE özelliği, ARM ögesinin tipini tanımlar ve armELEMENT özelliği, ARM ' nin kaydettiği öğenin adıdır:

- Aracı ELEMTYPE değerini SYSBFGAG olarak ayarlayabilir ve armELEMENT aracı adına karşılık gelecek şekilde ayarlanabilir.
- Kaydedici ELEMTYPE değerini SYSBFLG olarak ayarlayabilir ve armELEMENT , kaydedici adına karşılık gelecek şekilde ayarlanabilir.

**Not:** ARM tarafından yeniden başlatma için yapılandırılan araçlar ve kaydediciler yalnızca toplu işten ya da başlatılan bir görevden başarıyla çalıştırılabilir. Aracı ya da kaydediciyi doğrudan z/OS UNIX System Services komut satırından başlatma girişimleri ARM hata neden koduyla başarısız olur.

### Örnek

Aşağıdaki yeniden başlatma ilkesi örneği, BFGFT7CAG1 aracısını FT7C:

```
RESTART_ORDER
  LEVEL (3)
  ELEMENT_TYPE (SYSBFGAG, SYSBFLG)

RESTART_GROUP (GROUP7C)
  ELEMENT (SYSQMGRFT7C)
  ELEMENT (BFGFT7CAG1)
  RESTART_ATTEMPTS (3, 300)
```

## z/OS Örnek: z/OS üzerinde Managed File Transfer araçları için JCL oluşturma

IBM MQ for z/OS üzerinde aracı oluşturmak ve başlatmak için kullanılacak bazı JCL ' ler oluşturmak üzere bu bilgileri kullanın.

### Örnek kitaplığı kopyala

Aşağıdaki yordamı gerçekleştirin:

1. Kitaplığı açarak SCSQFCMD kitaplığının bir kopyasını alın (bkz. [“JCL kitaplığı yaratmak için SCSQFCMD ' yi kopyala”](#) sayfa 716).



Üyelerin çoğunluğu (BFGX, BFGY ya da BFGZ ile başlayanlar), daha sonra aracı için özelleştirilmiş JCL oluşturmak üzere kullandığınız şablonlardır.

Önemli üye BFGCOPY.

2. BFGCOPY dosyasını açın ve değiştirin:

**++ supplied\_library ++**

Ürünün bir parçası olarak kurulan SCSQFCMD kitaplığının adını girin.

**++ hizmet kitaplığı ++**

Aracınız (hedef kitaplık) için kullanmak istediğiniz kitaplığın adını belirtin.

3. İşi gönderin ve kullanabileceğiniz yeni bir kitaplığınız olsun.

## BFGCUSTM ' yi Düzenle

Aşağıdaki yordamı gerçekleştirin:

1. BFGCUSTM üyesini düzenleyebilmeniz için yeni kitaplığı açın (bkz. [“Üye BFGCUSTM düzenleniyor” sayfa 716](#))

2. ++ karakterleri içine alınmış üyedeki tüm parametreleri değiştirin ve bunları uygun değerlerle değiştirin. Örneğin, değişiklik:

**++ mq\_path ++**

z/OS UNIX System Services (z/OS UNIX) Components dizininin yolu. Örneğin, /mqm/V9R2M0.

**Not:** Bu değişkenin değiştirilecek üç eşgörünümü vardır.

**++ bfg\_data ++**

IBM MQ Managed File Transfer for z/OS yapılandırmanızın depolanacağı z/OS UNIX dizini olmak için.

**++ hizmet\_tipi ++**

AGENT sözcüğüne

**++ aracı\_adi ++**

Temsilcinizin adı olmak için

### Notlar:

1. CLEAN\_AGENT\_PROPS için gerekli olan ++options++ gibi bazı girişlere gerek yoktur, bu nedenle bunları kaldırmanız gerekir.
2. BFGCUSTM üyesindeki tüm parametrelerin tam listesi ve bunların hangi değerlere sahip olması gerektiğine ilişkin bir açıklama için bkz. [“MFT for z/OS ürününü yapılandırmaya başlamadan önce” sayfa 711](#).

## BFGCUSTM JCL ' yi Gönder

Aşağıdaki yordamı gerçekleştirin:

1. İşi gönderin.

2. ISPFiçindeki kitaplıktan çıkın.

BFGCUSTM işi kitaplığı güncellediğinden ve kitaplık açıkken bunu yapamadığından bu gereklidir.

3. İş tamamlandığında iş günlüğüne bakın.

Kitaplık içinde yeni üyelerin yaratıldığını gösteren bir dizi ileti görürsünüz.

Bu üyelerin her biri, aracınıza ilişkin belirli görevleri gerçekleştirmek için kullanılacak JCL ' yi içerir.

Bu üyelerin ve bunların karşılık geldikleri IBM MQ Managed File Transfer komutlarının bir listesi için bkz. [“z/OS aracı ve kaydedici komutu JCL komut dosyaları” sayfa 724](#).

## Aracıyı oluşturmak için BFGAGCR gönder

Yeni üye BFGAGCR, **fteCreateAgent** komutunu çağırarak [bir aracı oluşturan JCL ' yi](#) içerir.

Aşağıdaki yordamı gerçekleştirin:

1. BFGAGCR üyesini açın.

BFGAGCR ' nin şu adla doldurulduğunu görmelisiniz:

- Aracı
- Aracı kuyruk yöneticisi
- MFT topolojisi için koordinasyon kuyruk yöneticisi

2. Üye BFGAGCR gönder.

Üye çalıştığında:

- Aracınız için gerekli yapılandırma dosyalarını oluşturur.
- Aracı kuyruk yöneticisine bağlanır ve CSQUTIL kullanarak aracının gereksinim duyduğu sistem kuyruklarını yaratır.
- Aracıyı eşgüdüm kuyruk yöneticisine kaydeder.

## BFGAGST göndererek aracıyı başlat

Aşağıdaki yordamı gerçekleştirin:

1. BFGAGST üyesini gönderin. Aracının doğru çalıştığını gösteren çeşitli komutlar için [aracıyı kullanma](#) başlıklı konuya bakın.
2. İş tamamlandığında, iş günlüğünde aşağıdaki iletileri olup olmadığını denetleyin:

```
BFGAG0058I: The agent has successfully initialized.  
BFGAG0059I: The agent has been successfully started.
```

Bu, aracınızın çalışır durumda ve yönetilen aktarımları gerçekleştirmeye hazır olduğu anlamına gelir.

## MFT aracısını yeni bir z/OS LPAR ' na taşıma

Bir IBM MQ Managed File Transfer for z/OS aracısını aynı koordinasyon ve komut kuyruğu yöneticileriyle aynı IBM MQ Managed File Transfer topolojisinde tutarken bazen bir LPAR ' dan diğerine taşımak gerekir. Bunu yapmak için gereken adımlar, aracının ilk olarak nasıl oluşturulduğuna bağlıdır.

### Bu görev hakkında

IBM MQ Managed File Transfer for z/OS aracısını aşağıdaki yollardan biriyle taşıyın:

- Aracı, SCSQFCMD kitaplığının özelleştirilmiş bir sürümü kullanılarak yaratıldıysa, yeni bir LPAR ' da yeniden yaratmak için kitaplığı kullanın.
- Aracı başlangıçta z/OS UNIX System Services (z/OS UNIX) komutları çalıştırılarak oluşturulduysa, yeni bir LPAR ' da yeniden oluşturmak için komutları kullanın.

#### Not:

Zamanlanmış aktarımlar ve aktarma şablonları, bir IBM MQ Managed File Transfer topolojisi için koordinasyon kuyruğu yöneticisinde saklanır. Bu görev, eşgüdüm kuyruk yöneticisinin hareket işinin bir parçası olmadığını varsayar. Bu durumda, taşınmakta olan görevle ilişkili zamanlanmış aktarımlar ve aktarma şablonları, taşıma tamamlandıktan sonra var olan koordinasyon kuyruğu yöneticisinde kalır.

### Yordam

- SCSQFCMD kitaplığının uyarlanmış bir sürümü kullanılarak yaratılan bir aracıyı taşımanızı sağlar.

Aracı SCSQFCMD kitaplığının özelleştirilmiş bir sürümü kullanılarak yaratıldıysa, IBM MQ Managed File Transfer for z/OS ortamını ve yeni LPAR ' daki aracı yapılandırmasını yeniden yaratmak için bu kitaplığı kullanabilirsiniz. Bunu yapmak için aşağıdaki adımları tamamlayın:

1. Kitaplığın özelleştirilmiş sürümünü özgün LPAR 'dan yeni LPAR' a kopyalayın.
2. Yeni LPAR 'daki kitaplığın özelleştirilmiş sürümünde BFGCUSTM üyesini düzenleyin ve parametre değerlerinin hala geçerli olduğundan emin olun.
3. Ortamı yapılandırmak ve aracıyı oluşturmak için gereken tüm JCL 'yi oluşturmak üzere yeni LPAR' da BFGCUSTM üyesini çalıştırın.
4. Yeni LPAR 'da aracı tarafından kullanılacak koordinasyon kuyruğu yöneticisini tanımlamak için BFGCFR üyesini çalıştırın ve IBM MQ Managed File Transfer yapılandırmasını depolamak için gereken izin yapısını oluşturun.
5. Daha sonra, yeni LPAR üzerinde aracı tarafından kullanılacak komut kuyruğu yöneticisini tanımlamak için BFGCMCR üyesini çalıştırın.
6. Aracıyı ve yapılandırmasını yeniden oluşturmak için BFGAGCR üyesini çalıştırın.
7. Aracı tarafından kullanılan sistem kuyruklarının, o aracı için kuyruk yöneticisinde bulunduğundan emin olun.

Taşınmakta olan aracıyla ilişkilendirilmiş kaynak izleyicileri varsa, yeni aracıda izleyicileri yeniden yaratmanız gerekir. Bunu yapmak için aşağıdaki adımları tamamlayın:

1. Özgün LPAR 'da, özgün aracıyla ilişkili kaynak izleme programına ilişkin tanımlamaları XML dosyalarına aktarmak için BFGMNLİ üyesini çalıştırın.
  2. Kaynak izleme programı tanımlarını içeren XML dosyalarını yeni LPAR 'a kopyalayın.
  3. XML dosyalarında saklanan kaynak izleyicisi tanımlamalarını içe aktarmak için, yeni LPAR 'ın SCSQFCMD kitaplığındaki BFGMNCRS üyesini kullanın. Bu, yeni aracıda izleme programlarının oluşturulmasına neden olur.
- z/OS UNIX içinde komutları çalıştırarak oluşturulan bir aracıyı taşıyın.

Aracı başlangıçta z/OS UNIX komutları çalıştırılarak oluşturulduysa, yeni bir LPAR 'da aracıyı yeniden oluşturmak için komutları kullanabilirsiniz. Bunu yapmak için aşağıdaki adımları tamamlayın:

1. Yeni LPAR 'da fteSetupKoordinasyon komutunu çalıştırın, aracı tarafından kullanılacak koordinasyon kuyruğu yöneticisini tanımlayın ve IBM MQ Managed File Transfer yapılandırmasını depolamak için gerekli izin yapısını oluşturun.
2. Yeni LPAR 'da aracı tarafından kullanılacak komut kuyruğu yöneticisini tanımlamak için fteSetupCommands komutunu çalıştırın.
3. Aracıyı ve yapılandırmasını yeniden oluşturmak için fteCreateAgent komutunu çalıştırın.
4. Aracı tarafından kullanılan sistem kuyruklarının, o aracı için kuyruk yöneticisinde bulunduğundan emin olun.

Taşınmakta olan aracıyla ilişkilendirilmiş kaynak izleyicileri varsa, yeni aracıda izleyicileri yeniden yaratmanız gerekir. Bunu yapmak için aşağıdaki adımları tamamlayın:

1. Özgün LPAR 'da, özgün aracıyla ilişkili kaynak izleyici tanımlarını XML dosyalarına aktarmak için **-ox** parametresini belirterek fteListMonitors komutunu çalıştırın.
2. Kaynak izleme programı tanımlarını içeren XML dosyalarını yeni LPAR 'a kopyalayın.
3. XML dosyalarında saklanan kaynak izleyicisi tanımlarını içe aktarmak için yeni LPAR 'da **-ix** parametresini belirterek fteCreateMonitor komutunu çalıştırın. Bu, yeni aracıda izleme programlarının oluşturulmasına neden olur.

## **V 9.3.0 z/OS MFT altyapısının IBM MQ for z/OS kuyruk paylaşım gruplarıyla planlanması**

IBM MQ Managed File Transfer (MFT) kullanıyorsanız, bir ya da daha çok aracı, komut ya da eşgüdüm kuyruğu yöneticisi IBM MQ for z/OS kuyruk paylaşım grubunun bir parçası olduğunda aşağıdakileri göz önünde bulundurmanız gerekir.

Aracılara, komut kuyruğu yöneticilerine ve koordinasyon kuyruğu yöneticilerine ilişkin açıklamalar için MFT topolojisine genel bakış başlıklı konuya bakın.

## Aracı kuyruk yöneticileri

Normalde bir MFT aracı tek bir aracı kuyruk yöneticisine bağlanır ve yalnızca bu kuyruk yöneticisi tarafından erişilebilen yerel kuyrukları kullanır. Aracı ilk yaratıldığında kuyruk yöneticisi adıyla birlikte sağlanarak, aracıya hangi kuyruk yöneticisine bağlanacağı bildirilir.

IBM MQ for z/OS ile, aracı yaratılabilir ve kuyruk yöneticisi adını bir kuyruk paylaşım grubu (QSG) adıyla değiştirebilirsiniz. Bu, aracının dosya aktarımlarını gerçekleştirmek için QSG 'deki herhangi bir kullanılabilir kuyruk yöneticisine bağlanabileceği anlamına gelir. Aracının bağlı olduğu kuyruk yöneticisinde bir hata oluşması durumunda, aracı hatayı algılar ve QSG 'deki diğer bir kuyruk yöneticisine yeniden bağlanır.

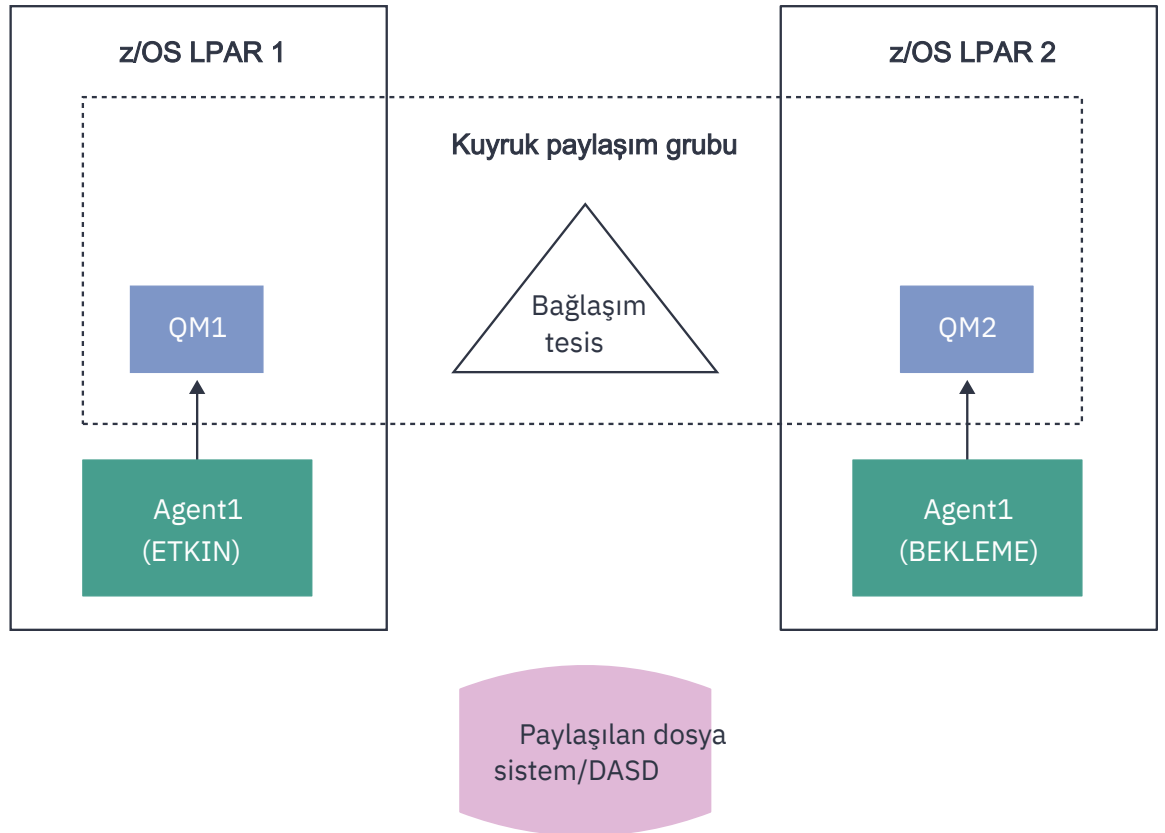
Bir aracıyı IBM MQ 9.2.0 tarafından sağlanan yüksek kullanılabilirlikli aracı desteğiyle birlikte QSG 'ye bağlamak, bkz. "Managed File Transfer içinde yüksek kullanılabilirlikli araçlar" sayfa 757, çok güçlü MFT topolojilerinin oluşturulmasına izin verir.

Örneğin, aşağıdaki şekilde *Agent1* aracı kuyruk yöneticisinin iki kuyruk yöneticisinden *QM1* ve *QM2* oluşan bir QSG olması için oluşturulmuştur. Aracı kuyrukları, bağlaşım olanağında saklanan paylaşılan kuyruklar olarak tanımlanır.

Bu, aracının *LPAR 1* ya da *LPAR 2* üzerinde çalışabileceği ve *QM1* ya da *QM2*'ye bağlanabileceği anlamına gelir. Aracının okuduğu ya da yazdığı dosyalar ve veri kümeleri paylaşılır; bu, bu dosyalara her iki LPAR 'dan da erişilebileceği anlamına gelir.

Buna ek olarak, aracı yüksek kullanılabilirlikli bir aracı olacak şekilde yapılandırıldı. Şemada aracı *LPAR 1* 'de etkin ve aracının bir beklemedeki eşgörünümlü *LPAR 2* 'de çalışıyor.

Bu topoloji yüksek esneklik sağlar. *LPAR 1* üzerinde çalışan aracı başarısız olursa ya da kuyruk yöneticisi *QM1* başarısız olursa ya da *LPAR 1* , *LPAR 2* üzerindeki aracının yedek eşgörünümlü başarısız olursa, dosya aktarımlarını hata noktasından devralabilir ve işlemeye devam edebilir.



Şekil 95. Bir kuyruk paylaşım grubu kullanan yüksek kullanılabilirlikli MFT aracı

## Aracı kuyruk yöneticisi olarak QSG kullanan bir aracı oluşturma

`fteCreateAgent` komutunu kullanarak bir aracı oluşturun. Bunu yaparken, aracı kuyruk yöneticisi için kuyruk paylaşım grubunun adı sağlanır. Örneğin:

```
fteCreateAgent -agentName Agent1 -agentQMGr QSG1
```

Bu, aracı kuyruk yöneticisi olarak QSG *QSG1* 'nin üyesi olan herhangi bir kuyruk yöneticisini kullanan *Agent1* adlı bir aracı oluşturur. Bu yapılandırma aracı, aracı ile kuyruk yöneticisinin aynı LPAR 'da olması gerektiği anlamına gelen bir çapraz bellek (bağ tanımlama kipi) bağlantısı kullanarak aracı kuyruk yöneticisine bağlanır. Bu, yukarıdaki şekil 1 'de gösterilen örneğe tam olarak benzer.

**fteCreateAgent** komutunu çalıştırdığınızda, aracı kuyruk yöneticisinde gerekli kuyrukları yaratmak için bir MQSC komutları kümesi oluşturur.

Aracı kuyruk yöneticisi bir QSG olduğunda, her bir kuyruğun paylaşılan bir kuyruk olarak yaratılması için bu komut kümesinin değiştirilmesi gerekir. Yani, her bir kuyruğun QSGDISP (SHARED) ile ve CFSTRUCT özneliği tarafından sağlanan uygun bir bağlaşım olanağı yapısıyla yaratılması gerekir.

Aşağıdaki örnekte, SYSTEM.FTE.COMMAND.AGENT1 kuyruğu. Varsayılanlarda yapılan değişiklikler koyu metindir.

**Önemli:** Aracının kullandığı diğer tüm kuyruklarda benzer değişiklikler yapmanız gerekir.

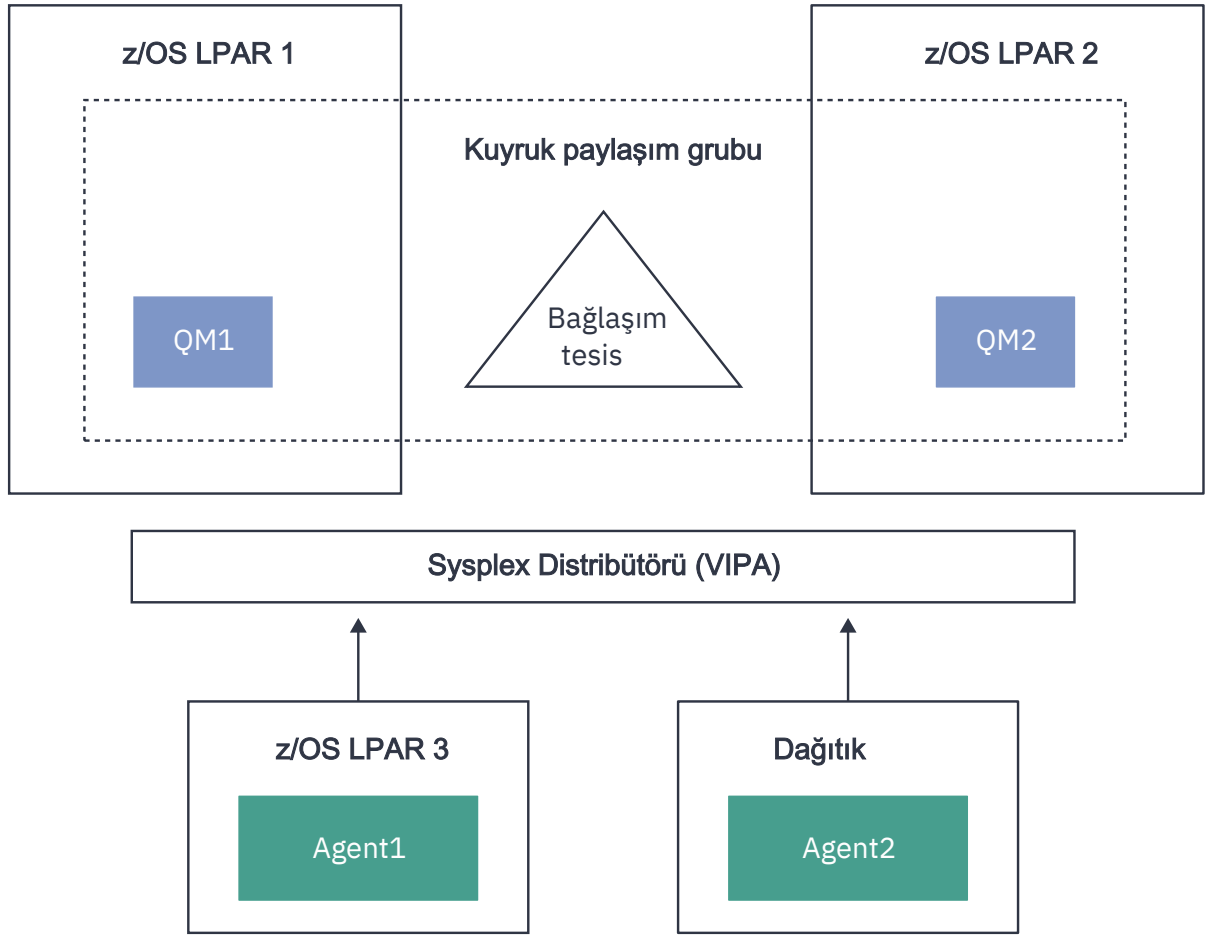
```
DEFINE QLOCAL(SYSTEM.FTE.COMMAND.AGENT1) +
  QSGDISP(SHARED) +
  CFSTRUCT(MFTSTRUCT) +
  DEFPRTY(0) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  INDXTYPE(CORRELID) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
  MSGDLVSQ(PRIORITY) +
  PUT(ENABLED) +
  RETINTVL(99999999) +
  SHARE +
  NOTRIGGER +
  USAGE(NORMAL) +
  REPLACE
```

## Aracı kuyruk yöneticisi olarak QSG kullanan ve istemci olarak bağlanan bir aracı oluşturma

Aracılar, bir istemci kanalı kullanarak Agent kuyruk yöneticisine bağlanabilir. Bu yaklaşımı, aracının bir QSG 'ye bağlanırken dağıtılmış platformlar üzerinde çalışmasına izin vermek için kullanabilirsiniz. QSG 'deki tüm kuyruk yöneticileri IBM MQ Advanced for z/OS Value Unit Edition için lisanslandıysa, aracı bunlara yerel kuyruk yöneticisi olmayan bir z/OS LPAR'ından da bağlanabilir.

Bu topoloji aşağıdaki şekilde gösterilir ve aracının QSGs esnekliğinden yararlanmasını sağlar. Aracının bağlı olduğu QSG 'deki kuyruk yöneticisi başarısız olursa, aracı otomatik olarak QSG 'nin farklı bir üyesine yeniden bağlanır ve işleme devam eder.

Sysplex distribütörü, aracından gelen bağlantıları QSG 'deki kullanılabilir kuyruk yöneticilerine dağıtmak için kullanılır.



Şekil 96. Bir kuyruk paylaşım grubuna istemci olarak bağlanan MFT araçları

Bu topolojiden yararlanmak için, QSG ' deki kuyruk yöneticilerinin her birinin aracı tarafından kullanılmak üzere tanımlanmış bir sunucu bağlantı kanalı olmalıdır. Bunun nasıl yapılacağını öğrenmek için bkz. [“İstemcinin kuyruk paylaşım grubuna bağlanması”](#) sayfa 61 .

Aracı oluşturulurken kuyruk yöneticilerinin QSG ' de tanımlanan kanalı kullanabilmeleri ve sysplex distribütörü aracılığıyla bu kanala erişebilmeleri için yapılandırılması gerekir. Örneğin:

```
fteCreateAgent -agentName Agent1 -agentQMgr QSG1 -agentQMgrHost vipaAddress
-agentQMgrPort sharedPort -agentQMgrChannel CHANNEL1
```

Daha önce belirtildiği gibi, **fteCreateAgent** komutu çalıştırılarak oluşturulan MQSC komutları, CFSTRUCT özniteliğinde QSGDISP (SHARED) ve uygun bir bağlaşım olanağı yapısı belirtilecek şekilde ayarlanmalıdır.

## Komut kuyruğu yöneticileri

MFT komut kuyruğu yöneticisi bir QSG ' nin parçası olabilir. Ancak, bir komut kuyruğu yöneticisi belirtilirken QSG adı kullanılamaz; belirli bir kuyruk yöneticisi adı kullanmanız gerekir.

## Eşgüdüm kuyruğu yöneticileri

MFT koordinasyon kuyruk yöneticisi bir QSG ' nin parçası olabilir. Ancak, bir komut kuyruğu yöneticisinde olduğu gibi, bir koordinasyon kuyruğu yöneticisi belirtilirken QSG adı kullanılamaz; belirli bir kuyruk yöneticisi adı kullanmanız gerekir.

## QSG ' ye bağlanan komutlar

MFT , yönetilen araçlar, aktarımlar ve aracı, komut ya da koordinasyon kuyruğu yöneticilerine bir dizi komut sağlar. Kuyruk yöneticisi bir QSG içindeyse, yalnızca bir aracı kuyruk yöneticisine bağlanan komutları kullanabilirsiniz.

Aşağıda, aracı kuyruk yöneticisine bağlanan komutların bir listesi yer almaktadır:

- **fteCleanAgent**
- **fteCreateAgent**
- **fteCreateBridgeAgent**
- **fteCreateCDAgent**
- **fteDeleteAgent**

Diğer MFT komutlarını çalıştırırken kuyruk yöneticisinin adını belirtmeniz gerektiğini unutmayın.

## Managed File Transfer for z/OS ' in JZOS Java başlatıcısıyla kullanılması

Bu konudaki yönergeleri, kuruluşunuzda, IBM MQ for z/OS sisteminizde Managed File Transfer komutunu kullanmanın alternatif bir yöntemi olarak uygulayabilirsiniz.

### Genel Bakış

Managed File Transfer for z/OS (MFT) standart z/OS kuruluş yordamını kullanır. MFT komutlarını çalıştırmanın diğer bir yolu JCL ve JZOS Java Launcher ' ı kullanmaktır.

Daha fazla ayrıntı için bkz. [JZOS Toplu İş Başlatıcısı ve Araç Takımı](#) .

JCL ' iniz düzgün çalışmazsa, [JZOS ile ortak MFT sorunları](#) başlıklı konuya bakın.

### Örnek JCL

```
//JOHNDOEA JOB 1,MSGCLASS=H
// JCLLIB ORDER=(SCEN.MFT.JCL) (1)
// INCLUDE MEMBER=BFGJCL8 (2)
// DD * (2A)
. ${BFG_PROD}/bin/fteBatch createAgent (3)
export IBM_JAVA_OPTIONS="${BFG_JAVA_OPTIONS} ${BFG_LANG}" (4)
export JZOS_MAIN_ARGS="${BFG_MAIN_ARGS}" (4)
//MAINARGS DD *
-agentName MYAGENT (5)
-f
-agentQMgr MQPD
-p MQPD
/*
```

Burada:

- (1) İçerilen JCL deyimlerinin yeridir
- (2) 1 'deki konumdan belirtilen JCL üyesini dahil et)
- (2A) Bu, // STDENV ' yi genişletir-aşağıya bakın
- (3) Bu, baştaki fte öneki olmadan yürütülecek komuttur
- (4) Bu satırlar gereklidir, JZOS için bilgileri ayarlar
- (5) Komutun parametreleri
- BFGJCL8 üyesi (kendi adınızı seçebilirsiniz) JZOS ' u çağırır. Bu üyede MFTkomutunu çalıştırmak için gereken STEPLIB ve diğer JCL var.

### Eklemeniz gereken diğer JCL

IBM MQ for z/OS kitaplıkları için JCL ' yi ve Db2 günlük kaydedicisini kullanıyorsanız Db2 kitaplıklarını eklemelisiniz.

## Örneğin:

```
//WMQFTE EXEC PGM=JVMLDM86,REGION=0M PARM='+T' (1)
//STÉPLIB DD DSN=SYS1.SIEALNKE,DISP=SHR (2)
//* MQ libraries
// DD DSN=MQM.V920.SCSQAUTH,DISP=SHR MQ Bindings
// DD DSN=MQM.V920.SCSQANLE,DISP=SHR MQ Bindings
// DD DSN=MQM.V920.SCSQLOAD,DISP=SHR MQ Bindings

//* DB2 libraries
// DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNEXIT.DBCP
// DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNLOAD
// DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNLOAD2
//SYSOUT DD SYSOUT=H
//SYSPRINT DD SYSOUT=H
//STDOUT DD SYSOUT=H
//STDERR DD SYSOUT=H

//STDENV DD DSN=SCEN.MFT.JCL(BFGZENV8),DISP=SHR (3)
```

Burada:

- (1) JZOS programının adıdır. Sisteminizdeki sürüm için SYS1.SIEALNKE içine bakın. Ek tanılama sağlamak için PARM = '+ T' ekleyin.
- (2) Bu, JZOS programını içeren veri kümesidir.
- (3) Bu, bir kabuk komut dosyasının üye adıdır. MFT için gerekli parametreleri tanımlar. Bkz. [“MFT dosyasını tanımlamak için kabuk komut dosyası” sayfa 740.](#)

Herhangi bir veri kümesi ve üye olabilir. JCL işi bunu genişlettiği için dosyada sonuncu olması gerekir. [“Örnek JCL” sayfa 739](#) içinde 2A 'ya bakın.

## MFT dosyasını tanımlamak için kabuk komut dosyası

“Eklemeniz gereken diğer JCL” sayfa 739 örneğinde BFGZENV8 üyesi kullanılır. Bu, JZOS profiline dayalıdır.

Bilmeniz gereken:

- Java ' in kurulu olduğu konum
- IBM MQ for z/OS Java kitaplıklarının ve MFT kitaplıklarının konumu.
- Bir kullanıcı kimliğinin IBM MQ for z/OS yöneticisi olarak kabul edilmesi için belirli bir grupta olması gerekir. Bu grubun adı gerekli
- İletiler için İngilizce kullanmıyorsanız, hangi dili belirtmeniz gerektiğini bilmeniz gerekir.

## Örnek dosya

```
# This is a shell script that configures
# any environment variables for the Java JVM.
# Variables must be exported to be seen by the launcher.
# Use PARM='+T' and set -x to debug environment script problems
set -x
# . /etc/profile
#
# Java configuration (including MQ Java interface)
#
export _BPXK_AUTOCVT="ON"
export JAVA_HOME="/java/java71_bit64_sr3_fp30/J7.1_64/"
export PATH="/bin:${JAVA_HOME}/bin/classic/"
LIBPATH="/lib:/usr/lib:${JAVA_HOME}/bin"
LIBPATH="$LIBPATH:${JAVA_HOME}/bin/classic"
LIBPATH=$LIBPATH: "/mqm/V9R2M0/java/lib/"
export LIBPATH

export BFG_JAVA_HOME="${JAVA_HOME}"
export BFG_WTO="YES"
export BFG_GROUP_NAME=MQADM
export BFG_PROD="/mqm/V9R2M0/mqft"
export BFG_CONFIG="/u/johndoe/fteconfig"
```



```
# export BFG_LANG=" -Duser.language=de "  
export BFG_LANG=" "
```

Burada:

```
export _BPXK_AUTOCVT = "ON "
```

Unicode dönüştürmesi için gereklidir

```
export JAVA_HOME = "/java/java71_bit64/J7.1_64/"
```

Java dizininin konumudur. Javayolunun adını belirtin. Bu dizin, bin ve diğer dizinleri içerir.

```
export PATH= "/bin: ${JAVA_HOME}/bin/classic/"
```

Java yürütülür deyimleri için yol deyimini ayarlar

```
LIBPATH= "/lib:/usr/lib:${JAVA_HOME}/bin"
```

Java yürütülür deyimleri için kitaplık yolunu ayarlar

```
LIBPATH=$LIBPATH: ${JAVA_HOME}/bin/classic"
```

LIBPATH deyimine daha fazla Java kitaplığı ekler.

```
LIBPATH=$LIBPATH: "/mqm/V9R2M0/java/lib/"
```

Kitaplık yoluna IBM MQ for z/OS kitaplıklarını ekler. z/OS UNIX System Services içinde IBM MQ for z/OS kitaplıklarınızın adını belirtin.

```
LIBPATH ' ı Dışa Aktar
```

LIBPATH değişkenini JZOS için kullanılır kılar

```
export BFG_JAVA_HOME = "${JAVA_HOME}"
```

BFG\_JAVA\_HOME değerini, yukarıda belirtilen JAVA\_HOME değerine ayarlar

```
export BFG_WTO = "YES "
```

BFG\_WTO değerinin YES olarak ayarlanması, iletilerin WTO kullanılarak iş günlüğünde görüntülenmesine neden olur

```
export BFG_GROUP_NAME=MQADM
```

Belirtilen grubun üyesi olan kullanıcı kimlikleri IBM MQ for z/OS yöneticileri olarak değerlendirilir

```
export BFG_PROD = "/mqm/V9R2M0/mqft"
```

MFT kodunun bulunduğu yol

```
export BFG_DATA= "/u/johndoe/fteconfig"
```

MFT yapılandırma bilgilerinin depolandığı yerdir

```
# export BFG_LANG = " -Duser.language= de"
```

Dili Almanca olarak tanımlamak için açıklama satırı yapılmış bir deyim

```
export BFG_LANG = ""
```

Dili varsayılan olarak (İngilizce) belirtir.

MFT ürününün /lib/messages/BFGNVMessages\_\*.properties içindeki içeriği, kullanılabilir dilleri listeler. Varsayılan değer, İngilizcenin kullanıldığı anlamına gelen değeri boş bırakmaktır.

### İlgili görevler

“yapılandırmaManaged File Transfer for z/OS” sayfa 710

Managed File Transfer for z/OS , bileşenin doğru şekilde çalışmasını sağlamak için özelleştirme gerektirir.

[Managed File Transfer planlaması](#)

IBM i

## IBM i üzerinde MFT ' ın yapılandırılması

Managed File Transfer komutunu kurduktan sonra kullanmaya başlamak için, eşgüdüm kuyruğu yöneticiniz ve aracısına ilişkin bir yapılandırmayı tamamlamanız gerekir.

### Bu görev hakkında

Kurduktan sonra, dosyaları aktarmak için eşgüdüm kuyruğu yöneticilerini ve araçlarını kullanabilmek için Managed File Transfer tarafından sağlanan yapılandırma komut dosyalarını yeni eşgüdüm kuyruğu yöneticileri ve yeni araçlar için çalıştırmanız gerekir. Yarattığınız araçları başlatmanız gerekir.

## Yordam

1. Tüm yeni eşgüdüm kuyruğu yöneticileri için: *coordination\_qmgr\_name.mqsc* dosyasındaki MQSC komutlarını eşgüdüm kuyruğu yöneticisine karşı çalıştırın. Eşgüdüm kuyruğu yöneticisi kuruluşla aynı bilgisayarda değilse, MQSC komut dosyasını kuyruk yöneticisinin bulunduğu bilgisayara kopyalayın ve komut dosyasını çalıştırın.

- a) IBM i komut satırından şu komutu kullanarak qshell 'i başlatın: CALL QSHELL
- b) Şu dizine geçin: /QIBM/UserData/mqm/mqft/config/*coordination\_qmgr\_name*
- c) *coordination\_qmgr\_name* ögesini kuyruk yöneticinizin adıyla değiştirerek şu komutu verin:

```
/QSYS.LIB/QMQM.LIB/RUNMQSC.PGM coordination_qmgr_name < coordination_qmgr_name.mqsc
```

Bunun yerine eşgüdüm kuyruğu yöneticisini el ile yapılandırabilirsiniz. Daha fazla bilgi için bkz [“MFT için koordinasyon kuyruk yöneticisini yapılandırma” sayfa 746](#).

2. Tüm yeni araçlar için: *agent\_name\_create.mqsc* dosyasındaki MQSC komutlarını aracı kuyruk yöneticisine karşı çalıştırın.

Aracı kuyruk yöneticisi, aracıyla aynı bilgisayarda değilse, MQSC komut dosyasını kuyruk yöneticisinin bulunduğu bilgisayara kopyalayın ve komut dosyasını çalıştırın.

- a) IBM i komut satırından şu komutu kullanarak qshell 'i başlatın: CALL QSHELL
- b) Şu dizine geçin: /QIBM/UserData/mqm/mqft/config/*agent\_qmgr\_name*/agents
- c) *agent\_qmgr\_name* ögesini aracı kuyruk yöneticinizin adıyla değiştirerek ve *agent\_name* ögesini aracınızın adıyla değiştirerek şu komutu verin:

```
/QSYS.LIB/QMQM.LIB/RUNMQSC.PGM agent_qmgr_name < agent_name_create.mqsc
```

Aracı kuyruk yöneticisini el ile yapılandırabilirsiniz. Daha fazla bilgi için, bkz. [“MFT Agent kuyruk yöneticilerinin yapılandırılması” sayfa 752](#).

3. QMFT altsistemini kuruluşun bir parçası olarak başlatmadıysanız, IBM i komut satırından aşağıdaki komutu kullanarak QMFT altsistemini başlatın: STRSBS SBSD (QMQMMFT/QMFT) ya da STRSBS QMQMMFT/QMFT

4. **fteStartAgent** komutunu kullanarak yeni araçlarınızı başlatın.

- a) IBM i komut satırından şu komutu kullanarak qshell 'i başlatın: CALL QSHELL
- b) Şu dizine geçin: /QIBM/ProdData/mqm/bin
- c) AGENT ' i aracınızın adıyla değiştirerek şu komutu verin:

```
./fteStartAgent AGENT
```

## Sonraki adım

Bir aracının erişebileceği dosya sistemi alanlarını sınırlamak için kum havuzları ayarlamanız önerilir. Bu özellik, [MFT aracı kum havuzlarıyla çalışmabaşlıklı](#) konuda açıklanmaktadır.

### İlgili kavramlar

[“MFT ' nin ilk kullanım için yapılandırılması” sayfa 742](#)

Managed File Transfer araçlarını ve kuyruk yöneticilerini ilk kez kullanmak istediğinizde bazı yapılandırma görevlerini bir kez gerçekleştirmeniz gerekir.

## MFT ' nin ilk kullanım için yapılandırılması

Managed File Transfer araçlarını ve kuyruk yöneticilerini ilk kez kullanmak istediğinizde bazı yapılandırma görevlerini bir kez gerçekleştirmeniz gerekir.

### İlgili kavramlar

[“bağlanmaIBM MQ” sayfa 743](#)

Managed File Transfer ile ilgili iletişim de dahil olmak üzere IBM MQ kuyruk yöneticileriyle tüm ağ iletişimi IBM MQ kanallarını içerir. IBM MQ kanalı, bir ağ bağlantısının bir ucunu temsil eder. Kanallar, ileti kanalları ya da MQI kanalları olarak sınıflandırılır.

“Çok eşgörünümlü bir kuyruk yöneticisinin MFT ile çalışacak şekilde yapılandırılması” sayfa 749  
IBM WebSphere MQ 7.0.1 , çok eşgörünümlü kuyruk yöneticilerinin yaratılmasını destekler. Çok eşgörünümlü bir kuyruk yöneticisi yedek sunucuda otomatik olarak yeniden başlatılır. Managed File Transfer , çok eşgörünümlü aracı kuyruk yöneticilerine, çok eşgörünümlü bir koordinasyon kuyruk yöneticisine ve çok eşgörünümlü bir komut kuyruk yöneticisine bağlantıyı destekler.

### **İlgili görevler**

“MFT ağ kuyruğu yöneticilerinin yapılandırılması” sayfa 744

Managed File Transfer ağınızda birden çok IBM MQ kuyruk yöneticisi varsa, bu IBM MQ kuyruk yöneticileri birbirleriyle uzaktan iletişim kurabilmelidir.

“MFT Agent kuyruk yöneticilerinin yapılandırılması” sayfa 752

Kuruluştan sonra, aracı kuyruk yöneticisi için gerekli yapılandırmayı gerçekleştirmek üzere `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` dizininde `agent_name_create.mqsc` komut dosyasını çalıştırın. Ancak, bu yapılandırmayı el ile yapmak istiyorsanız, aracı kuyruk yöneticisinde bu adımları tamamlayın.

“MFT için koordinasyon kuyruk yöneticisini yapılandırma” sayfa 746

**fteSetupCoordination** komutunu çalıştırdıktan sonra, koordinasyon kuyruk yöneticisi için gerekli yapılandırmayı gerçekleştirmek üzere `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` dizininde `coordination_qmgr_name.mqsc` komut dosyasını çalıştırın. Ancak, bu yapılanışı el ile yapmak istiyorsanız, eşgüdüm kuyruğu yöneticisinde aşağıdaki adımları izleyin.

“MFT Agent ya da Logger komut veri kümesi yaratılması” sayfa 709

Belirli bir Managed File Transfer Agent ya da Managed File Transfer Logger için Managed File Transfer komut şablonu veri kümesinden belirli bir eşgüdüm için bir PDSE veri kümesi yaratabilirsiniz.

“z/OS üzerinde var olan bir MFT Agent ya da Logger komut veri kümesinin güncellenmesi” sayfa 720  
Managed File Transfer komut şablonu veri kümesinden oluşturulan bir Managed File Transfer komutu PDSE kitaplığı veri kümesini güncelleyebilirsiniz.

### **İlgili başvurular**

MFT Aracı kuyruğu ayarları

MFT sistem kuyrukları ve sistem konusu

“MFT günlük iletilerini tutma” sayfa 751

Managed File Transfer , eşgüdüm kuyruğu yöneticisine dosya aktarma ilerleme durumunu ve günlük bilgilerini gönderir. Eşgüdüm kuyruğu yöneticisi bu bilgileri SYSTEM.FTE konusu. Abonelik yoksa, bu bilgiler korunmaz.

## **bağlanmaIBM MQ**

Managed File Transfer ile ilgili iletişim de dahil olmak üzere IBM MQ kuyruk yöneticileriyle tüm ağ iletişimi IBM MQ kanallarını içerir. IBM MQ kanalı, bir ağ bağlantısının bir ucunu temsil eder. Kanallar, ileti kanalları ya da MQI kanalları olarak sınıflandırılır.

## **Managed File Transfer ve kanallar**

Managed File Transfer , istemci kipinde araçları aracı kuyruk yöneticilerine bağlamak ve komut uygulamalarını (örneğin, **fteCreateTransfer**) komutlarına ve koordinasyon kuyruk yöneticilerine bağlamak için MQI kanallarını kullanır. Varsayılan yapılandırmada, bu bağlantılar SYSTEM.DEF.SVRCONN. Bu varsayılanlar nedeniyle, temel Managed File Transfer kuruluşu için herhangi bir MQI kanalını değiştirmeniz gerekmez.

Altı tip ileti kanalı uç noktası vardır, ancak bu konu yalnızca gönderen-alıcı çiftlerini kapsar. Diğer kanal birleşimleriyle ilgili bilgi için bkz. Distributed queuing components .

## Gerekli ileti yolları

IBM MQ iletileri yalnızca ileti kanalları üzerinden seyahat edebilir; bu nedenle, Managed File Transfer' in gerektirdiği tüm ileti yolları için kanalların kullanılabilir olduğundan emin olmanız gerekir. Bu yolların doğrudan olması gerekmez; iletiler gerektiğinde ara kuyruk yöneticilerinden geçebilir. Bu konu yalnızca doğrudan noktadan noktaya iletişimi kapsar. Bu seçeneklere ilişkin ek bilgi için [Uzak kuyruk yöneticisine nasıl erişir](#) başlıklı konuya bakın.

Managed File Transfer tarafından kullanılan iletişim yolları şunlardır:

### Aracıya

Dosyalar arasında aktarılan her iki aracı, ilişkili kuyruk yöneticileri arasında iki yönlü iletişim gerektirir. Bu yol toplu verileri taşıdığından, yolu gereksinimlerinize göre mümkün olduğunca kısa, hızlı ya da ucuz hale getirmeyi düşünün.

### Aracıyı koordinasyona

Bir aktarıma katılan araçlardan gelen günlük iletileri, koordinasyon kuyruğu yöneticisine ulaştırabilmelidir.

### Aracıya komut

Komut uygulamalarını ya da IBM MQ Explorer (komut kuyruğu yöneticisini kullanarak) bağlantı kuran herhangi bir kuyruk yöneticisinin, bu komut uygulamalarının denetlemek için kullanıldığı araçların kuyruk yöneticilerine ileti gönderebilmesi gerekir. Geri bildirim iletilerinin komutlarla gösterilmesini sağlamak için iki yönlü bir bağlantı kullanın.

Daha fazla bilgi için, kuruluşunuzun kullandığı platform ya da platformlar için *IBM MQ kuruluşunun doğrulanması* başlıklı konuya bakın.

## İlgili kavramlar

[“Çok eşgörünümlü bir kuyruk yöneticisinin MFT ile çalışacak şekilde yapılandırılması” sayfa 749](#) IBM WebSphere MQ 7.0.1 , çok eşgörünümlü kuyruk yöneticilerinin yaratılmasını destekler. Çok eşgörünümlü bir kuyruk yöneticisi yedek sunucuda otomatik olarak yeniden başlatılır. Managed File Transfer , çok eşgörünümlü aracı kuyruk yöneticilerine, çok eşgörünümlü bir koordinasyon kuyruk yöneticisine ve çok eşgörünümlü bir komut kuyruk yöneticisine bağlantıyı destekler.

## İlgili görevler

[“MFT ağ kuyruğu yöneticilerinin yapılandırılması” sayfa 744](#)

Managed File Transfer ağınızda birden çok IBM MQ kuyruk yöneticisi varsa, bu IBM MQ kuyruk yöneticileri birbirleriyle uzaktan iletişim kurabilmelidir.

[“MFT için koordinasyon kuyruk yöneticisini yapılandırma” sayfa 746](#)

**fteSetupCoordination** komutunu çalıştırdıktan sonra, koordinasyon kuyruk yöneticisi için gerekli yapılandırmayı gerçekleştirmek üzere `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` dizininde `coordination_qmgr_name.mqsc` komut dosyasını çalıştırın. Ancak, bu yapılanışı el ile yapmak istiyorsanız, eşgüdüm kuyruğu yöneticisinde aşağıdaki adımları izleyin.

## MFT ağ kuyruğu yöneticilerinin yapılandırılması

Managed File Transfer ağınızda birden çok IBM MQ kuyruk yöneticisi varsa, bu IBM MQ kuyruk yöneticileri birbirleriyle uzaktan iletişim kurabilmelidir.

## Bu görev hakkında

Kuyruk yöneticilerinizi birbirleriyle iletişim kurabilecek şekilde yapılandırmanın iki yolu vardır:

- Bir IBM MQ kuyruk yöneticisi kümesi ayarlayarak.

IBM MQ kuyruk yöneticisi kümeleri ve bunların nasıl yapılandırılacağı hakkında bilgi için bkz. [“Kuyruk yöneticisi kümesinin yapılandırılması” sayfa 283.](#)

- Kuyruk yöneticileri arasında aşağıdaki gibi açıklanan kanallar oluşturularak:

## Kuyruk yöneticileri arasındaki kanalları ayarlama

Kuyruk yöneticileriniz arasında aşağıdaki ileti kanallarını ayarlayın:

- Aracı kuyruk yöneticisinden koordinasyon kuyruk yöneticisine
- Komut kuyruğu yöneticisinden aracı kuyruk yöneticisine.
- Aracı kuyruk yöneticisinden komut kuyruğu yöneticisine (komutlar tarafından geribildirim iletilerinin gösterilmesini sağlamak için).
- Komut kuyruğu yöneticisinden koordinasyon kuyruğu yöneticisine
- Aracı kuyruk yöneticisinden Managed File Transfer ağındaki diğer herhangi bir aracı kuyruk yöneticisine

Bu iletişimi nasıl kuracağınıza ilişkin daha fazla bilgiye gereksinim duyarsanız, şu bilgilerle başlayın: [MQSC kullanarak uzak IBM MQ nesnelere yönetme](#).

Önerilen bazı adımlar şunlardır:

## Yordam

1. IBM MQ kuyruk yöneticisinde koordinasyon kuyruk yöneticisiyle aynı adı taşıyan bir iletim kuyruğu yaratın.

Aşağıdaki MQSC komutunu kullanabilirsiniz:

```
DEFINE QLOCAL(coordination-qmgr-name) USAGE(XMITQ)
```

2. IBM MQ kuyruk yöneticisinde, Managed File Transfer koordinasyon kuyruk yöneticisine bir gönderen kanalı oluşturun.

Önceki adımda yaratılan iletim kuyruğunun adı, bu kanal için gerekli bir parametredir.

Managed File Transfer for IBM MQ üzerindeki araçlar için iletiler boş bir biçimle yayınlanır.

Aşağıdaki MQSC komutunu kullanabilirsiniz:

```
DEFINE CHANNEL(channel-name) CHLTYPE(SDR) CONNAME('coordination-qmgr-host(coordination-qmgr-port)')
XMITQ(coordination-qmgr-name) CONVERT(NO)
```

**Not:** CONVERT (NO) değerini, yalnızca gerekliyse ayarlayın.

3. Managed File Transfer koordinasyon kuyruk yöneticisinde, IBM MQ kuyruk yöneticisine yönelik bir alıcı kanal yaratın. Bu alıcı kanalına, IBM MQ kuyruk yöneticisindeki gönderen kanalla aynı adı verin.

Aşağıdaki MQSC komutunu kullanabilirsiniz:

```
DEFINE CHANNEL(channel-name) CHLTYPE(RCVR)
```

## Sonraki adım

Daha sonra, eşgüdüm kuyruk yöneticiniz için yapılandırma adımlarını izleyin: [“MFT için koordinasyon kuyruk yöneticisini yapılandırma” sayfa 746](#).

### İlgili kavramlar

[“bağlanma IBM MQ” sayfa 743](#)

Managed File Transfer ile ilgili iletişim de dahil olmak üzere IBM MQ kuyruk yöneticileriyle tüm ağ iletişimi IBM MQ kanallarını içerir. IBM MQ kanalı, bir ağ bağlantısının bir ucunu temsil eder. Kanallar, ileti kanalları ya da MQI kanalları olarak sınıflandırılır.

[“Çok eşgörünümlü bir kuyruk yöneticisinin MFT ile çalışacak şekilde yapılandırılması” sayfa 749](#)

IBM WebSphere MQ 7.0.1, çok eşgörünümlü kuyruk yöneticilerinin yaratılmasını destekler. Çok eşgörünümlü bir kuyruk yöneticisi yedek sunucuda otomatik olarak yeniden başlatılır. Managed File Transfer, çok eşgörünümlü aracı kuyruk yöneticilerine, çok eşgörünümlü bir koordinasyon kuyruk yöneticisine ve çok eşgörünümlü bir komut kuyruk yöneticisine bağlantıyı destekler.

### İlgili görevler

[“MFT için koordinasyon kuyruk yöneticisini yapılandırma” sayfa 746](#)

**fteSetupCoordination** komutunu çalıştırdıktan sonra, koordinasyon kuyruk yöneticisi için gerekli yapılandırmayı gerçekleştirmek üzere `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` dizininde `coordination_qmgr_name.mqsc` komut dosyasını çalıştırın. Ancak, bu yapılanışı el ile yapmak istiyorsanız, eşgüdüm kuyruğu yöneticisinde aşağıdaki adımları izleyin.

## MFT için koordinasyon kuyruk yöneticisini yapılandırma

**fteSetupCoordination** komutunu çalıştırdıktan sonra, koordinasyon kuyruk yöneticisi için gerekli yapılandırmayı gerçekleştirmek üzere `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` dizininde `coordination_qmgr_name.mqsc` komut dosyasını çalıştırın. Ancak, bu yapılanışı el ile yapmak istiyorsanız, eşgüdüm kuyruğu yöneticisinde aşağıdaki adımları izleyin.

## Bu görev hakkında

### Yordam

1. SYSTEM.FTE.
2. SYSTEM.FTE kuyruğunu SYSTEM.QPUBSUB.QUEUE.NAMELIST ad listesine ekleyin.
3. SYSTEM.FTE konu dizisiyle SYSTEM.FTE adlı bir konu oluşturun.
4. SYSTEM.FTE konusu ALLAVAIL olarak ayarlandı.
5. Eşgüdüm kuyruğu yöneticisinin PSMODE (Yayınlama/Abone Olma kipi) özniteliğinin ENABLED olarak ayarlandığından emin olun.

### Sonraki adım

`strmqm -c` komutunu, koordinasyon kuyruğu yöneticisi olarak yapılandırılmış bir kuyruk yöneticisinde çalıştırırsanız, komut adım 2 içinde yapılan değişikliği siler (SYSTEM.FTE kuyruğunu SYSTEM.QPUBSUB.QUEUE.NAMELIST ad listesine ekler). Bunun nedeni, `strmqm -c` varsayılan IBM MQ nesnelere yeniden oluşturup Managed File Transfer değişikliklerini tersine çevirmesidir. Bu nedenle, kuyruk yöneticisini `strmqm -c` ile başlatırsanız, aşağıdaki adımlardan birini tamamlayın:

- Kuyruk yöneticisinde `coordination_qmgr_name.mqsc` komut dosyasını yeniden çalıştırın.
- [adım 2'](#) ni yineleyin.

### İlgili kavramlar

[“bağlanma IBM MQ” sayfa 743](#)

Managed File Transfer ile ilgili iletişim de dahil olmak üzere IBM MQ kuyruk yöneticileriyle tüm ağ iletişimi IBM MQ kanallarını içerir. IBM MQ kanalı, bir ağ bağlantısının bir ucunu temsil eder. Kanallar, ileti kanalları ya da MQI kanalları olarak sınıflandırılır.

[“Çok eşgörünümlü bir kuyruk yöneticisinin MFT ile çalışacak şekilde yapılandırılması” sayfa 749](#)

IBM WebSphere MQ 7.0.1, çok eşgörünümlü kuyruk yöneticilerinin yaratılmasını destekler. Çok eşgörünümlü bir kuyruk yöneticisi yedek sunucuda otomatik olarak yeniden başlatılır. Managed File Transfer, çok eşgörünümlü aracı kuyruk yöneticilerine, çok eşgörünümlü bir koordinasyon kuyruk yöneticisine ve çok eşgörünümlü bir komut kuyruk yöneticisine bağlantıyı destekler.

### İlgili görevler

[“MFT ağ kuyruğu yöneticilerinin yapılandırılması” sayfa 744](#)

Managed File Transfer ağınızda birden çok IBM MQ kuyruk yöneticisi varsa, bu IBM MQ kuyruk yöneticileri birbirleriyle uzaktan iletişim kurabilmelidir.

### İlgili başvurular

[fteSetupKoordinasyonu](#)

## IBM MQ Dosya Aktarımı Yapısının Yaratılması

Managed File Transfer yapısını, aynı makinedeki bir kuyruk yöneticisine bağlı tek bir aracıya dayalı olarak yapılandırabilirsiniz.

## Bu görev hakkında

MFT yapılandırması, aracının bulunacağı makinedeki IBM MQ DataPath altında bir dosya yapısında depolanır.

Aşağıdaki örnek yapılandırma, SAMPLECOORD adlı IBM MQ 8.0 kuyruk yöneticisinde (güvenlik geçersiz kılınmış) MFT ve SAMPLEAGENT adlı tek bir MFT aracı içerir:

```
+--- config
      +--- SAMPLECOORD
            +--- command.properties
            +--- coordination.properties
            +--- SAMPLECOORD.mqsc
            +--- agents
                  +--- SAMPLEAGENT
                        +--- agent.properties
                        +--- SAMPLEAGENT_create.mqsc
                        +--- SAMPLEAGENT_delete.mqsc

+--- logs
      +--- SAMPLECOORD
            +--- agents
                  +--- SAMPLEAGENT
                        +--- logs
```

Bu örnek, kuyruk yöneticisi güvenliğinin devre dışı bırakıldığını varsayar. **runmqsc** içinde çalıştırılan aşağıdaki komutlar, kuyruk yöneticisi yeniden başlatıldıktan sonra güvenliği devre dışı bırakır:

```
runmqsc queue manager
alter qmgr CONNAUTH(NONE);
alter qmgr CHLAUTH(DISABLED);
end;
```

IBM MQ 8.0 ya da sonraki bir sürümde MFT içinde güvenlik etkinleştirilmiş olarak yapılandırma için **CONNAUTH** , kullanıcı kimliği ve parola kimlik bilgilerini sağlamak üzere bir kuyruk yöneticisine bağlanan tüm MFT komutlarını gerektirir. Her komut için **-mquserid** ve **-mqpassword** ek parametrelerini uygulayabilir ya da bir **MQMFTCcredentials.xml** dosyası tanımlayabilirsiniz. Aşağıdaki örnek kimlik bilgileri dosyası, kuyruk yöneticisine bağlanırken MyPassword parolasının kullanılacağı fteuser kullanıcı kimliğini tanımlar: SAMPLECOORD:

```
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCcredentials"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/MQMFTCcredentials MQMFTCcredentials.xsd">
  <tns:qmgr mqPassword="MyPassword" MyUserId="fteuser" name="SAMPLECOORD"/>
</tns:mqmftCredentials>
```

Daha fazla bilgi için bkz. [MFT ve IBM MQ bağlantı kimlik doğrulaması](#).

### Notlar:

- MFT yapılandırma dizininizi bulmak için **fteDisplayVersion -v** komutunu kullanın.
- z/OS kullanıcıları için **MQMFTCcredential.xml** dosyası, değişken kayıt biçimi (RECFM = V) ya da tanımlanmamış kayıt biçimi (RECFM = U) olan bölümlenmiş bir veri kümesinde üye olarak bulunabilir.
- Güvenlik etkinleştirilmiş yapılandırmada, kimlik bilgilerini ilgili kuyruk yöneticisiyle ilişkilendirmek için aşağıdaki adımları izleyin: **-F full\_credential\_file\_path**.
- **MQMFTCcredential.xml** içindeki açık metin parolası aşağıdaki komut kullanılarak karartılabilir:

```
fteObfuscate -f full_file_path_to_MQMFTCcredentials.xml
```

## Yordam

1. Bir eşgüdüm kuyruğu yöneticisi yaratın.

Eşgüdüm kuyruğu yöneticisi, araçlarından tüm aktarma günlüğü ve durum bilgilerini almak için kullanılan tek bir kuyruk yöneticisidir. Aşağıdaki komutu çalıştırın:

```
fteSetupCoordination -coordinationQMGr coordination_qmgr_name
```

Bu, temel üst düzey yapılandırmayı oluşturur ve *coordination\_qmgr\_name.mqsc*' i çağırmak için bir IBM MQ komut dosyası oluşturur.

Bundan sonra, aşağıdaki IBM MQ komutu çalıştırılarak yapılan işin kuyruk yöneticisine yüklenmesi gerekir:

```
runmqsc queue manager name < coordination_qmgr_name.mqsc
```

**Not:** Bir kuyruk yöneticisine TCP istemcisi bağlantısı için aşağıdakileri kullanabilirsiniz:

```
fteSetupCoordination -coordinationQMGr coordination_qmgr_name  
-coordinationQMGrHost coordination_qmgr_host -coordinationQMGrPort coordination_qmgr_port  
-coordinationQMGrChannel coordination_qmgr_channel
```

Yaratılan *coordination\_qmgr\_name.mqsc* için, eşgüdüm kuyruk yöneticisinin çalıştığı makede **runmqsc** komutunu çalıştırmanız gerekir.

## 2. Komut kuyruğu yöneticisini yaratın.

Komut kuyruğu yöneticisi, IBM MQ altyapısının MFT isteklerini ilgili aracıya yönltebilmesi için önceden yapılandırılmış tek bir kuyruk yöneticisidir. Aşağıdaki komutu çalıştırın:

```
fteSetupCommands -connectionQMGr Command QM Name -p Coordination QM Name
```

Bu, koordinasyon dizininde bir *command.properties* dosyası oluşturur. -p ' in isteğe bağlı olduğunu ve komutlar varsayılan eşgüdüm için ayarlandıysa gerekli olmadığını unutmayın.

**Not:** Bir kuyruk yöneticisine TCP istemcisi bağlantısı için aşağıdakileri kullanabilirsiniz:

```
fteSetupCommands -p coordination_qmgr_name -commandQMGr connection_qmgr_name  
-commandQMGrHost connection_qmgr_host -commandQMGrPort connection_qmgr_port  
-commandQMGrChannel connection_qmgr_channel
```

## 3. Aracıyı oluşturun.

Aracı, dosya gönderip alabilen bir uygulamadır. Aşağıdaki komutu çalıştırın:

```
fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMGr agent_qmgr_name
```

Bu, aracı yapılandırmasını eşgüdüm altında oluşturur ve aracının yapılandırma dizininde *agent\_name.mqsc* çağrılmasını sağlamak için bir IBM MQ komut dosyası oluşturur.

IBM MQ komut dosyasını kuyruk yöneticisine yüklemek için aşağıdaki IBM MQ komutunu çalıştırın:

```
runmqsc agent_qmgr_name < agent_name_create.mqsc file
```

**Not:** Bir kuyruk yöneticisine TCP istemcisi bağlantısı için aşağıdakileri kullanabilirsiniz:

```
fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMGr agent_qmgr_name  
-agentQMGrHost agent_qmgr_host -agentQMGrPort agent_qmgr_port -agentQMGrChannel  
agent_qmgr_channel
```

## 4. Aracıyı başlatın.

Aşağıdaki komutu çalıştırın:

```
fteStartAgent -p coordination_qmgr_name agentName
```



Aracı arka planda başlar ve komut istemi döndürülür. Aracının çalışıp çalışmadığını denetlemek için şu komutu çalıştırın:

```
fteListAgents -p coordination_qmgr_name
```

Bu, araçların durumunu gösterir. Aracı başarıyla çalışıyorsa, READY (Hazır) durumunda olduğu bildirilir.

## Sonuçlar

Temel bir MFT altyapısı kullanıma hazırdır ve artık aktarım istemek için **fteCreateTransfer** komutunu kullanabilirsiniz. Diğer bir seçenek olarak, IBM MQ Explorer kullanılabilir, aktarımları yaratmak ve izlemek için MFT eklentilerini kullanın.

3. Adım yinelenerek yapılandırmaya daha fazla aracı eklenebilir: Aracı oluşturun. TCP istemci bağlantısı kullanılıyorsa, bunlar farklı makinelerde olabilir. Farklı makineler için her makine için **fteSetupCoordination** ve **fteSetupCommands** komutları yinelenmelidir, ancak mqsc komut dosyalarının çalıştırılması gerekmez.

Daha karmaşık yapılandırmalar, koordinasyon ve her aracı için ayrı kuyruk yöneticilerine sahip olabilir. Bu durumlarda, çeşitli kuyruk yöneticilerinin birbirine bağlanması gerekir.

### İlgili kavramlar

MFT aracınız **fteListAgents** komutuyla listelenmediğinde ne yapmanız gerekir?

### İlgili başvurular

[fteSetupKoordinasyonu](#)

[fteSetupKomutları: MFT command.properties dosyasını oluşturun](#)

[fteCreateAracı](#)

**fteObfuscate**: hassas verileri şifrele

[MFT kimlik bilgileri dosya biçimi](#)

[MFTagent.properties dosyası](#)

## Çok eşgörunümlü bir kuyruk yöneticisinin MFT ile çalışacak şekilde yapılandırılması

IBM WebSphere MQ 7.0.1 , çok eşgörunümlü kuyruk yöneticilerinin yaratılmasını destekler. Çok eşgörunümlü bir kuyruk yöneticisi yedek sunucuda otomatik olarak yeniden başlatılır. Managed File Transfer , çok eşgörunümlü aracı kuyruk yöneticilerine, çok eşgörunümlü bir koordinasyon kuyruk yöneticisine ve çok eşgörunümlü bir komut kuyruk yöneticisine bağlantıyı destekler.

## Çok eşgörunümlü kuyruk yöneticisinin yapılandırılması

**Önemli:** IBM MQ çok eşgörunümlü kuyruk yöneticisini yapılandırma hakkında bilgi için bkz. "[Çok eşgörunümlü kuyruk yöneticileri](#)" sayfa 494. Managed File Transfer ile çalışacak çok eşgörunümlü bir kuyruk yöneticisini yapılandırma girişiminde bulunmadan önce bu bilgileri okuduğunuzdan emin olun.

## Aracı kuyruk yöneticisi olarak çok eşgörunümlü bir kuyruk yöneticisinin kullanılması

Bir aracının çok eşgörunümlü kuyruk yöneticinizin hem etkin hem de yedek eşgörunümüne bağlanmasını sağlamak için aracının `agent.properties` dosyasına `agentQMgrStandby` özelliğini ekleyin. `agentQMgrStandby` özelliği, yedek kuyruk yöneticisi yönetim ortamına ilişkin istemci bağlantıları için kullanılan anasistem adını ve kapı numarasını tanımlar. Özelliğin değeri MQ CONNAME biçiminde (yani, `host_name(port_number)`) verilmelidir.

`agentQMgr` özelliği, çok eşgörunümlü kuyruk yöneticisinin adını belirtir. `agentQMgrHost` özelliği, etkin kuyruk yöneticisi yönetim ortamı için anasistem adını ve `agentQMgrPort` özelliği etkin kuyruk yöneticisi yönetim ortamı için kapı numarasını belirtir. Aracı, istemci kipinde çok eşgörunümlü kuyruk yöneticisinin hem etkin hem de yedek yönetim ortamına bağlanmalıdır.

Daha fazla bilgi için bkz. [MFT agent.properties dosyası](#).

Bu örnek, QM\_JUPITER adlı çok eşgörünümlü bir kuyruk yöneticisine bağlanan AGENT1 için agent.properties dosyasının içeriğini gösterir. QM\_JUPITER ' in etkin yönetim ortamı host1 sisteminde bulunur ve istemci bağlantıları için 1414 numaralı kapıyı kullanır. QM\_JUPITER yedek yönetim ortamı host2 sisteminde ve istemci bağlantıları için 1414 numaralı kapıyı kullanıyor.

```
agentName=AGENT1
agentDesc=
agentQMgr=QM_JUPITER
agentQMgrPort=1414
agentQMgrHost=host1
agentQMgrChannel=SYSTEM.DEF.SVRCONN
agentQMgrStandby=host2(1414)
```

## Eşgüdüm kuyruğu yöneticisi olarak çok eşgörünümlü kuyruk yöneticisinin kullanılması

Çok eşgörünümlü eşgüdüm kuyruk yöneticinizin hem etkin hem de yedek yönetim ortamına yönelik bağlantıları etkinleştirmek için, coordinationQMgrStandby özelliğini Managed File Transfer topolojizdeki tüm coordination.properties dosyalarına ekleyin.

Daha fazla bilgi için bkz. [MFT coordination.properties dosyası](#) .

Bu örnek, QM\_SATÜRN adlı çok eşgörünümlü eşgüdüm kuyruğu yöneticisine yönelik bağlantı ayrıntılarını belirten bir coordination.properties dosyasının içeriğini gösterir. QM\_SATÜRN ' un etkin örneği coordination\_host1 sisteminde ve istemci bağlantıları için 1420 numaralı kapıyı kullanır. QM\_SATÜRN yedek yönetim ortamı coordination\_host2 sisteminde ve istemci bağlantıları için 1420 numaralı kapıyı kullanır.

```
coordinationQMgr=QM_SATURN
coordinationQMgrHost=coordination_host1
coordinationQMgrPort=1420
coordinationQMgrChannel=SYSTEM.DEF.SVRCONN
coordinationQMgrStandby=coordination_host2(1420)
```

Managed File Transfer bağımsız kaydedicisi her zaman bağ tanımlama kipinde kuyruk yöneticisine bağlanmalıdır. Bağımsız günlük kaydediciyi çok eşgörünümlü bir eşgüdüm kuyruk yöneticisi ile kullanırken, bağ tanımlama kipinde farklı bir kuyruk yöneticisine bağlanın. Bunu yapmak için gereken adımlar “MFT bağımsız günlük kaydedicisi için alternatif yapılandırmalar” sayfa 775’inde açıklanmıştır. Bağımsız günlük kaydedicinin kuyruk yöneticisi ile koordinasyon kuyruk yöneticisi arasındaki kanalları, çok eşgörünümlü eşgüdüm kuyruk yöneticisinin her iki eşgörünümlünün de anasistem adı ve kapı numarası ile tanımlamanız gerekir. Bunun nasıl yapılacağını öğrenmek için bkz. “Çok eşgörünümlü kuyruk yöneticileri” sayfa 494.

IBM MQ Explorer için Managed File Transfer eklentisi, istemci kipinde eşgüdüm kuyruğu yöneticisine bağlanır. Çok eşgörünümlü eşgüdüm kuyruğu yöneticisinin etkin yönetim ortamı başarısız olursa, eşgüdüm kuyruğu yöneticisinin yedek yönetim ortamı etkinleşir ve eklenti yeniden bağlanır.

Managed File Transfer komutlar **fteList\*** ve **fteShowAgentDetails** doğrudan koordinasyon kuyruğu yöneticisine bağlanır. Çok eşgörünümlü eşgüdümün etkin eşgörünümlü kullanılamıyorsa, bu komutlar eşgüdüm kuyruğu yöneticisinin yedek yönetim ortamına bağlanmayı dener.

## Komut kuyruğu yöneticisi olarak çok eşgörünümlü kuyruk yöneticisinin kullanılması

Çok eşgörünümlü komut kuyruğu yöneticinizin hem etkin hem de yedek yönetim ortamına yönelik bağlantıları etkinleştirmek için, connectionQMgrStandby özelliğini Managed File Transfer topolojizdeki tüm command.properties dosyalarına ekleyin.

Daha fazla bilgi için bkz. [MFT command.properties dosyası](#) .

Bu örnek, QM\_MARS adlı çok eşgörünümlü bir komut kuyruğu yöneticisine yönelik bağlantı ayrıntılarını belirten bir command.properties dosyasının içeriğini gösterir. QM\_MARS etkin örneği command\_host1

sisteminde bulunur ve istemci bağlantıları için 1424 kapı numarasını kullanır. QM\_MARS yedek yönetim ortamı command\_host2 sisteminde bulunur ve istemci bağlantıları için 1424 kapı numarasını kullanır.

```
connectionQMgr=QM_SATURN
connectionQMgrHost=command_host1
connectionQMgrPort=1424
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
connectionQMgrStandby=command_host2(1424)
```

## İlgili kavramlar

[“bağlanmaIBM MQ” sayfa 743](#)

Managed File Transfer ile ilgili iletişim de dahil olmak üzere IBM MQ kuyruk yöneticileriyle tüm ağ iletişimi IBM MQ kanallarını içerir. IBM MQ kanalı, bir ağ bağlantısının bir ucunu temsil eder. Kanallar, ileti kanalları ya da MQI kanalları olarak sınıflandırılır.

## İlgili görevler

[“MFT ağ kuyruğu yöneticilerinin yapılandırılması” sayfa 744](#)

Managed File Transfer ağınızda birden çok IBM MQ kuyruk yöneticisi varsa, bu IBM MQ kuyruk yöneticileri birbirleriyle uzaktan iletişim kurabilmelidir.

[“MFT için koordinasyon kuyruk yöneticisini yapılandırma” sayfa 746](#)

**fteSetupCoordination** komutunu çalıştırdıktan sonra, koordinasyon kuyruk yöneticisi için gerekli yapılandırmayı gerçekleştirmek üzere `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` dizininde `coordination_qmgr_name.mqsc` komut dosyasını çalıştırın. Ancak, bu yapılanışı el ile yapmak istiyorsanız, eşgüdüm kuyruğu yöneticisinde aşağıdaki adımları izleyin.

## MFT günlük iletilerini tutma

Managed File Transfer , eşgüdüm kuyruğu yöneticisine dosya aktarma ilerleme durumunu ve günlük bilgilerini gönderir. Eşgüdüm kuyruğu yöneticisi bu bilgileri SYSTEM.FTE konusu. Abonelik yoksa, bu bilgiler korunmaz.

## Bilgilerin korunmasını sağlamanın yolları

Aktarma işleminin ilerleyişi ya da günlük bilgileri işiniz için önemliyse, bilgilerin saklandığından emin olmak için aşağıdaki adımlardan birini gerçekleştirmeniz gerekir:

- SYSTEM.FTE/Log konusunda yayınlanan iletileri bir Oracle ya da Db2 veritabanına kopyalamak için Managed File Transfer veritabanı kaydedicisini kullanın.
- Yayınları bir IBM MQ kuyruğunda depolayan SYSTEM.FTE konusuna abonelik tanımlayın. Tüm ilerleme ve günlük iletilerinin kuyrukta tutulduğundan emin olmak için dosya aktarımlarını aktarmadan önce bu aboneliği tanımlayın.
- Sürekli abonelik yaratmak ve aboneliğe teslim edilen yayınları işlemek için ileti kuyruğu arabirimini (MQI) ya da IBM MQ JMS kullanan bir uygulama yazın. Uygulamanın tüm aşama ve günlük iletilerini almasını sağlamak için, herhangi bir dosya aktarılmadan önce bu uygulamanın çalışır durumda olması gerekir.

Bu yaklaşımların her biri, takip eden bölümlerde daha ayrıntılı olarak açıklanmıştır.

Günlük bilgilerini saklamak için IBM MQ Explorer eklentisine güvenmeyin.

## Günlük iletilerini tutmak için Managed File Transfer veritabanı kaydedicisini kullanma

Veritabanı kaydedici, çözümlenme ve denetleme amacıyla günlük bilgilerini bir veritabanına kopyalamak için kullanabileceğiniz isteğe bağlı bir Managed File Transfer bileşenidir. Veritabanı kaydedici, eşgüdüm kuyruğu yöneticisini ve veritabanını barındıran bir sisteme kurduğunuz bağımsız bir Java uygulamasıdır. Veritabanı kaydedicisi hakkında daha fazla bilgi için bkz. [“MFT günlük kaydedicisi yapılandırılması” sayfa 763.](#)

## IBM MQ Explorer eklentisini kullanarak ilerleme durumunu ve günlük iletilerini tutma

IBM MQ Explorer eklentisinin bir yönetim ortamı ilk kez başlatıldığında, yönetim ortamı eşgüdüm kuyruğu yöneticisinde sürekli bir abonelik yaratır. Bu sürekli abonelik, **Aktarma Günlüğü** ve **Yürürlükteki Aktarma İlerleme Durumu** görünümünde görüntülenen bilgileri toplamak için kullanılır.

Sürekli aboneliğin adının başında, aboneliğin IBM MQ Explorer MFT eklentisi, anasistem adı ve kullanıcı adı tarafından oluşturulduğunu gösterir. Örneğin, MQExplorer\_MFT\_Plugin\_HOST\_TJWatson.

Bu örnek, yöneticinin IBM MQ Explorer eklentisinin bir eşgörünümü tarafından artık etkin olmayan kalıcı bir aboneliği silmek istemesi durumunda eklenir.

Eşgüdüm kuyruk yöneticisinde sürekli abonelik kullanılması, iletilerin SYSTEM.MANAGED.DURABLE (DURABLE) kuyrukları. Yüksek hacimli bir Managed File Transfer ağına varsa, IBM MQ Explorer eklentisini nadiren ya da her ikisini birden kullanın; bu ileti verileri yerel dosya sistemini doldurabilir.

Bunun gerçekleşmesini durdurmak için, IBM MQ Explorer eklentisinin eşgüdüm kuyruğu yöneticisine kalıcı olmayan bir abonelik kullandığını belirtin. IBM MQ Explorer içinde aşağıdaki adımları gerçekleştirin:

1. **Pencere > Tercihler > MQ Explorer > Yönetilen Dosya Aktarımı** seçeneğini belirleyin.
2. **Aktarma Günlüğü abonelik tipi** listesinden NON\_DURABLE seçeneğini belirleyin.

### Yayınların IBM MQ kuyruğunda saklanması

Günlük ya da aşama iletilerini bir IBM MQ kuyruğunda saklamak için, koordinasyon kuyruk yöneticisinde iletileri bu kuyruğa ileten bir abonelik yapılandırın. Örneğin, tüm günlük iletilerini LOG.QUEUE, şu MQSC komutunu gönderin:

```
define sub(MY.SUB) TOPICSTR('Log/#') TOPICOBJ(SYSTEM.FTE) DEST(LOG.QUEUE) WSCHEMA(TOPIC)
```

Günlük iletileri bir IBM MQ kuyruğuna iletdikten sonra, kuyruğu kullanan bir IBM MQ uygulaması tarafından işleninceye kadar kuyrukta kalırlar.

### SYSTEM.FTE konusu

IBM MQ tarafından desteklenen uygulama programlama arabirimlerinden birini kullanarak SYSTEM.FTE konusuna kendi sürekli aboneliklerini yöneten uygulamalar yazabilirsiniz. Bu uygulamalar, IBM MQ kuyruğunu alabilir ya da iletileri günlüğe kaydedebilir ve bunlar üzerinde iş gereksinimleriniz için uygun şekilde işlem yapabilir.

Kullanılabilir uygulama programlama arabirimlerine ilişkin ek bilgi için [Uygulamaları geliştirme](#) başlıklı konuya bakın.

## MFT Agent kuyruk yöneticilerinin yapılandırılması

Kuruluştan sonra, aracı kuyruk yöneticisi için gerekli yapılandırmayı gerçekleştirmek üzere `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` dizininde `agent_name_create.mqsc` komut dosyasını çalıştırın. Ancak, bu yapılandırmayı el ile yapmak istiyorsanız, aracı kuyruk yöneticisinde bu adımları tamamlayın.

### Yordam

1. Aracı işlemi kuyruklarını oluşturun.

Bu kuyrukların adı:

- SYSTEM.FTE.COMMAND.*araci\_adi*
- SYSTEM.FTE.DATA.*araci\_adi*
- SYSTEM.FTE.EVENT.*araci\_adi*
- SYSTEM.FTE.REPLY.*araci\_adi*

- SYSTEM.FTE.STATE.*aracı\_adi*

Kuyruk parametreleri ve kuyrukların nasıl kullanıldığına ilişkin bilgi için bkz. [MFT Aracı kuyruğu ayarları](#).

## 2. Aracı yetki kuyruklarını oluşturun.

Bu kuyrukların adı:

- SYSTEM.FTE.AUTHADM1.*aracı\_adi*
- SYSTEM.FTE.AUTHAGT1.*aracı\_adi*
- SYSTEM.FTE.AUTHMON1.*aracı\_adi*
- SYSTEM.FTE.AUTHOPS1.*aracı\_adi*
- SYSTEM.FTE.AUTHSCH1.*aracı\_adi*
- SYSTEM.FTE.AUTHTRN1.*aracı\_adi*

Kuyruk parametreleri ve kuyrukların nasıl kullanıldığına ilişkin bilgi için bkz. [MFT Aracı kuyruğu ayarları](#).

## Sonraki adım

Bir iletişim kuralı köprüsü aracısı oluşturma ve yapılandırma hakkında bilgi için bkz. [fteCreateBridgeAgent \( MFT iletişim kuralı köprüsü aracısı oluşturma ve yapılandırma\)](#) ve [Bir FTPS sunucusu için protokol köprüsünü yapılandırma](#).

### İlgili kavramlar

[“bağlanmaIBM MQ” sayfa 743](#)

Managed File Transfer ile ilgili iletişim de dahil olmak üzere IBM MQ kuyruk yöneticileriyle tüm ağ iletişimi IBM MQ kanallarını içerir. IBM MQ kanalı, bir ağ bağlantısının bir ucunu temsil eder. Kanallar, ileti kanalları ya da MQI kanalları olarak sınıflandırılır.

[“Çok eşgörünümlü bir kuyruk yöneticisinin MFT ile çalışacak şekilde yapılandırılması” sayfa 749](#)

IBM WebSphere MQ 7.0.1 , çok eşgörünümlü kuyruk yöneticilerinin yaratılmasını destekler. Çok eşgörünümlü bir kuyruk yöneticisi yedek sunucuda otomatik olarak yeniden başlatılır. Managed File Transfer , çok eşgörünümlü aracı kuyruk yöneticilerine, çok eşgörünümlü bir koordinasyon kuyruk yöneticisine ve çok eşgörünümlü bir komut kuyruk yöneticisine bağlantıyı destekler.

### İlgili görevler

[“MFT ağ kuyruğu yöneticilerinin yapılandırılması” sayfa 744](#)

Managed File Transfer ağınızda birden çok IBM MQ kuyruk yöneticisi varsa, bu IBM MQ kuyruk yöneticileri birbirleriyle uzaktan iletişim kurabilmelidir.

[“MFT için koordinasyon kuyruk yöneticisini yapılandırma” sayfa 746](#)

**fteSetupCoordination** komutunu çalıştırdıktan sonra, koordinasyon kuyruk yöneticisi için gerekli yapılandırmayı gerçekleştirmek üzere `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` dizininde `coordination_qmgr_name.mqsc` komut dosyasını çalıştırın. Ancak, bu yapılanışı el ile yapmak istiyorsanız, eşgüdüm kuyruğu yöneticisinde aşağıdaki adımları izleyin.

### İlgili başvurular

[MFT Aracı kuyruğu ayarları](#)

[fteSetupKoordinasyonu](#)

## Bir kümedeki birden çok kanal için MFT aracısının yapılandırılması

Kümelenmiş bir yapılandırmada IBM MQ çok kanallı desteğini kullanmak istiyorsanız, önce **agentMultipleChannelsEnabled** özelliğini `true` olarak ayarlayın ve bu konudaki adımları tamamlayın.

## Bu görev hakkında

Bir kümede, çok kanallı destek yalnızca hedef aracının kuyruk yöneticisindeki IBM MQ tanımlamaları tarafından etkinleştirilir.

“MFT ' nin ilk kullanım için yapılandırılması” sayfa 742 içinde listelenen bir Managed File Transfer aracı için gerekli olan standart IBM MQ yapılandırma adımlarına ek olarak bu konudaki adımları tamamlamanız gerekir.

Aşağıdaki yapılandırma örnekleri **runmqsc** komutlarını kullanır.

## Yordam

1. Kullanmak istediğiniz her kanal için bir küme alıcı kanalı tanımlayın. Örneğin, iki kanal kullanıyorsanız:

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
```

Burada:

- *DESTQMGRNAME* , hedef aracının kuyruk yöneticisinin adıdır.
- *MFTCLUSTER* , IBM MQ kümesinin adıdır.

Kanallar için *MFTCLUSTER.DESTMGRNAME\_n* adlandırma kuralını kullanmanız önerilir, ancak bu kural zorunlu değildir.

2. Her kanala karşılık gelen bir kuyruk yöneticisi diğer adı tanımlayın. Örneğin:

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
```

*SYSTEM.FTE.DESTQMGRNAME\_n* gönderen aracı bu biçimdeki kuyruk yöneticisi diğer adlarını aradığından kuyruk yöneticisi diğer adları için adlandırma kuralı. *n* için kullandığınız numaralar 1 'den başlayıp ardışık olmalıdır. Tanımlamaların kaynak aracının kuyruk yöneticisinde kullanılabilmesi için küme çapında kullanıma sunulması gerekir.

Hem kaynak aracı hem de hedef aracının kuyruk yöneticisi diğer adlarının sayısını doğru olarak belirlemesi için, kuyruk yöneticisi için varsayılan bir XMITQ **tanımlamayın** .

## İlgili görevler

“MFT aracısının birden çok kanal için yapılandırılması: kümelenmemiş” sayfa 754

Kümelenmemiş bir yapılandırmada IBM MQ çok kanallı desteği kullanmak istiyorsanız, önce `agentMultipleChannelsEnabled` özelliğini `true` olarak ayarlayın ve bu konudaki adımları tamamlayın.

## İlgili başvurular

MFT agent.properties dosyası

## MFT aracısının birden çok kanal için yapılandırılması: kümelenmemiş

Kümelenmemiş bir yapılandırmada IBM MQ çok kanallı desteği kullanmak istiyorsanız, önce `agentMultipleChannelsEnabled` özelliğini `true` olarak ayarlayın ve bu konudaki adımları tamamlayın.

## Bu görev hakkında

Kümelenmemiş bir yapılandırmada, hem kaynak aracının hem de hedef aracının kuyruk yöneticisindeki IBM MQ tanımlamaları tarafından çok kanallı destek etkinleştirilir.

“MFT ' nin ilk kullanım için yapılandırılması” sayfa 742 içinde listelenen bir Managed File Transfer aracı için gerekli olan standart IBM MQ yapılandırma adımlarına ek olarak bu konudaki adımları tamamlamanız gerekir.

Aşağıdaki adımlar, kaynak ve hedef kuyruk yöneticileri arasında iletişim kurmak için gönderen alıcı kanallarının kullanıldığını varsayar.

Aşağıdaki yapılandırma örnekleri **runmqsc** komutlarını kullanır.

## Yordam

1. Hedef aracının kuyruk yöneticisinde, kullanmak istediğiniz her kanal için bir alıcı kanal tanımlayın. Örneğin, iki kanal kullanıyorsanız:

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(RCVR) TRPTYPE(TCP)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(RCVR) TRPTYPE(TCP)
```

Burada: DESTQMGRNAME, hedef aracının kuyruk yöneticisinin adıdır.

TO.DESTMGRNAME\_n adlandırma kuralı, ancak bu kural zorunlu değildir. Alıcı kanal adları, kaynak aracının kuyruk yöneticisindeki ilgili gönderen kanallarıyla eşleşmelidir.

2. Kaynak aracının kuyruk yöneticisinde, kullanmak istediğiniz her kanal için bir iletim kuyruğu tanımlayın. Örneğin, iki kanal kullanıyorsanız:

```
DEFINE QLOCAL(DESTQMGRNAME_1) USAGE(XMITQ)
DEFINE QLOCAL(DESTQMGRNAME_2) USAGE(XMITQ)
```

İletim kuyrukları için DESTMGRNAME\_n adlandırma kuralını kullanmanız önerilir, ancak bu kural zorunlu değildir. Tanımladığınız iletim kuyruklarına, aşağıdaki adımlarda gönderen kanal tanımlarından ve kuyruk yöneticisi diğer ad tanımlarından gönderme yapılır.

3. Kaynak aracının kuyruk yöneticisinde, kullanmak istediğiniz her kanal için bir gönderen kanalı tanımlayın. Örneğin, iki kanal kullanıyorsanız:

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
XMITQ(DESTQMGRNAME_1)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
XMITQ(DESTQMGRNAME_2)
```

TO.DESTMGRNAME\_n adlandırma kuralı, ancak bu kural zorunlu değildir. Gönderen kanal adları, hedef aracının kuyruk yöneticisindeki ilgili alıcı kanallarıyla eşleşmelidir.

4. Kaynak aracının kuyruk yöneticisinde, her bir kanala karşılık gelen bir kuyruk yöneticisi diğer adı tanımlayın. Örneğin:

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_1)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_2)
```

SYSTEM.FTE.DESTQMGRNAME\_n adlandırma kuralı. n için kullandığınız numaralar 1 'den başlayıp ardışık olmalıdır.

Aracının kuyruk yöneticisi diğer adlarının sayısını doğru olarak saptaması için, kuyruk yöneticisi için varsayılan bir XMITQ **tanımlamayın** .

### İlgili görevler

“Bir kümedeki birden çok kanal için MFT aracısının yapılandırılması” sayfa 753

Kümelenmiş bir yapılandırmada IBM MQ çok kanallı desteğini kullanmak istiyorsanız, önce **agentMultipleChannelsEnabled** özelliğini true olarak ayarlayın ve bu konudaki adımları tamamlayın.

### İlgili başvurular

MFT agent.properties dosyası

## MFT araçlarını MSCS ile yapılandırma

Platform MFT tarafından destekleniyorsa ve Windows sürümlerinden birini çalıştırıyorsa Managed File Transfer (MFT) aracı Microsoft Cluster Service (MSCS) kurulumunu desteklenir.



## Bu görev hakkında

Bu görevde, bir MFT aracısının yedek sisteme geçmesini sağlamak için izleyebileceğiniz iki senaryo açıklanmaktadır:

- 1. senaryo: Aracıyı MSCS kaynağı olarak yapılandırma.
- 2. senaryo: Aracı kuyruk yöneticisini ve aracıyı MSCS kaynakları olarak yapılandırma.

## Yordam

1. senaryo: Aracıyı MSCS kaynağı olarak yapılandırma

- Aracıyı MSCS kaynağı olarak yapılandırmak için aşağıdaki adımları tamamlayın:
  - a) Managed File Transfer ' i kümedeki her makineye yerel olarak kurun.  
Bkz. [Kurma Managed File Transfer](#).
  - b) Kümedeki birincil makinede aracıyı yaratın.  
Aracı, CLIENT iletimini kullanarak Agent kuyruk yöneticisine bağlanacak şekilde yapılandırılmalıdır. Bu aracı için kuyruk yöneticisindeki tüm nesnelere oluşturduğunuzdan emin olun. Bunun nasıl yapılacağını öğrenmek için [Aracıyı ayarlamabaşlıklı](#) konuya bakın.
  - c) Aracıyı Windows hizmeti olarak çalışacak şekilde değiştirin ve Windows Services aracında Agent Service için **Startup Type** (Başlatma Tipi) alanını Manual (El ile) olarak ayarlayarak Windows yeniden başlatıldığında otomatik olarak başlatılmayacak şekilde yapılandırın.  
Daha fazla bilgi için [MFT aracısını Windows hizmeti olarak başlatmabaşlıklı](#) konuya bakın.
  - d) İkincil makinede Senaryo 1 'in "[2](#)" sayfa 756 ve "[3](#)" sayfa 756 adımlarını yineleyin.  
Bu, günlüklere, özelliklere vb. ilişkin dosya yapısının kümedeki diğer makinede bulunmasını sağlar. "[2](#)" sayfa 756. adımda olduğu gibi kuyruk yöneticisi nesnelere yaratmaya gerek olmadığını unutmayın.
  - e) Birincil makinede, aracıyı MSCS denetimi altında 'Soysal Hizmet' olarak ekleyin.  
Bunu yapmak için:
    - a. Kümeyi sağ tıklayın ve **Rol-> Kaynak Ekle-> 'Soysal Hizmet'** seçeneklerini belirleyin.
    - b. Windows hizmetleri listesinden Agent Service olanağını seçin ve **Next** (İleri) düğmesini tıklayarak yapılandırma sihirbazını tamamlayın.  
Aracı hizmeti şimdi bir MSCS kaynağı olarak eklenir. Hata durumunda yedek sisteme geçiş gerçekleşirse, Agent Service diğer makinede başlatılır.

2. senaryo: Aracı kuyruk yöneticisini ve aracıyı MSCS kaynakları olarak yapılandırma

- Aracı kuyruk yöneticisini ve aracıyı MSCS kaynakları olarak yapılandırmak için aşağıdaki adımları tamamlayın:
  - a) Aracı kuyruk yöneticisini MSCS kaynağı olarak çalışacak şekilde yapılandırın.  
Bunun nasıl yapılacağını öğrenmek için bkz. "[Kuyruk yöneticisinin MSCS denetimi altına konması](#)" sayfa 483.
  - b) Kümedeki birincil makinede aracıyı yaratın.  
Aracı, BINDINGS iletimini kullanarak aracı kuyruk yöneticisine bağlanacak şekilde yapılandırılmalıdır. Bu aracı için kuyruk yöneticisindeki tüm nesnelere oluşturduğunuzdan emin olun. Bunun nasıl yapılacağını öğrenmek için [Aracıyı ayarlamabaşlıklı](#) konuya bakın.
  - c) Aracıyı Windows hizmeti olarak çalışacak şekilde değiştirin ve Windows Services aracında Agent Service için **Startup Type** (Başlatma Tipi) alanını Manual (El ile) olarak ayarlayarak Windows yeniden başlatıldığında otomatik olarak başlatılmayacak şekilde yapılandırın.  
Daha fazla bilgi için [MFT aracısını Windows hizmeti olarak başlatmabaşlıklı](#) konuya bakın.
  - d) Aracı kuyruk yöneticisinin (MSCS denetimi altında) ikincil makinede çalıştığından emin olun.  
Bu makinede yaratılan aracı BINDINGS iletimini kullanarak kuyruk yöneticisine bağlanır ve aracı yaratıldığında kullanılabilir olması gerekir.
  - e) İkincil makinede Senaryo 2 'nin "[2](#)" sayfa 756 ve adım "[3](#)" sayfa 756 adımlarını yineleyin.



Bu, günlüklere, özelliklere vb. ilişkin dosya yapısının kümedeki diğer makinede bulunmasını sağlar. “2” sayfa 756. adımda olduğu gibi kuyruk yöneticisi nesnelere yaratmaya gerek olmadığını unutmayın.

f) Aracıyı MSCS denetimi altında 'Sosyal Hizmet' olarak ekleyin.

Bunu yapmak için:

a. Kümeyi sağ tıklayın ve **Rol-> Kaynak Ekle-> 'Sosyal Hizmet'** seçeneklerini belirleyin.

b. Windows hizmetleri listesinden Agent Service olanağını seçin ve **Next(İleri)** düğmesini tıklayarak yapılandırma sihirbazını tamamlayın.

g) Bağımlılık listesine kuyruk yöneticisi kaynağını eklemek için aracı hizmetinin kaynak özelliklerini değiştirin.

Bu, kuyruk yöneticisi kaynağının aracı başlatılmadan önce başlatılmasını sağlar.

h) Kuyruk yöneticisi kaynağını çevrimdışı duruma getirin ve aracı kaynağını çevrimiçi duruma getirin. Hem kuyruk yöneticisi kaynağının hem de aracının başlatılıp başlatılmadığını denetleyin.

Hata durumunda yedek sisteme geçiş gerçekleşirse, aracı hizmeti ve aracı kuyruk yöneticisi ikincil makinede başlatılır.

## Managed File Transfer içinde yüksek kullanılabilirlikli araçlar

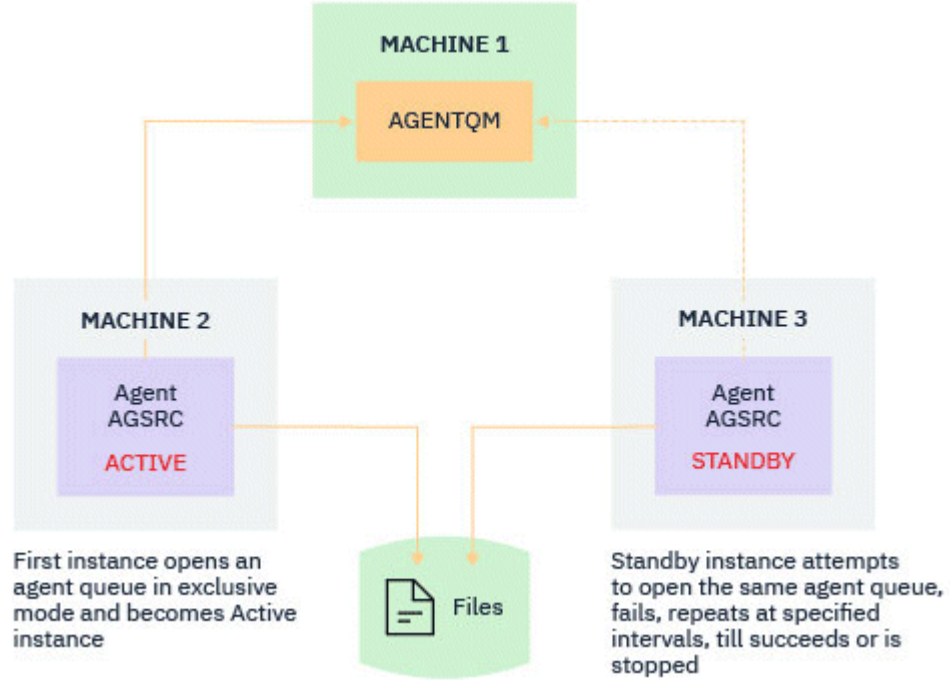
MFT içinde standart ya da köprü araçlarını yüksek kullanılabilirlik (HA) yapılandırmasında çalışacak şekilde yapılandırabilirsiniz. Aynı yapılandırmaya sahip bir çift aracı eşgörünümü, bir yönetim ortamının farklı bir makinede çalışırken bir yönetim ortamının bir makinede çalıştığı HA kurulumuna dahil olur. Her iki eşgörünüm de aynı aracı kuyruk yöneticisine bağlanacak şekilde yapılandırılır.

## Genel Bakış

*Etkin yönetim ortamı* adı verilen iki yönetim ortamından yalnızca biri kütük aktarımlarını işlerken, *yedek yönetim ortamı* adlı diğer yönetim ortamı kısmen kullanıma hazırlanmış durumda ve hiçbir kütük aktarımlarını işleyemiyor.

Etkin bir yönetim ortamı başarısız olduğunda ya da kuyruk yöneticisine bağımlılığını kaybettiğinde, yedek yönetim ortamı kullanıma hazırlama işlemini tamamlar, etkin duruma gelir ve kütük aktarımlarını işlemeye başlar. Etkin eşgörünüm başarısız olduğunda devam eden tüm inflight aktarımları, bilinen son denetim noktasından devam eder.

Aşağıdaki şekilde etkin ve beklemedeki araçların ortak bir yapılandırması



gösterilmektedir:

#### Notlar:

1. Bir aracının bir eşgörünümü iki farklı makinede çalışıyor; eşgörünümlerden biri *etkin eşgörünüm*, diğeri *yedek eşgörünüm* olarak.
2. Aracının her eşgörünümü farklı bir makinede çalışıyor; eşgörünümlerden biri etkin eşgörünüm, diğeri yedek eşgörünüm.
3. Aynı aracı kuyrukları kümesi, aracının her iki eşgörünümü arasında paylaşılır.
4. Yönetilen aktarımları gerçekleştirmek için aracının her iki örneğinin de aynı paylaşılan dosya sistemine erişmesi gerekir.

Etkin-yedek aracı eşgörünümü düzeneği, paylaşılan bir kaynak üzerinde kilit alınarak çalışır. Paylaşılan kaynakta kilit alan aracı eşgörünümü etkin eşgörünüm olurken, diğer eşgörünüm (kilit almayan) yedek eşgörünüm olur.

Burada paylaşılan kaynak yeni bir kuyruktur ( SYSTEM . FTE . HA . <agent name> ). Bu kuyruk, bir IBM MQ 9.1.4 ya da daha sonraki bir aracı yapılandırıldığında otomatik olarak oluşturulur.

#### Süreç nasıl çalışır

Bir HA aracı oluşturmak için, agent . properties dosya kümesinde true olarak ayarlanan **highlyAvailable** agent özelliğiyle birlikte ek **-x** parametresini kullanarak **fteCreateAgent** ya da **fteCreateBridgeAgent** komutunu çalıştırarak iki makinede aynı yapılandırma parametrelerine sahip bir aracı oluşturun.

#### Notlar:

- Her iki yapılanış da aynı aracı kuyruk yöneticisini göstermelidir.
- Gerekli aracı kuyrukları, aracı kuyruk yöneticisinde yalnızca bir kez yaratılmalıdır.

**highlyAvailable** aracı özelliği hakkında daha fazla bilgi için **-x** parametresi ve agent . properties dosyası hakkında daha fazla bilgi için **fteCreateAgent** komutuna bakın.

**Not: fteCreateAgent** ya da **fteCreateBridgeAgent** komutunun çalıştırılması, aracı kuyruk yöneticisinde ve SYSTEM . FTE . HA . agent name kuyruğunda IBM MQ nesnelere yaratmak için gereken

komut dosyalarını içeren bir MQSC dosyası yaratır. Bu MQSC dosyası, **-x** deęiřtirgesini belirtip belirtmedięinizden baęımsız olarak yaratılır.

Yüksek kullanılabilirlikli bir aracı yapılandırması oluřtururken **fteCreateAgent** ya da **fteCreateBridgeAgent** komutu, SYSTEM.FTE/Agents/*agent name* konusuna abone olarak başka bir yerde bulunan aynı aracının bir eřęörünümünün varlıęını denetler. Aynı aracının bir örneęi bulunursa, iki komuttan biri dosya sisteminde gerekli yapılandırmayı oluřturur, ancak aracının oluřturulmasını yeniden yayınlamaz.

Bir aracı HA kipinde bařladıęında:

1. Aracı, SYSTEM.FTE.HA.*agent name* kuyruęunu dıřlayıcı bir GET kipinde açmayı dener.
2. Aracı SYSTEM.FTE.HA.*agent name* kuyruęunu bařarıyla açarsa, bu, bir aracının *etkin eřęörünümü* olur ve başka bir bařlatma iřlemi devam eder.
3. SYSTEM.FTE.HA.*agent name* kuyruęunu dıřlayıcı GET kipinde açma giriřimi MQRC\_OBJECT\_IN\_USE neden koduyla bařarısız olursa, aracının başka bir yerde çalıřmakta olan etkin bir eřęörünümü olduęu anlamına gelir. Bu nedenle, bu eřęörünüm aracının *yedek eřęörünümü* olur.

Yedek eřęörünüm, belirtilen aralıklarla SYSTEM.FTE.HA.*agent name* kuyruęunu açmayı dener. *agent.properties* dosyasında bu amaçla bir ek aracı özellięi **standbyPollInterval** saęlanır.

Varsayılan deęerle, yedek eřęörünüm her beř saniyede bir SYSTEM.FTE.HA.*agent name* kuyruęunu açmayı dener. Bu iřlem, eřęörünüm SYSTEM.FTE.HA.*agent name* kuyruęunu açmayı bařarıncaya ya da **fteStopAgent** komutu kullanılarak durduruluncaya kadar yinelenir.

IBM MQ 9.2.4 ve IBM MQ 9.2.0 Fix Pack 5 özelliklerinden **standbyPollInterval** özellięi, bir yönetim ortamının aracı kuyruk yöneticisiyle baęlantısı kesilirse, yönetim ortamının yeniden baęlanma giriřimleri arasında ne kadar bekleyeceęini saptamak için tüm yönetim ortamları tarafından da kullanılır.

## Birden çok yedek yönetim ortamı

Tüm yedek yönetim ortamları SYSTEM.FTE.HA.*agent name* kuyruęunu dıřlayıcı bir GET kipinde almayı dener ve etkin yönetim ortamı bařarısız olduktan sonra bařarılı olan yönetim ortamı etkin yönetim ortamı olur.

Etkin yönetim ortamı, bilinen tüm yedek yönetim ortamlarına iliřkin bilgileri saklar ve bilgileri aracı durumu yayınının bir parçası olarak yayınlar. **fteShowAgentDetails** komutunun çıkıřı, aracı GET REST API yanıtı ve IBM MQ Explorer MFT eklentisi, tüm yedek yönetim ortamlarına iliřkin bilgileri görüntüler.

Daha fazla bilgi için **fteShowAgentDetails** komutu ve aracı [GET REST API](#) yanıtının örnek çıkıřlarına bakın.

XML biçiminde aracı durumu bilgileri örnekleri için bkz. [MFT aracı durumu iletileri](#).

## Sürüm gereksinimi

Etkin ve yedek araçlar IBM MQ 9.1.4 ya da daha yüksek olmalıdır.



### Uyarı:

- Yüksek kullanılabilirlik kiplerinde IBM MQ sürümlerini IBM MQ 9.1.4 sürümünden önce yapılandıramaz ya da bařlatamazsınız.
- Hem etkin hem de yedek yönetim ortamlarının kodun aynı sürümünü çalıřtırması gerekir.

Her iki yönetim ortamının da aynı sürümde olmasını saęlamak için, etkin ve yedek yönetim ortamlarının sürümünün geçerlilięi denetlenir. Eřęörünümler arasındaki iletiřim için geçici bir dinamik kuyruk kullanılır. *agent.properties* dosyasında tanımlanan iki aracı özellięi ( **dynamicQueuePrefix** ve **modelQueueName**), geçici dinamik kuyruęun adını oluřturur.

## **Managed File Transfer içinde yüksek kullanılabilirlikli araçlar için gerekli bilgiler**

Yüksek kullanılabilirlik yapılandırmasında çalışan standart ya da köprü MFT araçları hakkında bilmeniz gereken çeşitli bilgi tipleri vardır. Bu bilgiler, aracının başlatılacağı farklı yöntemleri, günlük dosyasındaki aracının eşgörünümünün nasıl tanımlanacağını ve aracıya ilişkin durum bilgilerini içerir.

### **Aracı başlatılması**

#### **Bir aracının eşgörünümü başka bir yerde HA olmayan kipte çalışıyor**

Aracının HA aracı olarak yapılandırılmamış başka bir eşgörünümünü başlatma girişiminde bulunulursa, önce `SYSTEM.FTE.HA.agent name` kuyruğunda kilit alınıp alınamayacağını görmek için bir denetim yapılır.

Diğer yönetim ortamı HA olmayan bir kipte başlatıldığından, `SYSTEM.FTE.HA.agent name` kuyruğundaki kilit bu yönetim ortamı tarafından edinilir. Aracı kullanıma hazırlama işlemini sürdürür, ancak komut kuyruğu özel olarak başka bir eşgörünüm tarafından açıldığı için daha sonraki bir noktada başarısız olur.

Bu durumda, aşağıdaki örnekte gösterilen iletiler aracının `output0.log` dosyasına kaydedilir ve aracı, komut kuyruğunu her 30 saniyede bir açma girişiminde bulunmaya devam eder:

```
BFGMQ1045I: Aracıyı sistem kuyruğu 'SYSTEM.FTE.COMMAND.SRC' konfigürasyonu NOSHARE ya da DEFSOPT (PAYLAŞILAN).
```

```
BFGAG0035W: Aracı, kuyruğu açma girişimi sırasında MQI neden kodu 2042 'yi aldı.  
'SYSTEM.FTE.COMMAND.SRC'  
ve kanal 'MFT_HA_CHN'. Aracı işlemi her 30 saniyede bir yeniden deneyecek.
```

#### **Bir aracının eşgörünümü başka bir yerde HA kipinde çalışıyor**

Aracının HA aracı olarak yapılandırılmamış başka bir eşgörünümünü başlatma girişiminde bulunulursa, önce `SYSTEM.FTE.HA.agent name` kuyruğunda kilit alınıp alınamayacağını görmek için bir denetim yapılır.

Diğer yönetim ortamı etkin bir yönetim ortamı olarak çalıştığı için, kilit alma girişimi başarısız oldu. Eşgörünüm başlatılamaz ve aşağıdaki hata iletileri aracının `output0.log` dosyasına kaydedilir:

```
BFGAG0194E: Bu aracının bir örneği zaten başka bir yerde çalışıyor.  
Bu nedenle bu eşgörünüm devam edemez ve sona erecek.
```

### **Windows**

#### **Aracıyı Windows hizmeti olarak başlatma**

Windows' da bir aracıyı Windows hizmeti olarak başlatabilirsiniz.

Başlatma sırasında Windows , MFT aracısını normal ya da HA kipinde başlatır. Aracı HA kipinde çalışacak şekilde yapılandırıldıysa, hangi yönetim ortamının önce kilidi edindiğine bağlı olarak hizmet etkin ya da yedek eşgörünüm olarak çalışır.

### **Günlük dosyasında bir aracının eşgörünüm tipinin tanımlanması**

Bilgi iletileri, eşgörünümün tipini belirtmek için aracının `output0.log` dosyasına yazılır. Bir aracı eşgörünümü etkin bir eşgörünüm olarak başladığında aşağıdaki ileti yazılır:

```
BFGAG0193I: Aracı etkin bir örnek olarak başarıyla başlatıldı.
```

Bir aracı eşgörünümü yedek eşgörünüm olarak başladığında aşağıdaki ileti yazılır:

```
BFGAG0193I: Aracı beklemedeki bir eşgörünüm olarak başarıyla kullanıma hazırlandı.
```

### **Aracı durumu güncelleştirmeleri**

Aynı aracının iki eşgörünümü çalıştığından, aracı durumu yayınında her iki eşgörünümle ilgili bilgilere sahip olmanız gerekir.

Etkin eşgörünümün, her iki eşgörünümün durumunu yayınlayan örnek olduğunu unutmayın.

## Yedekleme eşgörünümlü

Aracı durumu yayınlanırken etkin eşgörünüm, yedek eşgörünüm yayınının yaşını denetler.

`agent.properties` dosyasında bu amaçla iki ek özellik vardır:

- **standbyStatusExpiry** , bekleme durumu iletilsinin aracının komut kuyruğuna konması için süre bitimidir. Bir aracının etkin eşgörünümlü o dönemde bu iletiyi işlemezse, iletinin süresi dolar.

Varsayılan olarak **standbyStatusExpiry** değeri 30 saniyedir. İleti, bekleme durumu iletileri üzerinden aktarma isteklerinin öncelikli olarak işlenmesini sağlayan düşük öncelikli bir iletidir ( 9).

- **standbyStatusPublishInterval** , yedek eşgörünümlü durumunu yayınlama sıklığını ayarlar.

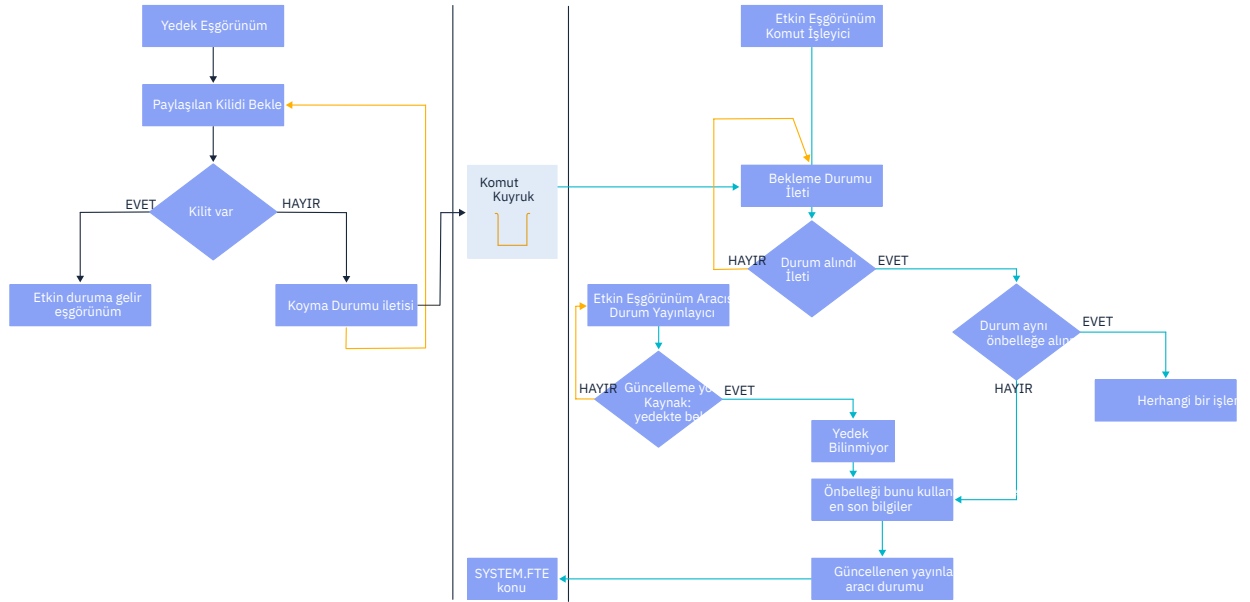
## Etkin eşgörünüm

Etkin yönetim ortamı, yedek yönetim ortamındaki durum güncellemelerini işlemek için aşağıdaki işlemleri gerçekleştirir:

1. `SYSTEM.FTE.COMMAND.<agent name>` kuyruğundan iletiyi alır ve ileti işlemeyi bir işçi iş parçacığına gönderir.
2. İşçi iş parçacığı, ileti gövdesinden içeriği alır, aracı durumu nesnesini yedek eşgörünümlü bilgileriyle günceller ve aracı durumu yayınlayıcıya durumu yayınlamasını bildirir.
3. Aracı durumu yayınlayıcısı durumu yayınlamaya başlar.

Beklemedeki durum bilgilerini önbelleğe almak için eniyilemelerin burada yapıldığını unutmayın. Bir istek yapıldığında, aracı durumu yayınlayıcısı yeni durumu önbelleğe alındı durumuyla denetler ve yalnızca bir fark varsa yayınlamaya başlar.

Aşağıdaki çizge, etkin ya da yedek eşgörünümlü bir aracının durumunu yayınlamak için izlediği akışı açıklar:



## Yüksek düzeyde kullanılabilir araçlarda eşgörünümlü atılması, hata durumunda yedek sisteme geçiş ve bakım

Yüksek düzeyde kullanılabilir Managed File Transfer eşgörünümlüleri atılabilir, çeşitli şekillerde başarısız olabilir ve bakım gerekebilir.

## Yedek eşgörünüm durumunun atılması

Etkin yönetim ortamının aktarımlarla meşgul olduğu ve yedek yönetim ortamı durum iletilerini işleyemediği ya da yedek yönetim ortamının başarısız olduğu ya da herhangi bir nedenle durum iletilerini yayınlamadığı durumlar olabilir.

Bu tür senaryolarda, yedek eşgörünümün varlığından haberdar olan etkin aracı, yedek eşgörünümün listesinden kaldırılmadan önce, `agent.properties` dosyasında **standbyStatusDiscardTime** özelliğinin belirlediği değeri bekler. Bu özelliğin varsayılan değeri, **standbyStatusPublishInterval** özelliğinin iki katı olan 600 saniyedir.

## Normal olarak bir eşgörünüm üzerinde hata oluşması

Normal bir hata durumunda yedek sisteme geçiş işlemi gerçekleştirilmek için **fteStopAgent** komutunu **-i** seçeneğiyle birlikte kullanmanız gerekir.

Bu, etkin eşgörünümün hemen durdurulmasını sağlar. Bir aracıyı **-i** seçeneği olmadan durdurursanız, etkin eşgörünüm tarafından devam eden tüm aktarımlar tamamlanıncaya kadar aracı çalışmaya devam eder; bu nedenle, yedek sisteme geçiş uzun sürebilir.

Bilinen son kontrol noktasından gelen herhangi bir inflight transfer devam ediyor.

## Diğer durumlarda bir eşgörünüm üzerinde hata oluşması

Etkin bir eşgörünüm olağan olmayan bir şekilde sona ererse ya da makinenin tamamı başarısız olursa, aracı kuyruğu bağlantısı kesilir ve kuyruk yöneticisi, `SYSTEM.FTE.HA.<agent name>` kuyruğu ve bağlantıları da içinde olmak üzere tüm açık kuyrukları kapatır.

Bu nedenle, yedek eşgörünüm özel GET işlemi alır ve aracının kullanıma hazırlanmasının geri kalanını tamamlar.

Yine, bilinen son kontrol noktalarından herhangi bir inflight transfer devam ediyor.

## Kuyruk yöneticisiyle bağlantı kesilirse

### İstemci kipi

Bir aracı işlemi birkaç iş parçacığını içerir. Varsayılan iş parçacıkları dışında, örneğin, düzenli aralıklarla aracı durumunu yayınlayan bir iş parçacığı, her aktarma isteği, aktarım tamamlandıktan sonra sona erecek bir iş parçacığı kümesiyle işlenir.

Bu iş parçacıklarının çoğu aracı kuyruk yöneticisine bağlanır ve iletileri koyar ve alır. Bir ağ sorunu ya da kuyruk yöneticisinin başarısız olması nedeniyle bu bağlantılardan herhangi biri kopabilir. Herhangi bir iş parçacığı bozuk bir bağlantı sorunu saptadığında, iş parçacığı ana iş parçacığını kurtarma işlemi başlatması için bilgilendirir ve sona erdirir.

Ana iş parçacığı, kurulmakta olan kuyruk yöneticisine bağlantı beklemek için başka bir iş parçacığı başlatır. Yeniden bağlandıktan sonra, aracı için özel GET alma girişiminde bulunuldu. Başarılı olursa, aracı kurtarma işlemi tamamlamaya devam eder ve etkin eşgörünüm olur. Dışlayıcı GET alma girişimi başarısız olursa, yönetim ortamı yedek olur.

### Bağ Tanımları kipi

Bağ tanımlama kipinde bağlantı kurulurken bir aracı bağlantıyı kaybederse aracı işlemi sona erer. Süreç denetleyicisi, aracının yeniden başlatılmasını ister. Bir aracı yeniden başlatıldığında, kendisi için özel GET 'yi edinmeye çalışılır.

Aracı başarılı olursa, etkin bir eşgörünüm olur; tersi durumda aracı yedek eşgörünüm olur.

## Bakım düzeyi yükseltmeleri uygulanıyor

Yüksek kullanılabilirlikli araçlara bakım uygulama adımları, çok eşgörünümlü kuyruk yöneticileri için belgelenenlere benzer. Daha fazla bilgi için bkz. [Windows üzerinde çok eşgörünümlü kuyruk yöneticilerine](#)

bakım düzeyi güncellemeleri uygulama ya da AIX üzerinde çok eşgörünümlü kuyruk yöneticilerine bakım düzeyi güncellemeleri uygulamaya da Linux üzerinde çok eşgörünümlü kuyruk yöneticilerine bakım düzeyi güncellemeleri uygulama.

Bakım uygulanmadan önce, bakım düzeyinin uygulanacağı makinede çalışan aracıyı durdurmanız gerekir. Etkin bir yönetim ortamını güncelliyorsanız, aktarımların sürekliliği için etkin yönetim ortamını yedek yönetim ortamına geçirmeniz gerekir.

Yükseltme tamamlandıktan sonra, aracı yönetim ortamını başlatmanız, yürürlükteki etkin yönetim ortamını yükseltiletilen yönetim ortamına yedeği devreye sokmanız ve yedek yönetim ortamını yükseltmeniz gerekir.


## Aracıların ürünün önceki bir sürümünden geçirilmesi

Agents migrated from versions of IBM MQ prior to IBM MQ 9.1.4 run as non highly available. Managed File Transfer araçlarını daha önceki bir sürümden yeni düzeye geçirmebaşlıklı konudaki yordamı izleyerek, bunları yüksek kullanılabilirlik kipinde çalıştırabilirsiniz.

## MFT günlük kaydedicisi yapılandırılması

Managed File Transfer dosyaları aktardığında, işlemleriyle ilgili bilgileri koordinasyon kuyruğu yöneticisindeki bir konuya yayınlır. Veritabanı kaydedici, analiz ve denetim amacıyla bu bilgileri bir veritabanına kopyalamak için kullanabileceğiniz isteğe bağlı bir Managed File Transfer bileşenidir.

Kaydedicinin üç sürümü vardır:

-  bağımsız dosya kaydedici
- bağımsız veritabanı kaydedicisi
- Java Platform, Enterprise Edition (Java EE) kaydedici

## IBM i üzerinde kaydediciler



Managed File Transfer kaydedicileri IBM i platformunda desteklenmez.

## Bağımsız dosya kaydedici



Bağımsız dosya kaydedici, eşgüdüm kuyruk yöneticisini barındıran sistemde ya da eşgüdüm kuyruk yöneticisine bağımlılığı olan bir kuyruk yöneticisini barındıran bir sistemde çalışan bir Java işlemidir. Bağımsız dosya kaydedici, ilişkili kuyruk yöneticisine bağlanmak için IBM MQ bağ tanımlarını kullanır. Bağımsız kaydedici, **fteCreateLogger** komutu kullanılarak oluşturulur.



Windows oturumunuzu kapattığınızda dosya kaydedicinin çalışmaya devam etmesini sağlamak için bağımsız dosya kaydediciyi bir Windows hizmeti olarak çalıştırabilirsiniz ve sistem yeniden başlatıldığında otomatik olarak başlatılacak şekilde yapılandırılabilir. Daha fazla bilgi için bkz [“MFT bağımsız dosya kaydedicinin kurulması” sayfa 764.](#)

Bağımsız dosya kaydedici aşağıdaki altyapılarda desteklenmez:

-  z/OS
-  IBM i

## Bağımsız veritabanı kaydedici

Bağımsız veritabanı kaydedici, bir kuyruk yöneticisini ve veritabanını barındıran bir sisteme kurduğunuz bir Java uygulamasıdır. Bağımsız veritabanı kaydedici genellikle koordinasyon kuyruk yöneticisiyle aynı sisteme kurulur, ancak koordinasyon kuyruk yöneticisiyle bağlantısı olan herhangi bir kuyruk yöneticisiyle



aynı sisteme de kurulabilir. Bağımsız veritabanı kaydedici, ilişkili kuyruk yöneticisine bağlanmak için IBM MQ bağ tanımlarını ve bir Db2 ya da Oracle veritabanına bağlanmak için tip 2 ya da tip 4 JDBC sürücüsünü kullanır. Bağımsız veritabanı kaydedicisi, kuyruk yöneticisi ve veritabanı üzerinde genel bir hareketi koordine etmek ve verileri korumak için kuyruk yöneticisinin XA desteğini kullandığından, bu bağlantı tipleri gereklidir.

**Windows** Windows sistemi kullanıyorsanız, Windows oturumunuzu kapattığınızda kaydedicilerin çalışmaya devam etmesini sağlamak için bağımsız kaydedicileri Windows hizmetleri olarak çalıştırabilirsiniz. Daha fazla bilgi için, bağımsız veritabanı kaydedicisi için bkz. [“MFT bağımsız veritabanı kaydedicisi kuruluyor” sayfa 771](#).

## Java EE veritabanı kaydedici

Java EE veritabanı kaydedicisi, uygulama sunucusuna kurduğunuz bir EAR dosyası olarak sağlanır. Java EE veritabanı günlük kaydedicisi, diğer kurumsal uygulamalarınızla birlikte yönetilebileceğinden, var olan bir Java EE uygulama sunucusu ortamınız varsa, bağımsız veritabanı günlük kaydedicisini kullanmaktan daha uygun olabilir. Java EE veritabanı kaydedicisi, IBM MQ sunucunuzu ve veritabanınızı barındıran sistemlere ayrı bir sisteme de kurabilirsiniz. Java EE veritabanı kaydedicisi, Db2 ve Oracle veritabanlarıyla birlikte kullanılmak üzere desteklenir. Java EE veritabanı kaydedici, WebSphere Application Server 7.0 üzerine kurulduğunda Oracle Real Application Clusters (Gerçek Uygulama Kümeleri) olanağını da destekler.

Günlüğe kaydedicinin nasıl yapılandırılacağına ilişkin yönergeler için aşağıdaki konulara bakın:

- [“MFT bağımsız dosya kaydedicinin kurulması” sayfa 764](#)
- [“MFT bağımsız veritabanı kaydedicisi kuruluyor” sayfa 771](#)
- [“MFT için Java EE veritabanı kaydedicisi kuruluyor” sayfa 776](#)

### İlgili görevler

[“MFT olanağının uzak veritabanıyla kullanılması” sayfa 773](#)

Uzak sistemdeki bir veritabanıyla iletişim kurmak için Managed File Transfer kaydedicisi kullanabilirsiniz.

### İlgili başvurular

[MFT kaydedici hata işleme ve ileti reddi](#)

[MFT kaydedici yapılandırma özellikleri](#)

## **ALW** MFT bağımsız dosya kaydedicinin kurulması

Bağımsız dosya kaydedici, IBM MQ bağ tanımlama kipini ya da istemci kipini kullanarak bir eşgüdüm kuyruğu yöneticisine bağlanması gereken bir Java işlemidir. Bağımsız bir dosya kaydedici tanımlamak için **ftCreateLogger** komutunu kullanın ve bu konudaki adımları izleyin.

### Bu görev hakkında

Bağımsız dosya kaydediciyle ilgili daha fazla bilgi için bkz. [“MFT günlük kaydedicisi yapılandırılması” sayfa 763](#). Bu konudaki adımlar, bir koordinasyon kuyruk yöneticisine bağlanmak için bir kaydediciyi yapılandırıyor. Alternatif kaydedici yapılandırmaları için bkz. [“MFT bağımsız günlük kaydedicisi için alternatif yapılandırmalar” sayfa 775](#)

Bağımsız dosya kaydedici aşağıdaki altyapılarda desteklenmez:

- **z/OS** z/OS
- **IBM i** IBM i

### Yordam

1. Managed File Transfer Logger bileşeninin kurulu olduğundan emin olun. Ek bilgi için [Managed File Transfer product options](#) başlıklı konuya bakın.



2. Koordinasyon kuyruğu yöneticisini belirterek **fteCreateLogger** komutunu çalıştırın ve bağımsız dosya kaydedicini yaratmak için `-loggerType` parametresini FILE olarak ayarlayın. Daha fazla bilgi için bkz. [fteCreateLogger](#).
3. İsteğe bağlı: Özel bir biçim kullanmak istiyorsanız, **fteCreateLogger** komutuyla yaratılan XML dosyasını değiştirebilirsiniz. Günlük biçimi tanımlaması `FileLoggerFormat.xml` dosyasında bulunur. Daha fazla bilgi için bkz. [“MFT bağımsız dosya kaydedici biçimi” sayfa 765](#).
4. Günlük kaydedici kuyruklarını yaratmak için, **fteCreateLogger** komutuyla sağlanan MQSC komutlarını eşgüdüm kuyruğu yöneticinizle karşılaştırarak çalıştırın.
5. Günlüğe kaydedici işlemini çalıştırmak için bir kullanıcı tanımlayın ve o kullanıcı için izinleri yapılandırın. Daha fazla bilgi için bkz. [“MFT bağımsız dosya kaydedicisi için kullanıcı erişimini yapılandırma” sayfa 771](#).
6. İsteğe bağlı: **fteCreateLogger** komutunu çalıştırdığınızda oluşturulan `logger.properties` dosyasını düzenleyerek bağımsız dosya kaydedicini daha fazla yapılandırabilirsiniz. Bu dosya, anahtar-değer çiftlerinden oluşan bir Java özellikler dosyasıdır. `logger.properties` dosyası `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` dizininde bulunur. Kullanılabilir özellikler ve bunların etkileri hakkında daha fazla bilgi için bkz. [MFT kaydedici yapılandırma özellikleri](#).
7. **Windows**  
İsteğe bağlı: Windows sistemi kullanıyorsanız, bağımsız dosya kaydedicini Windows hizmeti olarak çalıştırabilirsiniz. **fteModifyLogger** komutunu `-s` parametresiyle çalıştırın. Daha fazla bilgi için bkz. [fteModifyLogger](#).
8. **fteStartLogger** komutuyla bağımsız dosya kaydedicini başlatın. Daha fazla bilgi için bkz. [fteStartKaydedici](#).  
Önceki adımı gerçekleştirdiyseniz ve **fteModifyLogger** komutunu Windows üzerinde `-s` parametresiyle kullandıysanız, bağımsız dosya kaydedici bir Windows hizmeti olarak başlar.
9. Kaydedici çıkışı denetleyin. Bağımsız dosya kaydedici, dosya aktarma denetim verileri ve günlüğe kaydedici tanımlama verileri olmak üzere iki tip çıkış oluşturur. Dosya aktarımı denetim verileri `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs` içinde bulunabilir. Günlüğe kaydedici tanımlama verilerini `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name` içinde bulabilirsiniz.
10. **fteStopLogger** komutunu kullanarak kaydedicini durdurabilirsiniz. Daha fazla bilgi için bkz. [fteStopLogger](#).

## Sonuçlar

### İlgili görevler

[“MFT bağımsız dosya kaydedicisi için kullanıcı erişimini yapılandırma” sayfa 771](#)

Bir test ortamında, normal kullanıcı hesabınıza yeni ayrıcalıklar ekleyebilirsiniz. Bir üretim ortamında, işi yapmak için gereken en düşük izinlere sahip yeni bir kullanıcı yaratmanız önerilir.

### İlgili başvurular

[MFT kaydedici yapılandırma özellikleri](#)

[fteStartKaydedici](#)

[fteCreateKaydedici](#)

[fteModifyKaydedici](#)

[fteStopKaydedici](#)

[“MFT bağımsız dosya kaydedici biçimi” sayfa 765](#)

Dosya kaydedici tarafından yazılan ileti bilgilerinin biçimi `FileLoggerFormat.xml` dosyasında tanımlanabilir.

[MFT kaydedicisi için yetkiler](#)

### **ALW** **MFT bağımsız dosya kaydedici biçimi**

Dosya kaydedici tarafından yazılan ileti bilgilerinin biçimi `FileLoggerFormat.xml` dosyasında tanımlanabilir.

Günlüğe kaydediciye ilişkin yapılandırma dizini `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_named` dizininde bulunur. Yeni bir dosya kaydedici yaratılırken, bu dosyanın, dosya kaydedici tarafından kullanılan varsayılan bir tanımlama kümesini içeren bir sürümü yaratılır. Varsayılan günlük biçimi tanımlamasıyla ilgili daha fazla bilgi için bkz. [MFT bağımsız dosya kaydedici varsayılan günlük biçimi](#).

Kendi özel günlük biçiminizi belirtmek istiyorsanız, `FileLoggerFormat.xml` dosyasını düzenleyin.

## Özel bir günlük biçimi tanımlaması

Günlük biçimi tanımlaması, her ileti tipinin biçim tanımlaması olan bir ileti tipleri kümesinden oluşur. Bir ileti tipine ilişkin biçim tanımlaması, XPATH biçiminde sağlanan bir ekleme kümesi ve her bir eklemeyi ayırmak için kullanılan bir ayırıcıdan oluşur. Eklemeler sıralaması, içeriğin günlük dosyalarına çıkış için oluşturulan satırlara yerleştirilme sırasını belirler. Örneğin, bu, `callStarted` ileti tipinin tanımlamasıdır:

```
<callStarted>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/
        @time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/agent/
        @agent</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/agent/@QMgr</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/job/name</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/
        call/command/@type</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/
        call/command/@name</insert>
      <insert type="system" width="0" ignoreNull="true">callArguments</insert>
    </inserts>
    <separator></separator>
  </format>
</callStarted>
```

Bu biçim, günlük dosyasında şu şekilde bir satır üretir:

```
2011-11-25T10:53:04;414d5120514d5f67627468696e6b20206466cf4e20004f02; [CSTR];
AGENT1;AGENT_QM;Managed Call;executable;echo;call test;
```

Biçim tanımlamasında sağlanan eklemeler, bilgilerin günlük dosyasındaki satırda görüldüğü sırada olur. `FileLoggerFormat.xml` dosyasının biçimini tanımlayan XML şeması hakkında daha fazla bilgi için bkz. [Bağımsız dosya kaydedici biçimi XSD](#).

## İleti Tipleri

FTE araçları, `SYSTEM.FTE/Log` alt konusuna farklı ileti tipleri aralığı yazar. Daha fazla bilgi için bkz. [SYSTEM.FTE konu](#). Günlük dosyası tanımlaması şu ileti tiplerine ilişkin biçim tanımlamalarını içerebilir:

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
scheduleSubmitTransferSet
transferStarted
transferCancelled
transferComplete
transferDelete
transferProgress
```

İletilerin biçimi değişebilir. İleti tiplerinin çoğu, SYSTEM.FTE/Log alt konusundan tüketilen her günlük iletisi için günlük dosyasına tek bir satır yazar. Bu, günlük biçimi tanımında sağlanan XPATH adreslerinin iletinin köküyle ilişkili olduğu basit duruma yol açar. Çıkış yazmak için bu yöntemi kullanan ileti tipleri şunlardır:

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
transferStarted
transferCancelled
transferComplete
transferDelete
```

Günlük iletisi yazmak için kullanılan diğer yöntem, bir günlük iletisi içindeki aktarım kümesindeki öğeleri göstermek için birden çok satır kullanır. Bu durumda, sağlanan biçim günlük iletisi içindeki aktarım kümesindeki her öğeye uygulanır. Aktarma kümesindeki her öğeye özgü bilgileri içermek istiyorsanız, öğeyi XPATH kökü olarak kullanmak için belirtilen XPATH gereklidir. Çıkış yazmak için bu yöntemi kullanan ileti tipleri şunlardır:

```
scheduleSubmitTransferSet
transferProgress
```

Aktarma kümesindeki her öğe için bir çıkış satırı yazılır. Bir aktarma kümesindeki tüm öğeler için düzeltilmesini istediğiniz bilgiler, günlük iletisinin köküne göre XPATH adreslerini kullanmaya devam edebilir. Aşağıdaki basitleştirilmiş transferProgress biçim tanımlaması örneğinde, düzeltilen zaman damgası ve aktarım tanıtıcısıdır. Kök olarak bir öğeyle görelili olan tüm bilgiler, yazılan her satır için değişiklik gösterir. Bu örnekte, her öğe için kaynak ve hedef dosya bilgileri yazılır.

```
<transferProgress>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/
        @time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">source/file |
        source/queue</insert>
      <insert type="user" width="0" ignoreNull="false">source/file/@size |
        source/queue/@size</insert>
      <insert type="user" width="5" ignoreNull="true">source/@type</insert>
      <insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
      <insert type="user" width="0" ignoreNull="false">destination/file |
        destination/queue</insert>
      <insert type="user" width="0" ignoreNull="false">destination/file/@size |
        destination/queue/@size</insert>
      <insert type="user" width="5" ignoreNull="true">destination/@type</insert>
      <insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
      <insert type="user" width="0" ignoreNull="true">status/supplement</insert>
    </inserts>
    <separator></separator>
  </format>
</transferProgress>
```

Bu, bu biçimde bir ya da daha çok satırın günlük dosyası girişini üretir:

```
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
;/src/test1.file;3575;file;leave ;/dest/test1.file;3575;file;overwrite;;
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
;/src/test2.file;3575;file;leave ;/dest/test2.file;3575;file;overwrite;;
```

## Biçim Ekle

Bir ileti tipi için biçim tanımlanırken kullanılabilecek iki tip ekleme vardır: kullanıcı ve sistem. Araya ekleme (insert) tipi, araya ekleme (insert) öğesinin type özneliğinde tanımlanır. Her iki ekleme tipi de, araya ekleme öğesinin **width** ve **ignoreNull** özneliklerini kullanarak yerleşim düzenini özelleştirebilir. Örneğin:

```
<insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
```

Bu örnekte insert, /transaction/@ID adresindeki günlük iletisinde bulunan bilgileri alır ve günlüğe yazmadan önce kırıyor ya da 48 karaktere doldurur. /transaction/@ID içeriği boşsa, ignoreNull özneliği false olarak ayarlandığından, dizgiyi 48 karaktere doldurduktan sonra boş değer yazar. ignoreNull değeri true olarak ayarlanırsa, bunun yerine 48 karaktere doldurulmuş boş dizgi yazılır. Width="0" değeri, sütun genişliğinin kırılmadığı anlamına gelir; genişliğin 0 olarak kırıldığı anlamına gelmez. ignoreNull özneliği, beklenmediğinde boş değer bulunduğunda günlükte saptanması için bu şekilde kullanılabilir. Bu, yeni bir günlük dosyası tanımlamasında hata ayıklanırken yararlı olabilir.

## Kullanıcı tanımlı eklemeler

Kullanıcı ekleme işlemi, bu ekte yazılacak bilgiler için bir XPATH adresi içerir. Bu adres, FTE günlük iletisinde bulunan bir bilgi parçasını belirtir. Günlük iletisi biçimleriyle ilgili daha fazla bilgi için aşağıdaki konuya bakın:

- [Kütük aktarma günlüğü ileti biçimleri](#)
- [Zamanlanmış dosya aktarma günlüğü ileti biçimleri](#)
- [MFT monitör günlük iletisi biçimi](#)

## Sistem tarafından tanımlanan eklemeler

Sistem tarafından tanımlanan eklemeler, günlük iletisinde bulunamayan ya da XPATH dilini kullanarak tanımlanması kolay olmayan bir bilgi parçasına gönderme yapan bir anahtar sözcük içerir.

Desteklenen sistem eklemeleri şunlardır:

- type -Günlük iletisinin tipini kısa biçimde yazar.
- callArguments -Yönetilen bir çağrıya sağlanan bağımsız değişken kümesini boşlukla ayrılmış biçimde yazar.
- transferMetaData -Bir aktarma için tanımlanan meta veri girişleri kümesini virgülle ayrılmış *anahtar=değer* biçiminde yazar.

Aşağıdaki tabloda, her ileti tipi için sistem tarafından tanımlanan eklemeler için "type" değeri listelenmektedir.

Çizelge 50. Desteklenen ileti tiplerinin ve bunların "tip" sistem eklerinin özeti.	
İleti tipi	"type" sistem ekleme değeri
callCompleted	[ CCOM]
callStarted	[ CSTR]
monitorAction	[ EYLEM]
monitorCreate	[ MCRT]
monitorFired	[ MFIR]
notAuthorized	[ KIMLIK DEĞERİ]

Çizelge 50. Desteklenen ileti tiplerinin ve bunların "tip" sistem eklerinin özeti. (devamı var)

İleti tipi	"type" sistem ekleme değeri
scheduleDelete	[ SDEL]
scheduleExpire	[ SEXP]
scheduleSkipped	[ SSKP]
scheduleSubmitBilgileri	[ SSIN]
scheduleSubmitAktarma	[ SSTR]
scheduleSubmitTransferSet	[ SSTS]
transferStarted	[ TSTR]
transferCancelled	[ TCAN]
transferComplete	[ TCOM]
transferDelete	[ TDEL]
transferProgress	[ TPRO]

### İlgili başvurular

[MFT bağımsız dosya kaydedici varsayılan günlük biçimi](#)

[Bağımsız dosya kaydedici biçimi XSD](#)

[SYSTEM.FTE konusu](#)

[Dosya aktarma günlüğü ileti biçimleri](#)

[Zamanlanmış dosya aktarma günlüğü ileti biçimleri](#)

[MFT monitör günlük iletisi biçimi](#)

**ALW** MFT bağımsız dosya kaydedicisi ileti tiplerini kapsam dışı bırakma

Belirli bir ileti tipini dosya kaydedici çıkışından dışlamak istiyorsanız, boş ileti tipi öğeleri kullanabilirsiniz.

### Örnek

Örneğin, aşağıdaki biçim tanımlaması, dosya kaydedici tarafından çıkışa yazılan transferProgress iletilerini durdurur.

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferProgress></transferProgress>
  </messageTypes>
</logFormatDefinition>
```

**ALW** MFT bağımsız dosya kaydedicisi için özel biçimleri tanımlama

Günlük dosyası biçiminizi uyarlamak için gereken yapılandırma miktarını azaltmak için günlük biçimi tanımlamasında özel ileti tiplerinin bir alt kümesi tanımlanmalıdır.

## Bu görev hakkında

FileLoggerFormat.xml dosyasında bir messageType ögesi yoksa, o ileti tipinin biçimi varsayılan biçimi kullanır. Yalnızca varsayılandan farklı olmasını istediğiniz biçimleri belirtmeniz gerekir.

## Örnek


Bu örnekte, biçim tanımlaması transferStarted ileti tipine ilişkin varsayılan biçimi, yalnızca aktarımı başlatan kullanıcının çıkışını sağlayan azaltılmış sürümle değiştirir. Diğer tüm ileti tipleri, bu günlük biçimi tanımlamasında içerilmediği için varsayılan biçimi kullanır:

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
  xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferStarted>
      <format>
        <inserts>
          <insert type="user" width="19" ignoreNull="false">/transaction/action/
            @time</insert>
          <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6" ignoreNull="false">type</insert>
          <insert type="user" width="0" ignoreNull="true">/transaction/originator/
            userID</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </transferStarted>
  </messageTypes>
</logFormatDefinition>
```

## İlgili başvurular

[MFT bağımsız dosya kaydedici varsayılan günlük biçimi](#)

[Bağımsız dosya kaydedici biçimi XSD](#)

 *MFT bağımsız dosya kaydedicisinde yinelenen iletileri azaltma*

Bağımsız dosya kaydedicinin günlüğünde yinelenen günlük iletileri oluşabilir. logger.properties dosyasını kullanarak bağımsız dosya kaydedicisini ayarlayabilir ve yineleme sayısını azaltabilirsiniz.

## Dosya kaydedici günlüğünde yinelenen iletiler var

Bir hata durumunda, SYSTEM.FTE/Log# konusu IBM MQ için kesinleştiriliyor. Bu durumda, bağımsız dosya kaydedici yeniden başlatıldığında, aynı iletiyi ikinci kez alır ve günlük dosyasına yeniden yazar. Günlük dosyalarına el ile bakarken ya da bunları otomatik olarak işlerken bu yinelemelerin olasılığını işlemeyi planlayın. Yinelemelerin saptanmasına yardımcı olmak için, bağımsız dosya kaydedici, başlatılırken günlük dosyasına aşağıdaki iletiyi gönderir:

```
BFGDB0054I: The file logger has successfully started
```

Yinelemeler her zaman bağımsız dosya kaydedicinin başlangıç zamanı civarında gerçekleşir; bu, önceki eşgörünüm başarısız olmadan önce okunan son iletinin işlendiği zamandır. Yeni eşgörünümün ne zaman başladığını bilerek, yinelemelerin beklenip beklenmediğini ve bunların işlenmesi gerekip gerekmediğini saptayabilirsiniz.

## Yineleme sayısını azaltma

Bağımsız dosya kaydedici, başarıyı artırmak için işlediği günlük iletilerini hareketler halinde gruplar. Bu toplu iş boyutu, bir hata durumunda görebileceğiniz yinelenen ileti sayısı üst sınırdır. Yineleme sayısını azaltmak için logger.properties dosyasında aşağıdaki özelliği ayarlayabilirsiniz:

```
wmqfte.max.transaction.messages
```

Örneğin, bu değer 1 olarak ayarlandığında, yinelenen ileti sayısı üst sınırı 1'e düşürülür. Bu değer değiştirilmesinin, bağımsız dosya kaydedicinizin performansı üzerinde etkili olduğunu ve bunun sisteminizi olumsuz etkilemediğinden emin olmak için kapsamlı bir test gerektiğini unutmayın.

logger.properties dosyası, `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` dizininde bulunur. Kullanılabilir özellikler ve etkileri hakkında daha fazla bilgi için bkz. [MFT kaydedici yapılandırma özellikleri](#)

## **ALW** MFT bağımsız dosya kaydedicisi için kullanıcı erişimini yapılandırma

Bir test ortamında, normal kullanıcı hesabınıza yeni ayrıcalıklar ekleyebilirsiniz. Bir üretim ortamında, işi yapmak için gereken en düşük izinlere sahip yeni bir kullanıcı yaratmanız önerilir.

### **Bu görev hakkında**

Bağımsız dosya kaydediciyi ve IBM MQ dosyasını tek bir sisteme kurmanız gerekir. Kullanıcının izinlerini aşağıdaki gibi yapılandırın:

#### **Yordam**

1. Kullanıcının Managed File Transfer kuruluşunun bir parçası olarak kurulan dosyaları okuma ve yürütme iznine sahip olduğundan emin olun.
2. Kullanıcının yapılandırma dizinindeki logs dizinindeki herhangi bir dosyayı oluşturma ve bu dosyaya yazma izni olduğundan emin olun. Bu izin, bir olay günlüğü için ve gerekiyorsa tanılama izleme ve First Failure Data Capture (FFDC) dosyaları için kullanılır.
3. Kullanıcının kendi grubu olduğundan ve eşgüdüm kuyruk yöneticisinde geniş kapsamlı izinleri olan herhangi bir grupta olmadığından emin olun. Kullanıcı mqm grubunda olmamalıdır. Belirli platformlarda, personel grubuna otomatik olarak kuyruk yöneticisi erişimi verilir; bağımsız dosya kaydedici kullanıcı personel grubunda olmamalıdır. IBM MQ Explorer komutunu kullanarak, kuyruk yöneticisinin kendisine ve içindeki nesnelere ilişkin yetki kayıtlarını görüntüleyebilirsiniz. Nesneyi sağ tıklatın ve **Nesne Yetkileri > Yetki Kayıtlarını Yönet** seçeneklerini belirleyin. Komut satırında `dspmqaout` (görüntü yetkisi) ya da `dmpmqaut` (döküm yetkisi) komutlarını kullanabilirsiniz.
4. Kullanıcının kendi grubuna ilişkin yetkiler eklemek için IBM MQ Explorer ya da `setmqaut` (`grant or revoke authority`) komutundaki **Yetki Kayıtlarını Yönet** penceresini kullanın (AIX üzerinde, IBM MQ yetkileri tek tek kullanıcılarla değil, yalnızca gruplarla ilişkilendirilir). Gereken yetkiler şunlardır:
  - Kuyruk yöneticisine bağlanma ve sorma (IBM MQ Java kitaplıklarının çalışması için sorgu izni gerekir).
  - SYSTEM.FTE konusu.
  - SYSTEM.FTE.LOG.RJCT.günlük\_adı kuyruğu.
  - SYSTEM.FTE.LOG.CMD.günlük\_adı kuyruğu.

Belirtilen reddetme ve komut kuyruğu adları varsayılan adlardır. Bağımsız dosya kaydedici kuyruklarını yapılandırırken farklı kuyruk adları seçtiyseniz, izinleri bu kuyruk adlarına ekleyin.

### **MFT bağımsız veritabanı kaydedicisi kuruluyor**

Bağımsız veritabanı kaydedicisini kurmak ve yapılandırmak için bu adımları tamamlayın.

#### **Bu görev hakkında**

**Önemli:** Managed File Transfer kaydedicileri IBM i platformunda desteklenmez.

Bağımsız veritabanı kaydedicisi hakkında daha fazla bilgi için bkz. [“MFT günlük kaydedicisi yapılandırılması” sayfa 763.](#)

**Not:** Bir veritabanında aynı şema için aynı anda birden çok veritabanı kaydedicisi (bağımsız ya da Java EE) çalıştıramazsınız. Bunu yapma girişimi, aktarma günlüğü verilerini veritabanına yazma girişimi sırasında çakışmalara neden olur.

## Yordam

1. Veritabanınıza ilişkin belgeleri kullanarak veritabanı yazılımınızı kurun.

JDBC desteği, veritabanınız için isteğe bağlı bir bileşense, bu bileşeni kurmanız gerekir.

2. Bağımsız veritabanı kaydedicinizi yaratmak için **fteCreateLogger** komutunu **-loggerType** değiştirgesini DATABASE olarak ayarlayın. Daha fazla bilgi için bkz. [fteCreateLogger](#).

Varsayılan şema adı FTELOG 'tur. FTELOG dışında bir şema adı kullanırsanız, sonraki adıma geçmeden önce bu şema adını yansıtmak için veritabanınıza ( `ftelog_tables_db2.sql` ya da `ftelog_tables_oracle.sql`) uygun olan SQL dosyasını düzenlemeniz gerekir. Daha fazla bilgi için [MFT kaydedici yapılandırma özellikleri](#) içindeki `wmqfte.database.schema` konusuna bakın.

3. Veritabanınızın araçlarını kullanarak gerekli veritabanı tablolarını oluşturun.

### Multi

Çoklu platformlar sistemlerinde `ftelog_tables_db2.sql` ve `ftelog_tables_oracle.sql` kütükleri, çizelgeleri yaratmak için çalıştırabileceğiniz SQL komutlarını içerir.

### z/OS

z/OS' de çalıştırmanız gereken dosya, kullandığınız Db2 for z/OS sürümüne bağlıdır:

- Db2 for z/OS 9.0 ve öncesi için, çizelgeleri yaratmak üzere `ftelog_tables_zos.sql` dosyasını çalıştırın. Bu dosya, aktarılan dosyaların büyüklüklerini ve her aktarımla ilişkili çizelge tanıtıcısını gösteren alanlar için bir INTEGER veri tipi kullanarak çizelgeleri yaratır.
- Db2 for z/OS 9.1 ve sonraki sürümler için, çizelgeleri yaratmak üzere `ftelog_tables_zos_bigint.sql` dosyasını çalıştırın. Bu dosya, aktarılan dosyaların büyüklüklerini ve her aktarımla ilişkili çizelge tanıtıcısını gösteren alanlar için bir BIGINT veri tipi kullanarak çizelgeleri yaratır.

4. Günlüğe kaydedici kuyruklarını yaratmak için, **fteCreateLogger** komutuyla sağlanan MQSC komutlarını günlük kaydedici komut kuyruk yöneticinizle karşılaştırarak çalıştırın. Bağımsız veritabanı kaydedici, eşgüdüm kuyruğu yöneticisinde iki kuyruk kullanır. İlk kuyruk, bağımsız veritabanı kaydedicisinin işlemini denetleyen iletilerin yerleştirildiği bir komut kuyruğudur. Bu komut kuyruğunun varsayılan adı `SYSTEM.FTE.LOG.COMD.günlük_adi`. İkinci kuyruk bir reddetme kuyruğudur. Bağımsız veritabanı kaydedici günlük iletilerini hiçbir zaman atmadığı için, kaydedici işlemediği bir iletiyle karşılaşır, iletiyi inceleme ve olası yeniden işleme için reddetme kuyruğuna yerleştirir. Reddedilen iletiler DLH üstbilgisine sahip olmadığı ve reddedilen iletiler, başka nedenlerden ötürü gitmeyen iletiler kuyruğuna konan iletilerle birleştirilmemesi gerektiği için, bu amaçla kuyruk yöneticisinin gitmeyen iletiler kuyruğunu kullanmanız önerilmez. Reddetme kuyruğunun varsayılan adı `SYSTEM.FTE.LOG.RJCT.günlük_adi`. Bu iki kuyruk, **fteCreateLogger** komutu tarafından oluşturulan MQSC komut dosyalarında tanımlanır.

5. [Bir kullanıcı seçin ve izinleri yapılandırın](#)

6. İsteğe bağlı: “2” sayfa 772. adımda **fteCreateLogger** komutuyla yaratılan `logger.properties` dosyasını düzenleyerek, bağımsız veritabanı kaydedicisini daha fazla yapılandırabilirsiniz. Bu dosya, anahtar-değer çiftlerinden oluşan bir Java özellikler dosyasıdır. `logger.properties` dosyası, `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` dizininde bulunur. Kullanılabilir özellikler ve etkileri hakkında daha fazla bilgi için bkz. [MFT kaydedici yapılandırma özellikleri](#).

7. **Windows**

İsteğe bağlı: Windows sistemi kullanıyorsanız, bağımsız veritabanı günlük kaydedicisini Windows hizmeti olarak çalıştırabilirsiniz. **fteModifyLogger** komutunu **-s** parametresiyle çalıştırın. Daha fazla bilgi için bkz. [fteModifyLogger](#).

8. İsteğe bağlı: Kullanılmakta olan veritabanı Oracle ise ya da bir Db2 veritabanına uzaktan bağlanıyorsanız, kaydedicinin veritabanı sunucunuzla kimlik doğrulaması gerçekleştirmek için kullanacağı bir kullanıcı adı ve parola belirtmeniz gerekir. Bu kullanıcı adı ve parola, `MQMFTCredentials.xsd` şeması tarafından tanımlanan biçime uyan bir kimlik bilgileri dosyasında belirtilir. Daha fazla bilgi için bkz. [MFT kimlik bilgileri dosya biçimi](#). Kimlik bilgileri dosyasını oluşturduktan sonra, `wmqfte.database.credentials.file` özelliğini kullanarak `logger.properties` dosyasında kimlik bilgileri dosyasının konumunu belirtmeniz gerekir.



9. **fteStartLogger** komutunu kullanarak bağımsız veritabanı kaydedicisini başlatın. Varsayılan olarak, bağımsız veritabanı kaydedici artanda çalışır ve bağımsız veritabanı kaydedici çıkışı logs dizinindeki bir dosyaya yerleştirir. Bağımsız veritabanı kaydediciyi önalanda çalıştırmak ve konsola ve günlük dosyasına çıkış üretmek istiyorsanız, **fteStartLogger** komutuna **-F** parametresini ekleyin.

Önceki adımı gerçekleştirdiyse ve **fteModifyLogger** komutunu Windowsüzerinde **-s** parametresiyle kullandıysanız, bağımsız veritabanı kaydedicisi Windows hizmeti olarak başlar.

### İlgili görevler

“MFT bağımsız veritabanı kaydedicisi için kullanıcı erişiminin yapılandırılması” sayfa 773

Bir test ortamında, normal kullanıcı hesabınıza yeni ayrıcalıklar ekleyebilirsiniz. Bir üretim ortamında, işi yapmak için gereken en düşük izinlere sahip yeni bir kullanıcı yaratmanız önerilir.

### İlgili başvurular

[MFT kaydedici yapılandırma özellikleri](#)

[fteStartKaydedici](#)

[fteModifyKaydedici](#)

[MFT kaydedicisi için yetkiler](#)

### **MFT olanağının uzak veritabanıyla kullanılması**

Uzak sistemdeki bir veritabanıyla iletişim kurmak için Managed File Transfer kaydedicisi kullanabilirsiniz.

### **Bu görev hakkında**

Managed File Transfer ' in kurulu olduğu makineden farklı bir makinede kurulu bir veritabanınız varsa, aşağıdaki adımları tamamlayın. Aksi belirtilmedikçe, adımlar hem Db2 hem de Oracle için geçerlidir.

### **Yordam**

1. Managed File Transfer ürününü kurduğunuz sisteme bir veritabanı istemcisi kurun.
2. Uzak veritabanı sunucunuzu yerel veritabanı istemcisi yapılanışınıza ekleyin. Bu yapılandırma güncellemesi, Managed File Transfer ve IBM MQ ' in veritabanına doğru şekilde erişmesi için gereklidir.
3. Kimlik bilgileri dosyasını kullanarak veritabanına bağlanmak için `logger.properties` dosyasındaki yeni özellikleri belirtin: **wmfte.database.credentials.file**.

**Not:** Deprecated Managed File Transfer ' in önceki sürümleri **wmqfte.oracle.user** ya da **wmqfte.database.user** ve **wmqfte.oracle.password** ya da **wmqfte.database.password** özelliklerini kullanıyordu. Bu özellikler artık kullanımdan kaldırılmıştır. Bunun yerine **wmfte.database.credentials.file** kullanın.

4. **Oracle :** Veritabanıyla uzak bağlantıya izin vermek için, eşgüdüm kuyruğu yöneticisinin `qm.ini` dosyasındaki `XAResourceManager` kısmı değerini şu şekilde değiştirin (veritabanı adını, kullanıcı adını ve kullanıcı parolasını kendi bilgilerinizle eşleştirecek şekilde değiştirmenizi sağlar):

```
Oracle_XA+Acc=P/ftelog/  
qgw783jhT+SesTm=35+DB=FTEAUDIT1+SqlNet=FTEAUDIT1+threads=false,  
değişiklik kalın olarak vurgulanır.
```

5. **Oracle :** **wmqfte.oracle.host** ve **wmqfte.oracle.port** özelliklerini kullanarak `logger.properties` dosyasında bir anasistem ve kapı belirtin. Anasistem ve kapı için varsayılan değerler yerel bir veritabanı istemcisiyle çalışmanızı sağlar; daha önce yerel bir veritabanıyla çalıştıysanız, bu değerleri ayarlamamış olabilirsiniz.

### İlgili başvurular

[MFT kaydedici yapılandırma özellikleri](#)

### **MFT bağımsız veritabanı kaydedicisi için kullanıcı erişiminin yapılandırılması**

Bir test ortamında, normal kullanıcı hesabınıza yeni ayrıcalıklar ekleyebilirsiniz. Bir üretim ortamında, işi yapmak için gereken en düşük izinlere sahip yeni bir kullanıcı yaratmanız önerilir.

## Bu görev hakkında

Bağımsız veritabanı kaydediciyi çalıştırmak için gereken kullanıcı hesaplarının sayısı ve tipi, kullandığınız sistem sayısına bağlıdır. Bağımsız veritabanı kaydedicisini IBM MQ ve veritabanınızı tek bir sisteme ya da iki sisteme kurabilirsiniz. Bağımsız veritabanı kaydedicisi IBM MQ ile aynı sistemde olmalıdır. Bileşenler aşağıdaki topolojilere kurulabilir:

### Bağımsız veritabanı kaydedicisi, IBM MQ ve veritabanının tümü aynı sistemde

Üç bileşenle birlikte kullanılmak üzere tek bir işletim sistemi kullanıcısı tanımlayabilirsiniz. Bu, bağımsız veritabanı kaydedicisi için uygun bir yapılandırma değildir. Bağımsız veritabanı kaydedici, IBM MQ ' e bağlanmak için Bağlama kipini ve veritabanına bağlanmak için yerli bir bağlantıyı kullanır.

### Bağımsız veritabanı kaydedicisi ve IBM MQ tek bir sistemde, veritabanı ayrı bir sistemde

Bu yapılanış için iki kullanıcı yaratırsınız: Sistemde bağımsız veritabanı kaydedicisini çalıştıran bir işletim sistemi kullanıcısı ve veritabanı sunucusundaki veritabanına uzaktan erişimi olan bir işletim sistemi kullanıcısı. Bu, uzak veritabanı kullanan bağımsız veritabanı kaydedicisi için uygun bir yapılandırma değildir. Bağımsız veritabanı kaydedici, IBM MQ ' e bağlanmak için Bindings kipini ve veritabanına erişmek için bir istemci bağlantısını kullanır.

Örnek olarak, bu yönergelerin geri kalanı kullanıcının adının fte1logolduğunu varsayar, ancak herhangi bir kullanıcı adını kullanabilirsiniz. Kullanıcının izinlerini aşağıdaki gibi yapılandırın:

## Yordam

1. Kullanıcının Managed File Transfer Remote Tools and Documentation kuruluşunun bir parçası olarak kurulan dosyaları okuma ve yürütme iznine sahip olduğundan emin olun.
2. Kullanıcının Logs dizinindeki (yapılandırma dizininde) herhangi bir dosyayı oluşturma ve bu dosyaya yazma izni olduğundan emin olun. Bu izin, olay günlüğü için ve gerekiyorsa tanılama izleme ve FFDC dosyaları için kullanılır.
3. Kullanıcının kendi grubuna sahip olduğundan ve koordinasyon kuyruk yöneticisinde geniş kapsamlı izinlere sahip herhangi bir grupta olmadığından emin olun. Kullanıcı mqm grubunda olmamalıdır. Belirli platformlarda, personel grubuna otomatik olarak kuyruk yöneticisi erişimi verilir; bağımsız veritabanı kaydedici kullanıcısı personel grubunda olmamalıdır. IBM MQ Explorer komutunu kullanarak, kuyruk yöneticisinin kendisine ve içindeki nesnelere ilişkin yetki kayıtlarını görüntüleyebilirsiniz. Nesneyi sağ tıklatın ve **Nesne Yetkileri > Yetki Kayıtlarını Yönet** seçeneklerini belirleyin. Komut satırında `dspmqaout` (görüntü yetkisi) ya da `dmpmqaut` (döküm yetkisi) komutlarını kullanabilirsiniz.
4. Kullanıcının kendi grubuna ilişkin yetkiler eklemek için IBM MQ Explorer ya da `setmqaut` (`grant or revoke authority`) komutundaki **Yetki Kayıtlarını Yönet** penceresini kullanın ( AIX üzerinde, IBM MQ yetkileri tek tek kullanıcılarla değil, yalnızca gruplarla ilişkilendirilir). Gereken yetkiler şunlardır:
  - Kuyruk yöneticisine bağlanma ve sorma ( IBM MQ Java kitaplıklarının çalışması için sorgu izni gerekir).
  - SYSTEM.FTE konusu.
  - SYSTEM.FTE.LOG.RJCT.günlük\_adı kuyruğu.
  - SYSTEM.FTE.LOG.CMD.günlük\_adı kuyruğu.

Belirtilen reddetme ve komut kuyruğu adları varsayılan adlardır. Bağımsız veritabanı kaydedici kuyruklarını yapılandırırken farklı kuyruk adları seçtiyseniz, izinleri bu kuyruk adlarına ekleyin.

5. Kullanmakta olduğunuz veritabanına özgü kullanıcı yapılanışını gerçekleştirin.

- Veritabanınız Db2 ise, aşağıdaki adımları izleyin:

Db2 ile veritabanı kullanıcılarını yönetmek için çeşitli mekanizmalar vardır. Bu yönergeler, işletim sistemi kullanıcılarına dayalı olarak varsayılan şema için geçerlidir.

- fte1log kullanıcısının hiçbir Db2 denetim grubunda olmadığını doğrulayın (örneğin, db2iadm1, db2fadm1 ya da dasadm1).
- Kullanıcıya veritabanına bağlanma izni verin ve Adım 2: gerekli veritabanı tablolarını oluşturma ' nin bir parçası olarak oluşturduğunuz tabloları seçme, ekleme ve güncelleme izni verin.

- Veritabanınız Oracleise, aşağıdaki adımları izleyin:
  - fteLog kullanıcısının herhangi bir Oracle yönetim grubunda olmadığından emin olun (örneğin, Windows üzerinde ora\_dba ya da AIX and Linux üzerinde dba)
  - Kullanıcıya veritabanına bağlanma izni verin ve [Adım 2: gerekli veritabanı tablolarını oluşturma](#)'nin bir parçası olarak oluşturduğunuz tabloları seçme, ekleme ve güncelleme izni verin.

## MFT bağımsız günlük kaydedicisi için alternatif yapılandırmalar

İster bir dosya, ister veritabanı tipi olsun, genellikle Managed File Transfer bağımsız kaydedici, koordinasyon kuyruğu yöneticisiyle aynı sistemde bulunur ve IBM MQ bağ tanımları kipinde koordinasyon kuyruk yöneticisine bağlıdır. Ancak, eşgüdüm kuyruğu yöneticisine bağlantılılığı olan herhangi bir kuyruk yöneticisiyle aynı sisteme de kurulabilir. Bağımsız günlük kaydedici, bağımsız günlük kaydedicinin otomatik olarak yarattığı bir abonelik kullanarak iletileri alır. Bu, kuruluş yönergelerinde açıklanan yapılandırmadır.

Ancak, siteye özgü önemli noktalara sahipseniz, wmqfte.message.source.type özelliği tarafından denetlenen iletileri başka iki şekilde almak için bağımsız bir kaydedici yapılandırabilirsiniz. Bu özellik, [MFT kaydedici yapılandırma özelliklerinde](#) açıklanır.

## Yönetim aboneliği

Varsayılan olarak, bağımsız bir kaydedici SYSTEM.FTE/Log/# konusu, varsayılan sürekli abonelik seçeneklerini ve yönetilen aboneliği kullanarak (yani, kuyruk yöneticisi, uygulamaya geçirilmeden önce iletileri tutmak için kullanılan yedek kuyruğu denetler). Abonelikte ya da kuyrukta başka seçenekler gerekiyorsa, bunun yerine kendiniz bir abonelik oluşturabilir, gerek duyduğunuz seçenekleri ayarlayabilir ve bağımsız kaydediciyi o aboneliği kullanacak şekilde yapılandırabilirsiniz. Bağımsız günlük kaydedicinin oluşturduğunuz aboneliği kullanması için izin eklemeyi unutmayın.

Bu yapılanışın kullanılmasına bir örnek, günlük alanını iki genel arama karakteri aboneliğini kullanarak, adı FINANCE ile başlayan araçlardan bir veritabanına günlük ve MUHASEBE ile başlayan araçlardan diğerine günlük göndermektir. Bu yapılandırma tipi, her biri gerekli aboneliğe ve kendi komut kuyruğuna ve reddetme kuyruğuna gönderme yapan kendi logger.properties dosyasına sahip iki bağımsız kaydedici eşgörünümü gerektirir.

Yalnızca adları MUHASEBE ile başlayan araçlardan günlük iletileri toplamak için, koordinasyon kuyruğu yöneticinizde SYSTEM.FTE/Log/ACCOUNTING\*. **Genel arama karakteri kullanımı** değerini **Karakter düzeyi genel arama karakteri** olarak ayarlayın. Günlük kaydediciniz için logger.properties dosyasına girdiler de eklemeniz gerekir. Örneğin, ACCOUNTING.LOGS, logger.properties dosyasına aşağıdaki girdileri ekleyin:

```
wmqfte.message.source.type=administrative subscription
wmqfte.message.source.name=ACCOUNTING.LOGS
```

Bağımsız günlük kaydedici, SYSTEM.FTE/Log/. Daha kısıtlayıcı bir konu dizgisi belirtebilirsiniz, ancak daha az kısıtlayıcı bir dizgi belirtmezsiniz. Hata durumunda daha az kısıtlayıcı bir dizgi belirtirseniz, SYSTEM.FTE/Log/ Reddetme kuyruğuna gidin ve bağımsız kaydedici BFGDB0002Ehata iletisini üretir. Bu hata iletisi, bağımsız günlük kaydedici yapılandırmasıyla ilgili bir sorun olduğunu gösterir.

## Kuyruk

Tipik topoloji, bağımsız günlük kaydedicinin koordinasyon kuyruk yöneticisiyle aynı sistemde çalıştığı yerdir. Bu mümkün değilse, abonelik hedefi olarak başka bir kuyruk yöneticisindeki bir kuyruğu kullanarak (uzak kuyruk tanımlamasını kullanarak ya da aboneliğin DESTQMGR özelliğini kullanarak) eşgüdüm kuyruğu yöneticisinde bir abonelik yaratabilirsiniz. Kaydedici daha sonra ikinci kuyruk yöneticisini barındıran sistemde çalışabilir ve kuyruktan iletileri okuyabilir. Hareket bütünlüğünü sağlamak için, bağımsız günlük kaydedicinin her zaman bağ tanımlama kipinde kuyruk yöneticisine bağlanması gerekir. Bağımsız günlük kaydedicinin bağlandığı aynı kuyruk yöneticisinde reddetme kuyruğunu ve komut kuyruğunu tanımlamanız gerekir. Kuyruk yöneticileri IBM WebSphere MQ 7.5 ya da daha sonra olmalıdır.

Örneğin, USER.QUEUE ögesini abonelik ekleyin, bu girişleri `logger.properties` dosyasına ekleyin:

```
wmqfte.message.source.type=queue  
wmqfte.message.source.name=USER.QUEUE
```

## MFT için Java EE veritabanı kaydedicisi kuruluyor

JEE veritabanı kaydedicisini Managed File Transfer ile kullanmak üzere kurmak ve yapılandırmak için bu yönergeleri izleyin.

### Bu görev hakkında

Java EE veritabanı kaydedicisi hakkında daha fazla bilgi için bkz. [“MFT günlük kaydedicisi yapılandırılması” sayfa 763.](#)

**Not:** Bir Java EE veritabanı kaydedicisini, bağımsız bir kaydediciyle aynı anda çalıştıramazsınız; ancak, bu kaydediciler veritabanının ayrı eşgörunümlerini kullanırlar.

### Yordam

1. Java EE veritabanı kaydedicisi kurmadan önce ortamınızı hazırlamanız gerekir. [“MFT için Java EE veritabanı kaydedicisi kurulmaya hazırlanıyor” sayfa 776](#) başlıklı konudaki yönergeleri kullanın.
2. Java EE veritabanı kaydedicisi Java Platform, Enterprise Edition (Java EE) ya da Jakarta EE uyumlu bir uygulama sunucusuna kurun.  
Yönergeler için bkz. [“MFT için Java EE veritabanı kaydedicinin WebSphere Application Server traditional 9.0 ile kurulması” sayfa 779](#)

### İlgili görevler

[“MFT için Java EE veritabanı kaydedicisi kurulmaya hazırlanıyor” sayfa 776](#)

Java EE veritabanı kaydedicisi kurmadan önce Managed File Transfer ortamınızı hazırlamak için bu yönergeleri izleyin.

[“MFT için Java EE veritabanı kaydedicinin WebSphere Application Server traditional 9.0 ile kurulması” sayfa 779](#)

WebSphere Application Server traditional 9.0 ile Managed File Transfer için Java Platform, Enterprise Edition (Java EE) veritabanı kaydedicisini kurmak ve yapılandırmak için bu yönergeleri izleyin.

[“MFT için Java EE veritabanı kaydedicisi için kullanıcı erişiminin yapılandırılması” sayfa 784](#)  
Java Platform, Enterprise Edition (Java EE) veritabanı kaydedicisini Managed File Transfer için yapılandırdığınızda, IBM MQ' e, veritabanınıza ve işletim sisteminize erişmek için kullanıcı hesaplarına gereksinim duyarsınız. Gereken işletim sistemi kullanıcılarının sayısı, bu bileşenleri barındırmak için kullandığınız sistem sayısına bağlıdır.

[“Bağımsız veritabanı kaydedicisinden MFT için Java EE veritabanı kaydedicine geçiş” sayfa 785](#)

Bağımsız veritabanı kaydedicisinden Java EE veritabanı kaydedicine geçiş yapabilirsiniz. Bağımsız veritabanı kaydedicisi durdurmalı ve JEE veritabanı kaydedicisi kurmalısınız. Günlük girişlerini kaybetmemek ya da çoğaltmamak için, SYSTEM.FTE konusu ve Java EE veritabanı kaydedicisi kurduktan sonra yeniden başlatın. Geçişten önce veritabanınızı yedekleyin.

### İlgili başvurular

[MFT kaydedicisi için yetkiler](#)

## MFT için Java EE veritabanı kaydedicisi kurulmaya hazırlanıyor

Java EE veritabanı kaydedicisi kurmadan önce Managed File Transfer ortamınızı hazırlamak için bu yönergeleri izleyin.

### Bu görev hakkında

Java EE veritabanı kaydedicisi hakkında daha fazla bilgi için bkz. [“MFT günlük kaydedicisi yapılandırılması” sayfa 763.](#)

## Yordam

1. Veritabanınıza ilişkin belgeleri kullanarak veritabanı yazılımınızı kurun.  
JDBC desteği, veritabanınız için isteğe bağlı bir bileşense, bu bileşeni kurmanız gerekir.
2. Veritabanınız tarafından sağlanan araçları kullanarak bir veritabanı oluşturun. Veritabanının çizelge alanı ve arabellek havuzu sayfa büyüklüğü en az 8Kolmalıdır.  
Varsayılan şema adı FTELOG 'tur. FTELOG dışında bir şema adı kullanıyorsanız, sonraki adıma geçmeden önce bunu yansıtmak için veritabanınıza uygun olan `ftelog_tables_db2.sql` ya da `ftelog_tables_oracle.sql` dosyasını düzenlemeniz gerekir.  
**Not:** `ftelog_tables_db2.sql` ve `ftelog_tables_oracle.sql` dosyaları dosya yolunda `<MQ-installation-path>/mqft/sql`
3. Veritabanınızın araçlarını kullanarak gerekli veritabanı tablolarını oluşturun.

**Multi** Çoklu platformlarsistemlerinde `ftelog_tables_db2.sql` ve `ftelog_tables_oracle.sql` kütükleri, çizelgeleri yaratmak için çalıştırabileceğiniz SQL komutlarını içerir.

**z/OS** z/OS' de çalıştırmanız gereken dosya, kullandığınız Db2 for z/OS sürümüne bağlıdır:

- Db2 for z/OS 9.0 ve öncesi için, çizelgeleri yaratmak üzere `ftelog_tables_zos.sql` dosyasını çalıştırın. Bu dosya, aktarılan dosyaların büyüklüklerini ve her aktarımla ilişkili çizelge tanıtıcısını gösteren alanlar için bir INTEGER veri tipi kullanarak çizelgeleri yaratır.
  - Db2 for z/OS 9.1 ve sonraki sürümler için, çizelgeleri yaratmak üzere `ftelog_tables_zos_bigint.sql` dosyasını çalıştırın. Bu dosya, aktarılan dosyaların büyüklüklerini ve her aktarımla ilişkili çizelge tanıtıcısını gösteren alanlar için bir BIGINT veri tipi kullanarak çizelgeleri yaratır.
4. Şema adını FTELOG olarak değiştirdiyseniz, EAR dosyasındaki şema adını değiştirmeniz gerekir. Daha fazla bilgi için, bkz. [“MFT için Java EE veritabanı kaydedicisinde şema adını değiştirme” sayfa 777.](#)
  5. IBM MQ’inde bir reddetme kuyruğu oluşturun.  
Kaydedici günlük iletilerini hiçbir zaman atmadığı için, kaydedici işlemediği bir iletiyle karşılaşır, iletiyi inceleme ve olası yeniden işleme için reddetme kuyruğuna yerleştirir. Reddedilen iletiler DLH üstbilgisine sahip olmadığından ve reddedilen iletiler, başka nedenlerden ötürü gitmeyen iletiler kuyruğuna konan iletilerle birleştirilmemesi gerektiğinden, kuyruk yöneticisinin gitmeyen iletiler kuyruğunu bu amaçla kullanmayın. **fteCreateLogger** komutu bir reddetme kuyruğu yaratır. Bu reddetme kuyruğunun varsayılan adı `SYSTEM.FTE.LOG.RJCT.günlük_adi`
  6. [“MFT için Java EE veritabanı kaydedicisi için kullanıcı erişiminin yapılandırılması” sayfa 784](#) konusundaki yönergeleri izleyin.

## Sonraki adım

Java EE veritabanı kaydedicisi Java EE ya da Jakarta EE uyumlu bir uygulama sunucusuna kurun. [“MFT için Java EE veritabanı kaydedicinin WebSphere Application Server traditional 9.0 ile kurulması” sayfa 779](#) içindeki yönergeleri kullanın

## MFT için Java EE veritabanı kaydedicisinde şema adını değiştirme

Java Platform, Enterprise Edition (Java EE) veritabanı kaydedicisi, varsayılan şema adı olmayan bir veritabanını kullanabilir. Java EE veritabanı kaydedici EAR dosyasındaki şema adını değiştirmeniz gerekir.

## Bu görev hakkında

Java EE veritabanı kaydedicisinin kullandığı şemanın adını değiştirmek için aşağıdaki adımları izleyin:

## Yordam

1. Aşağıdaki komutu kullanarak EAR dosyasından JPA JAR dosyasını açın:

```
jar -xvf ear_file lib/jpa_file
```

Burada:

- *ear\_file* , Db2 ya da Oracle kullanıp kullanmadığınıza bağlı olarak `com.ibm.wmqfte.databaselogger.jee.oracle.ear` ya da `com.ibm.wmqfte.databaselogger.jee.ear` değeridir.
  - *jpa\_file* , Db2 ya da Oracle kullanıp kullanmadığınıza bağlı olarak `com.ibm.wmqfte.web.jpa.oracle.jar` ya da `com.ibm.wmqfte.web.jpa.jar` ' dir.
2. Aşağıdaki komutu kullanarak `persistence.xml` dosyasını JPA JAR dosyasından açın:

```
jar -xvf lib/jpa_file META_INF/persistence.xml
```

Burada:

- *jpa\_file* , Db2 ya da Oracle kullanıp kullanmadığınıza bağlı olarak `com.ibm.wmqfte.web.jpa.oracle.jar` ya da `com.ibm.wmqfte.web.jpa.jar` ' dir.
3. Aşağıdaki satırı değiştirmek için `persistence.xml` dosyasını düzenleyin:

```
<property name="openjpa.jdbc.Schema" value="schema_name" />
```

burada:

- *şema\_adi* , kullanmak istediğiniz şema adıdır.
4. Aşağıdaki komutu kullanarak JPA JAR dosyasını değiştirilen `persistence.xml` dosyasıyla güncelleyin:

```
jar -uvf lib/jpa_file META_INF/persistence.xml
```

Burada:

- *jpa\_file* , Db2 ya da Oracle kullanıp kullanmadığınıza bağlı olarak `com.ibm.wmqfte.web.jpa.oracle.jar` ya da `com.ibm.wmqfte.web.jpa.jar` ' dir.
5. Aşağıdaki komutu kullanarak EAR dosyasını değiştirilen JPA JAR dosyasıyla güncelleyin:

```
jar -uvf ear_file lib/jpa_file
```

Burada:

- *ear\_file* , Db2 ya da Oracle kullanıp kullanmadığınıza bağlı olarak `com.ibm.wmqfte.databaselogger.jee.oracle.ear` ya da `com.ibm.wmqfte.databaselogger.jee.ear` değeridir.
- *jpa\_file* , Db2 ya da Oracle kullanıp kullanmadığınıza bağlı olarak `com.ibm.wmqfte.web.jpa.oracle.jar` ya da `com.ibm.wmqfte.web.jpa.jar` ' dir.

## Sonraki adım

Java EE veritabanı kaydedicisini kurmak için değiştirilen EAR dosyasını kullanın.

### İlgili görevler

[“MFT için Java EE veritabanı kaydedicinin WebSphere Application Server traditional 9.0 ile kurulması” sayfa 779](#)

WebSphere Application Server traditional 9.0 ile Managed File Transfer için Java Platform, Enterprise Edition (Java EE) veritabanı kaydedicisini kurmak ve yapılandırmak için bu yönergeleri izleyin.

## **WebSphere Application Server traditional 9.0 içinde yerel kitaplık yolunu ayarlama**

Java Platform, Enterprise Edition (Java EE) veritabanı kaydedici uygulamasını WebSphere Application Server traditional 9.0 üzerinde konuşlandırılırsa ve uygulama ile IBM MQ arasındaki bağ tanımlama

kipi bağlantılarını kullanmak istiyorsanız, IBM MQ ileti alışverişi sağlayıcısını sistemdeki IBM MQ yerel kitaplıklarının yerile yapılandırmanız gerekir.

## Bu görev hakkında

Uygulama sunucunuzda yerel kitaplık yolunu ayarlamazsanız, WebSphere Application Server traditional 9.0 sistem çıkış günlüğünde aşağıdaki hata iletisini alabilirsiniz:

```
A connection could not be made to WebSphere MQ for the following reason:  
CC=2;RC=2495;AMQ8568: The native JNI library 'mqjbnb' was not found. [3=mqjbnb]
```

Aşağıdaki adımları tamamlamak için WebSphere Application Server traditional 9.0 yönetim konsolunu kullanın:

## Yordam

1. Gezinme bölmesinde **Resources > JMS > JMS Providers**(Kaynaklar JMS JMS Sağlayıcıları) seçeneğini genişletin.
2. Bağ tanımlama kipi bağlantısını yaratan bağlantı üreticisi ya da etkinleştirme belirtimi için doğru kapsamda olan IBM MQ ileti alışverişi sağlayıcısını seçin.

**Not:** Server kapsamındaki yerli yol bilgileri, daha yüksek kapsamlardaki yerli yol bilgileri ve Node kapsamındaki yerli yol bilgileri, Cell kapsamındaki yerel yol bilgileri için tercih edilir.

3. Genel Özellikler altında, **Yerel kitaplık yolu** alanına IBM MQ yerel kitaplıklarını içeren dizinin tam adını girin.

Örneğin, Linux yazın /opt/mqm/java/lib. Tek bir dizin adı girin.

4. **Tamam**'ı tıkkatın.

Yol ayarlandıktan sonra, değişikliklerin yürürlüğe girmesi için ana yapılandırmada yapılan değişiklikleri kaydetmeniz gerekir.

5. Yapılandırmayı yenilemek için uygulama sunucusunu yeniden başlatın.

6. Gerekli: Kitaplıkları yüklemek için uygulama sunucusunu ikinci kez yeniden başlatın.

## İlgili görevler

[“MFT için Java EE veritabanı kaydedicinin WebSphere Application Server traditional 9.0 ile kurulması” sayfa 779](#)

WebSphere Application Server traditional 9.0 ile Managed File Transfer için Java Platform, Enterprise Edition (Java EE) veritabanı kaydedicisini kurmak ve yapılandırmak için bu yönergeleri izleyin.

## **MFT için Java EE veritabanı kaydedicinin WebSphere Application Server traditional 9.0 ile kurulması**

WebSphere Application Server traditional 9.0 ile Managed File Transfer için Java Platform, Enterprise Edition (Java EE) veritabanı kaydedicisini kurmak ve yapılandırmak için bu yönergeleri izleyin.

## Başlamadan önce

JEE veritabanı kaydedici uygulamasını kurmadan önce [“MFT için Java EE veritabanı kaydedicisi kurulmaya hazırlanıyor” sayfa 776](#) ve [“WebSphere Application Server traditional 9.0 içinde yerel kitaplık yolunu ayarlama” sayfa 778](#) konularındaki yönergeleri izleyin.

## Bu görev hakkında


Java EE veritabanı kaydedicisi hakkında daha fazla bilgi için bkz. [“MFT günlük kaydedicisi yapılandırılması” sayfa 763](#).

## Yordam

1. XA JDBC sağlayıcısını ayarlayın:



- a) WebSphere Application Server traditional 9.0 yönetim konsolu gezinme bölmesinden **Resources > JDBC > JDBC Providers** (Kaynaklar JDBC) seçeneğini belirleyin.
  - b) **Yenidüğmesini** tıklatarak konsol sihirbazını kullanarak bir JDBC sağlayıcısı oluşturun.
  - c) Sihirbazın 1. Adımında, **Veritabanı tipi** listesinden kullanmakta olduğunuz veritabanını ve **Sağlayıcı tipi** listesinden ilişkili sağlayıcı tipini seçin. **Somutlama tipi** listesinde **XA veri kaynağı** öğesini seçin. **İleri**'yi tıklatın.
 

 db2jcc\_license\_cisuz.jar ile ilgili bir başvuruyu kaldırabilir ve db2jcc.jar dosyasını db2jcc4.jar olarak, yani en son Db2 sürümüyle ya da yerel sürümünüzle birlikte gönderilen jar dosyasının sürümünü değiştirebilirsiniz.
  - d) Sihirbazın 2. Adımında, gerekli veritabanı jar dosyalarının izin konumunun doğru şekilde ayarlandığından emin olun. **İleri**'yi tıklatın.
  - e) JDBC sağlayıcısını yaratmak için özet sayfasında **Son** düğmesini tıklatın.
2. Kimlik doğrulama diğer adları oluşturun. Veri kaynağı için bir diğer ad, IBM MQ için başka bir diğer ad oluşturursanız:
- a) WebSphere Application Server traditional 9.0 yönetim konsolu gezinme bölmesinde **Security (Güvenlik) > Global security** (Genel güvenlik) seçeneklerini belirleyin.
  - b) **Authentication** (Kimlik Doğrulama) başlığı altında **Java Authentication and Authorization Service** (Java Kimlik Doğrulaması ve Yetkilendirme Hizmeti) öğesini genişletin.
  - c) **J2C kimlik doğrulama verileri** öğesini tıklatın. Kimlik doğrulama diğer adı sayfası açılır.
  - d) Veri kaynağınız için bir kimlik doğrulama diğer adı oluşturun:
    - i) **Yeni**'yi tıklatın.
    - ii) **Diğer Ad, Kullanıcı Kimliği, Parola ve Açıklama** için ayrıntıları girin. **Kullanıcı Kimliği** ve **Parola** alanlarına girilen ayrıntılar, veritabanı kullanıcılarını oluşturduğunuzda girdiğiniz ayrıntılarla eşleşmelidir. Daha fazla bilgi için, bkz. ["MFT için Java EE veritabanı kaydedicisi için kullanıcı erişiminin yapılandırılması"](#) sayfa 784.
    - iii) **Tamam**'i tıklatın.
  - e) IBM MQ için bir kimlik doğrulama diğer adı oluşturun:
    - i) **Yeni**'yi tıklatın.
    - ii) **Diğer Ad, Kullanıcı Kimliği, Parola ve Açıklama** için ayrıntıları girin. **Kullanıcı Kimliği** ve **Parola** alanlarına girilen ayrıntılar, IBM MQ kuruluşunuza ilişkin kullanıcı ve parola ayarlarıyla eşleşmelidir.
    - iii) **Tamam**'i tıklatın.
3. Bir veri kaynağı oluşturun:
- a) WebSphere Application Server traditional 9.0 yönetim konsolu gezinme bölmesinden **Resources > JDBC > Data resources** (Kaynaklar JDBC Veri kaynakları) seçeneğini belirleyin.
  - b) **Kapsam** açılan listesini seçin ve kapsamı uygun değerle değiştirin. Örneğin, Node=yourNode , Server=yourServer.
  - c) **Yenidüğmesini** tıklatarak konsol sihirbazını kullanarak bir veri kaynağı oluşturun.
  - d) Sihirbazın 1. Adımında, **Veri kaynağı adı** alanına wmqfte-database yazın ve **JNDI adı** alanına jdbc/wmqfte-database girin. **İleri**'yi tıklatın.
  - e) Sihirbazın 2. Adımında, önceki adımlarda yaratılan JDBC sağlayıcısını seçmek için **Varolan bir JDBC sağlayıcısını seç** açılan listesini kullanın. **İleri**'yi tıklatın.
  - f) **Db2:** Sihirbazın 3. adımında, **Driver type** (Sürücü tipi) alanına 4 yazın.
  - g) **Db2: Veritabanı adı, Sunucu adı ve Kapı numarası** alanlarına ayrıntıları girin ve **İleri** düğmesini tıklatın.
 

**Oracle: URL** alanında URL bağlantısını girin ve **Veri deposu yardımcısı sınıf adı** alanında doğru veri deposu yardımcısını seçin.



- Oracle RAC:** Bir Oracle Real Application Cluster ürününe bağlanırken URL bağlantısı, veritabanının tüm kullanılabilir yönetim ortamlarına bağlanmak için gereken anasistem bilgilerini içermelidir.
- h) Sihirbazın 4. Adımında, **XA kurtarma için kimlik doğrulama diğer adı** listesinden 2d adımda tanımladığınız veri kaynağı kimlik doğrulama diğer adının adını seçin. **Component-managed authentication alias** (Bileşen tarafından yönetilen kimlik doğrulama diğer adı) ve **Container-managed authentication alias** (Taşıyıcı tarafından yönetilen kimlik doğrulama diğer adı) listelerinden aynı adı seçin.
- i) Veri kaynağını oluşturmak için özet sayfasında **Son** düğmesini tıklayın.
4. İsteğe bağlı: Veri kaynağının yapılandırmasını doğrulayın:
- a) WebSphere Application Server traditional 9.0 yönetim konsolu gezinme bölmesinden **Resources > JDBC > Data resources** (Kaynaklar JDBC Veri kaynakları) seçeneğini belirleyin.
- b) **Bağlantıyı Sına** düğmesini tıklayın.
5. Bir konu oluşturun.
- a) WebSphere Application Server traditional 9.0 yönetim konsolu gezinme bölümünde **Kaynaklar > JMS > Konular** seçeneklerini tıklayın.
- b) **Kapsam** açılan listesini seçin ve kapsamı uygun değerle değiştirin. Örneğin, Node=yourNode , Server=yourServer.
- c) **Yeni**'yi tıklayın.
- d) **IBM MQ ileti alışverişi sağlayıcısı** öğesini tıklayın.
- e) Konuya ilişkin özellik sayfasının **Denetim** panosunda, yapılandırmada daha sonra başvuracağınız **Ad** ve **JNDI adı** alanları için benzersiz değerler seçin.
- f) **IBM MQ konu** panosuna **Konu adı** alanına SYSTEM.FTE/Log/# yazın.
6. Etkinleştirme belirtimi yarat:
- a) WebSphere Application Server traditional 9.0 yönetim konsolu gezinme bölümünden **Kaynaklar > JMS > Etkinleştirme belirtileri** seçeneklerini tıklayın.
- b) **Kapsam** açılan listesini seçin ve kapsamı uygun değerle değiştirin. Örneğin, Node=yourNode , Server=yourServer.
- c) **Yeni**'yi tıklayın.
- d) **IBM MQ ileti alışverişi sağlayıcısı** öğesini tıklayın.
- e) Sihirbazın 1. Adımında, yapılandırmada daha sonra yeniden başvuracağınız **Ad** ve **JNDI adı** alanları için benzersiz değerler seçin.
- f) Adım 1.1' de, **Hedef JNDI adı** alanında 5. adımda ayarladığınız konuya ilişkin JNDI adını girin.
- g) **Destination type** (Hedef tipi) listesinden **Topic**(Konu) seçeneğini belirleyin.
- h) Sihirbazın Adım 1.2 ' de **Dayanıklı Abonelik** seçeneğini belirleyin. **Abonelik adı** alanına SYSTEM.FTE.DATABASELOGGER.AUTO yazın.
- i) Sihirbazın 2. Adımında **Bu sihirbaza gerekli tüm bilgileri girin** seçeneğini belirleyin.
- j) Adım 2.1' de, **Kuyruk yöneticisi ya da kuyruk paylaşım grubu adı** alanına kuyruk yöneticisi adınızı girin.
- k) Adım 2.2' de, **İletim** listesinden seçtiğiniz iletim yöntemini seçin. **Bağ Tanımları** seçeneğini belirlerseniz, başka bilgi gerekmez. **Client** ya da **Bindings then client** seçeneğini belirlerseniz, **Anasistem adı, Kapı ve Sunucu bağlantı kanalı** için ayrıntıları girin.
- l) İsteğe bağlı: Kuyruk yöneticisinin var olduğunu doğrulamak için **Bağlantıyı Sına** düğmesini tıklayın. Ancak, 6n. adımda kimlik doğrulama diğer adına başvuruncaya kadar NOT\_AUTHORIZED alabilirsiniz.
- m) **Kaydet** öğesini tıklayın.
- n) Oluşturduğunuz Etkinleştirme Belirtiminin adını tıklayın. **Configuration** (Yapılandırma) sekmesinin **General Properties** (Genel Özellikler) bölümünde, **Advanced** (Gelişmiş) panosuna gidin ve **Client ID** (İstemci Tanıtıcısı) alanında IBM MQ bağlantınızı tanımlamak için benzersiz bir ad girin. Bu

adımı tamamlamanız gerekir ya da bağlantınız IBM MQ tarafından JMSSC0101 hata koduyla reddedilir.

- o) İletim yönteminiz olarak **İstemci** ' yi seçtiyseniz, **Güvenlik Ayarları** panosuna gidin ve **Kimlik doğrulama diğer adı** listesinden 8. adımda tanımladığınız kimlik doğrulama diğer adını seçin.
- p) **Uygula**'yı tıklatın.
- q) **Yapılandırma** sekmesinin **Ek Özellikler** bölümünde **Gelişmiş Özellikler** seçeneğini tıklatın. **Gelişmiş Özellikler** panosunun **Bağlantı Tüketicisi** bölümüne, **Sunucu oturumu sayısı üst sınırı** alanına 1 girin.

**Not:** Devam etmeden önce bu adımı tamamladığınızdan emin olun. Bunun yapılmaması, günlüğe kaydedicinin düzgün çalışmamasına neden olabilir.

- r) **Yapılandırma** sekmesinin **Ek Özellikler** bölümünde **Gelişmiş Özellikler** seçeneğini tıklatın. **İleti teslimi başarısız olursa uç noktayı durdur** değerini en az 1 olarak ayarlayın.

**\_numberOfFailedAttemptsBeforeReject** özelliğinin değeri 1 'den fazlaysa (daha fazla bilgi için bkz. 9j), **İleti teslimi başarısız olursa uç noktayı durdur** seçeneğini

**\_numberOfFailedAttemptsBeforeReject** özelliğinin değerine ayarlayın. Bu, işlenemeyen bir ileti (örneğin, bozuk biçimli bir aktarma günlüğü iletisi) alındığında uç noktanın durmasını önler. Daha fazla bilgi için bkz. [MFT kaydedici hata işleme ve reddetme](#).

7. Bir kuyruk bağlantısı üreticisi yaratın.

- a) WebSphere Application Server traditional 9.0 yönetim konsolu gezinme bölümünde **Kaynaklar > JMS > Kuyruk bağlantısı üreticileri** seçeneklerini tıklatın.
- b) **Kapsam** açılan listesini seçin ve kapsamı uygun değerle değiştirin. Örneğin, Node=yourNode , Server=yourServer.
- c) **Yeni**'yi tıklatın.
- d) **IBM MQ ileti alışverişi sağlayıcısı** öğesini tıklatın.
- e) Sihirbazın 1. Adımında, yapılandırmada daha sonra yeniden başvuracağınız **Ad** ve **JNDI adı** alanları için benzersiz değerler seçin.
- f) Adım 2 'de **Bu sihirbaza gerekli tüm bilgileri girin** seçeneğini belirleyin.
- g) Adım 2.1' de, **Kuyruk yöneticisi ya da kuyruk paylaşım grubu adı** alanına kuyruk yöneticisi adınızı girin.
- h) Adım 2.2' de, **İletim** listesinden seçtiğiniz iletim yöntemini seçin. **Bağ Tanımları** seçeneğini belirlerseniz, başka bilgi gerekmez. **Client** ya da **Bindings then client** seçeneğini belirlerseniz, **Anasistem adı**, **Kapı** ve **Sunucu bağlantı kanalı** için ayrıntıları girin.
- i) İsteğe bağlı: Kuyruk yöneticisinin var olduğunu doğrulamak için **Bağlantıyı Sına** düğmesini tıklatın. Ancak, 7h numaralı adımda kimlik doğrulama diğer adına başvuruncaya kadar NOT\_AUTHORIZED alabilirsiniz.
- j) İletim yönteminiz olarak **İstemci** ya da **Bağ Tanımları** öğesini seçtiyseniz, az önce yarattığınız kuyruk bağlantısı üreticisinin adını tıklatın. **Configuration** (Yapılandırma) sekmesinin **Security Settings** (Güvenlik Ayarları) panosuna gidin ve **Authentication alias for XA recovery** (XA kurtarma için kimlik doğrulama diğer adı) ve **Container tarafından yönetilen kimlik doğrulama diğer adı** listelerinden 2e adımında tanımladığınız kimlik doğrulama diğer adını seçin.

8. WebSphere Application Serverinde bir reddetme kuyruğu oluşturun:

- a) WebSphere Application Server traditional 9.0 yönetim konsolu gezinme bölümünde **Kaynaklar > JMS > Kuyrukları** seçeneklerini tıklatın.
- b) **Kapsam** açılan listesini seçin ve kapsamı uygun değerle değiştirin. Örneğin, Node=yourNode , Server=yourServer.
- c) **Yeni**'yi tıklatın.
- d) **IBM MQ ileti alışverişi sağlayıcısı** öğesini tıklatın.
- e) Daha sonra yapılandırmada yeniden başvuracağınız **Ad** ve **JNDI adı** alanları için benzersiz değerler seçin.

- f) **Kuyruk adı** alanına SYSTEM.FTE.LOG.RJCT.*logger\_name* girin. Bu kuyruğu eşgüdüm kuyruk yöneticinizde yarattığınızdan emin olun.
- g) **Kuyruk yöneticisi adı** alanına kuyruk yöneticisi adınızı girin.
- h) **Tamam'**ı tıklatın.

9. JEE veritabanı kaydedici uygulamasını kurun:

- a) WebSphere Application Server traditional 9.0 yönetim konsolunda **Uygulamalar > Yeni Uygulamaseçeneği**ni belirleyin.
- b) **Kapsam** açılan listesini seçin ve kapsamı uygun değerle değiştirin. Örneğin, Node=yourNode, Server=yourServer.
- c) Seçenekler listesinden **Yeni Kurumsal Uygulamaseçeneği**ni belirleyin.
- d) **Uygulama kuruluşu için hazırlık** sayfasında, Managed File Transfer Service kuruluşunun *MQ\_INSTALLATION\_PATH/mqft/web* dizininden *com.ibm.wmqfte.databaselogger.jee.ear* dosyasını ya da *com.ibm.wmqfte.databaselogger.jee.oracle.ear* dosyasını seçin ve **İleri**düğmesini tıklatın.
- e) Aşağıdaki ekranda, tüm kuruluş seçeneklerini ve değiştirgelerini göstermek için **Detailed** (Ayrıntılı) seçeneğini belirleyin ve **Next**(İleri) düğmesini tıklatın.
- f) Varsayılan değerleri kabul etmek için 1-4 arasındaki sihirbaz adımlarıyla **Next** (İleri) düğmesini tıklatın.
- g) Sihirbazın 5. adımında **İletiyile yönlendirilen Bean 'ler için Bind dinleyicileri Dinleyici Bağları** bölümüne gidin. **Etkinleştirme Belirtimi'** ni tıklatın.  
Aşağıdaki alanlar için gerekli değerleri girin:

**Hedef Kaynak JNDI adı**

6d. adımda bir etkinleştirme belirtimi yaratırken belirttiğiniz JNDI adı.

**Hedef JNDI adı**

5d. adımda konu yaratırken belirttiğiniz JNDI adı.

**İleri'**yi tıklatın.

- h) Sihirbazın 6. adımında **Kaynak başvurularını kaynaklarla eşle, Hedef Kaynak JNDI adı** alanına ayrıntıları girin. Bu ad, 7c. adımda kuyruk bağlantısını reddetme üreticisi için belirttiğiniz JNDI adıdır. **İleri'**yi tıklatın.
- i) Sihirbazın 7. adımında **Kaynak ortamı girişi başvurularını kaynaklarla eşle, Hedef Kaynak JNDI adı** alanına ayrıntıları girin. Bu ad, 8d. adımda yarattığınız reddetme kuyruğunun JNDI adıdır. **İleri'**yi tıklatın.
- j) Sihirbazın 8. adımında **EJB modülleri için ortam girişlerini eşle, 1** varsayılan değerini kabul edin. **İleri'**yi tıklatın.

**Oracle RAC:** Bir Oracle Real Application Cluster ürününe bağlanırken, *\_numberOfFailedAttemptsBeforeReject* özelliğinin değerini **en az 2** olarak ayarlamamız gerekir. Bu özellik, bir hata oluşuktan sonra günlük kaydedicinin bir denetim iletisini kaç kez işlemeye çalışacağını belirler. Bir veritabanı yedeği devreye sokma durumunda en az bir hata oluşma olasılığı vardır. Bir iletinin gereksiz yere reddetme kuyruğuna taşınmasını önlemek için, bu değer artırılması ikinci bir girişimin yapılmasına izin verir; bu da genellikle yeni veritabanı örneğiyle bağlantı kurulmasıyla başarıya ulaşılmasına neden olur. Veritabanı örneğinizin yedeği devreye sokma sırasında iletilerin yine de reddetme kuyruğuna taşındığını bulursanız, bu değeri daha da artırın: Eşgörünüm arasındaki anahtarın zamanlaması aynı ileti için birden çok hataya neden olabilir. Ancak, bu değeri artırmanın yalnızca veritabanı yedeği devreye sokma değil, tüm başarısızlık durumlarını (örneğin, bozuk biçimli bir ileti) etkilediğini unutmayın; bu nedenle gereksiz yeniden denemeleri önlemek için değeri dikkatli bir şekilde artırın.

- k) Sihirbazın 9. adımında **Modüller için meta veriler, İleri'** yi tıklatın.

l) Sihirbazın 10. adımında **Özet, Sondüğmesini** tıklatın.

10. Artık uygulamayı WebSphere Application Server traditional 9.0 yönetim konsolundan başlatabilirsiniz:

- a) Konsol gezinme bölmesinden **Applications > Application Types > WebSphere enterprise applications** (Uygulamalar > Uygulama Tipleri) seçeneklerini belirleyin.
- b) Kaynak grubu tablosundan **Kaydedici** kurumsal uygulamasına ilişkin onay kutusunu seçin ve **Başlat** 'ı tıklatın.

### **MFT için Java EE veritabanı kaydedicisi için kullanıcı erişiminin yapılandırılması**

Java Platform, Enterprise Edition (Java EE) veritabanı kaydedicisini Managed File Transfer için yapılandırdığınızda, IBM MQ' e, veritabanınıza ve işletim sisteminize erişmek için kullanıcı hesaplarına gereksinim duyarsınız. Gereken işletim sistemi kullanıcılarının sayısı, bu bileşenleri barındırmak için kullandığınız sistem sayısına bağlıdır.

### **Bu görev hakkında**

Java EE veritabanı kaydediciyi çalıştırmak için gereken kullanıcı hesaplarının sayısı ve tipi, kullandığınız sistem sayısına bağlıdır. Kullanıcı hesaplarının aşağıdaki üç ortama erişmesi gerekir:

- Yerel işletim sistemi
- IBM MQ
- Veritabanı

JEE veritabanı kaydedicisi IBM MQ ve veritabanınızı tek bir sisteme ya da birden çok sisteme kurabilirsiniz. Bileşenler aşağıdaki örnek topolojilere kurulabilir:

#### **Java EE veritabanı kaydedici, IBM MQ ve veritabanının tümü aynı sistemde**

Üç bileşenle birlikte kullanılmak üzere tek bir işletim sistemi kullanıcısı tanımlayabilirsiniz. Kaydedici, IBM MQ ' e bağlanmak için Bağlama kipini ve veritabanına bağlanmak için yerli bir bağlantıyı kullanır.

#### **Java EE veritabanı kaydedici ve IBM MQ tek bir sistemde, veritabanı ayrı bir sistemde**

Bu yapılanış için iki kullanıcı yaratırsınız: Sistemde günlük kaydediciyi çalıştıran bir işletim sistemi kullanıcısı ve veritabanı sunucusundaki veritabanına uzaktan erişimi olan bir işletim sistemi kullanıcısı. Kaydedici, IBM MQ ' e bağlanmak için Bağlama kipini ve veritabanına erişmek için bir istemci bağlantısını kullanır.

#### **Java EE veritabanı kaydedicisi bir sistemde, IBM MQ başka bir sistemde, veritabanı başka bir sistemde**

Bu yapılanış için üç kullanıcı yaratırsınız: Uygulama sunucusunu başlatacak bir işletim sistemi kullanıcısı, kullanılmakta olan kuyruklara ve konulara erişecek bir IBM MQ kullanıcısı ve veritabanı çizerlerine erişecek ve eklenecek bir veritabanı sunucusu kullanıcısı. Günlüğe kaydedici, IBM MQ ' e erişmek için İstemci kipini ve veritabanına erişmek için bir istemci bağlantısını kullanır.

Örnek olarak, bu yönergelerin geri kalanı kullanıcının adının `ftelogo` olduğunu varsayar, ancak yeni ya da var olan herhangi bir kullanıcı adını kullanabilirsiniz. Kullanıcı izinlerini aşağıdaki gibi yapılandırın:

### **Yordam**

1. İşletim sistemi kullanıcısının kendi grubuna sahip olduğundan ve koordinasyon kuyruk yöneticisinde geniş kapsamlı izinlere sahip herhangi bir grupta olmadığından emin olun. Kullanıcı `mqm` grubunda olmamalıdır. Belirli platformlarda, personel grubuna otomatik olarak kuyruk yöneticisi erişimi verilir; kaydedici kullanıcı personel grubunda olmamalıdır. IBM MQ Explorer komutunu kullanarak, kuyruk yöneticisinin kendisine ve içindeki nesnelere ilişkin yetki kayıtlarını görüntüleyebilirsiniz. Nesneyi sağ tıklayın ve **Nesne Yetkileri > Yetki Kayıtlarını Yönet** seçeneklerini belirleyin. Komut satırında `dspmqaout` (görüntü yetkisi) ya da `dmpmqaut` (döküm yetkisi) komutlarını kullanabilirsiniz.
2. IBM MQ kullanıcısının kendi grubuna ilişkin yetkiler eklemek için IBM MQ Explorer ya da `setmqaut` (`grant or revoke authority`) komutundaki **Yetki Kayıtlarını Yönet** penceresini kullanın ( AIX üzerinde, IBM MQ yetkileri tek tek kullanıcılarla değil, yalnızca gruplarla ilişkilendirilir). Gereken yetkiler şunlardır:
  - Kuyruk yöneticisinde CONNECT ve INQUIRE ( IBM MQ Java kitaplıklarının çalışması için INQUIRE izni gerekir).
  - SYSTEM.FTE konusuna ilişkin ABONE OL izni.

- SYSTEM.FTE.LOG.RJCT.günlük\_adi kuyruğunda PUT izni.

Belirtilen reddetme ve komut kuyruğu adları varsayılan adlardır. Günlük kaydedici kuyruklarını yapılandırırken farklı kuyruk adları seçtiyseniz, izinleri bu kuyruk adlarına ekleyin.

### 3. Kullanmakta olduğunuz veritabanına özgü veritabanı kullanıcı yapılandırmasını gerçekleştirin.

- Veritabanınız Db2ise, aşağıdaki adımları izleyin:

**Not:** Db2ile veritabanı kullanıcılarını yönetmek için çeşitli mekanizmalar vardır. Bu yönergeler, işletim sistemi kullanıcılarına dayalı olarak varsayılan şema için geçerlidir.

- fte1log kullanıcısının hiçbir Db2 denetim grubunda olmadığını doğrulayın (örneğin, db2iadm1, db2fadm1ya da dasadm1).
- Kullanıcıya veritabanına bağlanma izni ve Adım 2: gerekli veritabanı tablolarını oluşturma' nin bir parçası olarak oluşturduğunuz tabloları seçme, ekleme ve güncelleme izni verin.

- Veritabanınız Oracleise, aşağıdaki adımları izleyin:

- fte1log kullanıcısının herhangi bir Oracle yönetim grubunda olmadığından emin olun (örneğin, Windows üzerinde ora\_dba ya da AIX and Linuxüzerinde dba).
- Kullanıcıya veritabanına bağlanma izni ve Adım 2: gerekli veritabanı tablolarını oluşturma' nin bir parçası olarak oluşturduğunuz tabloları seçme, ekleme ve güncelleme izni verin.

## Bağımsız veritabanı kaydedicisinden MFT için Java EE veritabanı kaydedicine geçiş

Bağımsız veritabanı kaydedicisinden Java EE veritabanı kaydedicine geçiş yapabilirsiniz. Bağımsız veritabanı kaydediciyi durdurmalı ve JEE veritabanı kaydediciyi kurmalısınız. Günlük girişlerini kaybetmemek ya da çoğaltmamak için, SYSTEM.FTE konusu ve Java EE veritabanı kaydediciyi kurduktan sonra yeniden başlatın. Geçişten önce veritabanınızı yedekleyin.

### Bu görev hakkında

#### Yordam


1. Veritabanını durdurmadan önce, eşgüdüm kuyruğu yöneticiniz için şu MQSC komutunu çalıştırın:  
ALTER QM PSMODE(COMPAT)

Bu, SYSTEM.FTE/Log konusu. Kaydedici, abonelikteki tüm iletileri işleinceye kadar bekleyin. Varsayılan olarak bu abonelik SYSTEM.FTE.LOGGER.AUTO(OTOMATIK).

2. **fteStopLogger** komutunu kullanarak veritabanı kaydedicisini durdurun.
3. Veritabanı yazılımıyla birlikte sağlanan araçları kullanarak veritabanını yedekleyin.
4. Bağımsız veritabanı kaydedicisine ait aboneliği silin.

Varsayılan olarak bu abonelik SYSTEM.FTE.LOGGER.AUTO(OTOMATIK).

5. Veritabanı şemanız daha önceki bir sürümdeyse, şemayı sonraki her düzeye sırayla geçirmeniz gerekir. Örneğin, veritabanı şemanız V7.0.1 ise ve V7.0.4düzeyine geçiyorsanız, şemanızı V7.0.1 sürümünden V7.0.2sürümüne, ardından V7.0.2 sürümünden V7.0.3sürümüne ve daha sonra V7.0.3 sürümünden V7.0.4sürümüne geçirmeniz gerekir. Veritabanı şemanızı *old* sürümünden *newsürümüne* geçirin; burada *old* ve *new*, geçmeniz gereken şemanın her sürümü için aşağıdaki işlemlerden birini gerçekleştirerek, şema sürümünü tanımlayan değişkenlerdir:

-  Veritabanınız Db2 on z/OS ise ve V7.0.2 ile V7.0.3 şemaları arasında ya da V7.0.3 ile V7.0.4 şemaları arasında geçiş yapıyorsanız, yeni bir veritabanı şeması oluşturmanız ve var olan verilerinizi bu şemaya kopyalamanız gerekir. Daha fazla bilgi için Db2 belgelerine bakın.
- Veritabanınız Db2 değilse ya da veritabanınızı 8Ksayfa büyüklüğünden daha büyük bir sayfa boyutuyla yarattıysanız, aşağıdaki adımları tamamlayarak şemayı diğer sürümlerle aynı şekilde geçirebilirsiniz.
- Başka bir koşulda veritabanı çizelgeleri arasında geçiş yapıyorsanız aşağıdaki adımları izleyin:

- a. Veritabanı altyapınıza uygun olan ve *old-new* dizgisini içeren bir adı olan dosyayı seçin. Bu dosya, Uzak Araçlar ve Belgeler kuruluşunun *MQ\_INSTALLATION\_PATH/mqft/sql* dizininde bulunur.
  - b. İlk şemada değişiklik yaptıysanız, dosyanın değiştirilen veritabanınızla uyumlu olduğundan emin olmak için geçiş dosyasını inceleyin.
  - c. SQL dosyasını veritabanınız için çalıştırın.
6. Java EE veritabanı kaydedici EAR dosyasını kurun.
  7. Java EE veritabanı kaydediciyi konuşlandırın. Daha fazla bilgi için bkz [“MFT için Java EE veritabanı kaydedicisi kuruluyor” sayfa 776](#).
  8. Eşgüdüm kuyruk yöneticiniz için şu MQSC komutunu çalıştırın: ALTER QMGR PSMODE(ENABLED)  
Bu, iletilerin SYSTEM.FTE/Log konusu.

## Sonuçlar

### Connect:Direct köprüsünün yapılandırılması

Connect:Direct köprüsünü, bir Managed File Transfer ağı ile bir Connect:Direct ağı arasında dosya aktarmak için yapılandırın. Connect:Direct köprüsünün bileşenleri bir Connect:Direct düğümü ve o düğümle iletişim kurmaya adanmış bir Managed File Transfer aracıdır. Bu aracıya Connect:Direct köprü aracı denir.

#### Başlamadan önce

Connect:Direct köprüsünü oluşturan aracı ve düğüm aynı sistemde olmalı ya da aynı dosya sistemine (örneğin, paylaşılan bir NFS bağlantısı aracılığıyla) erişmelidir. Bu dosya sistemi, Connect:Direct köprüsünü içeren dosya aktarımları sırasında dosyaları, **cdTmpDir** parametresi tarafından tanımlanan bir dizinde geçici olarak saklamak için kullanılır. Connect:Direct köprü aracı ve Connect:Direct köprü düğümü, aynı yol adını kullanarak bu dizini adresleyebilmelidir. Örneğin, aracı ve düğüm ayrı Windows sistemlerideyse, sistemlerin paylaşılan dosya sistemini bağlamak için aynı sürücü harfini kullanması gerekir. Aşağıdaki yapılandırmalar, aracının ve düğümün aynı yol adını kullanmasına izin verir:

- Aracı ve düğüm aynı sistemde; Windows ya da Linux for x86-64 çalıştırılıyor
- Aracı, x86-64 için Linux üzerinde ve düğüm AIX üzerdedir
- Aracı bir Windows sisteminde ve düğüm başka bir Windows sisteminde

Aşağıdaki yapılandırmalar, aracının ve düğümün aynı yol adını kullanmasına izin vermiyor:

- Aracı, x86-64 için Linux üzerinde ve düğüm Windows üzerdedir
- Aracı Windows üzerdedir ve düğüm UNIX üzerdedir

Connect:Direct köprüsünü kurmayı planlarken bu kısıtlamayı göz önünde bulundurun.

Connect:Direct köprüsü için desteklenen işletim sistemi sürümlerine ilişkin daha fazla ayrıntı için [IBM MQ için Sistem Gereksinimleri](#) web sayfasına bakın.

#### Bu görev hakkında

Connect:Direct köprü aracı, Connect:Direct düğümüyle iletişim kurmaya adanmış bir Managed File Transfer aracıdır.

Varsayılan olarak, Connect:Direct köprü aracı Connect:Direct düğümüne bağlanmak için TCP/IP iletişim kuralını kullanır. Connect:Direct köprü aracınız ile Connect:Direct düğümü arasında güvenli bir bağlantı istiyorsanız, SSL iletişim kuralını ya da TLS iletişim kuralını kullanabilirsiniz.

#### Yordam

1. Connect:Direct köprü aracı ve düğümü için işletim sistemlerini seçin:

- a) Connect:Direct köprü aracısını kurmak için x86-64 üzerinde Windows ya da Linux çalıştıran bir sistem seçin.
  - b) Connect:Direct for Windows tarafından desteklenen bir işletim sistemi seçin ya da Connect:Direct köprü düğümünü kurmak için Connect:Direct for UNIX seçeneğini belirleyin.
2. Bir Connect:Direct düğümü seçin ve yapılandırın.

Bu yönergeleri izlemeden önce bir Connect:Direct düğümünün kurulu olması gerekir.

- a) Managed File Transfer aracısının iletişim kurması için bir Connect:Direct düğümü seçin.
- b) Seçtiğiniz Connect:Direct düğümüne ilişkin ağ eşlemeni denetleyin. Ağ eşleminde Windows işletim sisteminde çalışan uzak düğümlere ilişkin girişler varsa, bu girişlerin düğümlerin Windows üzerinde çalıştığından emin olmanız gerekir.



Connect:Direct köprüsü için seçtiğiniz Connect:Direct düğümü Windows üzerinde çalışıyorsa, ağ eşlemeni düzenlemek için Connect:Direct İstekçiyi kullanın. Windows üzerinde çalışan uzak düğümler için **İşletim Sistemi** alanının **Windows** olarak ayarlandığından emin olun.

3. Bir Connect:Direct köprü aracısı oluşturun ve yapılandırın.

- a) **fteCreateCDAgent** komutunu kullanarak bir Connect:Direct köprü aracısı oluşturun.

- **cdNode** parametresi için bir değer sağlamalısınız. Bu parametre, aracının Connect:Direct köprüsünün bir parçası olan Connect:Direct düğümü için kullandığı adı belirtir. Önceki bölümde seçtiğiniz Connect:Direct düğümünün adını kullanın.
- Aracının iletişim kurduğu Connect:Direct düğümünü tanımlayan **cdNodeHost** ve **cdNodePort** parametreleri için değerler sağlayın.

**cdNodeHost** parametresi için bir değer sağlamazsanız, yerel sistemin anasistem adı ya da IP adresi kullanılır. **cdNodePort** parametresi için bir değer belirtmezseniz, 1363 değeri kullanılır.

- İsteğe bağlı olarak, **cdTmpDir** parametresi için bir değer belirtmeniz gerekip gerekmediğini belirlemek için **fteCreateAgent** içindeki bilgileri kullanın.

- b) Managed File Transfer tarafından kullanılan kullanıcı kimlik bilgilerini bir Connect:Direct düğümündeki kullanıcı kimlik bilgileriyle eşleyin. Aşağıdaki yöntemlerden birini kullanarak kimlik bilgilerini eşleyebilirsiniz:

- Kimlik bilgisi eşleme bilgilerini tanımlamak için bir **ConnectDirectCredentials.xml** dosyası oluşturun. Daha fazla bilgi için bkz [“Connect:Direct kimlik bilgilerini ConnectDirectCredentials.xml dosyasını kullanarak eşleme” sayfa 788.](#)
- Connect:Direct köprünüz için kimlik bilgileri eşlemesi gerçekleştirmek üzere bir kullanıcı çıkışı yazın. Daha fazla bilgi için bkz [“Çıkış sınıflarını kullanarak Connect:Direct kimlik bilgilerini eşleme” sayfa 791.](#)

4. **ConnectDirectNodeProperties.xml** dosyasını, uzak Connect:Direct düğümlerine ilişkin bilgileri içerecek şekilde yapılandırın.


Bu yönergeleri izlemeden önce bir Connect:Direct köprü aracısı oluşturmanız gerekir.

Connect:Direct köprü aracısı yapılandırma dizinindeki **ConnectDirectNodeProperties.xml** şablonunu düzenleyin. Hakkında bilgi tanımlamak istediğiniz her bir Connect:Direct düğümü ya da düğüm grubu için aşağıdaki adımları gerçekleştirin:

- a) **nodeProperties** ögesinin içinde bir **node** ögesi yaratın.
- b) **node** ögesine bir **name** özniteliği ekleyin. Bir ya da daha çok uzak Connect:Direct düğümünün adıyla eşleşecek bir kalıp olarak bu özniteliğin değerini belirtin.
- c) İsteğe bağlı: **name** özniteliğindeki değer ne tip bir kalıp olduğunu belirten **node** ögesine bir **pattern** özniteliği ekleyin. Geçerli değerler şunlardır: **regex** ve **wildcard**. Varsayılan seçenek **wildcard** dir.
- d) **node** ögesine, **name** özniteliği tarafından belirtilen uzak Connect:Direct düğümlerinin çalıştığı işletim sistemini belirten bir **type** özniteliği ekleyin.

Aşağıdaki değerler geçerlidir:



- Windows -düğüm Windows üzerinde çalışır
- UNIX -Düğüm AIX and Linux üzerinde çalışır
-  z/OS, zos, os/390ya da os390 -Düğüm z/OS üzerinde çalışır

Bu özneteliğin değeri büyük ve küçük harfe duyarlı değildir. Diğer işletim sistemlerinde uzak düğümlere yapılan aktarımlar, Connect:Direct köprüsü tarafından desteklenmez.

Daha fazla bilgi için bkz. [Connect:Direct düğüm özellikleri dosya biçimi](#).

5. Connect:Direct köprü aracı ile Connect:Direct düğümü arasında güvenli bir bağlantı yapılandırın. Bunun nasıl gerçekleştirileceğine ilişkin bir örnek için [Connect:Direct köprü aracı ve Connect:Direct düğümü arasında SSL ya da TLS ' nin yapılandırılması](#) başlıklı konuya bakın.

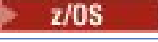
### İlgili görevler

[Connect:Direct köprüsünde sorun giderme](#)

[Connect:Direct köprü aracı ile Connect:Direct düğümü arasında SSL ya da TLS ' nin yapılandırılması](#)

[Connect:Direct düğümüne dosya aktarılması](#)

[Connect:Direct düğümünden dosya aktarılması](#)

 [Connect:Direct düğümünden birden çok dosyanın aktarılması](#)

### İlgili başvurular

[Connect:Direct köprüsü](#)

## Connect:Direct için kimlik bilgilerini eşleme

Managed File Transfer içindeki kullanıcı kimlik bilgilerini, Connect:Direct köprü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanarak ya da kendi kullanıcı çıkışınızı yazarak bir Connect:Direct düğümündeki kullanıcı kimlik bilgileriyle eşleyin. Managed File Transfer , kullanıcı kimlik bilgileri eşlemesi gerçekleştiren örnek bir kullanıcı çıkışı sağlar.

### İlgili görevler

“Connect:Direct kimlik bilgilerini ConnectDirectCredentials.xml dosyasını kullanarak eşleme” sayfa 788  
Connect:Direct köprü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanarak Managed File Transfer içindeki kullanıcı kimlik bilgilerini Connect:Direct düğümlerindeki kullanıcı kimlik bilgileriyle eşleyin. Managed File Transfer , kimlik bilgilerinizi dahil etmek için düzenleyebileceğiniz bir XML dosyası sağlar.

“Çıkış sınıflarını kullanarak Connect:Direct kimlik bilgilerini eşleme” sayfa 791

Connect:Direct Bridge aracısının varsayılan kimlik bilgisi eşleme işlevini kullanmak istemiyorsanız, kendi kullanıcı çıkışınızı yazarak Managed File Transfer içindeki kullanıcı kimlik bilgilerini bir Connect:Direct düğümündeki kullanıcı kimlik bilgileriyle eşleyebilirsiniz. Kendi kimlik bilgisi eşleme kullanıcı çıkışlarınızın yapılandırılması, varsayılan kimlik bilgisi eşleme işlevini devre dışı bırakır.

### İlgili başvurular

[CDCredentialExit.java arabirimi](#)

[Connect:Direct kimlik bilgileri dosya biçimi](#)

## Connect:Direct kimlik bilgilerini ConnectDirectCredentials.xml dosyasını kullanarak eşleme

Connect:Direct köprü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanarak Managed File Transfer içindeki kullanıcı kimlik bilgilerini Connect:Direct düğümlerindeki kullanıcı kimlik bilgileriyle eşleyin. Managed File Transfer , kimlik bilgilerinizi dahil etmek için düzenleyebileceğiniz bir XML dosyası sağlar.

## Bu görev hakkında

**ftCreateCDAgent** komutu kullanılarak bir Connect:Direct köprü aracı oluşturulduktan sonra, bir ConnectDirectCredentials.xml dosyasının el ile oluşturulması gerekir. Bir Connect:Direct köprü aracısını kullanmadan önce, bu dosyayı anasistem, kullanıcı ve kimlik bilgilerini içerecek şekilde düzenlemeniz gerekir. Daha fazla bilgi için bkz. [Connect:Direct kimlik bilgileri dosya biçimi](#). Varsayılan olarak, bu dosya geçerli kullanıcının ana dizininden (örneğin, /



home/fteuser/ConnectDirectCredentials.xml) yüklenir. Başka bir konum kullanmak için, ConnectDirectNodeProperties.xml dosyasındaki <credentialsFile> ögesini kullanarak belirtin.

## Yordam

1. <tns:pnode name="Connect:Direct node host" pattern="wildcard"> ögesindeki name özniteliğinin, Connect:Direct köprü aracısının bağlandığı Connect:Direct düğümünün adını içerdiğinden emin olun. Bu değer, **fteCreateCDAgent -cdNode** parametresi için belirttiğiniz değerle aynı olmalıdır.  
pattern özniteliğinin değeri wildcard ya da regex olabilir. Bu öznitelik belirtilmezse, varsayılan değer wildcard olur.
2. <tns:pnode>' un alt öğeleri olarak dosyaya kullanıcı kimliği ve kimlik bilgisi bilgilerini ekleyin. Dosyaya aşağıdaki <tns:user> ögesinin bir ya da daha çok örneğini ekleyebilirsiniz:

```
<tns:user name="name"
  pattern="pattern"
  ignorecase="ignorecase"
  cdUserId="cdUserId"
  cdPassword="cdPassword"
  pnodeUserId="pnodeUserId"
  pnodePassword="pnodePassword">
</tns:user>
```

Burada:

- *name* , MFT aktarma isteğiyle ilişkilendirilmiş MQMD kullanıcı kimliğiyle eşleşecek bir kalıptır.
- *pattern* , name özniteliği için belirtilen örüntünün bir genel arama karakteri ifadesi mi, yoksa Java düzenli ifadesi mi olduğunu belirtir. *pattern* özniteliğinin değeri *wildcard* ya da *regex* olabilir. Bu öznitelik belirtilmezse, varsayılan değer *wildcard* olur.
- *ignorecase* , name özniteliği tarafından belirtilen örüntünün büyük ve küçük harfe duyarlı olarak kabul edilip edilmeyeceğini belirtir. Bu öznitelik belirtilmezse, varsayılan değer *true* olur.
- *cdUserId* , Connect:Direct köprü aracı tarafından <tns:pnode> ögesinin name özniteliğiyle belirtilen Connect:Direct düğümüne bağlanmak için kullanılan kullanıcı kimliğidir. Olanaklıysa, *cdUserId* ' in bir Connect:Direct yönetici kullanıcı kimliği olduğundan emin olun. *cdUserId* bir Connect:Direct yöneticisi olamıyorsa, kullanıcı kimliğinin Connect:Direct köprü düğümünde aşağıdaki işlevsel yetkilere sahip olduğundan emin olun:
  - Windows düğümü için aşağıdaki yetkileri ayarlayın. Bu örnek, okunabilirliğe yardımcı olmak için satır başlarıyla biçimlendirilir:

```
View Processes in the TCQ value: yes
Issue the copy receive, copy send, run job, and run task Process statements value: yes
Issue the submit Process statement value: yes
Monitor, submit, change, and delete all Processes value: all
Access Process statistics value: all
Use the trace tool or issue traceon and traceoff commands value: yes
Override Process options such as file attributes and remote node ID value: yes
```

- Bir AIX ya da Linux düğümü için `userfile.cfg` dosyasında aşağıdaki parametreleri ayarlayın:

```
pstmt.copy           value: y
pstmt.upload         value: y
pstmt.download       value: y
pstmt.runjob         value: y
pstmt.runtask        value: y
cmd.submit           value: y
pstmt.submit         value: y
cmd.chgproc          value: y
cmd.delproc          value: y
cmd.flspoc           value: y
cmd.selproc          value: a
cmd.selstats         value: a
cmd.trace            value: y
snode.ovrd           value: y
```

- `cdPassword` , `cdUserId` özniteliği tarafından belirtilen kullanıcı kimliğiyle ilişkili paroladır.
- İsteğe bağlı olarak `pnodeUserId` özniteliğini belirtebilirsiniz. Bu özniteliğin değeri, Connect:Direct sürecini sunmak için `<tns:pnode>` öğesinin `name` özniteliği tarafından belirtilen Connect:Direct düğümü tarafından kullanılan kullanıcı kimliğidir. `pnodeUserId` özniteliğini belirtmezseniz, Connect:Direct düğümü Connect:Direct işlemi sunmak için `cdUserId` özniteliğiyle belirtilen kullanıcı kimliğini kullanır.
- İsteğe bağlı olarak `pnodePassword` özniteliğini belirtebilirsiniz. Bu özniteliğin değeri, `pnodeUserId` özniteliği tarafından belirtilen kullanıcı kimliğiyle ilişkili paroladır.

MQMD kullanıcı kimliğiyle eşleşen bir kullanıcı ögesi yoksa, aktarma başarısız olur.

3. İsteğe bağlı: Bir ya da daha çok `<tns:snode>` ögesini `<tns:user>` ögesinin alt ögeleri olarak ekleyebilirsiniz. `<tns:snode>` ögesi, Connect:Direct köprüsünün bir parçası olan Connect:Direct düğümü tarafından kullanılan kimlik bilgilerini belirtir. Bu kimlik bilgileri, Connect:Direct Bridge düğümünün dosya aktarımının kaynağı ya da hedefi olan Connect:Direct düğümüne bağlanmak için kullandığı kullanıcı kimliği ve paroladır.

Dosyaya aşağıdaki ögelerden birini ya da birkaçını ekleyin:

```
<tns:snode name="name"
  pattern="pattern"
  userId="userId"
  password="password" />
```

Burada:

- `name` , dosya aktarımının kaynağı ya da hedefi olan Connect:Direct düğümünün adıyla eşleşecek bir kalıptır.
- `pattern` , `name` özniteliği için belirtilen örüntünün bir genel arama karakteri ifadesi mi, yoksa Java düzenli ifadesi mi olduğunu belirtir. Örüntü özniteliğinin değeri `wildcard` ya da `regex` olabilir. Bu öznitelik belirtilmezse, varsayılan değer `wildcard` olur.
- `userId` , `<tns:pnode>` ögesinin `name` özniteliği tarafından `<tns:snode>` ögesinin `name` özniteliğiyle belirtilen kalıpla eşleşen bir Connect:Direct düğümüne bağlanmak için belirtilen Connect:Direct düğümü tarafından kullanılan kullanıcı kimliğidir.
- `password` , `userId` özniteliği tarafından belirtilen kullanıcı kimliğiyle ilişkili paroladır.

Dosya aktarımının ikincil düğümüyle eşleşen bir `<tns:snode>` ögesi yoksa, bu aktarma işleminin başarısız olmasına neden olmaz. Aktarma başlatılır ve `snode` ile kullanılmak üzere kullanıcı kimliği ve parola belirtilmez.

## Sonuçlar

Kullanıcı adları ya da Connect:Direct düğüm adları için bir örüntü eşleşmesi aranırken, Connect:Direct köprü aracı dosyanın başından sonuna kadar arama yapar. Bulunan ilk eşleşme, kullanılan eşleşti.

### İlgili görevler

[“Connect:Direct köprüsünün yapılandırılması” sayfa 786](#)

Connect:Direct köprüsünü, bir Managed File Transfer ağı ile bir Connect:Direct ağı arasında dosya aktarmak için yapılandırın. Connect:Direct köprüsünün bileşenleri bir Connect:Direct düğümü ve o düğümle iletişim kurmaya adanmış bir Managed File Transfer aracıdır. Bu aracıya Connect:Direct köprü aracı denir.

### İlgili başvurular

Connect:Direct kimlik bilgileri dosya biçimi

[fteCreateCDAgent: Connect:Direct köprü aracı oluşturma](#)

### Çıkış sınıflarını kullanarak Connect:Direct kimlik bilgilerini eşleme

Connect:Direct Bridge aracısının varsayılan kimlik bilgisi eşleme işlevini kullanmak istemiyorsanız, kendi kullanıcı çıkışınızı yazarak Managed File Transfer içindeki kullanıcı kimlik bilgilerini bir Connect:Direct düğümündeki kullanıcı kimlik bilgileriyle eşleyebilirsiniz. Kendi kimlik bilgisi eşleme kullanıcı çıkışlarınızın yapılandırılması, varsayılan kimlik bilgisi eşleme işlevini devre dışı bırakır.

### Bu görev hakkında

Connect:Direct kimlik bilgilerini eşlemek için oluşturduğunuz kullanıcı çıkışları `com.ibm.wmqfte.exitroutine.api.ConnectDirectCredentialExit` aracıyı uygulamalıdır. Daha fazla bilgi için bkz. [CDCredentialExit.java](#) aracıyı.

## IBM MQ Console ve REST API ' nin yapılandırılması

IBM MQ Console ve REST API öğelerini barındıran mqweb sunucusu varsayılan bir yapılandırma ile sağlanır. Bu bileşenlerden birini kullanabilmek için, kullanıcıların oturum açmasına izin verecek şekilde güvenlik yapılandırılması gibi, bir dizi yapılandırma görevinin tamamlanması gerekir. Bu konuda, kullanılabilir tüm yapılandırma seçenekleri açıklanmaktadır.

### Yordam

- [“mqweb sunucusu için temel yapılandırma” sayfa 791](#)
- [“Güvenliğin yapılandırılması” sayfa 797](#)
- [“HTTP anasistem adının yapılandırılması” sayfa 797](#)
- [“HTTP ve HTTPS kapılarının yapılandırılması” sayfa 798](#)
- [“Yanıt zaman aşımının yapılandırılması” sayfa 799](#)
- [“Otomatik başlatma yapılandırılıyor” sayfa 800](#)
- [“Günlüğe kaydetmenin yapılandırılması” sayfa 801](#)
- [“LTPA belirtecinin yapılandırılması” sayfa 805](#)
- [“IBM MQ Console için uzak kuyruk yöneticisi bağlantı davranışının yapılandırılması” sayfa 807](#)
- [“administrative REST API ağ geçidini yapılandırma” sayfa 808](#)
- [“messaging REST API ' nin yapılandırılması” sayfa 810](#)
- [“MFT için REST API ' in yapılandırılması” sayfa 816](#)
- [“mqweb sunucusu JVM 'sinin ayarlanması” sayfa 821](#)
- [“IBM MQ Console ve REST API kuruluş bileşeninin dosya yapısı” sayfa 823](#)

## mqweb sunucusu için temel yapılandırma

REST API ya da IBM MQ Console'ünü kullanmaya başlamadan önce, doğru bileşenleri kurmanız ve REST API ya da IBM MQ Console sunucusunu barındıran mqweb sunucusunu yapılandırmanız gerekir.

### Bu görev hakkında

Bu göreve ilişkin yordam, REST API ve IBM MQ Console ile hızlı bir şekilde çalışmaya başlayabilmeniz için mqweb sunucusu için temel bir yapılandırmaya odaklanır. Temel bir kullanıcı kayıt dosyasının

nasıl ayarlanacağına ilişkin güvenlik anahattını yapılandırma adımları, ancak kullanıcıların ve rollerin yapılandırılmasına ilişkin diğer seçenekler vardır. Mqweb sunucusu için güvenliği yapılandırma hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API güvenlik](#).

**Not:** Bu yordamı tamamlamak için mqwebuser.xml kütüğüne erişiminiz olmalıdır:

- **z/OS** z/OS' da, mqwebuser.xml dosyasına yazma erişimi olan bir kullanıcı olmanız gerekir.
- **Multi** Diğer tüm işletim sistemlerinde, mqwebuser.xml dosyasına erişmek için [ayrıcalklı kullanıcı](#) olmanız gerekir.
- **Linux** **V 9.3.5** mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.

## Yordam

1. IBM MQ Console ve REST API bileşenini kurun:

- **AIX** AIX' da mqm.web.rte dosya kümesini kurun. AIXüzerine dosya kümeleri kurulmasıyla ilgili daha fazla bilgi için bkz. [AIX kuruluş görevleri](#).
- **IBM i** IBM üzerinde WEB bileşenini kurun. Bu özelliği kullanmak için 5724L26 IBM MQ Java Messaging and Web Services ve 5770JV1 Java SE 8 önkoşullarını da kurmanız gerekir. Özellikleri IBM üzerine kurma hakkında daha fazla bilgi için bkz. [IBM i kuruluş görevleri](#).
- **Linux** Linux' da MQSeriesWeb bileşenini kurun. Bileşenlerin Linuxüzerine kurulmasıyla ilgili daha fazla bilgi için bkz. [Linux kuruluş görevleri](#).
- **V 9.3.5** IBM MQ 9.3.5olanağından, Linuxüzerinde bağımsız bir IBM MQ Web Server kuruluşunda mqweb sunucusunu çalıştırabilirsiniz. IBM MQ Web Serverkuruluşu hakkında daha fazla bilgi için bkz. [Bağımsız kuruluşu IBM MQ Web Server](#).
- **Windows** Windowsüzerinde, Web Administration özelliğini kurun. Özellikleri Windowsüzerine kurma hakkında daha fazla bilgi için bkz. [Windows kuruluş görevleri](#).
- **z/OS** IBM MQ for z/OS UNIX System Services Web Components özelliğini kurun. Bileşenlerin ve özelliklerin z/OSüzerine kurulmasıyla ilgili daha fazla bilgi için bkz. [z/OS kuruluş görevleri](#).

2. IBM MQ Console ve REST API' ı barındıran mqweb sunucusunu yaratın.

- **z/OS** z/OS' da **crtmqweb** komut dosyasını çalıştırın.  
Bu komut dosyası, mqweb sunucusu yapılandırmasını ve günlük dosyalarını içeren WebSphere Liberty kullanıcı dizinini yaratır. **crtmqweb** komut dosyasını çalıştırma hakkında daha fazla bilgi için bkz. [“mqweb sunucusu yaratılıyor” sayfa 943](#).
- **Linux** **V 9.3.5** Bağımsız bir IBM MQ Web Server kuruluşunda, [“Bağımsız IBM MQ Web Server ' in yapılandırılması” sayfa 795](#) içindeki adımları izleyin.
- Diğer tüm ortamlarda, mqweb sunucusunu yaratmak için herhangi bir işlemi tamamlamanız gerekmez.

3. **z/OS**

z/OSüzerinde, mqweb sunucusunu başlatmak için kataloğa alınmış bir yordam yaratın.  
Daha fazla bilgi için bkz [“mqweb sunucusu için yordam yaratılması” sayfa 945](#).

4. Var olan mqwebuser.xml yapılandırma dosyasını, temel güvenlik sunmak üzere yapılandırılan temel kayıt dosyası ile değiştirin. basic\_registry.xml dosyasını MQ\_INSTALLATION\_PATH/web/mq/samp/configuration dizininden sisteminiz için uygun dizine kopyalayın ve dosyayı mqwebuser.xmlolarak yeniden adlandırın:

- IBM MQ kuruluşunda, dosyayı aşağıdaki dizine kopyalayın:

- **Linux** **AIX** AIX and Linux işletim tarihinde: `/var/mqm/web/installations/installationName/servers/mqweb`
- **Windows** Windows işletim tarihinde:  
`MQ_DATA_PATH\web\installations\installationName\servers\mqweb`  
Burada `MQ_DATA_PATH` , IBM MQ veri yoludur; bu yol, IBM MQ kuruluşu sırasında seçilen veri yoludur. Varsayılan olarak bu yol `C:\ProgramData\IBM\MQ` olur.
- **z/OS** z/OS işletim tarihinde: `WLP_user_directory/servers/mqweb`  
Burada `WLP_user_directory` , `crtmqweb` komut dosyası `mqweb` sunucusu tanımlamasını oluşturmak için çalıştırıldığında belirtilen dizindir.
- **Linux** **V 9.3.5** Bağımsız bir IBM MQ Web Server kuruluşunda:  
`MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST/servers/mqweb`  
Burada `MQ_OVERRIDE_DATA_PATH` , `MQ_OVERRIDE_DATA_PATH` ortam değişkeninin gösterdiği IBM MQ Web Server veri dizidir.

`basic_registry.xml` örnek dosyası dört kullanıcıyı yapılandırır:

#### **mqadmin**

MQWebAdmin rolünün üyesi olan bir yönetici kullanıcı.

#### **mqreader**

MQWebAdminRO rolünün üyesi olan salt okunur yönetici kullanıcı.

#### **mftadmin**

MFTWebAdmin rolünün üyesi olan bir yönetici kullanıcı.

#### **mftreader**

MFTWebAdminRO rolünün üyesi olan salt okunur yönetici kullanıcı.

Tüm kullanıcılar `MQWebUser` rolünün de üyeleridir.

Kullanılabilir roller hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API üzerindeki roller](#)

5. İsteğe bağlı: Daha fazla kullanıcı ve grup eklemek için `mqwebuser.xml` dosyasını düzenleyin. Bu kullanıcılara ve gruplara, REST API ya da IBM MQ Console ürününü kullanma yetkisi vermek üzere uygun rolleri atayın. Varsayılan olarak tanımlanan kullanıcıların parolalarını değiştirebilir ve yeni parolaları kodlayabilirsiniz. Daha fazla bilgi için [Kullanıcıları ve rolleri yapılandırma](#) başlıklı konuya bakın.

#### **Not:**

- **z/OS** z/OS üzerinde, `MQWebUser` rolüne kullanıcı eklerseniz, `MQWebUser` rolüyle kullanıcı kimliklerine ilişkin `mqweb` tarafından başlatılan görev kullanıcı kimliğine diğer kullanıcı erişimi de vermeniz gerekir. Örneğin:

```
RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
```

- **z/OS** **Multi** messaging REST API ile çalışmaya başlama adımlarını tamamlamak için `mqwebuser.xml` dosyasına bir kullanıcı eklemeniz gerekir. Bu kullanıcı, sisteminizde var olan bir IBM MQ kullanıcısıyla aynı ada sahip olmalıdır. XML dosyasındaki diğer kullanıcılarla aynı biçimi izleyerek, XML dosyasında şu satırdan sonra kullanıcı kimliğini ve parolayı ekleyin: `<user name="mftreader" password="mftreader"/>`.
6. Ortamınızı `mqweb` sunucusu yapılandırmasını gösterecek şekilde ayarlayın.

- **z/OS** z/OS' ta, aşağıdaki komutu girerek `WLP_USER_DIR` ortam değişkenini, değişken `mqweb` sunucusu yapılandırmanızı gösterecek şekilde ayarlayın:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory* , **crtmqweb** komutuna geçirilen dizinin adidir. Örneđin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz “mqweb sunucusu yaratılıyor” sayfa 943.

- **Linux** **V 9.3.5** Bağımsız bir IBM MQ Web Server kuruluşunda, **MQ\_OVERRIDE\_DATA\_PATH** ortam değışkenini IBM MQ Web Server veri dizinine ayarlayın. Örneđin, IBM MQ Web Server veri dizininiz olarak /var/mqweb komutunu kullanmayı seçerseniz, ařađıdaki komutu verin:

```
export MQ_OVERRIDE_DATA_PATH=/var/mqweb
```

- Diđer tüm ortamlarda, ortamınızı ayarlamak için herhangi bir işlemleri tamamlamanız gerekmez.
7. Varsayılan olarak, REST API ve IBM MQ Console yalnızca mqweb sunucusu ile aynı anasistemden kullanılabilir. Ařađıdaki komutu girerek mqweb sunucusuna uzak bağlantıları etkinleştirin:

```
setmqweb properties -k httpHost -v hostname
```

Burada *anasistemadı* , etki alanı adı soneki olan IP adresini, etki alanı ad sunucusu (DNS) anasistem adını ya da IBM MQ ' in kurulu olduđu sunucunun DNS anasistem adını belirtir. Ařađıdaki örnekte gösterildiđi gibi, kullanılabilir tüm ağ arabirimlerini belirtmek için çift tırnak işareti içinde yıldız işareti (\*) kullanın:

```
setmqweb properties -k httpHost -v "*"
```

8. İsteđe bađlı: Varsayılan olarak, MFT için administrative REST API etkinleştirilmez. Bu özelliđi kullanmak istiyorsanız, bu özelliđi etkinleştirmeniz ve bir koordinasyon kuyruk yöneticisi yapılandırmanız gerekir:

- a) Ařađıdaki komutu girerek MFT için administrative REST API ' i etkinleştirin:

```
setmqweb properties -k mqRestMftEnabled -v true
```

- b) Ařađıdaki komutu girerek hangi kuyruk yöneticisinin koordinasyon kuyruk yöneticisi olduđunu yapılandırın:

```
setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
```

Burada *qmgrName* , koordinasyon kuyruđu yöneticisinin adıdır.

- c) POST çağrılarını etkinleřtirmek için, ařađıdaki komutu girerek hangi kuyruk yöneticisinin komut kuyruđu yöneticisi olduđunu yapılandırın:

```
setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
```

Burada *qmgrName* , komut kuyruđu yöneticisinin adıdır.

9. REST API ve IBM MQ Console' ı destekleyen mqweb sunucusunu başlatın:

- **ALW** AIX, Linux, and Windows işletim sistemlerinde ayrıcalklı kullanıcı olarak řu komutu girin:

```
stmqweb
```

- **IBM i** IBM işletim sistemlerinde, ayrıcalklı kullanıcı olarak Qshell 'de řu komutu girin:

```
/QIBM/ProdData/mqm/bin/stmqweb
```

- **z/OS** z/OS üzerinde, “mqweb sunucusu için yordam yaratılması” sayfa 945 içinde oluşturduđunuz yordamı başlatın.

Aşağıdaki iletiler, mqweb sunucusunun başarıyla başlatıldığını belirtmek için STDOUT DD ' ye gönderilir.

```
[AUDIT ] MQWB2019I: MQ Console level: 9.2.4 - V924-CD924-L211028
[AUDIT ] MQWB0023I: MQ REST API level: 9.2.4 - V924-CD924-L211028
[AUDIT ] CWWKZ0001I: Application com.ibm.mq.rest started in 1.763 seconds.
[AUDIT ] CWWKZ0001I: Application com.ibm.mq.console started in 2.615 seconds.
[AUDIT ] CWWKF0011I: The mqweb server is ready to run a smarter planet. The mqweb
server started in 10.016 seconds.
```

z/OS üzerinde mqweb sunucusunun başlattığı görevi durdurarak ya da **endmqweb** komutunu kullanarak istediğiniz zaman mqweb sunucusunu durdurabilirsiniz. Ancak, mqweb sunucusu çalışmıyorsa, REST API ya da IBM MQ Console kullanamazsınız.

## 10. z/OS

İsteğe bağlı: z/OS sistemlerinde, sistem otomasyonu ürünlerinin IBM MQ Console ve REST API başladığında yayınlanan MQWB2019I ve MQWB0023I iletilerini tuzağa düşürmesine izin vermek istiyorsanız, mqweb sunucusunu bu iletileri MVS konsoluna yazacak şekilde yapılandırın. mqweb sunucusunu MQWB2019I ve MQWB0023I iletilerini MVS konsoluna yazacak şekilde yapılandırmak için, “4” sayfa 792. adımda yarattığınız mqwebuser.xml dosyasını düzenleyin ve dosyaya aşağıdaki satırı ekleyin:

```
<zosLogging enableLogToMVS="true" wtoMessage="MQWB2019I,MQWB0023I"/>
```

z/OS mqweb sunucusunda günlüğe kaydetmeyi yapılandırma hakkında daha fazla bilgi için bkz. [z/OS Günlük Kaydı \(zosLogging\)](#).

## Sonraki adım

1. HTTP bağlantılarının etkinleştirilmesi ve kapı numarasının değiştirilmesi de içinde olmak üzere mqweb sunucusu ayarlarını yapılandırın. Daha fazla bilgi için bkz. “IBM MQ Console ve REST API ' nin yapılandırılması” sayfa 791.
2. İsteğe bağlı olarak, REST API' yi yapılandırın:
  - a. REST API için Köken Arası Kaynak Paylaşımını yapılandırın. Varsayılan olarak, REST API ile aynı etki alanında bulunmayan web kaynaklarından REST API ' e erişemezsiniz. Yani, kökler arası istekler etkinleştirilmez. Kökler Arası Kaynak Paylaşımını (CORS), belirtilen URL adreslerinden gelen kökler arası isteklere izin verecek şekilde yapılandırabilirsiniz. Daha fazla bilgi için bkz. [REST API için CORS ' nin yapılandırılması](#).
  - b. MFT için REST API ' i yapılandırın. Daha fazla bilgi için bkz. “MFT için REST API ' in yapılandırılması” sayfa 816.
3. REST API ya da IBM MQ Console kullanın:
  - [administrative REST API ile çalışmaya başlama](#)
  - [messaging REST API ile çalışmaya başlama](#)
  - [IBM MQ Console ile çalışmaya başlama](#)

Linux

V 9.3.5

## Bağımsız IBM MQ Web Server ' in yapılandırılması

IBM MQ 9.3.5 olanağından, IBM MQ Console ve REST API öğelerini barındıran mqweb sunucusunu bağımsız bir IBM MQ Web Server kuruluşunda çalıştırabilirsiniz.

## Başlamadan önce

Bağımsız IBM MQ Web Server yalnızca Linux üzerinde kullanılabilir.

mqweb sunucusunu yapılandırmadan önce, [Bağımsız kuruluşu IBM MQ Web Server](#) içindeki adımları izleyerek IBM MQ Web Server kurulmalıdır.

## Bu görev hakkında

Bağımsız bir IBM MQ Web Server kuruluşunda çalışan yeni bir mqweb sunucusu yaratmak ve yapılandırmak için bu görevdeki yordamı izleyin. Bu yordamı yineleyerek, bağımsız bir IBM MQ Web Server kuruluşunda çalışacak birden çok mqweb sunucusu yapılandırabilirsiniz.

## Yordam

1. IBM MQ Web Server veri dizinini oluşturun.

Veri dizini, IBM MQ Console ve REST API'yi çalıştıran mqweb sunucusuna ilişkin yapılandırma ve günlük dosyalarını saklamak için kullanılır. IBM MQ Web Server veri dizini olarak seçtiğiniz herhangi bir dizini kullanabilirsiniz.

mqweb sunucusunu başlatmak için kullandığınız kullanıcı kimliğine veri dizini için okuma ve yazma erişimi verilmelidir.

2. **MQ\_OVERRIDE\_DATA\_PATH** ortam değişkenini, "[1](#)" sayfa 796. adımda oluşturduğunuz veri dizinine ayarlayın.

Örneğin, IBM MQ Web Server veri dizininiz olarak /var/mqweb komutunu kullanmayı seçerseniz, aşağıdaki komutu verin:

```
export MQ_OVERRIDE_DATA_PATH=/var/mqweb
```

3. IBM MQ ortamını ayarlamak için **setmqenv** komutunu kullanın.

IBM MQ Web Server kuruluş dizininin bin dizinine geçin ve aşağıdaki komutu verin:

```
. setmqenv -s
```

4. Veri dizininde IBM MQ dizinlerini ve dosyalarını oluşturmak için **crtmqdir** komutunu kullanın. Yaratılan dosyalar, mqweb sunucusu için bir şablon tanımlaması içeriyor.

Şu komutu verin:

```
crtmqdir -s -f
```

5. İsteğe bağlı: Bu mqweb sunucusu, bağımsız IBM MQ Web Server kuruluşuyla çalışmak üzere yarattığınız ilk sunucuysa, IBM MQ lisansını gözden geçirmek ve kabul etmek için **mqlicense** komutunu kullanın.

Bu komutu, IBM MQ Web Server kuruluş dizinine yazma erişimi olan bir kullanıcı olarak çalıştırmanız gerekir.

Örneğin, IBM MQ lisansını görüntülemek için aşağıdaki komutu verin:

```
mqlicense
```

Daha fazla bilgi için bkz. [mqlicense](#).

6. İsteğe bağlı: Yeni yapılandırılan bağımsız IBM MQ Web Server kuruluşunda çalışmak üzere var olan bir mqweb sunucusunu yeni düzeye geçirmek için aşağıdaki adımları izleyin:

- a. Var olan mqweb sunucusu yapılandırmanızı yedekleyin.

- b. Dosyaları `MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST` dizinine geri yükleyin; burada `MQ_OVERRIDE_DATA_PATH`, "[1](#)" sayfa 796. adımda oluşturduğunuz IBM MQ Web Server veri dizinidir.

Daha fazla bilgi için bkz. "[mqweb sunucusu yapılandırmanızı yedekleme ve geri yükleme](#)" sayfa 825.

**Not:** IBM MQ Console ve REST API ürününün bazı özellikleri bağımsız bir IBM MQ Web Server kuruluşunda kullanılamaz. Bir mqweb sunucusunu IBM MQ kuruluşundan bağımsız bir IBM MQ Web Server kuruluşuna geçirirseniz, bu özellikler geçişten sonra kullanılamaz. Bağımsız bir IBM MQ Web Server kuruluşunda geçerli olan kısıtlamalar hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API](#).



## Sonraki adım

“mqweb sunucusu için temel yapılandırma” sayfa 791’inde açıklanan adımları izleyerek mqweb sunucusunu yapılandırın.

## Güvenliğin yapılandırılması

mqwebuser.xml dosyasını düzenleyerek IBM MQ Console ve REST API için güvenliği yapılandırabilirsiniz. Temel bir kullanıcı kayıt dosyasını ya da bir LDAP kaydını ya da WebSphere Liberty ile birlikte sağlanan diğer kayıt tiplerinden herhangi birini yapılandırarak kullanıcıları yapılandırabilir ve kimliklerini doğrulayabilirsiniz. Daha sonra, kullanıcılara ve gruplara bir rol atayarak bu kullanıcılara yetki verebilirsiniz.

### Bu görev hakkında

IBM MQ Console ve REST API için güvenliği yapılandırmak üzere kullanıcıları ve grupları yapılandırmanız gerekir. Bu kullanıcılar ve gruplar daha sonra IBM MQ Console, REST API ya da her ikisini kullanma yetkisine sahip olabilir. Kullanıcıları ve grupları yapılandırma ve kullanıcıların kimlik doğrulaması ve yetkilendirmesi hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API güvenliği](#).

Kullanıcılar IBM MQ Console ile kimlik doğrulaması yaptığında bir LTPA simgesi oluşturulur. Bu belirteç, kullanıcının belirteç süresi doluncaya kadar yeniden kimlik doğrulaması yapmadan IBM MQ Console 'yi kullanmasını sağlar.

REST API ile belirteç tabanlı kimlik doğrulamasını kullanırsanız, kullanıcı HTTP POST yöntemiyle /login REST API kaynağını kullanarak oturum açtığında farklı bir LTPA simgesi oluşturulur. Bu simgenin süresi dolduğunda ve bu simgenin hem HTTP hem de HTTPS bağlantıları için kullanılıp kullanılmayacağını yapılandırabilirsiniz. Daha fazla bilgi için bkz. [“LTPA belirtecini yapılandırılması” sayfa 805](#).

### Yordam





- [IBM MQ Console ve REST API güvenliği](#)
- [“LTPA belirtecini yapılandırılması” sayfa 805](#)

## HTTP anasistem adının yapılandırılması

Varsayılan olarak, IBM MQ Console ve REST API öğelerini barındıran mqweb sunucusu yalnızca yerel bağlantılara izin verecek şekilde yapılandırılır. Yani, IBM MQ Console ve REST API ürününe yalnızca IBM MQ Console ve REST API ürününün kurulu olduğu sistemde erişilebilir. **setmqweb** komutunu kullanarak anasistem adını uzak bağlantılara izin verecek şekilde yapılandırabilirsiniz.

### Başlamadan önce

Bu görevi tamamlamak için, **dspmweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

-  z/OS' da **dspmweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
-  Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.
-   mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.



**Uyarı:** 

z/OS üzerinde **setmqweb** ya da **dspmweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory* , *crtmqweb* ' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).



**Uyarı:** V 9.3.5 Linux

Bağımsız bir IBM MQ Web Server kuruluşunda **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, **MQ\_OVERRIDE\_DATA\_PATH** ortam değişkenini IBM MQ Web Server veri dizinine ayarlamanız gerekir.

## Yordam

- Aşağıdaki komutu kullanarak HTTP anasistem adının yürürlükteki yapılanışını görüntüleyin:

```
dspmqweb properties -a
```

httpHost alanı, HTTP anasistem adını gösterir.

- Aşağıdaki komutu kullanarak HTTP anasistem adını ayarlayın:

```
setmqweb properties -k httpHost -v hostName
```

Burada *hostName* , etki alanı adı sonekine sahip IP adresini, etki alanı ad sunucusu (DNS) anasistem adını ya da IBM MQ ' un kurulu olduğu sunucunun DNS anasistem adını belirtir. Kullanılabilir tüm ağ arabirimlerini belirtmek için çift tırnak içine alınmış bir yıldız işareti kullanın. Yalnızca yerel bağlantılara izin vermek için localhost değerini kullanın.

- Aşağıdaki komutu kullanarak HTTP anasistem adını ayarlayın:

```
setmqweb properties -k httpHost -d
```

## HTTP ve HTTPS kapılarının yapılandırılması

Varsayılan olarak, IBM MQ Console ve REST API öğelerini barındıran mqweb sunucusu, HTTPS 9443 kapısını kullanır. HTTP bağlantılarıyla ilişkili kapı devre dışı bırakıldı. HTTP kapısını etkinleştirebilir, farklı bir HTTPS kapısı yapılandırabilir ya da HTTP ya da HTTPS kapısını devre dışı bırakabilirsiniz. **setmqweb** komutunu kullanarak kapıları yapılandırabilirsiniz.

### Başlamadan önce

HTTP kapısını etkinleştirirseniz ve belirteç tabanlı kimlik doğrulamasını kullanıyorsanız, HTTP ve HTTPS bağlantıları için aynı LTPA simgesini etkinleştirmeniz gerekir. Daha fazla bilgi için bkz. [“LTPA belirtecinin yapılandırılması” sayfa 805](#).

Bu görevi tamamlamak için, **dspmqweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

- z/OS z/OS' da **dspmqweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
- Multi Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.
- V 9.3.5 Linux mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizindeki mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.



**Uyarı:** z/OS

z/OS üzerinde **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory* , *crtmqweb* ' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).



**Uyarı:** V 9.3.5 Linux

Bağımsız bir IBM MQ Web Server kuruluşunda **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, **MQ\_OVERRIDE\_DATA\_PATH** ortam değişkenini IBM MQ Web Server veri dizinine ayarlamanız gerekir.



**Uyarı:** Varsayılan olarak mqweb sunucusu, LTPA simgelerinin tüm istekler için güvenli kılınmasını gerektirir. mqweb sunucusu LTPA simgelerinin güvenli kılınmasını gerektirecek şekilde yapılandırıldıysa, HTTP kapısına bağlandığınızda aşağıdaki işlemleri tamamlayamazsınız:

- IBM MQ Console' da oturum açın.
- REST API ile belirteç tabanlı kimlik doğrulamasını kullanın.

LTPA simgelerinin HTTP istekleri tarafından kullanılmasına izin vermek için **secureLTPA** özelliğinin değerini *false* olarak ayarlayın. Daha fazla bilgi için bkz [“LTPA belirtecinin yapılandırılması” sayfa 805](#).

## Yordam

- Aşağıdaki komutu kullanarak HTTP ve HTTPS kapılarının geçerli yapılandırmasını görüntüleyin:

```
dspmqweb properties -a
```

*httpPort* alanında HTTP kapısı ve *httpsPort* alanında HTTPS kapısı gösterilir.

- HTTP kapısını etkinleştirmek ya da yapılandırmak için aşağıdaki komutu kullanın:

- Aşağıdaki komutu kullanarak HTTP kapısını etkinleştirin ya da ayarlayın:

```
setmqweb properties -k httpPort -v portNumber
```

Burada *portNumber* , HTTP bağlantıları için kullanmak istediğiniz kapıyı belirtir. -1 değerini kullanarak kapıyı devre dışı bırakabilirsiniz.

- Aşağıdaki komutu kullanarak HTTP kapı değerini -1 varsayılan değerine sıfırlayın:

```
setmqweb properties -k httpPort -d
```

- HTTPS kapısını yapılandırın:

- Aşağıdaki komutu kullanarak HTTPS kapı numarasını ayarlayın:

```
setmqweb properties -k httpsPort -v portNumber
```

Burada *portNumber* , HTTPS bağlantıları için kullanmak istediğiniz kapıyı belirtir. -1 değerini kullanarak kapıyı devre dışı bırakabilirsiniz.

- Aşağıdaki komutu kullanarak HTTPS kapı numarasını 9443 varsayılan değerine sıfırlayın:

```
setmqweb properties -k httpsPort -d
```

## Yanıt zaman aşımının yapılandırılması

Varsayılan olarak, bir istemciye yanıt göndermek için geçen süre 30 saniyeden uzunsa, IBM MQ Console ve REST API zamanaşımına uğrar. **setmqweb** komutunu kullanarak IBM MQ Console ve REST API ' yi farklı bir zamanaşımı değeri kullanacak şekilde yapılandırabilirsiniz.

## Başlamadan önce

Bu görevi tamamlamak için, **dspmweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

- **z/OS** z/OS' da **dspmweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
- **Multi** Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.
- **V 9.3.5** **Linux** mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.



### Uyarı: **z/OS**

z/OS üzerinde **setmqweb** ya da **dspmweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory*, crtmqweb' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).



### Uyarı: **V 9.3.5** **Linux**

Bağımsız bir IBM MQ Web Server kuruluşunda **setmqweb** ya da **dspmweb** komutlarını vermeden önce, **MQ\_OVERRIDE\_DATA\_PATH** ortam değişkenini IBM MQ Web Server veri dizinine ayarlamanız gerekir.

## Yordam

- Aşağıdaki komutu kullanarak istek zaman aşımının yürürlükteki yapılanışını görüntüleyin:  
`dspmweb properties -a`  
mqRestRequestTimeout alanı, yanıt zaman aşımı için geçerli değeri gösterir. Ek bilgi için bkz. [dspmweb özellikleri](#).
- Aşağıdaki komutu kullanarak istek zaman aşımını ayarlayın:  
`setmqweb properties -k mqRestRequestTimeout -v timeout`  
Burada *zaman aşımı*, zaman aşımından önceki süreyi saniye cinsinden belirtir.
- Aşağıdaki komutu kullanarak istek zaman aşımını 30 saniyelik varsayılan değere sıfırlayın:  
`setmqweb properties -k mqRestRequestTimeout -d`

## Otomatik başlatma yapılandırılıyor

Varsayılan olarak, mqweb sunucusu başlatıldığında IBM MQ Console otomatik olarak başlatılır. **setmqweb** komutunu kullanarak IBM MQ Console ve REST API ' un otomatik olarak başlatılıp başlatılmayacağını yapılandırabilirsiniz.

## Başlamadan önce

Bu görevi tamamlamak için, **dspmweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

- **z/OS** z/OS' da **dspmweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
- **Multi** Diğer tüm işletim sistemlerinde [ayrıcılık kullanmanız](#) gerekir.
- **V 9.3.5** **Linux** mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.



**Uyarı:** **z/OS**

z/OS üzerinde **setmqweb** ya da **dspmweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory*, c1tmqweb' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).



**Uyarı:** **V 9.3.5** **Linux**

Bağımsız bir IBM MQ Web Server kuruluşunda **setmqweb** ya da **dspmweb** komutlarını vermeden önce, **MQ\_OVERRIDE\_DATA\_PATH** ortam değişkenini IBM MQ Web Server veri dizinine ayarlamanız gerekir.

## Yordam

- Aşağıdaki komutu kullanarak otomatik başlatmanın yürürlükteki yapılanışını görüntüleyin:  
`dspmweb properties -a`  
mqRestAutostart alanı, REST API 'in otomatik olarak başlatılıp başlatılmadığını gösterir ve mqConsoleAutostart alanı IBM MQ Console ' un otomatik olarak başlatılıp başlatılmadığını gösterir.
- Aşağıdaki komutu kullanarak IBM MQ Console ' in otomatik olarak başlatılıp başlatılmayacağını yapılandırın:  
`setmqweb properties -k mqConsoleAutostart -v start`  
Burada *start* , IBM MQ Console ' in otomatik olarak başlamasını istiyorsanız *true* , tersi durumda *false* değeridir.
- Aşağıdaki komutu kullanarak REST API ' in otomatik olarak başlatılıp başlatılmayacağını yapılandırın:  
`setmqweb properties -k mqRestAutostart -v start`  
Burada *start* , REST API ' in otomatik olarak başlamasını istiyorsanız *true* , tersi durumda *false* değeridir.

## Günlüğe kaydetmenin yapılandırılması

Günlük kaydı düzeylerini, günlük dosyası boyutu üst sınırını ve IBM MQ Console ve REST API' i barındıran mqweb sunucusu tarafından kullanılan günlük dosyası sayısı üst sınırını yapılandırabilirsiniz. **setmqweb** komutunu kullanarak günlük kaydını yapılandırabilirsiniz.

## Başlamadan önce

Bu görevi tamamlamak için, **dspmweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

- **z/OS** z/OS' da **dspmweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
- **Multi** Diğer tüm işletim sistemlerinde ayrıcılık kullanıcılmanız gerekir.
- **V 9.3.5** **Linux** mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.



**Uyarı:** **z/OS**

z/OS üzerinde **setmqweb** ya da **dspmweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory*, crtmqweb' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. mqweb sunucusu yaratılması.



**Uyarı:** **V 9.3.5** **Linux**

Bağımsız bir IBM MQ Web Server kuruluşunda **setmqweb** ya da **dspmweb** komutlarını vermeden önce, **MQ\_OVERRIDE\_DATA\_PATH** ortam değişkenini IBM MQ Web Server veri dizinine ayarlamanız gerekir.

## Bu görev hakkında

mqweb sunucusu günlük iletilerini ve izlemeyi aşağıdaki günlük dosyalarına yazar:

### console.log ve messages.log

Bu dosyalar, IBM MQ Console, REST API ve bu bileşenleri çalıştıran mqweb sunucusu tarafından yayınlanan iletileri içerir.

### trace.log

Bu dosya, IBM MQ Console ve REST API için izleme içerir. İzleme yalnızca izleme etkinleştirildiyse bu dosyaya yazılır.

mqweb sunucusuna ilişkin günlük dosyalarını aşağıdaki dizinlerden birinde bulabilirsiniz:

- IBM MQ kuruluşunda:

- **Linux** **AIX** AIX ya da Linux sistemlerinde: */var/mqm/web/installations/installationName/servers/mqweb/logs*

- **Windows** Windows üzerinde:  
*MQ\_DATA\_PATH\web\installations\installationName\servers\mqweb\logs*; burada *MQ\_DATA\_PATH*, IBM MQ veri yoludur. Bu yol, IBM MQ kuruluşu sırasında seçilen veri yoludur. Varsayılan olarak bu yol C:\ProgramData\IBM\MQ olur.

- **z/OS** z/OS işletim tarihinde: *WLP\_user\_directory/servers/mqweb/logs*

Burada *WLP\_user\_directory*, **crtmqweb** komut dosyası mqweb sunucusu tanımlamasını oluşturmak için çalıştırıldığında belirtilen dizindir.

- **Linux** **V 9.3.5** Bağımsız bir IBM MQ Web Server kuruluşunda:  
*MQ\_OVERRIDE\_DATA\_PATH/web/installations/MQWEBINST/servers/mqweb/logs*

Burada *MQ\_OVERRIDE\_DATA\_PATH*, **MQ\_OVERRIDE\_DATA\_PATH** ortam değişkeninin gösterdiği IBM MQ Web Server veri dizindir.

mqweb sunucusunda çalışan ileti sistemi REST API koduna ilişkin ileti alışverişi izleme dosyaları aşağıdaki dizinlerden birinde bulunabilir:

- IBM MQ kuruluşunda:

– **Linux** **AIX** AIX ya da Linux sistemlerinde: `/var/mqm/web/installations/  
installationName/servers/mqweb`

– **Windows** Windows üzerinde:  
`MQ_DATA_PATH\web\installations\installationName\servers\mqweb`; burada `MQ_DATA_PATH`, IBM MQ veri yoludur. Bu yol, IBM MQ kuruluşu sırasında seçilen veri yoludur. Varsayılan olarak bu yol `C:\ProgramData\IBM\MQ` olur.

– **z/OS** z/OS işletim tarihinde: `WLP_user_directory/servers/mqweb`

Burada `WLP_user_directory`, `crtmqweb` komut dosyası mqweb sunucusu tanımlamasını oluşturmak için çalıştırıldığında belirtilen dizindir.

- **Linux** **V9.3.5** Bağımsız bir IBM MQ Web Server kuruluşunda:

`MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST/servers/mqweb`

Burada `MQ_OVERRIDE_DATA_PATH`, `MQ_OVERRIDE_DATA_PATH` ortam değişkeninin gösterdiği IBM MQ Web Server veri dizinidir.

İzlemeyi etkinleştirme hakkında daha fazla bilgi için:

- REST API, bkz. [REST API İzleme](#)
- IBM MQ Console, bkz. [IBM MQ Console İzleme](#)

## Yordam

- Aşağıdaki komutu kullanarak REST API günlük kaydının geçerli yapılandırmasını görüntüleyin:

```
dspmweb properties -a
```

- `maxTraceFileSize` alanı, günlük kütüğü büyüklüğü üst sınırını gösterir.
- `maxTraceFiles` alanı, günlük dosyası sayısı üst sınırını gösterir
- `traceSpec` alanı, kullanılan izleme düzeyini gösterir
- `maxMsgTraceFileSize` alanı, ileti alışverişi izleme dosyası büyüklüğü üst sınırını gösterir.
- `maxMsgTraceFiles` alanı, ileti alışverişi izleme kütüğü sayısı üst sınırını gösterir.

- `messages.log` ve `trace.log` dosyalarının boyut üst sınırını yapılandırın:

- Aşağıdaki komutu kullanarak günlük dosyası boyutu üst sınırını ayarlayın:

```
setmqweb properties -k maxTraceFileSize -v size
```

Burada `boyut`, her günlük dosyasının ulaşabileceği MB cinsinden boyutu belirtir.

- Aşağıdaki komutu kullanarak günlük dosyası boyutu üst sınırını 20 MB varsayılan değerine sıfırlayın:

```
setmqweb properties -k maxTraceFileSize -d
```

- `messages.log` ve `trace.log` dosya sayısı üst sınırını yapılandırın:

- Aşağıdaki komutu kullanarak her günlük dosyası sayısı üst sınırını ayarlayın:

```
setmqweb properties -k maxTraceFiles -v max
```

Burada `maks`, dosya sayısı üst sınırını belirtir.

- Aşağıdaki komutu kullanarak, her bir günlük dosyası için varsayılan değer olan 2 değerini sıfırlayın:

```
setmqweb properties -k maxTraceFiles -d
```

- İleti alışverişi izleme dosyası büyüklüğü üst sınırını yapılandırın:

- Aşağıdaki komutu kullanarak ileti alışverişi izleme dosyası büyüklüğü üst sınırını ayarlayın:

```
setmqweb properties -k maxMsgTraceFileSize -v size
```

Burada *boyut* , her ileti alışverişi izleme dosyasının ulaşabileceği MB cinsinden boyutu belirtir.

- Aşağıdaki komutu kullanarak ileti alışverişi izleme dosyası büyüklüğü üst sınırını 200 MB varsayılan değerine ayarlayın:

```
setmqweb properties -k maxMsgTraceFileSize -d
```

- Kullanılacak ileti sistemi izleme dosyası sayısı üst sınırını yapılandırın:

- Aşağıdaki komutu kullanarak ileti alışverişi izlemesi için kullanılacak dosya sayısı üst sınırını belirleyin:

```
setmqweb properties -k maxMsgTraceFiles -v max
```

Burada *maks* , dosya sayısı üst sınırını belirtir.

- Aşağıdaki komutu kullanarak , ileti alışverişi izlemesi için kullanılacak dosya sayısı üst sınırını 5 varsayılan değerine ayarlayın:

```
setmqweb properties -k maxMsgTraceFiles -d
```

- mqweb sunucusunun yazdığı izleme düzeyini yapılandırın:

- Aşağıdaki komutu kullanarak kullanılan izleme belirtimini ayarlayın:

```
setmqweb properties -k traceSpec -v level
```

Burada *düzy* , Çizelge 51 sayfa 804 içinde listelenen değerlerden biridir. Tablo, ayrıntı düzeyini artırarak sıralanan günlük kaydı düzeylerini özetler. Bir günlük kaydı düzeyini etkinleştirdiğinizde, ondan önceki her düzeyi de etkinleştirmiş oluyorsunuz. Örneğin, **\*=warning** günlük kaydı düzeyini etkinleştirirseniz, **\*=severe** ve **\*=fatal** günlük kaydı düzeylerini de etkinleştirirsiniz.

IBM Destek tarafından istendiğinde bu değeri değiştirin.

- Aşağıdaki komutu kullanarak , kullanılan izleme belirtimini **\*=info** varsayılan değerine sıfırlayın:

```
setmqweb properties -k traceSpec -d
```

Çizelge 51. Geçerli günlük kaydı düzeyleri	
Değer	Günlük kaydı düzeyi uygulandı
* =kapalı	Günlüğe kaydetme kapalı.
* =onulmaz	Görev devam edemiyor ve bileşen, uygulama ve sunucu çalışmıyor.
* =ciddi	Görev devam edemez, ancak bileşen, uygulama ve sunucu çalışmaya devam edebilir. Bu düzey, yaklaşan kurtarılamaz bir hatayı da gösterebilir.
* =uyarı	Olası hata ya da yaklaşan hata. Bu düzey, aşamalı bir başarısızlık (örneğin, kaynakların olası sızması) olduğunu da gösterebilir.
* =denetleme	Sunucu durumunu ya da kaynaklarını etkileyen önemli olay
* =bilgi	Genel görev ilerleme durumunu özetleyen genel bilgiler
* =config	Yapılandırma değişikliği ya da durumu
* =ayrıntı	Alt görevin ilerleyişini ayrıntılandıran genel bilgiler
* =iyi	İzleme bilgileri-Genel izleme + yöntem girişi, çıkış ve dönüş değerleri



Çizelge 51. Geçerli günlük kaydı düzeyleri (devamı var)	
Değer	Günlük kaydı düzeyi uygulandı
* =daha ince	İzleme bilgileri-Ayrıntılı izleme
* =en iyi	İzleme bilgileri-Sorunlarda hata ayıklamak için gereken tüm ayrıntıları içeren daha ayrıntılı bir izleme
* =tümü	Tüm olaylar günlüğe kaydedilir

## LTPA belirtecinin yapılandırılması

LTPA belirteçleri, mqweb sunucusuna her istekte kullanıcı adı ve parola kimlik bilgileri sağlamak için bir kullanıcıya ihtiyaç duymayı önlemek için kullanılabilir. **setmqweb** komutunu kullanarak LTPA belirteci tanımlama bilgisinin adını, LTPA kimlik doğrulama belirteçleri için süre bitimi aralığını ve LTPA belirteçlerinin HTTP bağlantıları tarafından kullanılıp kullanılmayacağını yapılandırabilirsiniz.

### Başlamadan önce

Bu görevi tamamlamak için, **dspmqweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

- **z/OS** z/OS' da **dspmqweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
- **Multi** Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.
- **V 9.3.5** **Linux** mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.

**Not:** REST API ile hem IBM MQ Console, hem de simge kimlik doğrulamasını kullanıyorsanız, süre bitimi aralığı paylaşılır.



**Uyarı:** **z/OS**

z/OS üzerinde **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory*, c1tmqweb' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).



**Uyarı:** **V 9.3.5** **Linux**




Bağımsız bir IBM MQ Web Server kuruluşunda **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, **MQ\_OVERRIDE\_DATA\_PATH** ortam değişkenini IBM MQ Web Server veri dizinine ayarlamanız gerekir.



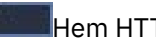
### Bu görev hakkında

Kullanıcılar IBM MQ Console uygulamasında oturum açtığında bir LTPA simgesi oluşturulur. REST API ile belirteç tabanlı kimlik doğrulamasını kullanırsanız, kullanıcı HTTP POST yöntemiyle /login REST API kaynağını kullanarak oturum açtığında bir LTPA simgesi oluşturulur. Bu simge bir tanımlama bilgisinde döndürülür. Belirteç, belirteç süresi doluncaya kadar kullanıcının kullanıcı kimliği ve parolasıyla yeniden

oturum açması gerekmeden kullanıcının kimliğini doğrulamak için kullanılır. Varsayılan süre bitimi aralığı 120 dakikadır.

LTPA simgesini içeren tanımlama bilgisinin adı altyapıya göre değişir:

-  IBM MQ Appliance üzerinde LTPA simgesi `LtpaToken2` olur. Bu değer değiştirilemez.
-   Varsayılan olarak, diğer tüm altyapılarda LTPA simgesini içeren tanımlama bilgisinin adı `LtpaToken2` ile başlar ve `mqweb` sunucusu yeniden başlatıldığında değişebilecek bir sonek içerir. Bu rasgele tanımlama bilgisi adı, aynı sistemde birden çok `mqweb` sunucusunun çalışmasına izin verir. Ancak tanımlama bilgisi adının tutarlı bir değer olarak kalmasını istiyorsanız, **setmqweb** komutunu kullanarak tanımlama bilgisinin sahip olduğu adı belirtebilirsiniz.

   Hem HTTP hem de HTTPS kapılarını etkinleştirirseniz, bir HTTP isteği için HTTPS isteği için yayınlanan bir LTPA simgesi yeniden kullanılabilir. Bu davranış varsayılan olarak devre dışıdır, ancak **setmqweb** komutunu kullanarak bu davranışı etkinleştirebilirsiniz.

## Yordam

- Aşağıdaki komutu kullanarak LTPA belirtecini geçerli süre bitimini, LTPA belirteci tanımlama bilgisinin adını ve LTPA belirtecini HTTP istekleri için kullanılıp kullanılmayacağını görüntüleyin:

```
dspmweb properties -a
```

- `LtpaCookieName` alanı, LTPA belirteci tanımlama bilgisinin adını gösterir. Tanımlama bilgisi adı ayarlamadıysanız, bu özelliğin değeri `AIX, Linux, and Windows`, ya da `LtpaToken2_{{httpsPort}} on z/OS`, üzerinde `LtpaToken2_{{env.MQWEB_LTPA_SUFFIX}}` olur. `LtpaToken2_` önekinden sonraki değişken, tanımlama bilgisi için benzersiz bir ad oluşturmak üzere `mqweb` sunucusu tarafından kullanılır. Bu değişkeni ayarlayamazsınız, ancak `LtpaCookieName` değerini seçtiğiniz bir değere değiştirebilirsiniz.
- `LtpaExpiration` alanı LTPA belirteci süre bitimini gösterir.
- LTPA belirteçleri HTTP istekleri tarafından kullanılabiliriyorsa, `secureLtpa` alanı `false` olarak ayarlanır.

- LTPA belirteci süre bitimini yapılandır:

- Aşağıdaki komutu girerek LTPA belirteci süre bitimini ayarlayın:

```
setmqweb properties -k ltpaExpiration -v time
```

Burada *zaman*, LTPA belirtecini süresi dolmadan ve kullanıcının oturumu kapatılmadan önce geçecek süreyi dakika cinsinden belirtir.

- Aşağıdaki komutu girerek LTPA belirteci süre bitimini varsayılan değer olan 120 dakika değerine sıfırlayın:

```
setmqweb properties -k ltpaExpiration -d
```

-  

LTPA belirteci tanımlama bilgisi adını yapılandır:

- Aşağıdaki komutu girerek LTPA belirteci tanımlama bilgisi adını ayarlayın:

```
setmqweb properties -k ltpaCookieName -v name
```

Burada *ad*, LTPA simgesi tanımlama bilgisi için benzersiz bir ad belirtir.

- Aşağıdaki komutu girerek LTPA belirteci tanımlama bilgisi adını, `LtpaToken2_` önekini rasgele karakterlerin izlediği varsayılan değere sıfırlayın:

```
setmqweb properties -k ltpaCookieName -d
```

-  

Aşağıdaki komutu girerek LTPA simgesinin HTTP bağlantıları tarafından kullanılıp kullanılmayacağını yapılandırın:

```
setmqweb properties -k secureLtpa -v secure
```

Burada *secure* , LTPA simgesinin hem güvenli olmayan HTTP bağlantıları hem de güvenli HTTPS bağlantıları tarafından kullanılıp kullanılmayacağını belirtir. *false* değeri, HTTP ve HTTPS bağlantılarının aynı LTPA simgesini kullanmasını sağlar.

## V 9.3.0 IBM MQ Console için uzak kuyruk yöneticisi bağlantı davranışının yapılandırılması

IBM MQ Console' yi kullandığınızda, uzak kuyruk yöneticilerine bağlantılar yaratabilirsiniz. Yani, IBM MQ Console' u çalıştıran mqweb sunucusu ile aynı kuruluşun parçası olmayan kuyruk yöneticilerine bağlanabilirsiniz. Uzak kuyruk yöneticisi bağlantılarının davranışını denetlemek için ayarlayabileceğiniz bir dizi yapılandırma seçeneği vardır.

### Başlamadan önce

Bu görevi tamamlamak için, **dspmweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

- ▶ **z/OS** z/OS' da **dspmweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
- ▶ **Multi** Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.
- ▶ **V 9.3.5** ▶ **Linux** mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.



**Uyarı:** ▶ **z/OS**

z/OS üzerinde **setmqweb** ya da **dspmweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory* , c1tmqweb' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).



**Uyarı:** ▶ **V 9.3.5** ▶ **Linux**

Bağımsız bir IBM MQ Web Server kuruluşunda **setmqweb** ya da **dspmweb** komutlarını vermeden önce, **MQ\_OVERRIDE\_DATA\_PATH** ortam değişkenini IBM MQ Web Server veri dizinine ayarlamanız gerekir.

### Bu görev hakkında

Aşağıdaki yapılandırma seçeneklerini ayarlayabilirsiniz:

- Uzak kuyruk yöneticisi bağlantılarına izin verilip verilmediğini belirler.
- Bağlantıların IBM MQ Console kullanılarak mı, yoksa yalnızca komut satırıyla mı ekleneceğini belirler.
- Uzak kuyruk yöneticisi bağlantılarına izin verildiğinde IBM MQ Console ' de yerel kuyruk yöneticilerinin görüntülenip görüntülenmeyeceğini belirler.
- Uzak kuyruk yöneticisi bağlantılarının IBM MQ Console başlatıldığında mı, yoksa bir bağlantı hatası olduğunda mı otomatik olarak kurulacağını belirler.
- IBM MQ Console içinde görüntülenen uzak kuyruk yöneticileri listesinin her yenilenmesi arasındaki süre.



## Yordam

- Yürürlükteki uzak kuyruk yöneticisi bağlantısı yapılanış ayarlarını görüntülemek için şu komutu girin:  
`dspmweb properties -a`
  - `mqConsoleRemoteSupportEnabled` alanı, uzak kuyruk yöneticisi bağlantılarına izin verilip verilmediğini belirtir.
  - `mqConsoleRemoteUIAdmin` alanı, IBM MQ Console kullanılarak uzak kuyruk yöneticisi bağlantılarının eklenip eklenemeyeceğini belirtir.
  - `mqConsoleRemoteAllowLocal` alanı, yerel kuyruk yöneticilerinin görüntülenip görüntülenmeyeceğini belirtir.
  - `mqConsoleRemotePollTime` alanı, uzak kuyruk yöneticileri listesinin her yenilenmesi arasında kaç saniye olduğunu gösterir.

- IBM MQ Console ile uzak kuyruk yöneticisi bağlantılarını önlemek ya da izin vermek için şu komutu girin:

```
setmqweb properties -k mqConsoleRemoteSupportEnabled -v true or false
```

Burada `true` , uzak kuyruk yöneticisi bağlantılarına izin verir ya da `false` uzak kuyruk yöneticisi bağlantılarını engeller.

**Not:**   `mqweb` sunucusu bağımsız bir IBM MQ Web Server kuruluşunda çalışıyorsa, **`mqConsoleRemoteSupportEnabled`** özelliği geçerli değildir. Bağımsız IBM MQ Web Server , yalnızca uzak kuyruk yöneticilerine yönelik bağlantıları destekler.

- Uzak kuyruk yöneticisi bağlantılarının IBM MQ Console kullanılarak ya da yalnızca komut satırıyla eklenmesini önlemek ya da izin vermek için aşağıdaki komutu girin:



```
setmqweb properties -k mqConsoleRemoteUIAdmin -v true or false
```

Burada `true` , IBM MQ Console ve komut satırı kullanılarak uzak kuyruk yöneticisi bağlantılarının eklenmesini sağlar ya da `false` uzak kuyruk yöneticisi bağlantılarının yalnızca komut satırına **`setmqweb remote`** komutu kullanılarak eklenmesini sağlar.

- Uzak kuyruk yöneticisi bağlantılarına izin verildiğinde IBM MQ Console ' de yerel kuyruk yöneticilerinin görüntülenmesini önlemek ya da izin vermek için şu komutu girin:

```
setmqweb properties -k mqConsoleRemoteAllowLocal -v true or false
```

Burada `true` yerel kuyruk yöneticilerinin görüntülenmesine izin verir ya da `false` yerel kuyruk yöneticilerini gizler.

**Not:**   `mqweb` sunucusu bağımsız bir IBM MQ Web Server kuruluşunda çalışıyorsa, **`mqConsoleRemoteAllowLocal`** özelliği geçerli değildir. Bağımsız IBM MQ Web Server , yalnızca uzak kuyruk yöneticilerine yönelik bağlantıları destekler.

- IBM MQ Console içinde görüntülenen uzak kuyruk yöneticileri listesinin her yenilenmesi arasındaki süreyi ayarlamak için aşağıdaki komutu girin:

```
setmqweb properties -k mqConsoleRemotePollTime -v seconds
```

Burada *saniye* değeri, uzak kuyruk yöneticileri listesinin her yenilenmesi arasındaki saniye sayısına ayarlanır.

## İlgili başvurular

[setmqweb](#)



[mqweb](#)

## administrative REST API ağ geçidini yapılandırma



administrative REST API ağ geçidi etkinleştirildiğinde, bir ağ geçidi kuyruk yöneticisini kullanarak REST API ile uzaktan yönetim gerçekleştirebilirsiniz. Varsayılan ağ geçidi kuyruk yöneticisi olarak kullanılan

kuyruk yöneticisini yapılandırabilir ya da **setmqweb** komutunu kullanarak administrative REST API ağ geçidini devre dışı bırakarak uzak yönetimi önleyebilirsiniz.

## Başlamadan önce

**Not:**   mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunda çalışıyorsa, bu görev uygulanamaz. administrative REST API bağımsız bir IBM MQ Web Server kuruluşunda kullanılamaz.

Bu görevi tamamlamak için, **dspmweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

-  z/OS' da **dspmweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
-  Diğer tüm işletim sistemlerinde ayrıcalıklı kullanıcı olmanız gerekir.



### Uyarı:

z/OS üzerinde **setmqweb** ya da **dspmweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory*, c1tmqweb'e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).

## Bu görev hakkında

mqweb sunucusu bir IBM MQ kuruluşunda çalıştığında, administrative REST API ağ geçidi varsayılan olarak etkinleştirilir.

Varsayılan ağ geçidi kuyruk yöneticisi, aşağıdaki deyimlerin her ikisi de doğru olduğunda kullanılır:

- Bir REST isteğinin `ibm-mq-rest-gateway-qmgr` üstbilgisinde kuyruk yöneticisi belirtilmedi.
- REST API kaynak URL adresinde belirtilen kuyruk yöneticisi yerel bir kuyruk yöneticisi değil.

REST API ile uzaktan yönetim hakkında daha fazla bilgi için bkz. [REST API kullanarak uzaktan yönetim](#).

## Yordam

- Aşağıdaki komutu kullanarak administrative REST API ağ geçidinin geçerli yapılandırmasını görüntüleyin:

```
dspmweb properties -a
```

mqRestGatewayEnabled alanı, ağ geçidinin etkinleştirilip etkinleştirilmediğini gösterir ve mqRestGatewayQmgr alanı varsayılan ağ geçidi kuyruk yöneticisinin adını gösterir.

- Aşağıdaki komutu kullanarak administrative REST API ağ geçidinin etkinleştirilip etkinleştirilmediğini yapılandırın:

```
setmqweb properties -k mqRestGatewayEnabled -v enabled
```

Burada *etkin*, administrative REST API ağ geçidini etkinleştirmek için **true** değeri, tersi durumda **false** değeridir.

- Varsayılan ağ geçidi kuyruk yöneticisi olarak kullanılacak kuyruk yöneticisini yapılandırın:
  - Aşağıdaki komutu kullanarak varsayılan ağ geçidi kuyruk yöneticisini ayarlayın:

```
setmqweb properties -k mqRestGatewayQmgr -v qmgrName
```

Burada *qmgrName* , mqweb sunucusuyla aynı kuruluştaki bir kuyruk yöneticisinin adıdır.

- Aşağıdaki komutu kullanarak varsayılan ağ geçidi kuyruk yöneticisinin ayarını geri alın:

```
setmqweb properties -k mqRestGatewayQmgr -d
```

## messaging REST API ' nin yapılandırılması

messaging REST API ürününü çeşitli şekillerde yapılandırabilirsiniz. messaging REST API özelliğini etkinleştirmeyi ya da devre dışı bırakmayı seçebilirsiniz. messaging REST API tarafından kullanılacak havuza yollanmış bağlantı sayısı üst sınırını ve tüm bağlantılar kullanımdayken messaging REST API ' in işleyişini seçebilirsiniz. Bir iletiyi göndermek, almak, göz atmak ya da yayınlamak için messaging REST API komutunu kullanırken yetkilendirme için hangi kullanıcı bağlamının kullanılacağını da seçebilirsiniz.

### Yordam

- [“messaging REST API ' in etkinleştirilmesi” sayfa 810](#)
- [“messaging REST API için bağlantı havuzunun yapılandırılması” sayfa 811](#)
- [V 9.3.2](#)  
[“messaging REST API içinde yetkilendirme için kullanılan kullanıcı bağlamının yapılandırılması” sayfa 814](#)

## messaging REST API ' in etkinleştirilmesi

**setmqweb** komutunu kullanarak messaging REST API ' in etkinleştirilip etkinleştirilmediğini yapılandırabilirsiniz. Varsayılan olarak messaging REST API etkindir.

### Başlamadan önce

Bu görevi tamamlamak için, **dspmweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

- [z/OS](#) z/OS' da **dspmweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
- [Multi](#) Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.
- [V 9.3.5](#) [Linux](#) mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.



**Uyarı:** [z/OS](#)

z/OS üzerinde **setmqweb** ya da **dspmweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory* , cıtmqweb' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).

### Yordam

- Aşağıdaki komutu kullanarak messaging REST API ' in geçerli yapılandırmasını görüntüleyin:

```
dspmqweb properties -a
```

mqRestMessagingEnabled alanı, messaging REST API özelliğinin etkinleştirilip etkinleştirilmediğini gösterir. Değer True ise messaging REST API etkinleştirilir.

- Aşağıdaki komutu kullanarak messaging REST API ' i etkinleştirin:

```
setmqweb properties -k mqRestMessagingEnabled -v true
```

- Aşağıdaki komutu kullanarak messaging REST API ' i devre dışı bırakın:

```
setmqweb properties -k mqRestMessagingEnabled -v false
```

## İlgili görevler

[“messaging REST API için bağlantı havuzunun yapılandırılması” sayfa 811](#)

messaging REST API tarafından kullanılacak havuza yollanmış bağlantı sayısı üst sınırını ve tüm bağlantılar kullanımdayken messaging REST API ' in işleyişini yapılandırabilirsiniz.

[“messaging REST API içinde yetkilendirme için kullanılan kullanıcı bağlamının yapılandırılması” sayfa 814](#)

**V 9.3.2** Bir iletiyi göndermek, almak, göz atmak ya da yayınlamak için messaging REST API komutunu kullanırken yetkilendirme için hangi kullanıcı bağlamının kullanılacağını yapılandırabilirsiniz. Başka bir deyişle, messaging REST API ' da oturum açan kullanıcının mı, yoksa mqweb sunucusunu başlatan kullanıcının mı yetkilendirme için kullanılacağını seçebilirsiniz.

[“messaging REST API için bağlantı kipinin yapılandırılması” sayfa 813](#)

messaging REST API ' i yerel ya da uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırabilirsiniz.

## messaging REST API için bağlantı havuzunun yapılandırılması

messaging REST API tarafından kullanılacak havuza yollanmış bağlantı sayısı üst sınırını ve tüm bağlantılar kullanımdayken messaging REST API ' in işleyişini yapılandırabilirsiniz.

## Başlamadan önce

Bu görevi tamamlamak için, **dspmqweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

- **z/OS** z/OS' da **dspmqweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
- **Multi** Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.
- **V 9.3.5** **Linux** mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.



**Uyarı:** **z/OS**

z/OS üzerinde **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory*, crtmqweb' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).



## Bu görev hakkında

messaging REST API başarımını en iyi duruma getirmek için, IBM MQ kuyruk yöneticilerine yönelik bağlantılar havuza gönderilir. Yani, her REST isteği kendi bağlantısını yaratmak, kullanmak ve silmek yerine, her REST isteği bir bağlantı havuzundan bir bağlantı kullanır. Varsayılan olarak, her kuyruk yöneticisi havuzu için 20 bağlantı vardır ve tüm bağlantılar kullanımdayken istekleri işlemek için üç seçenek arasından seçim yapabilirsiniz:

- messaging REST API , istek için kullanılacak yeni, havuzlanmamış bir bağlantı yaratabilir. Bu davranış varsayılan davranıştır.
- messaging REST API bir hata döndürebilir.
- messaging REST API havuzlanmış bağlantının kullanılabilir olmasını bekleyebilir. Bu bekleme belirsiz bir bekleme.

**setmqweb properties** komutunu kullanarak, tüm bağlantılar kullanımdayken messaging REST API ' in havuzlanmış bağlantı sayısı üst sınırını ve varsayılan davranışını değiştirebilirsiniz.

## Yordam

- Aşağıdaki komutu kullanarak geçerli yapılandırmayı görüntüleyin:

```
dspmqweb properties -a
```

- `mqRestMessagingFullPoolBehavior` alanı, havuz içindeki tüm bağlantılar kullanımdayken messaging REST API ' in davranışını gösterir. Değer `blockise`, messaging REST API bir bağlantının kullanılabilir olmasını beklemelidir. Değer `errorise`, messaging REST API bir hata döndürmelidir. Değer `overflowise`, messaging REST API , kullanmak için havuzlanmamış bir bağlantı yaratmalı ve kullanıldıktan sonra bağlantıyı atmalıdır.
- `mqRestMessagingMaxPoolSize` alanı, bağlantı havuzu boyutu üst sınırını gösterir.
- Aşağıdaki komutu kullanarak havuz içindeki tüm bağlantılar kullanımdayken messaging REST API davranışını yapılandırın:

```
setmqweb properties -k mqRestMessagingFullPoolBehavior -v action
```

Burada *işlem* yapılacak işlemi belirtir. *işlem* aşağıdaki değerlerden biri olabilir:

### **blok**

Havuzdaki tüm bağlantılar kullanımdayken, bir bağlantının kullanılabilir olmasını bekleyin.

### **hata**

Havuzdaki tüm bağlantılar kullanımdayken bir hata döndürür.

### **Taşma**

Havuzdaki tüm bağlantılar kullanımdayken, kullanılacak havuzlanmamış bir bağlantı yaratın ve kullanıldıktan sonra bağlantıyı imha edin.

- Aşağıdaki komutu kullanarak her kuyruk yöneticisi havuzu için bağlantı havuzu büyüklüğü üst sınırını yapılandırın:

```
setmqweb properties -k mqRestMessagingMaxPoolSize -v size
```

Burada *boyut* , havuzun boyutunu belirtir.

**Not:** `mqRestMessagingMaxPoolSize` için büyük bir değer ayarlandıysa ve birçok kuyruk yöneticisine bağlıysa, mqweb sunucusu öbeğinin büyüklük üst sınırını artırmayı düşünebilirsiniz. Daha fazla bilgi için bakınız: [tuning the mqweb server JVM](#) .

## İlgili görevler

“messaging REST API ' in etkinleştirilmesi” sayfa 810

**setmqweb** komutunu kullanarak messaging REST API ' in etkinleştirilip etkinleştirilmediğini yapılandırabilirsiniz. Varsayılan olarak messaging REST API etkindir.



[“messaging REST API içinde yetkilendirme için kullanılan kullanıcı bağlamının yapılandırılması” sayfa 814](#)

**V 9.3.2** Bir iletiyi göndermek, almak, göz atmak ya da yayınlamak için messaging REST API komutunu kullanırken yetkilendirme için hangi kullanıcı bağlamının kullanılacağını yapılandırabilirsiniz. Başka bir deyişle, messaging REST API' da oturum açan kullanıcının mı, yoksa mqweb sunucusunu başlatan kullanıcının mı yetkilendirme için kullanılacağını seçebilirsiniz.

[“messaging REST API için bağlantı kipinin yapılandırılması” sayfa 813](#)  
messaging REST API ' i yerel ya da uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırabilirsiniz.

### **V 9.3.3 messaging REST API için bağlantı kipinin yapılandırılması**

messaging REST API ' i yerel ya da uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırabilirsiniz.

## **Başlamadan önce**

**Not:** **Linux** **V 9.3.5** mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunda çalışıyorsa, bu görev uygulanamaz. Bağımsız IBM MQ Web Server , yalnızca uzak kuyruk yöneticilerine yönelik bağlantıları destekler.

Bu görevi tamamlamak için, **dspmqweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

- **z/OS** z/OS' da **dspmqweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
- **Multi** Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.



**Uyarı:** **z/OS**

z/OS üzerinde **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory* , crtmqweb' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).

## **Bu görev hakkında**

messaging REST API için varsayılan bağlantı kipi, mqweb sunucusunu çalıştıran kuruluş tipine bağlı olarak değişir:

- Bir IBM MQ kuruluşunda, varsayılan olarak messaging REST API yalnızca mqweb sunucusu ile aynı kuruluştaki yerel kuyruk yöneticilerine bağlanır. Bağlantı yapılandırmasını görüntülemek ve değiştirmek için bu görevdeki adımları tamamlayın.
- **Linux** **V 9.3.5** Bağımsız bir IBM MQ Web Server kuruluşunda, messaging REST API yalnızca uzak kuyruk yöneticilerine yönelik bağlantıları destekler. Bağlantı konfigürasyonu görüntülenemiyor ya da değiştirilemiyor.

## **Yordam**

- Aşağıdaki komutu kullanarak messaging REST API ' in geçerli yapılandırmasını görüntüleyin:

```
dspmqweb properties -a
```

mqRestMessagingConnectionMode alanı, geçerli bağlantı kipini gösterir. Değer localise, messaging REST API yalnızca mqweb sunucusu ile aynı kuruluştaki kuyruk yöneticilerine bağlanabilir. Değer remoteise, messaging REST API uzak kuyruk yöneticilerine bağlanabilir.

- mqweb sunucusunu, messaging REST API ' un yalnızca mqweb sunucusu ile aynı kuruluştaki kuyruk yöneticilerine aşağıdaki komutları kullanarak bağlanmasına izin verecek şekilde yapılandırın:

```
setmqweb properties -k mqRestMessagingConnectionMode -v local  
endmqweb  
strmqweb
```

- mqweb sunucusunu, messaging REST API ' un aşağıdaki komutu kullanarak uzak kuyruk yöneticilerine bağlanmasına izin verecek şekilde yapılandırın:

```
setmqweb properties -k mqRestMessagingConnectionMode -v remote  
endmqweb  
strmqweb
```

## Sonraki adım

mqweb sunucusunu, messaging REST API ' un uzak kuyruk yöneticilerine bağlanmasına izin verecek şekilde yapılandırırsanız, bağlanmak istediğiniz her kuyruk yöneticisi için bağlantı bilgileri sağlamanız gerekir. Bağlantı bilgilerinin nasıl sağlanacağıyla ilgili daha fazla bilgi için [messaging REST API ile kullanılmak üzere uzak kuyruk yöneticisinin ayarlanması](#) başlıklı konuya bakın.

### İlgili görevler

[“messaging REST API ' in etkinleştirilmesi” sayfa 810](#)

**setmqweb** komutunu kullanarak messaging REST API ' in etkinleştirilip etkinleştirilmediğini yapılandırabilirsiniz. Varsayılan olarak messaging REST API etkindir.

[“messaging REST API için bağlantı havuzunun yapılandırılması” sayfa 811](#)

messaging REST API tarafından kullanılacak havuza yollanmış bağlantı sayısı üst sınırını ve tüm bağlantılar kullanımdayken messaging REST API ' in işleyişini yapılandırabilirsiniz.

[“messaging REST API içinde yetkilendirme için kullanılan kullanıcı bağlamının yapılandırılması” sayfa 814](#)

**V 9.3.2** Bir iletiyi göndermek, almak, göz atmak ya da yayınlamak için messaging REST API komutunu kullanırken yetkilendirme için hangi kullanıcı bağlamının kullanılacağını yapılandırabilirsiniz. Başka bir deyişle, messaging REST API ' da oturum açan kullanıcının mı, yoksa mqweb sunucusunu başlatan kullanıcının mı yetkilendirme için kullanılacağını seçebilirsiniz.

## **V 9.3.2 messaging REST API içinde yetkilendirme için kullanılan kullanıcı bağlamının yapılandırılması**

**V 9.3.2** Bir iletiyi göndermek, almak, göz atmak ya da yayınlamak için messaging REST API komutunu kullanırken yetkilendirme için hangi kullanıcı bağlamının kullanılacağını yapılandırabilirsiniz. Başka bir deyişle, messaging REST API ' da oturum açan kullanıcının mı, yoksa mqweb sunucusunu başlatan kullanıcının mı yetkilendirme için kullanılacağını seçebilirsiniz.

## Başlamadan önce

Bu görevi tamamlamak için, **dspmqweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

- z/OS** z/OS' da **dspmqweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
- Multi** Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.
- V 9.3.5 Linux** mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.



## Uyarı: z/OS

z/OS üzerinde **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory*, *crtmqweb*' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).

## Bu görev hakkında

- Kullanılan kullanıcı kimliği messaging REST API'de oturum açan kullanıcı kimliği olduğunda, **MQMD.UserIdentifier** , REST API' de oturum açan kullanıcı kimliğine ayarlanır. **MQMD.AppIdentityData** , REST API 'sinde oturum açan kullanıcı kimliğine ayarlanır.
- Kullanılan kullanıcı kimliği mqweb sunucusunu başlatan kullanıcı kimliği olduğunda, **MQMD.UserIdentifier** boş bırakılır. **MQMD.AppIdentityData** , REST API 'sinde oturum açan kullanıcı kimliğine ayarlanır.

IBM MQ iletisinin ileti tanımlayıcı kısımlarıyla ilgili ek bilgi için [MQMD](#) kısmına bakın.

## Yordam

- Aşağıdaki komutu kullanarak messaging REST API ' in geçerli yapılandırmasını görüntüleyin:

```
dspmqweb properties -a
```

**mqRestMessagingAdoptWebUserContext** alanı, iletileri gönderirken, yayınlarken, alırken ya da göz atarken yetkilendirme için hangi kullanıcı kimliğinin kullanıldığını gösterir. Değer **True** ise, yetkilendirme için messaging REST API ' ta oturum açan kullanıcı kullanılır. Değer **False** ise, yetkilendirme için mqweb sunucusunu başlatan kullanıcı kullanılır.

- Aşağıdaki komutu kullanarak yetkilendirme için messaging REST API 'ta oturum açan kullanıcının kullanıcı kimliğini kullanmak üzere messaging REST API ' i yapılandırın:

```
setmqweb properties -k mqRestMessagingAdoptWebUserContext -v true
```

**mqRestMessagingAdoptWebUserContext** **true** olarak ayarlandığında, **MQMD.UserIdentifier** , REST API ' de oturum açan kullanıcı kimliğine ayarlanır. **MQMD.AppIdentityData** , REST API 'sinde oturum açan kullanıcı kimliğine ayarlanır.

- messaging REST API ' i, aşağıdaki komutu kullanarak mqweb sunucusunu başlatan kullanıcının kullanıcı kimliğini kullanacak şekilde yapılandırın:

```
setmqweb properties -k mqRestMessagingAdoptWebUserContext -v false
```

**mqRestMessagingAdoptWebUserContext** **false** olarak ayarlandığında, **MQMD.UserIdentifier** boş bırakılır. **MQMD.AppIdentityData** , REST API 'sinde oturum açan kullanıcı kimliğine ayarlanır.

## İlgili görevler

“messaging REST API ' in etkinleştirilmesi” sayfa 810

**setmqweb** komutunu kullanarak messaging REST API ' in etkinleştirilip etkinleştirilmediğini yapılandırabilirsiniz. Varsayılan olarak messaging REST API etkindir.

“messaging REST API için bağlantı havuzunun yapılandırılması” sayfa 811

messaging REST API tarafından kullanılacak havuza yollanmış bağlantı sayısı üst sınırını ve tüm bağlantılar kullanımdayken messaging REST API ' in işleyişini yapılandırabilirsiniz.

“messaging REST API için bağlantı kipinin yapılandırılması” sayfa 813

messaging REST API ' i yerel ya da uzak kuyruk yöneticilerine bağlanacak şekilde yapılandırabilirsiniz.

## MFT için REST API ' in yapılandırılması

Varsayılan olarak, MFT için REST API etkinleştirilmez. REST API for MFT olanağının etkinleştirilip etkinleştirilmediğini yapılandırabilir, eşgüdüm kuyruğu yöneticisini ayarlayabilir, komut kuyruğu yöneticisini ayarlayabilir ve **setmqweb properties** komutunu kullanarak MFT yeniden bağlanma zamanışımını belirtebilirsiniz.


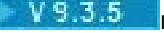
### Yordam

- “MFT için REST API ' in etkinleştirilmesi” sayfa 816
- “MFT için REST API koordinasyon kuyruk yöneticisini yapılandırma” sayfa 817
- “MFT için REST API komut kuyruğu yöneticisinin yapılandırılması” sayfa 818
- “REST API öğesinin MFT zamanışımı değerleri için yapılandırılması” sayfa 819



## MFT için REST API ' in etkinleştirilmesi

MFT için REST API ' i kullanmadan önce, MFT için REST API ' yi etkinleştirmeniz gerekir. **setmqweb** komutunu kullanarak MFT için REST API ' in etkinleştirilip etkinleştirilmediğini yapılandırabilirsiniz. Varsayılan olarak, MFT için REST API etkinleştirilmez.

### Başlamadan önce

**Not:**   mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunda çalışıyorsa, bu görev uygulanamaz. MFT için REST API bağımsız bir IBM MQ Web Server kuruluşunda kullanılamaz.

Bu görevi tamamlamak için, **dspmqweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

-  z/OS' da **dspmqweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
-  Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.



**Uyarı:** 

z/OS üzerinde **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory* , c1mqweb' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).

### Yordam

1. Aşağıdaki komutu kullanarak MFT için geçerli REST API yapılandırmasını görüntüleyin:

```
dspmqweb properties -a
```

mqRestMftEnabled alanı, MFT için REST API özelliğinin etkinleştirilip etkinleştirilmediğini gösterir. Değer, MFT için REST API etkinleştirildiyse True , tersi durumda False olur.

2. Aşağıdaki komutlardan birini kullanarak MFT için REST API ' i etkinleştirin ya da devre dışı bırakın:

- Aşağıdaki komutu kullanarak MFT için REST API ' i etkinleştirin:

```
setmqweb properties -k mqRestMftEnabled -v true
```

- Aşağıdaki komutu kullanarak MFT için REST API ' i devre dışı bırakın:

```
setmqweb properties -k mqRestMftEnabled -v false
```

3. Aşağıdaki komutları girerek mqweb sunucusunu yeniden başlatın:

```
endmqweb  
strmqweb
```



## Sonraki adım

MFT için REST API 'i etkinleştirdiyseniz, MFT için REST API 'yi kullanmadan önce eşgüdüm kuyruğu yöneticisi adını ayarlamanız gerekir. Eşgüdüm kuyruk yöneticisini ayarlama hakkında daha fazla bilgi için bkz. [“MFT için REST API koordinasyon kuyruk yöneticisini yapılandırma” sayfa 817.](#)



## MFT için REST API koordinasyon kuyruk yöneticisini yapılandırma

REST API for MFT programını kullanmadan önce, MFT hareketlerine ilişkin eşgüdüm kuyruğu yöneticisi olarak işlev görecektir bir kuyruk yöneticisi yapılandırmanız gerekir. **setmqweb** komutunu kullanarak hangi kuyruk yöneticisinin koordinasyon kuyruğu yöneticisi olduğunu ayarlayabilirsiniz.

## Başlamadan önce

**Not:**   mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunda çalışıyorsa, bu görev uygulanamaz. MFT için REST API bağımsız bir IBM MQ Web Server kuruluşunda kullanılamaz.

Bu görevi tamamlamak için, **dspmqweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

-  z/OS' da **dspmqweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
-  Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.



**Uyarı:** 

z/OS üzerinde **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory* , c1tmqweb' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması.](#)

## Yordam

1. Aşağıdaki komutu kullanarak MFT için geçerli REST API yapılandırmasını görüntüleyin:

```
dspmqweb properties -a
```

mqRestMftCoordinationQmgr alanı, koordinasyon kuyruk yöneticisinin adını gösterir.

2. Aşağıdaki komutu kullanarak eşgüdüm kuyruğu yöneticisini yapılandırın:

```
setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
```

Burada *qmgrName* , eşgüdüm kuyruğu yöneticisinin adıdır. Koordinasyon kuyruk yöneticisi, mqweb sunucusunun çalıştığı makinede olmalıdır. Varsayılan olarak, bu kuyruk yöneticisi adı boştur. Bir değer ayarlanmazsa, MFT için REST API çalışmaz.

3. Aşağıdaki komutları girerek mqweb sunucusunu yeniden başlatın:

```
endmqweb  
startmqweb
```


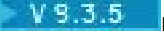
## Sonraki adım

- MFT için REST API ' in etkinleştirildiğinden emin olun. Daha fazla bilgi için bkz [“MFT için REST API ' in etkinleştirilmesi”](#) sayfa 816.
- Yaratma istekleri sunmak için REST API for MFT kullanmak istiyorsanız, komut kuyruğu yöneticisi adını ayarlamanız gerekir. Örneğin, **create transfer** gibi bir REST API komutu kullanmak istiyorsanız, komut kuyruğu yöneticisi adını ayarlamanız gerekir. Daha fazla bilgi için bkz [“MFT için REST API komut kuyruğu yöneticisinin yapılandırılması”](#) sayfa 818.
- REST API ögesini MFT zamanaşımı değerleri için yapılandırabilirsiniz. Varsayılan zamanaşımı 30 dakikadır. Daha fazla bilgi için bkz [“REST API ögesinin MFT zamanaşımı değerleri için yapılandırılması”](#) sayfa 819.
- MFT için REST API ' i kullanmak üzere bir kullanıcının kimliği mqweb sunucusunda doğrulanmalı ve MFTWebAdminya da MFTWebAdminRO rollerinden birinin ya da daha fazlasının üyesi olmalıdır. Kullanıcıları yapılandırma hakkında daha fazla bilgi için [REST API için kullanıcıları ve rolleri yapılandırmaya](#) başlıklı konuya bakın.



## MFT için REST API komut kuyruğu yöneticisinin yapılandırılması

Yaratma istekleri sunmak üzere REST API for MFT programını kullanabilmek için önce komut kuyruğu yöneticisi adını ayarlamanız gerekir. Örneğin, **create transfer** kaynağını kullanmak için komut kuyruğu yöneticisi adını ayarlamanız gerekir. **setmqweb** komutunu kullanarak komut kuyruğu yöneticisi adını ayarlayabilirsiniz.

## Başlamadan önce

**Not:**   mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunda çalışıyorsa, bu görev uygulanamaz. MFT için REST API bağımsız bir IBM MQ Web Server kuruluşunda kullanılamaz.

Bu görevi tamamlamak için, **dspmweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

-  z/OS' da **dspmweb** ve **setmqweb** komutlarını çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
-  Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.



**Uyarı:** 

z/OS üzerinde **setmqweb** ya da **dspmweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamanız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada `WLP_user_directory` , `crtmqweb` ' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).

## Yordam

1. Aşağıdaki komutu kullanarak MFT için geçerli REST API yapılandırmasını görüntüleyin:

```
dspmqweb properties -a
```

`mqRestMftCommandQmgr` alanı, komut kuyruğu yöneticisinin adını gösterir.

2. Aşağıdaki komutu kullanarak komut kuyruğu yöneticisini yapılandırın:

```
setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
```

Burada `qmgrName` , komut kuyruğu yöneticisinin adıdır. Komut kuyruğu yöneticisi, `mqweb` sunucusunun çalıştığı makinede olmalıdır. Varsayılan olarak, bu kuyruk yöneticisi adı boştur. Bir değer ayarlanmazsa, yaratma komutuna ilişkin REST API for MFT çalışmaz.

3. Aşağıdaki komutları girerek `mqweb` sunucusunu yeniden başlatın:

```
endmqweb  
strmqweb
```



## Sonraki adım

- MFT için REST API ' in etkinleştirildiğinden emin olun. Daha fazla bilgi için bkz [“MFT için REST API ' in etkinleştirilmesi”](#) sayfa 816.
- Bir eşgüdüm kuyruğu yöneticisinin ayarlandığından emin olun. Daha fazla bilgi için bkz [“MFT için REST API koordinasyon kuyruk yöneticisini yapılandırma”](#) sayfa 817.
- REST API ögesini MFT zamanaşımı değerleri için yapılandırabilirsiniz. Varsayılan zamanaşımı 30 dakikadır. Daha fazla bilgi için bkz [“REST API ögesinin MFT zamanaşımı değerleri için yapılandırılması”](#) sayfa 819.
- MFT için REST API ' i kullanmak üzere bir kullanıcının kimliği `mqweb` sunucusunda doğrulanmalı ve `MFTWebAdmin` ya da `MFTWebAdminRO` rollerinden birinin ya da daha fazlasının üyesi olmalıdır. Kullanıcıları yapılandırma hakkında daha fazla bilgi için [REST API için kullanıcıları ve rolleri yapılandırma](#) başlıklı konuya bakın.



## REST API ögesinin MFT zamanaşımı değerleri için yapılandırılması

Bağlantı kesildikten sonra, REST API for MFT olanağının koordinasyon kuyruğu yöneticisine bağlanmayı denemeyi durduracağı süreyi dakika cinsinden yapılandırabilirsiniz. Varsayılan zamanaşımı 30 dakikadır. Bu zamanaşımını `setmqweb` komutunu kullanarak yapılandırabilirsiniz.

## Başlamadan önce

**Not:**   `mqweb` sunucusu bağımsız bir IBM MQ Web Server kuruluşunda çalışıyorsa, bu görev uygulanamaz. MFT için REST API bağımsız bir IBM MQ Web Server kuruluşunda kullanılamaz.

Bu görevi tamamlamak için, `dspmqweb` ve `setmqweb` komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

-  `z/OS` ' da `dspmqweb` ve `setmqweb` komutlarını çalıştırma ve `mqwebuser.xml` dosyasına yazma erişiminiz olmalıdır.
-  Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.





## Uyarı: z/OS

z/OS üzerinde **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, WLP\_USER\_DIR ortam değişkenini, değişkenin mqweb sunucusu yapılandırmanızı işaret etmesi için ayarlamamız gerekir.

WLP\_USER\_DIR ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada *WLP\_user\_directory* , *crtmqweb*' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).

## Bu görev hakkında

MFT için REST API zaman aşımını yapılandırabilirsiniz.

REST API for MFT , koordinasyon kuyruk yöneticisiyle bağlantı kesildikten hemen sonra bağlantıyı yeniden kurmayı dener. Bu girişim başarısız olursa, zamanaşımı geçinceye kadar her yeniden bağlanma girişimi arasında beş dakika ara vardır. Bu nedenle, 0-5 arasında bir değer ayarlanması yeniden bağlanma girişimine neden olur.

Yeniden bağlanma zamanaşımına uğradıktan sonra, REST API for MFT kaynaklarından herhangi biri çağrıldığında yeniden bağlanma girişiminde bulunilir. Bu yeniden bağlanma girişimi başarısız olursa, MFT yeniden bağlanma zamanaşımı geçinceye kadar her beş dakikada bir yeniden bağlanma girişiminde bulunur.

## Yordam

1. Aşağıdaki komutu kullanarak MFT için geçerli REST API yapılandırmasını görüntüleyin:

```
dspmqweb properties -a
```

`mqRestMftReconnectTimeoutInMinutes` alanı, MFT Aktarma Rest hizmetleri eşgüdüm kuyruğu yöneticisine bağlanmayı denemeyi durduruncaya kadar yeniden bağlantı zamanaşımı değerini gösterir.

2. REST API for MFT olanağının eşgüdüm kuyruk yöneticisine bağlanmayı denemeyi durdurması için dakika cinsinden zamanaşımını yapılandırın:
  - Zamanaşımını 30 dakikalık varsayılan değere sıfırlayın:

```
setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -d
```

- Zamanaşımını ayarla:

```
setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -v time
```

Burada *zaman* , zamanaşımı oluşmadan önceki süreyi dakika cinsinden belirtir.

Bu değer 0 -5 arasında ayarlanırsa, MFT için REST API , eşgüdüm kuyruğu yöneticisine yalnızca bir kez yeniden bağlanmayı dener. Bağlantı başarısız olursa, REST API çağrılıncaya kadar bağlantıyı yeniden kurma girişiminde bulunulmaz.

Bu değer -1 olarak ayarlanırsa, MFT için REST API , bağlantı başarılı oluncaya kadar yeniden bağlanmayı dener.

3. Aşağıdaki komutları girerek mqweb sunucusunu yeniden başlatın:

```
endmqweb  
strmqweb
```



## Sonraki adım

- MFT için REST API ' in etkinleştirildiğinden emin olun. Daha fazla bilgi için bkz “[MFT için REST API ' in etkinleştirilmesi](#)” sayfa 816.
- Bir eşgüdüm kuyruğu yöneticisinin ayarlandığından emin olun. Daha fazla bilgi için bkz “[MFT için REST API koordinasyon kuyruk yöneticisini yapılandırma](#)” sayfa 817.
- Yaratma istekleri sunmak için REST API for MFT kullanmak istiyorsanız, komut kuyruğu yöneticisi adını ayarlamanız gerekir. Örneğin, **create transfer** gibi bir REST API komutu kullanmak istiyorsanız, komut kuyruğu yöneticisi adını ayarlamanız gerekir. Daha fazla bilgi için bkz “[MFT için REST API komut kuyruğu yöneticisinin yapılandırılması](#)” sayfa 818.
- MFT için REST API ' i kullanmak üzere bir kullanıcının kimliği mqweb sunucusunda doğrulanmalı ve MFTWebAdminya da MFTWebAdminRO rollerinden birinin ya da daha fazlasının üyesi olmalıdır. Kullanıcıları yapılandırma hakkında daha fazla bilgi için [REST API için kullanıcıları ve rolleri yapılandırma](#) başlıklı konuya bakın.

## mqweb sunucusu JVM 'sinin ayarlanması

Varsayılan olarak, mqweb sunucusu Java Virtual Machine (JVM), öbek büyüklüğü alt sınırı ve üst sınırı ve sınıf önbelleği büyüklüğü gibi yapılanış değiştirgeleri için altyapıya özgü varsayılan değerleri kullanır.

### Bu görev hakkında

Performansı artırmak ya da sorunları çözmek için varsayılan değerleri değiştirmeniz gerekebilir. Örneğin, mqweb sunucusu tarafından bir `java.lang.OutOfMemoryError` yayınlandıysa, yığın boyutu üst sınırını artırmamız gerekir. Çok sayıda kuyruk nesnesi yüklemeye çalışıyorsanız, öbek büyüklüğünü de artırmamız gerekir.

IBM MQ Console içinde gösterge panosu yapılandırma bilgilerinin görüntülenmesiyle ilgili sorunlarla karşılaşıyorsanız, yapılandırmanın dosya kodlamasını belirleyen bir değişken ayarlamanız gerekir. `jvm.options` dosyasındaki varsayılan değerleri değiştirebilirsiniz.

### Yordam

1. `jvm.options` dosyasını açın.

`jvm.options` dosyası aşağıdaki dizinlerden birinde bulunabilir:

- IBM MQ kuruluşunda:

– **Linux** / **AIX** AIX ya da Linux sistemlerinde: `/var/mqm/web/installations/installationName/servers/mqweb`

– **Windows** Windows üzerinde:  
`MQ_DATA_PATH\web\installations\installationName\servers\mqweb`; burada `MQ_DATA_PATH`, IBM MQ veri yoludur. Bu yol, IBM MQ kuruluşu sırasında seçilen veri yoludur. Varsayılan olarak bu yol `C:\ProgramData\IBM\MQ` olur.

– **IBM i** IBM işletim tarihinde: `MQ_DATA_PATH/web/installations/Installation1/`

– **z/OS** z/OS işletim tarihinde: `WLP_user_directory/servers/mqweb`

Burada `WLP_user_directory`, `crtmqweb` komut dosyası mqweb sunucusu tanımlamasını oluşturmak için çalıştırıldığında belirtilen dizindir.

- **Linux** / **V 9.3.5** Bağımsız bir IBM MQ Web Server kuruluşunda:  
`MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST/servers/mqweb`

Burada `MQ_OVERRIDE_DATA_PATH`, `MQ_OVERRIDE_DATA_PATH` ortam değişkeninin gösterdiği IBM MQ Web Server veri dizinidir.

2. İsteğe bağlı: Dosyaya aşağıdaki satırı ekleyerek yığın boyutu üst sınırını ayarlayın:

```
-XmxMaxSize
```

Burada *MaxSize* , MB cinsinden yığın boyutu üst sınırını belirtir.

Örneğin, aşağıdaki satır yığın boyutu üst sınırını 1GB:

```
-Xmx1024m
```

3. İsteğe bağlı: Dosyaya aşağıdaki satırı ekleyerek yığın boyutu alt sınırını ayarlayın:

```
-XmsMinSize
```

Burada *MinSize* , MB cinsinden yığın boyutu alt sınırını belirtir. Öbek büyüklüğü alt sınırının varsayılan değerden artırılması, mqweb sunucusunu başlatmak için gereken süreyi azaltabilir.

Örneğin, aşağıdaki satır yığın boyutu alt sınırını 512MB: olarak ayarlar:

```
-Xms512m
```

4. İsteğe bağlı: Dosyaya aşağıdaki satırı ekleyerek sınıf önbelleğinin boyutunu ayarlayın:

```
-XscmxSize
```

Burada *Boyut* , MB cinsinden sınıf önbelleği boyutunu belirtir.

Örneğin, aşağıdaki satır sınıf önbelleği boyutunu 100MB: olarak ayarlar:

```
-Xscmx100m
```

Java paylaşılan sınıf önbelleği, yüklenen sınıflar ve AOT (AOT) derlenmiş kodu gibi verileri saklamak için kullanılır.

Sınıf önbelleği, mqweb sunucusunu başlatma süresini önemli ölçüde kısaltır. mqweb sunucusunun ilk başlatılışında sınıf önbelleği yaratılır ve sunucunun başlatılması önemli bir zaman alabilir. Sınıflar paylaşılan sınıf önbelleğinden yüklenebildiği için, sunucunun sonraki yeniden başlatması çok daha hızlı olur.

Sınıf önbelleğinin varsayılan boyutunun artırılması, mqweb sunucusunun başlatılması için gereken süreyi azaltabilir.

**z/OS** Sınıf önbelleği, mqweb sunucusu farklı bir z/OS sisteminde başlatıldığında yeniden oluşturulur. Bu nedenle, sistem şebekesinde farklı bir z/OS sisteminde mqweb sunucusunun başlatılması, sunucunun aynı sistemde yeniden başlatılmasından çok daha uzun sürebilir.

Bu değerde yapılan değişikliklerin yalnızca sınıf önbelleği yaratıldığında geçerli olduğunu unutmayın. Sınıf önbelleği, mqweb sunucusu ilk başlatıldığında ya da sınıf önbelleği Java sınıf önbelleği yardımcı programı kullanılarak yok edildikten sonra yaratılır.

5. Gerekli: Dosyanın, REST API verileri işlediğinde ve IBM MQ Console'indeki kullanıcı gösterge panosu yapılandırma bilgileri için kullanılan dosya kodlamasını belirtmek üzere aşağıdaki satırları içerip içermediğini denetleyin:

```
-Dfile.encoding=UTF-8  
-Ddefault.client.encoding=UTF-8
```

6. mqweb sunucusunu yeniden başlatın.

**z/OS** z/OS' da, mqweb sunucusunun başlattığı görevi durdurun ve yeniden başlatın.

**Multi** Diğer tüm altyapılarda, komut satırına aşağıdaki komutları girin:

```
endmqweb  
startmqweb
```

## IBM MQ Console ve REST API kuruluş bileşeninin dosya yapısı

IBM MQ Console ve REST API kuruluş bileşeniyle ilişkilendirilmiş iki dizin yapısı kümesi vardır. Bir dizin yapısı, düzenlenebilecek dosyaları içerir. Diğer dizin yapısı düzenlenemeyen dosyalar içeriyor.

### Düzenlenebilir dosyalar

Kullanıcı tarafından düzenlenebilir dosyalar, IBM MQ Console ve REST API kuruluş bileşeninin ilk kuruluşunun bir parçası olarak yerleştirilmiştir. Bu dosyalar düzenlenebildiğinden, bakım uygulandığında dosyalar değiştirilmez.

Kullanıcı tarafından düzenlenebilir dosyaların konumu, işletim sistemine ve kuruluürününe bağlıdır.

- Bir IBM MQ kuruluşunda, kullanıcı tarafından düzenlenebilir dosyalar aşağıdaki dizinlerden birinde bulunur:

- **Linux** **AIX** AIX ya da Linux sistemlerinde: `/var/mqm/web/installations/installationName`
- **Windows** Windows üzerinde: `MQ_DATA_PATH\web\installations\installationName`; burada `MQ_DATA_PATH`, IBM MQ veri yoludur. Bu yol, IBM MQ kuruluşu sırasında seçilen veri yoludur. Varsayılan olarak bu yol `C:\ProgramData\IBM\MQ` olur.
- **z/OS** z/OS' da: `crtmqweb` komut dosyası mqweb sunucusu tanımlamasını yaratmak için çalıştırıldığında belirtilen dizin.

- **Linux** **V9.3.5** Bağımsız bir IBM MQ Web Server kuruluşunda: `MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST`

Burada `MQ_OVERRIDE_DATA_PATH`, `MQ_OVERRIDE_DATA_PATH` ortam değişkeninin gösterdiği IBM MQ Web Server veri dizinidir.

Bu üst düzey dizin altında aşağıdaki dizinler ve dosyalar bulunur:

Dizinler ve dosyalar	Açıklama
<code>angular.persistence/</code>	IBM MQ Console gösterge panosu yapılandırmasının depolandığı dizin.
<code>servers/</code>	WebSphere Liberty sunucuları dizini.
<code>servers/mqweb</code>	mqweb sunucusu dizin yapısını içeren dizin.
<code>servers/mqweb/logs</code>	mqweb sunucusuna ilişkin günlükleri içeren dizin.
<code>servers/mqweb/logs/console.log</code>	Temel sunucu durumu ve işlem iletilerinin günlüğü.
<code>servers/mqweb/logs/ffdc</code>	First Failure Data Capture (FFDC) çıkış dizini.
<code>servers/mqweb/logs/messages.log</code>	IBM MQ Console ve REST API'da içinde olmak üzere, mqweb sunucusundan gelen yürütme zamanı iletilerinin günlüğü. Eski iletiler, <code>messages_timestamp.log</code> adlı verilen dosyalarda saklanır.
<code>servers/mqweb/logs/trace.log</code>	IBM MQ Console ve REST API'da içinde olmak üzere, mqweb sunucusundan izleme günlüğü. Daha eski izleme, <code>trace_timestamp.log</code> adlı dosyalarda saklanır. Bu dosyalar yalnızca izleme etkinleştirildiyse var olur.
<code>servers/mqweb/logs/state</code>	Sunucuya özgü durum.
<code>servers/mqweb/server.xml</code>	Ana sunucu yapılandırma dosyası.




Dizinler ve dosyalar	Açıklama
	Bu dosya salt okunur. Varsayılan yapılandırmayı geçersiz kılmak için <code>mqwebuser.xml</code> dosyasını düzenleyin.
<code>servers/mqweb/mqwebuser.xml</code>	IBM MQ Console ve REST API için yapılandırma dosyası. Bu dosyada yapılandırılan ayarlar, varsayılan yapılandırmayı geçersiz kılar.  Bu dosyayı düzenlemek için <u>ayrıcalıklı kullanıcı</u> olmanız gerekir.
<code>servers/mqweb/resources</code>	Anahtar depoları gibi çeşitli sunucu kaynaklarını içeren dizin.
<code>servers/mqweb/workarea</code>	Sunucu tarafından çalışırken oluşturulan dizin. Bu dizin, sunucu ilk çalıştırıldıktan sonra oluşturulur.

## Düzenlenemez dosyalar



Düzenlenemeyen dosyalar, IBM MQ Console ve REST API kuruluş bileşeninin ilk kuruluşunun bir parçası olarak yerleştirilmiştir. Bu dosyalar, bakım uygulandığında güncellenir.

Düzenlenemeyen dosyaların konumu, işletim sistemine ve kuruluürününe bağlıdır.

- Bir IBM MQ kuruluşunda, düzenlenemeyen dosyalar aşağıdaki dizinlerden birinde bulunur:

-  AIX, Linux, and Windows sistemlerinde: `MQ_INSTALLATION_PATH/web`
-  IBM sistemlerinde: `MQ_INSTALLATION_PATH/web`
-  z/OS işletim tarihinde: `installation_directory/web/`

Burada `installation_directory`, IBM MQ for z/OS UNIX System Services Components kuruluş yoludur.

-   Bağımsız bir IBM MQ Web Server kuruluşunda, IBM MQ Web Server kuruluş dosyasının açıldığı dizin.

Bu konumda aşağıdaki dizin yapısı ve dosyalar var:

Dizinler ve dosyalar	Açıklama
<code>bin/</code>	WebSphere Liberty komutlarını içeren dizin.  Bu dizinde komut dosyalarını yürütmek için <u>ayrıcalıklı kullanıcı</u> olmanız gerekir.
<code>mq/</code>	Çeşitli IBM MQ kaynaklarını içeren dizin yapısı.
<code>mq/apps/</code>	IBM MQ Console ve REST API uygulamalarını içeren dizin.
<code>mq/etc/</code>	
<code>mq/etc/mqweb.xml</code>	<code>mqweb</code> sunucusu için salt okunur yapılandırma dosyası.  Yapılandırma değişiklikleri yapmak için <code>mqwebuser.xml</code> dosyasını düzenleyin.
<code>mq/libs</code>	IBM MQ Console ve REST API tarafından kullanılacak paylaşılan kitaplıkları içeren dizin.

Dizinler ve dosyalar	Açıklama
mq/samp	Örnekleri içeren dizin.
mq/samp/configuration	mqwwebuser.xml dosyasına kopyalanabilen örnek yapılandırma dosyalarını içeren dizin.

## mqwweb sunucusu yapılandırmanızı yedekleme ve geri yükleme

mqwweb sunucusu yapılandırmanızı yedekleyebilir ve aynı yere ya da farklı bir yere geri yükleyebilirsiniz.

### Başlamadan önce

mqwweb sunucusu yapılandırmanızı geri yükleyebilmeniz için, mqweb sunucusunu geri yüklemek istediğiniz sisteme IBM MQ'ya da bağımsız IBM MQ Web Server, kurulmalıdır. Bağımsız bir IBM MQ Web Server kuruluşunda, [“Bağımsız IBM MQ Web Server” in yapılandırılması](#) sayfa 795’indeki adımları izleyerek mqweb sunucusunu yaratmalısınız.

### Bu görev hakkında

mqwweb sunucusu yapılandırmanızı yedeklemek ve geri yüklemek için bu görevdeki yordamı izleyin. mqweb sunucunuzu farklı bir konuma geri yükleyinizse, dosyalara başvuruların doğru olduğundan emin olmak için mqweb sunucusu yapılandırmasını güncellemelisiniz.

**V 9.3.5** Bu yordamı, şu anda bir IBM MQ kuruluşunda çalışan bir mqweb sunucusunu bağımsız bir IBM MQ Web Server kuruluşta çalışacak şekilde geçirmek için de kullanabilirsiniz.

### Yordam

- mqwweb sunucusu yapılandırmanızı yedeklemek için, mqweb sunucusu yapılandırmanızı içeren dizindeki tüm dosyaları yedek yerinize kopyalayın.
  - Bir IBM MQ kuruluşunda, aşağıdaki dizinin içeriğini kopyalayın:
    - Linux** **AIX** AIX ya da Linux sistemlerinde: `/var/mqm/web/installations/installationName`
    - Windows** Windows üzerinde: `MQ_DATA_PATH\web\installations\installationName`; burada `MQ_DATA_PATH`, IBM MQ veri yoludur. Bu yol, IBM MQ kuruluşu sırasında seçilen veri yoludur. Varsayılan olarak bu yol `C:\ProgramData\IBM\MQ` olur.
    - z/OS** z/OS sistemlerinde: `crtmqweb` komut dosyası mqweb sunucusu tanımlamasını yaratmak için çalıştırıldığında belirtilen WebSphere Liberty kullanıcı dizini.
  - Linux** **V 9.3.5** Bağımsız bir IBM MQ Web Server kuruluşunda, `MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST` dizinin içeriğini kopyalayın; burada `MQ_OVERRIDE_DATA_PATH`, `MQ_OVERRIDE_DATA_PATH` ortam değişkeninin gösterdiği IBM MQ Web Server veri dizini.
- mqwweb sunucusu yapılandırmanızı geri yüklemek için, mqweb sunucusu yapılandırmanızı içeren dizinin içeriğini, [“1” sayfa 825](#). adımda kopyaladığınız dosyalarla değiştirin.
  - IBM MQ kuruluşunda, aşağıdaki dizinin içeriğini değiştirin:
    - Linux** **AIX** AIX ya da Linux sistemlerinde: `/var/mqm/web/installations/installationName`
    - Windows** Windows üzerinde: `MQ_DATA_PATH\web\installations\installationName`; burada `MQ_DATA_PATH`, IBM MQ veri yoludur. Bu yol, IBM MQ kuruluşu sırasında seçilen veri yoludur. Varsayılan olarak bu yol `C:\ProgramData\IBM\MQ` olur.

- **z/OS** z/OS sistemlerinde: **crtmqweb** komut dosyası mqweb sunucusu tanımlamasını yaratmak için çalıştırıldığında belirtilen WebSphere Liberty kullanıcı dizini.
  - **Linux V 9.3.5** Bağımsız bir IBM MQ Web Server kuruluşunda, **MQ\_OVERRIDE\_DATA\_PATH/web/installations/MQWEBINST** dizininin içeriğini değiştirin; burada **MQ\_OVERRIDE\_DATA\_PATH** , **MQ\_OVERRIDE\_DATA\_PATH** ortam değişkeninin gösterdiği IBM MQ Web Server veri dizinidir.
3. “2” sayfa 825 . adımda geri yüklediğiniz dosyaların sahipliğini, mqweb sunucusu kullanıcı kimliğinin dosyalara erişebilmesi için ayarlayın.
4. mqweb sunucusu yapısını farklı bir yere geri yüklerseniz, mqweb sunucusu yapılışında önceki mqweb sunucusu yapılış dizinindeki dosyalara gönderme yapan özelliklerin değerini değiştirin.
- a) **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, ortamınızı mqweb sunucusu yapılandırmanızı gösterecek şekilde ayarlayın.

- **z/OS** z/OS üzerinde, aşağıdaki komutu girerek **WLP\_USER\_DIR** ortam değişkenini, değişken mqweb sunucusu yapılandırmanızı gösterecek şekilde ayarlayın:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada **WLP\_user\_directory** , **crtmqweb** komutuna geçirilen dizinin adıdır.

Daha fazla bilgi için bakınız: [Create the mqweb server.](#)

- **Linux V 9.3.5** Bağımsız bir IBM MQ Web Server kuruluşunda, **MQ\_OVERRIDE\_DATA\_PATH** ortam değişkenini IBM MQ Web Server veri dizinine ayarlayın.
  - Diğer tüm ortamlarda, ortamınızı ayarlamak için herhangi bir işlemi tamamlamanız gerekmez.
- b) Bir kullanıcının değiştirdiği tüm yapılandırılabilir mqweb sunucusu özelliklerinin değerini görüntüleyin. Şu komutu verin:

```
dspmqweb properties -u
```

- c) **remoteKeyfile** özelliği görüntülenirse, özelliğin değerini denetleyin. Özelliğin değeri önceki mqweb sunucusu yapılış dizinindeki bir dosya yoluna gönderme yapıyorsa, değeri yeni mqweb sunucusu yapılış dizinindeki dosya yoluna gönderme olacak şekilde değiştirin. **remoteKeyfile** özelliğinin değerini değiştirmek için aşağıdaki komutu verin:

```
setmqweb properties -k remoteKeyfile -v path_to_keyfile
```

- d) mqweb sunucusu uzak kuyruk yöneticisi yapılışını görüntüleyin. Şu komutu verin:

```
dspmqweb remote -a
```

- e) Aşağıdaki özelliklerden herhangi biri görüntülenirse, özelliğin değerini denetleyin:

- **globalTrustStorePath**
- **globalKeyStorePath**
- **ccdtURL**
- **keyStorePath**
- **trustStorePath**

Önceki mqweb sunucusu yapılandırma dizinindeki bir dosya yoluna başvuran herhangi bir özelliğin değerini, yeni mqweb sunucusu yapılandırma dizinindeki dosya yoluna başvuracak şekilde değiştirin. Her özelliğin değerini değiştirmek için **setmqweb remote** komutunu yazın. Örneğin, uzak kuyruk yöneticisine ilişkin **keyStorePath** özelliğinin değerini **remote-QM1** benzersiz adıyla değiştirmek için şu komutu verin:

```
setmqweb remote -uniqueName remote-QM1 -keyStorePath new_keystore_path
```

Daha fazla bilgi için bakınız: [setmqweb remote \(set mqweb server remote queue manager configuration\)](#).

## Linux > MQ Adv. > Windows > MQ Adv. VUE > MQ Adv. z/OS **Linux ya da Windows platformlarında Aspera gateway bağlantısı tanımlanması**

IBM Aspera faspio Gateway , IBM MQ için ağ verimini önemli ölçüde artırabilen hızlı bir TCP/IP tüneli sağlar. Herhangi bir yetkili platformda çalışan bir kuyruk yöneticisi, Aspera gateway aracılığıyla bağlanabilir. Ağ geçidinin kendisi Red Hat ya da Ubuntu Linux ya da Windows üzerinde devreye alınır.

### Bu görev hakkında

Aspera gateway , kuyruk yöneticisi kanallarının performansını artırmak için kullanılabilir. Özellikle ağ yüksek gecikme süresine sahipse veya paketleri kaybetme eğilimindeyse etkilidir ve genellikle farklı veri merkezlerindeki kuyruk yöneticileri arasındaki bağlantıyı hızlandırmak için kullanılır.

**Not:** Paketleri kaybetmeyen hızlı bir ağ için Aspera gateway kullanırken başarımda bir düşüş vardır, bu nedenle Aspera gateway bağlantısı tanımlanmadan önce ve tanımlandıktan sonra ağ başarımının denetlenmesi önemlidir.

IP ağ bağlantısının her bir ucunda bir Aspera gateway tanımlarsınız, ardından kuyruk yöneticisi kanallarını her bir ağ geçidine bağlamak için TCP/IP ' yi kullanırsınız. Bir kuyruk yöneticisinin, kullandığı Aspera gateway ile aynı makinede çalışması gerekmez ve birden çok kuyruk yöneticisi aynı ağ geçidini kullanabilir.

Aspera gateway ürünü kullanmak için aşağıdaki yetkilerden birine ya da daha fazlasına sahip olmanız gerekir:

- IBM MQ Advanced for Multiplatforms
- IBM MQ Appliance
- IBM MQ Advanced for z/OS VUE
- **LTS** **V 9.3.4** IBM MQ Advanced for z/OS, Long Term Support ya da Continuous Delivery from IBM MQ 9.3.4

Aspera gateway uygulamasını aşağıdaki platformlardan herhangi birinde konuşlandırabilirsiniz:

- Linux for x86-64
- Linux on Power Systems - Little Endian
- Linux for IBM Z
- Windows - Windows üzerindeki platform desteğiyle ilgili daha fazla bilgi için [IBM Aspera faspio Gateway belgelerine](#) bakın.

Ağ geçidinin ayrı olarak yetkilendirilmediği sürece, Aspera gateway iletilerinin kullanımı IBM MQ iletileriyle sınırlıdır.

Aspera gateway kullanan kuyruk yöneticileri, desteklenen herhangi bir altyapıda çalışıyor olabilir. Desteklenen platformların tam listesi için bkz. [Ürün belgelerinde kullanılan simgeler](#).

Kullandığı Aspera gateway ile aynı makinede olmayan her kuyruk yöneticisi için, kuyruk yöneticisi ile Aspera gateway arasında hızlı bir ağ bağlantısına sahip olup olmadığınızı denetleyin.

Ağ geçidinin kullandığı gelen ve giden kapıları tanımlayan bir ağ geçidi tanımı oluşturmak için bir tom1 dosyası kullanırsınız. Aspera gateway ile birlikte örnek bir tom1 dosyası gönderilir. Giden ağ geçidi tanımlaması, yerel kuyruk yöneticisinden ağ geçidine ve yerel ağ geçidinden uzak ağ geçidine bağlantıyı tanımlar. Gelen ağ geçidi tanımlaması, uzak ağ geçidinden yerel ağ geçidine ve yerel ağ geçidinden yerel kuyruk yöneticisine olan bağlantıyı tanımlar.

Aşağıdaki adımlar, çalışmaya başlamaya ve çalışmaya ilişkin temel bir kılavuz sağlar. Daha ayrıntılı bilgi için [IBM Aspera faspio Gateway belgelerine](#) bakın.



## Yordam

### 1. Aspera gateway kuruluş görüntüsünü alın.

**Multi** Çoklu Platformlar için, Aspera gateway ürününü Passport Advantagesitesinden yükleyebilirsiniz. Yükleme "IBM Aspera faspio Continuous Delivery Release for IBM MQ V9.3 Multiplatform Çok Dilli eAssembly" etiketlidir. Yalnızca bu alandaki değişiklik hızı nedeniyle Continuous Delivery (CD) görüntüsü olarak teslim edilir; bu, CD yayınlarının sıklığı itibariyle güncellemelerin gerekli olduğu anlamına gelir ve bunu IBM MQ Advanced for Multiplatforms ya da IBM MQ Appliance yetkisi olan herhangi bir IBM MQ sistemine kurabilirsiniz. Bu eAssembly'yi yüklemek için Downloading IBM MQ 9.3 (Karşıdan Yükleme) seçeneğine gidin ve ardından gerekli yayın düzeyine ilişkin sekmeyi tıklayın. eAssembly, ağ geçidinin kullanılabilir olduğu tüm platformlara ilişkin kuruluş görüntülerini içerir. **V 9.3.0** **V 9.3.0** eAssembly, lisans dosyası içeren bir `ibm-faspio-license.zip` dosyası da içerir.

**LTS** **MQ Adv. VUE** **V 9.3.4** **MQ Adv. z/OS** IBM MQ sisteminiz IBM MQ Advanced for z/OS VUE yetkisine ya da IBM MQ Advanced for z/OS yetkisine sahipse Long Term Supportya da Continuous Delivery from IBM MQ 9.3.4, SMP/E kuruluşunun parçası olan Connector Pack bileşeninden Aspera gateway ürününü edinebilirsiniz.

**MQ Adv. VUE** **MQ Adv. z/OS** IBM MQ Advanced for z/OS VUE ve IBM MQ Advanced for z/OS ile ilgili dosyalar aşağıdaki gibidir:

Hizmet olarak sunulan	Dosya adı	faspio sürüm numarası
Linux for x86-64	<b>V 9.3.4</b> M0C5LEN.zip	1.3.3
Linux on Power Systems - Little Endian	<b>V 9.3.4</b> M0C5MEN.zip	1.3.3
Linux for IBM Z	<b>V 9.3.4</b> M0C5NEN.zip	1.3.3
Windows	<b>V 9.3.4</b> M0C5PEN.zip	1.3.3
Linux for x86-64	<b>V 9.3.3</b> M0B2XEN.zip	1.3.2
Linux on Power Systems - Little Endian	<b>V 9.3.3</b> M0B2YEN.zip	1.3.2
Linux for IBM Z	<b>V 9.3.3</b> M0B2ZEN.zip	1.3.2
Windows	<b>V 9.3.3</b> M0B30EN.zip	1.3.2
Linux for x86-64	<b>V 9.3.2</b> M090HEN.zip	1.3.1
Linux on Power Systems - Little Endian	<b>V 9.3.2</b> M090JEN.zip	1.3.1
Linux for IBM Z	<b>V 9.3.2</b> M090KEN.zip	1.3.1
Windows	<b>V 9.3.2</b> M090LEN.zip	1.3.1



Çizelge 52. Platforma ve IBM MQ sürümüne göre dosya adları ve faspio sürüm numaraları (devamı var)		
Hizmet olarak sunulan	Dosya adı	faspio sürüm numarası
Linux for x86-64	V 9.3.0 M0559EN.zip	1.3.0
Linux on Power Systems - Little Endian	V 9.3.0 M055BEN.zip	1.3.0
Linux for IBM Z	V 9.3.0 M055CEN.zip	1.3.0
Windows	V 9.3.0 M055DEN.zip	1.3.0

Aspera gateway ' in z/OS üzerinde yerel olarak çalışamayacaktır.

V 9.3.0 V 9.3.0 MQ Adv. VUE MQ Adv. z/OS Kuruluş görüntülerine ek olarak, fasp dizini bir lisans dosyası içeren M05QKEN.zip dosyasını içerir.

- Aspera gateway kuruluş görüntüsünü ağ geçidini çalıştıracak iki makineye kopyalayın ve ağ geçidini çıkarıp kurun.

V 9.3.0 V 9.3.0 ibm-faspio-license.zip (Multiplatforms) ya da M05QKEN.zip (z/OS) içinde bulunan lisans kütüğünü kullanın. Daha fazla bilgi için IBM Aspera faspio Gateway belgelerine bakın:

- Linux Linux üzerine kurma
- Windows Windows üzerine kurma

- Her bir ağ geçidini yapılandırın ve güvenliğini sağlayın.

V 9.3.0 V 9.3.0 Daha fazla bilgi için IBM Aspera faspio Gateway belgelerine bakın:

- Gateway yapılış kütüğünün yapılandırılması
- Ağ Geçidinin Güvenli Kılınması

- Ağ bağlantısının her bir ucunda, kanal tanımını, yerel ağ geçidinin dinlediği kapiya bağlanacak şekilde değiştirin.
- Her ağ geçidi hizmetini başlatın.

V 9.3.0 V 9.3.0 Daha fazla bilgi için IBM Aspera faspio Gateway belgelerine bakın:

- Linux Linux üzerinde başlatma
- Windows Windows üzerinde başlatma

- Kanalları yeniden başlatın.

Kuyruk yöneticileriniz şimdi bir Aspera gateway bağlantısı üzerinden iletişim kuruyor.

## Örnek

Bu örnek, Linux çalıştıran iki makinede bir Aspera gateway bağlantısını tanımlar. Yapılandırma aşağıdaki gibidir:

- Yerel ağ geçidi makinesinin IP adresi: 9.20.193.107. Uzak ağ geçidi makinesinin IP adresi: 9.20.192.115.
- Yerel kuyruk yöneticisi, 9.20.121.5 IP adresine sahip bir makinede çalışıyor. Uzak kuyruk yöneticisi, 9.20.121.25 IP adresine sahip bir makinede çalışıyor. Her iki kuyruk yöneticisi de 1414 numaralı kapıda dinliyor.

- Yerel kuyruk yöneticisindeki kuyruk yöneticisi kanalı, **conname** 9.20.193.107(1500) kullanılarak yerel Aspera gateway ' e bağlanacak şekilde değiştirilir. Uzak kuyruk yöneticisindeki kuyruk yöneticisi kanalı, **conname** 9.20.192.115(1500) kullanılarak uzak Aspera gateway ' e bağlanacak şekilde değiştirilir.
- **V 9.3.0** **V 9.3.0** IBM Aspera faspio Gateway 1.2' dan TLS varsayılan olarak etkindir. TLS ' yi ağ geçidiyle yapılandırmak istiyorsanız, IBM Aspera faspio Gateway belgelerinde [Ağ Geçidini Güvenlik](#) başlıklı konuya bakın.

1. Yerel ağ geçidi makinesinde bir Aspera gateway bağlantısı tanımlayın:

- Aspera gatewayürünü kurun:

– **Linux** Linux' de aşağıdaki komutu kullanın:

```
rpm -ivh ibm-faspio-gateway-<version>.x86_64.rpm
```

- Kuruluş tarafından oluşturulan dizindeki gateway .toml dosyasını değiştirin:

Yerel ağ geçidi tanımlamalarını ayarlamak için dosyayı düzenleyin.

```
[[bridge]]
  name = "Outbound"
  [bridge.local]
    protocol = "tcp"
    host = "9.20.193.107"
    port = 1500
  tls_enabled = false

  [bridge.forward]
    protocol = "fasp"
    host = "9.20.192.115"
    port = 1600
  tls_enabled = false

[[bridge]]
  name = "Inbound"
  [bridge.local]
    protocol = "fasp"
    host = "9.20.193.107"
    port = 1600
  tls_enabled = false

  [bridge.forward]
    protocol = "tcp"
    host = "9.20.121.5"
    port = 1414
  tls_enabled = false
```

- **V 9.3.0** **V 9.3.0** aspera-license dosyasını ibm-faspio-license.zip (Multiplatforms) ya da M05QKEN.zip (z/OS) olanağından /usr/local/etc/faspio/içine kopyalayın.

2. Uzak ağ geçidi makinesinde bir Aspera gateway bağlantısı tanımlamak için önceki adımı yineleyin.

- Kuruluş tarafından oluşturulan dizindeki gateway .toml dosyasını değiştirin. Uzak ağ geçidi tanımlarını ayarlamak için dosyayı düzenleyin:

```
[[bridge]]
  name = "Outbound"
  [bridge.local]
    protocol = "tcp"
    host = "9.20.193.107"
    port = 1500
  tls_enabled = false

  [bridge.forward]
    protocol = "fasp"
    host = "9.20.192.115"
    port = 1600
  tls_enabled = false

[[bridge]]
  name = "Inbound"
  [bridge.local]
```

```
protocol = "fasp"  
host = "9.20.193.107"  
port = 1600  
tls_enabled = false  
  
[bridge.forward]  
protocol = "tcp"  
host = "9.20.121.5"  
port = 1414  
tls_enabled = false
```

- **V9.3.0** **V9.3.0** aspera-license dosyasını ibm-faspio-license.zip (Multiplatforms) ya da M05QKEN.zip (z/OS) olanağından /usr/local/etc/faspio/içine kopyalayın.
3. Bağlantının her bir ucunda, kanal tanımını, yerel ağ geçidinin dinlediği kapiya bağlanacak şekilde değiştirin.
    - Yerel kuyruk yöneticisindeki kuyruk yöneticisi kanalını **conname** 9.20.193.107(1500) kullanarak yerel Aspera gateway ' ye bağlanacak şekilde değiştirin.
    - Uzak kuyruk yöneticisindeki kuyruk yöneticisi kanalını, **conname** 9.20.192.115(1500) kullanarak uzak Aspera gateway ' e bağlanacak şekilde değiştirin.
  4. Yerel ağ geçidi makinesinde aşağıdaki komutu çalıştırarak yerel ağ geçidini başlatın:
    - **Linux**

```
sudo systemctl start faspio-gateway
```
  5. Uzak ağ geçidi makinesinde aşağıdaki komutu çalıştırarak uzak ağ geçidini başlatın:
    - **Linux**

```
sudo systemctl start faspio-gateway
```
  6. Kanalları yeniden başlatın.

## Sonraki adım

Aspera gateway , aldığı verileri hiçbir şekilde yorumlamadan aktarır. Bu, ağ geçidi bağlantısı TLS el sıkışmasının farkında olmadığı için Aspera gateway kullanan kuyruk yöneticisi kanalları arasında TLS yapılandırabileceğiniz anlamına gelir. Bu, desteklenen herhangi bir IBM MQ platformundaki kuyruk yöneticilerinin Aspera gatewayplatformunu kullanabileceği anlamına da gelir.

Ağ geçidiyle çok eşgörünümlü bir kuyruk yöneticisini kullanmak için, kuyruk yöneticisinin her eşgörünümü için ağ geçidi tanımlamalarını yapılandırın.

**Not:** Aspera gateway yalnızca kuyruk yöneticisi kanallarıyla sınırlanmıştır. Müşteri kanallarıyla test edilmedi. Bunun nedeni, Aspera gateway ' in öngörülen kullanımının uzak kuyruk yöneticilerini yavaş bir ağ üzerinden bağlamasıdır; ancak istemci uygulamaları genellikle hızlı bir ağ üzerinden yerel bir veri merkezindeki kuyruk yöneticilerine bağlanır.

## İlgili kavramlar

[Aspera gateway Yol haritası](#)

## İlgili başvurular

[“Kullanılacak iletişim tipi” sayfa 15](#)

Farklı platformlar farklı iletişim protokollerini destekler. İletim iletişim kuralı seçiminiz, IBM MQ MQI client ve sunucu platformlarının birleşimine bağlıdır.

[IBM Aspera faspio Gateway belgeleri](#)

# IBM MQ ' in IBM Cloud Private ölçüm hizmetiyle kullanılmak üzere yapılandırılması

Kuyruk yöneticisi başlatma ve kullanım bilgilerini raporlamak ve görüntülemek için IBM MQ ürününü IBM Cloud Private ölçüm hizmetiyle kullanmak üzere yapılandırma.

## Başlamadan önce

IBM MQ kuyruk yöneticilerinizi IBM Cloud Private hizmetini kullanacak şekilde yapılandırmadan önce bir IBM Cloud hesabınızın olması gerekir. Hesabınızı oluşturmak için bkz. [IBM Cloud](#).

## Bu görev hakkında

IBM Cloud Private ölçüm hizmetini kullanarak, şirket içi IBM ürünlerinizi IBM Cloud Private içindeki hizmet örneğinize bağlayabilir ve kuruluşunuzdaki tüm kayıtlı ürünleri tek bir gösterge panosunda görebilirsiniz.

AIX, Linux ve Windows kuyruk yöneticilerinizi yapılandırabilir ve ölçüm hizmeti eşgörünümünüze bağlayabilir ve bunların başlangıç ve kullanım bilgilerini görebilirsiniz. Ancak, Linux Container ortamları dışındaki platformlarda veriler, Hourly Container-Based fiyatlandırma lisanslarını desteklemek için kullanılamaz.

Aylık bir VPC lisans tipine ilişkin kullanım verilerini kaydetmek için, varsayılan saatlik lisanslama ölçüsü yerine `AMQ_LICENSEING_METRIC=VPCMonthlyPeak` ortam değişkenini ayarlayın. Bu, kuyruk yöneticisinin saatlik kapsayıcı tabanlı lisanslarla ilgili verileri karşıya yükleme varsayılan davranışı yerine aylık VPC lisans tipleriyle ilgili verileri karşıya yüklemesine neden olur.

`qm.ini` dosyasında `ReportingService` kısmı ile aşağıdaki öznitelikleri kullanın:

### APIKeyFile

Ölçüm hizmeti eşgörünümü **APIKey** değerini içeren metin dosyasının konumu.

### CapacityReporting

Hata günlüğü iletilerini düzenli aralıklarla `AMQERR` günlüklerine aşağıdaki biçimde yazar:

```
4/22/2020 01:44:29 PM - Process(1274.1) User(bld-adm) Program(amqmgr0)
Host(8b3b83f2bc7d) Installation(Docker)
VRMF(9.2.0.0)
Time(2020-04-22T13:44:29.295Z)
ArithInsert1(300)
CommentInsert1(8.5)
CommentInsert2(IBM MQ Advanced)
```

**CapacityReporting** özniteliği tarafından üretilen bilgiler `AMQ5064` iletilisine eklenir; bu ileti, işletmenizin IBM MQ ne kadar kullandığını daha iyi anlamanızı sağlar:

### AMQ5064

Bu kuyruk yöneticisi 300 saniyedir çalışıyor. Şu anda 8.5 çekirdeği ile çalışıyor. Lisans tipi: IBM MQ Advanced.

### Önem Derecesi

0: Bilgi

### Açıklama

Bu, kullanım takibi için bir bilgi iletidir.

### Yanıt

Yok.

### LicensingGroup

Kuyruk yöneticisinin ait olduğu faturalama grubu. Bu, verilerin ölçüm hizmeti tarafından oluşturulan raporlarda gruplanma şeklini etkiler.

### ServiceURL

IBM Cloud Private hizmet adresi.

## ServiceProxy

Kuyruk yöneticilerinin ölçüm hizmetinin çalıştığı ağa doğrudan erişimi yoksa kullanılacak HTTP yetkili sunucusu için URL ve kapı.

Ürünlerinizin kurulu olduğu anasistemleri, kullanmakta olduğunuz ürün sürümlerini ve üzerinde çalıştıkları platformları görebilirsiniz. Her bir ürün için görüntülenen üst düzey kullanım ölçümlerinden, iş yüklerinin ne kadar ağır olduğuna ilişkin bir genel bakış elde edebilirsiniz. IBM MQ için, hangi kuyruk yöneticilerinin daha çok kullanıldığını ve hangilerinin daha hafif iş yüklerine sahip olduğunu görebilirsiniz.

Bir kuyruk yöneticisi ölçüm hizmetinin bir eşgörünümüne bağlanacak şekilde yapılandırıldığında, IBM Cloud Private' e aşağıdaki bilgiler bildirilir:

- IBM MQ Kuyruk yöneticisi adı
- IBM MQ kuyruk yöneticisi tanıtıcısı
- IBM MQ kuruluş kök dizini
- IBM MQ kurulu bileşenleri (ad ve sürüm)
- Anasistem adı
- Anasistem işletim sistemi adı
- Anasistem işletim sistemi sürümü
- IBM MQ kuyruk yöneticisi için Sanal İşlemci Çekirdeği (VPC) kullanım bilgileri

Ölçüm hizmeti eşgörünümü gösterge panonuzda kuyruk yöneticisi VPC kullanım ölçümlerinizi izleyebilirsiniz.

## Yordam

- IBM Cloud Private üzerinde ölçüm hizmeti eşgörünümüyle kullanılacak bir kuyruk yöneticisi yapılandırın.
- Bir HTTP yetkili sunucusu aracılığıyla IBM Cloud Private ölçüm hizmetine bağlanın.
- IBM Cloud Private ölçüm hizmeti bağlantısında sorun giderin.

## İlgili başvurular

[Sanal İşlemci Çekirdekleri \(VPC ' ler\) için fiyatlandırma ölçüsü](#)

## **Multi** IBM Cloud Private üzerinde ölçüm hizmeti eşgörünümüyle kullanılmak üzere bir kuyruk yöneticisinin yapılandırılması

Kuyruk yöneticiniz için güvenlik ve IBM Cloud kayıt bilgilerinizi ayarlayın ve önceden yarattığınız ölçüm hizmeti eşgörünümüne bağlanın.

## Bu görev hakkında

IBM Cloud Private ölçüm hizmeti eşgörünüm gösterge panonuz yalnızca güvenlik ve IBM Cloud Private kayıt bilgilerinizi içerecek şekilde yapılandırılmış kuyruk yöneticilerine ilişkin verileri gösterir.

## Yordam

1. Hizmet tanıtıcısı yaratmak için ICP belgelenmiş adımları izleyin:  
[IBM Cloud Private CLI kullanılarak hizmet tanıtıcısı yaratılması.](#)
2. Aşağıdaki yerde bir API anahtarı oluşturmak için ICP belgelenmiş adımları izleyin:  
[API anahtar yönetimi API ' leri.](#)
3. ICP kümesinden TLS sertifikalarını karşıdan yükleyin.  
Sertifikaları karşıdan yüklediğiniz yeri not edin. “9” sayfa 834. adımda, karşıdan yüklenen sertifikaları kuyruk yöneticinizin anahtar deposuna ekleyebilirsiniz.

4. Bir metin dosyası `apikeyfile.txt` oluşturun ve önceki görevde kopyaladığınız **API key** değerini ekleyin.  
apikeyfile.txt 'in konumunu not edin; böylece, adım 8' de yolu ekleyebilirsiniz. Bu dosya, kuyruk yöneticisi kullanıcısı tarafından okunabilir olmalıdır (AIX and Linux sistemlerinde 'mqm'). Dosya, JSON bilgi yükünü değil, yalnızca **API key** ' in kendisini içermelidir; örneğin, d9c11b45-4dda-4de4-c0b2-2e4e1004dc64.
5. Kuyruk yöneticisini yaratın; örneğin, *QM1*.  
Daha fazla bilgi için [Çoklu Platformda kuyruk yöneticileri oluşturma ve yönetme](#) başlıklı konuya bakın.
6. Kuyruk yöneticisini *QM1* başlatın.  
Ek bilgi için [Kuyruk yöneticisinin başlatılması](#) başlıklı konuya bakın.
7. IBM MQ komutlarını çalıştırmadan önce IBM MQ komut satırı ortamınızı ayarlarınızı unutmayın.  
**setmqenv** komutunu çalıştırın.

**AIX** AIX'ta:

```
. /usr/mqm/bin/setmqenv -s
```

**Linux** Linux'ta:

```
. /opt/mqm/bin/setmqenv -s
```

**Windows** Windows'ta:

```
"C:\Program Files\IBM\MQ\bin\setmqenv.cmd" -n installation name
```

8. *QM1* kuyruk yöneticisi için bir SSL güvenli deposu oluşturun.

**AIX** AIX üzerinde güvenli depo oluşturmaya başlayın:

```
runmqckm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms -expire 30 -stash
```

**Linux** Linux'ta:

```
runmqckm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms -expire 30 -stash
```

**Windows** Windows'ta:

```
runmqckm -keydb -create -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type cms -expire 30 -stash
```

9. "3" sayfa 833. adımda karşıdan yüklediğiniz dijital sertifikaları kuyruk yöneticisinin güvenli deposuna ekleyin.

**AIX** AIX'ta:

```
runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms -label RootCA -file Download_location/RootCA.crt -format ascii -trust enable  
runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms -label ServerCert -file Download_location/CERT.crt -format ascii -trust enable
```

**Linux** Linux'ta:

```
runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms -label RootCA -file Download_location/RootCA.crt -format ascii -trust enable
```

```
runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms  
-label ServerCert  
-file Download_location/CERT.crt -format ascii -trust enable
```

**Windows** Windows'ta:

```
runmqckm -cert -add -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type cms  
-label RootCA  
-file "Download_location\RootCA.crt" -format ascii -trust enable  
  
runmqckm -cert -add -db "C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl\key.kdb" -pw password -type  
cms -label ServerCert  
-file "Download_location\CERT.crt" -format ascii -trust enable
```

10. Kuyruk yöneticisinin qm.ini dosyasına apikeyfile yoluyla yeni ReportingService kısmı ekleyin:

```
ReportingService:  
APIKeyFile=APIKey file location/apikeyfile.txt
```

11. **API host** değerini qm.ini dosyasına ekleyin.

ReportingService kısmı artık apikeyfile ve **API host (ServiceURL)** değerlerinin yolunu içerir:

```
ReportingService:  
APIKeyFile=APIKey file location/apikeyfile.txt  
ServiceURL=https://productinsights-api.ng.bluemix.net
```

qm.ini dosyasını kaydedin ve dosyadan çıkın.

12. Değişikliklerin yürürlüğe girmesi için kuyruk yöneticisini yeniden başlatın.  
**amqzmux0** kuyruk yöneticisi işleminin ağı erişmesi için izin vermeniz istenebilir. Kuyruk yöneticisinin ölçüm hizmetiyle iletişim kurmasını sağlamak için erişim gereklidir.
13. Ölçüm hizmeti eşgörünümünüzdeki kuyruk yöneticisine *QM1* ilişkin bilgileri görüntüleyin.  
Raporlama durumu etkin olduğunda, belirtilen bütünleştirme düğümündeki tüm bütünleştirme sunucularına ilişkin başlatma ve kullanım bilgileri ölçüm hizmetine bildirilir. Kullanım bilgileri her 15 dakikada bir güncellenir.
14. İsteğe bağlı: ReportingService kısmı kuyruk yöneticisinin qm.ini dosyasından kaldırıp kuyruk yöneticisini yeniden başlatarak, kuyruk yöneticisinin ölçüm hizmetine raporlamasını durdurun.
15. İsteğe bağlı: Kuyruk yöneticisi başlatma ya da kullanım bilgilerini ölçüm hizmetine raporlamazsa, kuyruk yöneticisinin günlük dosyasındaki tanılama bilgilerini denetleyin.

AIX için değişiklik

**AIX** AIX'ta:

```
/var/mqm/qmgrs/QM1/errors/AMQERR0*.log
```

**Linux** Linux'ta:

```
/var/mqm/qmgrs/QM1/errors/AMQERR0*.log
```

**Windows** Windows'ta:

```
C:\ProgramData\IBM\MQ\errors\AMQERR0*.log
```

## Sonuçlar

Bir ölçüm hizmeti eşgörünümü yarattınız ve kuyruk yöneticinizi yönetim ortamına bağlanacak şekilde yapılandırdınız. Ölçüm hizmeti eşgörünümü gösterge panosunda kuyruk yöneticinizle ilgili bilgileri görebilirsiniz.

## IBM Cloud Private ölçüm hizmetine HTTP yetkili sunucusu aracılığıyla bağlanma

Kuyruk yöneticiniz ICP kümenize doğrudan erişimi olmayan bir sistemde çalışıyorsa, kuruluşunuzun IBM Cloud Private içinde ölçüm hizmeti eşgörünümünüze bağlanmak için sağladığı bir HTTP yetkili sunucusunu kullanabilirsiniz.

### Başlamadan önce

Güvenliği yapılandırdınız, kuyruk yöneticiniz için `qm.ini` dosyasına **API key** ve hizmet URL eklediniz.

### Bu görev hakkında

Kuyruk yöneticinizi, kuruluşunuz tarafından sağlanan bir HTTP yetkili sunucusu aracılığıyla IBM Cloud Private içindeki [ölçüm hizmeti eşgörünümüne](#) bağlanacak şekilde yapılandırmak için bu görevi kullanın.

### Yordam

- `qm.ini` dosyanızın IBM Cloud Private kayıt kısmına bir hizmet yetkili sunucusu özniteliği ekleyin. **ServiceProxy** özniteliğini aşağıdaki gibi ayarlayabilirsiniz:
  - `http://` önekini ve isteğe bağlı olarak kapıyı içeren bir URL . Kapıyı belirtmezseniz, `1080` kullanılır.

```
ReportingService:
  ServiceProxy=http://myorgproxy.net:1080
```

**Not: ServiceProxy** parametresi geçerli bir `http://` URL değerine ayarlanmalıdır. HTTPS ve SOCKS gibi diğer yetkili iletişim kuralları desteklenmez.

- Değişiklikler yürürlüğe girmeden önce kuyruk yöneticinizi yeniden başlatın.

## Ölçüm hizmeti bağlantısında sorun giderme

Kuyruk yöneticinizi bir ölçüm hizmeti eşgörünümüne bağlarken karşılaşılabileceğiniz hatalara ilişkin sorun giderme önerileri.

### Kuyruk yöneticisi, kullanım metriklerini yapılandırılan ölçüm hizmetine kaydedemiyor ya da bu hizmete yükleyemiyor

Kuyruk yöneticisinin ağa erişimi olup olmadığını denetleyin. 'nın API Key özniteliğindeki **APIKey** değeri yanlış. IBM Global Security Kit (GSKit) bileşeninin kurulu olduğundan emin olun.

### Geçersiz `qm.ini` kısmı

Geçersiz bir `qm.ini` kısmı bulundu. Daha fazla bilgi için hata günlüğünü denetleyin.

### Geçersiz HTTP hizmeti yetkili sunucu parametresi

Kuyruk yöneticisi `ReportingService` kısmına ilişkin **ServiceProxy** özniteliğinin değeri doğru yapılandırılmadı. Kuyruk yöneticisi hizmete kaydolmaz. **ServiceProxy** parametresi geçerli bir `http://` URL 'sine ayarlanmalıdır. HTTPS ve SOCKS gibi diğer yetkili sunucu protokolleri desteklenmez.

## IBM MQ ' in Salesforce push konuları ve platform olaylarıyla kullanılmak üzere yapılandırılması

IBM MQ Bridge to Salesforce ürününü yapılandırıp çalıştırarak Salesforce ve IBM MQ ağınıza güvenlik ve bağlantı kurmak için bu bilgileri kullanın.



## Başlamadan önce

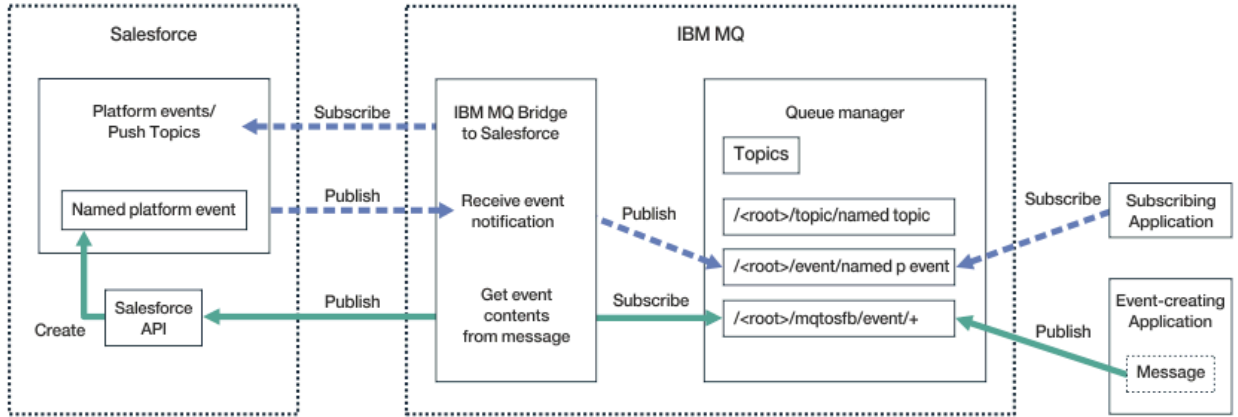
**Not:** **Deprecated** IBM MQ Bridge to Salesforce , 22 Kasım 2022 'deki tüm yayınlarda kullanımdan kaldırılmıştır (bkz. ABD Duyurusu mektubu 222-341). Salesforce bağlantılığı, IBM App Connect ile ya da IBM Cloud Pak for Integration ile sağlanan App Connect yetenekleriyle elde edilebilir.

- IBM MQ Bridge to Salesforce , x86-64 (64 bit) için Linux üzerinde kullanılabilir. Köprü, IBM WebSphere MQ 6.0 üzerinde ve daha önce çalışan kuyruk yöneticilerine bağlanmak için desteklenmez.
- IBM MQ 9.2.0 ' den bir kuyruk yöneticisi, uygun şekilde yapılandırıldığı birden çok köprü eşgörunümünü destekleyebilir. Daha fazla bilgi için bkz. [“IBM MQ Bridge to Salesforce için ek yapılandırma seçenekleri” sayfa 843](#) .
- **MQSeriesSFBridge** paketini kurun. Daha fazla bilgi için bkz. [Installing IBM MQ server on Linux and IBM MQ rpm components for Linux systems](#).

## Bu görev hakkında

Salesforce , bulut tabanlı bir müşteri ilişkileri yönetimi platformudur. Müşteri verilerini ve etkileşimlerini yönetmek için Salesforce kullanıyorsanız, daha sonra IBM MQ kuyruk yöneticinizde yayınlanabilecek Salesforce push konularına ve platform olaylarına abone olmak için IBM MQ Bridge to Salesforce kullanabilirsiniz. Bu kuyruk yöneticisine bağlanan uygulamalar, konu ve platform olay verilerini kullanışlı bir şekilde kullanabilir. Köprüyü, Salesforce içinde platform olayları için olay iletileri oluşturmak üzere de kullanabilirsiniz.

IBM MQ Bridge to Salesforce ürününe genel bakış için [Şekil 1](#) içindeki çizgeye bakın.



Şekil 97. IBM MQ Bridge to Salesforce

Push konuları, Salesforce içindeki kayıtlarda yapılan değişikliklere ilişkin bildirimleri almak için Force . com Streaming API ' yi kullanmak üzere tanımladığınız sorgulardır. Push konularını yapılandırma ve Streaming API ' yi kullanma hakkında daha fazla bilgi için bkz. [Introducing Streaming API](#) ve [Working with PushTopics](#).

Platform olayları, Force . com platformunun ürettiği ya da kullandığı olay verilerini belirlemek için tanımlanabilen özelleştirilebilir olay iletileridir. Platform olayları ve Salesforce olayları arasındaki fark hakkında daha fazla bilgi için bkz. [Kurumsal ileti sistemi platformu olayları](#) ve [Salesforce olayları](#) arasındaki fark nedir.

- Konuları ve platform olaylarını iletme için abone olma yapılandırması oluşturmak üzere bkz. [“IBM MQ Bridge to Salesforce ' nin yapılandırılması” sayfa 838](#).
- Salesforce platformu olaylarına ilişkin olay iletileri oluşturmaya ilişkin yapılandırmayı oluşturmak için bkz. [“Salesforce platformu olayları için olay iletileri oluşturma” sayfa 845](#).

Köprüdeki verileri, IBM MQ Console aracılığıyla ve **amqsrua** komutuyla **-p** parametresini kullanarak iki şekilde izleyebilirsiniz. Genel köprü durumu için bir veri kümesi yayınlanır:

- Bir aralıkta ( STATUS/PUSHTOPIC ağacının altında) işlenen gönderme konusu iletilerinin toplam sayısı.
- Bu aralıkta görülen gönderme konularının sayısı.
- Bir aralıkta ( STATUS/PLATFORM ağacının altında) işlenen toplam platform olayı sayısı.
- Bu aralıkta görülen platform olaylarının sayısı.
- Bir aralıkta ( STATUS/MQPE ağacının altında) işlenen toplam IBM MQ tarafından oluşturulan platform olayı sayısı.
- Bu aralıkta görülen IBM MQ tarafından oluşturulan platform olaylarının benzersiz sayısı.
- Bu aralıkta görülen IBM MQ tarafından oluşturulan platform olaylarının yayınlarının başarısız sayısı.

Yapılandırılan her Salesforce konusu için başka bir ileti yayınlanır. IBM MQ konusu, nesne adında tam Salesforce konu adını ve /event ya da /topic ögesini kullanır:

- Bir aralıkta işlenen iletilerin sayısı.

IBM MQ Console ürününü köprü verilerini izleyecek şekilde yapılandırmak için [IBM MQ Bridge to Salesforce](#) ürününün Yapılandırılması içindeki 9 ve 10 numaralı adımlara bakın. **amqsrua** komutunu kullanma hakkında bilgi için bkz. [IBM MQ Bridge to Salesforce](#) izleme.

IBM MQ Bridge to Salesforce' yi yapılandırmak ve çalıştırmak için aşağıdaki adımları izleyin:

## Yordam

1. IBM MQ Bridge to Salesforce' yi yapılandırın.
2. Salesforce platformu olayları için olay iletileri oluşturun.
3. IBM MQ Bridge to Salesforce komutunu çalıştırın.

## İlgili görevler

[IBM MQ Bridge to Salesforce izleme](#)

## İlgili başvurular

[runmqsfb \( IBM MQ Bridge to Salesforce komutunu çalıştırın\)](#)

Linux

Deprecated

## IBM MQ Bridge to Salesforce ' nin yapılandırılması

Yapılanış kütüğünü yaratmak ve Salesforce gönderme konularını ve altyapı olaylarını IBM MQ kuyruk yöneticinize bağlamak için IBM MQ değiştiricelerini yapılandırabilir ve IBM MQ Bridge to Salesforce değiştiricelerini girebilirsiniz.

## Başlamadan önce

**Not:** **Deprecated** IBM MQ Bridge to Salesforce , 22 Kasım 2022 tarihli tüm yayınlarda kullanımdan kaldırılmıştır (bkz. [ABD Duyuru mektubu 222-431](#)). Salesforce bağlantılığı, IBM App Connect ile ya da IBM Cloud Pak for Integration ile sağlanan App Connect yetenekleriyle elde edilebilir.

Bu görevi başlatmadan önce, MQSeriesSFBridge paketini IBM MQ kuruluşunuzda bir x86-64 Linux platformuna kurduğunuzdan emin olun.

Daha fazla bilgi için bkz. [Installing IBM MQ server on Linux and IBM MQ rpm components for Linux systems](#).

## Bu görev hakkında

Bu görev, IBM MQ Bridge to Salesforce yapılanış kütüğünü yaratmak ve Salesforce ve IBM MQ kütüğüne başarıyla bağlanmak için gereken en alt düzey kuruluşu gerçekleştirerek Salesforce gönderme konularına ve altyapı olaylarına abone olmanızı sağlar. Tüm parametrelere ilişkin anlam ve seçenekler hakkında daha fazla bilgi için [runmqsfb](#) komutuna bakın. Kendi güvenlik gereksinimlerinizi göz önünde bulundurmanız ve devreye almanıza uygun parametreleri özelleştirmeniz gerekir.

Salesforce platformu olaylarına ilişkin olay iletileri oluşturmaya ilişkin yapılandırmayı oluşturmak için bkz. [“Salesforce platformu olayları için olay iletileri oluşturma” sayfa 845.](#)

### **Salesforce push konularına ve platform olaylarına abone olma**

IBM MQ Bridge to Salesforce hem Salesforce hem de IBM MQ ile bağlantı kurduğunda, Salesforce push konularına ve platform olaylarına abonelikler oluşturur. Köprü'nün abone olmak istediği push konusu ya da platform olayı adı, bağlantı kurulmadan önce yapılandırma dosyasına eklenmeli ya da komut satırına eklenmelidir.

Yapılandırma özniteliklerinden biri IBM MQ konu ağacının köküdür ve olaylar bu kök altında yayınlanır. Köprü bu köke erişir ve tam Salesforce konu adını ekler; örneğin, /MQ/SF/ROOT/topic/EscalatedCases. IBM MQ 'e bağlanan izleme konusu ve uygulamaları, /event/NewCustomer\_\_ealtındaki /topic/EscalatedCases ve platform olayları altında push konularını arayabilir.

Yayınlanan ileti, denetim bilgilerini ve istenen veri alanlarını içeren veri yapısını içerir. Push konuları için veri yapısı bir **subject** ve platform olayları için yapı **payload** şeklindedir. Köprü, Salesforce içinde tanımlanmamışsa bir konuya ya da olaya abone olamaz. Köprü, bir konuya abone olmaya çalışırken bir hatayla karşılaşır, köprü durur.

Konu nesnesinin IBM MQ içinde tanımlanması gerekmez, ancak ağaçtaki en yakın üst öğeye dayalı olarak uygun yetkiler var olmalıdır. Yeniden yayınlanan ileti, varsayılan olarak özgün iletiden yalnızca ilgili veri yapısını içerir. Denetim bilgileri kaldırılır. Platform olayları için, yayın bir bilgi yükü yapısına sahiptir. **Köprü programının davranışı** yapılandırma parametreleri kümesindeki **Publish control data with the payload** yapılandırma seçeneği, denetim verileri de dahil olmak üzere tüm iletinin yeniden yayınlanmasını sağlar. Daha fazla bilgi için bkz. [Yapılandırma parametreleri](#).

Her bir push konusu ve platform olayı, Salesforce'den yayın sırasında ilişkili bir *ReplyID* 'ye sahiptir. *ReplyID* , sunucuyla bağlantı kurulduğunda yayının başlangıç noktasını istemek için kullanılabilir. Salesforce , 24 saate kadar bir geçmiş sağlar ve köprü'nün, oluşturuldukları sırada başlatılmamış olsa da en son anında iletme konularını ve platform olaylarını kaçırmamasına olanak tanır. Köprü, iki hizmet kalitesi kipini destekler:

#### **En çok bir kez**

Köprü, yeniden başlatma için *ReplyId* öğesini kullanmıyor. Köprü yeniden başlatıldıktan sonra yalnızca yeni oluşturulan push konuları ve platform olayları işlenir. Eksik yayınlarla ilgilenmek için başvurular hazırlanmalıdır. *ReplyId* , köprü tarafından izlenmeye devam eder ve bir kuyruğa katılaştırılır; böylece köprü, diğer hizmet kalitesiyle yeniden başlatılabilir ve geçerli durumu bilir.

#### **En az bir kez.**

*ReplyId* , köprü tarafından izlenir ve bir kuyruğa katılaştırılır. Köprü yeniden başlatıldığında, sunucudan yayınların başlangıç noktasını istemek için kalıcı *ReplyId* kullanılır. Aralığın 24 saatten fazla olmaması şartıyla, eski yayınlar gönderilir. Bir konuya ilişkin *ReplyId* her iletide sertleşmez. Düzenli aralıklarla ve köprü kapatıldığında kalıcı bir iletide yazılır. Tekrarlanan yayınları görmek için başvurular hazırlanmalıdır.

*ReplyId* , yeni tanımlanan bir kuyruğa ileti olarak yazılır. Köprü başlatılmadan önce bu kuyruğu ( **SYSTEM.SALESFORCE.SYNCQ**) tanımlamanız gerekir. **SYSTEM.SALESFORCE.SYNCQ** yoksa, hizmet kipinin kalitesinden bağımsız olarak köprü devam etmez. İlgili özniteliklerle kuyruk yaratmak için bir MQSC komut dosyası sağlanır. Köprü programının yalnızca bir eşgörünümünün **SYSTEM.SALESFORCE.SYNCQ** kuyruğunu güncelleyebildiğinden emin olmak için kuyruk DEFSOPT (EXCL) NOSHARE seçeneğiyle yapılandırılmalıdır.

Platform olaylarına ilişkin olay iletileri oluşturmak üzere yapılandırma oluşturmak için bkz. [“Salesforce platformu olayları için olay iletileri oluşturma” sayfa 845.](#)

### **Yordam**

1. Bir kuyruk yöneticisi yaratın ve başlatın.
  - a) Bir kuyruk yöneticisi yaratın; örneğin, SQM1.

strmqm SQM1

b) Kuyruk yöneticinizi başlatın.

strmqm SQM1

2. **Not:** Var olan oturum açma ve güvenlik Salesforce kimlik bilgilerini ve kendinden onaylı sertifikayı kullanmak için "3" sayfa 840. adıma geçin.

İsteğe bağlı: Salesforce hesabınız için bir güvenlik simgesi oluşturun.

a) Salesforce hesabınızda oturum açın.

b) Salesforce yardım makalesinde yer alan adımları izleyerek güvenlik belirtecini oluşturun ya da sıfırlayın: Güvenlik belirtecini sıfırlayın.

3. Salesforce'inde CA imzalı bir güvenlik sertifikası oluşturun.

a) **Force.com Ana Sayfa** sayfanızın **Yönetim** menüsünden **Güvenlik denetimleri** seçeneğini belirleyin ve **Sertifika ve Anahtar Yönetimi** seçeneğini belirleyin.

**Sertifika ve Anahtar Yönetimi** sayfası açılır.

b) **Create CA-Signed certificate**(CA İmzalı sertifika yarat) seçeneğini tıklattın.

**Sertifikalar** sayfası açılır.

c) **Etiket** alanına sertifika için bir ad girin, Tabtuşuna basın ve **Kaydet'** i tıklattın.

Sertifika ve Anahtar Ayrıntısı bilgileri görüntülenir.

d) **Listeye dön: Sertifikalar ve anahtarlar** seçeneğini tıklattın.

e) **Anahtar Deposuna Aktar** seçeneğini tıklattın.

f) Anahtar deposu için bir parola girin ve **Dışa Aktar** düğmesini tıklattın.

g) Dışa aktarılan anahtar deposunu yerel dosya sisteminize kaydedin.

4. Salesforce ' den dışa aktardığınız anahtar deposunu açmak ve imzalayıcı sertifikalarını doldurmak için IBM Anahtar Yönetimi GUI 'sini kullanın.

a) IBM Key Management GUI 'sini açmak için **strmqikm** komutunu çalıştırın.

Daha fazla bilgi için bkz. Sayısal sertifikaları yönetmek için runmqckm, runmqakm ve strmqikm kullanılması.

b) **Anahtar veritabanı dosyası aç** ' ı tıklattın ve Salesforce anahtar deposunun konumuna göz atın.

c) **Aç**' ı tıklattın, **Anahtar veritabanı tipi** seçeneklerinden **JKS** seçeneğini belirlediğinizden emin olun ve **Tamam** ' ı tıklattın.

d) 3fadımında anahtar deposu için oluşturduğunuz parolayı girin ve **Tamam** düğmesini tıklattın.

e) **Anahtar veritabanı içeriği** seçeneklerinden **İmzalayan Sertifikaları** seçeneğini belirleyin.

f) **Doldur** düğmesini tıklattın.

g) **CA Sertifikaları Ekle** listesinden **Verisign Inc.** onay kutusunu seçin ve **Tamam** düğmesini tıklattın.

5. İsteğe bağlı: Salesforce hesabınızda IBM MQ Bridge to Salesforce için bir uygulama bağlantısı oluşturarak OAuth tüketici anahtarı ve güvenlik dizgisi oluşturun.

IBM MQ Bridge to Salesforce ürününü üretim ortamlarında kullanırken **Consumer Key** ve **Consumer Secret** kodlarına gereksinim duyarsınız.

a) **Force.com Giriş** sayfanızın **Oluştur** menüsünden **Oluştur**, ardından **Uygulamalar** seçeneğini belirleyin.

Uygulamalar sayfası açılır.

b) **Bağlantılı Uygulamalar** bölümünden **Yeni** ' yi tıklattın.

**Yeni Bağlantılı Uygulama** sayfası açılır.

c) **Bağlantılı Uygulama Adı** alanında IBM MQ Bridge to Salesforce için bir ad girin; örneğin, **MQBridgeToSalesforce**.

d) **API Adı** ' nı girin.

Sekme tuşuyla sonraki alana geçerseniz, **Bağlı Uygulama Adı API Adı** ad alanına kopyalanır.

- e) **İletişim E-postası'** nı girin.
  - f) **API (OAuth Ayarlarını Etkinleştir)** bölümünde **OAuth Ayarlarını Etkinleştir** seçeneğini belirleyin. Daha sonra bu bölümdeki diğer seçenekler sunulur.
  - g) **Geri Çağırma URL** adresinizi ekleyin; örneğin, `https://www.ibm.com`.
  - h) **Seçilen OAuth Kapsamları** listesine tam erişim eklemek için **Seçilen OAuth Kapsamları** alt bölümündeki **Kullanılabilir OAuth Kapsamları** listesinden **Tam erişim (tam)** seçeneğini belirleyin ve **Ekle'** yi tıklayın.
  - i) **Kaydet** ögesini tıklayın.
  - j) **Continue** (Devam) seçeneğini tıklayın.
  - k) **Tüketici Anahtarı** ve **Tüketici Güvenlik Dizgisi** kodlarınızı not alın.
6. Kuyruk yöneticisinde gerekli eşitleme kuyruğunu yaratın.

```
cat /opt/mqm/mqsfc/samp/mqsfbSyncQ.mqsc | runmqsc SQM1
```

Eşitleme kuyruğu, uygulama ya da kuyruk yöneticisi yeniden başlatması sırasında olay durumunu korur. Kuyrukta tek bir ileti beklendiği için kuyruk derinliği küçük olabilir. Bu kuyrukta aynı anda köprünün yalnızca bir eşgörünümü çalışabilir; bu nedenle, varsayılan seçenekler dışlayıcı erişim için ayarlanır.

7. IBM MQ, Salesforce ve IBM MQ Bridge to Salesforce davranışı için bağlantı ve güvenlik parametreleriyle bir yapılandırma dosyası oluşturun.

```
runmqsf -o new_config.cfg
```

Varolan değerler köşeli ayraç içinde gösterilir. Var olan değerleri kabul etmek için Enter tuşuna basın, değerleri temizlemek için Space ardından Enter tuşuna basın ve yeni değerler eklemek için Enter tuşuna basın.

- a) SQM1: kuyruk yöneticisiyle bağlantı için değer girin

Bağlantı için gereken alt sınır değerleri kuyruk yöneticisi adı, IBM MQ temel konu kökü ve kanal adıdır.

```
Connection to Queue Manager
-----
Queue Manager or JNDI CF      : []SQM1
MQ Base Topic                : []/sf
MQ Channel                   : []A channel you have defined or for example
SYSTEM.DEF.SVRCONN
MQ Conname                   : []
MQ Publication Error Queue   : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL                  : []
JNDI implementation class    : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL           : []
MQ Userid                    : []
MQ Password                   : []
```

**Not:** Yerel olarak bağlanıyorsanız kanal adı gerekli değildir. Kuyruk yöneticisi adını ve temel konuyu, köprüyü çalıştırdığınızda daha sonra komut satırına eklenebileceği için yapılandırma dosyasında belirtmeniz gerekmez.

- b) Salesforce bağlantısı için değerleri girin:

Bağlantı için gerekli minimum değerler: Salesforce kullanıcı kimliği, parola, güvenlik simgesi ve oturum açma uç noktası. Üretim ortamlarında, OAuth güvenliği için tüketici anahtarını ve güvenlik dizgisini ekleyebilirsiniz.

```
Connection to Salesforce
-----
Salesforce Userid (reqd)     : []salesforce_login_email
Salesforce Password (reqd)  : []salesforce_login_password
Security Token (reqd)       : []Security_Token
Login Endpoint               : [https://login.salesforce.com]
```

```
Consumer ID : []
Consumer Secret Key : []
```

c) TLS bağlantıları için sertifika depoları değerlerini girin:

TLS bağlantıları için gerekli alt sınır değerleri, TLS sertifikaları ve anahtar deposu parolası için anahtar deposu yoludur. Güvenilir bir depo yolu ya da parolası sağlanmazsa, güvenilir depo ve parola için anahtar deposu ve parola parametreleri kullanılır. IBM MQ kuyruk yöneticisi bağlantınız için TLS kullanıyorsanız, aynı anahtar deposunu kullanabilirsiniz.

```
Certificate stores for TLS connections
-----
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/
SQM1/ssl/key.jks
Keystore password : []keystore_password
Trusted store for signer certificates : []
Trusted store password : []
Use TLS for MQ connection : [N]
```

d) IBM MQ Bridge to Salesforcedavranışını yapılandırmak için değerleri girin:

Bu değerlerden herhangi birini değiştirmeniz ya da sağlamanız gerekmez, ancak konu ya da platform olayı adlarını biliyorsanız, bunları buraya ekleyin. Köprüyü çalıştırmaya hazır olduğunuzda daha sonra komut satırına da eklenebilir. Günlük dosyasını yapılandırma dosyasında ya da komut satırında belirtmeniz gerekir.

```
Behaviour of bridge program
-----
PushTopic Names : []
Platform Event Names : []
MQ Monitoring Frequency : [30]
At-least-once delivery? (Y/N) : [Y]
Subscribe to MQ publications for platform events? (Y/N) : [N]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

8. İsteğe bağlı: Programın yürütülmesini denetlemek için IBM MQ hizmetini oluşturun. Örnek `mqsfbService.mqsc` dosyasını, yeni yaratılan yapılanış kütüğünü gösterecek şekilde düzenleyin ve komut değiştirgelerinde başka değişiklikler yapın.

```
cat modified mqsfbService.mqsc | runmqsc SQM1
```

9. İsteğe bağlı: IBM MQ Consoleürününü kurmak için [Getting started with the IBM MQ Console](#) başlıklı konudaki yönergeleri izleyin.
10. İsteğe bağlı: IBM MQ Bridge to Salesforce ' i köksüz kullanıcı olarak çalışacak şekilde yapılandırın.

IBM MQ Bridge to Salesforce ' i *köksüz kullanıcı* olarak çalıştırabilmek için, örneğin *köksüz kapsayıcı* içinde, köprü işlemi çalıştıran kullanıcı için okuma/yazma erişimi sağlamak üzere Java `userRoot` ve `systemRoot` dizinlerinin doğru bir şekilde ayarlanması gerekir. Bunu yapmak için aşağıdaki JVM özelliklerini ayarlayın:

```
export MQSFB_EXTRA_JAVA_OPTIONS="-
Djava.util.prefs.userRoot=directory_with_read_write_access"
```

```
export MQSFB_EXTRA_JAVA_OPTIONS="-
Djava.util.prefs.systemRoot=directory_with_read_write_access"
```

## Sonuçlar

IBM MQ Bridge to Salesforce ' in Salesforce ' in konuları ve platform olaylarını iletme ve bunları IBM MQ ağızında yayınlamak için kullandığı yapılandırma dosyasını oluşturdu.

## Sonraki adım

“IBM MQ Bridge to Salesforce ' in çalıştırılması” sayfa 851adımları boyunca çalışın.

### İlgili görevler

[IBM MQ Bridge to Salesforce izleme](#)

## İlgili başvurular

[runmqsfb \( IBM MQ Bridge to Salesforcekomutunu çalıştırın\)](#)

Linux

Deprecated

## IBM MQ Bridge to Salesforce için ek yapılandırma

### seçenekleri

IBM MQ 9.2.0'den, "gelen" ( Salesforce' dan oluşturulan, IBM MQ uygulamalarına yayınlanan olaylar) ve "giden" ( Salesforce' e gönderilen olayları yayınlayanIBM MQ uygulamaları) işleriyle ilgili olmak üzere iki ana ek topoloji sınıfına izin veren ek yapılandırma seçenekleri kullanılabilir. Ayrıca, izleme ve günlüğe kaydetme çalışma yönteminde de bir değişiklik vardır.

### IBM MQ 9.1.0 IBM MQ Bridge to Salesforce ürünündeki değişiklikler

IBM MQ 9.2.0' da, günlük dosyasının döndürmeye başlaması dışında, varsayılan olarak, IBM MQ 9.1.0 köprüsünden davranış üzerinde herhangi bir değişiklik yoktur. Ek bilgi için bkz. [“Dönen günlükler” sayfa 844](#) .

Ana değişiklik, bir kuyruk yöneticisinin birden çok köprü yönetim ortamını desteklemesidir. Bu özelliği ve ek topolojilerin geri kalanını etkinleştirmek için el ile yapılandırma değişiklikleri yapmanız gerekir.

Ek yapılandırma seçenekleri hakkında daha fazla bilgi için [runmqsfb](#) ve düzeltilen yapılandırma bilgilerinin bir örneği için [“IBM MQ Bridge to Salesforce için örnek yapılandırma çıkışı” sayfa 844](#) başlıklı konuya bakın.

### Ayrılmış gelen iş

Köprünün birden çok eşgörünümü Salesforce 'den IBM MQ' e gelen işleri işleyebilir, ancak bunlar bağımsız Salesforce gönderme konuları ve olayları kümeleri üzerinde çalışmalıdır. Aksi takdirde, olayların yinelenmesini durdurmak için köprüler arası iletişim kuralı olmadığından, IBM MQ uygulamaları tarafından görülen olayların yinelenme olasılığı vardır. Her eşgörünüm, **ReplayIdd** dosyasını tutmak için kendi yapılandırılır uyumlulaştırma kuyruğunu kullanır.

Bu, aşağıdaki durumlarda yararlı olabilir:

- Farklı Salesforce konularının farklı güvenlik yetkileri vardır. Her köprü örneğinin Salesforce' e erişmek için farklı bir kimlik bilgileri kümesi vardır.
- Salesforce ' in tek bir köprü için çok fazla olmasından gelen iş yükü hakkında endişeleriniz var. Bu nedenle, bir köprüden geçen "A-M" ve başka bir köprüden geçen "N-Z" ile bölümlenecek konuları düzenleyebilirsiniz.

### Paylaşılan giden işi

Köprü, IBM MQ 'den Salesforce' e gönderilen giden işleri desteklemek için birden çok örneği destekler. Köprünün bir eşgörünümü başarısız olursa, aynı kuyruk yöneticisinde aynı konulara abone olan diğer yönetim ortamları yayınları işlemeye devam edebilir.

**Not:** Bunun için IBM MQ konu yapılandırmasında herhangi bir değişiklik gerekmez.

Bu işbirliği yapan eşgörünümlerin, eşgörünümlerden birinin Salesforce' den gelen işleri işlemesi için ayarlanması gerekir; bu eşgörünümün uyumlulaştırma kuyruğuna özel erişimi olmalıdır.

Bu, aşağıdaki konularla ilgili endişeleriniz olduğunda yararlı olabilir:

- IBM MQ' den gelen iş yükü. Salesforce ' e yönelik istekler zamanuyumlu olduğundan köprü, bir iletiyi işlerken yeni işi işleyemez. Birden çok tüketicinin olması bu durumu hafifletir.
- Kullanılabilirlik mimarisi. Örneğin, artık daha iyi hata durumunda yedek sisteme geçiş ve olağanüstü durumdan kurtarma seçenekleriyle ayrı veri merkezlerinde birden çok eşgörünüm çalıştırılabilir. IBM MQ istemcisi olarak çalışılması, köprüyü kuyruk yöneticisi konumundan da ayırır.



## İzleme ve hata ayıklama etkileşimi

IBM MQ 9.2.0' dan hata ayıklama işareti, IBM MQ 9.1.0 için olduğu gibi davranmaya devam eder. -d1 köprüde hata ayıklama bilgileri verir ve -d2 önkoşul olan bileşenler için hata ayıklama günlüğünü açar. Ancak, köprüyü başlattığınızda IBM MQ izlemesini etkinleştirdiyse, -d2 düzey raporlaması otomatik olarak açılır.

## Dönen günlükler

IBM MQ 9.2.0' den günlük dosyasının varsayılan davranışı, her biri 2 MB boyutunda olan üç günlük dosyasına sahip olmaktır. Ek yapılandırma özelliklerini kullanarak bu değerleri geçersiz kılabilirsiniz. Günlük dosyasına ilişkin var olan yapılandırma özneliği ya da komut satırı parametresi, bir izin eklenmiş olarak günlüklerin temel adı olarak alınır.

Yapılandırılan günlük dosyası aşağıdaki özelliklere sahipse:

- Dosya tipi yok, izin dosya adının sonuna eklenir.  
Günlük dosyasının abcolarak ayarlanması, abc .0, abc .1vb. adlı günlüklerle sonuçlanır.
- Dosya tipi, izin, dosya tipinden önce eklenir.  
Günlük dosyasının abc .logolarak ayarlanması, abc .0 .log, abc .1 .logvb. adlı günlüklerle sonuçlanır.

### Notlar:

1. Köprüler isteğe bağlı kullanıcı izniyle çalışabileceğinden, günlükler için belirli bir dizini (örneğin, /var/mqm/qmgrs/<qm>/errors) zorlamak mümkün değildir.
2. Aynı bilgiler *stdout* ve *stderr* akışlarına yazılmaya devam eder.
3. Tek bir günlük dosyası yeniden açıldığında, temel yapılandırma bilgileri yeniden yazdırılır. Bilgiler, programın başlangıcında bir kez yazdırılmak yerine her zaman kullanılabilir olacaktır.

## Günlüklerin korunması

IBM MQ 9.2.0 topolojileri, köprünün belirli bir kuyruk yöneticisine karşı çalışan birden çok eşgörünümü olmasını daha olası hale getirmektedir.

Eşgörünümlerin birbiriyle etkileşimde bulunmasını önlemek ve köprünün önceki çalışmalarını önlemek için, .0 günlüğü zaten varsa köprü başlamaz.

Köprüyü başlatmadan önce günlüğün önceki kopyalarını silen ya da ada zaman damgası gibi bir şey ekleyen bir başlatma yordamı gerekir.

### İlgili görevler

[“IBM MQ ' ın Salesforce push konuları ve platform olaylarıyla kullanılmak üzere yapılandırılması” sayfa 836](#)

IBM MQ Bridge to Salesforceürünü yapılandırıp çalıştırarak Salesforce ve IBM MQ ağınıza güvenlik ve bağlantı kurmak için bu bilgileri kullanın.

[IBM MQ Köprüsünün Salesforce ' a izlenmesi](#)

### İlgili başvurular

[mqfsb çalıştırma](#)

## Linux IBM MQ Bridge to Salesforce için örnek yapılandırma çıkışı

IBM MQ 9.1.0 IBM MQ Bridge to Salesforceüründeki değişiklikleri gösteren örnek bir yapılandırma çıkışı.

```
IBM MQ Bridge to Salesforce
5724-H72 (C) Copyright IBM Corp. 2017, 2024.
Level : <<unknown>>
```

```
Enter new values for the configuration attributes. The
```



current settings are shown.  
Press ENTER to accept current values; use SPACE+ENTER  
to clear values.

#### Connection to Queue Manager

```
-----  
Queue Manager or JNDI CF : [V9000_A]  
MQ Base Topic           : [ /sf ]  
MQ Channel              : [ ]  
MQ Conname              : [ ]  
MQ Publication Error Queue : [SYSTEM.SALESFORCE.DEADQ]  
MQ Replay Status Queue  : [SYSTEM.SALESFORCE.SYNCQ]  
MQ CCDT URL             : [ ]  
JNDI implementation class : [com.sun.jndi.fscontext.RefFSContextFactory]  
JNDI provider URL       : [ ]  
MQ Userid               : [ ]  
MQ Password             : [ ]
```

#### Connection to Salesforce

```
-----  
Salesforce Userid (reqd) : [johndoe@yourenterprise.com]  
Salesforce Password (reqd) : [*****]  
Security Token           : [*****]  
Login Endpoint          : [https://login.salesforce.com]  
Consumer Key            : [3MVG9HxRZv05HarQhSy89qSKYNr1gDcv1wE3zN5kyFAa4Wxt]  
Consumer Secret        : [*****]
```

#### Certificate stores for TLS connections

```
-----  
Personal keystore for TLS certificates : [/var/mqm/ssl/key.jks]  
Keystore password                   : [*****]  
Trusted store for signer certificates : [ ]  
Trusted store password              : [ ]  
Use TLS for MQ connection          : [N]
```

#### Event processing

```
-----  
PushTopic Names                   : [ ]  
Platform Event Names              : [ ]  
At-least-once delivery for Salesforce events? (Y/N) : [N]  
At-least-once delivery for MQ publications? (Y/N) : [N]  
Subscribe to MQ publications for platform events? (Y/N) : [Y]  
Publish control data with the payload? (Y/N) : [Y]  
Treat unknown Salesforce topic as warning (Y/N) : [N]
```

#### Behaviour of bridge program

```
-----  
Bridge unique identifier          : [ ]  
MQ Monitoring Frequency           : [30]  
Delay before starting to process events : [0]  
Continue to retry after maximum reconnection attempts (Y/N) : [N]  
Runtime logfile for copy of stdout/stderr : [/tmp/runmqsfb.log]  
Number of logfiles                : [3]  
Maximum size of each logfile      : [2097152]  
Done.
```

## İlgili başvurular

[mqfsb çalıştırma](#)

Linux

Deprecated

## Salesforce platformu olayları için olay iletileri oluşturma

Yapılandırma dosyasını oluşturmak için IBM MQ 'i yapılandırabilir ve IBM MQ Bridge to Salesforce parameters girin ve Salesforce platformu olayları için olay iletileri oluşturmak üzere köprüyü kullanın.

### Başlamadan önce

- **MQSeriesSFBridge** paketini IBM MQ kuruluşunuzda bir x86-64 Linux platformuna kurduunuz.

### Bu görev hakkında

Bu görev, IBM MQ Bridge to Salesforce yapılandırma kütüğünü yaratmak ve Salesforce ve IBM MQ kütüklerine başarıyla bağlanmak için gereken kuruluş alt sınırı boyunca ilerlemenizi sağlar; böylece, Salesforce altyapısı olayları için olay iletileri yaratabilirsiniz. Tüm parametrelere ilişkin anlam ve seçenekler

hakkında daha fazla bilgi için `runmqsfb` komutuna bakın. Kendi güvenlik gereksinimlerinizi göz önünde bulundurmanız ve devreye alımınıza uygun parametreleri özelleştirmeniz gerekir.

Konuları ve platform olaylarını iletme için abone olma yapılandırması oluşturmak üzere bkz. [“IBM MQ Bridge to Salesforce” nin yapılandırılması](#) sayfa 838.

### Salesforce platformu olayları için olay iletileri oluşturma

Bir kuyruk yöneticisi konusuna konan iletiler yaratmak için IBM MQ uygulamasını kullanabilirsiniz `/root/mqtosfb/event/+`. Köprü konuya abone olur, iletilerden içerik alır ve bunu bir Salesforce platformu olayına ilişkin olay iletilerini yayınlamak için kullanır. Platform olaylarıyla ilgili daha fazla bilgi için Salesforce geliştirici belgelerinde [Platform olaylarıyla özel bildirimler sağlama](#) başlıklı konuya bakın.

Köprünün olay iletileri oluşturmasını sağlamak için, konuları ve platform olaylarını iletme üzere abone olmak için kullanılanlar için ek olan iki öznitelik sağlamanız gerekir:

- **Kuyruk Yöneticisine Bağlantı** için köprü yapılandırma özniteliklerinde **MQ Publication Error Queue** adını oluşturun ve ekleyin.
- **Köprü programının davranışı** 'nı tanımlamak için köprü yapılandırma özniteliklerinde **Subscribe to MQ publications for platform events** seçeneğini Yolarak ayarlayın.

Salesforce içinde bir platform olayı oluşturmanız ve o platform olayı için olay iletileri oluşturmak üzere köprüyü kullanmadan önce içerik alanlarını tanımlamanız gerekir. Platform olay adı ve içeriği, köprü tarafından işlenen IBM MQ iletisini nasıl biçimlendirmeniz gerektiğini belirler. Örneğin, Salesforce platform olayınız **Object name** `MQPlatformEvent1` ise ve iki özel tanımlı alanınız **API name** `MyText__c` ve `Name__c` içeren alanlarıysa, `/root/mqtosfb/event/MQPlatformEvent1__e` konusunda yayınlanan IBM MQ iletiniz aşağıdaki gibi doğru biçimlendirilmiş bir JSON olmalıdır:

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

İleti, IBM MQ Bridge to Salesforce tarafından MQFMT\_STRING biçimli ileti gövdesi olarak tanınacak şekilde biçimlendirilmelidir.

Salesforce içinde platform olayınızı oluşturmak için [“7” sayfa 848](#) . adımı bakın ya da olay iletileri oluşturmak istediğiniz bir platform etkinliğiniz varsa bu adımı atlayın. IBM MQ iletinizi, Salesforce platform etkinliğinizde ayarlanan alanlarla eşleşecek şekilde biçimlendirmeniz gerekir. Salesforce platformu olayındaki alanlar isteğe bağlı ya da zorunlu olarak atanabilir. Daha fazla bilgi için Salesforce geliştirici belgelerinde [Platform Event Fields](#) başlıklı konuya bakın.

Köprü çalışırken, belirtilen IBM MQ konusuna abone olur.

- Köprü yapılandırmasında **At-most-once** hizmet kalitesini belirtirseniz, köprünün yaptığı abonelik kalıcı değildir. Köprü çalışmamışken IBM MQ uygulamaları tarafından yapılan yayınlar işlenmez.
- Köprü yapılandırmasında **At-least-once** hizmet kalitesini belirtirseniz, köprünün yaptığı abonelik dayanıklıdır. Bu, köprünün köprü çalışmamışken IBM MQ uygulamaları tarafından yapılan yayınları işleyebileceği anlamına gelir. Sürekli abonelikler, bilinen bir abonelik ve istemci tanıtıcısı gerektirir. Köprü, abonelik adı olarak `D_SUB_RUNMQSFB` ve istemci tanıtıcısı olarak `runmqsfb_1` kullanır.

Köprü, olay iletileri oluşturmak için değil, Salesforce push konularına ve platform olaylarına abone olmak için kullanılıyorsa, yapılandırmanın değiştirilmesi ve aboneliğin artık kalmış olması durumunda sürekli aboneliği silmeye çalışır.

Köprünün yarattığı sürekli abonelikleri aşağıdaki gibi kaldırabilirsiniz:

### IBM MQ Explorer kullanın.

Köprünün kullandığı kuyruk yöneticisine ilişkin **abonelikler klasörünü** açın ve konu dizgisinin `/sf/mqtosfb/event+olduğu` `:D_SUB_RUNMQSFB` ile biten abonelik adını arayın. Abonelik adını sağ tıklayın ve Sil 'i tıklayın. Aboneliğin kullanımda olduğunu gösteren bir hata alırsanız, köprünüz hala çalışıyor olabilir. Köprüyü durdurun ve aboneliği silmeyi yeniden deneyin.

### Aboneliđi bulmak ve silmek için runmqsc öđesini kullanın.

**runmqsc** arabirimini bařlatın ve DISPLAY SUB (\*) komutunu alıřtırın. :D\_SUB\_RUNMQSFBile biten **SUB** abonelik adını arayın. Alt komutu silin ve silmek istediđiniz aboneliđin **SUBID** öđesini ekleyin. Örneđin, DELETE SUB SUBID(414D5120514D3120202020202020205C589459987E8620)

### Durdurun ve At-most-once hizmet kalitesi köprüyü bařlatın.

Köprüyü **At-least-once** hizmet kalitesi At-least-once delivery? (Y/N) : [Y] ile bařlattıysanız, oluřturulan abonelik dayanıklıdır. Aboneliđi silmek için, yapılandırma dosyanızdaki hizmet kalitesini At-least-once delivery? (Y/N) : [N] olarak deđiřtirin ve köprüyü yeniden bařlatın. Sürekli abonelik silinir ve kalıcı olmayan bir abonelik yaratılır.

## Yordam

1. Bir kuyruk yöneticisi yaratın ve bařlatın.

a) Bir kuyruk yöneticisi yaratın; örneđin, PEQM1.

```
crtmqm PEQM1
```

b) Kuyruk yöneticinizi bařlatın.

```
strmqm PEQM1
```

2. **Not:** Var olan oturum açma ve güvenlik Salesforce kimlik bilgilerini ve kendinden onaylı sertifikayı kullanmak için 4. adıma geçin.

İsteđe bađlı: Salesforce hesabınız için bir güvenlik simgesi oluřturun.

a) Salesforce hesabınızda oturum açın.

b) [Salesforce yardım makalesinde yer alan adımları izleyerek güvenlik belirtecinizi oluřturun ya da sıfırlayın: Güvenlik belirtecinizi sıfırlayın.](#)

3. Salesforce içinde kendinden onaylı bir güvenlik sertifikası oluřturun.

a) **Force.com Ana Sayfa** sayfanızın **Yönetim** menüsünden **Güvenlik denetimleri** seeneđini belirleyin ve **Sertifika ve Anahtar Yönetimi** seeneđini belirleyin.

**Sertifika ve Anahtar Yönetimi** sayfası açılır.

b) **Kendinden Onaylı Sertifika Oluřtur** ' u tıklatın.

**Sertifikalar** sayfası açılır.

c) **Etiket** alanına sertifika için bir ad girin, Tabtuřuna basın ve **Kaydet** ' i tıklatın.

Sertifika ve Anahtar Ayrıntısı bilgileri görüntülenir.

d) **Listeye dön: Sertifikalar ve anahtarlar** seeneđini tıklatın.

e) **Anahtar Deposuna Aktar** seeneđini tıklatın.

f) Anahtar deposu için bir parola girin ve **Dıřa Aktar** düđmesini tıklatın.

g) Dıřa aktarılan anahtar deposunu yerel dosya sisteminize kaydedin.

4. Salesforce ' den dıřa aktardığınız anahtar deposunu açmak ve imzalayıcı sertifikalarını doldurmak için IBM Anahtar Yönetimi GUI ' sini kullanın.

a) IBM Key Management GUI ' sini açmak için **strmqikm** komutunu alıřtırın. Daha fazla bilgi için bkz. [Sayısal sertifikaları yönetmek için runmqckm, runmqakm ve strmqikm kullanılması.](#)

b) **Anahtar veritabanı dosyası aç** ' ı tıklatın ve Salesforce anahtar deposunun konumuna göz atın.

c) **Aç** ' ı tıklatın, **Anahtar veritabanı tipi** seeneklerinden **JKS** seeneđini belirlediğinizden emin olun ve **Tamam** ' ı tıklatın.

d) 3.fadımında anahtar deposu için oluřturduđunuz parolayı girin ve **Tamam** düđmesini tıklatın.

e) **Anahtar veritabanı içeriđi** seeneklerinden **İmzalayan Sertifikaları** seeneđini belirleyin.

f) **Doldur** düđmesini tıklatın.

g) **CA Sertifikaları Ekle** listesinden **Verisign Inc.** onay kutusunu seđin ve **Tamam** düđmesini tıklatın.

5. İsteğe bağlı: Salesforce hesabınızda IBM MQ Bridge to Salesforce için bir uygulama bağlantısı oluşturarak OAuth tüketici anahtarı ve güvenlik dizgisi oluşturun.
- IBM MQ Bridge to Salesforce ürününü üretim ortamlarında kullanırken **Consumer Key** ve **Consumer Secret** kodlarına gereksinim duyarsınız.
- Force.com Giriş** sayfanızın **Oluştur** menüsünden **Oluştur**, ardından **Uygulamalar** seçeneğini belirleyin.  
**Uygulamalar** sayfası açılır.
  - Bağlantılı Uygulamalar** bölümünden **Yeni** 'yi tıklatın.  
**Yeni Bağlantılı Uygulama** sayfası açılır.
  - Bağlantılı Uygulama Adı** alanında IBM MQ Bridge to Salesforce için bir ad girin; örneğin, **MQBridgeToSalesforce**.
  - API Adı** 'nı girin.  
Sekme tuşuyla sonraki alana geçerseniz, **Bağlı Uygulama Adı API Adı** ad alanına kopyalanır.
  - İletişim E-postası** 'nı girin.
  - API (OAuth Ayarlarını Etkinleştir)** bölümünde **OAuth Ayarlarını Etkinleştir** seçeneğini belirleyin.  
Daha sonra bu bölümdeki diğer seçenekler sunulur.
  - Geri Çağırma URL** adresinizi ekleyin; örneğin, <https://www.ibm.com>.
  - Seçilen OAuth Kapsamları** listesine tam erişim eklemek için **Seçilen OAuth Kapsamları** alt bölümündeki **Kullanılabilir OAuth Kapsamları** listesinden **Tam erişim (tam)** seçeneğini belirleyin ve **Ekle** 'yi tıklatın.
  - Kaydet** öğesini tıklatın.
  - Continue** (Devam) seçeneğini tıklatın.
  - Tüketici Anahtarı** ve **Tüketici Güvenlik Dizgisi** kodlarınızı not alın.
6. Kuyruk yöneticisinde gereken uyumlulaştırma ve hata kuyruklarını yaratın.

```
cat /opt/mqm/mqsf/samp/mqsfSyncQ.mqsc | runmqsc PEQM1
```

Eşitleme kuyruğu, uygulama ya da kuyruk yöneticisi yeniden başlatması sırasında olay durumunu korur. Kuyrukta tek bir ileti beklediği için kuyruk derinliği küçük olabilir. Bu kuyrukta aynı anda köprünün yalnızca bir eşgörünümü çalışabilir; bu nedenle, varsayılan seçenekler dışlayıcı erişim için ayarlanır. Altyapı olaylarına ilişkin olay iletileri yaratmak için köprüyü kullanabilmeniz için önce hata kuyruğu yaratılmalıdır. Hata kuyruğu, Salesforce tarafından başarıyla işlenemeyen iletiler için kullanılır. “8.a” sayfa 849. adımda gösterildiği gibi, köprü yapılandırma parametresi bölümüne **Connection to Queue Manager** hata kuyruğu adını eklemeniz gerekir.

7. İsteğe bağlı: Salesforce hesabınızda bir platform olay nesnesi oluşturun.
- Force.com Ana Sayfa** sayfanızın **Geliştir** menüsünden **Platform Olayları** seçeneğini belirleyin ve **Yeni Platform Etkinliği** 'ni tıklatın.  
**Yeni Platform Etkinliği** sayfası açılır.
  - Etiket** ve **Çoğul Etiket** alanlarını doldurun.
  - Kaydet** 'i tıklatın.  
**Platform Olay Tanımlaması Ayrıntısı** sayfası açılır.
  - Özel Alanlar ve İlişkiler** 'i tanımlayın.  
Örneğin, *MyText* ve *Ad* etiketli iki metin alanı ekleyebilir ve **Veri Tipi** alan uzunluklarını sırasıyla *Metin (64)* ve *Metin (32)* olarak ayarlayabilirsiniz.

Bir platform olayı oluşturduanız ve bunun için **Custom Fields and Relationships** öğesini tanımladınız. Köprünün işlemlerini istediğiniz iletileri koyabileceğiniz IBM MQ konusu olarak platform olayı *Platform Nesnesi adı* 'nı ya da *API adı* 'nı kullanın. Örneğin, aşağıdaki JSON biçimli iletiyi /s/

mqtosfb/event/Salesforce Platform Object Name/API name konusuna eklemek için **AMQSPUBA** örneğini kullanabilirsiniz:

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

Köprü başlatıldıktan sonra ileti yaratmak için **AMQSPUBA** örneğini çalıştırabilirsiniz. *MQ installation location/samp/bin* dizininden aşağıdaki komutu verin:

```
./amqspub /sf/mqtosfb/event/Salesforce Platform Object Name/API name PEQM1
```

Komut isteminde, iletiyi JSON biçiminde girin.

8. IBM MQ, Salesforce ve IBM MQ Bridge to Salesforce davranışı için bağlantı ve güvenlik parametreleriyle bir yapılandırma dosyası oluşturun.

```
runmqsf -o new_config.cfg
```

Var olan değerler köşeli ayraç içinde gösterilir. Var olan değerleri kabul etmek için Enter tuşuna basın, değerleri temizlemek için Space ardından Enter tuşuna basın ve yeni değerler eklemek için Enter tuşuna basın.

- a) PEQM1: kuyruk yöneticisine yönelik bağlantı için değer girin:

Bağlantı için gereken en düşük değerler kuyruk yöneticisi adı, IBM MQ temel konu kökü, hata kuyruğu adı ve kanal adıdır.

```
Connection to Queue Manager
-----
Queue Manager or JNDI CF      : []PEQM1
MQ Base Topic                 : []/sf
MQ Channel                    : []A channel you have defined or for example
SYSTEM.DEF.SVRCONN
MQ Conname                    : []
MQ Publication Error Queue    : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL                   : []
JNDI implementation class     : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL             : []
MQ Userid                     : []
MQ Password                   : []
```

**Not:** Yerel olarak bağlanıyorsanız, kanal adı gerekli değildir. Kuyruk yöneticisi adını ve temel konuyu, köprüyü çalıştırdığınızda daha sonra komut satırına eklenebileceği için yapılandırma dosyasında belirtmeniz gerekmez.

- b) Salesforce bağlantısı için değerleri girin:

Bağlantı için gereken alt sınır değerleri şunlardır: Salesforce kullanıcı kimliği, parola, güvenlik simgesi ve oturum açma uç noktası. Üretim ortamlarında, OAuth güvenliği için tüketici anahtarını ve güvenlik dizgisini ekleyebilirsiniz.

```
Connection to Salesforce
-----
Salesforce Userid (reqd)      : []salesforce_login_email
Salesforce Password (reqd)    : []salesforce_login_password
Security Token (reqd)         : []Security_Token
Login Endpoint                 : [https://login.salesforce.com]
Consumer ID                   : []
Consumer Secret Key           : []
```

- c) TLS bağlantıları için sertifika depoları değerlerini girin:

TLS bağlantıları için gereken alt sınır değerleri, TLS sertifikaları ve anahtar deposu parolası için anahtar deposu yoludur. Güvenilir bir depo yolu ya da parolası sağlanmazsa, güvenilir depo ve parola için anahtar deposu ve parola parametreleri kullanılır. IBM MQ kuyruk yöneticisi bağlantınız için TLS kullanıyorsanız, aynı anahtar deposunu kullanabilirsiniz.

```
Certificate stores for TLS connections
-----
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/
PEQM1/ssl/key.jks
Keystore password                  : []keystore_password
Trusted store for signer certificates : []
```

```
Trusted store password : []
Use TLS for MQ connection : [N]
```

d) IBM MQ Bridge to Salesforcedavranışını yapılandırmak için değerleri girin:

Olay iletileri oluşturmak üzere köprüyü kullanmak için **Subscribe to MQ publications for platform events** seçeneğini varsayılan *N*'den *Y*'ye değiştirmeniz gerekir. Günlük kütüğünü yapılanış kütüğünde ya da komut satırında belirtmeniz de gerekir.

```
Behaviour of bridge program
-----
PushTopic Names : []
Platform Event Names : []
MQ Monitoring Frequency : [30]
At-least-once delivery? (Y/N) : [Y]
Subscribe to MQ publications for platform events? (Y/N) : [Y]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

9. İsteğe bağlı: Programın yürütülmesini denetlemek için IBM MQ hizmetini oluşturun. Örnek `mqsfbService.mqsc` dosyasını, yeni yaratılan yapılanış kütüğünü gösterecek şekilde düzenleyin ve komut değiştirgelerinde başka değişiklikler yapın.

```
cat modified mqsfbService.mqsc | runmqsc PEQM1
```

10. İsteğe bağlı: IBM MQ Consoleürününü kurmak için [Getting started with the IBM MQ Console](#) başlıklı konudaki yönergeleri izleyin.

11. İsteğe bağlı: IBM MQ Console yönetim ortamınızda Salesforce verilerini görüntülemek için gereçler ekleyin ve yapılandırın.

a) **Gereç ekle** öğesini tıklatın.

Yeni gereç açılır.

b) **Grafikler** seçeneğini belirleyin.

c) Yeni gerecin başlık çubuğundaki **Gereç yapılandır** simgesini tıklatın.

d) İsteğe bağlı: Bir **Pencere öğesi başlığı** girin.

e) **İzlenecek kaynak**, **Kaynak** açılan menüsünden **Salesforce Köprüsü** seçeneğini belirleyin.

f) **Resource class**(Kaynak sınıfı) açılan menüsünden **Bridge Status**(Köprü Durumu) seçeneğini belirleyin.

g) **Kaynak tipiaçılan menüsünden MQile yaratılan Platform Olayları** öğesini seçin.

h) **Kaynak öğesiaçılan menüsünden Toplam MQile yaratılan Platform Olayları** seçeneğini belirleyin.

i) **Kaydet** öğesini tıklatın.

IBM MQ Console öğesini, IBM MQ tarafından oluşturulan platform olaylarının toplam sayısını gösterecek şekilde yapılandırdınız. Köprü çalışırken ve `/sf/mqtosfb/event/Salesforce Platform Object Name/API name` konusuna ileti yerleştirmeye başladığınızda, gereç köprünün yarattığı toplam ileti olayı sayısını gösterir.

## Deprecated IBM MQ Bridge to Salesforce için ileti biçimi ve hata iletileri

IBM MQ Bridge to Salesforcetarafından işlenen iletilerin biçimlendirilmesine ilişkin bilgiler.

Bir uygulama belirli bir kuyruk yöneticisi konusuna bir ileti koyar; örneğin, `/root/mqtosfb/event/MQPlatformEvent1__e`. Köprü konuya abone olur, iletilerden içerik alır ve bunu bir Salesforce platformu olayına ilişkin olay iletilerini yayınlamak için kullanır.

Salesforce içinde bir platform olayı oluşturmanız ve o platform olayı için olay iletileri oluşturmak üzere köprüyü kullanmadan önce içerik alanlarını tanımlamanız gerekir. Platform olay adı ve içeriği, köprü tarafından işlenen IBM MQ iletisini nasıl biçimlendirmeniz gerektiğini belirler. Örneğin, Salesforce platform olayınız **Object name** `MQPlatformEvent1` ise ve iki özel tanımlı alanınız **API name** `MyText__c`

ve *Name\_\_* içeren metin alanlarıysa, */root/mqtosfb/event/MQPlatformEvent1\_\_e* konusunda yayınlanan IBM MQ iletiniz aşağıdaki gibi doğru biçimlendirilmiş bir JSON olmalıdır:

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

Köprü tarafından tüketilen ve üretilen iletiler, JSON biçimindeki metin (MQSTR) iletilerdir. Giriş iletisi basit bir JSON ' dur ve programlar bunu oluşturmak için dizgi birleştirmeyi kullanabilir.

## **hata iletileri**

Hatalar köprü tarafından saptanabilir; örneğin, ileti metin biçiminde değilse ya da Salesforce tarafından (örneğin, platform olay adı yoksa). Giriş iletisi işlenirken bir hata oluşursa, ileti, hatayı tanımlayan özelliklerle birlikte köprü hata kuyruğuna taşınır. Hata, köprüye ilişkin *stderr* akımına da yazılır.

Salesforce tarafından oluşturulan hatalar JSON ' dur. Aşağıda, yanlış biçimlendirilmiş iletilerin neden olduğu bazı hatalar yer almaktadır:

Hatalı platform olayı içeriği, durum 400 Metni

```
[{"message": "No such column 'Name__c' on subject of type MQPlatformEvent2__e", "errorCode": "INVALID_FIELD"}]
```

Geçersiz platform olay adı, durum 404 metni

```
{"errorCode": "NOT_FOUND", "message": "The requested resource does not exist"}
```

Hatalı JSON, durum 400 metni

```
{"errorCode": "NOT_FOUND", "message": "The requested resource does not exist"}
```

İleti JSON değil, durum 400 metni

```
[{"message": "Unexpected character ('h' (code 104)): expected a valid value (number, String, array, object, 'true', 'false' or 'null') at [line:1, column:2]", "errorCode": "JSON_PARSER_ERROR"}]
```

Bir metin iletisi değil ( Salesforce adresine gönderilmez)

```
Error: Publication on topic ' /sf/mqtosfb/event/MQPlatformEvent1' does not contain a text formatted message
```

Linux

Deprecated

## **IBM MQ Bridge to Salesforce ' in çalıştırılması**

Salesforce ve IBM MQ' e bağlanmak için IBM MQ Bridge to Salesforce komutunu çalıştırın. Bağlantı kurulduğunda, köprü Salesforce konularına abonelikler oluşturabilir ve iletileri IBM MQ konusuna yeniden yayınlatabilir. Köprü, Salesforce platformu olayları için olay iletileri de oluşturabilir.

### **Başlamadan önce**

görevinde yapılandırma adımlarını tamamladınız:

- [“IBM MQ Bridge to Salesforce ' nin yapılandırılması” sayfa 838](#)
- [“Salesforce platformu olayları için olay iletileri oluşturma” sayfa 845](#)

## Bu görev hakkında

IBM MQ Bridge to Salesforce'yi çalıştırmak için önceki görevde oluşturduğunuz yapılandırma dosyasını kullanın. Yapılanış kütüğünüze gerekli tüm değişiklikleri eklediyseniz, bunları komut satırına eklemeyi unutmayın.

## Yordam

1. Define the push topics or platform events in Salesforce that you want to subscribe to or the platform event that you want to create event messages for.
2. Salesforce 'e ve kuyruk yöneticinize bağlanmak için IBM MQ Bridge to Salesforce 'i başlatın. Salesforce olaylarına abone olmak için köprüyü çalıştırıyorsanız, 1. adımda tanımladığınız push konusunun ya da platform olayının adını ekleyin.

```
runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName
```

Köprü bağlandığında aşağıdaki iletiler döndürülür:

- Köprüyü Salesforce gönderme konusuna ve platform olaylarına abone olmak için kullanıyorsanız:

```
Successful connection to queue manager QM1
Warning: Subscribing to MQ-created platform events is not enabled.
Successful login to Salesforce at https://eu11.salesforce.com
Ready to process events.
```

- Köprüyü kullanarak Salesforce platformu olaylarına ilişkin olay iletileri yaratıyorsanız:

```
Successful connection to queue manager QM1
Successful login to Salesforce at https://eu11.salesforce.com
Successful subscription to '/sf/mqtsfb/event/+' for MQ-created platform events
Ready to process events.
```

3. İsteğe bağlı: Köprüyü çalıştırdıktan sonra döndürülen iletiler bir bağlantının başarılı olmadığını gösteriyorsa, kuyruk yöneticinizle ve Salesforce ile olan bağlantıda sorun giderin.

- a) Komutu hata ayıklama kipinde 1hata ayıklama seçeneğiyle çalıştırın.

```
runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 1
```

Köprü, bağlantı ayarlarında adım adım ilerler ve iletileri terse kipinde gösterir.

- b) Komutu hata ayıklama kipinde 2hata ayıklama seçeneğiyle çalıştırın.

```
runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 2
```

Köprü, bağlantı ayarlarında adım adım ilerler ve işleme iletilerini ayrıntılı kipte gösterir. Tam çıkış günlük dosyanıza yazılır.

4. Veritabanındaki kayıtları değiştirmek için Salesforce arabirimini kullanarak olaylar oluşturun.
5. Konuları iletmek üzere yapılan değişiklikleri görmek için IBM MQ Console adresine gidin. Önceki görevde yapılandığınız gereçte görüntülenir.

## Sonraki adım

Örneğin, IBM MQ izlemesini etkinleştirmek için JVM özelliklerini iletmek için `MQSFB_EXTRA_JAVA_OPTIONS` değişkenini kullanın. Daha fazla bilgi için bkz. [IBM MQ Bridge to Salesforceizleme](#).

### İlgili görevler

[IBM MQ Bridge to Salesforce ürününün izlenmesi](#)

### İlgili başvurular

[runmqsfb \( IBM MQ Bridge to Salesforcekomutunu çalıştırın\)](#)



## kullanılmak üzere yapılandırılması

Bir IBM MQ Advanced ya da IBM MQ Advanced for z/OS Value Unit Edition kuyruk yöneticisini ve IBM Blockchain'yi güvenli bir şekilde bağlamak için IBM MQ Bridge to blockchain ' ı ayarlayın ve çalıştırın. IBM MQ Advanced ya da IBM MQ Advanced for z/OS VUE kuyruk yöneticinizle bağlantı kuran bir ileti sistemi uygulamasını kullanarak blockchain teknolojinizdeki bir kaynağa zamanuysuz olarak bağlanmak, kaynağın durumunu aramak ve güncellemek için köprüyü kullanın.

### Başlamadan önce

#### Notlar:

- **Deprecated** IBM MQ Bridge to blockchain , 22 Kasım 2022 'deki tüm yayınlarda kullanımdan kaldırılmıştır (bkz. ABD Duyuru mektubu 222-341). Blockchain bağlantılığı, IBM App Connect ile ya da IBM Cloud Pak for Integration ile sağlanan App Connect yetenekleri aracılığıyla elde edilebilir.
- **V 9.3.2** **Removed** Continuous Delivery için IBM MQ Bridge to blockchain , IBM MQ 9.3.2 adresindeki üründen kaldırılır.
- **LTS** IBM , yeteneği Long Term Support yaklaşan düzeltme paketlerindeki yayınlardan kaldırmayı amaçlıyor. Bu değişiklikten etkilenen uygulamalarınız varsa, lütfen IBM Destek ile iletişim kurun.



**Uyarı:** Hyperledger Composer üzerinde oluşturulan IBM MQ Bridge to blockchain artık desteklenmiyor.

Hyperledger Fabric üzerinde oluşturulan IBM MQ Bridge to blockchain ögesini kullanmak için IBM MQ 9.1.4 ya da daha sonraki bir sürümü çalıştırmanız gerekir.

- IBM MQ Bridge to blockchain yalnızca aşağıdaki kuyruk yöneticilerine bağlanmak için kullanılabilir:
  - **Linux** IBM MQ Advanced, ya da
  - **z/OS** IBM MQ Advanced for z/OS VUE
- Kuyruk yöneticisi, köprüyle aynı komut düzeyinde ya da üstünde olmalıdır; örneğin, IBM MQ 9.3.0.
- IBM MQ Bridge to blockchain , Hyperledger Fabric 1.4 mimarisini temel alan blockchain ağıyla birlikte kullanılmak üzere desteklenir.

### Bu görev hakkında

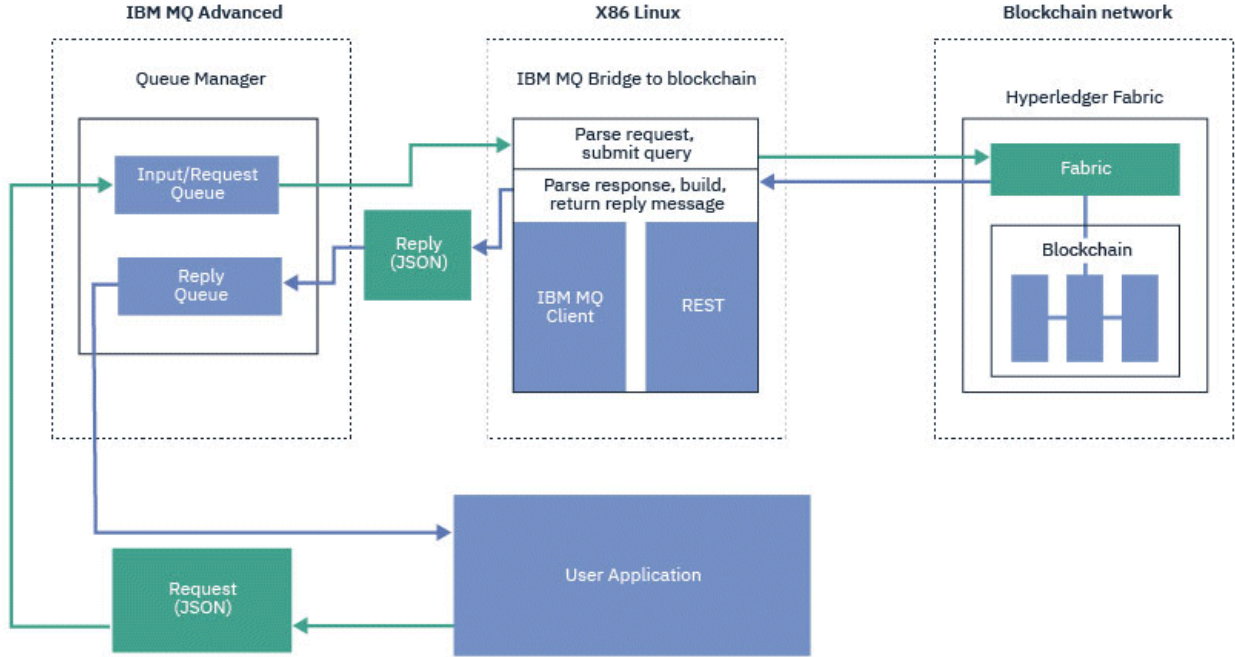
Blockchain , bir ağdaki eşler arasında kararlaştırılan işlemleri temsil eden bir blok zincirinden oluşan paylaşılan, dağıtılmış, dijital bir defterdir. Zincirdeki her blok bir önceki bloğa bağlanır ve böylece ilk işleme geri döner.

IBM Blockchain , Hyperledger Fabric temel alınarak oluşturulmuştur ve bununla yerel olarak Docker ile ya da IBM Cloud içinde bir kapsayıcı kümesinde geliştirilebilirsiniz. Yüksek güvenlik, gizlilik ve performans düzeylerine sahip bir iş ağı oluşturmak ve yönetmek için üretimde IBM Blockchain ağını etkinleştirebilir ve kullanabilirsiniz. Daha fazla bilgi için bkz. [IBM Blockchain Platform](#).

Hyperledger Fabric , ilk kod katılımcısı olarak IBM dahil olmak üzere Hyperledger Project üyeleri tarafından işbirliği içinde geliştirilen bir açık kaynak, kurumsal blockchain çerçevesidir. Hyperledger Project, veya Hyperledger, sektörler arası blockchain teknolojilerini geliştirmek için Linux Foundation açık kaynaklı, küresel ve işbirliğine dayalı bir girişimdir. Daha fazla bilgi için bkz. [IBM Blockchain](#), [Hyperledger Projeleri](#) ve [Hyperledger Fabric](#).

IBM MQ Advanced ya da IBM MQ Advanced for z/OS VUE ve IBM Blockchain kullanıyorsanız, basit sorgular, güncellemeler ve blockchain ağından yanıt almak için IBM MQ Bridge to blockchain kullanabilirsiniz. Bu şekilde, şirket içi IBM yazılımınızı bir bulut blockchain hizmetiyle bütünleştirebilirsiniz.

Şekil 1' de köprü işletim sürecine ilişkin kısa bir genel bakış görebilirsiniz. Bir kullanıcı uygulaması, IBM MQ Advanced'da IBM MQ Advanced for z/OS VUE kuyruk yöneticisindeki giriş/istek kuyruğuna JSON biçimli bir ileti koyar. Köprü kuyruk yöneticisine bağlanır, giriş/istek kuyruğundan iletiyi alır, JSON ' un doğru biçimlendirilip biçimlendirilmediğini denetler, ardından sorguyu ya da blok zincirinde bir güncelleme yayınlar. Blockchain tarafından döndürülen veriler, köprü tarafından ayrıştırılır ve özgün IBM MQ istek iletilisinde tanımlandığı şekilde yanıt kuyruğuna yerleştirilir. Kullanıcı uygulaması kuyruk yöneticisine bağlanabilir, yanıt iletilisini yanıt kuyruğundan alabilir ve bilgileri kullanabilir.



Şekil 98. IBM MQ Bridge to blockchain

IBM MQ Bridge to blockchain ' i bir blockchain ağına katılımcı ya da eş olarak bağlanmak için yapılandırabilirsiniz. Köprü çalışırken, bir ileti sistemi uygulaması köprüden kaynağın durumunu sorgulayan ya da güncelleyen ve sonuçları yanıt olarak ileti sistemi uygulamasına döndüren zincirleme yordamları yönlendirmesini ister.

## Yordam

1. Bir kuyruk yöneticisi yaratın ve başlatın ya da IBM MQ Bridge to blockchain ile kullanmak istediğiniz var olan bir kuyruk yöneticisini başlatın.

Kuyruk yöneticisi yarat:

```
crtmqm adv_qmgr_name
```

Kuyruk yöneticisini başlat:

```
strmqm adv_qmgr_name
```

2. **DefineQ.mqsc** komut dosyasında tanımlanan köprü için kuyruklar oluşturun.

Örnek köprü kuyruğu tanımlamaları, aşağıdakiler için kullanılan varsayılan adlandırılmış kuyruklar için sağlar:

- Kullanıcı kimlik bilgileri, örneğin SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE
- Köprüye ileti girişi, örneğin APPL1.BLOCKCHAIN.INPUT.QUEUE
- Blockchain 'den gelen yanıtlar, örneğin APPL1.BLOCKCHAIN.REPLY.QUEUE

/opt/mqm/mqbc/samp dizininden aşağıdaki komutu verin:

```
runmqsc adv_qmgr_name < ./DefineQ.mqsc
```

Farklı uygulamalar aynı giriş kuyruğunu kullanabilir, ancak uygulamalarınızın her biri için bir tane olacak şekilde birden çok yanıt kuyruğu belirleyebilirsiniz. Tanımlı yanıt kuyruklarını kullanmanız gerekmez. Yanıtlar için dinamik kuyruklar kullanmak istiyorsanız, bunların güvenlik yapılandırmasını göz önünde bulundurmanız gerekir.

## Sonuçlar

Köprünün IBM MQ ' den ve blockchain ağıңызdan gelen iletileri işlemek için gerektirdiği kuyrukları oluşturduz.

## Sonraki adım

IBM MQ Bridge to blockchain için bir yapılandırma dosyası oluşturmak üzere IBM MQ Advanced veya IBM MQ Advanced for z/OS VUE, kuyruk yöneticisi bilgilerinizi ve blockchain ağıңызdaki kimlik bilgilerinizi kullanın.

## Deprecated IBM MQ Bridge to blockchain için yapılandırma dosyası oluşturma

IBM MQ Bridge to blockchain ' in IBM MQ ve IBM Blockchain ağlarınıza bağlanması için yapılandırma dosyasını oluşturmak üzere kuyruk yöneticinizi ve blockchain ağ parametrelerinizi girin.

## Başlamadan önce

- Blockchain ağını oluşturup yapılandırdınız.
- Blockchain ağıңызdaki kimlik bilgileri dosyasına sahipsiniz.
- IBM MQ Bridge to blockchain ürününü x86 Linux ortamınıza kurdunuz.

Daha fazla bilgi için bkz. [Installing IBM MQ server on Linux and IBM MQ rpm components for Linux systems](#).

- IBM MQ Advanced kuyruk yöneticinizi başlattınız.

## Bu görev hakkında

Bu görev, IBM MQ Bridge to blockchain yapılandırma dosyasını oluşturmak ve IBM Blockchain ve IBM MQ ağlarınıza başarıyla bağlanmak için gereken en düşük düzeyde ayarları gerçekleştirmenizi sağlar.

Hyperledger Fabric 1.4 architecture tabanlı blockchain ağlarına bağlanmak için köprüyü kullanabilirsiniz. Köprüyü kullanmak için blockchain ağıңызdan yapılandırma bilgilerine gereksinim duyarsınız. Bu görevdeki her adımda, iki farklı yapılandırılmış blockchain ağına dayalı örnek yapılandırma ayrıntılarını bulabilirsiniz:

- Docker içinde çalışan Hyperledger Fabric ağı. Daha fazla bilgi için bkz. [Hyperledger Fabric ile çalışmaya başlama](#), [İlk uygulamanızı yazma](#) “[Örnek Hyperledger Fabric ağ kimlik bilgileri dosyası](#)” sayfa 856.
- IBM Cloud içinde bir Kubernetes kümesinde çalışan Hyperledger Fabric ağı. Daha fazla bilgi için bkz. [IBM Blockchain Platform](#).

## Yordam

1. Bir yapılandırma dosyası oluşturmak için köprüyü çalıştırın.

Blockchain ağ kimlik bilgileri dosyanızdaki ve IBM MQ Advanced kuyruk yöneticinizdeki parametrelere gereksinim duyarsınız.

```
runmqbcb -o config_file_name.cfg
```

Aşağıdaki örnekte gösterildiği gibi, var olan değerler köşeli ayraç içinde gösterilir. Var olan değerleri kabul etmek için Enter tuşuna basın, değerleri temizlemek için Space ardından Enter tuşuna basın

ve yeni değerler eklemek için köşeli ayraç içine Enter tuşuna basın. Değer listelerini (eşler gibi) virgüllerle ya da her bir değeri yeni bir satıra girerek ayırabilirsiniz. Boş bir satır, listeyi sonlandırır.

**Not:** Var olan değerleri düzenleyemezsiniz. Bunları saklayabilir, değiştirebilir ya da temizleyebilirsiniz.

2. IBM MQ Advanced kuyruk yöneticinizle bağlantı için değer girin.

Bağlantı için gereken alt sınır değerleri, kuyruk yöneticisi adı, tanımladığınız köprü girişi ve kimlik kuyrukları adlarıdır. Uzak kuyruk yöneticilerine yönelik bağlantılar için **MQ Channel** ve **MQ Conname** (kuyruk yöneticisinin çalıştığı anasistem adresi ve kapı) gereklidir. “4” sayfa 856. adımda IBM MQ 'a bağlanmak için TLS kullanmak üzere JNDI ya da CCDT' yi kullanmalı ve **MQ CCDT URL** ya da **JNDI implementation class** ve **JNDI provider URL** ' yi buna göre belirtmelisiniz.

3. 13. Ağınıza ilişkin Hyperledger Fabric sunucusu kimlik bilgilerini girin.

Beklemeniz gereken özelliklere ilişkin örnekler aşağıdaki kodda gösterilmiştir:

```
Fabric Server
-----
Network configuration file      : []connection-tls.json
Wallet                         : []
User Name                     : []User1
Certificate                   : []<path_to_user_certificate>
Private Key                   : []<path_to_private_key>/private_key.pem
Organisation                  : []Org1MSP
```

4. TLS bağlantıları için sertifika deposu değerlerini girin.

Yoksa bu alanı boş bırakın.

```
Certificate stores for MQ TLS connections
-----
Personal keystore             : []
Keystore password            : []
Trusted store for signer certs : []
Trusted store password       : []
```

5. Köprü günlüklerinin yazılacağı günlük dosyasının yolunu girin.

```
Behavior of bridge program
-----
Runtime logfile for copy of stdout/stderr : []bridgelog.log
Number of logfiles                       : [3]
Maximum size of each logfile (bytes)     : [2097152]
```



**Uyarı:** Daha önce, Akranlar, Siparişler ve Sertifika Yetkilisi ile ilgili ayrıntılar bu köprü kurulumunda saklanıyordu. Ancak, bu bilgiler artık kuruluşun Hyperledger Fabric sunucusu bölümünde bulunan *Ağ Yapılandırma Dosyası* 'nda saklanır.

## Sonuçlar

IBM MQ Bridge to blockchain ' in IBM Blockchain ağınıza ve IBM MQ Advanced kuyruk yöneticinize bağlanmak için kullandığı yapılandırma dosyasını yarattınız.



## Sonraki adım

“IBM MQ Bridge to blockchain ' in çalıştırılması” sayfa 860adımları boyunca çalışın.

## Deprecated Örnek Hyperledger Fabric ağ kimlik bilgileri dosyası

Docker içinde çalışan yerel olarak somutlaştırılan Hyperledger Fabric blockchain ağınızdaki .yaml dosyasının içeriği, IBM MQ Bridge to blockchain' inizi yapılandırmak için bu dosyayı kullanabilirsiniz.

IBM MQ Bridge to blockchain , aşağıdakilere bağlanmak için kullanılabilir:

-  IBM MQ Advanced, ya da
-  IBM MQ Advanced for z/OS VUE

yalnızca kuyruk yöneticileri.

Hyperledger Fabric ile çalışmaya başlama eğitmenleriyle çalıştıktan, anlaşıldı Sahne arkasında neler oluyor?ve Hyperledger Fabric örnekleri' dan birini kullanarak ağınızı başlattıktan sonra, /blockchain/fabric-samples/basic-network klasörünüzde aşağıdaki yapılandırma dosyası bulunmalıdır.

Blockchain ağınıza bağlanmak istiyorsanız, "IBM MQ Bridge to blockchain için yapılandırma dosyası oluşturma" sayfa 855oldüğünüzda bu dosyadaki yapılandırma ayrıntılarını kullanmanız gerekir.

```
{
  "name": "basic-network",
  "version": "1.0.0",
  "client": {
    "organization": "Org1",
    "connection": {
      "timeout": {
        "peer": {
          "endorser": "300"
        },
        "orderer": "300"
      }
    }
  },
  "channels": {
    "mychannel": {
      "orderers": [
        "orderer.example.com"
      ],
      "peers": {
        "peer0.org1.example.com": {
          "endorsingPeer": true,
          "chaincodeQuery": true,
          "ledgerQuery": true,
          "eventSource": true
        },
        "peer0.org2.example.com": {
          "endorsingPeer": true,
          "chaincodeQuery": false,
          "ledgerQuery": true,
          "eventSource": false
        }
      }
    }
  },
  "organizations": {
    "Org1": {
      "mspid": "Org1MSP",
      "peers": [
        "peer0.org1.example.com"
      ],
      "certificateAuthorities": [
        "ca-org1"
      ],
      "adminPrivateKeyPEM": {
        "path": "$<path_to_private_key>/admin_private_key"
      },
      "signedCertPEM": {
        "path": "<path_to_org_signed_cert>/Admin@org1.example.com-cert.pem"
      }
    },
    "Org2": {
      "mspid": "Org2MSP",
      "peers": [
        "peer0.org2.example.com"
      ],
      "certificateAuthorities": [
        "ca-org2"
      ]
    }
  },
  "orderers": {
    "orderer.example.com": {
      "url": "grpc://localhost:7050",
      "mspid": "OrdererMSP",
      "grpcOptions": {
        "ssl-target-name-override": "orderer.example.com",
        "hostnameOverride": "orderer.example.com"
      }
    }
  }
}
```

```

"tlsCACerts": {
  "path": "<path_to_orderer_cert>/ca.crt"
},
"adminPrivateKeyPEM": {
  "path": "<path_to_orderers_private_key>/<private_key>"
},
"signedCertPEM": {
  "path": "<path_to_orderer_signed_cert>/Admin@example.com-cert.pem"
}
},
"peers": {
"peer0.org1.example.com": {
  "url": "grpc://localhost:7051",
  "grpcOptions": {
    "ssl-target-name-override": "peer0.org1.example.com",
    "hostnameOverride": "peer0.org1.example.com",
    "request-timeout": 120001
  },
  "tlsCACerts": {
    "path": "<path_to_peer_cert>/ca.crt"
  }
},
"peer0.org2.example.com": {
  "url": "grpc://localhost:9051",
  "grpcOptions": {
    "ssl-target-name-override": "peer0.org2.example.com",
    "hostnameOverride": "peer0.org2.example.com",
    "request-timeout": 120001
  },
  "tlsCACerts": {
    "path": "<path_to_peer_cert>/ca.crt"
  }
}
},
"certificateAuthorities": {
"ca-org1": {
  "url": "https://localhost:7054",
  "grpcOptions": {
    "verify": true
  },
  "tlsCACerts": {
    "path": "<path_to_ca_cert>/ca.org1.example.com-cert.pem"
  },
  "registrar": [
    {
      "enrollId": "admin",
      "enrollSecret": "adminpw"
    }
  ]
},
"ca-org2": {
  "url": "https://localhost:8054",
  "grpcOptions": {
    "verify": true
  },
  "tlsCACerts": {
    "path": "<path_to_ca_cert>/ca.org2.example.com-cert.pem"
  },
  "registrar": [
    {
      "enrollId": "admin",
      "enrollSecret": "adminpw"
    }
  ]
}
}
}
}
}

```

## Linux Deprecated IBM MQ 9.2.0 ' den IBM MQ Bridge to blockchain için ileti biçimleri

IBM MQ Bridge to blockchaintarafından gönderilen ve alınan iletilerin biçimlendirilmesine ilişkin bilgiler.

Bir uygulama, IBM MQ Bridge to blockchain ' in Hyperledger Fabric sunucusunu blockchain üzerinde tutulan bilgilere göre hareket etmeye yönlendirmesini ister. Uygulama bunu, köprü istek kuyruğuna bir istek ileti yerleştirerek yapar. İsteğin sonuçları köprü tarafından bir yanıt iletiğine biçimlendirilir. Köprü,

İstek iletişimine ilişkin MQMD ' deki **ReplyToQ** ve **ReplyToQMGr** alanlarında bulunan bilgileri yanıt iletişinin hedefi olarak kullanır.

İstek ve yanıt iletileri, JSON biçimindeki metin (MQSTR) iletileridir ve dört öge içerir.

## İstek İletisi Biçimi

İstek iletileri aşağıdaki öznitelikleri içerir:

### işlem

- Dizgi-büyük ve küçük harfe duyarsız
- Güncellemeler için submit ya da sorgular için evaluate

### ağ

- Dizgi-bazen Hyperledger Fabric içinde channel olarak bilinir

### Sözleşme

- Dizgi-çağrılacak akıllı sözleşme ya da zincirleme kod paketi

### bağımsız arg

- Dizi-genellikle dizgilerden oluşur, ancak bazı öğeler içiçe JSON nesnelere olabilir.
- Yöntem adı da içinde olmak üzere, **contract** için gerçek bağımsız değişkenler.

Örneğin:

```
{
  "operation" : "Evaluate",
  "network"   : "mychannel",
  "contract"  : "marbles0",
  "args"     : [ "readMarble" , "marble1" ]
}
```

**Not:** Bu öğelerin var olduğundan ve iletinin geçerli JSON olduğundan emin olmak dışında, köprü tarafından içerik doğrulaması gerçekleştirilmez. Köprü, isteği işlemek ya da hataları döndürmek için Hyperledger Fabric ' e dayanır.

## Yanıt İletisi Biçimi

Yanıt iletilerinin ilinti tanıtıcısı, gelen iletinin ileti tanıtıcısına ayarlı. Kullanıcı tanımlı özellikler istek iletişiminden yanıt iletişimine kopyalanır. Yanıttaki kullanıcı kimliği, kaynak kullanıcının kullanıcı kimliğine ayarlanır.

**statusCode** , bir HTTP durum kodudur. Hata IBM MQ ya da köprüden ise, uygun bir **statusCode** kullanılır.

**statusType** , *SUCCESS* ya da *FAILURE* olan bir dizedir.

Başarılı istekler için, yanıt iletişimindeki **"data"** ögesi, çağrılan Hyperledger Composer REST API ' nin yanıtını içerir.

Başarılı işleme örneği:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "data": [
    {
      "$class": "org.example.trading",
      "firstName": "John",
      "lastName": "Doe",
      "tradeId": "Trader1"
    },
    {
      "$class": "org.example.trading",
      "firstName": "Jane",
      "lastName": "Doe",
      "tradeId": "Trader2"
    }
  ]
}
```

```
]
}
```

Tüm hata yanıtları, köprünün kendisi tarafından oluşturulup oluşturulmadıklarından bağımsız olarak, çağrılardan Hyperledger Composer REST sunucusuna, blockchain 'e ya da zincirleme kod çağrısından aynı alanlara sahiptir. Örneğin:

- Hatalı JSON giriş iletisi

```
{
  "statusCode": 400,
  "statusType": "FAILURE",
  "message": "[AMQBC021E] Error: Cannot parse input message or there are
  missing fields in the message. Missing fields appear to be: "method"."
}
```

- Hyperledger Composer REST sunucusu tarafından işlenmeyen istek

```
{
  "statusCode": 500,
  "statusType": "FAILURE",
  "message": "Error trying to invoke business network. Error: No valid responses
  from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
  error (status: 500, message: Error: Failed to add object with ID 'Trader1'
  as the object already exists)"
}
```

Uygulamalar, isteğin **statusType** dizgisine bakarak mı, yoksa veri alanının varlığından mı başarılı ya da başarısız olduğunu söyleyebilir. Giriş iletisi işlenirken bir hata oluştuğunda ve köprü bunu blockchain 'e göndermediğinde, köprüden döndürülen değer genellikle **MQRC\_FORMAT\_ERROR** olan bir MQRC değeridir.

## Deprecated IBM MQ Bridge to blockchain ' in çalıştırılması

IBM Blockchain ve IBM MQ' e bağlanmak için IBM MQ Bridge to blockchain komutunu çalıştırın. Bağlantı kurulduğunda köprü, istek iletilerini işlemeye, bunları Hyperledger Composer blockchain ağınıza göndermeye ve yanıtları almaya ve işlemeye hazırdır.

### Bu görev hakkında

IBM MQ Bridge to blockchain' yi çalıştırmak için önceki görevde oluşturduğunuz yapılandırma dosyasını kullanın.

### Yordam

1. Köprüyle kullanmak istediğiniz IBM MQ Advanced kuyruk yöneticisini başlatın.
2. Hyperledger Composer REST sunucunuza ve IBM MQ Advanced kuyruk yöneticinize bağlanmak için IBM MQ Bridge to blockchain ' i başlatın.

Köprü komutunu çalıştırın.

```
runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
```

Köprü bağlandığında, aşağıdakine benzer bir çıkış döndürülür:

```
2018-05-17 14:28:16.866 BST IBM MQ Bridge to Blockchain
5724-H72 (C) Copyright IBM Corp. 2017, 2024.
```

```
2018-05-17 14:28:19.331 BST Ready to process input messages.
```

3. İsteğe bağlı: Köprüyü çalıştırdıktan sonra döndürülen ileteler bir bağlantının başarılı olmadığını gösteriyorsa, IBM MQ Advanced kuyruk yöneticinizle ve blockchain ağınızla olan bağlantılarda sorun giderin.



a) Komutu hata ayıklama kipinde 1hata ayıklama seçeneğiyle çalıştırın.

```
runmqcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log  
-d 1
```

Köprü, bağlantı ayarlarında adım adım ilerler ve iletileri terse kipinde gösterir.

b) Komutu hata ayıklama kipinde 2hata ayıklama seçeneğiyle çalıştırın.

```
runmqcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log  
-d 2
```

Köprü, bağlantı ayarlarında adım adım ilerler ve işleme iletilerini ayrıntılı kipte gösterir. Tam çıkış günlük dosyanıza yazılır.

## Sonuçlar

IBM MQ Bridge to blockchain ürününü başlatmış ve Hyperledger Composer REST sunucunuzu kullanarak kuyruk yöneticisine ve blockchain ağınıza bağlandınız.

## Sonraki adım

- Blockchain ağınıza bir sorgu ya da güncelleme iletişi biçimlendirmek ve göndermek için “z/OS üzerinde IBM MQ Bridge to blockchain istemcisi örneğini çalıştırma” sayfa 863 içindeki adımları izleyin.
- JVM özelliklerini geçirmek için `MQBCB_EXTRA_JAVA_OPTIONS` değişkenini kullanın; örneğin, IBM MQ izlemesini etkinleştirmek için. Daha fazla bilgi için bkz. [IBM MQ Bridge to blockchainizleme](#).

## z/OS üzerinde IBM MQ Bridge to blockchain before IBM MQ 9.2.0 için ileti biçimleri

IBM MQ Bridge to blockchaintarafından gönderilen ve alınan iletilerin biçimlendirilmesine ilişkin bilgiler.



**Uyarı:** İleti biçimleri için var olan biçim eski. IBM MQ 9.2.0' de bir Hyperledger Fabric ağınıza varsa, “IBM MQ 9.2.0 ' den IBM MQ Bridge to blockchain için ileti biçimleri” sayfa 858içinde açıklanan iletilerin biçimini kullanın.

An application requests that the IBM MQ Bridge to blockchain drives the Hyperledger Composer defined REST API to act on information that is held on the blockchain. Uygulama bunu, köprü istek kuyruğuna bir istek iletişi yerleştirerek yapar. REST isteğinin sonuçları, köprü tarafından bir yanıt iletişine biçimlendirilir. Köprü, istek iletişine ilişkin MQMD ' deki **ReplyToQ** ve **ReplyToQMGr** alanlarında bulunan bilgileri yanıt iletişinin hedefi olarak kullanır.

İstek ve yanıt iletileri, JSON biçimindeki metin (MQSTR) iletileridir.

## İstek İletisi Biçimi

İstek iletileri üç öznitelik içerir:

### yöntem

POST, DELETE ya da GET gibi Hyperledger Composer REST API ' yı çağırmak için kullanılan REST komutu

### yol

Hyperledger Composer REST API yolu. Bu, URLtemel sunucusuna eklenir. Yol "api/" ile başlamalıdır.

### gövde

Yönteme özgü içerik. Bu genellikle bir JSON yapısıdır.

Aşağıdaki örnekte, yeni bir Trader nesnesi yaratmak için POST (hedef yol) `api/Trader`yöntemi kullanılmıştır. Gövde, kullanıcının Hyperledger Composer modeli tarafından tanımlandığı şekilde Traders

sınıfını belirtir ve blockchain ağı içinde yeni bir Trader nesnesi oluşturmak için gereken ek değerleri de belirtir.

```
{ "method": "POST",
  "path": "api/Trader",
  "body": {
    "$class": "org.example.trading",
    "tradeId": "Trader2",
    "firstName": "Jane",
    "lastName": "Doe"
  }
}
```

## Yanıt İletisi Biçimi

Yanıt iletilerinin ilinti tanıtıcısı, gelen iletinin ileti tanıtıcısına ayarlı. Kullanıcı tanımlı özellikler istek iletilisinden yanıt iletilisine kopyalanır. Yanıttaki kullanıcı kimliği, kaynak kullanıcının kullanıcı kimliğine ayarlanır.

**statusCode** , bir HTTP durum kodudur. Hata IBM MQ ya da köprüden ise, uygun bir **statusCode** kullanılır.

**statusType** , *SUCCESS* ya da *FAILURE* olan bir dizedir.

Başarılı istekler için, yanıt iletilisindeki **"data"** ögesi, çağrılan Hyperledger Composer REST API ' nin yanıtını içerir.

Başarılı işleme örneği:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "data": [
    {
      "$className": "org.example.trading",
      "firstName": "John",
      "lastName": "Doe",
      "tradeId": "Trader1"
    },
    {
      "$className": "org.example.trading",
      "firstName": "Jane",
      "lastName": "Doe",
      "tradeId": "Trader2"
    }
  ]
}
```

Tüm hata yanıtları, köprü'nün kendisi tarafından oluşturulup oluşturulmadıklarından bağımsız olarak, çağrılardan Hyperledger Composer REST sunucusuna, blockchain 'e ya da zincirleme kod çağrısından aynı alanlara sahiptir. Örneğin:

- Hatalı JSON giriş iletilisi

```
{
  "statusCode": 400,
  "statusType": "FAILURE",
  "message": "[AMQBC021E] Error: Cannot parse input message or there are missing fields in the message. Missing fields appear to be: "method"."
}
```

- Hyperledger Composer REST sunucusu tarafından işlenmeyen istek

```
{
  "statusCode": 500,
  "statusType": "FAILURE",
  "message": "Error trying to invoke business network. Error: No valid responses from any peers.\nResponse from attempted peer comms was an error: Error: chaincode error (status: 500, message: Error: Failed to add object with ID 'Trader1'"
}
```

```
    as the object already exists)"  
}
```

Uygulamalar, isteğin **statusType** dizgisine bakarak mı, yoksa veri alanının varlığından mı başarılı ya da başarısız olduğunu söyleyebilir. Giriş iletisi işlenirken bir hata oluştuğunda ve köprü bunu blockchain 'e göndermediğinde, köprüden döndürülen değer genellikle **MQRC\_FORMAT\_ERROR** olan bir MQRC değeridir.

## z/OS üzerinde IBM MQ Bridge to blockchain istemcisi örneğini çalıştırma

IBM MQ Bridge to blockchain ile birlikte sağlanan JMS istemci örneğini, blockchain köprüsünün denetlediğini belirten bir iletiyi giriş kuyruğuna yerleştirmek ve alınan yanıtı görmek için kullanabilirsiniz. Bu örnek, IBM MQ Bridge to blockchain tümleştirmesinin Hyperledger Composer Trader ağ örneğiyle kullanılmasına dayalıdır.

### Başlamadan önce


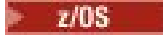
Daha fazla bilgi için bkz. [/trade\\_network](#)

IBM MQ Bridge to blockchain ürününüz çalışıyor ve IBM MQ Advanced ya da IBM MQ Advanced for z/OS VUE kuyruk yöneticinize ve blockchain ağınıza bağlı.

### Bu görev hakkında

JMS örnek uygulamasını (ComposerBCBSamp.java) IBM MQ Bridge to blockchain' in samp dizininde bulun.

Örneğin: <MQ\_INSTALL\_ROOT>/mqbc/samp/ComposerBCBSamp.java, burada <MQ\_INSTALL\_ROOT> :

-  IBM MQ ' in kurulu olduğu dizin
-  IBM MQ ürününün z/OS UNIX bileşenlerinin kurulu olduğu z/OS UNIX System Services dizini

### Yordam

1. İstemci örnek Java kaynak dosyasını düzenleyin.

IBM MQ ortamınızla ve blockchain ağınıza eşleşecek şekilde yapılandırmak için örnekteki yönergeleri izleyin.

Örnekteki aşağıdaki kod, köprüye gönderilecek üç JSON istek iletisini tanımlar:

- a. Öncelikle, var olan bir 'commodity' öğesini kaldırmak için
- b. İkinci olarak, yeni bir 'commodity', 'owner' ve ilişkili değerler oluşturmak için,
- c. Son olarak, önceki iki istek iletisini izleyerek 'commodity' ile ilgili yeni bilgileri görüntüler

```
private static JSONObject[] createMessageBodies() {  
    JSONObject[] msgs = new JSONObject[3]; // This method creates 3 messages  
    JSONObject m, m2;  
    String commodityName = "BC";  
  
    // Clean out the commodity in case it's already there. If  
    // it's not there, there will be an error returned from Composer.  
    m = new JSONObject();  
    m.put("method", "DELETE");  
    m.put("path", "api/Commodity/" + commodityName);  
    msgs[0] = m;  
  
    // To add the item to the table, the  
    // operation looks like this:
```

```

//
// { "method": "POST",
//   "path": "api/Commodity",
//   "body" : {
//     "$class": "org.example.trading.Commodity",
//     "tradingSymbol" : "BC",
//     "description" : "BC",
//     "mainExchange" : "HERE",
//     "owner" : "Me",
//     "quantity" : 100
//   }
// }
// You can see this structure in the API Explorer
m = new JSONObject();
m.put("method", "POST");
m.put("path", "api/Commodity");
m2 = new JSONObject();
m2.put("$class", " org.example.trading.Commodity");
m2.put("tradingSymbol", commodityName);
m2.put("description", "Blockchain Sample Description");
m2.put("mainExchange", "My Exchange");
m2.put("owner", "Me");
m2.put("quantity", 100);
m.put("body", m2);
msgs[1] = m;

// And list all items that have been created
m = new JSONObject();
m.put("method", "GET");
m.put("path", "api/Commodity");
msgs[2] = m;

return msgs;
}

```

## 2. Örneği derleyin.

Köprü dizininde gönderilen IBM MQ istemci sınıflarını ve JSON4J . jar dosyasını işaret edin.

```

javac -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar
ComposerBCClient.java

```

## 3. Derlenmiş sınıfı çalıştırın.

```

java -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar:.
ComposerBCClient

```

```

Starting Simple MQ Blockchain Bridge Client
Starting the connection.
Sent message:
{"method":"DELETE"," path ":"api/Commodity/BC"}
Response text:
{
  "statusCode": 204,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}
SUCCESS
Sent message:
{"body":
{"$class":"org.example.trading.Commodity","owner":"Me","quantity":100,"description":"Blockcha
in Sample Description","mainExchange":"My
Exchange","tradingSymbol":"BC"},"operation":"POST","url":"Commodity"}
Response text:
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": {
    "$class": "org.example.trading.Commodity",
    "description": "Blockchain Sample Description",
    "mainExchange": "My Exchange",
    "owner": "Me",
    "quantity": 100,
    "tradingSymbol": "BC"
  }
}

```

```

    }
  }
  SUCCESS
  Sent message:
  {"method":"GET","path":"/Commodity"}
  Response text:
  {
    "statusCode": 200,
    "statusType": "SUCCESS",
    "message": "OK",
    "data": [
      {
        "$class": "org.example.trading.Commodity",
        "description": "Blockchain Sample Description",
        "mainExchange": "My Exchange",
        "owner": "resource:org.example.trading.Trader#Me",
        "quantity": 100,
        "tradingSymbol": "BC"
      }
    ]
  }
}
SUCCESS

```

**message** alanı, başarıyla işlenen bir ileti için "OK" ya da başarısız bir istek durumunda, hatanın nedenine ilişkin bilgileri içerir.

İstemci yanıtı beklerken bir zamanaşımı hatası alırsa, köprünün çalışıp çalışmadığını denetleyin.

## Linux Deprecated IBM MQ Bridge to blockchain için ek yapılandırma seçenekleri

IBM MQ 9.2.0' da, IBM MQ Bridge to blockchain üzerinde izleme ve günlüğe kaydetme işleminin çalışma yönteminde bir değişiklik vardır.

### IBM MQ 9.1.0 IBM MQ Bridge to blockchain ürünündeki değişiklikler

Varsayılan olarak, IBM MQ 9.1.0 köprüsünde, günlük dosyasının döndürmeye başlamasından başka herhangi bir davranış değişikliği yoktur. Ek bilgi için bkz. [“Dönen günlükler” sayfa 865](#) .

### İzleme ve hata ayıklama etkileşimi

IBM MQ 9.2.0' dan hata ayıklama işareti, IBM MQ 9.1.0 için olduğu gibi davranmaya devam eder. *-d1* köprüde hata ayıklama bilgileri verir ve *-d2* önkoşul olan bileşenler için hata ayıklama günlüğünü açar. Ancak, köprüyü başlattığınızda IBM MQ izlemesini etkinleştirdiyse, *-d2* düzey raporlaması otomatik olarak açılır.

### Dönen günlükler

IBM MQ 9.2.0' den günlük dosyasının varsayılan davranışı, her biri 2 MB boyutunda olan üç günlük dosyasına sahip olmaktır. Ek yapılandırma özelliklerini kullanarak bu değerleri geçersiz kılabilirsiniz. Günlük dosyasına ilişkin var olan yapılandırma özniteliği ya da komut satırı parametresi, bir dizin eklenmiş olarak günlüklerin temel adı olarak alınır.

Yapılandırılan günlük dosyası aşağıdaki özelliklere sahipse:

- Dosya tipi yok, dizin dosya adının sonuna eklenir.

Günlük dosyasının abcolarak ayarlanması, abc .0, abc .1vb. adlı günlüklerle sonuçlanır.

- Dosya tipi, dizin, dosya tipinden önce eklenir.

Günlük dosyasının abc .logolarak ayarlanması, abc .0 .log, abc .1 .logvb. adlı günlüklerle sonuçlanır.

### Notlar:

1. Köprüler isteğe bağlı kullanıcı izniyle çalışabileceğinden, günlükler için belirli bir dizini (örneğin, /var/mqm/qmgrs/<qm>/errors) zorlamak mümkün değildir.
2. Aynı bilgiler *stdout* ve *stderr* akışlarına yazılmaya devam eder.

3. Tek bir günlük dosyası yeniden açıldığında, temel yapılandırma bilgileri yeniden yazdırılır. Bilgiler, programın başlangıcında bir kez yazdırılmak yerine her zaman kullanılabilir olacaktır.

## **z/OS** **MQ,Adv.VUE** **Deprecated** **IBM MQ Advanced for z/OS VUE ' nin Blockchain ile kullanılmak üzere yapılandırılması**

Bir IBM MQ ürününü z/OS kuyruk yöneticisine ve IBM Blockchain'e güvenli bir şekilde bağlamak için IBM MQ Bridge to blockchain ' ı ayarlayın ve çalıştırın. IBM MQ Advanced for z/OS VUE kuyruk yöneticinizle bağlantı kuran bir ileti sistemi uygulamasını kullanarak blockchain teknolojinizdeki bir kaynağa zamanuysuz olarak bağlanmak, aramak ve kaynağın durumunu güncellemek için köprüyü kullanın.

### **Başlamadan önce**

#### **Notlar:**

- **Deprecated** IBM MQ Bridge to blockchain , 22 Kasım 2022 'deki tüm yayınlarda kullanımdan kaldırılmıştır (bkz. [ABD Duyurusu mektubu 222-341](#)). Blockchain bağlantırlığı, IBM App Connect ile ya da IBM Cloud Pak for Integrationile sağlanan App Connect yetenekleriyle elde edilebilir.
- **V 9.3.2** **Removed** Continuous Delivery için IBM MQ Bridge to blockchain , IBM MQ 9.3.2 adresindeki üründen kaldırılır.
- IBM MQ Bridge to blockchain , IBM MQ Advanced for z/OS Value Unit Edition 9.1.0 üzerindeki bir Bağlayıcı Paketi 'nin bir parçası olarak kullanılabilir. Aynı komut düzeyinde ya da üstünde çalışan IBM MQ Advanced for z/OS VUE kuyruk yöneticilerine bağlanabilirsiniz.
- IBM MQ Bridge to blockchain , Hyperledger Fabric tabanlı Hyperledger Composer tabanlı blockchain ağınıza kullanım için desteklenir.
- IBM MQ Bridge to blockchain , bir z/OS UNIX System Services ortamına kurulmalı ve IBM' den Java runtime environment sürüm 8 'i gerektirir.

### **Bu görev hakkında**

Blockchain , bir ağdaki eşler arasında kararlaştırılan işlemleri temsil eden bir blok zincirinden oluşan paylaşılan, dağıtılmış, dijital bir defterdir. Zincirdeki her blok bir önceki bloğa bağlanır ve böylece ilk işleme geri döner.

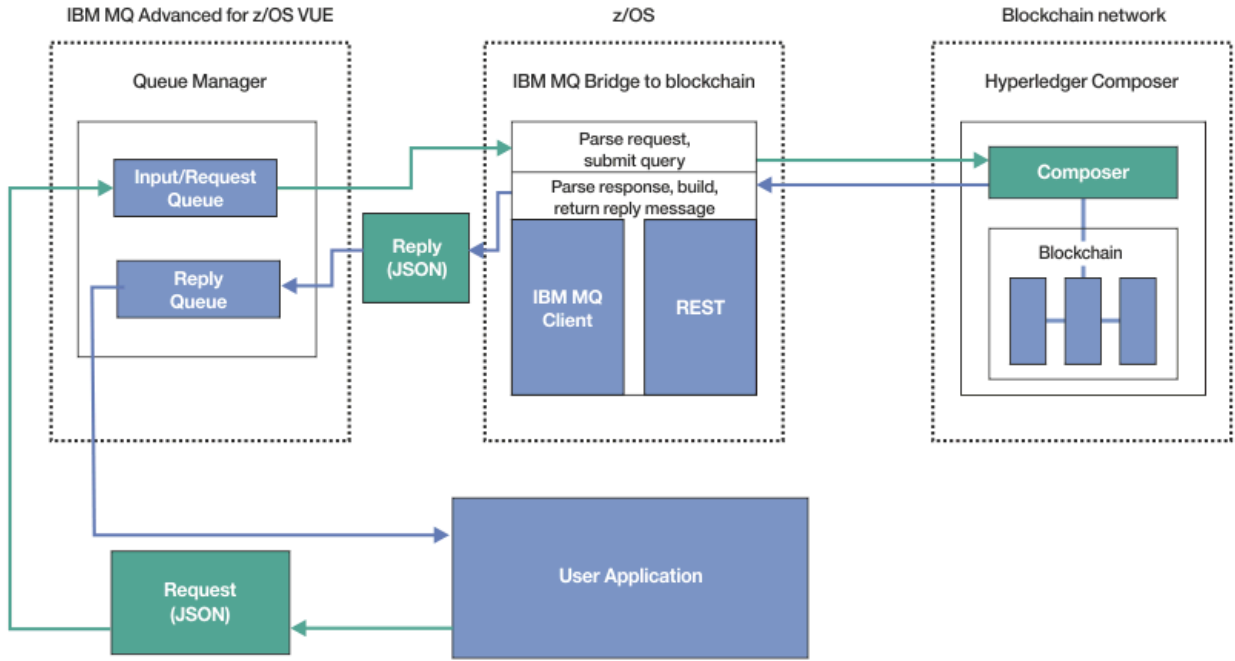
IBM Blockchain , Hyperledger Fabric ve Hyperledger Compose temel alınarak oluşturulmuştur. Bununla yerel olarak Docker ile ya da IBM Cloud içinde bir kapsayıcı kümesinde geliştirebilirsiniz. Yüksek güvenlik, gizlilik ve performans düzeylerine sahip bir iş ağı oluşturmak ve yönetmek için üretimde IBM Blockchain ağınıza etkinleştirebilir ve kullanabilirsiniz. Daha fazla bilgi için bkz. [IBM Blockchain Platform](#).

Hyperledger Fabric ve Hyperledger Composer , ilk kod katılımcısı olarak IBM dahil olmak üzere, Hyperledger Project üyeleri tarafından işbirliği içinde geliştirilen bir açık kaynak, kurumsal blockchain çerçevesidir. Hyperledger Project, veya Hyperledger, sektörler arası blockchain teknolojilerini geliştirmek için Linux Foundation açık kaynaklı, küresel ve işbirliğine dayalı bir girişimdir. Daha fazla bilgi için bkz. [IBM Blockchain](#), [Hyperledger Projeler](#), [Hyperledger Fabric](#) ve [Hyperledger Composer](#).

IBM MQ Advanced for z/OS VUE ve IBM Blockchain kullanıyorsanız, Hyperledger Composer iş modelinizi Hyperledger Composer REST arabirimi aracılığıyla yönetmek, blockchain ağınızdaki durumu güncelleme için ya da sorgulama ve blockchain ağından yanıt alma için IBM MQ Bridge to blockchain olanağını kullanabilirsiniz. Bu şekilde, şirket içi IBM yazılımınızı bir bulut blockchain hizmetiyle ya da yerel olarak yönetilen bir şirket içi çözümle bütünleştirebilirsiniz.

Köprü işletim sürecine ilişkin kısa bir genel bakış [Şekil 1](#) 'de görülebilir. Bir kullanıcı uygulaması, z/OS kuyruk yöneticisindeki giriş/istek kuyruğuna JSON biçimli bir ileti koyar. Köprü, Hyperledger Composer REST sunucusunu kullanarak kuyruk yöneticisine bağlanır, giriş/istek kuyruğundan iletiyi alır, JSON ' un doğru biçimlendirilip biçimlendirilmediğini denetler ve ardından REST isteğini blockchain 'e gönderir. Blockchain tarafından döndürülen veriler, köprü tarafından ayrıştırılır ve özgün IBM MQ istek iletisinde

tanımlandığı şekilde yanıt kuyruğuna yerleştirilir. Kullanıcı uygulaması kuyruk yöneticisine bağlanabilir, yanıt iletilerini yanıt kuyruğundan alabilir ve bilgileri kullanabilir.



Şekil 99. IBM MQ Bridge to blockchain

IBM MQ Bridge to blockchain ögesini, doğrudan temel Hyperledger Fabric katmanına değil, bir Hyperledger Composer REST sunucusuna bağlanacak şekilde yapılandırmanız gerekir. Köprü çalışırken, bir ileti sistemi uygulaması köprünün kullanıcı tanımlı iş ağı modeline dayalı olarak Hyperledger Composer REST API 'yi çalıştırmasını ister; bu model, kaynağın durumunu sorgulayabilen ya da güncelleyebilen temel zincir kod yordamlarını çalıştırır ve sonuçları, Hyperledger Composer REST sunucusunu kullanarak ileti sistemi uygulamasına yanıt olarak döndürür.

## Yordam

th1qua1.SCSQPROC (CSQ4BCBQ) içinde örnek JCL 'yi özelleştirerek ve göndererek köprüye ilişkin kuyrukları oluşturun.

Örnek köprü kuyruğu tanımlamaları, aşağıdakiler için kullanılan varsayılan adlandırılmış kuyruklar için sağlanır:

- Köprüye ileti girişi: SYSTEM.BLOCKCHAIN.INPUT.QUEUEve APPL1.BLOCKCHAIN.INPUT.QUEUE
- Blockchain 'den gelen yanıtlar: APPL1.BLOCKCHAIN.REPLY.QUEUE

Farklı uygulamalar aynı giriş kuyruğunu kullanabilir, ancak uygulamalarınızın her biri için bir tane olmak üzere birden çok yanıt kuyruğu belirleyebilirsiniz. Tanımlı yanıt kuyruklarını kullanmanız gerekmez. Yanıtlar için dinamik kuyruklar kullanmak istiyorsanız, bunların güvenlik yapılandırmasını göz önünde bulundurmanız gerekir.

## Sonuçlar

Köprünün IBM MQ 'den ve blockchain ağıңызdan gelen iletileri işlemek için gerektirdiği kuyrukları oluşturdunuz.

## Sonraki adım

IBM MQ Bridge to blockchain için bir yapılandırma dosyası oluşturmak üzere kuyruk yöneticinizle ilgili bilgileri ve blockchain ağıңызdaki kimlik bilgilerini kullanın.

## z/OS üzerinde IBM MQ Bridge to blockchain için yapılandırma dosyası oluşturma

IBM MQ Bridge to blockchain ' in IBM MQ ve IBM Blockchain ağlarınıza bağlanmasında için yapılandırma dosyasını oluşturmak üzere kuyruk yöneticinizi ve blockchain ağ parametrelerinizi girin.

### Başlamadan önce

#### Notlar:

- **Deprecated** IBM MQ Bridge to blockchain , 22 Kasım 2022 'deki tüm yayınlarda kullanımdan kaldırılmıştır (bkz. [ABD Duyurusu mektubu 222-341](#)). Blockchain bağlantılığı, IBM App Connect ile ya da IBM Cloud Pak for Integration ile sağlanan App Connect yetenekleriyle elde edilebilir.
- **V 9.3.2** **Removed** Continuous Delivery için IBM MQ Bridge to blockchain , IBM MQ 9.3.2 adresindeki üründen kaldırılır.
- **LTS** IBM , yeteneği Long Term Support yaklaşan düzeltme paketlerindeki yayınlardan kaldırmayı amaçlıyor. Bu değişiklikten etkilenen uygulamalarınız varsa, lütfen IBM Destek ile iletişim kurun.
- Hyperledger Composer blockchain ağınıza oluşturdu ve yapılandırdınız.
- IBM MQ Bridge to blockchain ürününü z/OS ortamınıza kurdu.
- IBM MQ Advanced for z/OS VUE kuyruk yöneticinizi başlattınız.

### Bu görev hakkında

Bu görev, IBM MQ Bridge to blockchain yapılandırma dosyasını oluşturmak ve IBM Blockchain ve IBM MQ ağlarınıza başarıyla bağlanmak için gereken en düşük düzeyde ayarları gerçekleştirmenizi sağlar.

Hyperledger Compositabanlı blockchain ağlarına bağlanmak için köprüyü kullanabilirsiniz. Köprüyü kullanmak için blockchain ağından yapılandırma bilgilerine gereksinim duyarsınız. Bu görevdeki her adımda, iki farklı yapılandırılmış blockchain ağına dayalı örnek yapılandırma ayrıntılarını bulabilirsiniz:

- Docker içinde çalışan Hyperledger Composer ağı. Daha fazla bilgi için bkz. [Installing Hyperledger Compose](#) ve [Generating a REST API](#).
- IBM Cloud içinde bir Kubernetes kümesinde çalışan Hyperledger Composer ağı. Daha fazla bilgi için bkz. [IBM Blockchain Platform](#).

### Yordam

1. Bir yapılandırma dosyası oluşturmak için köprüyü z/OS UNIX System Services (z/OS UNIX) ortamınızda çalıştırın.

Hyperledger Composer güvenlik bilgilerinizden ve IBM MQ Advanced for z/OS VUE kuyruk yöneticinizden gelen parametrelere gereksinim duyarsınız.

Köprü komut dosyasını, z/OS UNIX içinde IBM MQ ' in kurulu olduğu konumun mqbc/b in dizininden çalıştırın.

```
./runmqbc -o config_file_name.cfg
```

Aşağıdaki örnekte gösterildiği gibi, var olan değerler köşeli ayraç içinde gösterilir. Var olan değerleri kabul etmek için Enter tuşuna basın, değerleri temizlemek için Space ardından Enter tuşuna basın ve yeni değerler eklemek için köşeli ayraç içine Enter tuşuna basın. Değer listelerini (eşler gibi) virgüllerle ya da her bir değeri yeni bir satıra girerek ayırabilirsiniz. Boş bir satır, listeyi sonlandırır.

**Not:** Var olan değerleri düzenleyemezsiniz. Bunları saklayabilir, değiştirebilir ya da temizleyebilirsiniz.

2. IBM MQ Advanced for z/OS VUE kuyruk yöneticinizle bağlantı için değer girin.



Bağlantı için gereken alt sınır değerleri, kuyruk yöneticisi adı ve tanımladığınız köprü giriş kuyruklarının adlarıdır. Uzak IBM MQ Advanced for z/OS VUE kuyruk yöneticilerine yönelik bağlantılar için **MQ Channel** ve **MQ Conname** (kuyruk yöneticisinin çalıştığı anasistem adresi ve kapı) gereklidir.

“5” sayfa 869. adımda IBM MQ 'a bağlanmak için TLS kullanmak üzere JNDI ya da CCDT'yi kullanmalı ve **MQ CCDT URL** ya da **JNDI implementation class** ve **JNDI provider URL** 'yi buna göre belirtmelisiniz.

**Not: MQ CCDT** ya da **JNDI** değerleri, değerlerin çakıştığı yapıları kütüğüne göre önceliklidir.

```
Connection to Queue Manager
-----
Queue Manager                : [z/OS_ADV_VUE_qmgr_name]
Bridge Input Queue           : [APPL1.BLOCKCHAIN.INPUT.QUEUE]
MQ Channel                    : []
MQ Conname                    : []
MQ CCDT URL                   : []
JNDI implementation class     : []
JNDI provider URL            : []
MQ Userid                     : []
MQ Password                   : []
```

3. Blockchain ağıyla ilişkili Hyperledger Composer REST sunucusu için kimlik bilgilerini girin (yapılandırıldıysa).

Aşağıdaki örnekte, Hyperledger Composer REST sunucusu **passport-ldapauth NodeJS** modülü kullanılarak bir LDAP kimlik bilgileri deposuyla yapılandırılmıştır. Temel kullanıcı ve parola stili kimlik bilgilerini bu şekilde sağlayan **passport-\*** modüllerinden herhangi birini kullanabildiğinizi unutmayın. Daha fazla bilgi için [REST sunucusu için kimlik doğrulamasını etkinleştirme](#) başlıklı konuya bakın.

```
User Identification
-----
Userid                        : []admin
Password                       : []*****
API path for Login            : auth/ldap
```

4. Hyperledger Composer REST sunucusunun adresini girin.

Bu öznitelikte http ya da https protokolünün gerekli olmadığını ve kapı numarasının zorunlu olduğunu unutmayın. HTTP ya da HTTPS iletişim kuralının kullanılıp kullanılmayacağı REST sunucusunun güvenlik yapılandırmasına bağlıdır. REST sunucusuna bir sertifika ve özel anahtar çifti sağlanırsa, HTTPS kullanılır. HTTPS kullanılır. Ters durumda, HTTP kullanılır. Sertifika ve özel anahtar çiftinin nasıl belirtileceği hakkında bilgi için bkz. “5” sayfa 869.

```
REST Server
-----
Address for Composer REST server : [composer-rest-server-ip-address:3000]
```

5. TLS bağlantıları için sertifika deposu değerlerini girin.

Köprü, bir kuyruk yöneticisine bağlanan bir IBM MQ JMS istemcisi olarak işlev görür; bu, diğer herhangi bir IBM MQ JMS istemcisiyle aynı şekilde güvenli bir şekilde bağlanmak için TLS güvenliğini kullanacak şekilde yapılandırılabilir anlamına gelir. TLS bağlantı ayrıntılarının yapılandırması yalnızca “2” sayfa 868. adımda JNDI ya da CCDT bilgileri belirtildikten sonra gösterilir.

Sertifika depoları Hyperledger Compose IBM MQ Advanced for z/OS VUE kuyruk yöneticiniz için kullanılır. Sertifika depoları belirtilirse, köprü her zaman HTTPSkullanarak Hyperledger REST sunucusuna bağlanmayı dener. Ancak TLS, aşağıdaki seçenek kullanılarak Hyperledger Composer için TLS kullanılmaya devam ederken IBM MQ bağlantıları için devre dışı bırakılabilir.

```
Certificate stores for TLS connections
-----
Personal keystore             : []
Keystore password             : []
Trusted store for signer certs : []
Trusted store password        : []
Use TLS for MQ connection     : [N]
Timeout for Blockchain operations : [12]
```

Daha fazla bilgi için [HTTPS ve TLS](#) kullanılarak REST sunucusunun güvenliğini sağlama başlıklı konuya bakın.

6. İsteğe bağlı: IBM MQ Bridge to blockchain günlük dosyasının konumunu girin.  
Günlük kütüğü adını ve yerini yapılandırma kütüğünde ya da komut satırında belirtebilirsiniz.

```
Behavior of bridge program
-----
Runtime logfile for copy of stdout/stderr : [/var/mqm/errors/runmqbcb.log]
Done.
```

## Sonuçlar

IBM MQ Bridge to blockchain ' in IBM Blockchain ağınıza ve IBM MQ Advanced for z/OS VUE kuyruk yöneticinize bağlanmak için kullandığı yapılandırma dosyasını yarattınız.

## Sonraki adım

“IBM MQ Bridge to blockchain ' in z/OS üzerinde çalıştırılması” sayfa 871 adımları boyunca çalışma

## **z/OS** **MQ Adv. VUE** **Deprecated** **z/OS üzerinde IBM MQ Bridge to blockchain için IBM MQ güvenlik yapılandırması**

IBM MQ Bridge to blockchain ile IBM MQ güvenliğini ayarlarken dikkat edilmesi gereken noktalar.

### Notlar:

- **Deprecated** IBM MQ Bridge to blockchain , 22 Kasım 2022 'deki tüm yayınlarda kullanımdan kaldırılmıştır (bkz. ABD Duyurusu mektubu 222-341). Blockchain bağlantırlığı, IBM App Connect ile ya da IBM Cloud Pak for Integration ile sağlanan App Connect yetenekleriyle elde edilebilir.
- **V 9.3.2** **Removed** Continuous Delivery için IBM MQ Bridge to blockchain , IBM MQ 9.3.2 adresindeki üründen kaldırılır.

Aşağıdaki örnekler, IBM MQ Bridge to blockchain ' e gereksinim duyduğu kuyruklara erişim vermek için kullanılacak RACF tanımlamalarını göstermektedir. Tanımlamalar, köprünün MQBCBUSR kullanıcı kimliği altında çalıştığını varsayar.

Ayrıca, kuyruk yöneticisine bağlanmak için IBM MQ Bridge to blockchain ' e erişim verilmesi gerekir:

- Doğrudan bağ tanımlama kipini kullanma; bkz. [Toplu bağlantılar için bağlantı güvenliği profillerine](#) veya
- CHINIT aracılığıyla istemci kipinin kullanılması; bkz. [Client MQI istekleri](#)

## IBM MQ Bridge to blockchain istek kuyruğu için yetki

Varsayılan SYSTEM.BLOCKCHAIN.INPUT.QUEUE istek kuyruğundan ileti almak üzere MQBCBUSR kullanıcı kimliğine erişim vermek için aşağıdaki RACF komutlarını çalıştırın:

```
RDEFINE MQQUEUE SYSTEM.BLOCKCHAIN.INPUT.QUEUE UACC(NONE)
PERMIT SYSTEM.BLOCKCHAIN.INPUT.QUEUE CLASS(MQQUEUE) ID(MQBCBUSR) ACCESS(UPDATE)
```

## IBM MQ Bridge to blockchain yanıt kuyruğu için yetki

APPL1.BLOCKCHAIN.REPLY.QUEUE' e ileti göndermek üzere MQBCBUSR kullanıcı kimliğine erişim vermek için aşağıdaki RACF komutlarını verin. Bu kuyruk adı, istek iletilisinde yanıtın gönderileceği kuyruk adında belirlenir:

```
RDEFINE MQQUEUE APPL1.BLOCKCHAIN.REPLY.QUEUE UACC(NONE)
PERMIT APPL1.BLOCKCHAIN.REPLY.QUEUE CLASS(MQQUEUE) ID(MQBCBUSR) ACCESS(UPDATE)
PERMIT CONTEXT.APPL1.BLOCKCHAIN.REPLY.QUEUE CLASS(MQADMIN) ID(MQBCBUSR) ACCESS(UPDATE)
```

## İlgili kavramlar

[Kuyruk güvenliğine ilişkin profiller](#)

## İlgili görevler

[“z/OS üzerinde IBM MQ Bridge to blockchain istemcisi örneğini çalıştırma” sayfa 863](#)

IBM MQ Bridge to blockchain ile birlikte sağlanan JMS istemci örneğini, blockchain köprüsünün denetlediğini belirten bir iletiyi giriş kuyruğuna yerleştirmek ve alınan yanıtı görmek için kullanabilirsiniz. Bu örnek, IBM MQ Bridge to blockchain tümleştirmesinin Hyperledger Composer Trader ağ örneğiyle kullanılmasına dayalıdır.

## İlgili başvurular

[API-kaynak güvenliği erişimi hızlı başvurusu](#)

## z/OS MQ Adv. VUE **Deprecated** IBM MQ Bridge to blockchain ' in z/OS üzerinde çalıştırılması

IBM Blockchain ve IBM MQ' e bağlanmak için IBM MQ Bridge to blockchain komutunu çalıştırın. Bağlantı kurulduğunda köprü, istek iletilerini işlemeye, bunları Hyperledger Composer blockchain ağınıza göndermeye ve yanıtları almaya ve işlemeye hazırdır.

## Başlamadan önce

### Notlar:

- **Deprecated** IBM MQ Bridge to blockchain , 22 Kasım 2022 'deki tüm yayınlarda kullanımdan kaldırılmıştır (bkz. [ABD Duyurusu mektubu 222-341](#)). Blockchain bağlantırlığı, IBM App Connect ile ya da IBM Cloud Pak for Integration ile sağlanan App Connect yetenekleriyle elde edilebilir.
- **V 9.3.2** **Removed** Continuous Delivery için IBM MQ Bridge to blockchain , IBM MQ 9.3.2 adresindeki üründen kaldırılır.

## Bu görev hakkında

IBM MQ Bridge to blockchain' yi çalıştırmak için önceki görevde oluşturduğunuz yapılandırma dosyasını kullanın.

## Yordam

1. Köprüyle kullanmak istediğiniz IBM MQ Advanced for z/OS VUE kuyruk yöneticisini başlatın.
2. Blockchain ağınıza ve IBM MQ Advanced for z/OS VUE kuyruk yöneticinize bağlanmak için IBM MQ Bridge to blockchain uygulamasını başlatın.

Aşağıdakilerden birini yapın:

- a) Köprüyü doğrudan z/OS UNIX System Services (z/OS UNIX) içinde IBM MQ ' in kurulu olduğu z/OS UNIX konumundaki mqbc/b'n dizininden çalıştırın.

```
./runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
```

veya

- b) b. th1qua1 . SCSQPROC (CSQ4BCB) içinde sağlanan örnek JCL ' yi kullanarak z/OS sisteminizde köprüyü çalıştırın.

Ortamınıza özgü JCL ' de bir dizi güncelleme yapmanız gerekir:

- ++THLQUAL++ yerine, IBM MQ hedef kitaplık veri kümelerinin üst düzey niteleyicisini koyun.
- ++LANGLETTER++ yerine, iletilerin gösterilmesini istediğiniz dilin harfini koyun.
- ++PATHPREFIX++ ' i z/OS UNIX Bileşenleri kuruluş yoluyla değiştirin.

- ++CONFIGFILE++ dosyasını, z/OS UNIXiçindeki runmqbc -o <file> komutunu kullanarak oluşturulan bir yapılandırma dosyasının yoluyla değiştirin.
- ++JAVAHOME++ ' i, Java 8 ya da sonraki bir sürümde çalışan 64 bit Java Virtual Machine (JVM) konumuyla değiştirin.

Köprü bağlandığında, aşağıdakine benzer bir çıkış döndürülür:

```
2018-05-17 14:28:16.866 BST IBM MQ Bridge to Blockchain
5724-H72 (C) Copyright IBM Corp. 2017, 2024.

2018-05-17 14:28:19.331 BST Ready to process input messages.
```

3. İsteğe bağlı: Köprüyü çalıştırdıktan sonra döndürülen iletiler bir bağlantının başarılı olmadığını gösteriyorsa, IBM MQ Advanced for z/OS VUE kuyruk yöneticinizle ve blockchain ağınıza olan bağlantılarda sorun gidin.

a) Komutu hata ayıklama kipinde 1hata ayıklama seçeneğiyle çalıştırın.

```
./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/
logfile.log -d 1
```

Köprü, bağlantı ayarlarında adım adım ilerler ve iletileri terse kipinde gösterir.

b) Komutu hata ayıklama kipinde 2hata ayıklama seçeneğiyle çalıştırın.

```
./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/
logfile.log -d 2
```

Köprü, bağlantı ayarlarında adım adım ilerler ve işleme iletilerini ayrıntılı kipte gösterir. Tam çıkış günlük dosyanıza yazılır.

İsteğe bağlı olarak, '-d 0' ögesini '-d 1' ya da '-d 2' olarak değiştirerek JCL içindeki hata ayıklama kipi seçeneklerini de belirtebilirsiniz.

## Sonuçlar

IBM MQ Bridge to blockchain ürününü başlatmış ve kuyruk yöneticisine ve blockchain ağınıza bağlanmış bulunuyorsunuz.

## Sonraki adım

- Blockchain ağınıza bir sorgu ya da güncelleme iletisi biçimlendirmek ve göndermek için [“z/OS üzerinde IBM MQ Bridge to blockchain istemcisi örneğini çalıştırma” sayfa 863](#) içindeki adımları izleyin.
- JVM özelliklerini geçirmek için `MQBCB_EXTRA_JAVA_OPTIONS` değişkenini kullanın; örneğin, IBM MQ izlemesini etkinleştirmek için. Daha fazla bilgi için bkz. [IBM MQ Bridge to blockchainizleme](#).

## z/OS üzerinde IBM MQ Bridge to blockchain before IBM MQ 9.2.0 için ileti biçimleri

IBM MQ Bridge to blockchaintarafından gönderilen ve alınan iletilerin biçimlendirilmesine ilişkin bilgiler.



**Uyarı:** İleti biçimleri için var olan biçim eski. IBM MQ 9.2.0' de bir Hyperledger Fabric ağınıza varsa, [“IBM MQ 9.2.0 ' den IBM MQ Bridge to blockchain için ileti biçimleri” sayfa 858](#) içinde açıklanan iletilerin biçimini kullanın.

An application requests that the IBM MQ Bridge to blockchain drives the Hyperledger Composer defined REST API to act on information that is held on the blockchain. Uygulama bunu, köprü istek kuyruğuna bir istek iletisi yerleştirerek yapar. REST isteğinin sonuçları, köprü tarafından bir yanıt iletisine biçimlendirilir. Köprü, istek iletisine ilişkin MQMD ' deki **ReplyToQ** ve **ReplyToQMGR** alanlarında bulunan bilgileri yanıt iletisinin hedefi olarak kullanır.

İstek ve yanıt iletileri, JSON biçimindeki metin (MQSTR) iletileridir.

## İstek İletisi Biçimi

İstek iletileri üç öznitelik içerir:

### yöntem

POST, DELETE ya da GET gibi Hyperledger Composer REST API ' yı çağırmak için kullanılan REST komutu

### yol

Hyperledger Composer REST API yolu. Bu, URLtemel sunucusuna eklenir. Yol "api/" ile başlamalıdır.

### gövde

Yönteme özgü içerik. Bu genellikle bir JSON yapısıdır.

Aşağıdaki örnekte, yeni bir Trader nesnesi yaratmak için POST (hedef yol) api/Traderyöntemi kullanılmıştır. Gövde, kullanıcının Hyperledger Composer modeli tarafından tanımlandığı şekilde Traders sınıfını belirtir ve blockchain ağı içinde yeni bir Trader nesnesi oluşturmak için gereken ek değerleri de belirtir.

```
{
  "method": "POST",
  "path": "api/Trader",
  "body": {
    "$class": "org.example.trading",
    "tradeId": "Trader2",
    "firstName": "Jane",
    "lastName": "Doe"
  }
}
```

## Yanıt İletisi Biçimi

Yanıt iletilerinin ilinti tanıtıcısı, gelen iletinin ileti tanıtıcısına ayarlı. Kullanıcı tanımlı özellikler istek iletilerinden yanıt iletilisine kopyalanır. Yanıttaki kullanıcı kimliği, kaynak kullanıcının kullanıcı kimliğine ayarlanır.

**statusCode** , bir HTTP durum kodudur. Hata IBM MQ ya da köprüden ise, uygun bir **statusCode** kullanılır.

**statusType** , *SUCCESS* ya da *FAILURE* olan bir dizedir.

Başarılı istekler için, yanıt iletilisindeki **data** ögesi, çağrılan Hyperledger Composer REST API ' nin yanıtını içerir.

Başarılı işleme örneği:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "data": [
    {
      "$class": "org.example.trading",
      "firstName": "John",
      "lastName": "Doe",
      "tradeId": "Trader1"
    },
    {
      "$class": "org.example.trading",
      "firstName": "Jane",
      "lastName": "Doe",
      "tradeId": "Trader2"
    }
  ]
}
```

Tüm hata yanıtları, köprü'nün kendisi tarafından oluşturulup oluşturulmadıklarından bağımsız olarak, çağrılardan Hyperledger Composer REST sunucusuna, blockchain 'e ya da zincirleme kod çağrısından aynı alanlara sahiptir. Örneğin:

- Hatalı JSON giriş iletilisi

```
{
  "statusCode": 400,
  "statusType": "FAILURE",
  "message": "[AMQBC021E] Error: Cannot parse input message or there are
  missing fields in the message. Missing fields appear to be: "method"."
}
```

- Hyperledger Composer REST sunucusu tarafından işlenmeyen istek

```
{
  "statusCode": 500,
  "statusType": "FAILURE",
  "message": "Error trying to invoke business network. Error: No valid responses
  from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
  error (status: 500, message: Error: Failed to add object with ID 'Trader1'
  as the object already exists)"
}
```

Uygulamalar, isteğin **statusType** dizgisine bakarak mı, yoksa veri alanının varlığından mı başarılı ya da başarısız olduğunu söyleyebilir. Giriş iletisi işlenirken bir hata oluştuğunda ve köprü bunu blockchain 'e göndermediğinde, köprüden döndürülen değer genellikle **MQRC\_FORMAT\_ERROR**olan bir MQRC değeridir.

## z/OS üzerinde IBM MQ Bridge to blockchain istemcisi örneğini çalıştırma

IBM MQ Bridge to blockchain ile birlikte sağlanan JMS istemci örneğini, blockchain köprüsünün denetlediğini belirten bir iletiyi giriş kuyruğuna yerleştirmek ve alınan yanıtı görmek için kullanabilirsiniz. Bu örnek, IBM MQ Bridge to blockchain tümleştirmesinin Hyperledger Composer Trader ağ örneğiyle kullanılmasına dayalıdır.

### Başlamadan önce



Daha fazla bilgi için bkz. [/trade\\_network](#)

IBM MQ Bridge to blockchain ürününüz çalışıyor ve IBM MQ Advanced ya da IBM MQ Advanced for z/OS VUE kuyruk yöneticinize ve blockchain ağınıza bağlı.

### Bu görev hakkında

JMS örnek uygulamasını (ComposerBCBSamp.java) IBM MQ Bridge to blockchain' in samp dizininde bulun.

Örneğin: <MQ\_INSTALL\_ROOT>/mqbc/samp/ComposerBCBSamp.java, burada <MQ\_INSTALL\_ROOT> :

-  IBM MQ ' in kurulu olduğu dizin
-  IBM MQ ürününün z/OS UNIX bileşenlerinin kurulu olduğu z/OS UNIX System Services dizini

### Yordam

1. İstemci örnek Java kaynak dosyasını düzenleyin.

IBM MQ ortamınızla ve blockchain ağınıza eşleşecek şekilde yapılandırmak için örnekteki yönergeleri izleyin.

Örnekteki aşağıdaki kod, köprüye gönderilecek üç JSON istek iletisini tanımlar:

- a. Öncelikle, var olan bir 'commodity' ögesini kaldırmak için

- b. İkinci olarak, yeni bir 'commodity', 'owner' ve ilişkili değerler oluşturmak için,  
c. Son olarak, önceki iki istek iletisini izleyerek 'commodity' ile ilgili yeni bilgileri görüntüler

```
private static JSONObject[] createMessageBodies() {
    JSONObject[] msgs = new JSONObject[3]; // This method creates 3 messages
    JSONObject m, m2;
    String commodityName = "BC";

    // Clean out the commodity in case it's already there. If
    // it's not there, there will be an error returned from Composer.
    m = new JSONObject();
    m.put("method", "DELETE");
    m.put("path", "api/Commodity/" + commodityName);
    msgs[0] = m;

    // To add the item to the table, the
    // operation looks like this:
    //
    // { "method": "POST",
    //   "path": "api/Commodity",
    //   "body" : {
    //     "$class": "org.example.trading.Commodity",
    //     "tradingSymbol" : "BC",
    //     "description" : "BC",
    //     "mainExchange" : "HERE",
    //     "owner" : "Me",
    //     "quantity" : 100
    //   }
    // }
    // You can see this structure in the API Explorer
    m = new JSONObject();
    m.put("method", "POST");
    m.put("path", "api/Commodity");
    m2 = new JSONObject();
    m2.put("$class", " org.example.trading.Commodity");
    m2.put("tradingSymbol", commodityName);
    m2.put("description", "Blockchain Sample Description");
    m2.put("mainExchange", "My Exchange");
    m2.put("owner", "Me");
    m2.put("quantity", 100);
    m.put("body", m2);
    msgs[1] = m;

    // And list all items that have been created
    m = new JSONObject();
    m.put("method", "GET");
    m.put("path", "api/Commodity");
    msgs[2] = m;

    return msgs;
}
```

## 2. Örneği derleyin.

Köprü dizininde gönderilen IBM MQ istemci sınıflarını ve JSON4J . jar dosyasını işaret edin.

```
javac -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar
ComposerBCClient.java
```

## 3. Derlenmiş sınıfı çalıştırın.

```
java -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar:.
ComposerBCClient
```

```
Starting Simple MQ Blockchain Bridge Client
Starting the connection.
Sent message:
{"method":"DELETE"," path ":"api\\Commodity\\BC"}
Response text:
{
  "statusCode": 204,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}
```

```

}
SUCCESS
Sent message:
{"body":
{"$class": "org.example.trading.Commodity", "owner": "Me", "quantity": 100, "description": "Blockcha
in Sample Description", "mainExchange": "My
Exchange", "tradingSymbol": "BC"}, "operation": "POST", "url": "Commodity"}
Response text:
{
"statusCode": 200,
"statusType": "SUCCESS",
"message": "OK",
"data": {
"$class": "org.example.trading.Commodity",
"description": "Blockchain Sample Description",
"mainExchange": "My Exchange",
"owner": "Me",
"quantity": 100,
"tradingSymbol": "BC"
}
}
}
SUCCESS
Sent message:
{"method": "GET", "path": "api/Commodity"}
Response text:
{
"statusCode": 200,
"statusType": "SUCCESS",
"message": "OK",
"data": [
{
"$class": "org.example.trading.Commodity",
"description": "Blockchain Sample Description",
"mainExchange": "My Exchange",
"owner": "resource:org.example.trading.Trader#Me",
"quantity": 100,
"tradingSymbol": "BC"
}
]
}
}
SUCCESS

```

**message** alanı, başarıyla işlenen bir ileti için "OK" ya da başarısız bir istek durumunda, hatanın nedenine ilişkin bilgileri içerir.

İstemci yanıtı beklerken bir zamanaşımı hatası alırsa, köprünün çalışıp çalışmadığını denetleyin.

z/OS

## z/OS üzerinde kuyruk yöneticilerini yapılandırma

IBM MQ for z/OS üzerinde kuyruk yöneticilerini yapılandırmak için bu yönergeleri kullanın.

### Başlamadan önce

IBM MQ for z/OS' yi yapılandırmadan önce aşağıdakileri okuyun:

- [IBM MQ for z/OS kavramları](#)
- [IBM MQ ortamınızın z/OS üzerinde planlanması](#)

### Bu görev hakkında

IBM MQ' i kurduktan sonra, kullanıcıların kullanımına sunabilmeniz için önce birkaç görevi gerçekleştirmeniz gerekir.

### Yordam

- IBM MQ for z/OS' da kuyruk yöneticilerinin nasıl yapılandırılacağını öğrenmek için aşağıdaki alt başlıklara bakın.

### İlgili kavramlar

z/OS

[IBM MQ for z/OS üzerinde MQSC ve PCF komutlarını yayınlayabileceğiniz kaynaklar](#)



## İlgili görevler

[“Çoklu platformlarda kuyruk yöneticileri oluşturma” sayfa 7](#)

İletileri ve kuyrukları kullanabilmek için önce en az bir kuyruk yöneticisi ve ilişkili nesnelere yaratmanız ve başlatmanız gerekir. Bir kuyruk yöneticisi, kendisiyle ilişkili kaynakları, özellikle de sahip olduğu kuyrukları yönetir. IBM MQ nesnelere yaratmak, değiştirmek, görüntülemek ve silmek için İleti Kuyruklama Arabirimi (MQI) çağrılarını ve komutlarına ilişkin uygulamalara kuyruğa alma hizmetleri sağlar.

güvenlik

[“Dağıtılmış kuyruğa alma yapılandırılıyor” sayfa 189](#)

Bu bölümde, kuyruk tanımlaması, kanal tanımlaması, tetikleme ve eşitleme noktaları yordamları da içinde olmak üzere IBM MQ kuruluşları arasındaki iletişimle ilgili daha ayrıntılı bilgi sağlar.

[“İstemci ile sunucu arasındaki bağlantıların yapılandırılması” sayfa 14](#)

IBM MQ MQI clients ile sunucular arasındaki iletişim bağlantılarını yapılandırmak için iletişim protokolünüze karar verin, bağlantının her iki ucundaki bağlantıları tanımlayın, bir dinleyici başlatın ve kanalları tanımlayın.

[z/OS](#) YönetmeIBM MQ for z/OS

Planlama

## İlgili başvurular

[z/OS](#) IBM MQ for z/OS yardımcı programlarının kullanılması

## [z/OS](#) z/OS üzerinde kuyruk yöneticilerini özelleştirme hazırlığı

Kuyruk yöneticilerinizi kurulabilir özellikler, ulusal dil özellikleri ve sınama ve güvenlik ayarlarıyla ilgili bilgilerle uyarlarken bu konuyu kullanın.

## Uyarılama için hazırlık

Program Dizini, IBM MQ kuruluş manyetik bandının içeriğini, IBM MQ için program ve hizmet seviyesi bilgilerini listeler ve IBM MQ for z/OS ürününün, System Modifikasyon Programı Extended (SMP/E) kullanılarak nasıl kurulacağını açıklar. Program Dizinlerine ilişkin yükleme bağlantıları için bkz. [IBM MQ for z/OS Program Dizini PDF dosyaları](#).

IBM MQ' i kurduğunuzda, kullanıcıların kullanımına sunmadan önce bir dizi görevi gerçekleştirmeniz gerekir. Bu görevlere ilişkin açıklamalar için aşağıdaki bölümlere bakın:

- [“ayarılamaIBM MQ for z/OS” sayfa 881](#)
- [“z/OS üzerinde bir kuyruk yöneticisinin sınanması” sayfa 946](#)
- [z/OS üzerinde güvenliğin ayarlanması](#)

Önceki bir IBM MQ for z/OS sürümünden geçiş yapıyorsanız, özelleştirme görevlerinin çoğunu gerçekleştirmeniz gerekmez. Gerçekleştirmeniz gereken görevlerle ilgili ek bilgi için [Bakım ve Geçiş](#) başlıklı konuya bakın.

## IBM MQ for z/OS ürününün kurulabilir özellikleri

IBM MQ for z/OS aşağıdaki özelliklerden oluşur:

### Taban

Bu gereklidir; aşağıdakiler de dahil olmak üzere tüm ana işlevleri içerir:

- Yönetim ve yardımcı programlar
- IBM MQ Application Programming Interface ya da C++ kullanarak CICS, IMS ve toplu iş tipi uygulamalar için destek
- Dağıtılmış kuyruğa alma olanağı (hem TCP/IP, hem de APPC iletişimini destekler)

### Ulusal dil özellikleri

Bunlar, desteklenen tüm ulusal dillerdeki hata iletilerini ve panoları içerir. Her dille ilişkilendirilmiş bir dil harfi vardır. Diller ve harfler şunlardır:

**C**

Basitleştirilmiş Çince

**E**

U.S. İngilizce (büyük ve küçük harf karışık)

**F**

Fransızca

**K**

Japonca

**U**

U.S. İngilizce (büyük harf)

ABD İngilizcesi (büyük ve küçük harf karışık) seçeneğini kurmanız gerekir. Bir ya da daha çok dil de kurabilirsiniz. (Diğer dillere ilişkin kurulum süreci, ABD İngilizcesini (büyük ve küçük harf karışık) kullanmayacak olsanız bile, ABD İngilizcesinin (büyük ve küçük harf karışık) kurulmasını gerektirir.)

**IBM MQ for z/OS UNIX System Services Components**

Bu özellik isteğe bağlıdır. IBM MQ for z/OS' e bağlanmak için [Jakarta Messaging 3.0](#) ya da Java Message Service 2.0 kullanan Java uygulamaları oluşturmak ve çalıştırmak istiyorsanız bu özelliği seçin.

IBM MQ for z/OS UNIX System Services sürümünü kurmaya ilişkin bilgi için bkz. [IBM MQ for z/OS Program Dizini PDF dosyaları](#) .

**IBM MQ for z/OS UNIX System Services Web Components**

Bu özellik isteğe bağlıdır.

IBM MQ Consoleya da REST API kullanmak istiyorsanız bu özelliği seçin.

Bu özelliği kurmak için IBM MQ for z/OS UNIX System Services Components özelliğini kurmanız gerekir.

**IBM MQ for z/OS Managed File Transfer**

Bu aksam isteğe bağlıdır ve yalnızca IBM MQ Advanced for z/OS, IBM MQ for z/OS Value Unit Edition (VUE) ya da IBM MQ for z/OS Managed File Transfer için yetkiniz varsa kurulmalıdır.

IBM MQ for z/OS ürününün Managed File Transfer yeteneklerini kullanmak istiyorsanız bu özelliği seçin.

Bu özelliği kurmak için IBM MQ for z/OS UNIX System Services Components özelliğini kurmanız gerekir.

**Kuruluştan sonra var olan kitaplıklar**

IBM MQ ayrı yükleme kitaplıkları ile birlikte sağlanır. [Çizelge 53 sayfa 878](#) içinde, IBM MQ kurulduktan sonra var olabilecek kitaplıklar gösterilmektedir.

<i>Çizelge 53. Kuruluştan sonra var olan IBM MQ kitaplıkları</i>	
<b>Ad</b>	<b>Açıklama</b>
thlqual.SCSQANLC	IBM MQ' in Basitleştirilmiş Çince sürümüne ilişkin yükleme modüllerini içerir.
thlqual.SCSQANLE	U.S' ye ilişkin yükleme modüllerini içerir. IBM MQ' in İngilizce (büyük ve küçük harf karışık) sürümü.
thlqual.SCSQANLF	Fransızca IBM MQ sürümüne ilişkin yükleme modüllerini içerir.
thlqual.SCSQANLK	IBM MQ Japonca sürümüne ilişkin yükleme modüllerini içerir.
thlqual.SCSQANLU	U.S' ye ilişkin yükleme modüllerini içerir. IBM MQ İngilizce (büyük harf) sürümü.
thlqual.SCSQASMS	Çevirici örnek programlarına ilişkin kaynağı içerir.

Çizelge 53. Kuruluştan sonra var olan IBM MQ kitaplıkları (devamı var)

Ad	Açıklama
thlqual.SCSQAUTH	Tüm IBM MQ ürün yükleme modüllerine ilişkin ana havuz; CSQZPARM varsayılan değiştirge modülünü de içerir. Bu kitaplık APF yetkili ve PDS-E biçiminde olmalıdır.
thlqual.SCSQCICS	CICS DFHRPL birleştirmesine dahil edilmesi gereken ek yükleme modüllerini içerir. Bu kitaplık APF yetkili ve PDS-E biçiminde olmalıdır.
thlqual.SCSQCLST	Örnek programlar tarafından kullanılan CLIT ' leri içerir.
thlqual.SCSQCOBC	Örnek programlar için gerekli olan kopya kitapları da dahil olmak üzere COBOL kopya kitaplarını içerir.
thlqual.SCSQCOBS	COBOL örnek programları için kaynak içerir.
thlqual.SCSQCPPS	C++ örnek programlarına ilişkin kaynağı içerir.
thlqual.SCSQC37S	C örnek programlarına ilişkin kaynağı içerir.
thlqual.SCSQC370	Örnek programlar için gerekli üstbilgiler de içinde olmak üzere C üstbilgilerini içerir.
thlqual.SCSQDEFS	Paylaşılan kuyruğa alma için C++ ve Db2 DBRM ' leri için yan tanımlamaları içerir.
thlqual.SCSQEXEC	IBM MQ işlemleri ve denetim panolarını kullanıyorsanız, SYSEXEC ya da SYSPROC birleştirmesine eklenecek REXX yürütülür dosyalarını içerir.
thlqual.SCSQFCMD	Managed File Transfer görevlerini yaratmak ve çalıştırmak için kullanılacak işlere ilişkin şablonları içerir.
thlqual.SCSQHPPS	C + + için üstbilgi dosyalarını içerir.
thlqual.SCSQINST	Kuruluş işleri için JCL içerir.
thlqual.SCSQLINK	Erken kod kütüphanesi. Sistem ilk program yüklemesi (IPL) sırasında yüklenen yükleme modüllerini içerir. Kitaplık APF-yetkili olmalıdır.
thlqual.SCSQLOAD	Kitaplığı yükleyin. APF dışı kod, kullanıcı çıkışları, yardımcı programlar, örnekler, kuruluş doğrulama programları ve bağdaştırıcı sınırlı kod öbeklerine ilişkin yükleme modüllerini içerir. Kitaplığın APF yetkisi olması gerekmez ve bağlantı listesinde olması gerekmez. Bu kitaplık PDS-E biçiminde olmalıdır.
thlqual.SCSQMACS	Örnek makrolar, ürün makroları ve sistem parametresi makroları da içinde olmak üzere çevirici makrolarını içerir.
thlqual.SCSQMAPS	Örnek programlar tarafından kullanılan CICS eşkümelerini içerir.
thlqual.SCSQMSGC	IBM MQ işlemleri ve denetim panoları için Basitleştirilmiş Çince özelliğini kullanıyorsanız, ISPMLIB birleştirmesine eklenecek ISPF iletilerini içerir.
thlqual.SCSQMSGE	U.Skullanıyorsanız, ISPMLIB birleştirmesine eklenecek ISPF iletilerini içerir. IBM MQ işlemleri ve denetim panoları için İngilizce (büyük ve küçük harf karışık) dil özelliği.

<i>Çizelge 53. Kuruluştan sonra var olan IBM MQ kitaplıkları (devamı var)</i>	
<b>Ad</b>	<b>Açıklama</b>
thlqual.SCSQMSGF	IBM MQ işlemleri ve denetim panoları için Fransızca dil özelliğini kullanıyorsanız, ISPMLIB birleştirmesine eklenecek ISPF iletilerini içerir.
thlqual.SCSQMSGK	IBM MQ işlemleri ve denetim panoları için Japonca dil özelliğini kullanıyorsanız, ISPMLIB birleştirmesine eklenecek ISPF iletilerini içerir.
thlqual.SCSQMSGU	U.Skullanıyorsanız, ISPMLIB birleştirmesine eklenecek ISPF iletilerini içerir. IBM MQ işlemleri ve denetim panoları için İngilizce (büyük harfli) dil özelliği.
thlqual.SCSQMVR1	Dağıtılmış kuyruğa alma için yükleme modüllerini içerir. Bu kitaplık APF yetkili ve PDS-E biçiminde olmalıdır.
thlqual.SCSQPLIC	PL/I içerme dosyalarını içerir.
thlqual.SCSQPLIS	PL/I örnek programlarına ilişkin kaynağı içerir.
thlqual.SCSQPNLA	Döküm biçimleyicinin ISPPLIB birleştirmesine dahil edilmesi için IPCS panolarını içerir. IBM MQ örnek programlarına ilişkin panoları da içerir.
thlqual.SCSQPNL	IBM MQ işlemleri ve denetim panoları için Basitleştirilmiş Çince özelliğini kullanıyorsanız, ISPPLIB birleşiminin içereceği ISPF panolarını içerir.
thlqual.SCSQPNL	U.Skullanıyorsanız, ISPPLIB birleşiminin içereceği ISPF panolarını içerir. IBM MQ işlemleri ve denetim panoları için İngilizce (büyük ve küçük harf karışık) dil özelliği.
thlqual.SCSQPNLF	IBM MQ işlemleri ve denetim panoları için Fransızca dil özelliğini kullanıyorsanız, ISPPLIB birleşiminin içereceği ISPF panolarını içerir.
thlqual.SCSQPNLK	IBM MQ işlemleri ve denetim panoları için Japonca dil özelliğini kullanıyorsanız, ISPPLIB birleşiminin içereceği ISPF panolarını içerir.
thlqual.SCSQPNU	U.Skullanıyorsanız, ISPPLIB birleşiminin içereceği ISPF panolarını içerir. IBM MQ işlemleri ve denetim panoları için İngilizce (büyük harfli) dil özelliği.
thlqual.SCSQPROC	Örnek JCL ve varsayılan sistem kullanıma hazırlama veri kümelerini içerir.
thlqual.SCSQSNLC	Özel amaçlı işlev (örneğin, erken kod) için gerekli olan IBM MQ modüllerinin Basitleştirilmiş Çince sürümlerine ilişkin yükleme modüllerini içerir.
thlqual.SCSQSNLE	U.S' ye ilişkin yükleme modüllerini içerir. Özel amaçlı işlev için gerekli olan IBM MQ modüllerinin İngilizce (büyük ve küçük harf karışık) sürümleri (örneğin, erken kod).
thlqual.SCSQSNLF	Özel amaçlı işlev için gerekli IBM MQ modüllerinin Fransızca sürümlerine ilişkin yükleme modüllerini içerir (örneğin, erken kod).

Çizelge 53. Kuruluştan sonra var olan IBM MQ kitaplıkları (devamı var)	
Ad	Açıklama
thlqual.SCSQSNLK	Özel amaçlı işlev (örneğin, erken kod) için gerekli olan IBM MQ modüllerinin Japonca sürümlerine ilişkin yükleme modüllerini içerir.
thlqual.SCSQSNLU	U.S' ye ilişkin yükleme modüllerini içerir. Özel amaçlı işlev için gerekli olan IBM MQ modüllerinin İngilizce (büyük harfli) sürümleri (örneğin, erken kod).
thlqual.SCSQTBLC	IBM MQ işlemleri ve denetim panoları için Basitleştirilmiş Çince özelliğini kullanıyorsanız, ISPTLIB birleştirmesine dahil edilecek ISPF çizelgelerini içerir.
thlqual.SCSQTBLE	U.Skullanıyorsanız, ISPTLIB birleştirmesine eklenecek ISPF çizelgelerini içerir. IBM MQ işlemleri ve denetim panoları için İngilizce (büyük ve küçük harf karışık) dil özelliği.
thlqual.SCSQTBLE	IBM MQ işlemleri ve denetim panoları için Fransızca dil özelliğini kullanıyorsanız, ISPTLIB birleştirmesine eklenecek ISPF çizelgelerini içerir.
thlqual.SCSQTBLE	IBM MQ işlemleri ve denetim panoları için Japonca dil özelliğini kullanıyorsanız, ISPTLIB birleştirmesine eklenecek ISPF çizelgelerini içerir.
thlqual.SCSQTBLE	U.Skullanıyorsanız, ISPTLIB birleştirmesine eklenecek ISPF çizelgelerini içerir. IBM MQ işlemleri ve denetim panoları için İngilizce (büyük harfli) dil özelliği.

**Not:** Bu kitaplıkların hiçbirini değiştirmeyin ya da uyarlamayın. Değişiklik yapmak istiyorsanız, kitaplıkları kopyalayın ve kopyalarda değişiklik yapın.

### İlgili kavramlar

IBM MQ for z/OS kavramlar

“IBM MQ ' yi IMS ile kullanma” sayfa 984

IBM MQ -IMS bağdaştırıcısı ve IBM MQ - IMS köprüsü, IBM MQ ' in IMS ile etkileşimde bulunmasına izin veren iki bileşendir.

“IBM MQ ' yi CICS ile kullanma” sayfa 992

IBM MQ ürününü CICS ile kullanmak için IBM MQ CICS bağdaştırıcısını ve isteğe bağlı olarak IBM MQ CICS bridge bileşenlerini yapılandırmanız gerekir.

“IMS içinde OTMA çıkışlarının kullanılması” sayfa 994

IMS Open Transaction Manager Access ürününü IBM MQ for z/OS ile birlikte kullanmak istiyorsanız bu konuyu kullanın.

### İlgili görevler

“z/OS üzerinde diğer kuyruk yöneticileriyle iletişim kurulması” sayfa 954

Bu bölümde, dağıtılmış kuyruğa alma işlevini kullanmaya başlamadan önce yapmanız gereken IBM MQ for z/OS hazırlıkları açıklanmaktadır.

Yönetme IBM MQ for z/OS

### İlgili başvurular

“Hizmeti Dil Ortamına ya da z/OS Callable Services 'e yükseltme ve uygulama” sayfa 992

Gerçekleştirmeniz gereken işlemler, CALLLIBS mi, LINK mi ve SMP/E sürümünüze göre değişir.

## ► z/OS ayarlama IBM MQ for z/OS

IBM MQ for z/OS sisteminizi uyarlamak için adım adım bu konuyu kullanın.

Bir kuyruk yöneticisini yapılandırmanın en iyi yolu, aşağıdaki adımları gösterilen sırayla gerçekleştirmektir:

1. Temel kuyruk yöneticisini yapılandırın.
2. Kuyruk yöneticisi iletişimi ve uzak istemci uygulaması iletişimi için kuyruk yöneticisi gerçekleştiren kanal başlatıcısını yapılandırın.
3. İletileri şifrelemek ya da korumak istiyorsanız, Advanced Message Security for z/OS' i yapılandırın.
4. Dosyaları aktarmak için IBM MQ 'yi kullanmak istiyorsanız, Managed File Transfer for z/OS' yi yapılandırın.
5. Yönetim ya da ileti sistemini REST APIya da IBM MQ ' yi bir web tarayıcısından yönetmek için IBM MQ Console kullanmak istiyorsanız, mqweb sunucusunu yapılandırın.

Bu konu, IBM MQ ürününü başarıyla kurduktan sonra, bu konunun çeşitli aşamalarında size yol gösterecektir. Kuruluş işlemi Program Dizininde açıklanmıştır. Program Dizinlerine ilişkin yükleme bağlantıları için bkz. [IBM MQ for z/OS Program Dizini PDF dosyaları](#).

Örnekler, uyarılmanızda size yardımcı olması için IBM MQ ile birlikte sağlanır. Örnek veri kümesi üyelerinin adları CSQ4 ile başlar ve thlqual.SCSQPROCKitaplığında bulunur.

Bu konuda açıklanan özelleştirme görevlerini gerçekleştirmeden önce, IBM MQ for z/OS' in performans ve kaynak gereksinimlerini etkilediği için göz önünde bulundurmanız gereken bir dizi yapılandırma seçeneği vardır. Örneğin, hangi küreselleşme kitaplıklarını kullanmak istediğinize karar vermeniz gerekir.

Bazı özelleştirme adımlarını otomatikleştirmek istiyorsanız, bkz. [“IBM z/OSMF olanağının otomatikleştirmek için kullanılması IBM MQ” sayfa 998](#).

## Yapılandırma seçenekleri

Bu seçeneklerle ilgili daha fazla bilgi için bkz. [z/OS üzerinde planlama](#).

Bu bölümdeki her bir görevin tanımı aşağıdakilerden biri olup olmadığını gösterir:

- Görev, IBM MQ' yi ayarlama sürecinin bir parçasıdır. Yani, z/OS sisteminde IBM MQ ' i özelleştirdiğinizde görevi bir kez gerçekleştirmeniz gerekir. (Paralel bir sistem şebekesinde, sistem şebekesinde her z/OS sistemi için görevi gerçekleştirmeniz ve her bir z/OS sisteminin aynı şekilde ayarlandığından emin olmanız gerekir.)
- Görev, kuyruk yöneticisi eklenmesinin bir parçasıdır. Yani, bu kuyruk yöneticisini eklediğinizde, görevi her kuyruk yöneticisi için bir kez gerçekleştirmeniz gerekir.

Çeşitli z/OS sistem parametrelerini değiştirmek için komutlar kullanırsanız ve önerildiği gibi “SYS1.PARMLIB üyeleri” sayfa 896 işlemi gerçekleştirirseniz, görevlerin hiçbiri z/OS sisteminizin IPL işlemini gerçekleştirmenizi gerektirmez.

Operasyonları basitleştirmek ve sorun belirlemeye yardımcı olmak için bir sistem şebekesinde bulunan tüm z/OS sistemlerinin aynı şekilde kurulduğundan emin olun, böylece kuyruk yöneticileri acil durumlarda herhangi bir sistemde hızlı bir şekilde oluşturulabilir.

Bakım kolaylığı için IBM MQ kitaplıklarınıza başvurmak üzere diğer adları tanımlamayı düşünün; daha fazla bilgi için [IBM MQ kitaplığına başvurmak için diğer ad kullanmabaşlıklı konuya](#) bakın.

### İlgili kavramlar

[IBM MQ for z/OS kavramlar](#)

[“IBM MQ ' yi IMS ile kullanma” sayfa 984](#)

IBM MQ -IMS bağdaştırıcısı ve IBM MQ - IMS köprüsü, IBM MQ ' in IMS ile etkileşimde bulunmasına izin veren iki bileşendir.

[“IBM MQ ' yi CICS ile kullanma” sayfa 992](#)

IBM MQ ürününü CICS ile kullanmak için IBM MQ CICS bağdaştırıcısını ve isteğe bağlı olarak IBM MQ CICS bridge bileşenlerini yapılandırmanız gerekir.

[“IMS içinde OTMA çıkışlarının kullanılması” sayfa 994](#)

IMS Open Transaction Manager Access ürününü IBM MQ for z/OS ile birlikte kullanmak istiyorsanız bu konuyu kullanın.

## İlgili görevler

[“z/OS üzerinde diğer kuyruk yöneticileriyle iletişim kurulması” sayfa 954](#)

Bu bölümde, dağıtılmış kuyruğa alma işlevini kullanmaya başlamadan önce yapmanız gereken IBM MQ for z/OS hazırlıkları açıklanmaktadır.

Yönetme [IBM MQ for z/OS](#)

## İlgili başvurular

[“Hizmeti Dil Ortamına ya da z/OS Callable Services 'e yükseltme ve uygulama” sayfa 992](#)

Gerçekleştirmeniz gereken işlemler, CALLLIBS mi, LINK mi ve SMP/E sürümünüze göre değişir.

## **z/OS** z/OS sisteminin IBM MQ için yapılandırılması

IBM MQ for z/OS sisteminizi özelleştirmek için adım adım bu konuları kullanın.

## **z/OS** z/OS sistem parametrelerinin tanımlanması

Bazı görevler z/OS sistem parametrelerinin güncellenmesini içerir. Sistem IPL işlemi gerçekleştirilirken hangilerinin belirlendiğini bilmeniz gerekir.

- *Bu görevi, çalıştırmak istediğiniz her z/OS sistemi için bir kez gerçekleştirmeniz gerekir IBM MQ.*
- *Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekebilir.*

SYS1.PARMLIB(IEASYSpp), SYS1.PARMLIB (burada pp, sistemin IPL 'sini gerçekleştirmek için kullanılan z/OS sistem parametresi listesini gösterir).

Bulmanız gereken girdiler şunlardır:

### **“APF IBM MQ yükleme kitaplıklarını yetkilendirir” sayfa 883 için:**

PROG=xx ya da APF=aa, Yetkili Program Olanağı (APF) yetkili kitaplık listesini (üye PROGxx ya da IEFAPFaa) işaret eden

### **“z/OS bağlantı listesini ve Yaşam Çevrimi Projesi Yönetimi” sayfa 884 için:**

LNK=kk bağlantı listesini gösteriyor (üye LNKLSTkk) LPA=mm, LPA listesini gösteriyor (üye LPALSTmm)

### **“z/OS program özellikleri çizelgesinin güncellenmesi” sayfa 888 için:**

SCH=xx, Program Özellikleri Çizelgesini (PPT) (SCHEDxx üyesi) gösterir

### **“IBM MQ altsistemini şu şekilde tanımlayın: z/OS” sayfa 889 için:**

SSN=ss tanımlı altsistem listesini gösterir (üye IEFSSNss)

## **z/OS** APF IBM MQ yükleme kitaplıklarını yetkilendirir

APF-çeşitli kitaplıkları yetkilendir. Bazı yükleme modüllerinin yetkisi önceden verilmiş olabilir.

- *Bu görevi, çalıştırmak istediğiniz her z/OS sistemi için bir kez gerçekleştirmeniz gerekir IBM MQ.*
- *Kuyruk paylaşım grupları kullanıyorsanız, sistem şebekesinde her z/OS sisteminde IBM MQ ayarlarının aynı olduğundan emin olmanız gerekir.*
- *Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekebilir.*
- *Library Look aside (LLA) kullanımı:*
  - *Bazı IBM MQ kullanımı, yüksek Giriş/Çıkış (IO) modüllerinin kitaplıklardan yüklenmesine neden olabilir. Bu G/Ç, işletim sisteminin LLA olanağı kullanılarak azaltılabilir.*
  - *Bu yüksek G/Ç aşağıdaki durumlarda oluşabilir:*
    - *Yüksek MQCONN/MQDISC hızına sahip uygulamalar; örneğin, bir WLM saklanmış yordamında.*
    - *Kanal çıkışları yükleniyor. Sık sık başlayan ve durduran kanallarınız varsa ve kanal çıkışlarını kullanıyorsanız.*
  - *SYS1.PARMLIB, LLA kuruluşunu belirtir. LIBRARIES deyimine bir kitaplık adının eklenmesi, bir program kopyasının her zaman VLF ' den (Virtual Lookaside Facility) alınacağı ve bu nedenle yoğun bir şekilde kullanıldığında genellikle G/Ç gerektirmediği anlamına gelir.*

*FREEZE* deyimindeki dahil etme, ilgili *DD* deyimini birleştirme dizinlerini almak için *G/Ç* olmadığı anlamına gelir (bu genellikle programın kendisinden daha fazla *G/Ç* olabilir).

*İşletim sistemi komutunu kullan " F LLA, REFRESH " bu kitaplıklardan herhangi birinde herhangi bir değişiklik yapıldıktan sonra.*

IBM MQ yükleme kitaplıkları `thlqual.SCSQAUTH` ve `thlqual.SCSQLINK` APF-yetkili olmalıdır. Ayrıca, kitaplıkları ulusal dil özelliğinize (`thlqual.SCSQANLX` ve `thlqual.SCSQSNLX`) ve dağıtılmış kuyruğa alma özelliğine (`thlqual.SCSQMVR1`) ilişkin olarak da yetkilendirmeniz gerekir.

Ancak, LPA 'daki tüm yük modülleri otomatik olarak APF-yetkilendirilir. `SYS1.PARMLIB` üyesi `IEASYSpp` şu deyim içerir:

```
LNKAUTH=LNKLST
```

`LNKAUTH=LNKLST`, `LNKAUTH` belirtilmezse varsayılan değerdir.

Yaşam çevrimi projesi yönetimi ya da bağlantı listesine ne eklemeyi seçtiğine bağlı olarak (bkz. "[z/OS bağlantı listesini ve Yaşam Çevrimi Projesi Yönetimi](#)" sayfa 884 ), kitaplıkları APF bağlantı listesine yerleştirmeniz gerekmeyebilir

**Not:** IBM MQ STEPLIB 'de yer alan tüm kitaplıklara APF yetkisi vermeniz gerekir. STEPLIB 'e APF yetkisi olmayan bir kitaplık koyarsanız, tüm kitaplık birleştirme işlemi APF yetkisini kaybeder.

APF listeleri `SYS1.PARMLIB` üyesi `PROGxx` ya da `IEAAPFaa`. Listeler, APF yetkili z/OS kitaplıklarının adlarını içerir. Listelerdeki girdilerin sırası önemli değildir. APF listelerine ilişkin bilgi için [APF yetkili kitaplık listesi](#) başlıklı konuya bakın.

Sisteminizi ayarlama hakkında daha fazla bilgi için bkz. [SupportPac MP16](#)

Dinamik biçimde `PROGxx` üyelerini kullanıyorsanız, değişikliklerin yürürlüğe girmesi için yalnızca z/OS komutu `SETPROG APF ,ADD ,DSNAME=h1q .SCSQ XXXX ,VOLUME= YYYYYY` vermeniz gerekir: Burada `XXXX` , kitaplık adına göre değişir ve burada `YYYYY` birim olur. Ters durumda, statik biçim ya da `IEAAPFaa` üyeleri kullanıyorsanız, sisteminizde bir IPL gerçekleştirmeniz gerekir.

APF listesinde kitaplığın gerçek adını kullanmanız gerektiğini unutmayın. Kitaplığın veri kümesi diğer adını kullanma girişiminde bulunursanız, yetki başarısız olur.

### İlgili kavramlar

["z/OS bağlantı listesini ve Yaşam Çevrimi Projesi Yönetimi" sayfa 884](#)

LPA kitaplıklarını, erken kod kitaplıklarının yeni sürümüyle güncelleyin. Diğer kod bağlantı listesine ya da Yaşam Çevrimi Projesi Yönetimi listesine girebiliyor.

["z/OS üzerinde kuyruk yöneticilerini özelleştirme hazırlığı" sayfa 877](#)

Kuyruk yöneticilerinizi kurulabilir özellikler, ulusal dil özellikleri ve sinama ve güvenlik ayarlarıyla ilgili bilgilerle uyarlarken bu konuyu kullanın.

### **z/OS bağlantı listesini ve Yaşam Çevrimi Projesi Yönetimi**

LPA kitaplıklarını, erken kod kitaplıklarının yeni sürümüyle güncelleyin. Diğer kod bağlantı listesine ya da Yaşam Çevrimi Projesi Yönetimi listesine girebiliyor.

- Bu görevi, IBM MQ' un çalıştırılmasını istediğiniz her z/OS sistemi için bir kez gerçekleştirmeniz gerekir.
- Kuyruk paylaşım gruplarını kullanıyorsanız, kuyruk yöneticilerinden herhangi birini IBM MQ 9.3.0'e geçirmeden önce QSG' deki her kuyruk yöneticisindeki erken kodu IBM MQ 9.3.0 düzeyine yenilemeniz gerekir.

Her LPAR 'da en son erken kodu kurun ve geçiş öncesinde kuyruk yöneticilerini birer birer yenileyin. Tüm kuyruk yöneticilerini aynı anda geçirmeniz gerekmez.

- Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekebilir. Daha fazla ayrıntı için Program Dizinine bakın. Program Dizinine ilişkin yükleme bağlantıları için bkz. [IBM MQ for z/OS Program Dizini PDF dosyaları](#).



**Not:** Yaşam Çevrimi Projesi Yönetimi için veri kümesi, sürüme özgüdür. Sistemde varolan bir Yaşam Çevrimi Projesi Yönetimi (LPA) kullanıyorsanız, hangi Yaşam Çevrimi Projesi Yönetimi 'nin kullanılacağına karar vermek için sistem denetimcinize başvurun.

## Erken kod

IBM MQ 'in altsistem olarak işlev görmesi için bazı IBM MQ yükleme modüllerinin MVS ' e eklenmesi gerekir. Bu modüller Erken kod olarak bilinir ve bir kuyruk yöneticisi etkin olmasa da yürütülebilir. Örneğin, konsolda IBM MQ komut örneğiyle bir işletmen komutu verildiğinde, bu Erken kod denetimi alır ve bir kuyruk yöneticisi başlatması gerekip gerekmediğini ya da isteği çalışan bir kuyruk yöneticisine geçirip iletmediğini denetler. Bu kod, Bağlantı Paketi Alanı 'na (LPA) yüklenir. Tüm kuyruk yöneticileri için kullanılan bir Erken Dönem modülleri kümesi vardır ve bunların en yüksek IBM MQ düzeyinde olması gerekir. Daha yüksek bir IBM MQ sürümündeki ilk kod, daha düşük bir IBM MQ sürümüne sahip bir kuyruk yöneticisiyle çalışacaktır, ancak tersi çalışmayacaktır.

Erken kod aşağıdaki yükleme modüllerinden oluşur:

- CSQ3INI ve CSQ3EPX ( thqual.SCSQLINK kitaplığında)
- CSQ3ECMX kitaplıkta thqual.SCSQSNL x; burada x , dil harfinizi ifade eden bir harftir:
  - thlqual.SCSQSNLE, ABD İngilizcesi için karışık büyük/küçük harf
  - thlqual.SCSQSNLU, ABD İngilizcesi için büyük harf
  - thlqual.SCSQSNLK, Japonca için
  - thlqual.SCSQSNLF, Fransızca için
  - thlqual.SCSQSNLC, Çince için

IBM MQ , thqual.SCSQSNL i kitaplığının içeriğini thqual.SCSQLINK içine taşıyan ve SMP/E bilgilerini veren bir kullanıcı değişikliği içerir. Bu kullanıcı değişikliği CSQ8UERL olarak adlandırılır ve *IBM MQ for z/OS için Program Dizini* içinde Long Term Support ya da Continuous Delivery için açıklanır. Program Dizinine ilişkin yükleme bağlantıları için bkz. [IBM MQ for z/OS Program Dizini PDF dosyaları](#).

LPA kitaplıklarındaki erken dönem kodunu güncellediğinizde, bu kod sonraki z/OS IPL (CLPA seçeneğiyle) ile birlikte, SYS1.PARMLIB.

Daha sonra eklenen yeni kuyruk yöneticisi altsistemi için IPL olmadan hemen kullanılabilir kılabilir (açıklamalar: gelen: “IBM MQ altsistemini şu şekilde tanımlayın: z/OS” sayfa 889 ) Aşağıdaki şekilde Yaşam Çevrimi Projesi Yönetimi Yaşam Çevrimi Projesi Yönetimi olanağına ekleyerek:

- CSQ8UERL kullanmadıysanız, şu z/OS komutlarını verin:

```
SETPROG LPA,ADD,MODNAME=(CSQ3INI,CSQ3EPX),DSNAME=thqual.SCSQLINK
SETPROG LPA,ADD,MODNAME=(CSQ3ECMX),DSNAME=thqual.SCSQSNL x
```

- CSQ8UERL kullandıysanız, aşağıdaki z/OS komutunu kullanarak erken kodu LPA ' ya yükleyebilirsiniz:

```
SETPROG LPA,ADD,MASK=*,DSNAME=thqual.SCSQLINK
```

- Advanced Message Security kullanıyorsanız, Yaşam Çevrimi Projesi Yönetimi 'nde ek bir modül eklemek için aşağıdaki z/OS komutunu da vermeniz gerekir:

```
SETPROG LPA,ADD,MODNAME=(CSQ0DRTM),DSNAME=thqual.SCSQLINK
```

Bakım uyguladıysanız ya da IBM MQ' un daha sonraki bir sürümüyle ya da yayın düzeyiyle bir kuyruk yöneticisini yeniden başlatmak istiyorsanız, erken kod, aşağıdaki adımları kullanarak var olan kuyruk yöneticilerinin kullanımına sunulabilir. Bu adımları gerçekleştirmediğiniz kuyruk yöneticileri, kullanmakta oldukları erken kod sürümünü kullanmaya devam eder. Özellikle tümüne bakım uygulamayı ya da bunları IBM MQ' un daha yeni bir sürümüne ya da yayınına güncellemeyi denemediğiniz sürece, LPAR' daki tüm kuyruk yöneticileri için bu adımları gerçekleştirmeniz gerekmez.

1. Bu konuyu, bu konuda daha önce açıklandığı gibi z/OS SETPROG komutlarını kullanarak Yaşam Çevrimi Projesi Yönetimi Yaşam Çevrimi Projesi Yönetimi bölümüne ekleyin.
2. STOP QMGR IBM MQ komutunu kullanarak kuyruk yöneticisini durdurun.
3. qmgr.REFRESH.QMGR güvenlik profili ayarlandı. Bkz. [MQSC komutları, tanımları ve erişim düzeyleri](#).
4. IBM MQ REFRESH QMGR TYPE (EARLY) komutunu kullanarak kuyruk yöneticisine ilişkin erken kodu yenileyin.
5. START QMGR IBM MQ komutunu kullanarak kuyruk yöneticisini yeniden başlatın.

IBM MQ komutları STOP QMGR, REFRESH QMGR ve START QMGR, [MQSC komutlarında](#) açıklanmıştır.

## Diğer kod

Aşağıdaki kitaplıklarda IBM MQ tarafından sağlanan tüm yükleme modülleri yeniden girilir ve Yaşam Çevrimi Projesi Yönetimi (LPA) içine yerleştirilebilir:

- SCSQAUTH
- SCSQANL x; burada x , dil harfinizdir.
- SCSQMVR1

**Önemli:** Ancak, kitaplıkları Yaşam Çevrimi Projesi Yönetimi 'nde (LPA) yerleştirirseniz, bakım uyguladığınızda, değiştirilen modülleri Yaşam Çevrimi Projesi Yönetimi (LPA) içine el ile kopyalamanız gerekir. Bu nedenle, IBM MQ yükleme kitaplıklarının bağlantı listesine konması tercih edilir; bunlar bakım yapıldıktan sonra z/OS MODIFY LLA REFRESH komutu verilerek güncellenebilir.

Daha fazla bilgi için [LNKLST veri kümelerinin içeriğinin değiştirilmesi](#) ve [Dinamik LNKLST olanağının güvenli ve düzgün bir şekilde kullanılmasına](#) başlıklı konuya bakın.

Bu özellikle SCSQAUTH için önerilir; böylece, bu değeri birkaç STEPLIB' ye dahil etmek zorunda kalmayabilirsiniz. LPA ya da bağlantı listesine tek bir dil kitaplığı (SCSQANL x ) yerleştirilmelidir. Bağlantı listesi kitaplıkları, SYS1.PARMLIB.

Dağıtılmış kuyruğa alma olanağı ve CICS bridge (kuyruk yöneticisinin kendisi değil), SCEERUN (Dil Ortamı) yürütme ortamı kitaplığına erişmelidir. Bu olanaklardan birini kullanıyorsanız, bağlantı listesine SCEERUN eklemelisiniz.

**V 9.3.2** Bazı modüller, kuyruk yöneticisinin ECSA ' ya başlatılması sırasında yüklenir. ECSA kısıtlı ortamlarda, bu modüller LPA ' ya yerleştirilebilir. Ek bilgi için bkz. ["IBM MQ genel modüllerinin LPA ' ya yerleştirilmesi"](#) sayfa 886 .

**Önemli:** **LTS** IBM MQ 9.3 adresindeki bu olanağı kullanmak için APAR PH52358 uygulamanız gerekir.

## İlgili kavramlar

["z/OS program özellikleri çizelgesinin güncellenmesi"](#) sayfa 888

IBM MQ kuyruk yöneticisi için bazı ek PPT girişleri gerekir.

**V 9.3.2** **z/OS** *IBM MQ genel modüllerinin LPA ' ya yerleştirilmesi*


Bir IBM MQ for z/OS kuyruk yöneticisi başlatıldığında, bazı yükleme modüllerini (genel modüller) genişletilmiş ortak hizmet alanına (ECSA) yükler. Kuyruk yöneticisi-sona erdirildiği sırada ECSA serbest bırakıldı.

IBM MQ 9.3 adresinde, çalışan her kuyruk yöneticisi için yaklaşık 1.2 MB ECSA tüketen 19 genel modül vardır.

**Not:** CSQ7GPLM genel bir modül olsa da, Yaşam Çevrimi Projesi Yönetimi (LPA) olanağına eklenmemelidir.

Her LPAR için birden çok kuyruk yöneticisi çalıştıran ve ECSA ya da yüksek özel kısıtlamalar nedeniyle ECSA tüketiminde bir azalma gerektiren ortamlarda, genel modülleri Yaşam Çevrimi Projesi Yönetimi (LPA) içine yerleştirmek mümkündür. IBM MQ genel modüllerinin Yaşam Çevrimi Projesi Yönetimi (LPA) içine

yerleştirilmesi, bakım gerektiren el ile gerçekleştirilen bir işlemdir; bu nedenle, bu yordamı yalnızca ECSA ya da yüksek özel kısıtlamalara yanıt vermeniz gerektiğinde gerçekleştirmeniz gerekir.

**Önemli:**  IBM MQ 9.3 adresindeki bu olanağı kullanmak için APAR PH52358 uygulamanız gerekir.

Kuyruk yöneticisi STEPLIB 'de genel bir modül bulamazsa ve modülün LPA' da olduğunu saptarsa, modülün bir kopyasını ECSA ' ya yüklemek yerine, doğrudan LPA kopyasını kullanır. Diğer bir seçenek olarak, kuyruk yöneticileri kodu bağlantı listesinden yüklenirse, bağlantı listesindeki herhangi bir genel modüle tercih etmek için Yaşam Çevrimi Projesi Yönetimi içindeki tüm genel modüller yüklenir.

z/OS ortak depolama izleme işlevi (bkz. [Ortak depolama izleme işlevinin kullanılması](#)), her kuyruk yöneticisi için her kuyruk yöneticisinin MSTR adres alanı altındaki depolamayı izler ve genel modüller tarafından ne kadar yer kullanıldığını saptamak için kullanılabilir.

Varsayılan olarak, genel modüller SCSQAUTH yükleme kitaplığınca bulunur. Bir kuyruk yöneticisinin MSTR adres alanı, STEPLIB bitişirmesi yoluyla SCSQAUTH 'yi buluyorsa, genel modüller LPA' daki herhangi biri yerine kullanılır ve ECSA ' ya yüklenir.

Genel modüller şunlardır:

CSQ0GPLM, CSQ3AMGP, CSQ3SSGP, CSQ9PREP,  
CSQ9SCNB, CSQGGPLM, CSQMCGLM, CSQMGPLM, CSQRGLM1,  
CSQSLD1, CSQVGEPL, CSQVSRX, CSQWDL2, CSQWDL3,  
CSQWVZSA, CSQWZDGO, CSQWVZPS, CSQWVGTM, CSQZTDDM

### Önemli:

- IBM MQ için genel modüllerin adı, farklı IBM MQ sürümlerinde sabit kalır. Bu nedenle, genel modülleri Yaşam Çevrimi Projesi Yönetimi (LPA) içine yüklüyorsanız, bunlar tek bir IBM MQ sürümünden olmalıdır ve yalnızca aynı IBM MQ sürümünde çalışan kuyruk yöneticileri tarafından kullanılmalıdır.
- Aynı LPAR üzerinde birden çok IBM MQ sürümü çalıştırılırsa, herhangi bir zamanda bu sürümlerden yalnızca birinin genel modülleri Yaşam Çevrimi Projesi Yönetimi 'nde bulunabilir.
- Bakım, Yaşam Çevrimi Projesi Yönetimi (LPA) içine yüklenen genel modülleri olan bir IBM MQ kuruluşuna uygulanırsa ve bu bakım, genel modüllerden herhangi birini güncellerse, aşağıdaki metinde açıklanan yordamı yeniden gerçekleştirmeniz gerekir.

### Yordam

IBM MQ sürümündeki genel modülleri Yaşam Çevrimi Projesi Yönetimi (LPA) içine koymak için aşağıdaki adımları gerçekleştirin:

1. th1qua1 . SCSQAUTH yükleme kitaplığının ve içeriğinin bir kopyasını oluşturun, örneğin: th1qua1 . LOCAL . SCSQAUTH. Bu yükleme kitaplığının, dış güvenlik yöneticinizi (ESM) kullanarak yetkisiz erişimden korunduğundan emin olun.
2. APF, th1qua1 . LOCAL . SCSQAUTH yükleme kitaplığını yetkilendirir; bkz. [“APF IBM MQ yükleme kitaplıklarını yetkilendirir”](#) sayfa 883.
3. th1qua1 . LOCAL . SCSQAUTH ile aynı özniteliklere sahip yeni bir th1qua1 . GLOBAL . SCSQAUTH yükleme kitaplığı oluşturun.

**Not:** Bu yükleme kitaplığının APF yetkisi olması gerekmez. Bu yükleme kitaplığının ESM ' nizi kullanarak yetkisiz erişimden korunduğundan emin olun.

4. 19 genel modülü th1qua1 . LOCAL . SCSQAUTH içinden th1qua1 . GLOBAL . SCSQAUTH içine kopyalayın.
5. 19 genel modülü th1qua1 . LOCAL . SCSQAUTH içinden silin.
6. 19 genel modülü th1qua1 . GLOBAL . SCSQAUTH içinden Yaşam Çevrimi Projesi Yönetimi Yaşam Çevrimi Projesi Yönetimi (LPA) içine aşağıdaki seçeneklerden birini kullanarak yerleştirin:
  - a. a. SYS1.PARMLIB ögesinin LPALSTxx üyesine th1qua1 . GLOBAL . SCSQAUTH eklenmesi. Daha sonra, kitaplık içeriğinin PLPA ' ya yüklendiğinden emin olmak için sistemi CLPA seçeneğiyle IPL işlemi gerçekleştirmeniz gerekir.

b. Aşağıdaki komutu kullanarak modülleri LPA ' ya dinamik olarak ekleme:

```
SETPROG  
LPA,ADD,MODNAME=(CSQ0GPLM,CSQ3AMGP,CSQ3SSGP,CSQ9PREP,CSQ9SCNB,CSQGGLM,  
CSQMCGLM,CSQMGPLM,CSQRGLM1,CSQSLD1,CSQVGEPL,CSQVSRX,CSQWDL2,CSQWDL3,  
CSQWVZSA,CSQWZDGO,CSQWVZPS,CSQWVGTM,CSQZTDDM),DSNAME= thlqua1.GLOBAL.SCSQAUTH
```

**Not:** LPA LSTxx , modülleri Yaşam Çevrimi Projesi Yönetimi (LPA) içine yerleştirmenin tercih edilen uzun vadeli aracıdır.

7. Aşağıdaki komutu vererek modüllerin Yaşam Çevrimi Projesi Yönetimi 'nde olduğunu doğrulayın:

```
D PROG,LPA,MODNAME=CSQMCGLM
```

Komutun çıkışı, modülün giriş ve yükleme noktalarını (LPA ' ya başarıyla yüklendiye) belirtmelidir.

LPA 'daki genel modülleri kullanması gereken her kuyruk yöneticisi için, olağan durumda şunları yaparsanız:

1. thlqua1.SCSQAUTH bağlantı listesinde, kuyruk yöneticinizi durdurun da ürününe başlıyoruz). Genel modüller, Yaşam Çevrimi Projesi Yönetimi 'nden ve bağlantı listesinden yerel modüller yüklenir.
2. thlqua1.SCSQAUTH MSTR JCL STEPLIB içinde, JCL ' yi thlqua1.SCSQAUTHyerine thlqua1.LOCAL.SCSQAUTH kullanacak şekilde değiştirin. Kuyruk yöneticisini durdurun ve başlatın; genel modüller LPA 'dan ve STEPLIB' den yerel modüller yüklenir.

CHIN ve AM SM JCL, thlqua1.SCSQAUTH ' i IBM MQ uygulamaları gibi kullanmaya devam edebilir.

Kuyruk yöneticisini ECSA ' ya genel modülleri yüklemeye geri döndürmek için aşağıdaki adımları izleyin:

1. Kuyruk yöneticilerini durdur
2. LPA LSTxx tanımlarını kaldırarak ya da aşağıdaki komutu kullanarak genel modülleri sonraki IPL ' de kaldırın:

```
SETPROG LPA,DELETE,MODNAME=(xxx) FORCE=YES
```

3. thlqua1.LOCAL.SCSQAUTH kuyruk yöneticisinin STEPLIB ' indeyse, bunu thlqua1.SCSQAUTHile değiştirin.

4. Kuyruk yöneticilerini yeniden başlatın.

### İlgili kavramlar

[“z/OS bağlantı listesini ve Yaşam Çevrimi Projesi Yönetimi” sayfa 884](#)

LPA kitaplıklarını, erken kod kitaplıklarının yeni sürümüyle güncelleyin. Diğer kod bağlantı listesine ya da Yaşam Çevrimi Projesi Yönetimi listesine girebiliyor.

### z/OS program özellikleri çizelgesinin güncellenmesi

IBM MQ kuyruk yöneticisi için bazı ek PPT girişleri gerekir.

- Bu görevi, çalıştırmak istediğiniz her z/OS sistemi için bir kez gerçekleştirmeniz gerekir IBM MQ.
- Kuyruk paylaşım grupları kullanıyorsanız, sistem şebekesinde her z/OS sisteminde IBM MQ ayarlarının aynı olduğundan emin olmanız gerekir.
- Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekmez.
- Advanced Message Securitygerekirken bu görevin CSQ0DSRV kısmını gerçekleştirmeniz gerekir.

Gerekli tüm PPT girişlerini içeren bir örnek thlqua1.SCSQPROC(CSQ4SCHED) içinde sağlanır. Gerekli girişlerin, SYS1.PARMLIB(SCHEDxx).

z/OSiçinde, CSQYASCP, ayrıntılı özniteliklerle işletim sistemi için önceden tanımlanmıştır ve artık PARMLIB ' in bir SCHEDxx üyesine eklenmesi gerekmemektedir.

IBM MQ kuyruk yöneticisi kendi deęiş tokuşu denetler. Ancak, yüklü bir IBM MQ aęınız varsa ve yanıt süresi önemiye, CSQXJST PPT girişini z/OS sisteminizin geri kalanının başarımını etkileme riski altında ekleyerek IBM MQ kanal başlatıcısını deęiştirilemeyen duruma getirmek avantajlı olabilir.

Advanced Message Security gerekiyorsa, CSQODSRV PPT girişini ekleyin.

Bu değişikliklerin yürürlüğe girmesi için z/OS komutu **SET SCH=xx** verin; burada xx , PARMLIB ' in SCHEDxx üyesinin son ekidir.

### İlgili kavramlar

“IBM MQ altsistemini şu şekilde tanımlayın: z/OS” sayfa 889

Altsistem ad çizelgesini güncelleyin ve komut öneki dizgilerine ilişkin bir kurala karar verin.

### **z/OS** Kuyruk yöneticisinin ve kanal başlatıcısının yapılandırılması

Kuyruk yöneticisini ve kanal başlatıcısını yapılandırmak için adım adım bu konuları kullanın.

### **z/OS** IBM MQ altsistemini şu şekilde tanımlayın: z/OS

Altsistem ad çizelgesini güncelleyin ve komut öneki dizgilerine ilişkin bir kurala karar verin.

Bu görevi her IBM MQ kuyruk yöneticisi için yineleyin. Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekmez.

### İlgili kavramlar

“IBM MQ kuyruk yöneticisi için yordam yaratılması” sayfa 892

Her IBM MQ altsistemini kuyruk yöneticisini başlatmak için kataloğa alınmış bir yordam gerekir. Kendi yordam kitaplığınızı yaratabilir ya da IBM tarafından sağlanan yordam kitaplığını kullanabilirsiniz.

### **z/OS** Altsistem ad çizelgesinin güncellenmesi

IBM MQ altsistemini tanımlarken, altsistem adı çizelgesine bir giriş eklemeniz gerekir.

Başlangıçta SYS1.PARMLIB üyesi IEFSSNss 'ten alınan z/OS altsistem adı çizelgesi, resmi olarak tanımlanmış z/OS altsistemlerinin tanımlamalarını içerir. Her IBM MQ altsistemini tanımlamak için, SYS1.PARMLIB ya da tercihen, z/OS komutu SETSSI kullanılarak.

IBM MQ altsistem kullanıma hazırlama, koşut işlemeyi destekler; böylece IBM MQ altsistem tanımlama deyimleri, z/OS V1.12 ve sonraki yayın düzeylerinde bulunan IEFSSNss çizelgesindeki BEGINPARALLEL anahtar sözcüğünün üstüne ve altına eklenebilir.

SETSSI komutunu kullanırsanız, değişiklik hemen yürürlüğe girer ve sisteminizin IPL işlemini gerçekleştirmenize gerek kalmaz. SYS1.PARMLIB de, “SYS1.PARMLIB üyeleri” sayfa 896 içinde açıklandığı gibi, değişikliklerin sonraki IPL 'lerden sonra yürürlükte kalması için.

Bir IBM MQ altsistemini dinamik olarak tanımlamak için SETSSI komutu:

```
SETSSI ADD,S=ssid,I=CSQ3INI,P='CSQ3EPX,cpf,scope'
```

IEFSSNss içindeki ilgili bilgiler aşağıdaki iki yoldan biriyle belirtilebilir:

- IEFSSNss içindeki IBM MQ altsistem tanımlamasının anahtar sözcük değiştirgeci biçimi. Bu, önerilen yöntemdir.

```
SUBSYS SUBNAME(ssid) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,cpf,scope')
```

- IBM MQ altsistem tanımlamasının konumlu parametre biçimi.

```
ssid,CSQ3INI,'CSQ3EPX,cpf,scope'
```

İki formu bir IEFSSNss üyesinde karışık kullanmayın. Farklı formlar gerekiyorsa, yeni üyenin SSN işleneni IEASYSpp SYS1.PARMLIB üyesi. Birden çok SSN belirtmek için SSN = (aa, bb, ...) kullanın (IEASYSpp içinde).

Örneklerde,

### ssid

Altsistem tanıtıcısı. En çok dört karakter uzunluğunda olabilir. Tüm karakterler alfasayısal olmalıdır (büyük harf A-Z, 0-9), alfabetik bir karakterle başlamalıdır. Kuyruk yöneticisi altsistemle aynı ada sahip olur; bu nedenle, yalnızca z/OS altsistem adları ve IBM MQ nesne adları için izin verilen karakterleri kullanabilirsiniz.

### cpf

Komut öneki dizgisi (CPF ' lerle ilgili bilgi için bkz. [“Komut öneki dizgilerinin \(CPF\) tanımlanması” sayfa 890](#)).

### scope

Bir z/OS sistem şebekesinde çalışıyorsanız kullanılan sistem kapsamı (sistem kapsamına ilişkin bilgi için bkz. [“Sistem birleşimi \(sysplex\) ortamındaki CPF ' ler” sayfa 891](#)).

Şekil 100 sayfa 890 içinde birkaç IEFSSNss deyim örneği gösterilmektedir.

```
CSQ1,CSQ3INI, 'CSQ3EPX,+mqs1cpf,S'  
CSQ2,CSQ3INI, 'CSQ3EPX,+mqs2cpf,S'  
CSQ3,CSQ3INI, 'CSQ3EPX,++,S'
```

Şekil 100. Altsistemleri tanımlamak için örnek IEFSSNss deyimleri

**Not:** Altsistemde nesnelere yarattığınızda, altsistem adını değiştiremez ya da sayfa kümelerini başka bir altsistemdeki bir altsistemden kullanamazsınız. Bunlardan birini yapmak için, bir altsistemdeki tüm nesnelere ve iletileri boşa alıp başka bir altsisteme yeniden yüklemeniz gerekir.

Çizelge 54 sayfa 890 içinde, Şekil 100 sayfa 890 içindeki deyimler tarafından tanımlandığı şekilde, altsistem adları ve komut öneki dizgileri (CPF) ilişkilendirmelerini gösteren bir dizi örnek verilir.

IBM MQ altsistem adı	CPF
CSQ1	+mqs1cpf
CSQ2	+mqs2cpf
CSQ3	++

**Not:** SETSSI z/OS komutunun ACTIVATE ve DEACTIVATE işlevleri IBM MQ tarafından desteklenmez.

Değişikliklerin durumunu denetlemek için SDSF' de şu komutu verin: /D SSI, L. ETKİN durumuyla yaratılan yeni altsistemleri görürsünüz.

### z/OS Komut öneki dizgilerinin (CPF) tanımlanması

IBM MQ ' in her altsistem örneğinin, o altsistemi tanıttak bir komut öneki dizgisi olabilir.

Çakışmaları önlemek için tüm altsistemlere ilişkin CPF ' leriniz için sistem çapında bir kural benimseyin. Aşağıdaki yönergelere uyun:

- Bir CPF ' yi en çok sekiz karakterden oluşan bir dizgi olarak tanımlayın.
- Başka bir altsistem tarafından kullanılmakta olan bir CPF kullanmayın ve diziliminizin ilk karakteri olarak sisteminizde tanımlanan JES geri alma karakterini kullanmaktan kaçının.
- Çizelge 56 sayfa 891 içinde listelenen geçerli karakter kümesindeki karakterleri kullanarak CPF ' yi tanımlayın.

- Önceden tanımlanmış bir işlemin kısaltması olan ya da komut sözdizimiyle karıştırılmış olabilecek bir CPF kullanmayın. Örneğin, ' D' gibi bir CPF, DISPLAY gibi z/OS komutlarıyla çakışıyor. Bunun olmasını önlemek için özel karakterlerden birini kullanın ( Çizelge 56 sayfa 891 içinde gösterilir) ilk ya da tek karakter olarak.
- Var olan bir CPF ' nin altkümesi ya da üst kümesi olan bir CPF tanımlamayın. Örneğin, bkz. Çizelge 55 sayfa 891.

Çizelge 55. CPF altkümesi ve üst küme kuralları örneği		
Altsistem adı	CPF tanımı	Yöneltilen komutlar
MQA	!A	MQA
MQB	!B	MQB
MQC1	!C1	MQC1
MQC2	!C2	MQC2
MQB1	!B1	MQB

MQB1 altsistemine yönelik komutlar (CPF!B1), bu altsisteme ilişkin CPF!B, bir alt küme!B1. Örneğin, komutu girdiyeniz:

```
!B1 START QMGR
```

altsistem MQB şu komutu alır:

```
1 START QMGR
```

(bu durumda, bununla başa çıkamaz).

z/OS komutunu DISPLAY OPDATA komutunu vererek hangi öneklerin var olduğunu görebilirsiniz.

Bir sistem şebekesinde çalışıyorsanız, z/OS CPF kaydı sırasında bu tipteki çakışmaları tanımlar (CPF kaydına ilişkin bilgi için bkz. "Sistem birleşimi (sysplex) ortamındaki CPF ' ler" sayfa 891 ).

Çizelge 56 sayfa 891 içinde CPF dizgilerinizi tanımlarken kullanabileceğiniz karakterler gösterilmektedir:

Çizelge 56. CPF dizilimleri için geçerli karakter takımı	
Karakter kümesi	İçerik
Alfabetik	Büyük harf A-Z, küçük harf a-z
Sayısal	0 - 9
Ulusal (bkz. not)	@ \$# (Onaltılı değerler olarak gösterilebilen karakterler)
Özel	. [] * & + - = <   ! ; % _ ? : >

#### Not:

Sistem, ulusal karakterlerin şu onaltılı gösterimlerini tanımlar: @, X'7C', \$as X'5B' ve #, X'7B'. U.S.dışındaki ülkelerde, U.S. uçbirim klavyelerinde gösterilen ulusal karakterler farklı bir onaltılı gösterim oluşturabilir ve hataya neden olabilir. Örneğin, bazı ülkelerde \$karakterini bir X'4A' oluşturabilir.

Noktalı virgül (;) bir CPF olarak geçerlidir, ancak çoğu sistemde bu karakter komut sınırlayıcısıdır.

**z/OS** Sistem birleşimi (sysplex) ortamındaki CPF ' ler

Bir sistem şebekesi kapsamında CPF ' lerin nasıl kullanılacağını anlamak için bu konuyu kullanın.



Bir sysplex ortamında kullanılıyorsa, IBM MQ , sistem şebekesinde herhangi bir konsoldan bir komut girmenizi ve bu komutu yürütme için uygun sisteme yönlendirmenizi sağlamak üzere CPF ' lerinizi kaydeder. Komut yanıtları, kaynak konsola döndürülür.

## **Sysplex işlemleri için kapsamı tanımlama**

Kapsam, bir sistem şebekesi ortamında IBM MQ çalıştırılırken IBM MQ altsistemi tarafından gerçekleştirilen CPF kaydının tipini belirlemek için kullanılır.

Kapsam için olası değerler şunlardır:

### **M**

Sistem kapsamı.

CPF, sistem IPL sırasında IBM MQ tarafından z/OS ' e kaydedilir ve z/OS sisteminin etkin olduğu süre boyunca kayıtlı kalır.

IBM MQ komutları, hedef altsistemi çalıştıran z/OS görüntüsüne bağlı bir konsoldan girilmeli ya da komutu o görüntüye yönlendirmek için ROUTE komutlarını kullanmalısınız.

Bir sistem şebekesinde çalışmıyorsanız bu seçeneği kullanın.

### **S**

Sysplex kapsamı başlattı.

CPF, IBM MQ altsistemi başlatıldığında z/OS ' a kaydedilir ve IBM MQ altsistemi sona erinceye kadar etkin kalır.

Özgün START QMGR komutunu hedef sisteme yönlendirmek için ROUTE komutlarını kullanmanız gerekir, ancak diğer tüm IBM MQ komutları sistem şebekesine bağlı herhangi bir konsoldan girilebilir ve hedef sisteme otomatik olarak yönlendirilir.

IBM MQ sonlandırmasından sonra, sonraki START komutlarını hedef IBM MQ altsistemine yönlendirmek için ROUTE komutlarını kullanmanız gerekir.

### **X**

Sysplex IPL kapsamı.

CPF, sistem IPL sırasında IBM MQ tarafından z/OS ' e kaydedilir ve z/OS sisteminin etkin olduğu süre boyunca kayıtlı kalır.

IBM MQ komutları, sistem şebekesine bağlı herhangi bir konsoldan girilebilir ve hedef sistemi otomatik olarak yürüten görüntüye yönlendirilir.

S kapsamına sahip bir IBM MQ altsistemi, bir sistem şebekesindeki bir ya da daha çok z/OS görüntüde tanımlanabilir; böylece, bu görüntüler tek bir altsistem adı çizelgesini paylaşabilir. Ancak, IBM MQ altsisteminin çalışmasını istediğiniz z/OS görüntüsünde ilk START komutunun verildiğinden (ya da yöneltildiğinden) emin olmanız gerekir. Bu seçeneği kullanırsanız, altsistem adı çizelgesini değiştirmek ya da z/OS sisteminin IPL 'sini gerçekleştirmek zorunda kalmadan, IBM MQ altsistemini durdurabilir ve sysplex içindeki farklı bir z/OS görüntüsünde yeniden başlatabilirsiniz.

Kapsamı X olan bir IBM MQ altsistemi, bir sistem birleşimi (sysplex) içindeki tek bir z/OS görüntüsünde tanımlanabilir. Bu seçeneği kullanırsanız, X kapsamına sahip CPF ' leri olan IBM MQ altsistemlerini gerektiren her z/OS görüntüsü için benzersiz bir altsistem adı çizelgesi tanımlamanız gerekir.

Kuyruk yöneticilerini farklı z/OS görüntülerinde otomatik olarak yeniden başlatmak için z/OS otomatik yeniden başlatma yöneticisini (ARM) kullanmak istiyorsanız, kuyruk yöneticisinin yeniden başlatılabileceği her z/OS görüntüsünde her kuyruk yöneticisi tanımlanmalıdır. Her kuyruk yöneticisi, sysplex-wide, benzersiz 4 karakterli bir altsistem adıyla (CPF kapsamı S) tanımlanmalıdır.

## **z/OS IBM MQ kuyruk yöneticisi için yordam yaratılması**

Her IBM MQ altsisteminin kuyruk yöneticisini başlatmak için kataloğa alınmış bir yordam gerekir. Kendi yordam kitaplığınızı yaratabilir ya da IBM tarafından sağlanan yordam kitaplığını kullanabilirsiniz.



- Bu görevi her IBM MQ kuyruk yöneticisi için yineleyin.
- Önceki bir sürümden geçiş yaparken kataloğa alınmış yordamı değiştirmeniz gerekebilir.

Altsistem ad çizelgesinde tanımlanan her IBM MQ altsistemi için, kuyruk yöneticisini başlatmak üzere bir yordam kitaplığında kataloğa alınmış bir yordam yaratın. IBM tarafından sağlanan yordam kitaplığına SYS1.PROCLIB, ancak kuruluşunuz kendi adlandırma kuralını kullanabilir.

Kuyruk yöneticisi tarafından başlatılan görev yordamının adı, altsistem adının MSTR karakterleriyle birleştirilmesiyle oluşturulur. Örneğin, CSQ1 altsisteminde CSQ1MSTR yordam adı vardır. Tanımladığınız her altsistem için bir yordam gerekir.

İletileri içeren kitaplığı seçtiğiniz dilde eklemeniz gerekir:

- thlqual.SCSQSNLE, ABD İngilizcesi için karışık büyük/küçük harf
- thlqual.SCSQSNLU, ABD İngilizcesi için büyük harf
- thlqual.SCSQSNLK, Japonca için
- Fransızca için thlqual.SCSQSNLF
- thlqual.SCSQSNLC, Çince için

Bu ürün belgelerindeki birçok örnek ve yönerge, CSQ1 adlı bir altsisteminiz olduğunu varsayar. Başlangıçta kuruluş doğrulama ve sınama amacıyla CSQ1 adlı bir altsistem yaratıldıysa, bu örneklerin kullanılması daha kolay olabilir.

thlqual.SCSQPROCI içinde iki örnek başlatılan görev yordamı vardır. CSQ4MSTR üyesi, her ileti sınıfı için bir sayfa kümesi kullanır; CSQ4MSRR üyesi, iletinin ana sınıfları için birden çok sayfa kümesi kullanır. Bu yordamlardan birini SYS1.PROCLIB ürününüzdeki xxxxMSTR (burada xxxx, IBM MQ altsisteminizin adıdır) üyesine ya da SYS1.PROCLIB kullanmıyorsanız, yordam kitaplığınıza kopyalayın. Örnek yordamı, tanımladığınız her IBM MQ altsistemi için yordam kitaplığınızdaki bir üyeye kopyalayın.

Üyeleri kopyaladığınızda, üyelerinizi, üyelerdeki yönergeleri kullanarak her bir altsistemin gereksinimlerine göre uyarlayabilirsiniz. Kuyruk yöneticisi tarafından kullanılan depolama sınırlarını belirtme hakkında bilgi için bkz. [Depolama yapılandırması](#). Yordamın başlatıldığında değiştirilmesine izin vermek için JCL ' de sembolik parametreleri de kullanabilirsiniz. Birden çok IBM MQ altsisteminiz varsa, gelecekteki bakımı kolaylaştırmak için yordamın ortak kısımları için JCL içerme gruplarını kullanmayı avantajlı bulabilirsiniz.

Kuyruk paylaşım gruplarını kullanıyorsanız, STEPLIB birleştirmesi Db2 yürütme ortamı hedef kitaplığı SDSNLOAD ' u içermelidir ve bu kitaplık APF-yetkili olmalıdır. Bu kitaplık yalnızca, bağlantı listesinden ya da Yaşam Çevrimi Projesi Yönetimi 'nden (LPA) erişilemiyorsa, STEPLIB birleştirmesinde gereklidir.

#### Notlar:

1. Önyükleme veri kümenizin (BSDS) adlarını, günlüklerini ve sayfa kümelerini JCL ' de kullanmak üzere not edebilir ve daha sonra bu kümeleri işlemin sonraki bir adımında tanımlayabilirsiniz.
2. CSQ4MSTR ve CSQ4MSRR örnek başlatılan görev yordamları, iletim güvenliği (SSL ya da TLS özellikleri) içeren bir QMINI veri kümesini tanımlamak için kullanılacak CSQMINI DD kartını içerecek şekilde güncellendi, ancak bu kartta açıklama yapılmadı.

TLS 1.3 desteğini etkinleştirmek ya da devre dışı bırakmak için “QMINI veri kümesi” [sayfa 899](#) özelliğini kullanabilir ve/veya kanallar tarafından kullanılacak özel bir CipherSpecs listesi tanımlamak için kullanabilirsiniz.

#### İlgili kavramlar

[“Kanal başlatıcısı için yordamlar oluştur” sayfa 893](#)

Her IBM MQ altsistemi için bir CSQ4CHINKopyasını uyarlayabilirsiniz. Kullandığınız diğer ürünlere bağlı olarak, diğer veri kümelerine erişime izin vermeniz gerekebilir.

#### **Kanal başlatıcısı için yordamlar oluştur**

Her IBM MQ altsistemi için bir CSQ4CHINKopyasını uyarlayabilirsiniz. Kullandığınız diğer ürünlere bağlı olarak, diğer veri kümelerine erişime izin vermeniz gerekebilir.

- Bu görevi her IBM MQ kuyruk yöneticisi için yineleyin.
- Önceki bir sürümden geçiş yaparken kataloğa alınmış yordamı değiştirmeniz gerekebilir.

Dağıtım kuyruğa alma özelliğini kullanacak her IBM MQ altsistemi için kanal başlatıcı tarafından başlatılan bir görev yordamı yaratmanız gerekir.

Bunu yapmak için:

1. thlqual.SCSQPROC(CSQ4CHIN) adlı örnek başlatılan görev yordamını yordam kitaplığınıza kopyalayın. *xxxx* CHIN yordamını adlandırın; burada *xxxx* , IBM MQ altsisteminizin adıdır (örneğin, CSQ1CHIN , CSQ1kuyruk yöneticisi için kanal başlatıcı tarafından başlatılan görev yordamı olur).
2. Kullanacağınız her IBM MQ altsistemi için bir kopya oluşturun.
3. CSQ4CHINörnek yordamındaki yönergeleri kullanarak yordamları gereksinimlerinize göre uyarlayın. Yordamın başlatıldığında değiştirilmesine izin vermek için JCL ' de sembolik parametreleri de kullanabilirsiniz. Bu, Yönetme IBM MQ for z/OS başlıklı konudaki başlangıç seçenekleriyle açıklanır.

Dağıtılmış kuyruğa alma kitaplığını ( thlqual.SCSQMVR1) birleştirin.

Bağlantı listenizde (SYS1.PARMLIB(LNKLSTkk), STEPLIB DD deyiminde bunu birleştirin.

**V9.3.1**

Depolama yapılandırması içindeki bilgileri kullanarak MEMLIMIT parametresini ayarlayın.

4. Dış güvenlik yöneticiniz altında çalışmak üzere yordamları yetkilendirin.
5. İletileri içeren kitaplığı seçtiğiniz dilde eklemeniz gerekir:

- thlqual.SCSQSNLE, ABD İngilizcesi için karışık büyük/küçük harf
- thlqual.SCSQSNLU, ABD İngilizcesi için büyük harf
- thlqual.SCSQSNLK, Japonca için
- Fransızca için thlqual.SCSQSNLF
- thlqual.SCSQSNLC, Çince için

Kanal başlatıcı, uzun süre çalışan bir adres alanıdır. Sınırlı miktarda CPU tüketildikten sonra bitmesini önlemek için aşağıdakilerden birini onaylayın:

- z/OS sisteminizde başlatılan görevler için varsayılan değer sınırsız CPU 'dur; TIME = (1440,00) ile JOBCLASS (STC) için bir JES2 yapılandırma deyimini bunu gerçekleştirir ya da
- CSQXJST için EXEC deyimine belirttik olarak bir TIME=1440ya da TIME=NOLIMIT değiştirgesi ekleyin.

Kanal çıkışlarını kullanmak istiyorsanız, çıkış kitaplığını (CSQXLIB) daha sonra bu yordama ekleyebilirsiniz. Bunu yapmak için kanal başlatıcınızı durdurup yeniden başlatmanız gerekir.

TLS kullanıyorsanız, sistem TLS çalıştırma zamanı kitaplığına erişim gerekir. Bu kitaplığa SIEALNKE adı verilir. Kitaplık APF yetkisine sahip olmalıdır.

TCP/IP kullanıyorsanız, kanal başlatıcı adres alanının TCP/IP.DATA veri kümesi. Veri kümesinin ayarlanması gereken yollar, kullandığınız TCP/IP ürününe ve arabirimine bağlıdır. Bunlar şunları içerir:

- Ortam değişkeni, RESOLVER\_CONFIG
- Dosya sisteminde /etc/resolv.conf
- // SYSTCPD DD deyimini
- // SYSTCPDD DD deyimini
- *jobname/userid*.TCP/IP.DATA
- SYS1.TCPPARMS(TCPDATA)
- *zapname*.TCP/IP.DATA

Bunlardan bazıları, başlatılan görev yordamı JCL ' yi etkiler. Daha fazla bilgi için bkz. z/OS Communications Server: IP Configuration Guide.

### İlgili kavramlar

"IBM MQ altsistemini bir z/OS WLM hizmet sınıfına tanımlama" sayfa 895

z/OS sisteminde IBM MQ ' e uygun performans önceliği vermek için, kuyruk yöneticisi ve kanal başlatıcısı adres alanlarını uygun bir z/OS iş yükü yönetimi (WLM) hizmet sınıfına atamanız gerekir. Bunu belirttik olarak yapmazsanız, uygun olmayan varsayılanlar geçerli olabilir.

### **z/OS IBM MQ altsistemini bir z/OS WLM hizmet sınıfına tanımlama**

z/OS sisteminde IBM MQ ' e uygun performans önceliği vermek için, kuyruk yöneticisi ve kanal başlatıcısı adres alanlarını uygun bir z/OS iş yükü yönetimi (WLM) hizmet sınıfına atamanız gerekir. Bunu belirttik olarak yapmazsanız, uygun olmayan varsayılanlar geçerli olabilir.

- *Bu görevi her IBM MQ kuyruk yöneticisi için yineleyin.*
- *Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekmez.*

Aşağıdaki görevleri gerçekleştirmek için WLM ile birlikte sağlanan ISPF iletişim penceresini kullanın:

- z/OS WLM ilke tanımını WLM çift veri kümesinden çıkarın.
- Kuyruk yöneticisi ve kanal başlatıcısı tarafından başlatılan görev yordamı adlarını seçilen hizmet sınıfına ekleyerek bu ilke tanımlamasını güncelleyin
- Değiştirilen ilkeyi WLM çift veri kümesine kur

Daha sonra z/OS komutunu kullanarak bu ilkeyi etkinleştirin

```
V WLM,POLICY=policyname,REFRESH
```

Performans seçeneklerini ayarlama hakkında daha fazla bilgi için bkz. [z/OS üzerinde IBM MQ ortamınızın planlanması](#) .

### **İlgili kavramlar**

[“Db2 ortamını ayarlama” sayfa 932](#)

Kuyruk paylaşım gruplarını kullanıyorsanız, bir dizi örnek işi özelleştirerek ve çalıştırarak gerekli Db2 nesnelere yaratmanız gerekir.

### **z/OS ESM güvenlik denetimlerinizi uygulayın**

Kuyruk yöneticileri ve kanal başlatıcısı için güvenlik denetimlerini uygulayın.

- *Bu görevi her IBM MQ kuyruk yöneticisi için yineleyin.*
- *Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekebilir.*

Dış güvenlik yöneticiniz olarak RACF kullanıyorsanız, bu güvenlik denetimlerinin nasıl uygulanacağını açıklayan [z/OS üzerinde güvenliğin ayarlanması](#) başlıklı konuya bakın.

Kanal başlatıcısı kullanıyorsanız, aşağıdakileri de yapmanız gerekir:

- Altsisteminizde bağlantı güvenliği etkinse, dış güvenlik yöneticinizle ssid.CHIN bağlantı güvenliği tanımını tanımlayın (bu konuya ilişkin bilgi için [Kanal başlatıcısı için bağlantı güvenliği tanımları](#) konusuna bakın).
- TLS (Transport Layer Security; İletim Katmanı Güvenliği) ya da yuva arabirimi kullanıyorsanız, kanal başlatıcısının çalıştığı yetki altındaki kullanıcı kimliğinin, [z/OS UNIX System Services Planning](#) belgesinde açıklandığı gibi z/OS UNIX System Services kullanacak şekilde yapılandırıldığından emin olun.
- TLS kullanıyorsanız, kanal başlatıcısının çalıştığı yetki altındaki kullanıcı kimliğinin, ALTER QMGR komutunun SSLKEYR parametresinde belirtilen anahtarlık değerine erişecek şekilde yapılandırıldığından emin olun.

Kuyruk yöneticisini başlatmadan önce, IBM MQ veri kümesini ve sistem güvenliğini aşağıdaki şekilde ayarlayın:

- Kuyruk yöneticisi tarafından başlatılan görev yordamının dış güvenlik yöneticiniz altında çalışması için yetki verilmesi.
- Kuyruk yöneticisi veri kümelerine erişim yetkisi verilmesi.
- Gerekirse z/OS veri kümesi şifrelemesi yapılandırılıyor.

veri kümesi şifrelemesiyle IBM MQ for z/OS üzerinde atıl durumdaki veriler için gizlilik. başlıklı bölüme bakın. ek bilgi için.

Bunun nasıl gerçekleştirileceğine ilişkin ayrıntılar için bkz. [z/OS Güvenlik kuruluş görevleri](#).

RACF kullanıyorsanız, RACF BAŞLATAN sınıfını kullanmanız koşuluyla, sisteminize ilişkin bir IPL gerçekleştirmenize gerek yoktur (bkz. [RACF başlatılan görev yordamlarına ilişkin yetki](#)).

### İlgili kavramlar

“SYS1.PARMLIB üyeleri” sayfa 896

Bir IPL ' den sonra değişikliklerinizin yürürlükte kalmasını sağlamak için bazı SYS1.PARMLIB

“Kuyruk paylaşım grubu için ESM güvenlik denetimlerini uygula” sayfa 935

Db2 ' e ve bağlaım olanağı listesi yapılarına erişmek için bir kuyruk paylaşım grubundaki tüm kuyruk yöneticileri için güvenlik denetimleri uygulayın.

### **z/OS** **SYS1.PARMLIB üyeleri**

Bir IPL ' den sonra değişikliklerinizin yürürlükte kalmasını sağlamak için bazı SYS1.PARMLIB

- *Bu görevi, çalıştırmak istediğiniz her z/OS sistemi için bir kez gerçekleştirmeniz gerekir IBM MQ.*
- *Kuyruk paylaşım grupları kullanıyorsanız, sistem şebekesinde her z/OS sisteminde IBM MQ ayarlarının aynı olduğundan emin olmanız gerekir.*
- *Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekebilir.*

SYS1.PARMLIB üyeleri:

1. IEFSSNss üyesini “IBM MQ altsistemini şu şekilde tanımlayın: z/OS” sayfa 889 içinde açıklandığı gibi güncelleyin.
2. Bir IPL gerçekleştirilirken aşağıdaki üyelerin kullanılabilmesi için IEASYSpp değerini değiştirin:
  - “APF IBM MQ yükleme kitaplıklarını yetkilendirir” sayfa 883 içinde kullanılan PROGxx ya da IEAAPFaa üyeleri
  - “z/OS bağlantı listesini ve Yaşam Çevrimi Projesi Yönetimi” sayfa 884 içinde kullanılan LNKLSTkk ve LPALSTmm üyeleri
  - “z/OS program özellikleri çizelgesinin güncellenmesi” sayfa 888 içinde kullanılan SCHEDxx üyesi
  - “IBM MQ altsistemini şu şekilde tanımlayın: z/OS” sayfa 889 içinde kullanılan IEFSSNss üyesi

### İlgili kavramlar

“Kullanıma hazırlama giriş veri kümelerini özelleştirme” sayfa 896

Örnek kullanıma hazırlama giriş veri kümelerinin çalışma kopyalarını alın ve bunları sistem gereksinimlerinize uyacak şekilde uyarlayın.

### **z/OS** **Kullanıma hazırlama giriş veri kümelerini özelleştirme**

Örnek kullanıma hazırlama giriş veri kümelerinin çalışma kopyalarını alın ve bunları sistem gereksinimlerinize uyacak şekilde uyarlayın.

- *Bu görevi her IBM MQ kuyruk yöneticisi için yineleyin.*
- *Bu görevi, önceki bir sürümden geçiş yaparken gerçekleştirmeniz gerekir.*

Her IBM MQ kuyruk yöneticisi, ilk tanımlamalarını IBM MQ *kullanıma hazırlama giriş veri kümelerinde* bulunan bir dizi komuttan alır. Bu veri kümelerine, kuyruk yöneticisi tarafından başlatılan görev yordamında tanımlanan CSQINP1, CSQINP2 ve CSQINPT adları gönderme yapıyor.

Bu komutlara verilen yanıtlar, CSQOUT1, CSQOUT2 ve CSQOUTT DD adlarıyla gönderme yapılan kullanıma hazırlama çıkış veri kümelerine yazılır.

Orijinaleri korumak için her bir örneğin çalışma kopyalarını alın. Daha sonra, bu çalışma kopyalarında bulunan komutları sistem gereksinimlerinize uyacak şekilde uyarlayabilirsiniz.

Birden çok IBM MQ altsistemi kullanıyorsanız, altsistem adını, kullanıma hazırlama giriş veri kümesi adının üst düzey niteleyicisine ekleyecekseniz, her veri kümesiyle ilişkili IBM MQ altsistemini daha kolay tanıyabilirsiniz.

Örneklerle ilgili daha fazla bilgi için aşağıdaki konulara bakın:

- [Veri kümesi biçimlerini kullanıma hazırlama](#)
- [CSQINP1 örneğinin kullanılması](#)
- [CSQINP2 örneklerinin kullanılması](#)
- [CSQINPX örneğinin kullanılması](#)
- [CSQINPT örneğinin kullanılması](#)

## Kullanıma hazırlama veri kümesi biçimleri

Kullanıma hazırlama giriş veri kümeleri, bölümlenmiş veri kümesi (PDS) üyeleri ya da sıralı veri kümeleri olabilir. Bunlar birleştirilmiş veri kümeleri serisi olabilir. Bunları 80 baytlık bir kayıt uzunluğuyla tanımlayın; burada:

- Yalnızca 1-72 arası sütunlar anlamlıdır. 73-80 arası sütunlar yoksayılr.
- 1. kolonda yıldız imi (\*) olan kayıtlar açıklama olarak yorumlanır ve yoksayılr.
- Boş kayıtlar yoksayılr.
- Her komut yeni bir kayıta başlamalıdır.
- Sondaki bir ifade, sonraki kaydın 1. kolonundan devam edeceği anlamına gelir.
- Sondaki + işareti, sonraki kaydın ilk boş olmayan sütunundan devam edeceği anlamına gelir.
- Bir komutta izin verilen karakter sayısı üst sınırı 32 762 'dir.

Başlatma çıkışı veri kümeleri, kayıt uzunluğu 125, kayıt biçimi VBA ve blok boyutu 629 olan sıralı veri kümeleridir.

## CSQINP1 örneğinin kullanılması

Veri kümesi th1qua1.SCSQPROC , arabellek havuzlarının tanımlarını, arabellek havuzu ilişkilendirmelerini ve ALTER SECURITY komutunu içeren iki üyeyi içerir.

CSQ4INP1 üyesi, her ileti sınıfı için bir sayfa kümesi kullanır. İletiler aşağıdaki sınıflara ayrılır:

- Sistemle ilgili iletiler.
- Önemli uzun ömürlü mesajlar.
- Kısa ömürlü mesajlar.
- Çeşitli iletiler.

CSQ4INPR üyesi, ana ileti sınıflarının her biri için birden çok sayfa kümesi ve diğer sınıf için bir sayfa kümesi kullanır. Aşağıda ana ileti sınıfları yer almaktadır:

- Önemli uzun ömürlü mesajlar.
- Kısa ömürlü mesajlar.

Kuyruk yöneticisi tarafından başlatılan görev yordamının CSQINP1 birleşiminde uygun örneği ekleyin.

### Notlar:

1. IBM MQ , sıfır ile 99 arasında en çok 100 arabellek havuzunu destekler. DEFINE BUFFPOOL komutu yalnızca bir CSQINP1 kullanıma hazırlama veri kümesinden verilebilir. Örnekteki tanımlamalar dört arabellek havuzu belirtir.
2. Kuyruk yöneticisi tarafından kullanılan her sayfa kümesi, DEFINE PSID komutu kullanılarak CSQINP1 kullanıma hazırlama verileri kümesinde tanımlanmalıdır. Sayfa kümesi tanımlaması, bir arabellek havuzu tanıtıcısını sayfa kümesiyle ilişkilendirir. Arabellek havuzu belirtilmezse, varsayılan olarak arabellek havuzu sıfır kullanılır.

Sayfa kümesi sıfır (00) tanımlanmalıdır. Tüm nesne tanımlamalarını içerir. Her kuyruk yöneticisi için en çok 100 sayfa kümesi tanımlayabilirsiniz.

- ALTER SECURITY komutu, TIMEOUT ve INTERVAL güvenlik özniteliklerini değiştirmek için kullanılabilir. CSQ4INP1' de varsayılan değerler TIMEOUT için 54, INTERVAL için 12 olarak tanımlanır.

Arabellek havuzlarının ve sayfa kümelerinin düzenlenmesine ilişkin bilgi için [Sayfa kümelerinizin ve arabellek havuzlarının planlanması](#) başlıklı konuya bakın.

Kuyruk yöneticisi çalışırken arabellek havuzu ve sayfa kümesi tanımlamalarını devingen olarak değiştirirseniz, CSQINP1 tanımlamalarını da güncellemeniz gerekir. Arabellek havuzu tanımlaması REPLACE öznitelikliğini içermedikçe, değişiklikler yalnızca IBM MQsoğuk başlangıcı için alıkonur.

## CSQINP2 örneklerinin kullanılması

Bu çizelge, kuyruk yöneticisi tarafından başlatılan görev yordamının CSQINP2 birleştirmesinde içerilebilecek thlqua1 .SCSQPROC üyelerini, işlevlerinin bir tanımıyla listeler. Adlandırma kuralı CSQ4IN\*şeklinindedir. CSQ4INY\* üyeleri yapılandırmanız için değiştirilmelidir. Sonraki yayına geçtiğinizde değişiklikleri yeniden uygulamanız gerekeceği için CSQINS\* üyelerini değiştirmekten kaçınmalısınız. Bunun yerine, CSQ4INY\* üyelerine DEFINE ya da ALTER komutlarını koyabilirsiniz.

Çizelge 57. thlqua1 .SCSQPROC üyeleri	
Üye adı	Açıklama
CSQ4INSG	Sistem nesnesi tanımlamaları.
CSQ4INSA	Kanal kimlik doğrulamasına ilişkin sistem nesnesi ve varsayılan kurallar.
CSQ4INSX	Sistem nesnesi tanımlamaları.
CSQ4INSS	Kuyruk paylaşım grupları kullanıyorsanız, bu üyeyi özelleştirin ve ekleyin.
CSQ4INSJ	JMSkullanarak yayınlama/abone olma kullanıyorsanız, bu üyeyi özelleştirin ve ekleyin.
CSQ4INSM	Advanced Message Securityiçin sistem nesnesi tanımlamaları.
CSQ4INSR	WebSphere Application Serverkullanıyorsanız bu üyeyi özelleştirin ve ekleyin ya da IBM MQiçinde kuyruğa yollanmış yayınlama/abone olma yardımcı programı tarafından desteklenen kuyruğa alınmış yayınlama/abone olma arabirimini kullanın.
CSQ4DISP	Nesne tanımlamalarını görüntülemek için CSQINP2 örneği.
CSQ4INYC	Kümeleme tanımlamaları.
CSQ4INYD	Dağıtılmış kuyruğa alma tanımlamaları.
CSQ4INYG	Genel tanımlar.
CSQ4INYR	Depolama sınıfı tanımlamaları, iletinin ana sınıfları için birden çok sayfa kümesi kullanılıyor.
CSQ4INYS	Depolama sınıfı tanımlamaları, her ileti sınıfı için bir sayfa kümesi kullanılıyor.

Bir kuyruk yöneticisini her başlattığınızda değil, nesnelere yalnızca bir kez tanımlamanız gerekir; bu nedenle, bu tanımlamaların CSQINP2 ' ye her seferinde eklenmesi gerekmez. Bunları her seferinde eklerseniz, önceden var olan nesnelere tanımlamaya çalışıyorsanız ve aşağıdakilere benzer iletiler alırsınız:

```
CSQM095I +CSQ1 CSQMAQLC QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) ALREADY EXISTS
CSQM090E +CSQ1 CSQMAQLC FAILURE REASON CODE X'00D44003'
CSQ9023E +CSQ1 CSQMAQLC ' DEFINE QLOCAL' ABNORMAL COMPLETION
```

Nesneler bu hata nedeniyle zarar görmez. CSQINP2 birleşiminde SYSTEM tanımlama verileri kümesini bırakmak istiyorsanız, her nesne için REPLACE öznitelikliğini belirterek hata iletilerini önleyebilirsiniz.

## CSQINPX örneğinin kullanılması

Örnek `thlqua1.SCSQPROC(CSQ4INPX)`, kanal başlatıcısı her başlatıldığında yürütmek isteyebileceğiniz bir komut kümesi içerir. Bunlar genellikle, kuyruk yöneticisi her başlatıldığında değil, kanal başlatıcısı her başlatıldığında ve CSQINP1 ya da CSQINP2 giriş veri kümelerinde izin verilmeyen kanalla ilgili komutlardır. Bu örneği kullanmadan önce özelleştirmeniz gerekir; daha sonra kanal başlatıcısı için CSQINPX veri kümesine ekleyebilirsiniz.

Veri kümesinde bulunan IBM MQ komutları kanal başlatıcı kullanıma hazırlama sonunda yürütülür ve çıkış, CSQOUTX DD deyimi tarafından belirlenen veri kümesine yazılır. Çıkış, IBM MQ yardımcı programının (CSQUTIL) COMMAND işlevi tarafından üretilen çıkışa benzer. Bkz. [IBM MQ for z/OS için CSQUTIL yardımcı programının kullanılması](#).

Yalnızca kanal komutlarından değil, CSQUTIL ' den verilebilecek IBM MQ komutlarından herhangi birini belirtebilirsiniz. CSQINPX işlenirken diğer kaynaklardan komut girebilirsiniz. Önceki komutun başarısına bakılmaksızın, tüm komutlar sırayla verilir.

Bir komut yanıt süresi belirtmek için, veri kümesindeki ilk komut olarak sözde komut COMMAND komutunu kullanabilirsiniz. Bu, isteğe bağlı tek bir RESPTIME ( *nnn* ) anahtar sözcüğünü alır; burada *nnn* , her bir komuta yanıt bekleneceği saniye cinsinden zamandır. Bu değer 5-999 aralığındadır; varsayılan değer 30 'dur.

IBM MQ dört komuta verilen yanıtların çok uzun sürdüğünü saptarsa, CSQINPX işlemi durdurulur ve başka komut gönderilmez. Kanal başlatıcı durdurulmadı, ancak CSQOUTX veri kümesine [CSQU052E](#) iletisi yazıldı ve konsola [CSQU013E](#) iletisi gönderildi.

IBM MQ , CSQINPX ' i işlemeyi başarıyla tamamladığında konsola [CSQU012I](#) iletisi gönderilir.

## CSQINPT örneğinin kullanılması

Bu çizelge, kuyruk yöneticisi tarafından başlatılan görev yordamının CSQINPT birleştirmesinde içerilebilecek `thlqua1.SCSQPROC` üyelerini, işlevlerinin bir tanımıyla listeler.

Çizelge 58. <code>thlqua1.SCSQPROC</code> üyeleri	
Üye adı	Açıklama
CSQ4INST	Sistemin varsayılan abonelik tanımlaması.
CSQ4INYT	Tanımları yayınlama/abone ol.

Veri kümesinde bulunan IBM MQ komutları, yayınlama/abone olma işlemi tamamlandığında ve çıkış, CSQOUTT DD deyimi tarafından belirtilen veri kümesine yazıldığında yürütülür. Çıkış, IBM MQ yardımcı programının (CSQUTIL) COMMAND işlevi tarafından üretilen çıkışa benzer. Bkz. [IBM MQ for z/OS için CSQUTIL yardımcı programının kullanılması](#).

### İlgili kavramlar

[“Önyükleme ve günlük veri kümelerinin yaratılması” sayfa 901](#)

Sağlanan CSQJU003 programını kullanarak önyükleme veri kümelerini (BSDs) ve günlük veri kümelerini hazırlayın.

### QMINI veri kümesi

Kuyruk yöneticisi kullanıma hazırlanırken okunacak ve işlenecek özellikleri belirtmek için QMINI veri kümesini kullanabilirsiniz.

## QMINI veri kümesinin özellikleri

QMINI veri kümesi, kayıt uzunluğu üst sınırı 80 bayt (veri için 72 bayt ve satır numarası için sekiz bayt) olan sıralı bir veri kümesi ya da bölümlenmiş veri kümesinin bir üyesidir.

Aşağıdaki örnek, sıralı bir QMINI veri kümesine ilişkin özellikleri göstermektedir. Bazı özellikler, elbette, çevreye bağlıdır.



```

Data Set Name . . . . : QM01.QMINI
General Data
Management class . . : STANDARD      Current Allocation
Storage class . . . . : STANDARD      Allocated tracks . : 1
Volume serial . . . . : P5P21E       Allocated extents . : 1
Device type . . . . . : 3390
Data class . . . . . : **None**
Organization . . . . : PS            Current Utilization
Record format . . . . : FB           Used tracks . . . . : 0
Record length . . . . : 80          Used extents . . . . : 0
Block size . . . . . : 3120
1st extent tracks . . : 1
Secondary tracks . . . : 1          Dates
Data set name type . . :             Creation date . . . : 2020/08/11
Data set encryption . : NO          Referenced date . . : ***None***
SMS Compressible . . . : NO         Expiration date . . : ***None***

```

thlqual.SCSQPROC, aşağıdakileri içerir:

- CSQ4QMINI içindeki bir QMINI veri kümesine ilişkin örnek içerik.
- Başlatılan CSQ4MSTR ve CSQ4MSRR görev yordamlarında, kuyruk yöneticisi başlatma JCL ' de // CSQMINI DD kartını kullanarak QMINI veri kümesini belirtme örneği.

#### Notlar:

- Veri kümesini ayırtan kod, her kaydın ilk 72 baytını ayrıştırır.
- Satır numaraları dikkate alınmaz, bu nedenle satır numaralarının belirtilmesi gerekmez.
- Bir satır yıldız işaretiyle (\*) başlıyorsa, satır bir açıklama olarak işlenir.
- Kuyruk yöneticisi başlatılırken QMINI veri kümesinin içeriği ayrıştırılır. İçerik başarıyla ayrıştırılırsa, kuyruk yöneticisi iş günlüğünde CSQM578I iletisi yayınlanır. Ayrıştırma sırasında herhangi bir hatayla karşılaşırsa, kuyruk yöneticisi iş günlüğünde CSQM573E gibi hata iletileri yayınlanır, ancak kuyruk yöneticisi yine de başlatılır.

Hata iletilerini denetleyin ve QMINI veri kümesinin içeriğindeki sorunları çözün.

Kuyruk yöneticisi QMINI veri kümesini ayrıştıramazsa, kanal başlatıcısını başlatabilirsiniz, ancak güvenlik yapılandırma ayarları bilinmediği için SSL ya da TLS kullanacak şekilde yapılandırılmış kanalları başlatamayabilirsiniz.

- Kuyruk yöneticisini başlattıktan sonra veri kümesinde güncelleme yaparsanız, değişiklikleri almak için kuyruk yöneticisini yeniden başlatmanız gerekir.

## TransportSecurity kısmı

IBM MQ for z/OS 9.2.0' den QMINI veri kümesi, TransportSecurity dörtünü destekler. Bu kıta, IBM MQ for Multiplatforms üzerindeki qm.ini dosyasında SSL kısmı tarafından sağlanana benzer bir işlev sağlar.

TransportSecurity kısmı aşağıdaki özellikleri destekler:

#### AllowTLSV13

Bir kuyruk yöneticisinin TLS 1.3 CipherSpecs ' i kullanıp kullanamayacağı; geçerli değerler şunlardır: TRUE/T/YES/Y ya da FALSE/F/NO/N.

Geçirilen kuyruk yöneticileri için TLS 1.3 varsayılan olarak etkinleştirilmez. TransportSecurity stanza ve **AllowTLSV13=TRUE** ile bir QMINI veri kümesi tanımlayarak TLS 1.3 ' ü etkinleştirebilirsiniz.

Yeni oluşturulan kuyruk yöneticileri için TLS 1.3 varsayılan olarak etkindir.

#### AllowedCipherBelirtileri

Etkinleştirilen özel bir CipherSpecs listesi.

Bu özelliğe ilişkin ek bilgi için bkz. [IBM MQ for z/OS üzerinde sıralı ve etkin CipherSpecs ' in özel bir listesini sağlama](#) .

Listedeki yinelenen CipherSpec adları yoksayılır.



## V 9.3.0 OutboundSNI

Sunucu Adı Göstergesi (SNI), bir TLS bağlantısı başlatılırken hedef IBM MQ kanal adına mı, yoksa anasistem adına mı ayarlanır; geçerli değerler şunlardır: CHANNEL ya da HOSTNAME.

Hedef kanal CERTLABL alanında bir sertifika etiketiyle yapılandırıldıysa, CERTLABL ' yi kanal değerine ayarlamamız gerekir. HOSTNAME , CERTLABL ayarına sahip bir kanalla bağlantı kurulursa, bağlantı başarısız olur ve uzak kuyruk yöneticisi hata günlüklerine AMQ9673 iletisi yazılır.

Aşağıdaki örnekte TransportSecurity kısmı nasıl belirtildiği gösterilmektedir:

```
TransportSecurity:  
AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,  
                    ECDHE_RSA_AES_256_GCM_SHA384  
AllowTLSV13=TRUE
```

## z/OS Önyükleme ve günlük veri kümelerinin yaratılması

Sağlanan CSQJU003 programını kullanarak önyükleme veri kümelerini (BSDSs) ve günlük veri kümelerini hazırlayın.

### Not:

- Bu görevi her IBM MQ kuyruk yöneticisi için yineleyin.
- BSDS ya da etkin günlük veri kümelerini korumak için z/OS veri kümesi şifrelemesi kullanıyorsanız, veri kümeleri bu adımda ayrılmadan önce bu seçeneği yapılandırmanız gerekir.
- Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekmez.
- Bir kuyruk yöneticisini geçirip etkin günlük veri kümeleri ya da BSDS için z/OS veri kümesi şifrelemesi ekliyorsanız, veri kümelerini dönüştürmeniz gerekir.
- z/OS veri kümesi şifrelemesinin yapılandırılması ve var olan IBM MQ veri kümelerinin şifrelenmek üzere dönüştürülmesi hakkında daha fazla bilgi için bkz. [IBM MQ for z/OS veri kümesi şifrelemesi ile atıl durumdaki veriler için gizlilik.](#)

Tek ya da çift günlük kaydı ortamı yaratmak için CSQJU003 komutunu çalıştıracak örnek JCL ve Erişim Yöntemi Hizmetleri (AMS) denetim deyimleri thlqual.SCSQPROC(CSQ4BSDS) içinde tutulur. BSDS'lerinizi ve günlüklerinizi oluşturmak ve günlükleri önceden biçimlendirmek için bu işi özelleştirin ve çalıştırın.

**Önemli:** CSQ4BSDS'nin en yeni sürümünü kullanmalı ya da JCL' yi RECORDS (850 60) kullanacak şekilde el ile güncellemelisiniz.

Başlatılan görev yordamı ( CSQ4MSTR), [“IBM MQ kuyruk yöneticisi için yordam yaratılması” sayfa 892](#) içinde açıklanmıştır ve formun deyimlerindeki BSDS'lere başvurur:

```
//BSDS1 DD DSN=++HLQ++.BSDS01,DISP=SHR  
//BSDS2 DD DSN=++HLQ++.BSDS02,DISP=SHR
```

Günlük veri kümelerine BSDSler tarafından başvurulur.

### Not:

1. BLKSIZE, LOGDEF adımıdaki SYSPRINT DD deyiminde belirtilmelidir. BLKSIZE 629 olmalıdır.
2. Farklı kuyruk yöneticilerinden gelen önyükleme veri kümelerinin ve günlük veri kümelerinin belirlenmesine yardımcı olmak için, altsistem adını bu veri kümelerinin üst düzey niteleyicisine ekleyin.
3. Kuyruk paylaşım grupları kullanıyorsanız, önyükleme ve günlük veri kümelerini SHAREOPTIONS (2 3) ile tanımlamanız gerekir.

Önyükleme ve günlük veri kümelerinin ve boyutlarının planlanmasına ilişkin bilgi için bkz. [z/OS üzerinde planlama](#) .

IBM MQ 8.0' den itibaren 8 baytlık günlük RBA geliştirmesi, [Daha büyük günlük Relative Byte Address](#)(Görelî Bayt Adresi) başlıklı konuda açıklandığı gibi bir kuyruk yöneticisinin kullanılabilirliğini

artırır. Kuyruk yöneticisi ilk başlatılmadan önce bir kuyruk yöneticisinde 8 baytlık günlük RBA ' yı etkinleştirmek için günlük kaydı ortamınızı yarattıktan sonra aşağıdaki adımları gerçekleştirin.

**Not:** **V9.3.0** IBM MQ 9.3.0 ya da sonraki 8 baytlık günlük RBA ' da oluşturulan kuyruk yöneticileri için zaten etkinleştirilmiş olduğundan aşağıdaki adımlar gerekli değildir.

1. **IDCAMS ALTER**olanağını kullanarak, sürüm 1 biçimi BSDS'lerini ( CSQJU003 programı kullanılarak yaratılır) ++HLQ++ . V1 . BSDS01gibi bir biçimde yeniden adlandırın.

**Not:** VSAM kümesinin yanı sıra veri ve izin bileşenlerini yeniden adlandırdığınızdan emin olun.

2. Önceden tanımlananla aynı özniteliklere sahip yeni BSDSs ayırın. Bunlar, kuyruk yöneticisi başlatıldığında kullanılacak sürüm 2 biçimli BSDS'lere dönüşür.
3. BSDS dönüştürme yardımcı programını (CSQJUCNV) kullanarak, sürüm 1 biçimli BSDS ' leri yeni sürüm 2 biçimli BSDS'lere dönüştürün.
4. Dönüştürme başarıyla tamamlandıktan sonra, sürüm 1 biçimindeki BSDS' leri silin.

**Not:** Kuyruk yöneticisi bir kuyruk paylaşım grubundaya, kuyruk paylaşım grubundaki tüm kuyruk yöneticileri, 8 baytlık günlük RBA etkinleştirilmeden önce aşağıdaki gibi başlatılmalıdır:

- Kuyruk yöneticisi IBM MQ 9.0.0 LTS ise, **OPMODE**(NEWFUNC,900) ya da **OPMODE**(NEWFUNC,800) ile başlatılmalıdır.
- Kuyruk yöneticisi IBM MQ 9.0.n CDise, ya da IBM MQ 9.1.0 LTSya da daha sonraki bir yayın düzeyinin bu düzeyde başlatılması gerekir

### İlgili kavramlar

“Sayfa kümelerinizi tanımlayın” sayfa 902

Sağlanan örneklerden birini kullanarak her kuyruk yöneticisi için sayfa kümeleri tanımlayın.

### **z/OS** Sayfa kümelerinizi tanımlayın

Sağlanan örneklerden birini kullanarak her kuyruk yöneticisi için sayfa kümeleri tanımlayın.

- *Bu görevi her IBM MQ kuyruk yöneticisi için yineleyin.*

*Sayfa kümelerini korumak için z/OS veri kümesi şifrelemesi kullanıyorsanız, bu adımda veri kümeleri ayrılmadan önce bu seçeneği yapılandırmanız gerekir.*

- *Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekmez.*

*Bir kuyruk yöneticisini geçirip sayfa kümeleri için z/OS veri kümesi şifrelemesi ekliyorsanız, sayfa kümelerini dönüştürmeniz gerekir.*

*Veri kümesi şifrelemesiyle IBM MQ for z/OS üzerinde atıl durumdaki veriler için gizlilik bölümüne bakın. z/OS veri kümesi şifrelemesini yapılandırma ve var olan IBM MQ veri kümelerini şifrelenecek şekilde dönüştürme hakkında daha fazla bilgi için.*

Her IBM MQ kuyruk yöneticisi için ayrı sayfa kümeleri tanımlayın. thlqual.SCSQPROC(CSQ4PAGE) ve thlqual.SCSQPROC(CSQ4PAGR), sayfa kümelerini tanımlamak ve biçimlendirmek için JCL ve z/OS erişim yöntemi hizmetleri (AMS) denetim deyimlerini içerir. CSQ4PAGE üyesi, her ileti sınıfı için bir sayfa kümesi kullanır; CSQ4PAGR üyesi, iletinin ana sınıfları için birden çok sayfa kümesi kullanır. JCL, sağlanan CSQUTIL yardımcı programını çalıştırır. Örnekleri gözden geçirin ve bunları istediğiniz sayfa kümesi sayısı ve kullanılacak boyutlar için özelleştirin. Sayfa kümeleri ve uygun boyutların nasıl hesaplanacağına ilişkin bilgi için [Sayfa kümelerinizin ve arabellek havuzlarınızın planlanması](#) başlıklı konuya bakın.

Başlatılan görev yordamı CSQ4MSTR , “IBM MQ kuyruk yöneticisi için yordam yaratılması” sayfa 892 içinde açıklanan biçimde sayfa kümelerine gönderme yapıyor:

```
//CSQP00nn DD DISP=OLD,DSN=xxxxxxxx
```

Burada nn 00 ile 99 arasında sayfa kümesi numarası, xxxxxxxx ise tanımladığınız veri kümeidir.

**Not:**

1. Dinamik sayfa kümesi genişletme özelliğini kullanmayı amaçlıyorsanız, her sayfa kümesi için ikincil kapsamların tanımlandığından emin olun. thlqual.SCSQPROC(CSQ4PAGE), bunun nasıl yapılacağını gösterir.
2. Farklı kuyruk yöneticilerinden sayfa kümelerini tanımlamaya yardımcı olmak için, altsistem adını her sayfa kümesiyle ilişkili veri kümesinin üst düzey niteleyicisine ekleyin.
3. FORCE seçeneğinin CSQUTIL yardımcı programının FORMAT işleviyle birlikte kullanılmasını istiyorsanız, AMS DEFINE CLUSTER deyiminde REUSE özniteliğini eklemeniz gerekir.  
REUSE ile ilgili daha fazla bilgi için z/OS DEFINE CLUSTER komutunun [İsteğe Bağlı Parametreler](#) bölümüne bakın.
4. Sayfa kümeleriniz 4 GB ' den büyük olacaksa, SMS (Storage Management System; Depolama Yönetimi Sistemi) EXTENDED ADDRESSABILITY işlevini kullanmanız gerekir.

### İlgili kavramlar

“IBM MQ girdilerini Db2 tablolarına ekleme” sayfa 935

Kuyruk paylaşım gruplarını kullanıyorsanız, Db2 veri paylaşım grubundaki IBM MQ çizelgelerine kuyruk paylaşım grubu ve kuyruk yöneticisi girişleri eklemek için CSQ5PQSG yardımcı programını çalıştırın.

### z/OS Sistem parametre modülünüzü uyarlama, da,-

IBM MQ sistem parametresi modülü, IBM MQ ' un işletiminde kullandığı günlüğe kaydetme, arşivleme, izleme ve bağlantı ortamlarını denetler. Varsayılan bir modül sağlanır. Kendi sistem parametre modülünüzü bazı parametreler (örneğin, veri kümesi adları) olarak yaratmanız gerekir; bunlar genellikle siteye özgüdür.

- Bu görevi, her IBM MQ kuyruk yöneticisi için gerektiği şekilde yineleyin.
- Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekebilir. Ayrıntılar için bkz. [z/OS](#) üzerinde IBM MQ geçişi yapılıyor.
- Var olan bir kuyruk yöneticisinde Advanced Message Security for z/OS özelliğini etkinleştirmek için, SPLCAP ' ı “CSQ6SYSP ' nin kullanılması” sayfa 905 içinde açıklandığı gibi YES değerine ayarlamanız gerekir. Bu kuyruk yöneticisini ilk kez yapılandırıyorsanız, bu görevin tamamını tamamlayın.

Sistem parametre modülünde aşağıdaki gibi dört makro vardır:

Makro adı	Amaç
CSQ6SYSP	Bağlantı ve izleme parametrelerini belirtir, bkz. “CSQ6SYSP ' nin kullanılması” sayfa 905
CSQ6LOGP	Günlük başlatma işlemini denetler, bkz. “CSQ6LOGP ' nin kullanılması” sayfa 914
CSQ6ARVP	Arşiv başlatmayı denetler, bkz. “CSQ6ARVP ' nin kullanılması” sayfa 919
CSQ6USGP	Kullanım kaydını denetler, bkz. “CSQ6USGP ' nin kullanılması” sayfa 925

IBM MQ , IBM MQyönetim ortamını başlatmak için START QMGR komutunu (PARM değiştirgesi olmadan) çalıştırdığınızda otomatik olarak çağrılan varsayılan bir sistem değiştirgesi modülü (CSQZPARM) sağlar. CSQZPARM, IBM MQile birlikte verilen thlqual.SCSQAUTH APF yetkili kitaplığında yer alır. Bu parametrelerin değerleri, IBM MQ' i başlattığınızda bir ileti dizisi olarak görüntülenir.

Bu komutun nasıl kullanıldığına ilişkin ek bilgi için [START QMGR](#) başlıklı konuya bakın.

### Kendi sistem parametre modülünüzü oluşturma

CSQZPARM istediğiniz sistem değiştirgelerini içermiyorsa, thlqual.SCSQPROC(CSQ4ZPRM) içinde sağlanan örnek JCL ' yi kullanarak kendi sistem değiştirgesi biriminizi yaratabilirsiniz.

Kendi sistem parametresi modülünüzü yaratmak için:

1. JCL örneğinin çalışma kopyasını alın.
2. Kopyadaki her makroya ilişkin parametreleri gerektiği gibi düzenleyin. Makro çağrılarında herhangi bir parametre kaldırırsanız, yürütme sırasında varsayılan değerler otomatik olarak alınır.
3. ++NAME++ yer tutucusunu, yükleme modülünün alacağı adla değiştirin (bu CSQZPARM olabilir).
4. Çevirici üst düzey çevirici değilse, çevirici tarafından gerektiği gibi JCL ' yi değiştirin.
5. Yükleme modülü oluşturmak üzere sistem parametresi makrolarının uyarlanmış sürümlerini derlemek ve bağlamak için JCL ' yi çalıştırın. Bu, belirttiğiniz adı taşıyan yeni sistem parametresi modülüdür.
6. APF yetkili kullanıcı kitaplığına üretilen yükleme modülünü yerleştirin.
7. APF yetkili kullanıcı kitaplığına kullanıcı READ erişimi ekleyin.
8. Bu kitaplığı IBM MQ kuyruk yöneticisi tarafından başlatılan STEPLIB görev yordamına ekleyin. Bu kitaplık adı, STEPLIB içindeki thlqual.SCSQAUTH kitaplığından önce gelmelidir.
9. Kuyruk yöneticisini başlattığınızda yeni sistem değiştirgesi modülünü çağırın. Örneğin, yeni modülün adı NEWMODS ise, şu komutu verin:

```
START QMGR PARM(NEWMODS)
```

10. İş günlüğünü denetleyerek komutun başarıyla tamamlandığından emin olun. Günlükte aşağıdakine benzer bir giriş olmalıdır:

```
CSQ9022I CDL1 CSQYASCP 'START QMGR' NORMAL COMPLETION
```

Kuyruk yöneticisi başlatma JCL ' de parametre modülü adını da belirtebilirsiniz. Daha fazla bilgi için bkz. [z/OSüzerinde bir kuyruk yöneticisini başlatmak ve durdurmak için MQSC ' nin kullanılması](#).

**Not:** CSQZPARM modülünüzü adlandırmayı seçerseniz, START QMGR komutunda PARM parametresini belirtmeniz gerekmez.

## Sistem parametresi modülünün ince ayarlı olarak ayarlanması

IBM MQ ayrıca, var olan bir sistem parametresi modülünü ince ayar yapmak için kullanılabilir üç çevirici kaynak modülü de sağlar. Bu modüller thlqual.SCSQASMS kitaplığında bulunur. Genellikle, bu modülleri bir sınama ortamında sistem değiştirgesi makrolarındaki varsayılan değiştirgeleri değiştirmek için kullanırsınız. Her kaynak modül farklı bir sistem parametresi makrosunu çağırır:

Bu çevirici kaynak modülü ...	Bu makroyu çağırır ...
CSQFSYP	CSQ6SYSP (bağlantı ve izleme değiştirgeleri)
CSQJLOGP	CSQ6LOGP (günlüğü kullanıma hazırlama)
CSQJARVP	CSQ6ARVP (arşiv kullanıma hazırlama)

Bu modülleri bu şekilde kullanırsınız:

1. Bir kullanıcı çevirici kitaplığında her bir çevirici kaynak modülünün çalışma kopyalarını çıkarın.
2. İsteddiğiniz parametrelerin değerlerini ekleyerek ya da değiştirerek kopyalarınızı düzenleyin.
3. Bir kullanıcı nesnesi kitaplığında nesne modülleri yaratmak için, düzenlenen modüllerin kopyalarını derleyebilirsiniz.
4. Yeni sistem değiştirgesi modülü olan bir yükleme modülü üretmek için bu nesne kodu modüllerini var olan bir sistem değiştirgesi modülüyle bağlantılandırın.
5. Yeni sistem parametresi modülünün, kullanıcı yetkili kitaplığının üyesi olduğundan emin olun.

6. Bu kitaplığı kuyruk yöneticisi tarafından başlatılan STEPLIB görev yordamına ekleyin. Bu kitaplık, STEPLIB içindeki thlqual.SCSQAUTH kitaplığından önce gelmelidir.
7. Yeni sistem değiştirgesi modülünü çağırmak için, PARM değiştirgesinde yeni modül adını belirterek bir START QMGR komutu verin.

SMP/E denetimi altında özelleştirilmiş sistem parametrelerinin nasıl yönetileceğini gösteren SCSQPROC ' nin CSQ4UZPR üyesinde örnek bir usermod sağlanır.

## Sistem parametrelerinin değiştirilmesi

Bir kuyruk yöneticisi çalışırken bazı sistem parametrelerini değiştirebilirsiniz; [SET SYSTEM](#), [SET LOG](#) ve [SET ARCHIVE](#) komutlarına bakın.

SET komutlarını, kuyruk yöneticisini her başlattığınızda yürürlüğe girmesi için kullanıma hazırlama giriş veri kümelerinize koyun.

### İlgili kavramlar

“Kanal başlatıcı değiştirgelerini uyarla” sayfa 926

Kanal başlatıcıyı gereksinimlerinize uyacak şekilde uyarlamak için ALTER QMGR işlevini kullanın.

### CSQ6SYSP ' nin kullanılması

CSQ6SYSP kullanılarak sistem parametrelerinin nasıl ayarlanacağına ilişkin başvuru olarak bu konuyu kullanın.

CSQ6SYSP için varsayılan parametreler ve SET SYSTEM komutunu kullanarak her parametreyi değiştirip değiştiremeyeceğiniz Çizelge 59 sayfa 905 içinde gösterilir. Bu değerlerden herhangi birini değiştirmek istiyorsanız, parametrelerin ayrıntılı açıklamalarına bakın.

Parametre	Açıklama	Varsayılan değer	SET komutu
<a href="#">V9.3.0</a> “[MQ 9.3.0 Temmuz 2021]ACCTIME ” sayfa 907	Muhasebe verilerinin her bir toplanması arasındaki dakika ve saniye cinsinden süre.	-1	✓
“ACELIM.” sayfa 907	1 KB ' lik bloklar halinde ACE depolama havuzunun boyutu.	0 (sınır yok)	✓
“CLCACHE” sayfa 908	Kullanılacak küme önbelleğinin tipini belirtir.	STATIC	-
“CMDUSER” sayfa 908	Komut güvenliği denetimleri için varsayılan kullanıcı kimliği.	CSQOPR	-
“EXCLMSG” sayfa 908	Herhangi bir günlükten dışlanacak iletilerin listesini belirtir. Bu listedeki iletiler z/OS konsoluna ve basılı günlüğe gönderilmez. Sonuç olarak, iletileri dışlamak için EXCLMSG parametresinin kullanılması, CPU perspektifinden “Bilgi iletilerini engelle” sayfa 931 içinde açıklanan yöntemleri kullanmaktan daha verimlidir.	( )	✓
“Çıkış” sayfa 909	Her çağırma sırasında kuyruk yöneticisi çıkışlarının çalışabileceği süre üst sınırı (saniye).	30	-

Çizelge 59. CSQ6SYSP değıştirtgelerinin varsayılan değeri (devamı var)

Parametre	Açıklama	Varsayılan değeri	SET komutu
<u>“Çıkış” sayfa 909</u>	Kuyruk yöneticisi çıkışlarını çalıştırmak için kullanılacak başlatılmış sunucu görevlerinin sayısı.	8	-
<u>“YükLEME” sayfa 909</u>	Bir denetim noktasının başlangıcı ile sonraki arasında IBM MQ tarafından yazılan günlük kaydı sayısı.	500 000	✓
<u>“MULCCAPT” sayfa 909</u>	Ölçülen Kullanım Lisansı Ücretlendirmesi (MULC) tarafından kullanılan verileri toplamaya ilişkin algoritmayı denetleyen Ölçülü Kullanım Fiyatlandırması özelliğini belirler.	Bkz. <u>parametre tanımı</u>	-
<u>“OTMACON” sayfa 910</u>	OTMA bağlantı parametreleri.	Bkz. <u>parametre tanımı</u>	-
<u>“QINDXBLD” sayfa 910</u>	Kuyruk yöneticisi yeniden başlatmasının, tüm dizinler yeniden oluşturuluncaya kadar mı, yoksa tüm dizinler yeniden oluşturulmadan önce mi tamamlanacağını belirler.	Bekle	-
<u>“QMCCSID” sayfa 911</u>	Kuyruk yöneticisine ilişkin kodlanmış karakter takımı tanıtıcısı.	Sıfır	-
<u>“QSGDATA” sayfa 911</u>	Kuyruk paylaşım grubu parametreleri.	Bkz. <u>parametre tanımı</u>	-
<u>“İLK DURUMA GEÇİR” sayfa 912</u>	RESLEVEL denetim parametresi.	EVET	-
<u>“ROUTCDE” sayfa 912</u>	Belirli bir konsoldan istenmeyen iletilere atanan ileti yönlendirme kodu.	1	-
<u>“HİZMET” sayfa 912</u>	IBM tarafından kullanılmak üzere ayrılmıştır.	0	✓
<u>“SMFACCT” sayfa 912</u>	Kuyruk yöneticisi başlatıldığında SMF muhasebe verilerinin toplanıp toplanmayacağını belirtir. Sınıf 4 kanal muhasebe verilerinin yalnızca kanal başlatıcı başlatıldığında toplandığını unutmayın.	HAYIR	-
<u>SMFSTAT</u>	Kuyruk yöneticisi başlatıldığında SMF istatistiklerinin toplanıp toplanmayacağını belirtir. Sınıf 4 kanal başlatıcı istatistik verilerinin yalnızca kanal başlatıcı başlatıldığında toplandığını unutmayın.	HAYIR	-
<u>SPLCAP</u>	Bu kuyruk yöneticisinde kuyruk güvenliği ilkesi yeteneğinin etkinleştirilip etkinleştirilmediğini belirtir. Advanced Message Security for z/OS için bu parametreyi YES olarak ayarlayın.	HAYIR	-

Çizelge 59. CSQ6SYSP değıştirtgelerinin varsayılan değeri (devamı var)

Parametre	Açıklama	Varsayılan değeri	SET komutu
DURUM	V 9.3.0 İstatistiklerin her bir toplanması arasındaki dakika ve saniye cinsinden süre.	30	✓
TRACSTR	İzlemenin otomatik olarak başlatılıp başlatılmayacağını belirtir.	HAYIR	-
TRACTBL	Genel izleme olanağı tarafından kullanılacak izleme çizelgesinin büyüklüğü (4 KB).	99 (396 KB)	✓
WLMTIME	WLM tarafından yönetilen kuyruklar için kuyruk dizininin taranması arasındaki süre.	30	-
WLMTIMU	WLMTIME için birimler (dakika ya da saniye).	dak	-

### V 9.3.0 ACCTIME

Muhasebe verilerinin ardışık toplamaları arasındaki aralığı dakika ve saniye cinsinden belirtir.

-1ya da 0 -1440 dakika 'mmm' biçiminde ya da 0 -1440 dakika ve 0 -59 saniye 'mmm.ss' aralığında bir sayı belirtin.

#### Notlar:

- Yalnızca saniye aralığı belirtirken, aralığın başına 0 önekini eklemeniz gerekir. Olası en küçük aralık bir saniyedir: '0.01'.
- 0 değerini belirtirseniz, muhasebe verileri SMF genel kayıt aralığında toplanır. Ek bilgi için [Using System Management Facility](#) başlıklı konuya bakın.
- Varsayılan değeri olan -1 değerini belirtirseniz, muhasebe verileri STATIME değeri tarafından belirlenen aralıkta toplanır.

Örneğin:

'0.30', 30 saniyelik bir aralık ayarlar.

'5.30', 5 dakika 30 saniye aralığını ayarlar.

'30', 30 dakikalık bir aralık ayarlar.

#### ACELIM.

1 KB'lik bloklar halinde ACE depolama havuzu boyutu üst sınırını belirtir. Sayı 0-999999 aralığında olmalıdır. Varsayılan değeri olan sıfır, sistemde bulunanın ötesinde zorunlu kısıtlama olmadığı anlamına gelir.

Yalnızca ECSA depolamasının aşırı miktarlarını kullandığı belirlenen kuyruk yöneticilerindeki ACELIM için bir değeri ayarlamanız gerekir. ACE saklama havuzunun sınırlandırılması, sistemdeki bağlantı sayısını sınırlama etkisine neden olur; bu nedenle, bir kuyruk yöneticisi tarafından kullanılan ECSA saklama alanı miktarı.

Kuyruk yöneticisi sınıra ulaştığında, uygulamaların yeni bağlantılar edinmesi mümkün değildir. Yeni bağlantıların olmaması MQCONN işleminde başarısızlıklara neden olur ve RRS ile eşgüdümlü uygulamalar herhangi bir IBM MQ API 'sinde hat-Hayır seni Sürati ile karşılaşacak.

ACE, bir bağlantıya ilişkin iş parçacığıyla ilgili denetim öbekleri için gereken toplam ECSA'nın yaklaşık %12.5'ini temsil eder. Örneğin, ACELIM=5120 değerinin belirlenmesi, kuyruk yöneticisi tarafından ayrılan toplam ECSA miktarını (iş parçacığıyla ilgili denetim öbekleri için) yaklaşık 40960K; 5120 değerinin 8 ile çarpılması beklenmektedir.

Kuyruk yöneticisi tarafından ayrılan toplam ECSA miktarını sınırlamak için, 5120Kiş parçacığıyla ilgili denetim blokları için 640 ACELIM değeri gereklidir.

'ACE/PEB' depolama havuzunun büyüklüğünü izlemek için, CLASS (3) istatistiği tarafından üretilen SMF 115 alt tip 5 kayıtlarını kullanabilir ve ACELIM için uygun bir değeri ayarlayabilirsiniz.



Kuyruk yöneticisi tarafından kullanılan toplam ECSA saklama alanı miktarını, denetim öbekleri için, CLASS (2) istatistiği tarafından yazılan SMF 115 alt tip 7 kayıttan elde MSbelli edebilirsiniz. Kullanılan toplam ECSA saklama alanı miktarı, QSRSPHBGF ve QSRSPHBGV alanlarının toplamıdır.

SMF 115 istatistik kayıtlarıyla ilgili ek bilgi için [IBM MQ performans istatistiklerini yorumlamabaşlıklı](#) konuya bakın.

Bir z/OS görüntüsünü, bir kuyruk yöneticisine yönelik uygulama bağlantılarını denetlemek yerine, kötü davranışlı bir kuyruk yöneticisinden korumak için bir mekanizma olarak ACELIM ' i ayarlamanız gerektiğini unutmayın.

## **CLCACHE**

Kullanılacak küme önbelleğinin tipini belirtir.

Küme önbelleği, kümeyle ilgili bilgileri depolamak için kullanılan bir depolama alanıdır.

Küme önbelleği durağansa, kuyruk yöneticisi başlatma sırasında ayrılmış sabit bir büyüklüğü vardır. Önbellek doluysa, CSQM060E iletisi yayınlanır ve daha fazla yer gerektiren uygulama isteği bir MQRC\_CLUSTER\_RESOURCE\_ERROR alır.

CLCACHE dinamik olarak ayarlanırsa, küme önbelleği gerektiği şekilde genişletilebilir. Ancak, önce kurulu küme iş yükü çıkışlarının dinamik bir önbellekle çalışabildiğinden emin olmanız gerekir.

Kurulu bir küme iş yükü çıkışı dinamik önbellek iletisiyle CSQM061E çalışmazsa.

MQXCLWLN, küme iş yükü çıkışlarının, dinamik ya da durağan önbelleklerin kullanılıp kullanılmadığına bakılmaksızın, küme önbelleğinde çalışacağı şekilde dolaşması için sağlanır.

Yeni kuyruk yöneticileri için, dinamik önbelleği desteklemeyen bir küme iş yükü çıkışı kullanmayacaksanız, CLCACHE=DYNAMIC ayarını yapın.

Durağan önbellek kullanan ve çok sayıda yeni kuyruk ve kuyruk yöneticisi eklenmemiş bir kümede bulunan var olan kuyruk yöneticileri için, CLCACHE=STATIC komutunu kullanmaya devam etmek makul olur.

Durağan önbellek kullanan ve ona birçok yeni kuyruk ya da kuyruk yöneticisi eklenecek bir kümede bulunan kuyruk yöneticileri için CLCACHE=DYNAMIC değerini kullanmaya başlayın.

## **STATIC**

Küme önbelleği durağan olduğunda, büyüklüğü kuyruk yöneticisi başlatma sırasında sabitlenerek, yürürlükteki küme bilgisi miktarı artı genişletme alanı için yeterli olur. Kuyruk yöneticisi etkinken büyüklük artırılmaz. Bu varsayılandır.

## **Devingen**

Küme önbelleği dinamik olduğunda, kuyruk yöneticisi başlatılırken ayrılan ilk büyüklük, kuyruk yöneticisi etkinken gerektiğinde otomatik olarak artırılabilir.

## **CMDUSER**

Komut güvenliği denetimleri için kullanılan varsayılan kullanıcı kimliğini belirtir. Bu kullanıcı kimliği ESM ' ye tanımlanmalıdır (örneğin, RACF ). 1-8 alfasayısal karakterden oluşan bir ad belirleyin. İlk karakter olarak bir harf kullanılmalıdır.

Varsayılan değer CSQOPR ' dır.

## **EXCLMSG**

Dışlanacak hata iletilerinin listesini belirtir.

Bu liste dinamiktir ve SET SYSTEM komutu kullanılarak güncellenir.

Varsayılan değer boş bir list () değeridir.

İletiler CSQ öneki ve işlem kodu soneki (I-D-E-A) olmadan sağlanır. Örneğin, CSQX500Iiletisini dışlamak için bu listeye X500 ekleyin. Bu liste en çok 16 ileti tanıtıcısı içerebilir.

Listeye eklenmeye hak kazanmak için, ileti MSTR ya da CHIN adres boşlukları normal başlatıldıktan sonra yayınlanmalı ve E, H, I, J, L, M, N, P, R, T, V, W, X, Y, 2, 3, 5, 9 karakterlerinden biriyle başlamalıdır.



İşleme komutlarının sonucu olarak yayınlanan ileti tanıtıcıları listeye eklenebilir, ancak listeye eklenmez. Örneğin, DISPLAY USAGE PSID (\*) komutunun sonucu olarak bir ileti tanıtıcısı yayınlanır, ancak bu ileti gizlenemez.

### ÇIKIŞ

Kuyruk yöneticisi çıkışlarının her çağrılması için izin verilen süreyi saniye cinsinden belirtir. (Bu parametrenin kanal çıkışları üzerinde etkisi yoktur.)

5-9999 aralığında bir değer belirleyin.

Varsayılan 30'dur. Kuyruk yöneticisi, her 30 saniyede bir çalışan çıkışları yoklar. Her ankette, EXITLIM tarafından belirlenen süreden daha uzun süre çalışan tüm anketler zorla sonlandırılır.

### ÇIKIŞ

Kuyruk yöneticisinde çıkışları çalıştırmak için kullanılacak başlatılan sunucu görevlerinin sayısını belirtir. (Bu parametrenin kanal çıkışları üzerinde etkisi yoktur.) Kuyruk yöneticisinin çalışması gereken çıkış sayısı üst sınırı (kanal çıkışları dışında) kadar yüksek bir sayı belirtmelisiniz; tersi durumunda, 6c6 olağandışı bitmesiyle başarısız olur.

Sıfır ile 99 arasında bir değer belirleyin. Sıfır değeri, çıkış çalıştırılmadığı anlamına gelir.

Varsayılan değer 8 'dir.

### YÜKLEME

IBM MQ ' in bir denetim noktasının başlangıcı ile sonraki arasında yazdığı günlük kaydı sayısını belirtir. IBM MQ , belirlediğiniz kayıt sayısı yazıldıktan sonra yeni bir denetim noktası başlatır.

200-16 000 000 aralığında bir değer belirtin.

Varsayılan değer 500 000 'dir.

Değer ne kadar büyük olursa, IBM MQ performansı da o kadar iyi olur; Ancak, parametre büyük bir değere ayarlanırsa yeniden başlatma daha uzun sürer.

Önerilen ayarlar:

<b>Test sistemi</b>	10 000
<b>Üretim sistemi</b>	500 000

Bir üretim sisteminde, sağlanan varsayılan değer çok yüksek bir denetim noktası sıklığıyla sonuçlanabilir.

LOGLOAD değeri, kuyruk yöneticisi denetim noktalarının sıklığını belirler. Çok büyük bir değer, denetim noktaları arasında günlüğe büyük miktarda veri yazılması anlamına gelir ve bu, bir hatanın ardından kuyruk yöneticisi ileriye işleyerek kurtarma yeniden başlatma süresini artırmasına neden olur. Çok küçük bir değer, yoğun yükleme sırasında denetim noktalarının çok sık oluşmasına neden olur ve yanıt sürelerini ve işlemci kullanımını olumsuz etkiler.

LOGLOAD için 500 000 başlangıç değeri önerilir. Saniyede 100 iletiden oluşan 1 KB kalıcı ileti hızı için (yani, kesinleştirme içeren 100 MQPUT ve kesinleştirme içeren 100 MQGET ), denetim noktaları arasındaki aralık yaklaşık 5 dakikadır.

**Not:** Bu, yalnızca bir kılavuz olarak amaçlanır ve bu parametre için en iyi değer, tek tek sistemin özelliklerine bağlıdır.

### MULCCAPT

Ölçülen Kullanım Lisansı Ücreti (MULC) tarafından kullanılan verileri toplamak için kullanılacak algoritmayı belirtir.

#### Standart

MULC, IBM MQ API MQCONN çağrısından IBM MQ API MQDISC çağrısının zamanına kadar olan süreyi temel alır.

#### İyileştirilmiş

MULC, bir IBM MQ API çağrısının başlangıcından IBM MQ API çağrısının sonuna kadar olan süreyi temel alır.

Varsayılan değer STANDARD değeridir.

## **OTMACON**

OTMA parametreleri. Bu anahtar sözcük beş konumlu parametre alır:

**OTMACON = ( Group , Member , Druexit , Age , Tpipepfx )**

### **Grup**

Bu, bu IBM MQ yönetim ortamının ait olduğu XCF grubunun adıdır.

1-8 karakter uzunluğunda olabilir ve büyük harfli karakterler girilmelidir.

Varsayılan değer boşluktur; bu, IBM MQ ' in bir XCF grubunu birleştirme girişiminde bulunmaması gerektiğini gösterir.

### **Üye**

Bu, XCF grubu içindeki bu IBM MQ yönetim ortamının üye adıdır.

1-16 karakter uzunluğunda olabilir ve büyük harflerle girilmelidir.

Varsayılan değer, 4 karakterlik kuyruk yöneticisi adıdır.

### **Druexit**

Bu, IMStarafından çalıştırılacak OTMA hedef çözünürlük kullanıcı çıkışının adını belirtir.

1-8 karakter uzunluğunda olabilir.

Varsayılan değer DFSYDRUO' dır.

Bu parametre isteğe bağlıdır; IBM MQ , IBM MQtarafından başlatılmamış bir IMS uygulamasından ileti almaktır. Ad, IMS sisteminde kodlanmış hedef çözüm kullanıcı çıkışına karşılık gelmelidir. Daha fazla bilgi için bkz. [“IMS içinde OTMA çıkışlarının kullanılması” sayfa 994.](#)

### **Yaş**

Bu, IBM MQ kullanıcı kimliğinin daha önce IMStarafından doğrulandığı kabul edilen süreyi saniye cinsinden gösterir.

0-2 147 483 647 aralığında olabilir.

Varsayılan değer 2 147 483 647 'dir.

Anabilgisayarda güvenlik önbellegi ayarlarının tutarlılığını korumak için ALTER SECURITY komutunun `interval` parametresiyle birlikte bu parametreyi ayarlamanız önerilir.

### **Tpipepfx**

Bu, Tpipe adları için kullanılacak öneki gösterir.

Üç karakterden oluşur; ilk karakter A-Z aralığındadır, sonraki karakterler A-Z ya da 0-9 arasındadır. Varsayılan değer CSQ 'dur.

Bu, IBM MQ her Tpipe oluşturduğunda kullanılır; adın geri kalanı IBM MQtarafından atanır. IBM MQtarafından oluşturulan herhangi bir Tpipe için tam Tpipe adını ayarlayamazsınız.

## **QINDXBLD**

Kuyruk yöneticisi yeniden başlatmasının, tüm kuyruk izinleri yeniden oluşturuluncaya kadar mı, yoksa tüm izinler yeniden oluşturulmadan önce mi tamamlanacağını belirler.

### **Bekle**

Kuyruk yöneticisi yeniden başlatma işlemi, tüm kuyruk dizini oluşturmalarının tamamlanmasını bekler. Bu, herhangi bir uygulamanın kuyruk yöneticisine bağlanabilmesi için önce tüm izinler oluşturulduğu için, izin oluşturulurken normal IBM MQ API işlemi sırasında geciken uygulama olmadığı anlamına gelir.

Bu varsayılandır.

### **HAYIR BEKLE**

Kuyruk yöneticisi, tüm kuyruk dizini oluşturma işlemi tamamlanmadan önce yeniden başlayabilir.

## QMCCSID

Kuyruk yöneticisinin (ve dağıtılmış kuyruğa alma işleminin) kullanacağı varsayılan kodlanmış karakter takımı tanıtıcısını belirler.

Sıfır ile 65535 aralığında bir değer belirleyin. Değer, Ulusal dillerdeseçtiğiniz dil için yerel z/OS kod sayfası olarak listelenen bir EBCDIC kod sayfasını temsil etmelidir.

Varsayılan değer olan sıfır, belirlenen CCSID ' yi kullanın ya da değer belirlenmezse CCSID 500 'ü kullanın. Bu, CCSID 'yi sıfır dışında bir değere belirttik olarak ayarladıysanız, QMCCSID değerini sıfır olarak ayarlayarak ilk durumuna getiremezsiniz; şimdi doğru sıfır olmayan CCSID' yi kullanmanız gerekir. QMCCSID değeri sıfırsa, DISPLAY QMGR CCSID komutunu vererek, gerçekte hangi CCSID ' nin kullanılmakta olduğunu denetleyebilirsiniz.

**Not:** Bir kuyruk paylaşım grubundaki tüm kuyruk yöneticileri aynı QMCCSID ' yi kullanmalıdır.

## QSGDATA

Kuyruk paylaşım grubu verileri. Bu anahtar sözcük beş konumlu parametre alır:

**QSGDATA = ( Qsgname , Dsgname , Db2name , Db2serv , Db2blob)**

### Qsgname

Bu, kuyruk yöneticisinin ait olduğu kuyruk paylaşım grubunun adıdır.

Geçerli karakterler için IBM MQ nesnelere adlandırma kuralları konusuna bakın. Ad:

- 1-4 karakter uzunluğunda olabilir
- Sayısal bir değerle başlamamalıdır
- @ ile bitmemelidir.

Bunun nedeni, somutlama nedenleriyle, dört karakterden az olan adların @ simgeleriyle içeriden doldurulması,

Varsayılan değer, kuyruk yöneticisinin herhangi bir kuyruk paylaşım grubunun üyesi olmadığını gösteren boşluklardır.

### Dsgname

Bu, kuyruk yöneticisinin bağlanacağı Db2 veri paylaşım grubunun adıdır.

1-8 karakter uzunluğunda olabilir ve büyük harfli karakterler girilmelidir.

Varsayılan değer, kuyruk paylaşım gruplarını kullanmadığınızı gösteren boşluklardır.

### Db2name

Bu, kuyruk yöneticisinin bağlanacağı Db2 altsisteminin ya da grup bağlantısının adıdır.

1-4 karakter uzunluğunda olabilir ve büyük harflerle girilmelidir.

Varsayılan değer, kuyruk paylaşım gruplarını kullanmadığınızı gösteren boşluklardır.

**Not:** Db2 altsistemi (ya da grup bağlantısı), Dsgname içinde belirtilen Db2 veri paylaşım grubunda olmalıdır ve tüm kuyruk yöneticileri aynı Db2 veri paylaşım grubunu belirtmelidir.

### Db2serv

Bu, Db2' e erişmek için kullanılan sunucu görevlerinin sayısıdır.

4-10 aralığında olabilir.

Varsayılan 4 'tür.

### Db2blob

Bu, İkili Büyük Nesnelere (BLOB) erişmek için kullanılan Db2 görevlerinin sayısıdır.

4-10 aralığında olabilir.

Varsayılan 4 'tür.

Ad değiştirgelerinden birini ( **Qsgname**, **Dsgname** ya da **Db2name** ) belirtirseniz, diğer adlar için değer girmeniz gerekir; tersi durumda IBM MQ başarısız olur.

## İLK DURUMA GEÇİR

Bağlantı işleme sırasında gerçekleştirilen RESLEVEL güvenlik denetimleri için RACF denetim kayıtlarının yazılıp yazılmayacağını belirtir.

Aşağıdakilerden birini belirtin:

### HAYIR

RESLEVEL denetimi gerçekleştirilmedi.

### EVET

RESLEVEL denetimi gerçekleştirilir.

Varsayılan değer Evet'tir.

## ROUTCDE

Bir MQSC komutuna doğrudan yanıt olarak gönderilmeyen iletilere atanan varsayılan z/OS ileti yöneltme kodunu belirtir.

Aşağıdakilerden birini belirtin:

- 1-16 aralığındaki bir değer (bu değerler de içinde olmak üzere).
- Virgülle ayrılmış ve parantez içinde yer alan değerlerin listesi. Her değer, 1-16 aralığında (bu değerler de içinde olmak üzere) olmalıdır.

Varsayılan değer 1'dir.

z/OS yöneltme kodlarıyla ilgili daha fazla bilgi için *z/OS MVS System Messages* elkitablarının birimlerinden birinde [Message description](#) içindeki *Routing codes* (Yönlendirme kodları) başlıklı konuya bakın.

## HİZMET

Bu alan IBM tarafından kullanılmak üzere ayrılmıştır.

## SMFACCT

Kuyruk yöneticisi başlatıldığında IBM MQ 'un muhasebe verilerini SMF' ye otomatik olarak gönderip göndermeyeceğini belirtir.

Aşağıdakilerden birini belirtin:

### HAYIR

Hesap verilerini otomatik olarak toplamaya başlamayın.

### EVET

Varsayılan sınıf 1 için hesap verilerini otomatik olarak toplamaya başlayın.

### Tamsayılar

**V9.3.0** Muhasebe verilerinin otomatik olarak 1-4 aralığında toplandığı sınıfların listesi.

**V9.3.0** \* 1, 2 ve 3. sınıflar için SMF muhasebesini otomatik olarak başlatın.

Varsayılan değer NO değeridir.

## SMFSTAT

Kuyruk yöneticisi başlatıldığında SMF istatistiklerinin otomatik olarak toplanıp toplanmayacağını belirtir.

Aşağıdakilerden birini belirtin:

### HAYIR

İstatistikleri otomatik olarak toplamaya başlamayın.

### EVET

Varsayılan sınıf 1 için istatistikleri otomatik olarak toplamaya başlayın.

### Tamsayılar

**V9.3.0** İstatistiklerin otomatik olarak 1-5 aralığında toplandığı sınıfların listesi.

Sınıf 2 ya da 3 istatistiklerini toplamak için sınıf 1 de belirtilmelidir.

**V 9.3.0**

\* 1, 2 ve 3. sınıflar için SMF istatistiklerini otomatik olarak başlatın.

Varsayılan değer NO değeridir.

**SPLCAP**

Güvenlik ilkesi yeteneği, iletilerin yazıldığı ve kuyruklardan okunduğu gibi imzalandığını ya da şifrelendiğini denetleyen ilkeler aracılığıyla daha yüksek düzeyde ileti güvenliği sağlar.

SPLCAP aşağıdaki değerlerden birine ayarlanarak, bu kuyruk yöneticisi için güvenlik ilkesi işleme yapılandırılır:

**HAYIR**

Kuyruk yöneticisi kullanıma hazırlanırken kuyruklar için ileti güvenliği ilkeleri uygulama yeteneği etkinleştirilmez.

**EVET**

Kuyruk yöneticisi kullanıma hazırlanırken ileti güvenliği yetenekleri etkinleştirilir.

Kuyruk yöneticisi, AMSPROD özneteliğinin AMS, ADVANCED ya da ADVANCED özneteliklerinden birine ayarlı olup olmadığını denetler; bu durumda, AMS için lisanslanmıştır. Yoksa başlamaz.

Kuyruk yöneticisi, gerekli AMS yapılandırmasının yerinde olup olmadığını da denetler. Değilse, kuyruk yöneticisi başlamaz.

Kuyruk yöneticisinin her ikisi de AMS için lisanslandıysa ve gerekli yapılandırma varsa, kuyruk yöneticisi, kuyruk yöneticisinin kullanıma hazırlanması sırasında etkinleştirilen ileti güvenliği yetenekleriyle başlar ve AMSM adres alanı başlatılır.

Varsayılan değer NO değeridir.

**DURUM****V 9.3.0**

IBM MQ for z/OS 9.3.0' dan, istatistik verilerinin ardışık toplamaları arasındaki süreyi dakika ve saniye cinsinden belirtir. ACCTIME ayarlanmamışsa ya da -1 ise, muhasebe verilerinin ardışık toplamaları arasındaki süreyi de belirtir.

0 -1440 dakika 'mmmm' aralığında ya da 0 -1440 dakika ve 0 -59 saniye aralığında 'mmmm.ss' biçiminde bir sayı belirtin. Varsayılan değer 30 dakikadır.

**Notlar:**

- Yalnızca saniye aralığı belirtirken, aralığın başına 0 önekini eklemeniz gerekir. Olası en küçük aralık bir saniyedir: '0.01'.
- **V 9.3.0** IBM MQ for z/OS 9.3.0' den 0 değerini belirlerseniz, istatistik verileri SMF veri toplama yayınında toplanır. ACCTIME belirtilmezse ya da -1 ise, SMF veri toplama yayınında muhasebe verileri de toplanır. Ek bilgi için [Using System Management Facility](#) başlıklı konuya bakın.
- Varsayılan değer olan -1 değerini belirtirseniz, muhasebe verileri STATIME değeri tarafından belirlenen aralıkta toplanır.

**TRACSTR**

Genel izlemenin otomatik olarak başlatılıp başlatılmayacağını belirtir.

Aşağıdakilerden birini belirtin:

**HAYIR**

Genel izlemeyi otomatik olarak başlatmayın.

**EVET**

Varsayılan sınıf (sınıf 1) için genel izlemeyi otomatik olarak başlat.

**Tamsayılar**

1-4 aralığında, genel izlemenin otomatik olarak başlatılacağı sınıfların listesi.

\*

Tüm sınıflar için genel izlemeyi otomatik olarak başlat.

Makroda anahtar sözcüğü belirtmezseniz, varsayılan değer NO olur.

**Not:** Sağlanan varsayılan sistem parametresi yükleme modülünde (CSQZPARM) TRACSTR=YES (CSQFSYSP çevirici modülünde ayarlanır) değeri var. İzlemeyi otomatik olarak başlatmak istemiyorsanız, kendi sistem değiştirgesi biriminizi yaratın ya da kuyruk yöneticisi başlatıldıktan sonra STOP TRACE komutunu verin.

STOP TRACE komutuna ilişkin ayrıntılar için [STOP TRACE](#) konusuna bakın.

### TRACTBL

Genel izleme olanağının IBM MQ izleme kayıtlarını sakladığı izleme çizelgesinin varsayılan büyüklüğünü 4 KB ' lik blok cinsinden belirtir.

1-999 aralığında bir değer belirleyin.

Varsayılan değer 99 'dur. Bu, 396 KB ' ye eşdeğerdir.

**Not:** İzleme çizelgesine ilişkin saklama alanı ECSA ' da ayrılır. Bu nedenle, bu değeri dikkatli bir şekilde seçmeniz gerekir.

### WLMTIME

WLM tarafından yönetilen kuyruklara ilişkin dizinlerin her taraması arasındaki süreyi (WLMTIMU değerine bağlı olarak dakika ya da saniye cinsinden) belirler.

1-9999 aralığında bir değer belirleyin.

Varsayılan 30'dur.

### WLMTIMU

WLMTIME parametresiyle kullanılan zaman birimleri.

Aşağıdakilerden birini belirtin:

#### **dak**

WLMTIME, dakika sayısını gösterir.

#### **saniye**

WLMTIME saniye sayısını gösterir.

Varsayılan değer MINS 'dir.

### İlgili başvurular

[“CSQ6LOGP ' nin kullanılması” sayfa 914](#)

CSQ6LOGPkullanılarak günlüğe kaydetme seçeneklerinin nasıl belirtileceğini öğrenmek için bu konuyu kullanın.

[“CSQ6ARVP ' nin kullanılması” sayfa 919](#)

CSQ6ARVP kullanarak arşivleme ortamınızın nasıl belirtileceğini belirtmek için bu konuyu başvuru olarak kullanın.

### CSQ6LOGP ' nin kullanılması

CSQ6LOGPkullanılarak günlüğe kaydetme seçeneklerinin nasıl belirtileceğini öğrenmek için bu konuyu kullanın.

Günlüğe kaydetme seçeneklerinizi belirlemek için CSQ6LOGP komutunu kullanın.

CSQ6LOGPiçin varsayılan değiştirgeler ve SET LOG komutunu kullanarak her değiştirgeyi değiştirip değiştiremeyeceğiniz CSQ6LOGP değiştirgelerinin varsayılan değerleri içinde gösterilir. Bu değerlerden herhangi birini değiştirmeniz gerekirse, parametrelerin ayrıntılı açıklamalarına bakın.

Çizelge 60. CSQ6LOGP değiştirgelerinin varsayılan değerleri			
Değiştirge	Açıklama	Varsayılan değer	SET komutu
ŞİRKET Günlüğü	Günlük sıkıştırmanın etkinleştirilip etkinleştirilmediğini denetler.	YOK	X

Çizelge 60. CSQ6LOGP değıştirgelerinin varsayılan değeri (devamı var)

Değıştirge	Açıklama	Varsayılan değeri	SET komutu
<u>DEALLCT</u>	Arşiv manyetik bant biriminin serbest bırakılmadan önce kullanılmayan süre.	sıfır	X
<u>BAĞLAMAZ</u>	Etkin ve arşiv günlüğü veri kümeleri için giriş arabelleği saklama alanı büyüklüğü.	60 KB	-
<u>MAXARCH</u> (MAXARCH)	Kaydedilebilecek arşiv günlüğü birimi sayısı üst sınırı.	500	X
<u>MAXCNOFF</u>	Koşut olarak çalıştırılacak CSQJOFF7 boşaltma görevi sayısı üst sınırı.	31	-
<u>MAXRTU</u>	Arşiv günlüğü manyetik bant birimlerini eşzamanlı olarak okumak için ayrılan adanmış manyetik bant birimi sayısı üst sınırı.	2	X
<u>YÜKLEME</u>	Arşivleme açık ya da kapalı.	EVET (Açık)	-
<u>ÇIKIŞ</u>	Etkin ve arşiv günlüğü veri kümeleri için çıkış arabelleği saklama alanı büyüklüğü.	4 000 KB	-
<u>TWOACTV</u>	Tek ya da çift etkin günlük kaydı.	YES (ikili)	-
<u>İKİ ARAMA</u>	Tek ya da çift arşiv günlük kaydı.	YES (ikili)	-
<u>TWOBSDS</u>	Tek ya da çift BSIDS.	EVET (çİFT BSIDS)	-
<u>YANLIŞ ITME</u>	Etkin günlük veri kümelerine yazılmadan önce doldurulacak çıkış arabelleklerinin sayısı.	20	X
<u>ZHYAZIN</u>	zHyperYazma özelliğinin etkinleştirilip etkinleştirilmediğini belirtir.	HAYIR	X

### DERLEME GÜNLÜĞÜ

Günlük sıkıştırmanın etkinleştirilip etkinleştirilmediğini belirtir.

Aşağıdakilerden birini belirtin:

#### YOK

Günlük sıkıştırma etkinleştirilmedi.

#### RLE

Günlük sıkıştırma, çalıştırma uzunluğu kodlaması kullanılarak etkinleştirilir.

#### Fark Etmez

Kuyruk yöneticisi, en yüksek günlük kaydı sıkıştırma derecesini veren sıkıştırma algoritmasını seçer. Bu seçenek RLE sıkıştırmayla sonuçlanır.

Varsayılan değeri NONE değeridir.

Günlük sıkıştırmayla ilgili daha fazla ayrıntı için bkz. [Günlük sıkıştırması](#).

### DEALLCT (KESİK)

Arşiv okuma manyetik bant biriminin serbest bırakılmadan önce kullanılmamasına izin verilen süreyi dakika cinsinden belirtir.

Aşağıdakilerden birini belirtin:

- Süre, dakika cinsinden, sıfır ile 1440 aralığında
- NOLIMIT

1440 ya da NOLIMIT deęerinin belirlenmesi, manyetik bant biriminin hiçbir zaman serbest bırakılmadıęı anlamına gelir.

Varsayılan deęer sıfırdır.

Arşiv günlüęü verileri manyetik banttan okunurken, bu deęeri IBM MQ ' in birden çok okuma uygulaması için manyetik bant işlemeyi eniyilemesini sağlayacak kadar yüksek bir deęere ayarlamanız önerilir.

### **INBUFF**

Kurtarma sırasında etkin ve arşiv günlüklerini okumak için kullanılan giriş arabelleęinin büyüklüęünü kilobayt cinsinden belirler. 28-60 aralıęında bir ondalık sayı kullanın. Belirtilen deęer, 4 'ün katlarına yuvarlanır.

Varsayılan deęer 60 KB 'dir.

Önerilen ayarlar:

**Test sistemi** 28 KB

**Üretim sistemi** 60 KB

Bu deęeri, en iyi günlük okuma başarımı üst sınırına ayarlayın.

### **MAXARCH**

BSDS ' ye kaydedilebilecek arşiv günlüęü birimi sayısı üst sınırını belirtir. Bu sayı aşıldıęında, kayıt BSDS ' nin bařlangıcında yeniden bařlar.

10-1000 aralıęında bir ondalık sayı kullanın.

Varsayılan deęer 500 'dür.

Önerilen ayarlar:

**Test sistemi** 500 (varsayılan)

**Üretim sistemi** 1 000

BSDS ' nin mümkün olduęunca çok günlük kaydedebilmesi için bunu üst sınıra ayarlayın.

Günlükler ve BSDS ile ilgili bilgi için [IBM MQ kaynaklarının yönetilmesi](#) başlıklı konuya bakın.

### **MAXCNOFF**

Kořut olarak çalıştırılacak CSQJOFF7 boşaltma görevlerinin sayısını belirtir.

Bu, bir kuyruk yöneticisinin ya da kuyruk yöneticilerinin, kullanılabilir tüm manyetik bant birimlerini kullanmayacak şekilde ayarlanmasını sağlar.

Kuyruk yöneticisi, yeni arşiv veri kümeleri ayırmayı denemeden önce CSQJOFF7 boşaltma görevi tamamlanıncaya kadar bekler.

Kuyruk yöneticisi manyetik banda arşivleniyorsa, kořutzamanlı manyetik bant isteklerinin sayısı, kullanılabilir manyetik bant birimlerinin sayısına eřit ya da bu sayıyı aşmayacak şekilde bu parametreyi ayarlayın; tersi durumda sistem askıda kalacaktır.

İkili arşivleme kullanılırsa, her boşaltma görevinin her iki arşivi de gerçekleřtirdięini ve deęiřtirgenin buna göre ayarlanması gerektięini unutmayın. Örneęin, kuyruk yöneticisi manyetik banda ikili arşivleme yaparsa, MAXCNOFF=2 deęeri en çok iki etkin günlüęün eşzamanlı olarak dört manyetik banda arşivlenmesine izin verir.

Birkaç kuyruk yöneticisi manyetik bant birimlerini paylaşıyorsa, her kuyruk yöneticisi için MAXCNOFF deęerini buna göre ayarlamanız gerekir.

Varsayılan deęer 31 'dir.

1-31 aralıęında bir deęer belirtin.



## MAXRTU

Arşiv günlüğü manyetik bant birimlerini eşzamanlı olarak okumak için ayrılabilir adanmış manyetik bant birimi sayısı üst sınırını belirtir.

Bu parametre ve DEALLCT parametresi, IBM MQ ' in manyetik bant aygıtlarından arşiv günlüğü okumasını eniyilemesini sağlar.

1-99 aralığında bir değer belirtin.

Varsayılan, 2'dir.

Değeri, IBM MQ' in kullanabileceği manyetik bant birimi sayısından en az bir az bir olacak şekilde ayarlamamız önerilir. Tersi durumda, boşaltma işlemi gecikebilir ve bu da sisteminizin başarımını etkileyebilir. Arşiv günlüğü işleme sırasında verim üst sınırı için, boşaltma işlemi için en az bir manyetik bant birimine gereksinim duyduğunuz anımsayarak, bu seçenek için mümkün olan en büyük değeri belirtin.

## YÜKLEME

Arşivlemenin açık ya da kapalı olduğunu belirtir.

Aşağıdakilerden birini belirtin:

### EVET

Arşivleme açık

### HAYIR

Arşivleme kapalı

Varsayılan değer Evet'tir.

**Dikkat:** Bir test ortamında çalışmadığınız sürece arşivlemeyi **kapatmayın** . Bunu kapatırsanız, bir sistem ya da hareket hatası durumunda verilerin kurtarılacağına garantisini veremezsiniz.

## ÇIKIŞ

Etkin ve arşiv günlüğü veri kümelerini yazmak için IBM MQ tarafından çıkış arabellekleri için kullanılacak saklama alanının toplam büyüklüğünü kilobayt cinsinden belirtir. Her çıkış arabelleği 4 KB 'dir.

Parametre, 128-4000 aralığında olmalıdır. Belirtilen değer, 4 'ün katlarına yuvarlanır. Uyumluluk nedeniyle 40 ile 128 arasındaki değerler kabul edilir ve 128 değeri olarak kabul edilir.

Varsayılan değer 4000 KB 'dir.

Önerilen ayarlar:

<b>Test sistemi</b>	400 KB
<b>Üretim sistemi</b>	4 000 KB

Günlük çıkış arabelleklerinin tükenmesini önlemek için bu değeri üst sınıra ayarlayın.

## TWOACTV

Tek ya da çift etkin günlük kaydını belirtir.

Aşağıdakilerden birini belirtin:

### HAYIR

Tek etkin günlük

### EVET

İkili etkin günlükler

Varsayılan değer Evet'tir.

Tek ve ikili günlük kaydı kullanımı hakkında daha fazla bilgi için bkz. [IBM MQ kaynaklarını yönetme](#).

## ARAMA

Etkin günlük boşaltıldığında IBM MQ ' in ürettiği arşiv günlüklerinin sayısını belirtir.

Aşağıdakilerden birini belirtin:

**HAYIR**

Tek arşiv günlükleri

**EVET**

İkili arşiv günlükleri

Varsayılan değer Evet'tir.

Önerilen ayarlar:

**Test sistemi** HAYIR

**Üretim sistemi** YES (varsayılan)

Tek ve ikili günlük kaydı kullanımı hakkında daha fazla bilgi için bkz. [IBM MQ kaynaklarını yönetme](#).

**TWOBSDS**

Önyükleme veri kümelerinin sayısını belirtir.

Aşağıdakilerden birini belirtin:

**HAYIR**

Tek BSDS

**EVET**

İkili BSDS

Varsayılan değer Evet'tir.

Tek ve ikili günlük kaydı kullanımı hakkında daha fazla bilgi için bkz. [IBM MQ kaynaklarını yönetme](#).

**YANLIŞ**

Etkin günlük veri kümelerine yazılmadan önce doldurulacak 4 KB çıkış arabelleklerinin sayısını belirtir.

Arabelleklerin sayısı ne kadar büyük olursa, yazma da o kadar az gerçekleşir ve bu, IBM MQ başarımını artırır. Kesinleştirme noktası gibi önemli olaylar oluşursa, arabellekler bu sayıya ulaşılmadan önce yazılabilir.

1-256 aralığındaki arabelleklerin sayısını belirleyin.

Varsayılan değer 20 'dir.

**ZHY Yazın**

Etkin günlüklere zHyperYazma etkinleştirilirken yazma işleminin yapılıp yapılmayacağını belirtir.

zHyperWrite ile etkin günlüklerin etkinleştirilmesine ilişkin ek bilgi için [Using zHyperWrite with IBM MQ active logs](#) başlıklı konuya bakın.

Değer şöyle olabilir:

**HAYIR**

zHyperYazma etkinleştirilmedi.

**EVET**

zHyperYazma etkinleştirildi.

**İlgili başvurular**

[“CSQ6SYSP ' nin kullanılması” sayfa 905](#)

CSQ6SYSP kullanılarak sistem parametrelerinin nasıl ayarlanacağına ilişkin başvuru olarak bu konuyu kullanın.

[“CSQ6ARVP ' nin kullanılması” sayfa 919](#)

CSQ6ARVP kullanarak arşivleme ortamınızın nasıl belirtileceğini belirtmek için bu konuyu başvuru olarak kullanın.

CSQ6ARVP kullanarak arşivleme ortamınızın nasıl belirtileceğini belirtmek için bu konuyu başvuru olarak kullanın.

Arşivleme ortamınızı oluşturmak için CSQ6ARVP komutunu kullanın.

CSQ6ARVP için varsayılan parametreler ve SET ARCHIVE komutunu kullanarak her parametreyi değiştirip değiştiremeyeceğiniz Çizelge 61 sayfa 919 içinde gösterilir. Bu değerlerden herhangi birini değiştirmeniz gerekirse, parametrelerin ayrıntılı açıklamalarına bakın. Depolamanın planlanmasıyla ilgili daha fazla bilgi için bkz. z/OS üzerinde depolama ve performans gereksinimlerinizin planlanması.

Çizelge 61. CSQ6ARVP değiştirgelerinin varsayılan değerleri			
Değiştirge	Açıklama	Varsayılan değer	SET komutu
<u>ALCUNIT</u>	Birincil ve ikincil alan ayırmaları yapılan birimler.	BLK (bloklar)	X
<u>ARCPFX1</u>	İlk arşiv günlüğü veri kümesi adının öneki.	CSQARC1	X
<u>ARCPFX2</u>	İkinci arşiv günlüğü veri kümesi adının öneki.	CSQARC2	X
<u>ARCRETN</u>	Arşiv günlüğü veri kümesinin alıkoyma süresi (gün).	9999	X
<u>ARCWRTC</u>	Arşiv günlüğü veri kümeleriyle ilgili işletmene iletilerin rota kodlarının listesi.	1.3.4	X
<u>ARCWTOR</u>	Bir arşiv günlüğü veri kümesini bağlamayı denemeden önce işletmene ileti gönderilip gönderilmeyeceğini ve yanıtın beklenip beklenmeyeceğini belirler.	EVET	X
<u>SIYAH BÜYÜKLÜK</u>	Arşiv günlüğü veri kümesinin blok boyutu.	28 672	X
<u>KATALOG</u>	Arşiv günlüğü veri kümelerinin ICF ' de kataloğa alınıp alınmayacağını belirler.	HAYIR	X
<u>Temel</u>	Arşiv günlüğü veri kümelerinin sıkıştırılıp sıkıştırılmayacağını.	HAYIR	X
<u>PRIQTY</u>	DASD veri kümeleri için birincil alan ayırma.	25 715	X
<u>KORU</u>	Arşiv günlüğü veri kümelerinin, veri kümeleri yaratıldığında ESM tanımları tarafından korunup korunmayacağını belirler.	HAYIR	X
<u>QUIESCE</u>	MODE (QUIESCE) ile ARCHIVE LOG belirlendiğinde saniye cinsinden susturma için izin verilen süre üst sınırı.	5	X
<u>SECQTY</u>	DASD veri kümeleri için ikincil alan ayırma. Kullanılacak birimler için ALCUNIT parametresine bakın.	540	X
<u>DAMGA</u>	Arşiv veri kümesi adının bir zaman damgası içerip içermeyeceğini belirler.	HAYIR	X
<u>birim</u>	Arşiv günlüğü veri kümelerinin ilk kopyasının saklandığı aygıt tipi ya da birim adı.	MANYETIK BANT	X
<u>UNIT2</u>	Arşiv günlüğü veri kümelerinin ikinci kopyasının saklandığı aygıt tipi ya da birim adı.	Boş	X

**ALCUNIT.**

Birincil ve ikincil alan ayırmaları yapılan birimi belirtir.

Aşağıdakilerden birini belirtin:

**CYL**

Silindirler

**trk**

İzler

**BLK**

Bloklar

BLK, aygıt tipinden bağımsız olduğu için kullanmanız önerilir.

Varsayılan değer BLK ' dir.

Arşiv DASD birimlerinde boş alan büyük olasılıkla bölünecek ise, daha küçük bir birincil kapsam belirtmeniz ve ikincil kapsamlara genişletilmesine izin vermeniz önerilir. Etkin günlüklere ilişkin alan ayırma hakkında daha fazla bilgi için [Günlük arşivi depolamanın planlanması](#) başlıklı konuya bakın.

**ARCPFX1**

İlk arşiv günlüğü veri kümesi adının önekini belirtir.

Veri kümelerinin nasıl adlandırıldığına ve ARCPFX1 uzunluğuna ilişkin kısıtlamalar için TSTAMP parametresine bakın.

Bu parametre boş bırakılamaz.

Varsayılan değer CSQARC1' dir.

Bu önekle arşiv günlükleri oluşturmak için IBM MQ kuyruk yöneticisi adres alanıyla ilişkili kullanıcı kimliğini yetkilendirmeniz gerekebilir.

**ARCPFX2**

İkinci arşiv günlüğü veri kümesi adının önekini belirtir.

Veri kümelerinin nasıl adlandırıldığına ve ARCPFX2 uzunluğuna ilişkin kısıtlamalar için TSTAMP parametresine bakın.

TWOARCH parametresi NO olarak belirtile bile bu parametre boş olamaz.

Varsayılan değer CSQARC2' dir.

Bu önekle arşiv günlükleri oluşturmak için IBM MQ kuyruk yöneticisi adres alanıyla ilişkili kullanıcı kimliğini yetkilendirmeniz gerekebilir.

**ARCRETN**

Arşiv günlüğü veri kümesi yaratıldığında kullanılacak alıkoyma süresini gün olarak belirtir.

Parametre, sıfır ile 9999 aralığında olmalıdır.

Varsayılan değer 9999 'dur.

Önerilen ayarlar:

**Test sistemi**

3

Bir test sisteminde, arşiv günlükleri büyük olasılıkla uzun dönemler boyunca gerekli değildir.

**Üretim sistemi**

9 999 (varsayılan)

Otomatik arşiv günlüğü silme işlemini etkili bir şekilde kapatmak için bu değeri yüksek bir değere ayarlayın.

Arşiv günlüğü veri kümelerinin atılması hakkında daha fazla bilgi için bkz. [Arşiv günlüğü veri kümelerinin atılması](#).

## ARCWRTC

Arşiv günlüğü veri kümeleriyle ilgili iletilere ilişkin z/OS yöneltme kodlarının listesini işletmene belirtir. ARCWTOR NO olarak ayarlanırsa bu alan yoksayılır.

Her biri 1-16 aralığında bir değere sahip en çok 14 yöneltme kodu belirleyin. En az bir kod belirtmelisiniz. Listedeki kodları boşluklarla değil, virgüllerle ayırın.

Varsayılan değer, değer listesidir: 1,3,4.

z/OS yöneltme kodlarıyla ilgili daha fazla bilgi için z/OS MVS System Messages elkitablarının birimlerinden birinde *Message description* içindeki *Routing codes* (Yönlendirme kodları) başlıklı konuya bakın.

## ARCWTOR

Bir arşiv günlüğü veri kümesini bağlama girişiminde bulunmadan önce işletmene bir iletinin gönderilip gönderilmeyeceğini ve bir yanıtın alınıp alınmayacağını belirtir.

Diğer IBM MQ kullanıcıları veri kümesi bağlanıncaya kadar beklemeye zorlanabilirler, ancak IBM MQ iletiye yanıt beklerken bu kullanıcılar etkilenmezler.

Aşağıdakilerden birini belirtin:

### EVET

Aygıt, arşiv günlüğü veri kümelerini bağlamak için uzun bir zamana gereksinim duyar. Örneğin, bir manyetik bant sürücüsü.

### HAYIR

Cihazda uzun gecikme yok. Örneğin, DASD.

Varsayılan değer Evet'tir.

Önerilen ayarlar:

**Test sistemi** HAYIR

**Üretim sistemi** YES (varsayılan)

Bu, işletim yordamlarına bağlıdır. Manyetik bant robotları kullanılırsa, NO daha uygun olabilir.

## BLKSIZE

Arşiv günlüğü veri kümesinin blok boyutunu belirtir. Belirlediğiniz blok boyutu, UNIT parametresinde belirlediğiniz aygıt tipiyle uyumlu olmalıdır.

Parametre, 4 097-28 672 aralığında olmalıdır. Belirttiğiniz değer, 4 096 'nın katlarına yuvarlanır.

Varsayılan değer 28 672 'dir.

Bu parametre, varsa, depolama yönetimi altsistemi (SMS) veri sınıfı blok boyutu tarafından geçersiz kılınır.

Arşiv günlüğü veri kümesi DASD ' ye yazılırsa, her iz için iki bloğa izin veren blok büyüklüğü üst sınırını seçmeniz önerilir. Örneğin, 3390 aygıtı için 24 576 blok boyutunu kullanmanız gerekir.

Arşiv günlüğü veri kümesi manyetik bantta yazılırsa, olası en büyük blok boyutunun belirtilmesi arşiv günlüğünü okuma hızını artırır. 28 672 'lik bir blok boyutu kullanmalısınız.

Önerilen ayarlar:

**Test sistemi** Arşiv günlükleri için kullanılan ortama bağlı olarak blok boyutu önerisini kullanın.

Yani, disk 24 576 ve manyetik bant 28 672 için.

**Üretim sistemi** Arşiv günlükleri için kullanılan ortama bağlı olarak blok boyutu önerisini kullanın.

Yani, disk 24 576 ve manyetik bant 28 672 için.

## KATALOG

Arşiv günlüğü veri kümelerinin birincil tümleşik katalog olanağı (ICF) kataloğunda kataloğa alınıp alınmayacağını belirtir.

Aşağıdakilerden birini belirtin:

### HAYIR

Arşiv günlüğü veri kümeleri kataloğa alınmadı

### EVET

Arşiv günlüğü veri kümeleri kataloğa alınmış

Varsayılan değer NO değeridir.

DASD ' de ayrılmış tüm arşiv günlüğü veri kümeleri kataloğa alınmalıdır. CATALOG değıştirgesi NO olarak ayarlanarak DASD dosyasına arşivlenirse, CSQJ072E iletisi, her arşiv günlüğü veri kümesi ayrıldığında ve IBM MQ veri kümesini kataloğa aldığında görüntülenir.

Önerilen ayarlar:

<b>Test sistemi</b>	EVET
<b>Üretim sistemi</b>	DASD ' de arşivler ayrıldığında EVET

## sıkıştırılmış

Arşiv günlüklerine yazılan verilerin sıkıştırılıp sıkıştırılmayacağını belirtir. Bu seçenek yalnızca, geliştirilmiş veri kaydetme yeteneğine (IDRC) sahip bir 3480 ya da 3490 aygıtı için geçerlidir. Bu özellik açıldığında, manyetik bant denetim birimindeki donanım verileri normalden çok daha yüksek yoğunlukta yazarak her birimde daha fazla veri sağlar. IDRC aksamıyla birlikte bir 3480 aygıtı ya da 3490Edışında bir 3490 temel modeli kullanmıyorsanız NO değerini belirtin. Verilerin sıkıştırılmasını istiyorsanız YES değerini belirleyin.

Aşağıdakilerden birini belirtin:

### HAYIR

Veri kümelerini düzenleme

### EVET

Veri kümelerini sıkıştır

Varsayılan değer NO değeridir.

YES değerinin belirlenmesi başarımı olumsuz etkiler. Ayrıca, manyetik bantta sıkıştırılan verilerin yalnızca IDRC özelliğini destekleyen bir aygıt kullanılarak okunabileceğini de unutmayın. Uzaktan kurtarma için arşiv manyetik bantlarını başka bir siteye göndermeniz gerekirse bu sorun olabilir.

Önerilen ayarlar:

<b>Test sistemi</b>	Geçerli değildir
<b>Üretim sistemi</b>	NO (varsayılan)

Bu, yalnızca 3480 ve 3490 IDR sıkıştırması için geçerlidir. Bunun YES olarak ayarlanması, kurtarma ve yeniden başlatma sırasında arşiv günlüğü okuma başarımını düşürebilir; ancak, manyetik bantta yazılmasını etkilemez.

## PRIQTY (MIKTAR)

ALCUNIT ' lerdeki DASD veri kümeleri için birincil alan ayırmayı belirtir.

Değer sıfırdan büyük olmalıdır.

Varsayılan değer 25 715 'tir.

Bu değer, hangisi daha büyükse, günlük veri kümesinin ya da buna karşılık gelen BSDS ' nin bir kopyası için yeterli olmalıdır. Gerekli değeri belirlemek için aşağıdaki yordamı izleyin:

1. “Önyükeme ve günlük veri kümelerinin yaratılması” sayfa 901’inde açıklandığı şekilde, ayrılmış etkin günlük kaydı sayısını ( c ) belirleyin.
2. Her arşiv günlüğü bloğunda 4096 baytlık blok sayısını belirleyin:

$$d = \text{BLKSIZE} / 4096$$

burada BLKSIZE, yuvarlanmış değerdir.

3. ALCUNIT = BLK ise:

$$\text{PRIQTY} = \text{INT}(c / d) + 1$$

Burada INT, bir tamsayıya yuvarlama anlamına gelir.

ALCUNIT = TRK ise:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}))) + 1$$

Burada e, her iz için bayt sayısıdır (3390 aygıtı için 56664) ve INT, bir tamsayıya yuvarlama anlamına gelir.

ALCUNIT = CYL ise:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}) * f)) + 1$$

Burada f, her silindir için iz sayısıdır (3390 aygıtı için 15) ve INT, bir tamsayıya yuvarlanır.

Günlük ve arşiv veri kümelerinizi ne kadar büyük yapacağınızla ilgili bilgi için bkz. “Önyükeme ve günlük veri kümelerinin yaratılması” sayfa 901 ve “Sayfa kümelerinizi tanımlayın” sayfa 902.

Önerilen ayarlar:

**Test sistemi**

1680

Tüm etkin günlüğü tutmak için yeterli:

$$10 \ 080 / 6 = 1 \ 680 \ \text{blocks}$$

**Üretim sistemi**

Manyetik banda arşivlenirken geçerli değildir.

Arşiv DASD birimlerinde boş alan büyük olasılıkla bölünecek ise, daha küçük bir birincil kapsam belirtmeniz ve ikincil kapsamlara genişletilmesine izin vermeniz önerilir. Etkin günlüklere ilişkin alan ayırma hakkında daha fazla bilgi için bkz. [Günlük arşivi depolamanın planlanması](#).

**KORU**

Veri kümeleri yaratıldığında arşiv günlüğü veri kümelerinin ayrı ESM (dış güvenlik yöneticisi) tanımları tarafından korunup korunmayacağını belirler.

Aşağıdakilerden birini belirtin:

**HAYIR**

Profiller oluşturulmaz.

**EVET**

Günlükler boşaltıldığında ayrı veri kümesi profilleri oluşturulur. EVET değerini belirtirseniz:

- ESM koruması IBM MQ için etkin olmalıdır.
- IBM MQ kuyruk yöneticisi adres alanıyla ilişkili kullanıcı kimliğinin bu tanımları yaratma yetkisi olmalıdır.
- Manyetik banda arşivleme için TAPEVOL sınıfı etkin olmalıdır.

Tersi durumda, boşaltma başarısız olur.

Varsayılan değer NO'dur.

### QUIESCE

MODE (QUIESCE) belirtilirken ARCHIVE LOG komutu verildiğinde susturma için izin verilen süre üst sınırını saniye cinsinden belirtir.

Parametre, 1-999 aralığında olmalıdır.

Varsayılan 5'tir.

### SECQTY

ALCUNIT 'lerdeki DASD veri kümeleri için ikincil alan ayırmasını belirtir. İkincil kapsam en çok 15 kez ayrılabilir; ALCUNIT hakkında daha fazla bilgi için [IBM z/OS Management Facility Programming Guide](#) belgesine bakın.

Parametre sıfırdan büyük olmalıdır.

Varsayılan değer 540'dur.

### DAMGA

Arşiv günlüğü veri kümesi adının içinde bir zaman damgası olup olmadığını belirtir.

Aşağıdakilerden birini belirtin:

#### HAYIR

Adlar zaman damgası içermez. Arşiv günlüğü veri kümeleri şu şekilde adlandırılır:

```
arcpxi.A nnnnnn
```

Burada *arcpxi*, ARCPFX1 ya da ARCPFX2 tarafından belirtilen veri kümesi adı önekidir. *arcpxi* en çok 35 karakter içerebilir.

#### EVET

Adlar bir zaman damgası içerir. Arşiv günlüğü veri kümeleri şu şekilde adlandırılır:

```
arcpxi.cyyddd.T hhmsst.A nnnnnn
```

Burada *c*, 1999 yılına kadar olan yıllar için 'D' ya da 2000 yılı ve sonrası için 'E' ve *arcpxi*, ARCPFX1 ya da ARCPFX2 tarafından belirtilen veri kümesi adı önekidir. *arcpxi* en çok 19 karakter içerebilir.

#### Uzt

Adlar bir zaman damgası içerir. Arşiv günlüğü veri kümeleri şu şekilde adlandırılır:

```
arcpxi.D yyyddd.T hhmsst.A nnnnnn
```

Burada *arcpxi*, ARCPFX1 ya da ARCPFX2 tarafından belirtilen veri kümesi adı önekidir. *arcpxi* en çok 17 karakter içerebilir.

Varsayılan değer NO'dur.

### birim

Arşiv günlüğü veri kümesinin ilk kopyasını saklamak için kullanılan aygıtın aygıt tipini ya da birim adını belirtir.

1-8 alfasayısal karakterden oluşan bir aygıt tipi ya da birim adı belirleyin. İlk karakter olarak bir harf kullanılmalıdır.



Bu parametre boş olamaz.

Varsayılan değer TAPE 'dir.

DASD ' de arşivleme yaparsanız, sınırlı birim aralığına sahip bir soysal aygıt tipi belirtebilirsiniz; örneğin, UNIT=3390.

DASD ' de arşivleme yapıyorsanız, aşağıdakilerden emin olun:

- Birincil alan ayırma, etkin günlük veri kümelerindeki tüm verileri içerecek kadar büyük.
- Arşiv günlüğü veri kümesi kataloğu seçeneği (CATALOG) YES olarak ayarlandı.
- BLKSIZE için uygun bir değer kullandınız.

TAPE ' de arşivleme yaparsanız, IBM MQ en çok 20 birim genişletebilir.

Önerilen ayarlar:

<b>Test sistemi</b>	DASD
<b>Üretim sistemi</b>	MANYETİK BANT

Arşiv günlükleri için bir konum seçmeyle ilgili daha fazla bilgi için [Günlük arşivi depolamanızı planlamabaşlıklı konuya](#) bakın.

## UNIT2

Arşiv günlüğü veri kümelerinin ikinci kopyasını saklamak için kullanılan aygıtın aygıt tipini ya da birim adını belirtir.

1-8 alfasayısal karakterden oluşan bir aygıt tipi ya da birim adı belirleyin. İlk karakter olarak bir harf kullanılmalıdır. Bu parametre boşsa, UNIT parametresi için ayarlanan değer kullanılır.

Varsayılan değer boşluktur.

## İlgili başvurular

“CSQ6SYSP ' nin kullanılması” sayfa 905

CSQ6SYSPkullanılarak sistem parametrelerinin nasıl ayarlanacağına ilişkin başvuru olarak bu konuyu kullanın.

“CSQ6LOGP ' nin kullanılması” sayfa 914

CSQ6LOGPkullanılarak günlüğe kaydetme seçeneklerinin nasıl belirtileceğini öğrenmek için bu konuyu kullanın.

 CSQ6USGP ' nin kullanılması

CSQ6USGP komutunu kullanarak sistem deęiřtirgelerinizin nasıl ayarlanacağına ilişkin başvuru olarak bu konuyu kullanın.

Ürün kullanımı kaydını denetlemek için CSQ6USGP komutunu kullanın.

CSQ6USGP için varsayılan parametreler [Çizelge 62 sayfa 925](#) içinde gösterilmektedir. Bu deęerlerden herhangi birini deęiřtirmeniz gerekirse, parametrelerin ayrıntılı açıklamalarına bakın.



**Uyarı:** SET SYSTEM komutunu kullanarak bu parametrelerin hiçbirini deęiřtiremezsiniz.

Çizelge 62. CSQ6USGP parametrelerinin varsayılan deęerleri		
Deęiřtirge	Açıklama	Varsayılan deęer
<a href="#">QMGRPROD</a>	Kuyruk yöneticisi kullanımının kaydedileceęi ürün	Boş
<a href="#">AMSPROD</a>	Advanced Message Security (AMS) kullanımının kaydedileceęi ürün	Boş

## QMGRPROD

Kuyruk yöneticisi kullanımının kaydedileceęi ürünü belirtir.

Aşağıdakilerden birini belirtin:

#### **MQ**

Kuyruk yöneticisi kullanımı, 5655-MQ9ürün tanıtıcısıyla bağımsız bir IBM MQ for z/OS ürünü olarak kaydedilir.

#### **VUE**

Kuyruk yöneticisi kullanımı, 5655-VU9ürün tanıtıcısıyla bağımsız bir IBM MQ for z/OS Value Unit Edition (VUE) ürünü olarak kaydedilir.

#### **GELİŞEN**

Kuyruk yöneticisi kullanımı, ürün tanıtıcısı 5655-AV1olan bir IBM MQ Advanced for z/OS Value Unit Edition ürününün bir parçası olarak kaydedilir.

#### **AMSPROD**

Bu parametre ayarlanmazsa, AMS adres alanı başlatılamaz ve [CSQY024I](#) iletisi çıkar.

Kullanılırsa, Advanced Message Security kullanımının kaydedileceği ürünü belirtir.

Aşağıdakilerden birini belirtin:

#### **AMS**

AMS kullanımı, 5655-AM9ürün tanıtıcısıyla bağımsız bir Advanced Message Security for z/OS ürünü olarak kaydedilir.

#### **GELİŞMİŞ**

AMS kullanımı, 5655-AV9ürün tanıtıcısıyla IBM MQ Advanced for z/OS ürününün bir parçası olarak kaydedilir.

#### **GELİŞEN**

AMS kullanımı, 5655-AV1ürün tanıtıcısıyla IBM MQ Advanced for z/OS Value Unit Edition ürününün bir parçası olarak kaydedilir.

Ürün kullanımı kaydı hakkında daha fazla bilgi için bkz. [Ürün bilgilerinin raporlanması](#) .

#### **İlgili başvurular**

“CSQ6SYSP ' nin kullanılması” sayfa 905

CSQ6SYSPkullanılarak sistem parametrelerinin nasıl ayarlanacağına ilişkin başvuru olarak bu konuyu kullanın.

“CSQ6LOGP ' nin kullanılması” sayfa 914

CSQ6LOGPkullanılarak günlüğe kaydetme seçeneklerinin nasıl belirtileceğini öğrenmek için bu konuyu kullanın.

#### **z/OS Kanal başlatıcı değiştirgelerini uyarla**

Kanal başlatıcıyı gereksinimlerinize uyacak şekilde uyarlamak için ALTER QMGR işlevini kullanın.

- *Bu görevi her IBM MQ kuyruk yöneticisi için gerektiği şekilde yineleyin.*
- *Bu görevi, önceki bir sürümden geçiş yaparken gerçekleştirmelisiniz.*

Bir dizi kuyruk yöneticisi özniteliği, dağıtılmış kuyruğun nasıl çalışacağını denetler. ALTER QMGR MQSC komutunu kullanarak bu öznitelikleri ayarlayın. Kullanıma hazırlama veri kümesi örneği thlqual.SCSQPROC(CSQ4INYG), uyarlayabileceğiniz bazı ayarları içerir. Ek bilgi için bkz. [ALTER QMGR](#).

Kanal başlatıcısını her başlattığınızda, bu parametrelerin değerleri bir ileti dizisi olarak görüntülenir.

#### **Bağdaştırıcılar, dağıtıcılar ve kanal sayısı üst sınırı arasındaki ilişki**

ALTER QMGR parametreleri CHIADAPS ve CHIDISPS, kanal başlatıcı tarafından kullanılan görev denetim öbeklerinin (TCB) sayısını tanımlar. CHIADAPS (bağdaştırıcı) TCB ' leri kuyruk yöneticisine IBM MQ API çağrıları yapmak için kullanılır. CHIDISPS (dağıtıcı) TCB ' leri iletişim ağına çağrı yapmak için kullanılır.

ALTER QMGR parametresi MAXCHL, dağıtıcı TCB ' ler üzerindeki kanalların dağıtımını etkiler.

#### **ŞİDİSLERİ**

Az sayıda kanalınız varsa, varsayılan değeri kullanın.

Her işlemci için bir görev, sistem başarımını eniyiler. Dağıtıcı görevleri CPU yoğun olduğundan, ilke, iş parçacıklarını bulmak ve başlatmak için geçen süre en aza indirgenecek şekilde mümkün olduğunca az sayıda görevi meşgul tutmaktır.

CHIDISPS (20), 100 'den fazla kanalı olan sistemler için uygundur. Bunun gerekenden daha fazla dağıtıcı TCB olduğu durumlarda CHIDISPS (20) ' ye sahip olmanın önemli bir dezavantajı yoktur.

Bir kılavuz olarak, 1000 'den fazla kanalınız varsa, her 50 mevcut kanal için bir dağıtıcıya izin verin. Örneğin, 2000 'e kadar etkin kanalı işlemek için CHIDISPS (40) belirleyin.

TCP/IP kullanıyorsanız, CHIDISPS ' de daha büyük bir değer belirleseniz bile, TCP/IP kanalları için kullanılan dağıtıcı sayısı üst sınırı 100 'dür.

## CHIADAPS

Kuyruk yöneticisine yönelik her IBM MQ API çağrısı diğerlerinden bağımsızdır ve herhangi bir bağıdaştırıcı TCB 'si üzerinde yapılabilir. Kalıcı ileti kullanan çağrılar, günlük G/Ç nedeniyle kalıcı olmayan iletiler için olandan çok daha uzun sürebilir. Bu nedenle, birçok kanalda çok sayıda kalıcı ileti işleyen bir kanal başlatıcısı, en iyi performans için varsayılan 8 bağıdaştırıcı TCB ' den daha fazlasına gereksinim duyabilir. Toplu işleme sonu da günlük G/Ç gerektirdiğinden ve ince istemci kanallarının kullanıldığı durumlarda, bu özellikle ulaşılan toplu iş boyutu küçüktür.

Üretim ortamı için önerilen değer: CHIADAPS (30). Bundan daha fazlasının kullanılması, önemli bir ekstra avantaj sağlamaz ve bu, gerekenden daha fazla bağıdaştırıcı TCB 'si ise, CHIADAPS (30)' a sahip olmanın önemli bir dezavantajı yoktur.

## MAXCHL

Her kanal, kanal başlangıcında belirli bir dağıtıcı TCB ile ilişkilendirilir ve kanal duruncaya kadar bu TCB ile ilişkili kalır. Birçok kanal her bir TCB ' yi paylaşabilir. MAXCHL, kanalları kullanılabilir dağıtıcı TCB ' lerine dağıtmak için kullanılır. Birinci (MIN (MAXCHL/CHIDISPS) , 10)) başlangıç kanalları ilk dağıtıcı TCB ile ilişkilendirilir ve tüm dağıtıcı TCB ' leri kullanılıncaya kadar bu şekilde devam eder.

Bunun az sayıda kanal ve büyük bir MAXCHL için etkisi, kanalların dağıtıcılar arasında eşit olarak dağıtılmamasıdır. Örneğin, CHIDISPS 'yi (10) ayarlarsanız ve MAXCHL' yi varsayılan değeri 200 olarak bırakırsanız, ancak yalnızca 50 kanalı varsa, beş dağıtıcı her biri 10 kanal ile ilişkilendirilir ve beş kanal kullanılmaz. Bunun küçük bir sabit sayı olduğu durumlarda, MAXCHL ' nin gerçekte kullanılacak kanal sayısına ayarlanmasını öneririz.

Bu kuyruk yöneticisi özelliğini değiştirirseniz, değerlerin uyumlu olduğundan emin olmak için ACTCHL, LU62CHLve TCPCHL kuyruk yöneticisi özelliklerini de gözden geçirmeniz gerekir. Bu özelliklerin ve ilişkilerinin tam açıklaması için [Kuyruk yöneticisi parametreleri](#) konusuna bakın.

## Kanal başlatıcıları için z/OS UNIX System Services ortamınızı ayarlama

Kanal başlatıcı (CHINIT) OMVS iş parçacıklarını kullanır. Yeni bir CHINIT oluşturmadan önce ya da dağıtıcı ya da SSLTASKS sayısını değiştirmeden önce OMVS yapılandırma parametrelerini gözden geçirin.

Her bir CHINIT, 3 + CHIDISP + SSLTASKS OMVS iş parçacıklarını kullanır. Bunlar, LPAR ' da kullanılan OMVS iş parçacıklarının toplam sayısına ve CHINIT tarafından başlatılan görev kullanıcı kimliği tarafından kullanılan iş parçacıklarının sayısına katkıda bulunur.

**D OMVS, L** ürününü kullanabilir ve MAXPROCSYS ' nin geçerli kullanım, yüksek su kullanımı ve sistem sınırını (sistemin izin verdiği işlem sayısı üst sınırı) gözden geçirebilirsiniz.

Yeni bir CHINIT ekliyorsanız ya da CHIDISPS ya da SSLTASKS değerlerini artırılıyorsanız, iş parçacıklarındaki artışı hesaplamalı ve MAXPROCSYS değerleri üzerindeki etkiyi gözden geçirmelisiniz. MAXPROCSYS ' yi devingen olarak değiştirmek ya da BPXPRCxx parmlib değerini ya da her ikisini birden güncellemek için **SETOMVS** komutunu kullanabilirsiniz.

OMVS parametresi MAXPROCUSER, aynı UID ' ye sahip olan tek bir OMVS kullanıcısının sahip olduğu OMVS iş parçacıklarının sayısıdır. İş parçacıkları bu değere doğru sayılır. Bu nedenle, her biri 10 dağıtıcı ve 3 SSLTASKS olan, aynı başlatılan görev kullanıcı kimliğine sahip 2 CHINITs varsa, OMVS uid için  $2 * (3 + 10 + 3) = 32$  iş parçacığı vardır.

**D OMVS, O** komutunu vererek varsayılan MAXPROCUSER 'i görüntüleyebilir ve MAXPROCUSER' i dinamik olarak değiştirmek için **SETOMVS** komutunu kullanabilir ya da BPXPRCxx parmlib değerini ya da her ikisini birden güncelleyebilirsiniz.

Bu değeri, RACF komut **ALTUSER userid OMVS(PROCUSERMAX(nnnn))** ya da eşdeğeri ile kullanıcı temelinde geçersiz kılabilirsiniz.

Kanal başlatıcıyı başlatmak için aşağıdaki komutu verin:

```
START CHINIT
```

Kanal başlatıcının başarıyla başlatıldığından emin olmak için, xxxxCHIN(ssidCHIN) iş günlüğünde ICH408I hatası olup olmadığını denetleyin.

### İlgili kavramlar

[“Toplu İş, TSO ve RRS bağdaştırıcılarını ayarla” sayfa 928](#)

Uygun STEPLIB birleşimlerine kitaplık ekleyerek bağdaştırıcıları uygulamalar için kullanılabilir kılın. Bir bağdaştırıcı tarafından yayınlanan SNAP dökümlerine hitap etmek için bir CSQSNAP DDname ayırın. Uygulama programlarınızın taşınabilirliğini artırmak için CSQBDEFV kullanmayı düşünün

### İlgili başvurular

[Kanal başlatıcı istatistikleri veri kayıtları](#)

### **Toplu İş, TSO ve RRS bağdaştırıcılarını ayarla**

Uygun STEPLIB birleşimlerine kitaplık ekleyerek bağdaştırıcıları uygulamalar için kullanılabilir kılın. Bir bağdaştırıcı tarafından yayınlanan SNAP dökümlerine hitap etmek için bir CSQSNAP DDname ayırın. Uygulama programlarınızın taşınabilirliğini artırmak için CSQBDEFV kullanmayı düşünün

- *Bu görevi her IBM MQ kuyruk yöneticisi için gerektiği şekilde yineleyin.*
- *Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekebilir.*

Toplu iş bağlantılarını kullanarak bağdaştırıcıları toplu iş ve diğer uygulamalar için kullanılabilir kılmak üzere toplu iş uygulamanıza ilişkin STEPLIB birleştirmesine aşağıdaki IBM MQ kitaplıklarını ekleyin:

- thlqual.SCSQANL x
- thlqual.SCSQAUTH

Burada x , ulusal dilinize ilişkin dil harfidir. (Kitaplıklar LPA ya da bağlantı listesindeyse bunu yapmanıza gerek yoktur.)

TSO uygulamaları için, TSO oturum açma yordamındaki STEPLIB bitişirmesine kitaplıkları ekleyin ya da TSO komutunu kullanarak bunları etkinleştirin.

Bağdaştırıcı beklenmeyen bir IBM MQ hatası saptarsa, DDname CSQSNAP ' ye bir z/OS SNAP dökümü verir ve uygulamaya neden kodu MQR\_Unexpected\_Error verir. CSQSNAP DD deyimini uygulamada yoksa JCL ya da CSQSNAP TSO altındaki bir veri kümesine ayrılmazsa döküm alınmaz. Bu durumda, CSQSNAP DD deyimini JCL uygulamasına ekleyebilir ya da CSQSNAP ' yi TSO altındaki bir veri kümesine ayırıp uygulamayı yeniden çalıştırabilirsiniz. Ancak, bazı sorunlar kesintili olduğu için, uygulamada JCL 'ye bir CSQSNAP deyimini eklemeniz ya da CSQSNAP' yi TSO oturum açma yordamındaki bir veri kümesine ayırmanız önerilir.

Sağlanan program CSQBDEFV, uygulama programlarınızın taşınabilirliğini artırır. CSQBDEFV ' de, bir uygulama programındaki MQCONN ya da MQCONNX çağrısında belirtmek yerine, bağlanılacak kuyruk yöneticisinin ya da kuyruk paylaşım grubunun adını belirtebilirsiniz. Her kuyruk yöneticisi ya da kuyruk paylaşım grubu için yeni bir CSQBDEFV sürümü yaratabilirsiniz. Bunu yapmak için aşağıdaki adımları izleyin:

1. IBM MQ çevirici programı CSQBDEFV ' yi thlqual.SCSQASMS içinden bir kullanıcı kitaplığına kopyalayın.
2. Sağlanan program, CSQ1varsayılan altsistem adını içeriyor. Bu adı sınama ve kuruluş doğrulaması için saklayabilirsiniz. Üretim altsistemlerinde NAME=CSQ1 değerini, bir-dört karakterlik altsistem adınızla değiştirebilir ya da CSQ1kullanabilirsiniz.

Kuyruk paylaşım grupları kullanıyorsanız, CSQ1yerine bir kuyruk paylaşım grubu adı belirtebilirsiniz. Bunu yaparsanız, program o grup içindeki etkin bir kuyruk yöneticisine bağlanma isteği verir.

3. CSQBDEFV yükleme modülünü üretmek için programı derleyip düzenleyin. Yapıbirimi için, SYSLIB bağlamanızda thlqual.SCSQMACS kitaplığını ekleyin; RENT , AMODE=31 , RMODE=ANYbağ düzenleme değiştirgelerini kullanın. Bu, thlqual.SCSQPROC(CSQ4DEFV) içindeki örnek JCL ' de gösterilir. Daha sonra, yükleme kitaplığını thlqual.SCSQAUTHdeğerinin önünde z/OS Toplu İş ya da TSO STEPLIB içine ekleyin.

### İlgili kavramlar

“İşlemleri ve denetim panolarını ayarlama” sayfa 929

İşlemleri ve denetim panolarını ayarlamak için öncelikle gerekli panoları, EXEC ' leri, iletileri ve çizelgeleri içeren kitaplıkları ayarlamanız gerekir. Bunu yapmak için, paneller için hangi ulusal dil özelliğinin kullanılacağını göz önünde bulundurmanız gerekir. Bunu yaptığınızda, isteğe bağlı olarak IBM MQ işlemleri ve denetim panoları için ana ISPF menüsünü güncelleyebilir ve işlev tuşu ayarlarını değiştirebilirsiniz.

### **z/OS İşlemleri ve denetim panolarını ayarlama**

İşlemleri ve denetim panolarını ayarlamak için öncelikle gerekli panoları, EXEC ' leri, iletileri ve çizelgeleri içeren kitaplıkları ayarlamanız gerekir. Bunu yapmak için, paneller için hangi ulusal dil özelliğinin kullanılacağını göz önünde bulundurmanız gerekir. Bunu yaptığınızda, isteğe bağlı olarak IBM MQ işlemleri ve denetim panoları için ana ISPF menüsünü güncelleyebilir ve işlev tuşu ayarlarını değiştirebilirsiniz.

- *Bu görevi, çalıştırmak istediğiniz her z/OS sistemi için bir kez gerçekleştirmeniz gerekir IBM MQ.*
- *Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekebilir.*

### Kitaplıkların ayarlanması

IBM MQ işlemlerini ve denetim panolarını ayarlamak için aşağıdaki adımları izleyin:

1. Birleştirmelerinizdeki tüm kitaplıkların aynı biçimde (F, FB, V, VB) olduğundan ve aynı blok boyutuna sahip olduğundan ya da blok boyutlarını azalttığından emin olun. Ters durumda, bu panoları kullanmaya çalışırken sorunlarla karşılaşabilirsiniz.
2. thlqual.SCSQEXEC kitaplığını SYSEXEC ya da SYSPROC birleştirmenize ekleyin ya da TSO ALTLIB komutunu kullanarak kitaplığı etkinleştirin. Kuruluş sırasında sabit blok 80 kayıt biçimiyle ayrılan bu kitaplık, gerekli EXEC ' leri içerir.

Kitaplığı SYSEXEC birleştirmeniz için tercih edilir. Ancak, SYSPROC içine koymak istiyorsanız, kitaplığın kayıt uzunluğu 80 byte olmalıdır.

3. thlqual.SCSQAUTH ve thlqual.SCSQANLx öğelerini STEPLIB TSO oturum açma yordamına ekleyin ya da TSO TSOLIB komutunu kullanarak etkinleştirin (bağlantı listesinde ya da Yaşam Çevrimi Projesi Yönetimi 'nde (LPA) yoksa).
4. IBM MQ panosu kitaplıklarını ISPF kitaplık ayarınıza kalıcı olarak ekleyebilir ya da panolar kullanıldığında bunların dinamik olarak ayarlanmasına izin verebilirsiniz. Önceki seçim için aşağıdakileri yapmanız gerekir:
  - a. ISPLIB birleştirmenizin işlemleri ve denetim panosu tanımlamalarını içeren kitaplığı içermesini sağlar. Ad thlqual.SCSQPNLx, burada x, ulusal dilinizin dil harfidir.
  - b. Gerekli çizelgeleri içeren kitaplığı ISPTLIB birleştirmeniz için ekleyin. Ad thlqual.SCSQTBLx' dir; burada x, ulusal dilinizin dil harfidir.
  - c. Gerekli iletileri içeren kitaplığı ISPLIB birleştirmeniz için ekleyin. Ad: thlqual.SCSQMSGx; burada x, ulusal dilinize ilişkin dil harfidir.
  - d. Gerekli yükleme modüllerini içeren kitaplığı ISPLIB birleştirmeniz içine ekleyin. Bu kitaplığın adı thlqual.SCSQAUTH.

İkinci seçenek için z/OS LIBDEF komutunu kullanın. Kullanabileceğiniz çeşitli anahtar sözcüklere ilişkin bir bağlantı için [Örnekler](#) konusuna bakın.

5. IBM MQ panolarına TSO Komut İşlemcisi panosundan erişebileceğinizi deneyin. Bu genellikle ISPF/PDF Birincil Seçenekler Menüsü 'nde seçenek 6 'dir. Çalıştırdığınız EXEC ' nin adı CSQOREXX. IBM MQ

kitaplıklarını, adım 4' te olduğu gibi, ISPF kuruluşunuza kalıcı olarak yerleştirip yerleştirmedeğinizi belirtecek parametre yoktur. Yapmadıysanız, aşağıdakileri kullanın:

```
CSQOREXX thlqual langletter
```

Burada langletter , kullanılacak ulusal dili tanımlayan bir harftir:

**C**

Basitleştirilmiş Çince

**E**

U.S. İngilizce (büyük ve küçük harf karışık)

**F**

Fransızca

**K**

Japonca

**U**

U.S. İngilizce (büyük harf)

## ISPF menüsünün güncellenmesi

ISPF ana menüsünü, ISPF' den IBM MQ işlemlerine ve denetim panolarına erişime izin verecek şekilde güncelleyebilirsiniz. & ZSEL için gerekli ayar:

```
CMD(%CSQOREXX thlqual langletter)
```

thlqual ve langletter hakkında bilgi için bkz. Adım "5" sayfa 929.

Daha fazla ayrıntı için bkz. [z/OS: ISPF İletişim Kutusu Geliştirici Kılavuzu ve Başvurusu](#).

## İşlev tuşlarının ve komut ayarlarının güncellenmesi

Paneller tarafından kullanılan işlev tuşlarını ve komut ayarlarını değiştirmek için olağan ISPF yordamlarını kullanabilirsiniz. Uygulama tanıtıcısı CSQO.

Ancak, yardım bilgileri yaptığınız değişiklikleri yansıtacak şekilde güncellenmediğinden, bu önerilmez.

### İlgili kavramlar

"IBM MQ döküm biçimlendirme üyesini dahil et" sayfa 930

IPCS (Interactive Problem Control System; Etkileşimli Sorun Denetim Sistemi) kullanarak IBM MQ dökümlerini biçimlendirebilmek için bazı sistem kitaplıklarını güncellemelisiniz.

### **z/OS IBM MQ döküm biçimlendirme üyesini dahil et**

IPCS (Interactive Problem Control System; Etkileşimli Sorun Denetim Sistemi) kullanarak IBM MQ dökümlerini biçimlendirebilmek için bazı sistem kitaplıklarını güncellemelisiniz.

- Bu görevi, çalıştırmak istediğiniz her z/OS sistemi için bir kez gerçekleştirmeniz gerekir IBM MQ.
- Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekir.

IBM MQ dökümlerini IPCS (Interactive Problem Control System; Etkileşimli Sorun Denetim Sistemi) kullanarak biçimlendirebilmek için, thlqual.SCSQPROC(CSQ7IPCS) veri kümesini SYS1.PARMLIB. Bu veri kümesini düzenlemenize gerek yoktur.

TSO yordamını IPCS için uyardıysanız, thlqual.SCSQPROC(CSQ7IPCS) IPCSPARM tanımındaki herhangi bir kitaplığa kopyalanabilir. Daha fazla bilgi için bkz. [z/OS MVS IPCS Kullanıcı Kılavuzu](#) .

ISPLLIB bağlamanızda thlqual.SCSQPPLA kitaplığını da eklemeniz gerekir.

Döküm biçimlendirme programlarını TSO oturumunuz ya da IPCS işinizin kullanımına sunmak için, thlqual.SCSQAUTH kitaplığını da STEPLIB birleşiminize eklemeniz ya da TSO TSOLIB komutunu kullanarak etkinleştirmeniz gerekir (bağlantı listesinde ya da LPA ' da olsa bile).

### İlgili kavramlar

“Bilgi iletilerini engelle” sayfa 931

IBM MQ sisteminiz çok sayıda bilgi iletisi üretebilir. Seçilen iletilerin konsola ya da basılı günlüğe gönderilmesini önleyebilirsiniz.

### **z/OS** Bilgi iletilerini engelle

IBM MQ sisteminiz çok sayıda bilgi iletisi üretebilir. Seçilen iletilerin konsola ya da basılı günlüğe gönderilmesini önleyebilirsiniz.

- *Bu görevi, çalıştırmak istediğiniz her z/OS sistemi için bir kez gerçekleştirmeniz gerekir IBM MQ.*
- *Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekmez.*

IBM MQ sisteminiz yoğun bir şekilde kullanılıyorsa, birçok kanal durduruluyor ve başlatılıyorsa, z/OS konsoluna ve basılı günlüğe çok sayıda bilgi iletisi gönderilir. IBM MQ - IMS köprüsü ve arabellek yöneticisi çok sayıda bilgi iletisi de üretebilir.

Gerekirse, SYS1.PARMLIBMPFLSTxx üyeleri tarafından belirlenen z/OS ileti işleme olanağı listesini kullanarak bu konsol iletilerinin bazılarını engelleyebilirsiniz. Belirlediğiniz iletiler basılı günde görüntülenmeye devam ediyor, ancak konsolda görüntülenmiyor.

Örnek thlqua1.SCSQPROC(CSQ4MPFL), MPFLSTxx için önerilen ayarları gösterir. Ek bilgi için bkz. [MPFLSTxx \(ileti işleme olanağı listesi\)](#) .

Basılı kopya günlüğünde seçilen bilgi iletilerini engellemek istiyorsanız, z/OS kuruluş çıkışı (IEAVMXIT) kullanabilirsiniz. Gerekli iletiler için aşağıdaki bit anahtarlarını AÇIK olarak ayarlayabilirsiniz:

#### **CTXTRDTM**

İletiyi silin.

İleti konsollarda ya da oturum açmış basılı kopyalarda görüntülenmez.

#### **CTXTESJL**

İş günlüğünden gizleme.

İleti JES iş günlüğüne girmiyor.

#### **CTXTNWTP**

WTP işlemlerini gerçekleştirmeyin.

İleti, bir TSO uçbirimine ya da toplu işin sistem ileti verileri kümesine gönderilmez.

#### **Not:**

1. Diğer deęiřtirgelerle ilgili tüm ayrıntılar için [MVS Installation Exits](#)(MVS Kuruluş Çıkışları) konusuna bakın.
2. Önerilen CSQ4MPFLengelleme listesindeki iletilerden başka iletileri gizlemeniz önerilmez.

Ek olarak, ek parametreyi de belirtebilirsiniz:

#### **EXCLMSG**

Herhangi bir günden dışlanacak iletilerin listesini belirtir.

Bu listedeki iletiler z/OS konsoluna ve basılı günlüğe gönderilmez. Daha fazla bilgi için “[CSQ6SYSP ' nin kullanılması](#)” sayfa 905 içindeki [EXCLMSG](#) bölümüne bakın.

### İlgili görevler

“z/OS üzerinde bir kuyruk yöneticisinin sınanması” sayfa 946

Kuyruk yöneticinizi özelleřtirdiğinizde ya da yeni düzeye geçirdiğinizde, kuruluş doğrulama programlarını ve IBM MQ for z/OS ile birlikte gönderilen bazı örnek uygulamaları çalıştırarak bu programı sınavabilirsiniz.



## Kuyruk paylaşım grubunun yapılandırılması

Yüksek kullanılabilirlik için paylaşılan kuyrukları kullanmak istiyorsanız, kuyruk paylaşım grubunu yapılandırmak için adım adım bu konuları kullanın.

IBM MQ for z/OS sisteminizi ayarlamaya ilişkin işlemin bu bölümündeki adımları tamamladığınızda, kuyruk paylaşım grubu verilerini eklemek için “Sistem parametre modülünüzü uyarlama, da,-” sayfa 903 gerçekleştirmeniz gerekir. [CSQ6SYSP](#) ' yi QSGDATA parametresini belirtecek şekilde değiştirmeniz gerekir.

## Db2 ortamını ayarlama

Kuyruk paylaşım gruplarını kullanıyorsanız, bir dizi örnek işi özelleştirerek ve çalıştırarak gerekli Db2 nesnelerini yaratmanız gerekir.

### Db2 ortamını ayarlama

Bir dizi örnek işi özelleştirerek ve çalıştırarak gerekli Db2 nesnelerini yaratmanız ve bağlamanız gerekir.

- Her Db2 veri paylaşım grubu için bu görevi yineleyin.
- Önceki bir sürümden geçiş yaparken bind ve grant adımlarını gerçekleştirmeniz gerekir.
- Kuyruk paylaşım gruplarını kullanmıyorsanız bu görevi atlayın.

Daha sonra kuyruk paylaşım gruplarını kullanmak istiyorsanız, o zaman bu görevi gerçekleştirin.

IBM MQ , iki eşdeğer iş kümesi sağlar. CSQ45 önekinde sahip olanlar, önceki IBM MQ sürümleriyle uyumluluk ve IBM MQ sürüm 11 ve önceki sürümlerle kullanım içindir. Db2 V12 ya da sonraki bir yayın düzeyiyle yeni bir veri paylaşım grubu kuruyorsanız, bu işler dinamik boyutlandırma ve UTS (Universal Table Spaces; Evrensel Çizelge Alanları) için daha yeni Db2 yeteneklerinden yararlandığı için, işleri CSQ4X önekiyle kullanmanız önerilir.

Her yeni Db2 veri paylaşım grubu için aşağıdaki adımlar gerçekleştirilmelidir. Tüm örnek JCL, thlqual.SCSQPROÇinde bulunur.

1. IBM MQ veritabanı, çizelge alanları ve çizelgeleri için kullanılacak saklama yeri grubunu yaratmak için örnek JCL CSQ4XCSG ' yi uyarlayın ve yürütün.
2. Bu Db2 veri paylaşım grubuna bağlanan tüm kuyruk yöneticileri tarafından kullanılacak veritabanını yaratmak için örnek JCL CSQ4XCDB ' yi uyarlayın ve yürütün.
3. Kuyruk paylaşım grupları için kullanılan kuyruk yöneticisi ve kanal başlatıcı çizelgelerini içeren çizelge alanlarını yaratmak için örnek JCL CSQ4XCTS ' yi uyarlayın ve yürütün.
4. 15 Db2 çizelgelerini ve ilişkili izinleri yaratmak için örnek JCL CSQ4XCTB ' yi uyarlayın ve yürütün. Satır adlarından ya da özniteliklerinden hiçbirini değiştirmeyin.
5. Kuyruk yöneticisi, yardımcı programlar ve kanal başlatıcısı için Db2 planlarını bağlamak üzere örnek JCL CSQ45BPL ' yi özelleştirin ve yürütün.
6. Kuyruk yöneticisi, yardımcı programlar ve kanal başlatıcısı tarafından kullanılan kullanıcı kimlikleri için yürütme yetkisi vermek üzere örnek JCL CSQ45GEX ' i uyarlayın ve yürütün. Kuyruk yöneticisi ve kanal başlatıcısına ilişkin kullanıcı kimlikleri, başlatılan görev yordamlarının çalıştırıldığı kullanıcı kimlikleridir. Yardımcı programlara ilişkin kullanıcı kimlikleri, toplu işlerin sunulabileceği kullanıcı kimlikleridir.

Uygun planların adları aşağıdaki tabloda gösterilir.



Kullanıcı	Planlar ( <b>LTS</b> )	Planlar ( <b>CD</b> )
Kuyruk yöneticisi	CSQ5A 930, CSQ5C 930, CSQ5D 930, CSQ5K 930, CSQ5L 930, CSQ5M 930, CSQ5P 930, CSQ5R 930, CSQ5S 930, CSQ5T 930, CSQ5U 930, CSQ5W 930	CSQ5A 9X0, CSQ5C 9X0, CSQ5D 9X0, CSQ5K 9X0, CSQ5L 9X0, CSQ5M 9X0, CSQ5P 9X0, CSQ5R 9X0, CSQ5S 9X0, CSQ5T 9X0, CSQ5U 9X0, CSQ5W 9X0
CSQUTIL toplu iş yardımcı programının SDEFS işlevi	CSQ52 930	CSQ52 9X0
CSQ5PQSG ve CSQJUCNV toplu iş yardımcı programları	CSQ5B 930	CSQ5B 9X0
CSQUZAP hizmet yardımcı programı	CSQ5Z 930	CSQ5Z 9X0

Db2 kuruluşu sırasında bir hata oluşması durumunda, aşağıdaki işler özelleştirilebilir ve yürütülebilir:

- Çizelgeleri ve dizinleri atmak için CSQ45DTB .
- Çizelge alanlarını atmak için CSQ4XDTS .
- Veritabanını atmak için CSQ4XDDB .
- Depolama grubunu atmak için CSQ4XDSG .

**Not:** Bu işler bir Db2 kilitleme sorunu nedeniyle başarısız olursa, bunun nedeni büyük olasılıkla Db2 kaynağı için çekişme (özellikle de sistem yoğun olarak kullanılıyorsa) olabilir. İşleri daha sonra yeniden sunun. Sistem hafif kullanıldığında ya da susturulduğunda bu işlerin çalıştırılması tercih edilir.

Db2kurulumu hakkında daha fazla bilgi için *Db2 for z/OS 12.0.0* içinde Db2 Yönetimi başlıklı konuya bakın.

Db2 tablo boyutlarıyla ilgili bilgi için bkz. z/OS üzerinde planlama .

### İlgili kavramlar

“Bağlaşım olanağını ayarla” sayfa 933

Kuyruk paylaşım grupları kullanıyorsanız, IXCMIAPU kullanarak, bağlaşım olanağı Kaynak Yönetimi (CFRM) ilke veri kümesindeki kuyruk paylaşım grubunda (QSG) kuyruk yöneticileri tarafından kullanılan bağlaşım olanağı yapılarını tanımlayın.

### **z/OS** **Bağlaşım olanağını ayarla**

Kuyruk paylaşım grupları kullanıyorsanız, IXCMIAPU kullanarak, bağlaşım olanağı Kaynak Yönetimi (CFRM) ilke veri kümesindeki kuyruk paylaşım grubunda (QSG) kuyruk yöneticileri tarafından kullanılan bağlaşım olanağı yapılarını tanımlayın.

IXCMIAPU ile ilgili ek bilgi için Administrative data utility başlıklı konuya bakın.

- Bu görevi her kuyruk paylaşım grubu için yineleyin.
- Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekebilir.
- Kuyruk paylaşım gruplarını kullanmıyorsanız bu görevi atlayın.

Daha sonra kuyruk paylaşım gruplarını kullanmak istiyorsanız, o zaman bu görevi gerçekleştirin.

Kuyruk paylaşım grubuna ilişkin tüm yapılar, kuyruk paylaşım grubunun adıyla başlar. Aşağıdaki yapıları tanımlayın:

- *qsg-name* CSQ\_ADMIN adlı bir yönetim yapısı. Bu yapı IBM MQ ' in kendisi tarafından kullanılır ve herhangi bir kullanıcı verisi içermez.
- *qsg-name* CSQSYSAPPL adlı bir sistem uygulama yapısı. Bu yapı, IBM MQ sistem kuyrukları tarafından durum bilgilerini saklamak için kullanılır.
- Paylaşılan kuyruklara ilişkin iletileri tutmak için kullanılan bir ya da daha çok yapı. Bu adların en çok 16 karakter uzunluğunda olması gerekir.
  - İlk dört karakter, kuyruk paylaşım grubu adı olmalıdır. (Kuyruk paylaşım grubu adı dört karakterden azsa, @ simgeleriyle dört karaktere doldurulacaktır.)
  - Beşinci karakter alfabetik ve sonraki karakterler alfabetik ya da sayısal olmalıdır. Adın bu bölümü (kuyruk paylaşım grubu adı olmadan), paylaşılan bir kuyruk ya da CF yapısı nesnesi tanımladığınızda CFSTRUCT adı için belirlediğiniz addır.

Paylaşılan kuyruklara ilişkin iletileri tutmak için kullanılan yapıların adlarında yalnızca alfabetik ve sayısal karakterler kullanabilirsiniz, başka karakterler kullanamazsınız (örneğin, yönetim yapısının adında kullanılan \_ karakteri).

IXCMIAPU ' ya ilişkin örnek denetim deyimleri thlqual.SCSQPROC(CSQ4CFRM) veri kümesinde bulunur. Bunları özelleştirin ve bağlaşım olanağı için IXCMIAPU işinize ekleyin ve çalıştırın.

Yapılarınızı başarıyla tanımladığınızda, kullanılmakta olan CFRM ilkesini etkinleştirin. Bunu yapmak için aşağıdaki z/OS komutunu verin:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME= policy-name
```

CF yapılarının ve boyutlarının planlanmasına ilişkin bilgi için [Eşleme olanağı kaynaklarının tanımlanması](#) başlıklı konuya bakın.

### İlgili kavramlar

“ESM güvenlik denetimlerinizi uygulayın” sayfa 895

Kuyruk yöneticileri ve kanal başlatıcısı için güvenlik denetimlerini uygulayın.

### SMDS ortamını ayarlama

Paylaşılan kuyruklardaki iletileri boşaltmak için SMDS kullanmak istiyorsanız, SMDS boşaltma depolama ortamını ayarlayın.

- SMDS ' ye veri boşaltmak üzere yapılandırmak istediğiniz kuyruk paylaşım grubundaki her kuyruk yöneticisi ve yapı için bu görevi gerçekleştirin.
- Daha sonra SMDS ' ye veri boşaltmak için ek yapılar yapılandırmak istiyorsanız, bu görev o zaman yeniden gerçekleştirilebilir.
- Kuyruk paylaşım gruplarını kullanmıyorsanız bu görevi kaldırın.

*Daha sonra kuyruk paylaşım gruplarını kullanmak istiyorsanız, bu görevi o zaman gerçekleştirin.*

### SMDS ortamını ayarlama

1. Yapı ve veri kümesi alanı gereksinimlerini tahmin edin. Bkz. [Paylaşılan ileti veri kümesi kapasitesiyle ilgili önemli noktalar](#).
2. Veri kümelerini ayır ve önceden biçimlendir. Bkz. [Paylaşılan ileti veri kümesi oluşturma](#).
3. CF yapısını IBM MQolarak tanımlarken, CFSTRUCT ' yi CFLEVEL (5) ve OFFLOAD (SMDS) ile tanımladığınızdan emin olun.

### İlgili kavramlar

“Bağlaşım olanağını ayarla” sayfa 933

Kuyruk paylaşım grupları kullanıyorsanız, IXCMIAPU kullanarak, bağlaşım olanağı Kaynak Yönetimi (CFRM) ilke veri kümesindeki kuyruk paylaşım grubunda (QSG) kuyruk yöneticileri tarafından kullanılan bağlaşım olanağı yapılarını tanımlayın.

## **z/OS IBM MQ girdilerini Db2 tablolarına ekleme**

Kuyruk paylaşım gruplarını kullanıyorsanız, Db2 veri paylaşım grubundaki IBM MQ çizelgelerine kuyruk paylaşım grubu ve kuyruk yöneticisi girişleri eklemek için CSQ5PQSG yardımcı programını çalıştırın.

- *Bu görevi her IBM MQ kuyruk paylaşım grubu ve her kuyruk yöneticisi için yineleyin.*
- *Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekebilir.*
- *Kuyruk paylaşım gruplarını kullanmıyorsanız bu görevi kaldırın.*

*Daha sonra kuyruk paylaşım gruplarını kullanmak istiyorsanız, o zaman bu görevi gerçekleştirin.*

Her kuyruk paylaşım grubu ve bir kuyruk paylaşım grubunun üyesi olacak her kuyruk yöneticisi için CSQ5PQSG komutunu çalıştırın.

Belirtilen sırada aşağıdaki işlemleri gerçekleştirin:

1. CSQ5PQSG programının ADD QSG işlevini kullanarak IBM MQ Db2 çizelgelerine bir kuyruk paylaşım grubu girişi ekleyin. thlqual.SCSQPROC(CSQ45AQS) içinde bir örnek sağlanır.

Bu işlevi, Db2 veri paylaşım grubunda tanımlanan her kuyruk paylaşım grubu için bir kez gerçekleştirin. Kuyruk paylaşım grubuna başvuran kuyruk yöneticisi girişleri eklenmeden önce kuyruk paylaşım grubu girişi var olmalıdır.

2. CSQ5PQSG programının ADD QMGR işlevini kullanarak IBM MQ Db2 çizelgelerine bir kuyruk yöneticisi girişi ekleyin. thlqual.SCSQPROC(CSQ45AQM) içinde bir örnek verilmiştir.

Bu işlevi, kuyruk paylaşım grubunun üyesi olacak her kuyruk yöneticisi için gerçekleştirin.

### **Not:**

- a. Kuyruk yöneticisi yalnızca bir kuyruk paylaşım grubunun üyesi olabilir.
- b. Kuyruk paylaşım gruplarını kullanabilmek için çalışan RRS ' nin olması gerekir.

### **İlgili kavramlar**

“Sistem parametre modülünüzü uyarlama, da,-” sayfa 903

IBM MQ sistem parametresi modülü, IBM MQ ' un işletiminde kullandığı günlüğe kaydetme, arşivleme, izleme ve bağlantı ortamlarını denetler. Varsayılan bir modül sağlanır. Kendi sistem parametre modülünüzü bazı parametreler (örneğin, veri kümesi adları) olarak yaratmanız gerekir; bunlar genellikle siteye özgüdür.

## **z/OS Kuyruk paylaşım grubu için ESM güvenlik denetimlerini uygula**

Db2 ' e ve bağlaşımları listesi yapılarına erişmek için bir kuyruk paylaşım grubundaki tüm kuyruk yöneticileri için güvenlik denetimleri uygulayın.

- *Bir kuyruk paylaşım grubundaki her IBM MQ kuyruk yöneticisi için bu görevi yineleyin.*
- *Önceki bir sürümden geçiş yaparken bu görevi gerçekleştirmeniz gerekebilir.*

Kuyruk yöneticisi, kanal başlatıcısı ve yardımcı programlarla ilişkili kullanıcı kimliklerinin, bağlantı kurmak istediğiniz her Db2 altsistemiyle bir RRSF bağlantısı kurma yetkisine sahip olduğundan emin olun. Kuyruk yöneticisi ve kanal başlatıcısına ilişkin kullanıcı kimlikleri, başlatılan görev yordamlarının çalıştırıldığı kullanıcı kimlikleridir.

Yardımcı programlara ilişkin kullanıcı kimlikleri, toplu işlerin sunulabileceği kullanıcı kimlikleridir. Kullanıcı kimliğinin OKUMA erişimi gerektirdiği RACF profili, DSNR kaynak sınıfında Db2ssid . RRSF

Bir kuyruk paylaşım grubundaki her kuyruk yöneticisiyle ilişkilendirilmiş kullanıcı kimliklerinin, bağlaşımları listesi yapılarına uygun erişim düzeyine sahip olması gerekir. RACF sınıfı FACILITY.

Aşağıdaki kullanıcı kimlikleri için ALTER erişimi gerekir:

- IXLSTR . structure - name tanıtımının kuyruk yöneticisi kimliği
- CSQ5PQSG komutunu çalıştıran kullanıcı kimliği

### **İlgili kavramlar**

“ESM güvenlik denetimlerinizi uygulayın” sayfa 895

Kuyruk yöneticileri ve kanal başlatıcısı için güvenlik denetimlerini uygulayın.

## **z/OS** yapılandırma **Advanced Message Security for z/OS**

Advanced Message Security ' i yapılandırmak için adım adım bu konuları kullanın (AMS).

### **Başlamadan önce**

AMSürününü yapılandırmaya başlamadan önce, aşağıdaki kuyruk yöneticisi yapılandırma adımlarının gerçekleştirildiğinden emin olun:

1. CSQ0DRTM modülünü LPA ' ya ekleyin (açıklamalar için bkz. [“z/OS bağlantı listesini ve Yaşam Çevrimi Projesi Yönetimi” sayfa 884](#)).
2. [“z/OS program özellikleri çizelgesinin güncellenmesi” sayfa 888](#) içinde açıklandığı gibi, z/OS program özellikleri çizelgesine (PPT) CSQ0DSRV için bir giriş ekleyin.
3. CSQ4INSM üyesini [“Kullanıma hazırlama giriş veri kümelerini özelleştirme” sayfa 896](#) içinde açıklandığı gibi, kuyruk yöneticisi tarafından başlatılan görev yordamının CSQINP2 birleştirmesine ekleyin.
4. AMBPROD özniteliğini kullanarak AMS özelliğini etkinleştirin. Daha fazla ayrıntı için bkz. [IBM MQ for z/OS ürünleriyle ürün kullanımı kaydı](#) .

### **Sonraki adım**

AMS ile korunan kuyruklar için ilkeleri yapılandırın. Güvenlik ilkeleri [Advanced Message Security güvenlik ilkelerinin denetlenmesi](#) başlıklı konuda açıklanmıştır.

[z/OS üzerindeki örnek yapılandırmaları](#) içinde AMS yapılandırmalarına ilişkin örnekler vardır.

## **z/OS** **Advanced Message Security için yordamlar oluşturma**

Advanced Message Security (AMS) kullanacak şekilde yapılandırılacak her IBM MQ altsistemi, AMS adres alanını başlatmak için kataloğa alınmış bir yordam gerektirir. Kendi yordam kitaplığınızı yaratabilir ya da IBM tarafından sağlanan yordam kitaplığını kullanabilirsiniz.

### **Yordam**

1. *thlqual.SCSQPROC* (CSQ4AMSM) örnek başlatılan görev yordamını SYS1.PROCLIB ya da SYS1.PROCLIB, yordam kitaplığınız. xxxxAMSM yordamını adlandırın; burada xxxx, IBM MQ altsisteminizin adıdır. Örneğin, CSQ1AMSM , CSQ1 kuyruk yöneticisi için AMS başlatılan görev yordamı olur.
2. Kullanacağınız her IBM MQ altsistemi için bir kopya oluşturun.
3. CSQ4AMSM örnek yordamındaki yönergeleri kullanarak yordamları gereksinimlerinize göre uyarlayın. Yordamın başlatıldığında değiştirilmesine izin vermek için JCL ' de sembolik parametreleri de kullanabilirsiniz.
4. Dil Ortamı ® \_CEE\_ENVFILE dosyasını kullanarak AMS görevine geçirilen parametreleri gözden geçirin ve isteğe bağlı olarak değiştirin. Örnek *thlqual.SCSQPROC(CSQ40ENV)*, desteklenen değıştirgeleri listeler.
5. Her IBM MQ kuyruk yöneticisi için 1-4 arasındaki adımları yineleyin.

### **Sonraki adım**

[“Advanced Message Security tarafından başlatılan görev kullanıcı kimliğini ayarlama” sayfa 936](#)

## **z/OS** **Advanced Message Security tarafından başlatılan görev kullanıcı kimliğini ayarlama**

Advanced Message Security (AMS) görevi, z/OS UNIX System Services (z/OS UNIX) süreci olarak bilinmesini sağlayan bir kullanıcı kimliği gerektirir.

## Bu görev hakkında

Ayrıca, görevin adına çalıştığı kullanıcıların, bu kullanıcıların z/OS UNIX System Services kullanıcıları olarak bilinmesi için uygun bir UNIX UID (kullanıcı kimliği) ve GID (grup kimliği) tanımlaması da olmalıdır. z/OS UNIX System Services UID 'leri ve GID' leri tanımlama hakkında daha fazla bilgi için bkz. [z/OS: Security Server RACF Security Administrator's Guide](#).

Geleneksel UNIX güvenliği ile z/OS UNIX güvenliği arasındaki güvenlik farklarını anladığınızdan emin olmak için [z/OS UNIX System Services Planning](#) (Planlama) başlıklı konuya bakın. Bu, Advanced Message Security görevini kuruluşunuzun ayrıcalıklı z/OS UNIX System Services süreçlerin konuşlandırılmasına ve çalıştırılmasına ilişkin güvenlik ilkesine göre yönetmenizi sağlar.

Geleneksel UNIX güvenlik ve z/OS güvenliği arasındaki birincil fark, Kernel hizmetlerinin iki uygun ayrıcalık düzeyini desteklemeleridir: UNIX düzey ve z/OS UNIX düzeyi.

Kuruluşunuzun güvenlik ilkesine bağlı olarak, Advanced Message Security görevi ayrıcalıklı kullanıcı yetkisiyle (uid (0)) ya da RACF kimliğiyle RACF FACILITY sınıfı BPX.DAEMON ve BPX.SERVER tanıtları için izin verilen tanıtlarla çalışabilir; bu görevin, kullanıcılarının RACF kimliğini üstlenebilmesi gerekir.

İkinci yöntem kullanılırsa ya da BPX.DAEMON ya da BPX.SERVER tanıtları, Advanced Message Security görev programı (thlqual.SCSQAUTH(CSQ0DSRV)) RACF program denetimli kitaplıklarda bulunmalıdır.

**Not:** Advanced Message Security alıcı sertifikaları bu kullanıcı kimliğiyle ilişkili bir anahtarlığa yüklendiğinden, bu göreve ilişkin kullanıcı kimliğini dikkatli bir şekilde seçin. Bu husus, [z/OS üzerinde sertifikaların kullanılması](#) başlıklı konuda ele alınmıştır.

Burada gösterilen adımlar, Advanced Message Security tarafından başlatılan görev kullanıcısının nasıl ayarlanacağını açıklar. Adımlar örnek olarak RACF komutlarını kullanır. Farklı bir güvenlik yöneticisi kullanıyorsanız, eşdeğer komutlar kullanmanız gerekir.

**Not:** Bu bölümdeki örnekler, RACF BAŞLATAN, FACILITY ve TAŞIYICI sınıfları ve sosyal tanıtların denetimi için sosyal tanıtlar komutunu işlemeyi etkinleştirdiğinizi varsayar. RACF ' in genel profilleri nasıl işlediğine ilişkin daha fazla bilgi için bkz. [z/OS: Security Server RACF Command Language Reference](#).

## Yordam

1. Advanced Message Security ' in başlattığı görev kullanıcısını RACFolarak tanımlayın. Bu bölümdeki örnekler WMQAMSM kullanıcı kimliğini kullanır.

```
ADDUSER WMQAMSM NAME('AMS user') OMVS (UID(0)) DFLTGRP(group)
```

Kuruluş standartlarınıza uygun bir varsayılan 'grup' seçin.

**Not:** z/OS UNIX ayrıcalıklı kullanıcı yetkisi (UID (0)) vermek istemiyorsanız, Advanced Message Security kullanıcı kimliğine BPX.DAEMON ve BPX.SERVER olarak sınıfı tanıtları:

```
PERMIT BPX.DAEMON CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
```

ve Advanced Message Security görev programı (thlqual.SCSQAUTH (CSQ0DSRV)) RACF program denetimli bir kitaplıkta bulunmalıdır.

SCSQAUTH kitaplık programınızın denetlenmesi için aşağıdaki komutu kullanabilirsiniz:

```
RALTER PROGRAM * ADDMEM('thlqual.SCSQAUTH'//NOPADCHK) -or-  
RALTER PROGRAM ** ADDMEM('thlqual.SCSQAUTH'//NOPADCHK)  
SETOPTS WHEN(PROGRAM) REFRESH
```

Advanced Message Security görevi tarafından kullanılan ulusal dil kitaplığı (thlqual.SCSQANLx) için de program denetimini etkinleştirmeniz gerekir.

2. RACF BAŞLATMA sınıfının etkin olup olmadığını belirleyin. Değilse, RACF BAŞLADI sınıfını etkinleştirin:

```
SETOPTS CLASSACT(STARTED)
```

3. Adım 1 'de seçtiğiniz ya da yarattığınız kullanıcı kimliğini belirterek, Advanced Message Security görevleri için başlatılan bir sınıf profili tanımlayın:

```
RDEFINE STARTED qmgrAMSM.* STDATA(USER(WMQAMSM))
```

Burada *qmgr* , başlatılan görev adının önekidir. Örneğin, başlatılan görev CSQ1AMSMolarak adlandırılabilir. Bu durumda, *qmgrAMSM.\** yerine CSQ1AMSM.\* yazın.

AMS ' in başlattığı görevler *qmgrAMSM*olarak adlandırılmalıdır.

4. Depolama alanı içindeki RACLİSTed BAŞLATILMIŞ sınıf profillerini yenilemek için **SETROPTS** RACF komutunu kullanın:

```
SETROPTS RACLIST(STARTED) REFRESH
```

5. Advanced Message Security görevi, IBM MQ iletilerinin korunması sırasında istekte bulunanın anasistem kullanıcı kimliğini geçici olarak varsayar. Bu nedenle, istekte bulunabilecek her kullanıcı kimliği için TAŞIYICI sınıfında tanımlar tanımlanması gerekir.

RACF VEKIL sınıfı etkinse, tek bir sosyal profil tanımlanması Advanced Message Security görevinin herhangi bir kullanıcının kimliğini üstlenmesini sağlar. VEKILGAT sınıfı etkin değilse, denetim yoksayılr. Gerekli VEKIL profilleri, [z/OS UNIX System Services Planning](#)belgesinde açıklanmıştır.

VEKILGAT sınıfında tanımları tanımlamak için:

- a) RACF SETROPTS komutunu kullanarak RACF VEKILGAT sınıfını etkinleştirin:

```
SETROPTS CLASSACT(SURROGAT)
```

- b) RACF VEKILGAT sınıfı için sosyal tanım işlemeyi etkinleştir:

```
SETROPTS GENERIC(SURROGAT)
```

- c) RACF VEKILGAT sınıfı için sosyal tanım komutunun işlenmesini etkinleştir:

```
SETROPTS GENCMD(SURROGAT)
```

- d) VEKILGAT sınıfında sosyal bir tanım tanımlayın:

```
RDEFINE SURROGAT BPX.SRV.* UACC(NONE)
```

- e) Advanced Message Security kullanıcı kimliğinin sosyal TAŞIYICI sınıfı tanımına izin ver:

```
PERMIT BPX.SRV.* CLASS(SURROGAT) ID(WMQAMSM) ACCESS(READ)
```

**Not:** Belirli kullanıcıların Advanced Message Security görevi tarafından işlenmesini kısıtlamak istiyorsanız, [z/OS UNIX System Services Planlamak](#) konusunda açıklandığı gibi daha özel profiller tanımlayabilirsiniz.

Örneğin, BPX.SRV.MQUSER1 adlı bir profil, AMS görevinin MQUSER1 kullanıcı kimliğinin üstlenip üstlenemeyeceğini denetler.

- f) Advanced Message Security kullanıcı kimliğine BPX.SERVER olanağı için izin verin (önceden Sertifika ve anahtar halkaları oluşturma içinde yapılmadıysa):

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
```

- g) Depolama alanı içindeki RACLİSTed başlatılan sınıf profillerini yenilemek için **SETROPTS** RACF komutunu kullanın:

```
SETROPTS RACLIST(SURROGAT) REFRESH
SETROPTS RACLIST(FACILITY) REFRESH
```

6. Advanced Message Security görevi, SAF tarafından yönetilen anahtar halkalarını açmak için z/OS System SSL hizmetleri tarafından sağlanan olanakları kullanır. Anahtar halkalarının içeriğine erişen temel Sistem Yetki Olanağı (SAF), RACFya da eşdeğer bir güvenlik yöneticisi tarafından denetlenir.

Bu hizmet, IRRSDL00 (R\_data lib) çağrılabilir hizmetidir. Bu çağrılabilir hizmet, RACF FACILITY sınıfında tanımlanan RACF RACDCERT komutlarını korumak için kullanılan tanıtımlarla korunur. Bu nedenle, Advanced Message Security kullanıcı kimliğinin aşağıdaki komutları kullanarak tanıtımlara izin verilmelidir:

- a) Henüz yapmadıysanız, RACF FACILITY sınıfına, RACDCERT komutunu ve IRRSDL00 çağrılabilir hizmetini koruyan bir RACF soysal tanıtımları tanımlayın:

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
SETROPTS RACLIST(FACILITY) REFRESH
```

- b) Başlatılan görev kullanıcı kimliği için RACF soysal tanıtımlarına yetki verin:

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(WMQMSM) ACC(READ)
```

Diğer bir seçenek olarak, RDATA LIB sınıfında veri hizmeti görevi kullanıcısının anahtarlığı için aşağıdaki gibi okuma erişimi verebilirsiniz:

```
PERMIT WMQMSM.DRQ.AMS.KEYRING.LST CLASS(RDATA LIB) ID(WMQMSM) ACC(READ)
```

7. Kaynak güvenliğini yapılandır:

- a) Advanced Message Security başlatılan görev kullanıcısı, toplu iş uygulaması olarak kuyruk yöneticisine bağlanmak için yetki gerektirir.

Kuyruk yöneticinizde bağlantı güvenliği etkinleştirildiyse, AMS görev yetkisine aşağıdaki komutu kullanarak kuyruk yöneticisine bağlanması için yetki verin:

```
PERMIT hlq.BATCH CLASS(MQCONN) ID(WMQMSM) ACC(READ)
```

burada *hlq* , kuyruk yöneticisi ad kuyruğu paylaşım grubu adı olabilir.

Daha fazla bilgi için bkz. [Toplu bağlantılar için bağlantı güvenlik profilleri](#).

- b) Advanced Message Security başlatılan görev kullanıcısı, SYSTEM.PROTECTION.POLICY.QUEUE.

Kuyruk yöneticisinde kuyruk güvenliği etkinse, AMS kullanıcısına şu komutlarla kuyruğa erişmesi için yetki verin:

```
RDEFINE MQQUEUE hlq.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
PERMIT hlq.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE) ID(WMQMSM) ACCESS(READ)
```

burada *hlq* , kuyruk yöneticisi ad kuyruğu paylaşım grubu adı olabilir.

Kuyruk yöneticisi büyük ve küçük harf karışık profiller kullanıyorsa, bunun yerine MXQUEUE sınıfında profili tanımlayın.

AMS güvenlik ilkelerini CSQOUTIL yardımcı programını kullanarak yönetmek için denetimcilerin SYSTEM.PROTECTION.POLICY.QUEUE. Bu, kuyruğu koruyan tanıtımlar için UPDATE erişimi verilmesiyle gerçekleştirilir.

Daha fazla bilgi için bkz. [Kuyruk güvenliği için profiller](#).

## Sonraki adım

[“Advanced Message Security için güvenlik yöneticisine RACDCERT izinleri verin” sayfa 940](#)



## **z/OS** **Advanced Message Security için güvenlik yöneticisine RACDCERT izinleri verin**

Advanced Message Security güvenlik denetimciniz, sayısal sertifikaları yaratmak ve yönetmek için RACDCERT komutunu kullanma yetkisine sahip olmalı.

### **Yordam**

- Bu rol için uygun kullanıcı kimliğini saptayın ve RACDCERT komutunu kullanma izni verin. Örneğin:

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(admin) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

Burada admin , Advanced Message Security güvenlik yöneticinizin kullanıcı kimliğidir.

### **Sonraki adım**

[“Advanced Message Security için kullanıcılara kaynak izinleri verin” sayfa 940](#)

## **z/OS** **Advanced Message Security için kullanıcılara kaynak izinleri verin**

Advanced Message Security kullanıcıları ilgili kaynak izinlerine gereksinim duyar.

### **Bu görev hakkında**

Advanced Message Security kullanıcıları (yani, Advanced Message Security korumalı iletileri yerleştirecek ya da alacak olan kullanıcılar) aşağıdaki işlemleri gerektirir:

- Kullanıcı kimliğiyle ilişkilendirilmiş bir OMVS bölümü
- IRR.DIGTCERT.LISTRING ya da RDATA LIB
- ICSF sınıfı CSFSERV ve CSFKEYS tanımları için izinler
- SYSTEM.PROTECTION.ERROR.QUEUE

Advanced Message Security görevi geçici olarak istemcilerinin kimliğini varsayar; yani, görev, Advanced Message Security tarafından korunan kuyruklara gönderilen IBM MQ iletilerinin işlenmesi sırasında Advanced Message Security kullanıcısının z/OS kullanıcı kimliğinin yerine geçer.

Görevin bir kullanıcının z/OS kimliğini üstlenebilmesi için, istemci z/OS kullanıcı kimliğinin kullanıcı tanımıyla ilişkilendirilmiş tanımlı bir OMVS kesimi olmalıdır.

Bir yönetim yardımı olarak RACF , RACF kullanıcı ve grup profilleriyle ilişkilendirilebilecek varsayılan bir OMVS kesimi tanımlama yeteneği sağlar. z/OS kullanıcı kimliği ya da grup tanımının açık olarak tanımlanmış bir OMVS kesimi yoksa, bu varsayılan değer kullanılır. Advanced Message Security' u kullanan çok sayıda kullanıcıya sahip olmayı planlıyorsanız, her kullanıcı için OMVS kesimini belirttik olarak tanımlamak yerine bu varsayılan değeri kullanmayı seçebilirsiniz.

[z/OS: Security Server RACF Security Administrator's Guide](#) , varsayılan OMVS bölümlerinin tanımlanmasına ilişkin ayrıntılı yordamı içerir. RACF Kullanıcı ve Grup profillerindeki varsayılan OMVS bölümlerinin tanımının kuruluşunuz için uygun olup olmadığını belirlemek için bu yayında açıklandığı şekilde yordamı gözden geçirin.

### **Yordam**

1. IRR.DIGTCERT.LISTRING tanıtımı:

- IRR.DIGTCERT.LISTRING tanıtımı, şu komutu verin:

```
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(READ)
```

- IRR.DIGTCERT.LISTRING tanıtımı, kullanıcı temelinde şu komutu verin:



```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(READ)
```

Burada kullanıcı kimliği, Advanced Message Security kullanıcısının adıdır.

- Diğer bir seçenek olarak, belirli anahtar halkaları için erişim vermek üzere RDATA LIB sınıfını kullanın. RDATA LIB izinleri, IRR.DIGTCERT.LISTRING izinleri. Örneğin:

```
PERMIT user.DRQ.AMS.KEYRING.LST CLASS(RDATA LIB) ID(user) ACC(READ)
```

2. ICSF tarafından yönetilen sertifikaları ve özel anahtarları kullanıyorsanız, Advanced Message Security kullanıcılarının belirli sınıf CSFSERV ve CSFKEYS profillerine erişmeleri gerekir. Bu erişim aşağıdaki tabloda ayrıntılı olarak açıklanmıştır:

*Çizelge 63. Sınıf CSFSERV ve CSFKEYS tanımları için gerekli kullanıcı erişimi*

Sınıf	Profil	İzin
CSFSERV	CSFDSG	READ
CSFSERV	CSFPKE	READ
CSFSERV	CSFPKD	READ
CSFSERV	CSFDSV	READ
CSFKEYS	ICSF PKDS Etiketi	READ

3. AMS ilkeleri tanımlanmış kuyruklarda işlem gerçekleştiren uygulamaların, iletileri SYSTEM.PROTECTION.ERROR.QUEUE. Şu komutlarla kuyruğa koyma erişimi ver:

```
RDEFINE MQQUEUE h1q.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)  
PERMIT h1q.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE) ID(userId) ACCESS(UPDATE)
```

Burada *h1q*, kuyruk yöneticisi adı kuyruğu paylaşım grubu adı, *userId* ise uygulama kullanıcı kimliği olabilir.

## Sonraki adım

[“Advanced Message Security için anahtar halkaları oluşturma” sayfa 941](#)

## Advanced Message Security için anahtar halkaları oluşturma

Advanced Message Security (AMS) tarafından imzalama ve şifreleme için kullanılan sertifikalar, z/OS SAF anahtar halkalarında saklanır. AMS' u kullanmadan önce bu anahtar halkalarını ve sertifikaları oluşturmanız gerekir.

## Bu görev hakkında

Advanced Message Security , aşağıdaki anahtar halkasındaki sertifikalara erişir:

- AMS adres alanı kullanıcısının sahip olduğu tek bir anahtarlık.
- AMS ilkeleri tanımlanmış olan kuyruklarda ileti gönderen ya da alan tek tek kullanıcıların sahip olduğu anahtarlar.

Bu anahtar halkalarının tümü `drq.ams.keyring` olarak adlandırılmalıdır.

Anahtar halkaları ve AMStarafından kullanılan sertifikalar ve örnek bir senaryo hakkında daha fazla bilgi için [z/OSüzerinde sertifikaları kullanmak](#) kısmına bakın.

AMS' in gerektirdiği anahtar halkalarını oluşturmak ve sertifikaları anahtar halkalarına bağlamak için bu adımları izleyin. AMS' u başlatmadan önce AMS adres alanı kullanıcısının sahip olduğu anahtarlığı oluşturmanız gerekir. İstedığınız zaman ileti gönderen ya da alan kullanıcıların sahip olduğu anahtar halkalarını oluşturabilirsiniz.

## Yordam

1. AMS adres alanı kullanıcısının sahip olduğu bir anahtarlık oluşturmak için aşağıdaki komutu verin:

```
RACDCERT ID(amsUser) ADDRING(dıq.ams.keyring)
```

Burada *amsUser* , AMS adres alanının kullanıcı kimliğidir.

2. AMS ile korunan iletileri gönderen ya da alan her kullanıcı için, her kullanıcı kimliği için 1. adımda komutu vererek bir anahtarlık oluşturun.
3. Kullanıcı sertifikalarını veren kişi için sertifika yetkilisi (CA) sertifikasını, AMS adres alanı kullanıcı kimliğinin sahip olduğu anahtar halkasına bağlayın. Şu komutu verin:

```
RACDCERT ID(amsUser) CONNECT(CERTAUTH LABEL('caLabel') RING(dıq.ams.keyring))
```

Burada *amsUser* AMS adres alanının kullanıcı kimliği ve *caLabel* CA sertifikasının etiketidir.

CA olarak RACF kullanıyorsanız ve bir sertifika yetkilisi sertifikası oluşturmanız gerekiyorsa, [Yerel Sertifika Yetkilisi sertifikası tanımlamabaşlıklı](#) konudaki örneği izleyin.

4. AMStarafından korunan kuyruklardaki iletileri şifrelemek için gizlilik ya da gizlilik güvenlik ilkelerini kullanıyorsanız, ileti alıcılarının sertifikalarını AMS adres alanı kullanıcı kimliğinin sahip olduğu anahtarlığa bağlayın. Şu komutu verin:

```
RACDCERT ID(amsUser) CONNECT(ID(userId) LABEL('certLabel')  
RING(dıq.ams.keyring) USAGE(SITE))
```

Burada *amsUser* AMS adres alanının kullanıcı kimliği, *userId* ileti alıcısı ve *certLabel* kullanıcı sertifikasının etiketidir.

USAGE (SITE) özniteliği, özel anahtarın anahtarlık içinde erişilebilir olmasını önler.

RACFile kendi sertifikalarınızı oluşturuyorsanız, sertifikayı oluşturmak için [Özel anahtarla dijital sertifika oluşturma](#) başlıklı konudaki örneği izleyin.

5. AMS tarafından korunan iletileri gönderen ya da alan her kullanıcının sertifikalarını, kullanıcının sahip olduğu bir anahtarlık anahtarına bağlayın. Sertifika, anahtar halkasında varsayılan sertifika olarak bağlanmalıdır. Şu komutu verin:

```
RACDCERT ID(userId) CONNECT(ID(userId) LABEL('certLabel')  
RING(dıq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

Burada *userId* ileti gönderen ya da alan kullanıcı, *certLabel* ise kullanıcının sertifikasının etiketidir.

### Notlar:

- a. Uygulama yalnızca çıkış için bir kuyruk açarsa ve bir AMS gizlilik ilkesiyle korunan kuyruklara ileti gönderirse, [“2” sayfa 942](#) ve [“5” sayfa 942](#) adımlarına gerek yoktur.
- b. Uygulama bir kuyruğu yalnızca giriş/göz atma için açarsa ve bir AMS bütünlük ilkesiyle korunan kuyruklardan ileti alıyorsa, [“2” sayfa 942](#) ve [“5” sayfa 942](#) adımlarına gerek yoktur.

## Sonraki adım

[“EtkinleştirAdvanced Message Security” sayfa 942](#)

## EtkinleştirAdvanced Message Security

Bir kuyruk yöneticisine ilişkin güvenlik ilkesi yeteneği, sistem parametre modülündeki SPLCAP parametresi tarafından denetlenir.

## Bu görev hakkında

Advanced Message Security (AMS) olanağını tek bir kuyruk yöneticisi için etkinleştirmek üzere aşağıdaki adımları izleyin.

Bu görev, sistem parametresi modülünde değişiklik yapmanızı gerektirir. Sistem değiştirgesi modülünün yaratılması ve uyarlanmasıyla ilgili ek bilgi için bkz. [“Sistem parametre modülünüzü uyarlama, da,-” sayfa 903](#).

## Yordam

1. CSQ6SYSP' de **SPLCAP** değerini YES olarak ayarlayın. CSQ6SYSP makrosuna ilişkin ek bilgi için bkz. [“CSQ6SYSP ' nin kullanılması” sayfa 905](#).
2. Lisans yetkinize bağlı olarak **AMSPROD** değerini AMS, ADVANCED ya da ADVANCED olarak ayarlayın. CSQ6USGP makrosuna ilişkin ek bilgi için [CSQ6USGP ' nin kullanılması](#) konusuna bakın.
3. Sistem parametresi modülünü yeniden derleyin.
4. Güncellenen sistem parametresi modülüyle kuyruk yöneticisini yeniden başlatın. AMS adres alanı, kuyruk yöneticisi başlatıldığında otomatik olarak başlatılır.

## **mqweb sunucusunu yapılandırma**

mqweb sunucusunu yapılandırmak için adım adım bu konuları kullanın.

### İlgili görevler

[“IBM MQ Console ve REST API ' nin yapılandırılması” sayfa 791](#)

IBM MQ Console ve REST API öğelerini barındıran mqweb sunucusu varsayılan bir yapılandırma ile sağlanır. Bu bileşenlerden birini kullanabilmek için, kullanıcıların oturum açmasına izin verecek şekilde güvenlik yapılandırılması gibi, bir dizi yapılandırma görevinin tamamlanması gerekir. Bu konuda, kullanılabilir tüm yapılandırma seçenekleri açıklanmaktadır.

## **mqweb sunucusu yaratılıyor**

IBM MQ for z/OS UNIX System Services Web Components ürününü kurduysanız ve IBM MQ Console'ya da REST API'olanağını kullanmak istiyorsanız, mqweb sunucusunu yaratmanız ve uyarmanız gerekir.

## Başlamadan önce

mqweb sunucusunu yaratmak için **crtmqweb** komut dosyasını çalıştırmadan önce, JAVA\_HOME ortam değişkenini sisteminizdeki Java ' un 64 bit sürümüne gönderme yapacak şekilde ayarlayın.

IBM MQ Console ve administrative REST API , SYSTEM.REST.REPLY.QUEUE kuyruğu. [“Kullanıma hazırlama giriş veri kümelerini özelleştirme” sayfa 896](#) içindeki **CSQ4INSG** örneğini kullanarak bu kuyruğu oluşturun.



**Uyarı:** mqweb sunucusunu başlatırken, aşağıdaki çıkışta gösterildiği gibi CWWKG0014Ehata iletilisiyle karşılaşabilirsiniz:

```
Launching mqweb (MQM MVS/ESA V9 R2.0/wlp..) (en_US)
YAUDIT   CWWKE0001I: The server mqweb has been
launched.
          YWARNING  CWWKF0009W: The server has not been configured to install any
features.
          YAUDIT   CWWKF0011I: The mqweb server is ready to run a smarter planet.
The mqweb server started in 6.348 seconds.
          YERROR   CWWKG0014E: The configuration parser detected an XML syntax
error while parsing the root of the configuration and the referenced configuration
documents.
Error: An invalid XML character (Unicode: 0x4c) was found
in the prolog of the document.
          File: file:<your filepath>/servers/mqweb/server.xml Line:
1 Column: 1
```

AUTOCVT z/OS ayarını denetleyip (dosyaları otomatik olarak bir kod kümesinden diğerine dönüştürün) aşağıdakilerden birini yaparak değeri gereken şekilde ayarlamalısınız.

### USS uçbiriminde:

Bu ortam değişkeninin değerini görüntülemek için şu komutu girin: echo **\$\_BPXK\_AUTOCVT** .  
Ortam değişkeni tanımlı değilse, değer görüntülenmez.

Ortam değişkenini ayarlamak için bkz. [\\_BPXK ortam değişkenleri](#).

### Sistem genelinde:

z/OS UNIX System Services (OMVS) durumunun görüntülenmesi başlıklı 6. örnek, BPXPRMxx içindeki sistem genelinde AUTOCVT deyiminin değerinin nasıl görüntüleneceğini gösterir.

Ortam değişkeni sistemini geniş bir alana ayarlamak için BPXPRMxx içindeki [AUTOCVT](#) deyimini kullanın.

USS uçbiriminde \_BPXX\_AUTOCVT ortam değişkeni ayarlanırsa, BPXPRMxx içindeki sistem genelinde deyim AUTOCVT ayarını geçersiz kılar.

## Bu görev hakkında

- IBM MQ Console ya da REST API programını çalıştırmak istediğiniz her z/OS sistemi için bu görevi bir kez tamamlayın.
- administrative REST API olanağını kullanmak için, çalışmakta olan her IBM MQ sürümü için bir mqweb sunucusu gerekir. Örneğin, IBM MQ 9.3.0, 9.2.5 ve 9.2.0 işletim sistemlerinde çalışıyorsanız, üç farklı mqweb sunucusu gerekir.
- Önceki bir sürümden geçiş yaparken sunucu yapılandırmasını yenilemeniz ya da değiştirmeniz gerekebilir.

IBM MQ Console ve REST API , mqweb adı verilen tek bir WebSphere Liberty sunucusunun yaratılmasını gerektirir.

Sunucu yapılandırma ve günlük dosyalarının tümü Liberty kullanıcı dizini altında saklanır.

mqweb sunucusunun, altında çalıştığı bir ürün tanıtıcısıyla (PID) yapılandırılması gerekir. PID, mqweb sunucusu yaratıldığında ayarlanır. mqweb sunucusunun bağlandığı yerel kuyruk yöneticilerini çalıştırmak için kullanılan PID ' yi kullanın.

**Not:** Yerel kuyruk yöneticileri birden çok farklı PID altında çalışıyorsa, altında mqweb sunucusunu çalıştırmak için bunlardan birini seçin.

PID ' ler ve bunların z/OS üzerinde nasıl kullanıldıkları hakkında daha fazla bilgi için bkz. [IBM MQ for z/OS ürünleriyle ürün kullanımı kaydı](#).

[setmqweb](#) komutunu kullanarak, mqweb sunucusu yaratıldıktan sonra altında çalışan PID ' yi değiştirebilirsiniz.

mqweb sunucusunu yaratmak için aşağıdaki adımları izleyin:

## Yordam

1. mqweb sunucusunun hangi PID altında çalıştığına karar verin.
2. Liberty kullanıcı dizini için uygun bir konum seçin.

mqweb sunucusunun altında çalıştığı kullanıcı kimliği, bu kullanıcı dizini ve içeriği için okuma ve yazma erişimi gerektirir. Bu kullanıcı dizini günlük dosyalarını içerdiğinden, sunucu yapılandırmasına ek olarak bu dizini ayrı bir dosya sisteminde oluşturun.

**Not:** mqweb sunucusu başlatıldığında önemli miktarda disk G/Ç vardır. mqweb sunucusunu başlatma süresini kısaltmak için hem IBM MQ kuruluş z/OS UNIX dosya sisteminin hem de Liberty kullanıcı dizini dosya sisteminin sysplex-farkındalığı olduğundan ya da mqweb sunucusunun çalıştığı sistemde yerel olarak sisteme bağlandığından emin olun.

3. z/OS UNIX System Services içinde, aşağıdaki komutu vererek geçerli çalışma dizininizi PathPrefix/web/bin olarak değiştirin:

```
cd PathPrefix/web/bin
```

Burada PathPrefix , IBM MQ for z/OS UNIX System Services Components kuruluş yoludur.

4. **crtmqweb** komut dosyasını çalıştırarak, şablon mqweb sunucusu tanımlamasını içeren Liberty kullanıcı dizinini yaratın.

**crtmqweb** komutunun biçimi şöyledir:

```
crtmqweb user_directory -p pid_value
```

Burada:

### **kullanıcı\_dizini**

“2” sayfa 944. adımda Liberty kullanıcı dizini kararlaştırılıyor. Bu parametre isteğe bağlıdır. If this parameter is not specified, then a default Liberty user directory of /var/mqm/web/installation1 is used.

### **pid\_değeri**

mqweb sunucusunun altında çalıştığı PID 'yi gösterir. Bu PID, “1” sayfa 944. adımda seçtiğiniz PID 'dir. *pid\_value* aşağıdaki değerlerden biridir:

#### **MQ**

mqweb sunucusu PID IBM MQ for z/OS (5655-MQ9) altında çalışır.

#### **VUE**

mqweb sunucusu PID IBM MQ for z/OS Value Unit Edition (VUE) (5655-VU9) altında çalışır.

#### **GELİŞEN**

mqweb sunucusu PID IBM MQ Advanced for z/OS VUE (5655-AV1) altında çalışır,

Örneğin, mqweb sunucusunu Liberty /usr/mqweb kullanıcı dizini ve PID IBM MQ Advanced for z/OS VUE (5655-AV1) ile yaratmak için aşağıdaki komutu çalıştırın:

```
./crtmqweb /usr/mqweb -p ADVANCEDVUE
```

5. Liberty kullanıcı dizinindeki dizinlerin ve dosyaların sahipliğini, mqweb sunucusunun altında çalıştığı kullanıcı kimliğine ve gruba ait olması için aşağıdaki komutu kullanarak değiştirin:

```
chown -R userid:group path
```

Gruba yol için yazma erişimi vermek üzere şu komutu verin:

```
chmod -R 770 path
```

## **Sonraki adım**

“mqweb sunucusu için yordam yaratılması” sayfa 945

### **İlgili görevler**

“IBM MQ Console ve REST API 'nin yapılandırılması” sayfa 791

IBM MQ Console ve REST API öğelerini barındıran mqweb sunucusu varsayılan bir yapılandırmayla sağlanır. Bu bileşenlerden birini kullanabilmek için, kullanıcıların oturum açmasına izin verecek şekilde güvenlik yapılandırılması gibi, bir dizi yapılandırma görevinin tamamlanması gerekir. Bu konuda, kullanılabilir tüm yapılandırma seçenekleri açıklanmaktadır.

### **z/OS mqweb sunucusu için yordam yaratılması**

IBM MQ for z/OS UNIX System Services Web Componentskurulursa ve IBM MQ Consoleya da REST APIkullanmak istiyorsanız, mqweb sunucusunu başlatmak için kataloğa alınmış bir yordam yaratmanız gerekir. mqweb sunucusu, IBM MQ Console ve REST API'yi barındıran bir Liberty sunucusudur.

- Bu görevi, IBM MQ Console ya da REST APIprogramını çalıştırmak istediğiniz her z/OS sistemi için bir kez gerçekleştirmeniz gerekir.
- Çalışmakta olan her IBM MQ sürümü için bir mqweb sunucusu gerekir. Örneğin, IBM MQ for z/OS 9.1.0 'deki kuyruk yöneticileri için MQWB0910 adlı başlatılan bir görev ve IBM MQ for z/OS 9.0.5adresindeki kuyruk yöneticileri için MQWB0905 adlı bir görev.

z/OS sisteminde tek bir kuyruk yöneticiniz varsa, tek bir Liberty sunucusu başlatma görevini çalıştırabilir ve kuyruk yöneticinizi geçirirken kullandığı kitaplıkları değiştirebilirsiniz.

- Önceki bir sürümden geçiş yaparken kataloğa alınmış yordamı değiştirmeniz gerekebilir.

Kataloğa alınmış bir yordam yaratmak için aşağıdaki yordamı gerçekleştirin:

1. th1qua1.SCSQPROC (CSQ4WEBS) adlı örnek başlatılan görev yordamını yordam kitaplığınıza kopyalayın.

Yordamı işletmenizin standartlarına göre adlandırın.

Örneğin, bunun IBM MQ for z/OS 9.1.0 mqweb sunucusu için kataloğa alınmış yordam olduğunu gösteren MQWB0910.

2. CSQ4WEBSörnek yordamındaki yönergeleri kullanarak yordamı gereksinimlerinize göre uyarlayın.

Liberty kullanıcı dizininin, mqweb sunucusu tanımlamasını yaratmak için **crtmqweb** komut dosyası çalıştırıldığında belirtilen izin olduğunu unutmayın.

Ayrıntılar için bkz. [“mqweb sunucusu yaratılıyor” sayfa 943.](#)

**Not:** Dosyayı küçük harfli veriler içerdiği için üyeyi düzenlerken **Caps off** (kapalı) değerini belirttiğinizden emin olun.

3. Yordamın dış güvenlik yöneticiniz altında çalıştırılması için yetki verin.

4. Bu adres alanını sınıflandırmak için IBM Workload Manager (WLM) olanağını kullanın.

mqweb sunucusu bir IBM MQ uygulamasıdır ve kullanıcılar bu uygulamayla etkileşimde bulunur. Uygulamanın WLM ' de yüksek önem taşıması gerekmez ve **STCUSER** hizmet sınıfı uygun olabilir.

## Sıradaki eylem

mqweb sunucusunu yapılandırmayı tamamlamak için [“mqweb sunucusu için temel yapılandırma” sayfa 791](#) içindeki adımları izleyin.

### İlgili görevler

[“IBM MQ Console ve REST API ' nin yapılandırılması” sayfa 791](#)

IBM MQ Console ve REST API öğelerini barındıran mqweb sunucusu varsayılan bir yapılandırma ile sağlanır. Bu bileşenlerden birini kullanabilmek için, kullanıcıların oturum açmasına izin verecek şekilde güvenlik yapılandırılması gibi, bir dizi yapılandırma görevinin tamamlanması gerekir. Bu konuda, kullanılabilir tüm yapılandırma seçenekleri açıklanmaktadır.

z/OS

## z/OS üzerinde bir kuyruk yöneticisinin sınanması

Kuyruk yöneticinizi özelleştirdiğinizde ya da yeni düzeye geçirdiğinizde, kuruluş doğrulama programlarını ve IBM MQ for z/OS ile birlikte gönderilen bazı örnek uygulamaları çalıştırarak bu programı sınavabilirsiniz.

### Bu görev hakkında

IBM MQ for z/OS ürününü kurduktan ve uyarladıktan sonra, IBM MQ for z/OS ' in çalıştığını doğrulamak için sağlanan kuruluş doğrulama programını ( CSQ4IVP1) kullanabilirsiniz.

Temel kuruluş doğrulama programı CSQ4IVP1 , paylaşılmayan kuyrukları sınar ve C, COBOL ya da CICS örneklerini kullanmadan IBM MQ tabanını doğrular.

Temel kuruluş doğrulamasını çalıştırdıktan sonra, CSQ4IVP1 komutunu farklı kuyruklarla kullanarak paylaşılan kuyrukları sınavabilir ve Db2 ile bağlantı olanağının doğru kurulduğunu sınavabilirsiniz. Dağıtılmış kuyruğa alma işleminin çalıştığını doğrulamak için, sağlanan kuruluş doğrulama programını ( CSQ4IVPX) kullanabilirsiniz.

CSQ4IVP1 , bir yükleme modülü olarak verilir ve İleti Kuyruğu Arabirimi 'nin (MQI) tipik kullanımlarını gösteren kaynak modüller olarak bir dizi yordamsal örnek uygulama sağlar. Farklı programlama dili ortamlarını sınamak için bu kaynak modülleri kullanabilirsiniz. Sağlanan örnek JCL ' yi kullanarak, diğer örneklerden hangisi kuruluşunuz için uygunsa, bunları derleyebilir ve düzenleyebilirsiniz.

### Yordam

- Kuyruk yöneticinizi z/OS üzerinde nasıl sınavacağınıza ilişkin bilgi için aşağıdaki alt konulara bakın:
  - [“Temel kuruluş doğrulama programının çalıştırılması” sayfa 947](#)

- [“Kuyruk paylaşım grupları için test ediliyor” sayfa 950](#)
- [“Dağıtılmış kuyruğa alma testi” sayfa 951](#)
- [“IBM MQ for z/OS ile C, C + +, COBOL, PL/I ve CICS programları için test” sayfa 954](#)

### **İlgili kavramlar**

[IBM MQ for z/OS kavramlar](#)

### **İlgili görevler**

[z/OS üzerinde IBM MQ ortamının planlanması](#)

[“z/OS üzerinde kuyruk yöneticilerini yapılandırma” sayfa 876](#)

[IBM MQ for z/OS üzerinde kuyruk yöneticilerini yapılandırmak için bu yönergeleri kullanın.](#)

[Yönetme IBM MQ for z/OS](#)

## **z/OS Temel kuruluş doğrulama programının çalıştırılması**

IBM MQ ürününü kurduktan ve uyarladıktan sonra, IBM MQ ' in çalıştığını doğrulamak için sağlanan kuruluş doğrulama programını ( CSQ4IVP1) kullanabilirsiniz.

Temel kuruluş doğrulama programı, temel IBM MQ ' i C, COBOL ya da CICS örneklerini kullanmadan doğrulayan bir toplu iş çevirici IVP ' dir.

Toplu İş çevirici IVP ' si SMP/E tarafından düzenlenir ve yükleme modülleri thlqual.SCSQLOAD kitaplığında gönderilir.

Hem SMP/E APPLY adımını, hem de özelleştirme adımlarını tamamladıktan sonra Toplu Assembler IVP ' yi çalıştırın.

Daha fazla ayrıntı için bu bölümlere bakın:

- [CSQ4IVP1 uygulamasına genel bakış](#)
- [CSQ4IVP1 ürününü çalıştırmaya hazırlanıyor](#)
- [CSQ4IVP1](#)
- [CSQ4IVP1 sonuçlarının denetlenmesi](#)

### **CSQ4IVP1 uygulamasına genel bakış**

CSQ4IVP1 , IBM MQ altsisteminize bağlanan ve aşağıdaki temel işlevleri gerçekleştiren bir toplu iş uygulamasıdır:

- Sorunlar IBM MQ çağrılarını
- Komut sunucusuyla iletişim kurar
- Tetikleyicinin etkin olduğunu doğrular
- Dinamik bir kuyruk oluşturur ve siler
- İleti süre bitimi işlemlerini doğrular
- İleti kesinleştirme işlemlerini doğrular

### **CSQ4IVP1 çalıştırılmaya hazırlanıyor**

CSQ4IVP1:

1. IVP girişlerinin kuyruk yöneticisi başlatma programındaki CSQINP2 veri kümesi birleşiminde olup olmadığını denetleyin. IVP girişleri thlqual.SCSQPROC(CSQ4IVPQ) üyesinde sağlanır. Yoksa, thlqual.SCSQPROC(CSQ4IVPQ) içinde sağlanan tanımlamaları CSQINP2 birleşiminize ekleyin. Kuyruk yöneticisi çalışıyorsa, bu tanımlamaların yürürlüğe girmesi için yeniden başlatmanız gerekir.
2. Kuruluş doğrulama programını çalıştırmak için gereken örnek JCL ( CSQ4IVPR) thlqual.SCSQPROCK kitaplığında bulunur.



CSQ4IVPR JCL ' yi IBM MQ kitaplıkları için üst düzey niteleyici, kullanmak istediğiniz ulusal dil, dört karakterli IBM MQ kuyruk yöneticisi adı ve iş çıkışına ilişkin hedef ile uyarlayın.

3. RACF dosyasını, IBM MQ güvenliği etkinse CSQ4IVP1 ' in kaynaklarına erişmesine izin verecek şekilde güncelleyin.

IBM MQ güvenliği etkinleştirildiğinde CSQ4IVP1 komutunu çalıştırmak için, nesnelere erişim yetkisi olan bir RACF kullanıcı kimliği gerekir. RACF'e kaynak tanımlanmasına ilişkin ayrıntılar için [z/OS ' da güvenliği ayarlama](#) başlıklı konuya bakın. IVP ' yi çalıştıran kullanıcı kimliğinin aşağıdaki erişim yetkisi olmalıdır:

Authority	Profil	Sınıf
READ	ssid.DISPLAY.PROCESS	MQCMDS
GÜNCELLE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
GÜNCELLE	ssid.SYSTEM.COMMAND.REPLY.MODEL	MQQUEUE
GÜNCELLE	ssid.CSQ4IVP1.**	MQQUEUE
READ	ssid.BATCH	MQCONN

Bu gereksinimler, tüm IBM MQ güvenliğinin etkin olduğunu varsayar. IBM MQ güvenliğini etkinleştirmek için RACF komutları **Şekil 101** sayfa 948'içinde gösterilir. Bu örnekte, kuyruk yöneticisi adının CSQ1 olduğu ve örnek CSQ4IVP1 ' i çalıştıran kişinin kullanıcı kimliğinin TS101 olduğu varsayılmıştır.

```
RDEFINE MQCMDS CSQ1.DISPLAY.PROCESS
PERMIT CSQ1.DISPLAY.PROCESS CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.REPLY.MODEL
PERMIT CSQ1.SYSTEM.COMMAND.REPLY.MODEL CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.CSQ4IVP1.**
PERMIT CSQ1.CSQ4IVP1.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)
```

*Şekil 101. CSQ4IVP1 için RACF komutları*

## CSQ4IVP1 ' nin çalıştırılması

Bu adımları tamamladığınızda, kuyruk yöneticinizi başlatın. Kuyruk yöneticisi zaten çalışıyorsa ve CSQINP2değiştirdiyse, kuyruk yöneticisini durdurup yeniden başlatmanız gerekir.

IVP, toplu iş olarak çalışır. İş kartını, kuruluşunuzun gönderim gereksinimlerini karşılayacak şekilde özelleştirin.

## CSQ4IVP1 sonuçlarının denetlenmesi

IVP 10 aşamaya bölünür; sonraki aşama çalıştırılmadan önce her aşama sıfır tamamlanma koduyla tamamlanmalıdır. IVP bir rapor oluşturur ve şunları listeler:

- Bağlanılmakta olan kuyruk yöneticisinin adı.
- Tamamlanma kodunu ve her aşamadan döndürülen neden kodunu gösteren tek satırlık bir ileti.
- Uygun olduğunda, tek satırlık bir bilgi iletisi.



Şekil 102 sayfa 950 içinde örnek bir rapor sağlanmıştır

**z/OS** Tamamlanma ve neden kodlarına ilişkin açıklamalar için IBM MQ for z/OS iletilerine, tamamlanma ve neden kodlarına bakın.

Bazı aşamalarda birden çok IBM MQ çağrısı vardır ve hata oluşması durumunda, hatayı döndürebilecek IBM MQ çağrısını belirten bir ileti yayınlanır. Ayrıca, bazı aşamalar için IVP açıklayıcı ve tanılama bilgilerini bir yorum alanına koyar.

IVP işi, belirli kuyruk yöneticisi nesnelerinin dışlayıcı denetimini istediğinden, sistem üzerinden tek iş parçacıklı olmalıdır. Ancak, IVP 'nin kuyruk yöneticiniz için kaç kez çalıştırılabileceğine ilişkin bir sınır yoktur.

Her aşama tarafından gerçekleştirilen işlevler şunlardır:

#### **Aşama 1**

MQCONN API çağrısını yayınlayarak kuyruk yöneticisine bağlanın.

#### **Aşama 2**

İstek iletilerini almak için komut sunucusu tarafından kullanılan sistem komutu giriş kuyruğunun adını belirleyin. Bu kuyruk, 5. Aşama 'dan görüntü istekleri alır.

Bunu yapmak için, çağrılarının sırası şöyledir:

1. Kuyruk yöneticisi nesnesini açmak için, kuyruk yöneticisi adını belirterek bir MQOPEN çağrısı yayınlayın.
2. Sistem komutu giriş kuyruğunun adını öğrenmek için bir MQINQ çağrısı yayınlayın.
3. Çeşitli kuyruk yöneticisi olay anahtarlarına ilişkin bilgi için bir MQINQ çağrısı yayınlayın.
4. Kuyruk yöneticisi nesnesini kapatmak için bir MQCLOSE çağrısı yayınlayın.

Bu aşama başarıyla tamamlandığında, açıklama alanında sistem komutu giriş kuyruğunun adı görüntülenir.

#### **Aşama 3**

**MQOPEN** çağrısıyla bir başlatma kuyruğu açın.

Bu kuyruk, 5. Aşama 'dan gelen isteğe yanıt veren komut sunucusunun sonucu olarak gelen bir tetikleyici ileti beklenirken bu aşamada açılır. Girişin tetikleme ölçütlerini karşılaması için kuyruk açılmalıdır.

#### **Aşama 4**

CSQ4IVP1.MODEL kuyruğu. Dinamik kuyruk, yaratıldığı modelle aynı özniteliklere sahiptir. Bu, 5. Aşama 'daki komut sunucusu isteğinden gelen yanıtlar bu kuyruğa yazıldığında, 3. Aşama 'da açılan başlatma kuyruğuna bir tetikleyici iletisinin yazılacağı anlamına gelir.

Bu aşama başarıyla tamamlandığında, kalıcı dinamik kuyruğun adı açıklama alanında gösterilir.

#### **Aşama 5**

Komut sunucusu komut kuyruğuna bir MQPUT1 isteği gönderin.

Sistem komut giriş kuyruğuna, CSQ4IVP1 işleminin görüntülenmesini isteyen MQMT\_REQUEST tipinde bir ileti yazılır. İletiyeye ilişkin ileti tanımlayıcı, komut sunucusunun yanıtı için yanıt kuyruğu olarak 4. Aşama 'da yaratılan kalıcı dinamik kuyruğu belirtir.

#### **Aşama 6**

Başlatma kuyruğundan bir **MQGET** isteği yayınlayın. Bu aşamada, 3. Aşama 'da açılan başlatma kuyruğuna karşı 1 dakikalık bir aralığa sahip GET WAIT verilir. Döndürülen iletinin, yanıt kuyruğuna yazılmakta olan komut sunucusunun yanıt iletileri tarafından oluşturulan tetikleyici ileti olması beklenir.

#### **Aşama 7**

Aşama 4 'te yaratılan kalıcı dinamik kuyruğu silin. Kuyrukta hala ileti olduğundan, MQCO\_PURGE\_DELETE seçeneği kullanılır.

## Aşama 8

1. Dinamik bir kuyruk açın.
2. Süre bitimi aralığı ayarlanmış bir MQPUT iletisi.
3. İletinin süresinin dolmasını bekleyin.
4. Süresi dolan iletii MQGET işlemi denendi.
5. Kuyruk MQCLOSE.

## Aşama 9

1. Dinamik bir kuyruk açın.
2. MQPUT bir ileti.
3. Yürürlükteki iş birimini kesinleştirmek için MQCMIT komutunu verin.
4. İletiyi MQGET.
5. İletiyi geriletmek için MQBACK komutunu verin.
6. MQGET aynı ileti ve geriletme sayısının 1 olarak ayarlandığından emin olun.
7. Kuyruğu kapatmak için MQCLOSE komutunu verin.

## Aşama 10

**MQDISC**komutunu kullanarak kuyruk yöneticisiyle bağlantıyı kesin.

IVP ' yi çalıştırdıktan sonra, artık gerek duymadığınız nesnelere silebilirsiniz.

IVP başarılı bir şekilde çalışmazsa, hangi işlevin başarısız olduğunu öğrenmek için her adımı el ile deneyin.

```
DATE : 2005.035          IBM MQ for z/OS - V6          PAGE : 0001
INSTALLATION VERIFICATION PROGRAM
PARAMETERS ACCEPTED. PROGRAM WILL CONNECT TO : CSQ1
,OBJECT QUALIFER : CSQ4IVP1
INSTALLATION VERIFICATION BEGINS :
STAGE 01 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR BRIDGE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS EXCP FOR CHANNEL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR SSL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR INHIBITED EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR LOCAL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR PERFORMANCE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR REMOTE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR START/STOP EVENTS
STAGE 02 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 SYSTEM.COMMAND.INPUT
STAGE 03 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 04 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810EFEAC8980
STAGE 05 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 06 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 07 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 08 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810F0070E645
STAGE 09 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9812BA8706803
STAGE 10 COMPLETE. COMPCODE : 0000 REASON CODE : 0000>>>>>>>>>> END OF REPORT <<<<<<<<<<<<
```

Şekil 102. CSQ4IVP1 ' den örnek rapor

## z/OS Kuyruk paylaşım grupları için test ediliyor

Temel kuruluş doğrulama programı CSQ4IVP1 , paylaşılmayan kuyrukları sınar.

CSQ4IVP1 , kuyruk yöneticisinin bir kuyruk paylaşım grubunun üyesi olup olmaması dikkate alınmaksızın kullanılabilir. Temel IVP ' yi çalıştırdıktan sonra, CSQ4IVP1 kuruluş doğrulama programını farklı kuyruklarla kullanarak paylaşılan kuyrukları sınavabilirsiniz. Bu ayrıca, Db2 ve bağlaşım olanağının doğru kurulduğunu da sınar.

## Bir kuyruk paylaşım grubu için CSQ4IVP1 komutunu çalıştırma hazırlığı

CSQ4IVP1:

1. IVP ' nin kullandığı bağlaşım olanağı yapısını “Bağlaşım olanağını ayarla” sayfa 933başlıklı konuda açıklandığı şekilde CFRM ilke veri kümenize ekleyin. Sağlanan örnekler APPLICATION1adlı bir yapı kullanıyor, ancak isterseniz bunu değiştirebilirsiniz.
2. IVP girişlerinin kuyruk yöneticisi başlatma programındaki CSQINP2 veri kümesi birleşiminde olup olmadığını denetleyin. IVP girişleri thlqual.SCSQPROC(CSQ4IVPG) üyesinde sağlanır. Değilse, thlqual.SCSQPROC(CSQ4IVPG) içinde sağlanan tanımlamaları CSQINP2 birleşiminize ekleyin. Kuyruk yöneticisi çalışıyorsa, bu tanımlamaların yürürlüğe girmesi için yeniden başlatmanız gerekir.
3. Gerekliyse, thlqual.SCSQPROC(CSQ4IVPG) içinde kullanılan bağlaşım olanağı yapısının adını değiştirin.
4. Bir kuyruk paylaşım grubuna ilişkin kuruluş doğrulama programını çalıştırmak için gereken örnek JCL ( CSQ4IVPS) thlqual.SCSQPROCKitaplığında yer alır.

CSQ4IVPS JCL ' yi IBM MQ kitaplıkları için üst düzey niteleyici, kullanmak istediğiniz ulusal dil, dört karakterli IBM MQ kuyruk yöneticisi adı ve iş çıkışına ilişkin hedef ile uyarlayın.

5. RACF dosyasını, IBM MQ güvenliği etkinse CSQ4IVP1 ' in kaynaklarına erişmesine izin verecek şekilde güncelleyin.

IBM MQ güvenliği etkinleştirildiğinde CSQ4IVP1 komutunu çalıştırmak için, nesnelere erişim yetkisi olan bir RACF kullanıcı kimliği gerekir. RACF'e kaynak tanımlanmasına ilişkin ayrıntılar için [z/OS ' da güvenliği ayarlama](#) başlıklı konuya bakın. Temel IVP 'yi çalıştırmak için gereken erişim yetkisine ek olarak, IVP' yi çalıştıran kullanıcı kimliğinin de aşağıdaki erişim yetkisi olmalıdır:

Authority	Profil	Sınıf
GÜNCELLE	ssid.CSQ4IVPG.**	MQQUEUE

Bu gereksinimler, tüm IBM MQ güvenliğinin etkin olduğunu varsayar. IBM MQ güvenliğini etkinleştirmek için RACF komutları [Şekil 103 sayfa 951](#) içinde gösterilir. Bu örnekte, kuyruk yöneticisi adının CSQ1 olduğu ve örnek CSQ4IVP1 ' i çalıştıran kişinin kullanıcı kimliğinin TS101 olduğu varsayılmıştır.

```
RDEFINE MQQUEUE CSQ1.CSQ4IVPG.**
PERMIT CSQ1.CSQ4IVPG.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)
```

*Şekil 103. Bir kuyruk paylaşım grubuna ilişkin CSQ4IVP1 için RACF komutları*

## Kuyruk paylaşım grubu için CSQ4IVP1 ' in çalıştırılması

Bu adımları tamamladığınızda, kuyruk yöneticisini başlatın. Kuyruk yöneticisi zaten çalışıyorsa ve CSQINP2değiştirdiyse, kuyruk yöneticisini durdurup yeniden başlatmanız gerekir.

IVP, toplu iş olarak çalışır. İş kartını, kuruluşunuzun gönderim gereksinimlerini karşılayacak şekilde özelleştirin.

## Kuyruk paylaşım grubu için CSQ4IVP1 sonuçlarının denetlenmesi

Kuyruk paylaşım gruplarına ilişkin IVP, yaratılan kuyruklara CSQIVPG adı verilmesi dışında, temel IVP ile aynı şekilde çalışır. xx. IVP ' nin sonuçlarını kuyruk paylaşım grupları için denetlemek üzere “[CSQ4IVP1 sonuçlarının denetlenmesi](#)” sayfa 948 içinde verilen yönergeleri izleyin.

## ► z/OS Dağıtılmış kuyruğa alma testi

Sağlanan kuruluş doğrulama programını ( CSQ4IVPX), dağıtılmış kuyruğa alma işleminin çalıştığını doğrulamak için kullanabilirsiniz.

## CSQ4IVPX işine genel bakış

CSQ4IVPX , kanal başlatıcısını başlatan ve IBM MQ DISPLAY CHINIT komutunu veren bir toplu işidir. Bu, kanal ve ağ tanımlarının ayarlanmasına gerek kalmayarak, dağıtılmış kuyruğa almanın tüm önemli yönlerinin çalışır durumda olduğunu doğrular.

## CSQ4IVPX ' i çalıştırma hazırlığı yapılıyor

CSQ4IVPX: komutunu çalıştırmadan önce:

1. Kuruluş doğrulama programını çalıştırmak için gereken örnek JCL ( CSQ4IVPX) thlqual.SCSQPROCKitaplığında bulunur.

CSQ4IVPX JCL ' yi IBM MQ kitaplıkları için üst düzey niteleyici, kullanmak istediğiniz ulusal dil, dört karakterli kuyruk yöneticisi adı ve iş çıkışına ilişkin hedef ile uyarlayın.

2. RACF dosyasını, IBM MQ güvenliği etkinse CSQ4IVPX ' in kaynaklarına erişmesine izin verecek şekilde güncelleyin. IBM MQ güvenliği etkinleştirildiğinde CSQ4IVPX komutunu çalıştırmak için, nesnelere erişim yetkisi olan bir RACF kullanıcı kimliği gerekir. RACF'e kaynak tanımlanmasına ilişkin ayrıntılar için z/OS ' da güvenliği ayarlama başlıklı konuya bakın. IVP ' yi çalıştıran kullanıcı kimliğinin aşağıdaki erişim yetkisi olmalıdır:

Authority	Profil	Sınıf
CONTROL	ssid.START.CHINIT ve ssid.STOP.CHINIT	MQCMDS
GÜNCELLE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
GÜNCELLE	ssid.SYSTEM.CSQUTIL.*	MQQUEUE
READ	ssid.BATCH	MQCONN
READ	ssid.DISPLAY.CHINIT	MQCMDS

Bu gereksinimler, ssid.CHIN bağlantı güvenliği tanımının tanımlandığını ( Kanal başlatıcısı için bağlantı güvenliği tanımları içinde gösterildiği gibi) ve tüm IBM MQ güvenliğinin etkin olduğunu varsayar. Bunu yapmak için RACF komutları Şekil 104 sayfa 953 içinde gösterilir. Bu örnek şunları varsayar:

- Kuyruk yöneticisi adı: CSQ1
- Örnek CSQ4IVPX çalıştıran kişinin kullanıcı kimliği: TS101
- Kanal başlatıcı adres alanı CSQ1MSTR kullanıcı kimliği altında çalışıyor

3. Kanal başlatıcı adres alanını aşağıdaki erişim yetkisine izin verecek şekilde RACF adresini güncelleyin:

Authority	Profil	Sınıf
READ	ssid.CHIN	MQCONN
GÜNCELLE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
GÜNCELLE	ssid.SYSTEM.CHANNEL.INITQ	MQQUEUE
GÜNCELLE	ssid.SYSTEM.CHANNEL.SYNCQ	MQQUEUE
Çeviri	ssid.SYSTEM.CLUSTER.COMMAND.QUEUE	MQQUEUE
GÜNCELLE	ssid.SYSTEM.CLUSTER.TRANSMIT.QUEUE	MQQUEUE
Çeviri	ssid.SYSTEM.CLUSTER.REPOSITORY.QUEUE	MQQUEUE
CONTROL	ssid.CONTEXT.**	MQADMIN

Bunu yapmak için RACF komutları da Şekil 104 sayfa 953 içinde gösterilir.

```

RDEFINE MQCMDS CSQ1.DISPLAY.DQM
PERMIT CSQ1.DISPLAY.DQM CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQCMDS CSQ1.START.CHINIT
PERMIT CSQ1.START.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQCMDS CSQ1.STOP.CHINIT
PERMIT CSQ1.STOP.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101,CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CSQUTIL.*
PERMIT CSQ1.SYSTEM.CSQUTIL.* CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)

RDEFINE MQCONN CSQ1.CHIN
PERMIT CSQ1.CHIN CLASS(MQCONN) ID(CSQ1MSTR) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.SYNCQ
PERMIT CSQ1.SYSTEM.CHANNEL.SYNCQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.INITQ
PERMIT CSQ1.SYSTEM.CHANNEL.INITQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQADMIN CSQ1.CONTEXT.**
PERMIT CSQ1.CONTEXT.** CLASS(MQADMIN) ID(CSQ1MSTR) ACCESS(CONTROL)

```

Şekil 104. CSQ4IVPX için RACF komutları

## CSQ4IVPX ' i çalıştırma

Bu adımları tamamladığınızda, kuyruk yöneticinizi başlatın.

IVP, toplu iş olarak çalışır. İş kartını, kuruluşunuzun gönderim gereksinimlerini karşılayacak şekilde özelleştirin.

## CSQ4IVPX sonuçlarının denetlenmesi

CSQ4IVPX , üç MQSC komutu vermek için CSQUTIL IBM MQ yardımcı programını çalıştırır. SYSPRINT çıkış veri kümesi Şekil 105 sayfa 954 gibi görünmelidir; ancak, ayrıntılar kuyruk yöneticisi özniteliklerinizle bağlı olarak farklılık gösterebilir.

- Her biri birden çok iletinin izlediği **(1)** komutlarını görmeniz gerekir.
- Her komutun son iletisi "CSQ9022I olmalıdır ... NORMAL TAMAMLANMA" **(2)**.
- Bir bütün olarak iş, dönüş kodu sıfır **(3)** ile tamamlanmalıdır.

```

CSQU000I CSQUTIL IBM MQ for z/OS - V6
CSQU001I CSQUTIL Queue Manager Utility - 2005-05-09 09:06:48
COMMAND
CSQU127I CSQUTIL Executing COMMAND using input from CSQUCMD data set
CSQU120I CSQUTIL Connecting to queue manager CSQ1
CSQU121I CSQUTIL Connected to queue manager CSQ1
CSQU055I CSQUTIL Target queue manager is CSQ1
START CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM138I +CSQ1 CSQMSCHI CHANNEL INITIATOR STARTING
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' START CHINIT' NORMAL COMPLETION
(2)
DISPLAY CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMDDQM DISPLAY CHINIT COMMAND ACCEPTED
CSQN205I COUNT= 12, RETURN=00000000, REASON=00000000
CSQX830I +CSQ1 CSQXRDM Channel initiator active
CSQX002I +CSQ1 CSQXRDM Queue sharing group is QSG1
CSQX831I +CSQ1 CSQXRDM 8 adapter subtasks started, 8 requested
CSQX832I +CSQ1 CSQXRDM 5 dispatchers started, 5 requested
CSQX833I +CSQ1 CSQXRDM 0 SSL server subtasks started, 0 requested
CSQX840I +CSQ1 CSQXRDM 0 channel connections current, maximum 200
CSQX841I +CSQ1 CSQXRDM 0 channel connections active, maximum 200,
including 0 paused
CSQX842I +CSQ1 CSQXRDM 0 channel connections starting,
0 stopped, 0 retrying
CSQX836I +CSQ1 Maximum channels - TCP/IP 200, LU 6.2 200
CSQX845I +CSQ1 CSQXRDM TCP/IP system name is TCP/IP
CSQX848I +CSQ1 CSQXRDM TCP/IP listener INDISP=QMGR not started
CSQX848I +CSQ1 CSQXRDM TCP/IP listener INDISP=GROUP not started
CSQX849I +CSQ1 CSQXRDM LU 6.2 listener INDISP=QMGR not started
CSQX849I +CSQ1 CSQXRDM LU 6.2 listener INDISP=GROUP not started
CSQ9022I +CSQ1 CSQXCRPS ' DISPLAY CHINIT' NORMAL COMPLETION
(2)
STOP CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMTCHI STOP CHINIT COMMAND ACCEPTED
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' STOP CHINIT' NORMAL COMPLETION
(2)
CSQU057I CSQUCMDS 3 commands read
CSQU058I CSQUCMDS 3 commands issued and responses received, 0 failed
CSQU143I CSQUTIL 1 COMMAND statements attempted
CSQU144I CSQUTIL 1 COMMAND statements executed successfully
CSQU148I CSQUTIL Utility completed, return code=0
(3)

```

Şekil 105. CSQ4IVPX örnek çıktısı

## z/OS IBM MQ for z/OS ile C, C + +, COBOL, PL/I ve CICS programları için test

IBM MQ ile verilen örnek uygulamaları kullanarak C, C + +, COBOL, PL/I ya da CICS için test yapabilirsiniz.

IVP (CSQ4IVP1) yükleme modülü olarak sağlanır ve örnekleri kaynak modül olarak sağlar. Farklı programlama dili ortamlarını sınamak için bu kaynak modülleri kullanabilirsiniz.

Örnek uygulamalarla ilgili daha fazla bilgi için bkz. [IBM MQ for z/OS için örnek programlar](#).

## z/OS z/OS üzerinde diğer kuyruk yöneticileriyle iletişim kurulması

Bu bölümde, dağıtılmış kuyruğa alma işlevini kullanmaya başlamadan önce yapmanız gereken IBM MQ for z/OS hazırlıkları açıklanmaktadır.

## Bu görev hakkında

Dağıtılmış kuyruğa alma gereksinimlerinizi tanımlamak için aşağıdaki öğeleri tanımlamanız gerekir:

- Kanal başlatıcı yordamları ve veri kümeleri
- Kanal tanımları
- Kuyruklar ve diğer nesnelere
- Erişim güvenliği

Kuyruk paylaşım gruplarını kullanıyorsanız, [Dağıtılmış kuyruğa alma ve kuyruk paylaşım grupları](#) başlıklı konuya bakın.

IBM MQ for z/OS ile dağıtılmış kuyruğa alma işlemini ayarlamaya hazırlanırken göz önünde bulundurmanız gereken ek noktalar için bkz. [“z/OS üzerinde dağıtılmış kuyruğa alma özelliğinin kullanılmasıyla ilgili dikkat edilecek noktalar”](#) sayfa 955.

## Yordam

Dağıtılmış kuyruğa alma özelliğini etkinleştirmek için aşağıdaki adımları tamamlayın:

- Dağıtılmış kuyruğa alma olanağını uyarlayın ve Sistem nesnelere tanımlama ve [“z/OS üzerinde kuyruk yöneticilerini özelleştirme hazırlığı”](#) sayfa 877 konusunda açıklandığı gibi gereken IBM MQ nesnelere tanımlayın.
- z/OS üzerinde kanal başlatıcısına ilişkin güvenlikle ilgili önemli noktalar içinde açıklandığı gibi erişim güvenliğini tanımlayın.
- [“z/OS için iletişimi ayarlama”](#) sayfa 974 başlıklı konuda açıklandığı gibi iletişimlerinizi ayarlayın.

## İlgili kavramlar

[“ayarlar IBM MQ for z/OS”](#) sayfa 881

IBM MQ for z/OS sisteminizi uyarlamak için adım adım bu konuyu kullanın.

## İlgili görevler

[“Dağıtılmış kuyruğa alma yapılandırılıyor”](#) sayfa 189

Bu bölümde, kuyruk tanımlaması, kanal tanımlaması, tetikleme ve eşitleme noktaları yordamları da içinde olmak üzere IBM MQ kuruluşları arasındaki iletişimle ilgili daha ayrıntılı bilgi sağlanır.

## z/OS üzerinde dağıtılmış kuyruğa alma özelliğinin kullanılmasıyla ilgili dikkat edilecek noktalar

z/OS üzerinde dağıtılmış kuyruğa alma işlevini kullanmaya hazırlanırken göz önünde bulundurulması gereken noktalar.

Kuyruk paylaşım gruplarını kullanıyorsanız, [Dağıtılmış kuyruğa alma ve kuyruk paylaşım grupları](#) başlıklı konuya bakın.

## İşletmen iletileri

Kanal başlatıcısı zaman uyumsuz olarak çalışan dağıtıcılar kullandığından, kronolojik sırada oturum kapatılırken işletmen iletileri oluşabilir.

## Kanal işlemi komutları

Kanal operasyon komutları genellikle iki aşamadan oluşur. Komut sözdizimi denetlendiğinde ve kanalın varlığı doğrulandığında, kanal başlatıcısına bir istek gönderilir. Birinci aşamanın tamamlandığını belirtmek için komut yayıncısına [CSQM134I](#) ya da [CSQM137I](#) iletileri gönderilir. Kanal başlatıcısı komutu işlediğinde, komutun başarılı olduğunu ya da başka bir şekilde komut veren kuruluşa [CSQ9022I](#) ya da [CSQ9023E](#) iletileriyle birlikte başka iletiler de gönderilir. Oluşturulan hata iletileri z/OS konsoluna da gönderilebilir.

Ancak, **DISPLAY CLUSQMR** dışındaki tüm küme komutları zamanuyumsuz olarak çalışır. Nesne özniteliklerini değiştiren komutlar nesneyi günceller ve kanal başlatıcısına bir istek gönderir. Kümelerle çalışmaya ilişkin komutların sözdizimi denetlenir ve kanal başlatıcısına bir istek gönderilir. Her iki durumda da, komut veren tarafa bir isteğin gönderildiğini belirten **CSQM130I** iletisi gönderilir. Bu iletiyi, bir isteğin gönderilmesiyle ilgili olarak komutun başarıyla tamamlandığını belirtmek için **CSQ9022I** iletisi görüntülenir. Bu, küme isteğinin başarıyla tamamlandığını göstermez. Kanal başlatıcısına gönderilen istekler, kümenin diğer üyelerinden alınan küme istekleriyle birlikte zamanuyumsuz olarak işlenir. Bazı durumlarda, bu isteklerin başarılı olup olmadıklarını belirlemek için tüm kümeye gönderilmesi gerekir. Hatalar, kanal başlatıcının çalıştığı sistemdeki z/OS ' e raporlanır. Bunlar, komutu veren kuruluşa gönderilmez.

## Teslim edilmedi-ileti kuyruğu

IBM MQ for z/OS ile birlikte bir Gitmeyen mektup işleyicisi sağlanır. Daha fazla bilgi için bakınız: [The dead-letter queue handler utility \(CSQUDLQH\)](#).

## Kullanılan kuyruklar

Alıcı kanallarına ilişkin MCA ' lar, iletiler iletilmediğinde bile hedef kuyrukları açık tutabilir. Bu davranış, kuyrukların 'kullanımda' olarak görünmesine neden olur.

## Güvenlik değişiklikleri

Bir kullanıcı kimliği için güvenlik erişimini değiştirirseniz, değişiklik hemen yürürlüğe girmeyebilir. Daha fazla bilgi için bkz. [z/OS üzerinde kanal başlatıcısına ilişkin güvenlikle ilgili önemli noktalar](#) , [Kuyruk güvenliğine ilişkin profillerle "ESM güvenlik denetimlerinizi uygulayın" sayfa 895](#).

## İletişim durduruldu-TCP

TCP bir nedenle durdurulup yeniden başlatılırsa, TCP kapısında bekleyen IBM MQ for z/OS TCP dinleyicisi durdurulur.

Otomatik kanal yeniden bağlanma, kanal başlatıcısının TCP/IP 'nin kullanılmadığını saptamasını ve TCP/IP' nin geri dönmesi durumunda TCP/IP dinleyicisini otomatik olarak yeniden başlatmasını sağlar. Bu otomatik yeniden başlatma işlemi, işlem personelinin TCP/IP ile ilgili sorunu fark etmesi ve dinleyiciyi el ile yeniden başlatması gereksinimini ortadan kaldırmaktadır. Dinleyici işlem dışı durumdayken, kanal başlatıcısı, LSTRTMR tarafından belirlenen aralıkta dinleyiciyi yeniden denemek için de kullanılabilir. Bu girişimler, TCP/IP dönüncüye ve dinleyici otomatik olarak yeniden başlatılıncaya kadar devam edebilir. LSTRTMR ile ilgili ek bilgi için bkz. [ALTER QMGR](#) ve [Distributed queuing messages \(CSQX ...\)](#).

## İletişim durduruldu- LU6.2

APPC durdurulursa, dinleyici de durdurulur. Bu durumda, dinleyici LSTRTMR aralığında otomatik olarak yeniden dener; böylece, APPC yeniden başlatılırsa, dinleyici de yeniden başlatılabilir.

Db2 başarısız olursa, çalışmakta olan paylaşılan kanallar çalışmaya devam eder, ancak yeni kanal başlatma istekleri başarısız olur. Db2 geri yüklendiğinde yeni istekler tamamlanabilir.

## z/OS Otomatik Yeniden Başlatma Yönetimi (ARM)

Otomatik yeniden başlatma yönetimi (ARM), belirli toplu işlerin ya da başlatılan görevlerin (örneğin, altsistemlerin) kullanılabilirliğini artırabilen bir z/OS kurtarma işlevidir. Bu nedenle, üretken çalışmanın daha hızlı bir şekilde sürdürülmesine neden olabilir.

ARM ' yi kullanmak için kuyruk yöneticilerinizi ve kanal başlatıcılarınızı otomatik olarak yeniden başlatmaları için belirli bir şekilde ayarlamanız gerekir. Daha fazla bilgi için bkz. [z/OS Automatic Restart Manager \(ARM\) olanağının kullanılması](#).

## İlgili kavramlar

["ayarlama IBM MQ for z/OS" sayfa 881](#)



IBM MQ for z/OS sisteminizi uyarlamak için adım adım bu konuyu kullanın.

## İlgili görevler

“Dağıtılmış kuyruğa alma yapılandırılıyor” sayfa 189

Bu bölümde, kuyruk tanımlaması, kanal tanımlaması, tetikleme ve eşitleme noktası yordamları da içinde olmak üzere IBM MQ kuruluşları arasındaki iletişimle ilgili daha ayrıntılı bilgi sağlanır.

## z/OS z/OS üzerinde IBM MQ nesnelerini tanımlama

z/OS sistemlerinde, IBM MQ nesnelerini tanımlamak için IBM MQ komut giriş yöntemlerinden birini kullanın.

Nesneleri tanımlama hakkında daha fazla bilgi için bkz. “z/OS üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 958.

## İletim kuyrukları ve tetikleme kanalları

Aşağıdakileri tanımlayın:

- Her bir gönderen ileti kanalı için XMITQ kullanımını içeren yerel bir kuyruk.
- Uzak kuyruk tanımlamaları.

Uzak kuyruk nesnesinin, adın ve içeriğin belirlenmiş şekline bağlı olarak üç ayrı kullanımı vardır:

- Uzak kuyruk tanımlaması
- Kuyruk yöneticisi diğer adı tanımlaması
- Yanıtın gönderileceği kuyruk diğer adı tanımlaması

Bu üç yöntem Uzak kuyruk tanımlaması nesnesini kullanmanın üç yolu olarak gösterilir.

Belirlenen kanalı tetiklemek için iletim kuyruğundaki TRIGDATA alanını kullanın. Örneğin:

```
DEFINE QLOCAL(MYXMITQ) USAGE(XMITQ) TRIGGER +  
INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(MYCHANNEL)  
DEFINE CHL(MYCHANNEL) CHLTYPE(SDR) TRPTYPE(TCP) +  
XMITQ(MYXMITQ) CONNAME('9.20.9.30(1555)')
```

Sağlanan örnek CSQ4INYD , gerekli tanımlamalara ilişkin ek örnekler sağlar.

Paylaşılan kanallar için eşitleme kuyruğunun tanımlandığı CF yapısıyla bağlantı kaybı ya da benzer sorunlar, bir kanalın başlatılmasını geçici olarak engelleyebilir. Sorun çözüldükten sonra, FIRST tetikleyici tipi kullanıyorsanız ve kanal tetiklendiğinde başlatılamıyorsa, kanalı el ile başlatmanız gerekir. Sorun çözüldükten sonra tetiklenen kanalları otomatik olarak başlatmak istiyorsanız, kuyruk yöneticisi TRIGINT özniteliğini varsayılan dışında bir değere ayarlayın. TRIGINT özniteliğinin varsayılan değerden farklı bir değere ayarlanması, iletim kuyruğunda ileti varken kanal başlatıcısının belirli aralıklarla kanalı başlatmayı yeniden denemesine neden olur.

## Eşitleme kuyruğu

DQM, sıra numaraları ve mantıksal iş tanıtıcısı birimleriyle (LUWID) kullanılmak üzere bir kuyruk gerektirir. SYSTEM.CHANNEL.SYNCQ (bkz. [z/OS üzerinde planlama](#) ). Bu kuyruk kullanılabilir olmalıdır, aksi halde kanal başlatıcı başlatılamıyor.

Bu kuyruğu INDXTYPE (MSGID) komutunu kullanarak tanımladığınızdan emin olun. Bu öznitelik, bunlara erişilme hızını artırır.

## Kanal komut kuyrukları

Sisteminiz için SYSTEM.CHANNEL.INITQ.

Kanal başlatıcı SYSTEM.CHANNEL.INITQ, sorun düzeltilinceye kadar olağan şekilde devam edemiyor. Sorun aşağıdakilerden biri olabilir:

- Kuyruk dolu
- Kuyruk koyma için etkinleştirilmedi
- Kuyruğun bulunduğu sayfa kümesi dolu
- Kanal başlatıcısı, kuyruk için doğru güvenlik yetkisine sahip değil

Kanal başlatıcısı çalışırken kuyruğun tanımı GET (DISABLED) olarak değiştirilirse, başlatıcı kuyruktan ileti alamaz ve sonlandırılır.

## Kanal başlatıcı başlatılıyor

Tetikleme, kanal başlatıcısı kullanılarak gerçekleştirilir. IBM MQ for z/OS' da başlatıcı, START CHINITMQSC komutuyla başlatılır.

## Kanal başlatıcısı durdurma

Kuyruk yöneticisini durdurduğunuzda kanal başlatıcı otomatik olarak durdurulur. Kuyruk yöneticisini değil, kanal başlatıcısı durdurmanız gerekiyorsa, STOP CHINITMQSC komutunu kullanabilirsiniz.

## z/OS üzerinde kanalların izlenmesi ve denetlenmesi

Uzak kuyruk yöneticilerine kanallar yaratmak, izlemek ve denetlemek için DQM komutlarını ve panolarını kullanın.

Her z/OS kuyruk yöneticisinin bir DQM programı vardır ( *kanal başlatıcısı* ) yerel z/OS olanaklarını kullanarak uzak kuyruk yöneticilerine yönelik ara bağlantıları denetlemek için.

Bu panoların ve komutların z/OS üzerindeki somutlaması, işlemler ve denetim panolarıyla ve MQSC komutlarıyla tümleştirilmiştir. Bu iki pano ve komutun düzenlenmesinde herhangi bir ayırım yapılmaz.

Komutları, Programlanabilir Komut Biçimi (PCF) komutlarını kullanarak da girebilirsiniz. Bu komutların kullanılmasıyla ilgili bilgi için [Yönetim görevlerinin otomatikleştirilmesi](#) başlıklı konuya bakın.

Bu bölümdeki bilgiler, kanal başlatıcısının dağıtılmış kuyruğa alma için kullanıldığı tüm durumlarda geçerlidir. Bu, kuyruk paylaşım gruplarını mı, yoksa grup içi kuyruğa alma mı kullandığınızı belirler.

## DQM kanal denetimi işlevi

Dağıtılmış kuyruk yönetimi modeline genel bakış için bkz. "[İleti gönderme ve alma](#)" sayfa 210.

Kanal denetim işlevi, panolar, komutlar ve programlar, iki eşzamanlama kuyruğu, kanal komut kuyrukları ve kanal tanımlarından oluşur. Bu konu, kanal denetimi işlevinin bileşenlerinin kısa bir açıklamasıdır.

- Kanal tanımlamaları, z/OSiçindeki diğer IBM MQ nesnelere gibi, sayfa kümesi sıfır ya da Db2içinde nesnelere tutulur.
- Aşağıdaki işlemleri ve denetim panolarını, MQSC komutlarını ya da PCF komutlarını kullanabilirsiniz:
  - Kanal tanımlarının yaratılması, kopyalanması, görüntülenmesi, değiştirilmesi ve silinmesi
  - Kanal başlatıcıların ve dinleyicilerin başlatılması ve durdurulması
  - Başlatma, durdurma ve ping kanalları, kanal sıra numaralarını sıfırlama ve bağlantılar yeniden oluşturulmadığı zaman belirsiz iletileri çözme
  - Kanallara ilişkin durum bilgilerini görüntüle
  - DQM ile ilgili bilgileri görüntüle

Özellikle, MQSC komutlarınızı yayınlamak için CSQINPX kullanıma hazırlama giriş verileri kümesini kullanabilirsiniz. Bu küme, kanal başlatıcısını her başlattığınızda işlenebilir. Daha fazla bilgi için bkz. [Başlatma komutları](#).

- İki kuyruk vardır (SYSTEM.CHANNEL.SYNCC ve SYSTEM.QSG.CHANNEL.SYNCC) kanal yeniden eşitleme amacıyla kullanılır. Başarımla ilgili nedenlerden ötürü bu kuyrukları INDXTYPE (MSGID) ile tanımlayın.

- Kanal komut kuyruğu (SYSTEM.CHANNEL.INITQ), kanal başlatıcılarına, kanallara ve dinleyicilere ilişkin komutları tutmak için kullanılır.
- Kanal denetim işlevi programı, kuyruk yöneticisinden ayrı olarak kendi adres alanında çalışır ve kanal başlatıcısı, dinleyiciler, MCA 'lar, tetikleyici izleyicisi ve komut işleyicisinden oluşur.
- Kuyruk paylaşım grupları ve paylaşılan kanallar için [Paylaşılan kuyruklar ve kuyruk paylaşım grupları](#) başlıklı konuya bakın.
- Grup içi kuyruğa alma için bkz. [Grup içi kuyruğa alma](#)

## z/OS üzerinde kanallarınızın yönetilmesi

Kanallarınızı, kanal başlatıcılarınızı ve dinleyicilerinizi nasıl yöneteceğinizi öğrenmek için aşağıdaki çizelgedeki bağlantıları kullanın:

<i>Çizelge 64. Kanal görevleri</i>	
<b>Gerçekleştirilecek görev</b>	<b>MQSC komutu</b>
<a href="#">Kanal tanımlanması</a>	<a href="#">KANALı TANıMLAYIN</a>
<a href="#">Kanal tanımını değiştir</a>	<a href="#">KANAL DEĞİŞTİN</a>
<a href="#">Kanal Tanımının Görüntülenmesi</a>	<a href="#">KANALı GÖRÜNTÜLE</a>
<a href="#">Kanal tanımının silinmesi</a>	<a href="#">KANALı SIL</a>
<a href="#">Kanal başlatıcı başlatılması</a>	<a href="#">CHINIT ' I BAŞLAYIN</a>
<a href="#">Kanal Başlatıcının Durdurulması</a>	<a href="#">CHINIT ' I DURDUR</a>
<a href="#">Kanal Başlatıcı Bilgilerinin Görüntülenmesi</a>	<a href="#">CHINIT ' I GÖRÜNTÜLE</a>
<a href="#">Kanal dinleyicisinin başlatılması</a>	<a href="#">DINLEYICIYI BAŞLAYIN</a>
<a href="#">Kanal dinleyicisinin durdurulması</a>	<a href="#">DINLEYICIYI DURDUR</a>
<a href="#">Kanal başlatma</a>	<a href="#">KANALı BAŞLAYIN</a>
<a href="#">Bir kanalı sınaama</a>	<a href="#">PING KANALı</a>
<a href="#">Bir kanala ilişkin ileti sıra numaralarının ilk durumuna getirilmesi</a>	<a href="#">KANALı SıFIRLAYIN</a>
<a href="#">Kanaldaki belirsiz iletilerin çözülmesi</a>	<a href="#">KANALı ÇÖZÜMLE</a>
<a href="#">Kanalın durdurulması</a>	<a href="#">KANALı DURDUR</a>
<a href="#">Kanal Durumunun Görüntülenmesi</a>	<a href="#">CHSTATUS ' U GÖRÜNTÜLE</a>
<a href="#">Küme kanallarını görüntüle</a>	<a href="#">CLUSQMGR ' YI GÖRÜNTÜLE</a>

### İlgili kavramlar

[“Panoların ve komutların kullanılması” sayfa 960](#)

DQM ' yi yönetmek için MQSC komutlarını, PCF komutlarını ya da işlemleri ve denetim panolarını kullanabilirsiniz.

[“ayarlar” IBM MQ for z/OS” sayfa 881](#)

IBM MQ for z/OS sisteminizi uyarlamak için adım adım bu konuyu kullanın.

[“z/OS için iletişimi ayarlama” sayfa 974](#)

Bir dağıtım kuyruğa alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmayı dener. Başarılı olmak için, bağlantının tanımlanması ve kullanılabilir olması gerekir. Bu bölümde, bir bağlantının nasıl tanımlanacağı açıklanmaktadır.

[“Kuyruk paylaşım gruplarıyla DQM için IBM MQ for z/OS hazırlanıyor” sayfa 979](#)

IBM MQ for z/OS üzerinde kuyruk paylaşım gruplarıyla dağıtılmış kuyruğa alma yapılandırmak için bu bölümdeki yönergeleri kullanın.

[“Kuyruk paylaşım gruplarını kullanarak IBM MQ for z/OS için iletişim ayarlama” sayfa 983](#)

Bir dağıtılmış kuyruğa alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmayı dener. Bu girişimin başarılı olması için, bağlantının tanımlanması ve kullanılabilir olması gerekir.

### İlgili görevler

[“Dağıtılmış kuyruğa alma yapılandırılıyor” sayfa 189](#)

Bu bölümde, kuyruk tanımlaması, kanal tanımlaması, tetikleme ve eşitleme noktası yordamları da içinde olmak üzere IBM MQ kuruluşları arasındaki iletişimle ilgili daha ayrıntılı bilgi sağlar.

[“z/OS üzerinde diğer kuyruk yöneticileriyle iletişim kurulması” sayfa 954](#)

Bu bölümde, dağıtılmış kuyruğa alma işlevini kullanmaya başlamadan önce yapmanız gereken IBM MQ for z/OS hazırlıkları açıklanmaktadır.

## **z/OS Panoların ve komutların kullanılması**

DQM 'yi yönetmek için MQSC komutlarını, PCF komutlarını ya da işlemleri ve denetim panolarını kullanabilirsiniz.

MQSC komutlarıyla ilgili bilgi için bakınız: [Administering IBM MQ using MQSC commands](#). PCF komutlarına ilişkin bilgi için [Programlanabilir Komut Biçimlerini kullanarak denetimi otomatikleştirin](#) başlıklı konuya bakın.

## İlk panonun kullanılması

İşlemleri ve denetim panolarını çağırmaya, işlev tuşlarını kullanmaya ve yardım almaya giriş için bkz. [IBM MQ for z/OS Yönetimi](#).

**Not:** İşlemleri ve denetim panolarını kullanmak için doğru güvenlik yetkisine sahip olmanız gerekir; daha fazla bilgi için [Yönetme IBM MQ for z/OS](#) başlıklı konuya ve alt konulara bakın. [Şekil 106 sayfa 960](#), bir pano oturumunu başlattığınızda görüntülenen panoyu gösterir. Panodan sonraki metin, bu panoda gerçekleştirdiğiniz işlemleri açıklar.

```
IBM MQ for z/OS - Main Menu
Complete fields. Then press Enter.

Action . . . . . 1 0. List with filter 4. Manage
1. List or Display 5. Perform
2. Define like 6. Start
3. Alter 7. Stop
8. Command
Object type . . . . . CHANNEL +
Name . . . . . *
Disposition . . . . . A Q=Qmgr, C=Copy, P=Private, G=Group,
S=Shared, A=All

Connect name . . . . . MQ25 - local queue manager or group
Target queue manager . . . MQ25
- connected or remote queue manager for command input
Action queue manager . . . MQ25 - command scope in group
Response wait time . . . . 10 5 - 999 seconds

(C) Copyright IBM Corporation 1993, 2024. All rights reserved.

Command ==>
F1=Help F2=Split F3=Exit F4=Prompt F9=SwapNext F10=Messages
F12=Cancel
```

Şekil 106. İşlemler ve denetimler başlangıç panosu

Bu panoda şunları yapabilirsiniz:

- **Eylem** alanına uygun sayıyı yazarak gerçekleştirmek istediğiniz işlemi seçin.

- Çalışmak istediğiniz nesne tipini belirleyin. Nesne tiplerinin ne olduğundan emin değilseniz, bunların listesini görmek için F4 tuşuna basın.
- Belirlenen tipte nesnelerin listesini görüntüler. **Ad** alanına bir yıldız işareti (\*) yazın ve bu altsistemde önceden tanımlanmış olan nesnelerin listesini (belirtilen tipte) görüntülemek için Enter tuşuna basın. Daha sonra sırayla çalışmak için bir ya da daha çok nesne seçebilirsiniz. Şekil 107 sayfa 961 içinde bu şekilde üretilen kanalların bir listesi gösterilmektedir.
- **Yok etme** alanında, çalışmak istediğiniz nesnelerin kuyruk paylaşım grubundaki yok etme işlemi belirleyin. Yok etme, nesnenin nerede tutulduğunu ve nesnenin nasıl davrandığını belirler.
- **Bağlantı adı** alanında, bağlanmak istediğiniz yerel kuyruk yöneticisini ya da kuyruk paylaşım grubunu seçin. Komutların uzak bir kuyruk yöneticisinde verilmesini istiyorsanız, uzak kuyruk yöneticisinin bir kuyruk paylaşım grubunun üyesi olup olmadığına bağlı olarak, **Hedef kuyruk yöneticisi** alanını ya da **İşlem kuyruğu yöneticisi** alanını seçin. Uzak kuyruk yöneticisi bir kuyruk paylaşım grubunun üyesi değilse, **Hedef kuyruk yöneticisi** alanını seçin. Uzak kuyruk yöneticisi bir kuyruk paylaşım grubunun üyesiyse, **İşlem kuyruğu yöneticisi** alanını seçin.
- **Yanıt bekleme süresi** alanında, yanıtların alınmasına ilişkin bekleme süresini seçin.

List Channels - MQ25

Row 1 of 8

Type action codes, then press Enter. Press F11 to display connection status.  
1=Display 2=Define like 3=Alter 4=Manage 5=Perform  
6=Start 7=Stop

```
Name          Type      Disposition Status
<> *          CHANNEL  ALL      MQ25
- SYSTEM.DEF.CLNTCONN CLNTCONN QMGR MQ25
- SYSTEM.DEF.CLUSRCVR CLUSRCVR QMGR MQ25 INACTIVE
- SYSTEM.DEF.CLUSSDR  CLUSSDR  QMGR MQ25 INACTIVE
- SYSTEM.DEF.RECEIVER RECEIVER  QMGR MQ25 INACTIVE
- SYSTEM.DEF.REQUESTER REQUESTER QMGR MQ25 INACTIVE
- SYSTEM.DEF.SENDER   SENDER   QMGR MQ25 INACTIVE
- SYSTEM.DEF.SERVER   SERVER   QMGR MQ25 INACTIVE
- SYSTEM.DEF.SVRCONN  SVRCONN  QMGR MQ25 INACTIVE
***** End of list *****
```

Command ==>

F1=Help F2=Split F3=Exit F4=Filter F5=Refresh F7=Bkwd  
F8=Fwd F9=SwapNext F10=Messages F11=Status F12=Cancel

Şekil 107. Kanalları listeleme

## z/OS üzerinde kanal tanımlanması

z/OS üzerinde, MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak bir kanal tanımlayabilirsiniz.

### Yordam

- MQSC komutlarını kullanarak bir kanal tanımlamak için **DEFINE CHANNEL** komutunu kullanın.
- İlk panodan başlayarak işlemleri ve denetim panolarını kullanmak için aşağıdaki alanlara veri girin ve Enter tuşuna basın:

Çizelge 65. İşlemler ve denetim panoları: ilk pano alanları	
Alan	Alana girilecek değer
Eylem	2 (Beğenmeyi tanımla)
Nesne tipi	Kanal tipi (örneğin, SENDER) ya da CHANNEL
Ad	

Çizelge 65. İşlemler ve denetim panoları: ilk pano alanları (devamı var)	
<b>Alan</b>	<b>Alana girilecek değer</b>
Süre	Yeni nesnenin konumu.

Size, tanımlamakta olduğunuz kanal için istediğiniz ad ve özniteliklerle ilgili bilgileri içeren bazı panolar sunulur. Bunlar varsayılan öznitelik değerleriyle kullanıma hazırlanır. Enter tuşuna basmadan önce istediğiniz değişiklikleri değiştirin.

**Not: Nesne tipi** alanına CHANNEL girdiyse, önce **Geçerli Bir Kanal Tipi Seç** panosu görüntülenir.

Var olan bir kanalla aynı özniteliklere sahip bir kanal tanımlamak istiyorsanız, kopyalamak istediğiniz kanalın adını ilk panodaki **Name** alanına yazın. Panolar, var olan nesnenin öznitelikleriyle kullanıma hazırlanır.

Kanal öznitelikleri hakkında bilgi için bkz. [Kanal öznitelikleri](#).

**Not:**

1. Ağınızdaki tüm kanalları benzersiz bir şekilde adlandırın. Tüm kanalları gösteren ağ çizgesinde gösterildiği gibi, kanal adında kaynak ve hedef kuyruk yöneticisi adları da dahil olmak üzere, bu adlandırma için iyi bir yoldur.

## Sonraki adım

Kanalınızı tanımladıktan sonra kanalın güvenliğini sağlamanız gerekir. Daha fazla bilgi için bkz. "[Kanal güvenliğinin sağlanması](#)" sayfa 964.

### **Kanal tanımının değiştirilmesi**

Bir kanal tanımlamasını MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak değiştirebilirsiniz.

MQSC komutlarını kullanarak bir kanal tanımlamasını değiştirmek için ALTER CHANNEL komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

<b>Alan</b>	<b>Değer</b>
Eylem	3 (Değiştir)
Nesne tipi	kanal tipi (örneğin SENDER) ya da CHANNEL
Ad	CHANNEL.TO.ALTER
Süre	Saklanan nesnenin konumu.

Size, kanalın yürürlükteki özniteliklerine ilişkin bilgi içeren bazı panolar sunulur. Yeni değeri yazarak istediğiniz korunmayan alanları değiştirin ve kanal tanımını değiştirmek için Enter tuşuna basın.

Kanal öznitelikleri hakkında bilgi için bkz. [Kanal öznitelikleri](#).

### **Kanal tanımının görüntülenmesi**

Bir kanal tanımlamasını MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak görüntüleyebilirsiniz.

Bir kanal tanımlamasını MQSC komutlarını kullanarak görüntülemek için DISPLAY CHANNEL komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

<b>Alan</b>	<b>Değer</b>
Eylem	1 (Listele ya da Görüntüle)
Nesne tipi	kanal tipi (örneğin SENDER) ya da CHANNEL
Ad	CHANNEL.TO.DISPLAY
Süre	Nesnenin konumu.

Kanalın yürürlükteki özniteliklerine ilişkin bilgileri görüntüleyen bazı panolar görüntülenir.

Kanal öznitelikleri hakkında bilgi için bkz. [Kanal öznitelikleri](#).

### **Kanal tanımının silinmesi**

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak bir kanal tanımlamasını silebilirsiniz.

MQSC komutlarını kullanarak bir kanal tanımlamasını silmek için DELETE CHANNEL komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

<b>Alan</b>	<b>Değer</b>
Eylem	4 (Yönet)
Nesne tipi	kanal tipi (örneğin SENDER) ya da CHANNEL
Ad	CHANNEL.TO.DELETE
Süre	Nesnenin konumu.

Başka bir paneliniz var. Bu panoda işlev tipi 1 'i seçin.

Kanal tanımını silmek için Enter tuşuna basın; yeniden Enter tuşuna basarak kanal tanımını silmek istediğinizi doğrulamanız istenir.

**Not:** Kanal tanımının silinebilmesi için kanal başlatıcının çalışıyor olması gerekir (istemci-bağlantı kanalları dışında).

### **Kanal başlatıcısına ilişkin bilgilerin görüntülenmesi**

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak kanal başlatıcısına ilişkin bilgileri görüntüleyebilirsiniz.

Kanal başlatıcısına ilişkin bilgileri MQSC komutlarını kullanarak görüntülemek için DISPLAY CHINIT komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

<b>Alan</b>	<b>Değer</b>
Eylem	1 (Görüntü)
Nesne tipi	SYSTEM
Ad	Boş

Başka bir paneliniz var. Bu panoda işlev tipi 1 'i seçin.

#### **Not:**

1. Çok sayıda kanalınız varsa, dağıtılmış kuyruğa alma bilgilerinin görüntülenmesi biraz zaman alabilir.
2. Dağıtılmış kuyruğa alma ile ilgili bilgileri görüntüleyebilmek için kanal başlatıcının çalışıyor olması gerekir.

## **z/OS** Kanal güvenliğini sağlanması

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak bir kanalı koruyabilirsiniz.

MQSC komutlarını kullanarak bir kanalı güvenli kılmak için SET CHLAUTH komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

Alan	Değer
Eylem	8

Size içinde MQSC komutu sağlayabileceğiniz bir düzenleyici sunulur; bu durumda CHLAUTH komutu, bkz. Şekil 108 sayfa 964. Komutu yazmayı tamamladığınızda artı işaretleri (+) gerekir. Düzenleyiciden çıkmak ve komutu komut sunucusuna sunmak için PF3 yazın.

```
***** Top of Data *****
000001 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000002 TYPE(SSLPEERMAP) +
000003 SSLPEER('CN="John Smith"') +
000004 MCAUSER('PUBLIC')
***** Bottom of Data *****

Command ==>                               Scroll ==> PAGE
F1=Help   F3=Exit   F4=LineEdit F12=Cancel
```

Şekil 108. Komut Girişi

Daha sonra komutun çıktısı size sunulur, bkz. Şekil 109 sayfa 964

```
***** Top of Data *****
000001 CSQU000I CSQUTIL IBM MQ for z/OS V7.1.0
000002 CSQU001I CSQUTIL Queue Manager Utility - 2011-04-20 14:42:58
000003 COMMAND TGTQMGR(MQ23) RESPTIME(30)
000004 CSQU127I Executing COMMAND using input from CSQUCMD data set
000005 CSQU120I Connecting to MQ23
000006 CSQU121I Connected to queue manager MQ23
000007 CSQU055I Target queue manager is MQ23
000008 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000009 TYPE(SSLPEERMAP) +
000010 SSLPEER('CN="John Smith"') +
000011 MCAUSER('PUBLIC')
000012 CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
000013 CSQ9022I !MQ23 CSQMCSA ' SET CHLAUTH' NORMAL COMPLETION
000014 CSQU057I 1 commands read
000015 CSQU058I 1 commands issued and responses received, 0 failed
000016 CSQU143I 1 COMMAND statements attempted
000017 CSQU144I 1 COMMAND statements executed successfully
000018 CSQU148I CSQUTIL Utility completed, return code=0
Command ==>                               Scroll ==> PAGE
F1=Help   F3=Exit   F5=Rfind   F6=Rchange F9=SwapNext F12=Cancel
```

Şekil 109. Komut Çıkışı

## **z/OS** Kanal başlatıcı başlatılması

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak bir kanal başlatıcı başlatabilirsiniz.

MQSC komutlarını kullanarak bir kanal başlatıcıyı başlatmak için START CHINIT komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

Alan	Değer
Eylem	6 (Başlat)
Nesne tipi	SYSTEM



Alan	Değer
Ad	Boş

Start a System Function (Sistem İşlevi Başlat) panosu görüntülenir. Aşağıdaki panoyu izleyen metin hangi işlemin yapılması gerektiğini açıklar:

```

Start a System Function

Select function type, complete fields, then press Enter to start system
function.

Function type . . . . . _ 1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
JCL substitution . . . . . -----
-----

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type . . . . . _ L=LU6.2, T=TCP/IP
LU name (LU6.2) . . . . . -----
Port number (TCP/IP) . . . 1414
IP address (TCP/IP) . . . -----

Command ==>-----
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel

```

Şekil 110. Sistem İşlevi Başlatılması

İşlev tipi 1 'i (kanal başlatıcı) seçin ve Enter tuşuna basın.

### Kanal Başlatıcının Durdurulması

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak bir kanal başlatıcıyı durdurabilirsiniz.

MQSC komutlarını kullanarak bir kanal başlatıcıyı durdurmak için STOP CHINIT komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

Alan	Değer
Eylem	7 (Durdur)
Nesne tipi	SYSTEM
Ad	Boş

Stop a System Function (Sistem İşlevini Durdur) panosu görüntülenir. Panoyu izleyen metin, bu panoyu nasıl kullanacağınızı açıklar:

```

Stop a System Function

Select function type, complete fields, then press Enter to stop system
function.

Function type . . . . . _ 1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
Restart shared channels Y Y=Yes, N=No

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type . . . . . _ L=LU6.2, T=TCP/IP

Port number (TCP/IP) . . . -----
IP address (TCP/IP) . . . -----

Command ==> -----
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel

```

Şekil 111. İşlev denetiminin durdurulması

İşlev tipi 1 'i (kanal başlatıcı) seçin ve Enter tuşuna basın.

Kanal başlatıcı, durmadan önce tüm çalışan kanalların susturma kipinde durmasını bekler.

**Not:** Kanallardan bazıları çalışan ancak etkin olmayan alıcı ya da istekçi kanallarıysa, alıcı ya da gönderen kanal başlatıcısına verilen bir durdurma isteği hemen durmasına neden olur.

Ancak, iletiler akıyorsa, kanal başlatıcısı durmadan önce yürürlükteki ileti kümesinin tamamlanmasını bekler.

### z/OS Kanal dinleyicisinin başlatılması

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak bir kanal dinleyicisi başlatabilirsiniz.

MQSC komutlarını kullanarak bir kanal dinleyicisini başlatmak için START LISTENER komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

Alan	Değer
Eylem	6 (Başlat)
Nesne tipi	SYSTEM
Ad	Boş

Start a System Function (Sistem İşlevi Başlat) panosu görüntülenir (bkz. Şekil 110 sayfa 965).

İşlev tipi 2 'yi (kanal dinleyicisi) seçin. Gelen yok etme seçeneğini belirleyin. İletim tipini seçin. İletim tipi L ise LU adını seçin. İletim tipi T ise, Kapı numarasını ve (isteğe bağlı olarak) IP adresini seçin. Enter tuşuna basın.

**Not:** TCP/IP dinleyicisi için, birden çok Kapı ve IP adresi birleşimi başlatabilirsiniz.

### z/OS Kanal dinleyicisinin durdurulması

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak bir kanal dinleyicisini durdurabilirsiniz.

MQSC komutlarını kullanarak bir kanal dinleyicisini durdurmak için STOP LISTENER komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

<b>Alan</b>	<b>Değer</b>
Eylem	7 (Durdur)
Nesne tipi	SYSTEM
Ad	Boş

Stop a System Function (Sistem İşlevini Durdur) panosu görüntülenir (bkz. Şekil 111 sayfa 966 ).

İşlev tipi 2 'yi (kanal dinleyicisi) seçin. Gelen yok etme seçeneğini belirleyin. İletim tipini seçin. İletim tipi 'T' ise, Kapı numarasını ve (isteğe bağlı olarak) IP adresini seçin. Enter tuşuna basın.

**Not:** TCP/IP dinleyicisi için, belirli Kapı ve IP adresi birleşimlerini durdurabilir ya da tüm birleşimleri durdurabilirsiniz.

### **Kanal başlatılması**

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak bir kanal başlatabilirsiniz.

MQSC komutlarını kullanarak bir kanal başlatmak için START CHANNEL komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

<b>Alan</b>	<b>Değer</b>
Eylem	6 (Başlat)
Nesne tipi	kanal tipi (örneğin SENDER) ya da CHANNEL
Ad	CHANNEL.TO.USE
Süre	Nesnenin atılması.

Start a Channel (Kanal Başlat) panosu görüntülenir. Panoyu izleyen metinde panonun nasıl kullanılacağı açıklanır:

```
Start a Channel
```

```
Select disposition, then press Enter to start channel.
```

```
Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE
```

```
Disposition . . . . . P P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager
```

```
Command ==>
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel
```

Şekil 112. Kanal başlatılması

Kanal eşgörünümünün atılacağı ve başlatılacağı kuyruk yöneticisini seçin.

Kanalı başlatmak için Enter tuşuna basın.

### **z/OS Paylaşılan kanal başlatılması**

Paylaşılan bir kanalı başlatmak ve bu kanalı aday gösterilen bir kanal başlatıcısında tutmak için, disposition = S (START CHANNEL komutunda CHLDISP (FIXSHARED) seçeneğini belirleyin).

Paylaşılan kanalın aynı anda çalışan tek bir örneği olabilir. Kanalın ikinci bir örneğini başlatma girişimleri başarısız olur.

Bir kanalı bu şekilde başlattığınızda, söz konusu kanal için aşağıdaki kurallar geçerlidir:

- Kanalı, kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisinden durdurabilirsiniz. Başlatıldığı kanal başlatıcısı, durdurma isteği sunulduğu sırada çalışmıyorsa da durdurabilirsiniz. Kanal durduğunda, aynı ya da başka bir kanal başlatıcısında yok etme = S (CHLDISP (FIXSHARED)) belirleyerek kanalı yeniden başlatabilirsiniz. Ayrıca, atma = A (CHLDISP (SHARED)) belirleyerek de başlatabilirsiniz.
- Kanal başlangıç ya da yeniden deneme durumundaysa, aynı ya da farklı bir kanal başlatıcısında yok etme = S (CHLDISP (FIXSHARED)) belirleyerek kanalı yeniden başlatabilirsiniz. Ayrıca, atma = A (CHLDISP (SHARED)) belirleyerek de başlatabilirsiniz.
- Kanal, etkin değil durumuna geçtiğinde başlatılabilir. Tetikleyici olarak başlatılan paylaşılan kanallar her zaman paylaşılan bir yok etme (CHLDISP (SHARED)) içerir.
- Kanal, etkin değil durumuna geçtiğinde, herhangi bir kanal başlatıcısında CHLDISP (FIXSHARED) ile başlatılabilir. Ayrıca, atma = A (CHLDISP (SHARED)) belirleyerek de başlatabilirsiniz.
- Kanal, başlatıldığı kanal başlatıcısı SHARED (RESTART) ile durdurulduğunda ya da kanal başlatıcısı olağandışı sona erdiğinde, kuyruk paylaşım grubundaki başka bir etkin kanal başlatıcısı tarafından kurtarılmaz. Kanal, yalnızca başlatıldığı kanal başlatıcısı yeniden başlatıldığında kurtarılır. Bu, başarısız kanal kurtarma girişimlerinin kuyruk paylaşım grubundaki diğer kanal başlatıcılarına iletilmesini durdurur ve bu da iş yüklerine eklenir.

### **z/OS Kanalın sınanması**

Bir kanalı MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak sınavabilirsiniz.

Bir kanalı MQSC komutlarını kullanarak sınamak için PING CHANNEL komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

<b>Alan</b>	<b>Değer</b>
Eylem	5 (Gerçekleştir)
Nesne tipi	SENDER, SERVER ya da KANAL
Ad	CHANNEL.TO.USE
Süre	Kanal nesnesinin atılması.

Perform a Channel Function (Kanal İşlevi Gerçekleştir) panosu görüntülenir. Panoyu izleyen metinde panonun nasıl kullanılacağı açıklanır:

Perform a Channel Function

Select function type, complete fields, then press Enter.

Function type . . . . . \_ 1. Reset 3. Resolve with commit  
2. Ping 4. Resolve with backout

Channel name . . . . . : CHANNEL.TO.USE  
Channel type . . . . . : SENDER  
Description . . . . . : Description of CHANNEL.TO.USE

Disposition . . . . . P P=Private on MQ25  
S=Shared on MQ25  
A=Shared on any queue manager

Sequence number for reset . . 1 1 - 99999999  
Data length for ping . . . 16 16 - 32768

Command ==> \_\_\_\_\_  
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel

Şekil 113. Kanalın sınanması

İşlev tipi 2 'yi (ping) seçin.

Sınamanın yapılacağı ve sınanacağı kuyruk yöneticisini seçin.

Veri uzunluğu başlangıçta 16 olarak ayarlanır. İstiyorsanız değiştirin ve Enter tuşuna basın.

### **z/OS Bir kanala ilişkin ileti sıra numaralarının ilk durumuna getirilmesi**

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak bir kanala ilişkin ileti sıra numaralarını ilk durumuna getirebilirsiniz.

MQSC komutlarını kullanarak kanal sıra numaralarını sıfırlamak için RESET CHANNEL komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

Alan	Değer
Eylem	5 (Gerçekleştir)
Nesne tipi	kanal tipi (örneğin SENDER) ya da CHANNEL
Ad	CHANNEL.TO.USE
Süre	Kanal nesnesinin atılması.

Perform a Channel Function (Kanal İşlevi Gerçekleştir) panosu görüntülenir (bkz. [Şekil 113 sayfa 969](#)).

İşlev tipi 1 'i (ilk durumuna getir) seçin.

İlk duruma getirme işleminin yapılacağı ve hangi kuyruk yöneticisinde yapılacağı kanalı seçin.

**Sıra numarası** alanı başlangıçta bir olarak ayarlanır. İstiyorsanız bu değeri değiştirin ve Enter tuşuna basın.

### **z/OS Kanaldaki belirsiz iletilerin çözülmesi**

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak bir kanaldaki belirsiz iletileri çözebilirsiniz.

MQSC komutlarını kullanarak bir kanaldaki belirsiz iletileri çözmek için RESOLVE CHANNEL komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

<b>Alan</b>	<b>Değer</b>
Eylem	5 (Gerçekleştir)
Nesne tipi	SENDER, SERVER ya da KANAL
Ad	CHANNEL.TO.USE
Süre	Nesnenin atılması.

Perform a Channel Function (Kanal İşlevi Gerçekleştir) panosu görüntülenir (bkz. Şekil 113 sayfa 969 ).

İşlev tipi 3 ya da 4 'ü seçin (kesinleştirme ya da geriletme ile çözümlayin). (Ek bilgi için bkz. "[Belirsiz kanalların işlenmesi](#)" sayfa 228 .)

Çözünürlüğün yapılacağı ve hangi kuyruk yöneticisi üzerinde yapılacağı kanal için yok etme seçeneğini belirleyin. Enter tuşuna basın.

### **Kanalı durdurma**

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak bir kanalı durdurabilirsiniz.

MQSC komutlarını kullanarak bir kanalı durdurmak için STOP CHANNEL komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

<b>Alan</b>	<b>Değer</b>
Eylem	7 (Durdur)
Nesne tipi	kanal tipi (örneğin SENDER) ya da CHANNEL
Ad	CHANNEL.TO.USE
Süre	Nesnenin atılması.

Stop a Channel (Kanal Durdur) panosu görüntülenir. Panoyu izleyen metinde panonun nasıl kullanılacağı açıklanır:

```
Stop a Channel
Complete fields, then press Enter to stop channel.

Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE

Disposition . . . . . P   P=Private on MQ25
A=Shared on any queue manager

Stop mode . . . . . 1   1. Quiesce  2. Force
Stop status . . . . . 1   1. Stopped  2. Inactive

Queue manager . . . . . -----
Connection name . . . . . -----

Command ==> -----
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

Şekil 114. Kanalı durdurma

Durdurma işlemleri yapılacak kanalı ve durdurulacak kuyruk yöneticisini seçin.

Gerek duyduğunuz durdurma kipini seçin:

#### **Susturma**

Yürürlükteki ileti tamamlandığında ve toplu iş sona erdirildiğinde, toplu iş değerine ulaşılmadığında ve iletim kuyruğunda bekleyen iletiler olduğunda kanal durur. Yeni toplu işler başlatılmaz. Varsayılan kip bu kiptir.

#### **Zorla**

Kanal hemen durur. Devam eden bir ileti kümesi varsa, 'belirsiz' durumu ortaya çıkabilir.

Durdurmak istediğiniz kanal için kuyruk yöneticisini ve bağlantı adını seçin.

Gerek duyduğunuz durumu seçin:

#### **Durduruldu**

Kanal otomatik olarak yeniden başlatılmaz ve el ile yeniden başlatılması gerekir. Kuyruk yöneticisi ya da bağlantı adı belirtilmezse, bu kip varsayılan kiptir. Bir ad belirtilirse, bu ada izin verilmez.

#### **Etkin Değil**

Kanal gerektiğinde otomatik olarak yeniden başlatılır. Bir kuyruk yöneticisi ya da bağlantı adı belirtilirse, bu kip varsayılan kiptir.

Kanalı durdurmak için Enter tuşuna basın.

Ek bilgi için bkz. "[Kanalları durdurma ve susturma](#)" sayfa 226 . Durdurulan kanalları yeniden başlatma hakkında bilgi için bkz. "[Durdurulan kanallar yeniden başlatılıyor](#)" sayfa 228.

**Not:** Paylaşılan bir kanal yeniden deneme durumundaysa ve başlatıldığı kanal başlatıcısı çalışmıyorsa, komutun girildiği kuyruk yöneticisinde kanal için STOP isteği yayınlanır.

### **Kanal durumunun görüntülenmesi**

MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak kanal durumunu görüntüleyebilirsiniz.

MQSC komutlarını kullanarak bir kanalın ya da kanal kümesinin durumunu görüntülemek için DISPLAY CHSTATUS komutunu kullanın.

**Not:** Çok sayıda kanalınız varsa, kanal durumu bilgilerinin görüntülenmesi biraz zaman alabilir.

Kanal Listele panosundaki işlemleri ve denetim panolarını kullanarak (bkz. [Şekil 107 sayfa 961](#) ), her bir kanal için kanal durumunun bir özeti aşağıdaki gibi gösterilir:

ETKİNDEĞİL	Etkin bağlantı yok
<i>durum</i>	Bir bağlantı etkin
<i>nnn durum</i>	Birden çok bağlantı güncel ve tüm yürürlükteki bağlantılar aynı durumda
<i>Nnn</i>	Birden çok bağlantı güncel ve yürürlükteki bağlantıların tümü aynı durumda değil
YÜRÜRLÜKTEKİ	
Boş	IBM MQ kaç bağlantının etkin olduğunu saptayamıyor (örneğin, kanal başlatıcısı çalışmadığından)

**Not:** Yok etme GRUBU olan kanal nesnelere için durum görüntülenmez.

Burada *nnn* etkin bağlantıların sayısıdır ve *durum* aşağıdakilerden biridir:

INIT	kullanıma hazırlama
Bağ Tanımla	Bağ Tanımı
START	BAŞLATILYOR
Çalıştır	ÇALIŞTIRILYOR

DUR DURDURULUYOR ya da DURDURULDU  
Yeniden deneme YENIDEN ÇALIŞMA  
GEREKÇE İSTEKLE

Kanal durumuna ilişkin ek bilgi görüntülemek için, Liste Kanalı ya da Görüntü Birimi 'ndeki Durum tuşuna (F11) ya da Kanal panolarını Değiştir düğmesine basarak Liste Kanalları-Yürürlükteki Durum panosunu görüntüleyin (bkz. Şekil 115 sayfa 972 ).

```
List Channels - Current Status - MQ25          Row 1 of 16
Type action codes, then press Enter. Press F11 to display saved status.
1=Display current status

Channel name      Connection name      State
Start time      Messages Last message time Type Disposition
<> *           CHANNEL ALL      MQ25

_ RMA0.CIRCUIT.ACL.F RMA1          STOP
_ 2005-03-21 10.22.36 557735 2005-03-24 09.51.11 SENDER PRIVATE MQ25
_ RMA0.CIRCUIT.ACL.N RMA1          STOP
_ 2005-03-21 10.23.09 378675 2005-03-24 09.51.10 SENDER PRIVATE MQ25
_ RMA0.CIRCUIT.CL.F RMA2          STOP
_ 2005-03-24 01.12.51 45544 2005-03-24 09.51.08 SENDER PRIVATE MQ25
_ RMA0.CIRCUIT.CL.N RMA2          STOP
_ 2005-03-24 01.13.55 45560 2005-03-24 09.51.11 SENDER PRIVATE MQ25
_ RMA1.CIRCUIT.CL.F RMA1          STOP
_ 2005-03-21 10.24.12 360757 2005-03-24 09.51.11 RECEIVER PRIVATE MQ25
_ RMA1.CIRCUIT.CL.N RMA1          STOP
_ 2005-03-21 10.23.40 302870 2005-03-24 09.51.09 RECEIVER PRIVATE MQ25
***** End of list *****
Command ==>
F1=Help F2=Split F3=Exit F4=Filter F5=Refresh F7=Bkwd
F8=Fwd F9=SwapNext F10=Messages F11=Saved F12=Cancel
```

Şekil 115. Kanal bağlantılarının listelenmesi

Durum değerleri aşağıdaki gibidir:

INIT kullanıma hazırlama  
Bağ Tanımla Bağ Tanımı  
START BAŞLATILYOR  
Çalıştır ÇALIŞTIRILYOR  
DUR DURDURULUYOR ya da DURDURULDU  
Yeniden deneme YENIDEN ÇALIŞMA  
GEREKÇE İSTEKLE  
ŞÜPHE DURDURULDU ve INDOUBT (YES)

Ek bilgi için bkz. “Kanal durumları” sayfa 219 .

Kaydedilmiş durumdaki kanal bağlantılarının benzer bir listesini görmek için F11 tuşuna basabilirsiniz; yürürlükteki listeye dönmek için F11 tuşuna basın. Kaydedilen durum, kanalda en az bir ileti kümesi iletilinceye kadar geçerli olmaz.

Bir bağlantı seçmek için işlem kodu 1 ya da eğik çizgi (/) kullanın ve Enter tuşuna basın. Display Channel Connection Current Status (Kanal Bağlantısını Geçerli Durumunu Görüntüle) panoları görüntülenir.



## **z/OS Küme kanallarının görüntülenmesi**

Küme kanallarını MQSC komutlarını kullanarak ya da işlemleri ve denetim panolarını kullanarak görüntüleyebilirsiniz.

Tanımlanmış tüm küme kanallarını (belirtik olarak ya da otomatik tanımlama kullanılarak) görüntülemek için MQSC DISPLAY CLUSQMGR komutunu kullanın.

İlk panodan başlayarak işlemleri ve denetim panolarını kullanarak bu alanları doldurun ve Enter tuşuna basın:

Alan	Değer
Eylem	1 (Listele ya da Görüntüle)
Nesne tipi	CLUSCHL.
Ad	*

Size, her küme kanalına ilişkin bilgilerin üç satırı kapladığı ve kanal, küme ve kuyruk yöneticisi adlarını içerdiği şekil [Şekil 116 sayfa 973](#) gibi bir pano sunulur. Küme-gönderen kanalları için genel durum gösterilir.

```
List Cluster queue manager Channels - MQ25      Row 1 of 9
Type action codes, then press Enter. Press F11 to display connection status.
1=Display 5=Perform 6=Start 7=Stop

Channel name      Connection name      State
Type      Cluster name      Suspended
Cluster queue manager name      Disposition
<> *          -      MQ25
- TO.MQ90.T      HURSLEY.MACH90.COM(1590)
- CLUSRCVR      VJH01T              N
- MQ90          -      MQ25
- TO.MQ95.T      HURSLEY.MACH95.COM(1595)      RUN
- CLUSSDRA      VJH01T              N
- MQ95          -      MQ25
- TO.MQ96.T      HURSLEY.MACH96.COM(1596)      RUN
- CLUSSDRB      VJH01T              N
- MQ96          -      MQ25
***** End of list *****

Command ==>-----
F1=Help  F2=Split  F3=Exit  F4=Filter  F5=Refresh  F7=Bkwd
F8=Fwd   F9=SwapNext  F10=Messages  F11=Status  F12=Cancel
```

Şekil 116. Küme kanalları listeleniyor

Bir ya da daha çok kanal hakkında tam bilgi görüntülemek için, adlarına işlem kodu 1 yazın ve Enter tuşuna basın. İşlevleri (ping, resolve ve reset) gerçekleştirmek ve bir küme kanalını başlatmak ya da durdurmak için 5, 6 ya da 7 numaralı işlem kodlarını kullanın.

Kanal durumuna ilişkin ek bilgi görüntülemek için Status (Durum) tuşuna basın (F11).

## **z/OS IBM MQ for z/OS Olanacağının zEnterprise Data Compression Express Olanacağını Kullanmak Üzere Hazırlanması**

zEnterprise Veri Sıkıştırma (zEDC) Express olanağı, IBM zEC12 GA2' den başlayarak, en az z/OS z/OS 2.1 düzeyi kullanılarak IBM Z makinelerinin belirli modellerinde kullanılabilir.

Daha fazla bilgi için bkz. [zEnterprise Veri Sıkıştırma \(zEDC\)](#) .

## Önkoşullar

IBM z15 ve sonraki sürümlerinde, zEnterprise Data Compression (zEDC) Express olanağı, zEDC için Integrated Accelerator olarak yongada olacak donanım sisteminin PCIe G/Ç çekmecesindeki isteğe bağlı bir aksamdan taşınmıştır. Bu değişikliklerle, yapılandırma önkoşulları güncellenir ve donanım sisteminize bağlıdır.

### IBM z15 ya da sonraki sürümü

z/OS düzeyinize göre aşağıdaki PTF 'lerden birini uygulayın:

- z/OS 2.4: UJ00636
- z/OS 2.3: UJ00635
- z/OS 2.2: UJ00638
- z/OS 2.1: UJ00639

z15 ya da sonraki sistemler için donanım gereksinimi yoktur. Bu sistemlerdeki zEDC için Integrated Accelerator çözümü, yerleşik veri hızlandırma sağlar ve bu nedenle ayrı bir bağdaştırıcıya artık gerek kalmaz.

### IBM zEC12 GA2 - IBM z14

Sisteminizin aşağıdaki gereksinimleri de olmalıdır:

- Donanım sisteminin PCIe G/Ç çekmecelerine takılı bir zEDC Express<sup>®</sup> bağdaştırıcısı.
- zEDC yazılım yeteneği (isteğe bağlı, ücretli bir özellik) bir IFAPRDxx parmlib üyesinde etkinleştirilmelidir.

## Yordam

### IBM zEC12 GA2 - IBM z14

Kanal başlatıcı kullanıcı kimliğinin RACF FACILITY CLASS 'taki FPZ.ACCELERATOR.COMPRESSION profili için ya da dış güvenlik yöneticisinde (ESM) işletmenizin kullandığı eşdeğer bir yetkiye sahip olduğundan emin olun.



**Uyarı:** IBM z15 ya da sonraki yayın sonlarında ebilir.

### IBM zEnterprise zEC12 GA2 ya da sonraki sürümü

Hem gönderme hem de alma uçlarında COMPMSG (ZLIBFAST) ile kanalı yapılandırın. Yapılandırıldıktan sonra, kanal boyunca akan iletileri sıkıştırmak ve açmak için zlib sıkıştırması kullanılır.

Sıkıştırılacak verilerin boyutu eşik alt sınırının üzerindeyse, zEDC 'de sıkıştırma gerçekleştirilir. Eşik, kullanılmakta olan IBM z donanımına bağlıdır

- IBM zEC12 GA2 - IBM z14 , en az 4KB eşiğine sahiptir
- IBM z15 ya da sonraki bir yayın düzeyinin eşığı alt sınırı 1KB

Eşik boyutunun altındaki iletiler için yazılımda sıkıştırma veya enflasyon gerçekleştirilir.

## z/OS için iletişimi ayarlama

Bir dağıtım kuyruğa alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmayı dener. Başarılı olmak için, bağlantının tanımlanması ve kullanılabilir olması gerekir. Bu bölümde, bir bağlantının nasıl tanımlanacağı açıklanmaktadır.

DQM, IBM MQ için uzak bir kuyruğa alma olanağıdır. İletişim bağlantılarına arabirim oluşturan kuyruk yöneticisi için kanal denetim programları sağlar. Bu bağlantılar sistem işletmeni tarafından denetlenebilir. Dağıtılmış kuyruk yönetimi tarafından tutulan kanal tanımları bu bağlantıları kullanır.

z/OS için kullanılacak iki iletişim protokolü biçiminden birini seçin:

- “z/OS üzerinde TCP bağlantısı tanımlanması” sayfa 975
- “APPC/MVS kullanılarak z/OS için LU6.2 bağlantısı tanımlanması” sayfa 978

**MQ Adv.** **CD** TCP/IP kullanan bir ileti kanalı, ağ verimini önemli ölçüde artıracak hızlı bir TCP/IP tüneli sağlayan bir IBM Aspera faspio Gateway' e işaret edilebilir. Herhangi bir yetkili platformda çalışan bir kuyruk yöneticisi, Aspera gateway aracılığıyla bağlanabilir. Ağ geçidinin kendisi Red Hat ya da Ubuntu Linux ya da Windows üzerinde devreye alınır. Bkz. [Linux ya da Windows üzerinde Aspera gateway bağlantısı tanımlanması](#).

Her kanal tanımı, iletim protokolü (İletim Tipi) özniteliği olarak tek bir protokol belirlemelidir. Bir kuyruk yöneticisi iletişim kurmak için birden çok iletişim kuralı kullanılabilir.

Örnek yapılandırma- IBM MQ for z/OS konusuna başvurmanız da yararlı olabilir. Kuyruk paylaşım gruplarını kullanıyorsanız, bkz. [“Kuyruk paylaşım gruplarını kullanarak IBM MQ for z/OS için iletişim ayarlama” sayfa 983](#).

### İlgili kavramlar

“Panoların ve komutların kullanılması” sayfa 960

DQM ' yi yönetmek için MQSC komutlarını, PCF komutlarını ya da işlemleri ve denetim panolarını kullanabilirsiniz.

“ayarlama IBM MQ for z/OS” sayfa 881

IBM MQ for z/OS sisteminizi uyarlamak için adım adım bu konuyu kullanın.

“z/OS üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 958

Uzak kuyruk yöneticilerine kanallar yaratmak, izlemek ve denetlemek için DQM komutlarını ve panolarını kullanın.

“Kuyruk paylaşım gruplarıyla DQM için IBM MQ for z/OS hazırlanıyor” sayfa 979

IBM MQ for z/OS üzerinde kuyruk paylaşım gruplarıyla dağıtılmış kuyruğa alma yapılandırmak için bu bölümdeki yönergeleri kullanın.

“Kuyruk paylaşım gruplarını kullanarak IBM MQ for z/OS için iletişim ayarlama” sayfa 983

Bir dağıtılmış kuyruğa alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmayı dener. Bu girişimin başarılı olması için, bağlantının tanımlanması ve kullanılabilir olması gerekir.

### İlgili görevler

“Dağıtılmış kuyruğa alma yapılandırılıyor” sayfa 189

Bu bölümde, kuyruk tanımlaması, kanal tanımlaması, tetikleme ve eşitleme noktası yordamları da içinde olmak üzere IBM MQ kuruluşları arasındaki iletişimle ilgili daha ayrıntılı bilgi sağlanır.

“z/OS üzerinde diğer kuyruk yöneticileriyle iletişim kurulması” sayfa 954

Bu bölümde, dağıtılmış kuyruğa alma işlevini kullanmaya başlamadan önce yapmanız gereken IBM MQ for z/OS hazırlıkları açıklanmaktadır.

### **z/OS** **z/OS üzerinde TCP bağlantısı tanımlanması**

TCP bağlantısı tanımlamak için, konfigürasyonu tanımlanacak birçok ayar vardır.

TCP adres alanı adı, TCP sistem parametreleri veri kümesinde ( *tcpip.TCPIP.DATA*. Veri kümesinde bir "TCPIPJOBNAME *TCPIP\_proc*" deyimi bulunmalıdır.

Bir güvenlik duvarı kullanıyorsanız, kanal başlatıcısından kanallardaki adreslere ve uzak bağlantılardan kuyruk yöneticisine allow bağlantılarını yapılandırmanız gerekir.

Genellikle bir güvenlik duvarı tanımı, gönderen IP adresini ve bağlantı noktasını hedef IP adresine ve bağlantı noktasına yapılandırır:

- Bir z/OS görüntüsünün birden çok anasistem adı olabilir ve güvenlik duvarını kaynak adresi olarak birden çok anasistem adresiyle yapılandırmanız gerekebilir.

Bu adları ve adresleri görüntülemek için NETSTAT HOME komutunu kullanabilirsiniz.

- Bir kanal başlatıcısının farklı kapılarda birden çok dinleyicisi olabilir, bu nedenle bu kapıları yapılandırmanız gerekir.
- Bir kuyruk paylaşım grubu için paylaşılan bir kapı kullanıyorsanız, paylaşılan kapıyı da yapılandırmanız gerekir.

Kanal başlatıcı adres alanı, veri kümesini okuma yetkisine sahip olmalıdır. TCPIP.DATA veri kümesi:

- Ortam değişkeni, RESOLVER\_CONFIG
- Dosya sisteminde /etc/resolv.conf
- // SYSTCPD DD deyiimi
- // SYSTCPDD DD deyiimi
- *jobname/userid*.TCPIP.DATA
- SYS1.TCPPARMS(TCPDATA)
- *zapname*.TCPIP.DATA

TCPIP ' ye ilişkin üst düzey niteleyiciyi doğru olarak belirtmeye de dikkat etmeniz gerekir.

Uygun şekilde yapılandırılmış bir DNS (Domain Name System; Etki Alanı Ad Sistemi) sunucusu gerekir; bu sunucu hem Ad-IP adresi çevirisi hem de IP adresi-Ad çevirisi yapabilir.

**Not:** Çözücü yapılandırmasında yapılan bazı değişiklikler, bunu kullanan uygulamaların geri dönüştürülmesini gerektirir; örneğin, IBM MQ.

Daha fazla bilgi için aşağıdaki konuya bakın:

- [Temel TCP/IP sistemi](#)
- [z/OS UNIX System Services](#).

Başlatıldığında her TCP kanalı TCP kaynaklarını kullanır; PROFILE.TCPIP konfigürasyon verileri kümesi:

#### **ACBPOOLSIZE**

Başlatılan TCP kanalı başına bir artı bir ekle

#### **CCBPOOLSIZE**

Başlatılan TCP kanalı başına bir tane ve DQM dağıtıcısı başına bir tane artı bir tane ekleyin

#### **DATABUFFERPOOLSIZ**


Başlatılan TCP kanalı başına iki, artı bir ekle

#### **MAXFILEPROC**

Kanal başlatıcısındaki her dağıtıcının kaç kanalı işleyebileceğini denetler.

Bu parametre, SYSI.PARMLIB. Gereksinimleriniz için yeterince büyük bir değer belirttiğinizden emin olun.

Varsayılan olarak, kanal başlatıcı yalnızca TCPNAME kuyruk yöneticisi özniteliğinde adı belirtilen yığıyla ilişkilendirilmiş IP adreslerine bağlanabilir. Kanal başlatıcısının sistemdeki ek TCP/IP yığınlarını kullanarak iletişim kurmasını sağlamak için TCPSTACK kuyruk yöneticisi özniteliğini MULTIPLE olarak değiştirin.

 TCP/IP kullanan bir ileti kanalı, ağ verimini önemli ölçüde artıracak hızlı bir TCP/IP tüneli sağlayan bir IBM Aspera faspio Gateway' e işaret edilebilir. Herhangi bir yetkili platformda çalışan bir kuyruk yöneticisi, Aspera gateway aracılığıyla bağlanabilir. Ağ geçidinin kendisi Red Hat ya da Ubuntu Linux ya da Windows üzerinde devreye alınır. Bkz. [Linux ya da Windows üzerinde Aspera gateway bağlantısı tanımlanması](#).

#### **İlgili kavramlar**

[“Gönderme sonu” sayfa 977](#)

TCPIP bağlantısının gönderildiği yerde, konfigürasyonu tanımlanacak bazı ayarlar vardır.

[“TCP ' de alma” sayfa 977](#)

TCPIP bağlantısının alıcı ucunda, konfigürasyonu tanımlanacak bazı ayarlar vardır.

[“z/OS üzerinde TCP dinleyici birikim listesi seçeneğinin kullanılması” sayfa 977](#)

TCP/IP ' de alma sırasında, işlem bekleyen bağlantı isteği sayısı üst sınırı belirlenir. Bu bekleyen istekler, dinleyicinin isteği kabul etmesi için TCP/IP kapısında bekleyen isteklerin *birikim listesi* olarak kabul edilebilir.

### ► z/OS Gönderme sonu

TCP/IP bağlantısının gönderildiği yerde, konfigürasyonu tanımlanacak bazı ayarlar vardır.

Kanal tanımlamasındaki bağlantı adı (CONNNAME) alanı, anasistem adına (örneğin, MVSHUR1) ya da hedefin TCP ağ adresine ayarlanmalıdır. TCP ağ adresi IPv4 noktalı onlu biçimde (örneğin, 127.0.0.1) ya da IPv6 onaltılı biçimde (örneğin, 2001:DB8:0:0:0:0:0:0) olabilir. Bağlantı adı bir anasistem adıyla, anasistem adını TCP anasistem adresine dönüştürmek için bir TCP ad sunucusu gerekir. (Bu gereksinim, IBM MQdeğil, TCP ' nin bir işlevidir.)

Bir bağlantının başlatılırken (gönderen, istekte bulunan ve sunucu kanal tipleri), bağlantı için isteğe bağlı bir kapı numarası sağlanmalıdır; örneğin:

**Bağlantı adı**  
192.0.2.0(1555)

Bu durumda, başlatıcı uç, 1555 numaralı kapıda dinleyen bir alıcı programa bağlanmayı dener.

**Not:** İsteğe bağlı bir kapı numarası belirtilmezse, varsayılan kapı numarası 1414 kullanılır.

Kanal başlatıcı, etkin ve kullanılabilir olan herhangi bir TCP/IP yığını kullanabilir. Varsayılan olarak, kanal başlatıcı giden kanallarını TCPNAME kuyruk yöneticisi özniteliğinde adı belirtilen TCP/IP yığına ilişkin varsayılan IP adresine bağlar. Farklı bir yığın üzerinden bağlanmak için, kanalın LOCLADDR özniteliğinde yığının anasistem adını ya da IP adresini belirtmeniz gerekir.

### ► z/OS TCP ' de alma

TCP/IP bağlantısının alıcı ucunda, konfigürasyonu tanımlanacak bazı ayarlar vardır.

Kanal programlarını alma işlemi, gönderen kanaldan gelen başlatma isteğine yanıt olarak başlatılır. Bunu yapmak için, gelen ağ isteklerini saptamak ve ilişkili kanalı başlatmak için bir dinleyici programının başlatılması gerekir. Bu dinleyici programını START LISTENER komutuyla ya da işlemler ve denetim panolarını kullanarak başlatabilirsiniz.

Varsayılan olarak:

- TCP Dinleyici programı 1414 numaralı kapıyı kullanır ve TCP yığınının kullanabileceği tüm adresleri dinler.
- TCP/IP dinleyicileri yalnızca, TCPNAME kuyruk yöneticisi özniteliğinde adı belirtilen TCP/IP yığını ile ilişkilendirilmiş adreslere bağ tanımlayabilir.

Diğer adreslere ya da kullanılabilir tüm TCP yığınlarına ilişkin dinleyicileri başlatmak için TCPSTACK kuyruk yöneticisi özniteliğinizi 'MULTIPLE' olarak ayarlayın.

TCP dinleyici programınızı, START LISTENER komutunda IPADDR belirleyerek yalnızca belirli bir adres ya da anasistem adını dinleyecek şekilde başlatabilirsiniz. Daha fazla bilgi için bkz. Dinleyiciler.

### ► z/OS z/OS üzerinde TCP dinleyici birikim listesi seçeneğinin kullanılması

TCP/IP ' de alma sırasında, işlem bekleyen bağlantı isteği sayısı üst sınırı belirlenir. Bu bekleyen istekler, dinleyicinin isteği kabul etmesi için TCP/IP kapısında bekleyen isteklerin *birikim listesi* olarak kabul edilebilir.

z/OS üzerindeki varsayılan dinleyici birikim listesi değeri 10000 'dir. Birikim listesi bu değerlere ulaşırsa, TCP/IP bağlantısı reddedilir ve kanal başlatılamaz.

MCA kanalları için bu, kanalın RETRY durumuna girmesine ve daha sonra bağlantıyı yeniden denemesine neden olur.

İstemci bağlantıları için, istemci MQCONN ' dan bir MQRC\_Q\_MGR\_NOT\_VAR olan neden kodunu alır ve bağlantıyı daha sonra yeniden deneyebilir.

## İlgili kavramlar

“IBM MQ for Multiplatforms üzerinde TCP dinleyici birikim listesi seçeneğinin kullanılması” sayfa 259 TCP ' de, sunucu ile istemci arasında üç yönlü tokalaşma gerçekleşmediği sürece bağlantılar eksik olarak işlenir. Bu bağlantılara, bekleyen bağlantı istekleri denir. Bu bekleyen bağlantı istekleri için bir değer üst sınırı belirlenir ve dinleyicinin isteği kabul etmesi için TCP kapısında bekleyen isteklerin birikim listesi olarak kabul edilebilir.

## z/OS **APPC/MVS kullanılarak z/OS için LU6.2 bağlantısı tanımlanması**

LU6.2 bağlantısı tanımlamak için, yapılandırılacak bazı ayarlar vardır.

## APPC/MVS ayarı

Kanal başlatıcının her yönetim ortamı, SYS1.PARMLIB, aşağıdaki örnekteki gibi:

```
LUADD ACBNAME( Luname ) NOSCHED TPDATA(CSQ.APPCTP)
```

*Luname* , kullanılacak mantıksal birimin adıdır. NOSCHED gereklidir; TPDATA kullanılmaz. ASCHPMxx üyesi ya da APPC/MVS TP tanıtımı veri kümesi için ekleme yapılması gerekmez.

DQM tarafından kullanılan bağlantıları tanımlamak için yan bilgi veri kümesi genişletilmelidir. APPC yardımcı programı ATBSDFMU kullanılarak nasıl yapılacağını öğrenmek için sağlanan CSQ4SIDE örneğine bakın. Kullanılacak TPNAME değerlerinin ayrıntıları için aşağıdaki çizelgeye bakın:

Çizelge 66. Uzak kuyruk yöneticisi platformuna ilişkin yerel z/OS sistemindeki ayarlar	
Uzak altyapı	TPNAME
z/OS ya da MVS	Uzak kuyruk yöneticisiyle ilgili yan bilgilerdeki TPNAME ile aynı.
IBM i	IBM i sistemindeki yöneltme girişindeki karşılaştırma değeriyle aynı.
AIX and Linux sistemleri	Uzak kuyruk yöneticisiyle ilgili yan bilgilerdeki TPNAME ile aynı.
Windows	Windows Run Listener komutunda ya da Windows üzerinde TpSetup kullanılarak tanımlanan çağrılabilir Hareket Programı 'nda belirtildiği şekilde.

Aynı makinede birden çok kuyruk yöneticiniz varsa, kanal tanımlamalarındaki TPdalarının benzersiz olmasına dikkat edin.

Kuyruk yöneticisinin aynı ya da başka bir z/OS sisteminde bir kuyruk yöneticisiyle APPC kullanarak iletişim kurduğu bir ortamda, iletişim kuran LU 'ya ilişkin VTAM tanımının SECACPT (ALREADYV) belirttiğinden ya da LU' lar arasındaki bağlantıya ilişkin bir RACF APPCLU tanıtımı olduğundan emin olun. Bu tanıtım CONVSEC (ALREADYV) belirtiliyor.

Gelen ya da giden iletişimi başlatma girişiminde bulunmadan önce, temel ve dinleyici LU ' ları için z/OS komutu VARY ACTIVE verilmelidir.



**Uyarı:** APPC kuruluşuna ek olarak aşağıdaki komutu da vermeniz gerekir:

```
ALTER QMGR LUNAME(Luname)
```

ve kanal başlatıcıyı yeniden başlatın.

Daha fazla bilgi için bkz. [LUNAME](#) .

## İlgili kavramlar

“LU 6.2 ile bağlantı kuruluyor” sayfa 979

LU 6.2' ye bağlanmak için, yapılandırılacak bazı ayarlar vardır.

“LU 6.2 üzerinde alma” sayfa 979

LU 6.2 üzerinde almak için, yapılandırılacak birçok ayar vardır.

## **z/OS** LU 6.2 ile bağlantı kuruluyor

LU 6.2' ye bağlanmak için, yapılandırılacak bazı ayarlar vardır.

Kanal tanımındaki bağlantı adı (CONNNAME) alanı, APPC/MVS için ayarlanan yan bilgi verilerinde belirtildiği gibi, simgesel hedef adına ayarlanmalıdır.

Kullanılacak LU adı (APPC/MVS için daha önce açıklandığı gibi) kanal başlatıcı değiştirgelerinde de belirtilmelidir. Bu, dinleyici tarafından almak için kullanılan LU ' ya ayarlanmalıdır.

Kanal başlatıcı "SECURITY (SAME)" APPC/MVS seçeneğini kullanır; bu nedenle, giden iletimler için kullanılan kanal başlatıcı adres alanının kullanıcı kimliği olur ve alıcıya sunulur.

## **z/OS** LU 6.2 üzerinde alma

LU 6.2 üzerinde almak için, yapılandırılacak birçok ayar vardır.

Gönderen kanaldan bir başlatma isteğine yanıt olarak MCA ' ların alınması başlatılır. Bunu yapmak için, gelen ağ isteklerini saptamak ve ilişkili kanalı başlatmak için bir dinleyici programının başlatılması gerekir. Dinleyici programı bir APPC/MVS sunucusudur. Bunu START LISTENER komutuyla ya da işlemler ve denetim panolarını kullanarak başlatabilirsiniz. Kullanılacak LU adını, kenar bilgisi veri kümesinde tanımlı bir simgesel hedef adıyla belirtmeniz gerekir. Tanıtılan yerel LU, kanal başlatıcı değiştirgelerinde belirlendiği şekilde, giden iletimlerde kullanılanla aynı olmalıdır.

## **z/OS** Kuyruk paylaşım gruplarıyla DQM için IBM MQ for z/OS hazırlanıyor

IBM MQ for z/OS üzerinde kuyruk paylaşım gruplarıyla dağıtılmış kuyruğa alma yapılandırmak için bu bölümdeki yönergeleri kullanın.

Kuyruk paylaşım gruplarını kullanan örnek bir yapılandırma için bkz. [Örnek yapılandırma- IBM MQ for z/OS kuyruk paylaşım gruplarını kullanma](#). Kuyruk paylaşım gruplarını kullanan bir ileti kanalı planlama örneği için [Kuyruk paylaşım gruplarını kullanma z/OS için ileti kanalı planlama örneği](#) başlıklı konuya bakın.

Kuyruk paylaşım gruplarıyla dağıtılmış kuyruğa alma özelliğini etkinleştirmek için aşağıdaki bileşenleri yaratmanız ve yapılandırmanız gerekir:

- [LU 6.2 ve TCP/IP dinleyicileri](#)
- [İletim kuyrukları ve tetikleme](#)
- [İleti kanalı araçları](#)
- [Eşitleme kuyruğu](#)

İletişimi kurmanız için gereken bileşenleri oluşturduktan sonra bkz. [“Kuyruk paylaşım gruplarını kullanarak IBM MQ for z/OS için iletişim ayarlama” sayfa 983](#).

Kuyruk paylaşım gruplarını kullanırken kanalların nasıl izleneceğine ve denetleneceğine ilişkin bilgi için bkz. [“z/OS üzerinde kanalların izlenmesi ve denetlenmesi” sayfa 958](#).

Kuyruk paylaşımı grubu kavramları ve yararları için aşağıdaki bölümlere bakın.

### **Hizmet sınıfı**

Paylaşılan kuyruk, farklı bir hizmet sınıfı sunan bir yerel kuyruk tipidir. Paylaşılan bir kuyruktaki iletiler, kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından erişilmesine olanak sağlayan bir bağlaşım olanağında (CF) saklanır. Paylaşılan kuyruktaki bir ileti, en çok 100 MB uzunluğunda bir ileti olmalıdır.

### **Soysal arabirim**

Bir kuyruk paylaşım grubu, ağın grubu tek bir varlık olarak görüntülemesini sağlayan soysal bir arabirime sahiptir. Bu görünüm, grup içindeki herhangi bir kuyruk yöneticisine bağlanmak için kullanılabilir tek bir soysal adresle gerçekleştirilir.

Kuyruk paylaşım grubundaki her kuyruk yöneticisi, soysal adresle mantıksal olarak ilişkili bir adresteki gelen oturum isteklerini dinler. Daha fazla bilgi için bkz. [“Kuyruk paylaşım grupları için LU 6.2 ve TCP/IP dinleyicileri” sayfa 981.](#)

## Yük dengeli kanal başlangıcı

Paylaşılan iletim kuyruğuna, kuyruk paylaşım grubundaki herhangi bir kanal başlatıcısında çalışan bir giden kanal hizmet verebilir. Yük dengeli kanal başlatma, kanal başlatma komutunun hedeflendiği yeri belirler. Gerekli iletişim altsistemine erişimi olan uygun bir kanal başlatıcı seçilir. Örneğin, TRPTYPE (LU6.2) ile tanımlanan bir kanal, yalnızca TCP/IP altsistemine erişimi olan bir kanal başlatıcısında başlatılamaz.

Kanal başlatıcısı seçimi, kanal yüküne ve kanal başlatıcısının ana odasına bağlıdır. Kanal yükü, kanal başlatıcı parametrelerinde tanımlandığı şekilde izin verilen etkin kanal sayısı üst sınırının yüzdesi olarak etkin kanal sayısıdır. Başlık, etkin kanal sayısı ile izin verilen üst sınır arasındaki farktır.

Gelen paylaşılan kanallar, [“Kuyruk paylaşım grupları için LU 6.2 ve TCP/IP dinleyicileri” sayfa 981](#) içinde açıklandığı gibi soysal bir adres kullanarak kuyruk paylaşım grubu genelinde yük dengelemeli olabilir.

## Paylaşılan kanal kurtarma

Aşağıdaki tabloda, paylaşılan kanal arızası tipleri ve her tipin nasıl işlendiği gösterilmektedir.

Arıza tipi:	Neler oluyor:
Kanal başlatıcı iletişim altsistemi hatası	İletişim altsistemine bağımlı kanallar kanal yeniden deneme işlemini girer ve yük dengeli başlatma komutuyla uygun bir kuyruk paylaşım grubu kanal başlatıcısında yeniden başlatılır.
Kanal başlatıcı hatası	Kanal başlatıcı başarısız olur, ancak ilişkili kuyruk yöneticisi etkin kalır. Kuyruk yöneticisi hatayı izler ve kurtarma işlemini başlatır.
Kuyruk yöneticisi hatası	Kuyruk yöneticisi başarısız oldu (ilişkili kanal başlatıcısı başarısız oldu). Kuyruk paylaşım grubundaki diğer kuyruk yöneticileri olayı izler ve eşdüzey kurtarma işlemini başlatır.
Paylaşılan durum hatası	Kanal durumu bilgileri Db2 içinde saklanır; bu nedenle, kanal durumu değişikliği gerçekleştiğinde Db2 bağlantısı kesilir. Çalışan kanallar, bu kaynaklara erişmeden çalışmaya devam edebilir. Db2 erişimi başarısız olursa, kanal yeniden dene 'ye girer.

Başarısız bir sistem adına paylaşılan kanal kurtarma işlemi, paylaşılan kanal durumunu almak için kurtarma işlemini yöneten sistemde Db2 bağlantısının bulunmasını gerektirir.

## Müşteri kanalları

İstemci bağlantı kanalları, belirli bir kuyruk yöneticisine bağlanmak yerine soysal arabirime bağlı kuyruk paylaşım gruplarındaki iletilerin yüksek kullanılabilirliğinden yararlanabilir. Daha fazla bilgi için bkz. [İstemci bağlantı kanalları.](#)

### İlgili kavramlar

[Paylaşılan kuyruklar ve kuyruk paylaşım grupları](#)

[“ayarlarIBM MQ for z/OS” sayfa 881](#)

IBM MQ for z/OS sisteminizi uyarlamak için adım adım bu konuyu kullanın.

[“Kümeler ve kuyruk paylaşım grupları” sayfa 982](#)

Paylaşılan kuyruğunuzu tek bir tanımlamada bir kümenin kullanımına açabilirsiniz. Bunu yapmak için, paylaşılan kuyruğu tanımladığınızda kümenin adını belirtirsiniz.

[“Kanallar ve diziselleştirme” sayfa 983](#)



Paylaşılan kuyruk eşdüzey kurtarma işlemi sırasında, paylaşılan kuyruklardaki iletileri işleyen ileti kanalı araçları, kuyruklara erişimlerini diziselleştirirler.

[Grup içi kuyruğa alma](#)

### İlgili görevler

[“Dağıtılmış kuyruğa alma yapılandırılıyor” sayfa 189](#)

Bu bölümde, kuyruk tanımlaması, kanal tanımlaması, tetikleme ve eşitleme noktası yordamları da içinde olmak üzere IBM MQ kuruluşları arasındaki iletişimle ilgili daha ayrıntılı bilgi sağlanır.

[“z/OS üzerinde diğer kuyruk yöneticileriyle iletişim kurulması” sayfa 954](#)

Bu bölümde, dağıtılmış kuyruğa alma işlevini kullanmaya başlamadan önce yapmanız gereken IBM MQ for z/OS hazırlıkları açıklanmaktadır.

### ► z/OS **Kuyruk paylaşım grupları için LU 6.2 ve TCP/IP dinleyicileri**

Grup LU 6.2 ve TCP/IP dinleyicileri, soysal adrese mantıksal olarak bağlı bir adresi dinler.

LU 6.2 dinleyicisi için, belirtilen LUGROUP, kuyruk paylaşım grubuyla ilişkilendirilmiş VTAM soysal kaynağıyla eşlenir. Bu teknolojinin ayarlanmasına ilişkin bir örnek için bkz. [“APPC/MVS kullanılarak z/OS için LU6.2 bağlantısı tanımlanması” sayfa 978.](#)

TCP/IP dinleyicisinde, belirtilen kapı soysal adrese aşağıdaki yollardan biriyle bağlanabilir:

- IBM Network Dispatchergibi bir ön uç yönleticisi için, gelen bağlantı istekleri yönlendiriciden kuyruk paylaşım grubunun üyelerine iletilir.
- TCP/IP Sistem Şebekesi Distribütörü için, çalışan ve Dağıtılmış DVIPA olarak ayarlanan belirli bir adreste dinleyen her bir dinleyiciye, gelen isteklerin bir oranı tahsis edilir. Bu teknolojinin kurulmasına ilişkin bir örnek için [Sysplex Distributor 'un Kullanılması](#) başlıklı konuya bakın.

### ► z/OS **Kuyruk paylaşım grupları için iletim kuyrukları ve tetikleme**

Paylaşılan iletim kuyruğu, iletileri kuyruk paylaşım grubundan hedefe taşınmadan önce saklamak için kullanılır.

Bu, paylaşılan bir kuyruktur ve kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından erişilebilir.

## Tetikleme

Tetiklenen bir paylaşılan kuyruk, karşılanmış bir tetikleyici koşulu için birden çok tetikleyici iletileri oluşturabilir. Tetiklenen paylaşılan kuyrukle ilişkilendirilmiş kuyruk paylaşım grubundaki bir kuyruk yöneticisinde tanımlı olan her yerel başlatma kuyruğu için bir tetikleyici ileti üretilir.

Dağıtılmış kuyruğa alma için, her kanal başlatıcısı, karşılanmış bir paylaşılan iletim kuyruğu tetikleyicisi koşulu için bir tetikleyici ileti alır. Ancak, yalnızca bir kanal başlatıcı tetiklenen başlangıç işler ve diğerleri güvenli bir şekilde başarısız olur. Tetiklenen kanal daha sonra yük dengeli başlatma ile başlatılır (bkz. [“Kuyruk paylaşım gruplarıyla QM2 için IBM MQ for z/OS hazırlanıyor” sayfa 979](#)) bu, QSG. TO. QM2kanalını başlatmak için tetiklenir. Paylaşılan bir iletim kuyruğu yaratmak için, aşağıdaki örnekte gösterildiği gibi IBM MQ komutlarını (MQSC) kullanın:

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') +
USAGE(XMITQ) QSGDISP(SHARED) +
CFSTRUCT(APPLICATION1) INITQ(SYSTEM.CHANNEL.INITQ) +
TRIGGER TRIGDATA(QSG.TO.QM2)
```

**Not:** Paylaşılan bir kuyruk tetikleme ve paylaşılan kuyruğu barındıran Coupling Facility bağlantısı için ayarlandıysa, bir tetikleme olayı oluşturulabilir ve başlatma kuyruğuna bir ileti konabilir. Tetikleme için özgün paylaşılan kuyruk ayarlarına hiçbir ileti konmadığında da bu durum oluşabilir. Bunun nedeni, [Liste Bildirim Vektörü](#)'nde belgelendiği gibi, IXLVECTR makrosunun bitlerin aşırı gösteriminden kaynaklanır.

### ► z/OS **Kuyruk paylaşım grupları için ileti kanalı araçları**

Bir kanal, kanal başlatıcısında ancak bu ada sahip bir kanala ilişkin bir kanal tanımına erişimi varsa başlatılabilir.

İleti kanalı aracısı, iletilerin gönderilmesini ve alınmasını denetleyen bir IBM MQ programıdır. İleti kanalı araçları iletileri bir kuyruk yöneticisinden diğerine taşır; bir kanalın her ucunda bir ileti kanalı aracısı vardır.

Bir kanal tanımlaması, bir kuyruk yöneticisine özel olarak tanımlanabilir ya da paylaşılan havuzda saklanabilir ve herhangi bir yerde (grup tanımlaması) kullanılabilir. Bu, grup tanımlı bir kanalın kuyruk paylaşım grubundaki herhangi bir kanal başlatıcısında kullanılabileceği anlamına gelir.

**Not:** Grup tanımlamasının özel kopyası değiştirilebilir ya da silinebilir.

Grup kanalı tanımlamaları yaratmak için, aşağıdaki örneklerde gösterildiği gibi IBM MQ komutlarını (MQSC) kullanın:

```
DEFINE CHL(QSG.TO.QM2) CHLTYPE(SDR) +  
TRPTYPE(TCP) CONNAME(QM2.MACH.IBM.COM) +  
XMITQ(QM2) QSGDISP(GROUP)
```

```
DEFINE CHL(QM2.TO.QSG) CHLTYPE(RCVR) TRPTYPE(TCP) +  
QSGDISP(GROUP)
```

Kuyruk paylaşım gruplarıyla dağıtılmış kuyruğa alma için kullanılan ileti kanalı araçlarına bakılacak iki perspektif vardır:

## Gelen

Gelen kanal, grup dinleyicisi aracılığıyla kuyruk yöneticisine bağlıysa, paylaşılan bir kanaldır. Soysal arabirim aracılığıyla kuyruk paylaşım grubuna bağlanır, daha sonra grup içindeki bir kuyruk yöneticisine yönlendirilir ya da belirli bir kuyruk yöneticisinin grup kapısını ya da grup dinleyicisi tarafından kullanılan mantıksal birim adını hedefler.

## Giden

Giden kanal, iletileri paylaşılan bir iletim kuyruğundan taşıyorsa, paylaşılan bir kanaldır. Örnek komutlarda, QSG.TO.QM2 gönderen kanalı, iletim kuyruğu QM2 QSGDISP (SHARED) ile tanımlandığından, paylaşılan bir kanaldır.

## **Kuyruk paylaşım grupları için eşitleme kuyruğu**

Paylaşılan kanalların SYSTEM.QSG.CHANNEL.SYNCQ.

Bu eşitleme kuyruğuna, kuyruk paylaşım grubunun herhangi bir üyesi erişebilir. (Özel kanallar özel eşitleme kuyruğunu kullanmaya devam eder. Bkz. [“z/OS üzerinde IBM MQ nesnelere tanımlama” sayfa 957](#)). Bu, iletişim altsisteminin, kanal başlatıcısının ya da kuyruk yöneticisinin başarısız olması durumunda, kanalın kuyruk paylaşım grubu içindeki farklı bir kuyruk yöneticisi ve kanal başlatıcısı yönetim ortamında yeniden başlatılabileceği anlamına gelir. Daha fazla bilgi için bkz. [“Kuyruk paylaşım gruplarıyla DQM için IBM MQ for z/OS hazırlanıyor” sayfa 979](#).

Kuyruk paylaşım gruplarına sahip DQM, SYSTEM.QSG.CHANNEL.SYNCQ. Bir grup dinleyicinin başarıyla başlatılabilmesi için bu kuyruk kullanılabilir olmalıdır.

Kuyruk kullanılmadığı için bir grup dinleyicisi başarısız olursa, kuyruk tanımlanabilir ve dinleyici kanal başlatıcısı geri dönüştürülmeden yeniden başlatılabilir. Paylaşılmayan kanallar etkilenmez.

Bu kuyruğu INDXTYPE (MSGID) komutunu kullanarak tanımladığınızdan emin olun. Bu tanım, kuyruktaki iletilere erişilme hızını artırır.

## **Kümeler ve kuyruk paylaşım grupları**

Paylaşılan kuyruğunuzu tek bir tanımlamada bir kümenin kullanımına açabilirsiniz. Bunu yapmak için, paylaşılan kuyruğu tanımladığınızda kümenin adını belirtirsiniz.

Ağdaki kullanıcılar, paylaşılan kuyruğu, kuyruk paylaşım grubu içindeki her kuyruk yöneticisi tarafından barındırıldığını görürler. (Paylaşılan kuyruk, kuyruk paylaşım grubu tarafından barındırılıyor olarak

duyurulmaz). İstemciler, iletileri aynı paylaşılan kuyruğa koymak için kuyruk paylaşım grubunun tüm üyeleriyle oturum başlatabilir.

Daha fazla bilgi için bkz. "[Kuyruk yöneticisi kümesinin yapılandırılması](#)" sayfa 283.

### **z/OS Kanallar ve diziselleştirme**

Paylaşılan kuyruk eşdüzey kurtarma işlemi sırasında, paylaşılan kuyruklardaki iletileri işleyen ileti kanalı araçları, kuyruklara erişimlerini diziselleştirirler.

Bir ileti kanalı aracısı bir ya da daha çok paylaşılan kuyrukte kesinleştirilmemiş iletilerle uğraşırken kuyruk paylaşım grubundaki bir kuyruk yöneticisi başarısız olursa, kanal ve ilişkili kanal başlatıcısı sona erer ve kuyruk yöneticisi için paylaşılan kuyruk eşini kurtarma işlemi gerçekleşir.

Paylaşılan kuyruk eşini kurtarma zamanuyumsuz bir etkinlik olduğundan, eş kanal kurtarma, paylaşılan kuyruk eşini kurtarma işlemi tamamlanmadan önce kanalı kuyruk paylaşım grubunun başka bir bölümünde eşzamanlı olarak yeniden başlatmayı deneyebilir. Bu olay gerçekleşirse, kesinleştirilen iletiler kurtarılmaya devam eden iletilerden önce işlenebilir. İletilerin bu şekilde işlenmediğinden emin olmak için, paylaşılan kuyruklardaki iletileri işleyen ileti kanalı araçları bu kuyruklara erişimlerini diziselleştirirler.

Paylaşılan kuyruk eşdüzey kurtarma işleminin devam ettiği bir kanalı başlatma girişimi bir hataya neden olabilir. Kurtarma işleminin devam ettiğini ve kanalın yeniden deneme durumuna getirildiğini belirten bir hata iletilisi gönderilir. Kuyruk yöneticisi eşdüzey kurtarma işlemi tamamlandığında, kanal bir sonraki yeniden deneme sırasında yeniden başlatılabilir.

RESOLVE, PING ya da DELETE girişimi aynı nedenle başarısız olabilir.

### **z/OS Kuyruk paylaşım gruplarını kullanarak IBM MQ for z/OS için iletişim ayarlama**

Bir dağıtım kuyruğa alma yönetim kanalı başlatıldığında, kanal tanımında belirtilen bağlantıyı kullanmayı dener. Bu girişimin başarılı olması için, bağlantının tanımlanması ve kullanılabilir olması gerekir.

Kullanılabilecek iki iletişim protokolü biçiminden birini seçin:

- [TCP](#)
- [LU 6.2 -APPC/MVS](#)

Örnek yapılandırma- [IBM MQ for z/OS kuyruk paylaşım gruplarını kullanmaseçeneğine başvurmanız yararlı olabilir.](#)

### **z/OS Kuyruk paylaşım grupları için TCP bağlantısı tanımlanması**

Bir kuyruk paylaşım grubuna ilişkin TCP bağlantısı tanımlamak için, gönderen ve alan uçtaki belirli özniteliklerin konfigürasyonunun tanımlanması gerekir.

TCP ' yi ayarlamaya ilişkin bilgi için bkz. "[z/OS üzerinde TCP bağlantısı tanımlanması](#)" sayfa 975.

## **Gönderme sonu**

Kuyruk paylaşım grubunuza bağlanmak için kanal tanımındaki bağlantı adı (CONNNAME) alanı, kuyruk paylaşım grubunuza ilişkin soysal arabirime ayarlanmalıdır (bkz. [Kuyruk paylaşım grupları](#) ). Daha fazla ayrıntı için bkz. [Sysplex Distributor 'un Kullanılması](#).

## **Kuyruk paylaşım grubu kullanılarak TCP üzerinden alma**

Paylaşılan kanal programlarının alınması, gönderen kanaldan gelen başlatma isteğine yanıt olarak başlatılır. Bunu yapmak için, gelen ağ isteklerini algılamak ve ilişkili kanalı başlatmak için bir dinleyici başlatılmalıdır. Bu dinleyici programını START LISTENER komutuyla, grubun gelen yok etme özelliğini kullanarak ya da işlemleri ve denetim panolarını kullanarak başlatabilirsiniz.

Kuyruk paylaşım grubundaki tüm grup dinleyicileri aynı kapıda dinliyor olmalıdır. Tek bir MVS görüntüsünde çalışan birden çok kanal başlatıcısı varsa, sanal IP adreslerini tanımlayabilir ve TCP dinleyici

programını, START LISTENER komutunda IPADDR belirleyerek yalnızca belirli bir adres ya da anasistem adını dinleyecek şekilde başlatabilirsiniz. (Daha fazla bilgi için bkz. [START LISTENER](#).)

## **z/OS** z/OS üzerinde LU 6.2 bağlantısı tanımlanması

Bir kuyruk paylaşım grubuna ilişkin LU 6.2 bağlantısı tanımlamak için, gönderme ve alma ucundaki bazı özniteliklerin yapılandırılması gerekir.

APPC/MVS ayarlarına ilişkin bilgi için [z/OS için iletişim kurulması](#) başlıklı konuya bakın.

## **APPC/MVS ile bağlantı kuruluyor (LU 6.2)**

Kuyruk paylaşım grubunuza bağlanmak için kanal tanımındaki bağlantı adı (CONNNAME) alanı, APPC/MVS için yan bilgi veri kümesinde belirtildiği gibi, simgesel hedef adına ayarlanmalıdır. Bu simgesel hedefte belirtilen ortak LU sosyal kaynak adı olmalıdır. Daha fazla ayrıntı için bkz. [Sosyal kaynakları kullanarak ağda kendinizi tanımlama](#).

## **Sosyal arabirim kullanılarak LU 6.2 üzerinde alma**

Paylaşılan MCA ' ların alınması, gönderen kanaldan gelen başlatma isteğine yanıt olarak başlatılır. Bunu yapmak için, gelen ağ isteklerini saptamak ve ilişkili kanalı başlatmak üzere bir grup dinleyici programı başlatılmalıdır. Dinleyici programı bir APPC/MVS sunucusudur. Bunu START LISTENER komutuyla, gelen yok etme grubunu kullanarak ya da işlemleri ve denetim panolarını kullanarak başlatabilirsiniz. Yan bilgi veri kümesinde tanımlı bir simgesel hedef adını kullanmak için LU adını belirtmelisiniz. Daha fazla ayrıntı için bkz. [Sosyal kaynakları kullanarak ağda kendinizi tanımlama](#).

## **z/OS** IBM MQ ' yi IMS ile kullanma

IBM MQ -IMS bağdaştırıcısı ve IBM MQ - IMS köprüsü, IBM MQ ' in IMS ile etkileşimde bulunmasına izin veren iki bileşendir.

IBM MQ ve IMS ' ı birlikte çalışacak şekilde yapılandırmak için aşağıdaki görevleri tamamlamanız gerekir:

- [“IMS bağdaştırıcısının ayarlanması” sayfa 984](#)
- [“IMS köprüsünün ayarlanması” sayfa 991](#)

### **İlgili kavramlar**

IBM MQ ve IMS

[“IBM MQ ' yi CICS ile kullanma” sayfa 992](#)

IBM MQ ürününü CICS ile kullanmak için IBM MQ CICS bağdaştırıcısını ve isteğe bağlı olarak IBM MQ CICS bridge bileşenlerini yapılandırmanız gerekir.

[“IMS içinde OTMA çıkışlarının kullanılması” sayfa 994](#)

IMS Open Transaction Manager Access ürününü IBM MQ for z/OS ile birlikte kullanmak istiyorsanız bu konuyu kullanın.

[IBM MQ for z/OS üzerinde IMS ve IMS köprü uygulamaları](#)

### **İlgili görevler**

[“z/OS üzerinde kuyruk yöneticilerini yapılandırma” sayfa 876](#)

IBM MQ for z/OS üzerinde kuyruk yöneticilerini yapılandırmak için bu yönergeleri kullanın.

### **İlgili başvurular**

[“Hizmeti Dil Ortamına ya da z/OS Callable Services 'e yükseltme ve uygulama” sayfa 992](#)

Gerçekleştirmeniz gereken işlemler, CALLIBS mi, LINK mi ve SMP/E sürümünüze göre değişir.

## **z/OS** IMS bağdaştırıcısının ayarlanması

IBM MQ ürününü IMS içinde kullanmak için IBM MQ - IMS bağdaştırıcısı gerekir (genellikle IMS bağdaştırıcısı olarak adlandırılır).

Bu konuda, IMS baędařtırıcısının IMS altsistemimizin kullanımına nasıl sunacaęınız anlatılır. Bir IMS altsistemini uyarlama konusunda bilgi sahibi deęilseniz, [IMS belgelerine](#) bakın.

IMS baędařtırıcısını IMS uygulamalarının kullanımına sunmak için ařaęıdaki adımları izleyin:

1. IMS dıř altsistem baęlantı olanaęını (ESAF) kullanarak dıř altsistem olarak IBM MQ deęerini IMS olarak tanımlayın.  
Bkz. "IBM MQ öęesini IMS olarak tanımlama" sayfa 986.
2. IBM MQ load library thlqual.SCSQAUTH öęesini IMS denetim bölgeniz için JCL 'de JOBLIB ya da STEPLIB birleřiminde ve IBM MQ ' e (Yařam Çevrimi Projesi Yönetimi (LPA) ya da baęlantı listesinde yoksa) baęlanan herhangi bir baęımlı bölgede ekleyin. JOBLIB ya da STEPLIB yetkiniz yoksa, bunu IMS modüllerini içeren kitaplıktan (genellikle IMS RESLIB) sonra DFSESL birleřtirmesine de ekleyin. thlqual.SCSQANLx öęesini de ekleyin (burada x, dil harfidir).  
DFSESL varsa, SCSQAUTH ve SCSQANLx 'in birleřtirmeye eklenmesi ya da LNKLIST ' e eklenmesi gerekir. JCL ' de STEPLIB ya da JOBLIB bititirmesine ekleme yeterli deęildir.
3. IBM MQ çevirici programı CSQQDEFV ' yi thlqual.SCSQASMS içinden bir kullanıcı kitaplığına kopyalayın.
4. Saęlanan program (CSQQDEFV), MQM1 IMS dil arabirimi simgesi (LIT) ile varsayılan olarak tanımlanan CSQ1 altsistem adını içerir. Bu adı sına ve kuruluş doęrulaması için saklayabilirsiniz.  
Üretim altsistemlerinde NAME=CSQ1 deęerini kendi altsistem adınıza çevirin ya da CSQ1komutunu kullanın. Gerektięinde ek altsistem tanımlamaları ekleyebilirsiniz. LIT ' lerle ilgili daha fazla bilgi için bkz. "IMS baędařtırıcısına IBM MQ kuyruk yöneticilerinin tanımlanması" sayfa 989 .
5. CSQQDEFV yükleme modülünü oluřturmak için programı derleyip düzenleyin. Yapıbirimi için, SYSLIB baęlamanızda thlqual.SCSQMACS kitaplığını ekleyin; RENTbaę düzenleme deęiřtirgesini kullanın. Bu, thlqual.SCSQPROC(CSQ4DEFV) içindeki örnek JCL ' de gösterilir.
6. IBM MQile baęlantı kuran herhangi bir baęımlı bölge için JCL ' de JOBLIB ya da STEPLIB birleřtirmesinde yarattıęınız CSQQDEFV modülünü içeren kullanıcı kitaplığını ekleyin. SCSQAUTH 'nin varsayılan bir yükleme modülü olduęu için, bu kitaplığı SCSQAUTH' nin önüne koyun. Bunu yapmazsanız, IMS' den 3041 olaęandıřı sonlanma kullanıcısı alırsınız.
7. IMS baędařtırıcısı beklenmeyen bir IBM MQ hatası saptarsa, z/OS SNAP dökümünü DD adı CSQSNAP ' ye verir ve uygulamaya MQRC\_UNEXPECTED\_ERROR neden kodunu verir. CSQSNAP DD deyimini IMS baęımlı bölgesi JCL ' de deęilse, döküm alınmaz. Bu durumda, JCL ' ye CSQSNAP DD deyimini ekleyebilir ve uygulamayı yeniden çalıřtırabilirdiniz. Ancak, bazı sorunlar kesintili olabileceęi için, oluřtuęunda hatanın nedenini yakalamak için CSQSNAP DD deyimini eklemeniz önerilir.
8. Dinamik IBM MQ çağrılarını kullanmak istiyorsanız ( [IBM MQ sınırlı kod öbeęini dinamik olarak çağırma](#) başlıklı konuda açıklanmaktadır), [Şekil 117 sayfa 986](#) içinde gösterildięi gibi dinamik sınırlı kod öbeęini oluřturun.
9. IMS tetikleyici izleme programını kullanmak istiyorsanız, IMS tetikleyici izleme uygulaması CSQQTRMN 'yi tanımlayın ve PSBGEN ile ACBGEN' yi gerçekleştirin. Bkz. "[IMS tetikleyici izleme programının ayarlanması](#)" sayfa 990.
10. OPERCMDS sınıfındaki kaynakları korumak için RACF kullanıyorsanız, IBM MQ kuyruk yöneticisi adres alanınızla iliřkili kullanıcı kimlięinin baęlanabileceęi herhangi bir IMS sisteminde MODIFY komutunu verme yetkisi olduęundan emin olun.

```

//DYNSTUB EXEC PGM=IEWL,PARM='RENT,REUS,MAP,XREF'
//SYSPRINT DD SYSOUT=*
//ACSQMOD DD DISP=SHR,DSN=thlqual.SCSQLOAD
//IMSLIB DD DISP=SHR,DSN=ims.reslib
//SYSLMOD DD DISP=SHR,DSN=private.load1
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,1)
//SYSLIN DD *
INCLUDE ACSQMOD(CSQQSTUB)
INCLUDE IMSLIB(DFSLI000)
ALIAS MQCONN,MQCONN,MQDISC MQI entry points
ALIAS MQGET,MQPUT,MQPUT1 MQI entry points
ALIAS MQOPEN,MQCLOSE MQI entry points
ALIAS MQBACK,MQCMIT MQI entry points
ALIAS CSQBBAK,CSQBBAK MQI entry points
ALIAS MQINQ,MQSET MQI entry points
ALIAS DFSPALI,PLITDLI IMS entry points
ALIAS DFSCOBOL,CBLTDLI IMS entry points
ALIAS DFSFOR,FORTDLI IMS entry points
ALIAS DFSASM,ASMTDLI IMS entry points
ALIAS DFSPASCL,PASTDLI IMS entry points
ALIAS DFHEI01,DFHEI1 IMS entry points
ALIAS DFSAIBLI,AIBTDLI IMS entry points
ALIAS DFSESS,DSNWLI,DSNHLI IMS entry points
ALIAS MQCRTMH,MQDLTMH,MQDLTMP IMS entry points
ALIAS MQINQMP,MQSETMP,MQMHBUFF,MQBUFMH IMS entry points
MODE AMODE(31),RMODE(24) Note RMODE setting
NAME CSQDYNS(R)
/*

```

<sup>1</sup>Specify the name of a library accessible to IMS applications that want to make dynamic calls to IBM MQ.

Şekil 117. Devingen çağrı sınırlı kod öbeğini düzenlemek için örnek JCL

## İlgili kavramlar

IBM MQ ve IMS

“IMS köprüsünün ayarlanması” sayfa 991

IBM MQ - IMS köprüsü, IBM MQ ' in IBM MQetkin olmayan var olan programlara ve hareketlere giriş ve çıkış yapmasını sağlayan isteğe bağlı bir bileşendir.

IBM MQ for z/OS üzerinde IMS ve IMS köprü uygulamaları

## z/OS IBM MQ ögesini IMS olarak tanımlama

IBM MQ , IMS denetim bölgesinde ve bu IBM MQ kuyruk yöneticisine erişen her bağımlı bölgede tanımlanmalıdır. Bunu yapmak için, IMSiçinde bir altsistem üyesi (SSM) oluşturmanız gerekir.PROCLIB kitaplığı ve ilgili IMS bölgelerine SSM ' yi tanımlayın.

## Altsistem üyesi girişinin IMSiçine yerleştirilmesi.PROCLIB

IMSiçindeki her SSM girdisi.PROCLIB, IMS bölgesinden farklı bir kuyruk yöneticisine bağlantı tanımlar.

Bir SSM ' yi adlamak için, IMS IMSCTRL makrosunun IMSID alanının değerini (bir ila dört alfasayısal karakter) siteniz tarafından tanımlanan herhangi bir adla (bir ila dört alfasayısal karakter) birleştirin.

Bir SSM tüm IMS bölgeleri tarafından paylaşılabilir ya da her bölge için belirli bir üye tanımlanabilir. Bu üye, dış altsistemlere yönelik bağlantı sayısı kadar giriş içerir. Her giriş 80 karakterlik bir kayıttır.

## Konumsal parametreler

Bu girdideki alanlar şunlardır:

SSN, LIT, ESMT, RTT, REO, CRC

Burada:

**SGN**

IBM MQ kuyruk yöneticisi adını belirtir. Gereklidir ve bir ile dört arasında karakter içermelidir.

**LIT.**

IMS' e sağlanan dil arabirim simgesini (LIT) belirtir. Bu alan gereklidir, değeri CSQQDEFV modülündeki bir alanla eşleşmelidir.

**ESMT**

Dış altsistem modülü çizelgesini (ESMT) belirtir. Bu tablo, IMStarafından hangi bağlantı modüllerinin yüklenmesi gerektiğini belirtir. CSQQESMT, bu alan için gerekli değerdir.

**RTT**

Bu seçenek IBM MQ tarafından desteklenmez.

**REO.**

Bir IMS uygulaması işletim dışı bir dış altsisteme başvurursa ya da kaynaklar iş parçacığı yaratma sırasında kullanılamıyorsa, kullanılacak bölge hatası seçeneğini (REO) belirtir. Bu alan isteğe bağlıdır ve tek bir karakter içerir:

**R**

Uygulamaya, IBM MQ hizmetleri isteğinin başarısız olduğunu gösteren bir dönüş kodu iletir.

**Q**

Uygulamayı U3051olağandışı bitiş koduyla sona erdirir, etkinliği son kesinleştirme noktasına kadar geriletir, hareketin PSTOP değerini gerçekleştirir ve giriş iletisini yeniden gönderir. Bu seçenek yalnızca IMS uygulaması çalışmayan bir dış altsisteme başvurduğunda ya da kaynaklar iş parçacığı yaratma sırasında kullanılamıyorsa geçerlidir.

IBM MQ tamamlanma ve neden kodları, IBM MQ sorun IBM MQ isteği işlerken oluşursa uygulamaya döndürülür; yani, bağdaştırıcı isteği IBM MQ' e geçirdikten sonra.

**A**

Uygulamayı U3047 olağandışı bitiş koduyla sonlandırır ve giriş iletisini atar. Bu seçenek yalnızca IMS uygulaması çalışmayan bir dış altsisteme başvurduğunda ya da kaynaklar iş parçacığı yaratma sırasında kullanılamıyorsa geçerlidir.

IBM MQ tamamlanma ve neden kodları, IBM MQ sorun IBM MQ isteği işlerken oluşursa uygulamaya döndürülür; yani, bağdaştırıcı isteği IBM MQ' e geçirdikten sonra.

**CRC**

Bu seçenek belirtilebilir, ancak IBM MQ tarafından kullanılmaz.

**Not:** Tüm konumsal parametrelerin tam ayrıntıları için [Dış altsistemlerin IMS için nasıl belirtildiği](#) başlıklı konuya bakın.

Örnek bir SSM girişi:

```
CSQ1, MQM1, CSQQESMT, , R,
```

Burada:

**CSQ1**

IBM MQ ile birlikte verilen varsayılan altsistem adı. Bunu kuruluşunuza uyacak şekilde değiştirebilirsiniz.

**MQM1**

CSQQDEFV ' de belirtilen varsayılan LIT.

**CSQQESMT**

Dış altsistem modülü adı. Bu değeri kullanmalısınız.

**Y**

REO seçeneği.



## Anahtar sözcük parametreleri

IBM MQ deęiřtirgeleri anahtar sözcük biçiminde belirtilebilir. SST deęiřtirgesinin deęeri DB2 ya da MQolabilir. IMS 14 'e MQ deęeri için destek eklenmiřtir. MQ kullanımı netlięe yardımcı olur ve IMS altsistem komutu artık SST deęerini içerir, ancak bunun dıřında önemli bir etkisi olmaz. Gerekiyorsa, DB2 deęeri kullanılabilir. Diđer parametreler Positional parameters(Konumsal parametreler) içinde açıklanmıřtır ve ařaęıdaki örnekte gösterilmiřtir:

```
SST=MQ,SSN=SYS3,LIT=MQM3,ESMT=CSQQESMT
```

Burada:

<b>SYS3</b>	Altsistem adı
<b>MQM3</b>	CSQQDEFV içinde saęlanan LIT
<b>CSQQESMT</b>	Dıř altsistem modülü adı

## SSM EXEC deęiřtirgesinin belirtilmesi

IMS denetim bölgesinin bařlatma yordamında SSM EXEC parametresini belirtin. Bu parametre, bir karakterden dört karakterlik altsistem üyesi adına (SSM) kadar olan bir karakteri belirtir.

IMS denetim bölgesi için SSM belirtirseniz, denetim bölgesi altında çalışan herhangi bir baęımlı bölge IMSadlı IBM MQ kuyruk yöneticisine baęlanabilir.SSM parametresi tarafından belirtilen PROCLIB üyesi. IMS.PROCLIB üye adı, SSM EXEC parametresinde belirlenen 1-4 karakterle bitiriřtirilmiř IMS tanıtıcısıdır (IMSID= xxxx). IMS tanıtıcısı, IMSCTRL oluřturma makrosunun IMSID parametresidir.

IMS , gerektięi kadar dıř altsistem baęlantısı tanımlamanızı saęlar. Farklı IBM MQ kuyruk yöneticileri için birden çok baęlantı tanımlanabilir. Tüm IBM MQ baęlantıları aynı z/OS sisteminde olmalıdır. Baęımlı bir bölge için, baęımlı bir bölge SSM belirtebilir ya da denetim bölgesi için belirtilen bölgeyi kullanabilirsiniz. Baęımlı bölge SSM 'de ve denetim bölgesi SSM' de farklı bölge hata seęenekleri (REO) belirtebilirsiniz. Çizelęe 67 sayfa 988 içinde SSM belirtimlerinin farklı olasılıkları gösterilmektedir.

Çizelęe 67. SSM belirtimleri seęenekleri			
Denetim bölgesi için SSM	Baęımlı bölge için SSM	Eylem	Yorumlar
Hayır	Hayır	Yok	Hiçbir dıř altsistem baęlanamaz.
Hayır	Evet	Yok	Hiçbir dıř altsistem baęlanamaz.
Evet	Hayır	Denetim bölgesi SSM ' yi kullan	Bölgede zamanlanan uygulamalar, denetim bölgesi SSM ' de tanımlanan dıř altsistemlere eriřebilir. Her bir eke iliřkin çıkıřlar ve denetim blokları denetim bölgesine ve baęımlı bölge adres alanlarına yüklenir.
Evet	Evet (boř)	Baęımlı bölge için SSM kullanılmıyor	Bu bölgede zamanlanan uygulamalar yalnızca DL/I veritabanlarına eriřebilir. Her bir eke iliřkin çıkıřlar ve denetim blokları denetim bölgesi adres alanına yüklenir.
Evet	Evet (boř deęil)	Denetim bölgesi SSM ile baęımlı bölge SSM ' yi denetleyin	Bu bölgede zamanlanan uygulamalar yalnızca her iki SSM ' de tanımlanan dıř altsistemlere eriřebilir. Her bir eke iliřkin çıkıřlar ve denetim blokları denetim bölgesine ve baęımlı bölge adres alanlarına yüklenir.

SSM belirtimi olasılıkları üst sınırını denetlemek için özel bir parametre yoktur.



## IMS baędařtırıcısının önceden yüklenmesi

IMS baędařtırıcısının performansı, IMStarafından önceden yüklendiyse artırılabilir. Ön yükleme, IMSDFSMP Lxx üyesi tarafından denetlenir. PROCLIB: bkz. " Daha fazla bilgi için IMS Yönetim Kılavuzu: Sistem ". Belirtilecek IBM MQ modülü adları şunlardır:

CSQACLST	CSQAMLST	CSQAPRH	CSQAVICM	CSQFSALM	CSQQDEFV
CSQQCONN	CSQQDISC	CSQQTERM	CSQQINIT	CSQQGERI	CSQQCMMT
CSQQESMT	CSQQPREP	CSQQTTHD	CSQQWAIT	CSQQNORM	CSQQSSOF
CSQQSSON	CSQFSTAB	CSQQRESV	CSQQSNOP	CSQQCMND	CSQQCVER
CSQQTMID	CSQQTRGI	CSQQCON2	CSQBPAPI	CSQBCRMH	CSQBAPPL

IBM MQ classes for JMS kullanımı hakkında daha fazla bilgi için bkz. [IMS içinde IBM MQ classes for JMS ' in kullanılması](#).

Geçerli IMS yayınları, IBM MQ modüllerinin yalnızca MPP, BMP, IFP, JMP ve JBP bölgelerindeki PDS-E biçim kitaplıklarından önceden yüklenmesini destekler. Başka bir IMS bölgesi tipi, PDS-E kitaplıklarından önyüklemeyi desteklemez. Başka bir bölge tipi için önyükleme gerekiyorsa, sağlanan IBM MQ modüllerinin bir PDS biçim kitaplığına kopyalanması gerekir.

### **z/OS** IMS baędařtırıcısına IBM MQ kuyruk yöneticilerinin tanımlanması

IBM MQ kuyruk yöneticilerinin adları ve bunlara karşılık gelen dil arabirimi belirteçleri (LIT), kuyruk yöneticisi tanımlama çizelgesinde tanımlanmalıdır.

CSQQDEFV yükleme modülünü yaratmak için, belirtilen CSQQDEFX makrosunu kullanın. [Şekil 118 sayfa 989](#) içinde bu çevirici makronun sözdizimi gösterilmektedir.

```
CSQQDEFX TYPE=ENTRY | DEFAULT , NAME=qmgr-name , LIT=token
or
CSQQDEFX TYPE=END
```

Şekil 118. CSQQDEFX makro sözdizimi

### Parametreler

#### **TYPE=ENTRY | VARSAYILAN**

TYPE=ENTRY ya da TYPE=DEFAULT değerini aşağıdaki gibi belirleyin:

#### **TYPE=ENTRY**

Bir IMS uygulaması tarafından kullanılacak bir IBM MQ kuyruk yöneticisini tanımlayan bir çizelge girişinin oluşturulacağını belirtir. Bu ilk girişse, çizelge üstbilgisi, CSQQDEFV CSECT deyimini de içinde olmak üzere oluşturulur.

#### **TYPE=DEFAULT**

TYPE=ENTRY için. Belirtilen kuyruk yöneticisi, MQCONN ya da MQCONNX boş olan bir adı belirttiğinde kullanılacak varsayılan kuyruk yöneticisidir. Çizelgede böyle bir giriş olmalıdır.

#### **NAME= qmgr-name**

MQCONN ya da MQCONNX ile belirtildiği şekilde, kuyruk yöneticisinin adını belirtir.

#### **LIT = simge**

IMS ' in kuyruk yöneticisini tanıtmak için kullandığı dil arabirimi simgesinin (LIT) adını belirtir.

MQCONN ya da MQCONNX çağrısı, *ad* giriş değiştirgesini ve *hconn* çıkış değiştirgesini ad etiketiyle ve dolayısıyla CSQQDEFV girişindeki LIT ile ilişkilendirir. *hconn* değiştirgesini aktaran diğer IBM MQ çağrıları, aynı LIT ile IMS SSM PROCLIB üyesinde tanımlanan IBM MQ kuyruk yöneticisine

çağrılarını yönlendirmek için MQCONN ya da MQCONNX çağrısında tanımlanan CSQQDEFV girişindeki LIT ' yi kullanır.

Özette, MQCONN ya da MQCONNX çağrısındaki **name** değiştirgesi CSQQDEFV 'deki bir LIT' yi tanıtır ve SSM üyesindeki aynı LIT bir IBM MQ kuyruk yöneticisini tanıtır. ( MQCONN çağrıyla ilgili bilgi için bkz. [MQCONN-Connect kuyruk yöneticisi](#). MQCONNX çağrıyla ilgili bilgi için bkz. [MQCONNX-Connect kuyruk yöneticisi \(genişletilmiş\).](#))

#### TYPE=SON

Çizelgenin tamamlandığını belirtir. Bu parametre atlanırsa, TYPE=ENTRY varsayılr.

#### CSQQDEFX makrosunun kullanılması

Şekil 119 sayfa 990 , bir kuyruk yöneticisi tanımlaması çizelgesinin genel yerleşim düzenini gösterir.

```
CSQQDEFX NAME=subsystem1,LIT=token1
CSQQDEFX NAME=subsystem2,LIT=token2,TYPE=DEFAULT
CSQQDEFX NAME=subsystem3,LIT=token3
...
CSQQDEFX NAME=subsystemN,LIT=tokenN
CSQQDEFX TYPE=END
END
```

Şekil 119. Kuyruk yöneticisi tanımlaması çizelgesinin yerleşim düzeni

#### **z/OS** IMS tetikleyici izleme programının ayarlanması

Bir IBM MQ başlatma kuyruğunu izlemek için IMS toplu iş yönelimli bir program ayarlayabilirsiniz.

thlqual.SCSQPROC kitaplığındaki CSQQTAPL modelini kullanarak IMS uygulamasını tanımlayın ( [CSQQTRMN için örnek hareket tanımlaması](#) konusuna bakın).

thlqual.SCSQPROC kitaplığındaki CSQQTPSB modelini kullanarak PSB ve ACB ' yi oluşturun ( [CSQQTRMN için örnek PSB tanımı](#) konusuna bakın).

```
* This is the application definition *
* for the IMS Trigger Monitor BMP *

APPLCTN PSB=CSQQTRMN,
PGMTYPE=BATCH,
SCHDTYP=PARALLEL
```

Şekil 120. CSQQTRMN için örnek hareket tanımlaması

```
PCB TYPE=TP,          ALTPCB for transaction messages
MODIFY=YES,          To "triggered" IMS transaction
PCBNAME=CSQQTRMN
PCB TYPE=TP,          ALTPCB for diagnostic messages
MODIFY=YES,          To LTERM specified or "MASTER"
PCBNAME=CSQQTRMG,
EXPRESS=YES
PSBGEN LANG=ASSEM,
PSBNAME=CSQQTRMN,   Runs program CSQQTRMN
CMPAT=YES
```

Şekil 121. CSQQTRMN için örnek PSB tanımı

IMS tetikleyici izleme programının başlatılması ve durdurulmasına ilişkin ek bilgi için [IMS tetikleyici izleme programının denetlenmesibaşlıklı konuya](#) bakın.

IBM MQ - IMS köprüsü, IBM MQ ' in IBM MQ etkin olmayan var olan programlara ve hareketlere giriş ve çıkış yapmasını sağlayan isteğe bağlı bir bileşendir.

Bu konuda, IBM MQ - IMS köprüsünü özelleştirmek için ne yapmanız gerektiği açıklanmaktadır.

**IBM MQ için XCF ve OTMA parametrelerini tanımlayın.**

Bu adım, IBM MQ sisteminize ve diğer OTMA değiştirgelerine ilişkin XCF grup ve üye adlarını tanımlar. IBM MQ ve IMS aynı XCF grubuna ait olmalıdır. Sistem değiştirgesi yükleme modülünde bu değiştirgeleri uyarlamak için CSQ6SYSP makrosunun OTMACON anahtar sözcüğünü kullanın.

Ek bilgi için [CSQ6SYSP' nin Kullanılması](#) başlıklı konuya bakın.

**IMS için XCF ve OTMA değiştirgelerini tanımlayın.**

Bu adım, IMS sistemine ilişkin XCF grup ve üye adlarını tanımlar. IMS ve IBM MQ aynı XCF grubuna ait olmalıdır.

IMS parametre listenize aşağıdaki parametreleri ekleyin: JCL ' de ya da IMS PROCLIB içindeki DFSPBxxx üyesinde:

**OTMA=Y**

Bu, IMS başlatıldığında OTMA ' yı otomatik olarak başlatır. (İsteğe bağlıdır; OTMA=N belirtirseniz, IMS komutunu /START OTMA komutunu vererek de OTMA ' yı başlatabilirsiniz.)

**GRAD=**

Bu değiştirge XCF grubu adını verir.

Bu, depolama sınıfı tanımlamasında (sonraki adıma bakın) ve CSQ6SYSP makrosunun OTMACON anahtar sözcüğünün **Group** değiştirgesinde belirtilen grup adıyla aynıdır.

**OTMANM=**

Bu değiştirge, IMS sisteminin XCF üyesi adını verir.

Bu, depolama sınıfı tanımlamasında belirtilen üye adıyla aynıdır (sonraki adıma bakın).

**IBM MQ ' a IMS sisteminin XCF grubunu ve üye adını belirtin.**

Bu, bir kuyruğun depolama sınıfı tarafından belirlenir. IBM MQ - IMS köprüsünden ileti göndermek istiyorsanız, kuyruk için depolama sınıfını tanımlarken bunu belirtmeniz gerekir. Depolama sınıfında, XCF grubunu ve hedef IMS sisteminin üye adını tanımlamanız gerekir. Bunu yapmak için IBM MQ işlemlerini ve denetim panolarını kullanın ya da IBM MQ komutlarını [Programlanır Komut Biçimine Giriş](#) başlıklı konuda açıklandığı gibi kullanın.

**Gereken güvenliği ayarlayın.**

/SECURE OTMA IMS komutu, OTMA aracılığıyla IMS ' e bağlanan **her** IBM MQ kuyruk yöneticisine uygulanacak güvenlik düzeyini belirler. Ek bilgi için bkz. [IMS ile IBM MQ kullanımı için güvenlikle ilgili önemli noktalar](#) .

**Aynı kuyruk yöneticisine ek IMS bağlantısı eklenmesi**

Aynı kuyruk yöneticisine IMS bağlantısı eklemek için yeni IMS' yi gösterecek ikinci bir depolama sınıfı (STGCLASS) tanımlamanız gerekir; ek bilgi için [DEFINE STGCLASS](#) başlıklı konuya bakın.

**Önemli:**

- Bir yerel kuyruk iki depolama sınıfını gösteremez.
- Bir depolama sınıfı iki IMS köprüsüne işaret edemez.
- IBM MQ ve IMS aynı XCF grubuna ait olmalıdır. Sistem değiştirgesi yükleme modülünde bu değiştirgeleri uyarlamak için CSQ6SYSP makrosunun OTMACON anahtar sözcüğünü kullanın.

Ek bilgi için [CSQ6SYSP' nin Kullanılması](#) başlıklı konuya bakın.

**İlgili kavramlar**

[IBM MQ ve IMS](#)

[“IMS bağdaştırıcısının ayarlanması” sayfa 984](#)

IBM MQ ürününü IMS içinde kullanmak için IBM MQ - IMS bağdaştırıcısı gerekir (genellikle IMS bağdaştırıcısı olarak adlandırılır).

IBM MQ for z/OS üzerinde IMS ve IMS köprü uygulamaları

## z/OS IBM MQ ' yi CICS ile kullanma

IBM MQ ürününü CICS ile kullanmak için IBM MQ CICS bağdaştırıcısını ve isteğe bağlı olarak IBM MQ CICS bridge bileşenlerini yapılandırmanız gerekir.

IBM MQ CICS bağdaştırıcısının ve IBM MQ CICS bridge bileşenlerinin yapılandırılmasıyla ilgili daha fazla bilgi için CICS belgelerinin [MQ bağlantılarının yapılandırılması](#) bölümüne bakın.

### İlgili kavramlar

IBM MQ ve CICS

[“IBM MQ ' yi IMS ile kullanma” sayfa 984](#)

IBM MQ -IMS bağdaştırıcısı ve IBM MQ - IMS köprüsü, IBM MQ ' in IMS ile etkileşimde bulunmasına izin veren iki bileşendir.

### İlgili başvurular

[“Hizmeti Dil Ortamına ya da z/OS Callable Services 'e yükseltme ve uygulama” sayfa 992](#)

Gerçekleştirmeniz gereken işlemler, CALLLIBS mi, LINK mi ve SMP/E sürümünüze göre değişir.

## z/OS Hizmeti Dil Ortamına ya da z/OS Callable Services 'e yükseltme ve uygulama

Gerçekleştirmeniz gereken işlemler, CALLLIBS mi, LINK mi ve SMP/E sürümünüze göre değişir.

Aşağıdaki çizelgede, aşağıdaki ürünlerin düzeyini yükseltir ya da bu ürünlere hizmet uygularsanız, IBM MQ for z/OS ' e ne yapmanız gerektiği gösterilmektedir:

- Dil Ortamı
- z/OS Çağrılabilir Hizmetler (örneğin, APPC ve RRS)

<i>Çizelge 68. Hizmet uygulandı ya da ürün yeni bir yayına yükseltildi</i>		
Ürün	CALLLIBS ve SMP/E V3r2 ya da üstü kullanılıyorsa işlem	LINK kullanılıyorsa işlem
Dil Ortamı	<p><b>Not: Dil Ortamı ve Çağrılabilir hizmetler için ayrı işler çalıştırmanız gerekmez. Bir iş yeterli olacak.</b></p> <ol style="list-style-type: none"><li>1. SMP/E işinizdeki Sınırı Hedef bölgeye ayarlayın.</li><li>2. SMP/CNTL kartında LINK LMODS CALLLIBS değerini belirleyin. CHECK, RETRY (YES) ve RC gibi diğer parametreleri de belirtebilirsiniz. Daha fazla bilgi için bkz. <a href="#">z/OS SMP/E Komutları</a> .</li><li>3. SMP/E işini çalıştırın.</li></ol>	SMP/E bölgelerinin otomatik yeniden bağlama için ayarlanması ve CSQ8SLDQ işinin çalıştırılması için herhangi bir işlem yapılması gerekmez.

Çizelge 68. Hizmet uygulandı ya da ürün yeni bir yayına yükseltildi (devamı var)		
<b>Ürün</b>	<b>CALLLIBS ve SMP/E V3r2 ya da üstü kullanılıyorsa işlem</b> <b>Not: Dil Ortamı ve Çağrılabilir hizmetler için ayrı işler çalıştırmanız gerekmez. Bir iş yeterli olacak.</b>	<b>LINK kullanılıyorsa işlem</b>
Çağrılabilir Hizmetler	<ol style="list-style-type: none"> <li>1. SMP/E işinizdeki Sınırı Hedef bölgeye ayarlayın.</li> <li>2. SMP/CNTL kartında LINK LMODS CALLLIBS değerini belirleyin. CHECK, RETRY (YES) ve RC gibi diğer parametreleri de belirtebilirsiniz. Daha fazla bilgi için bkz. <u>z/OS SMP/E Komutları</u> .</li> <li>3. SMP/E işini çalıştırın.</li> </ol>	SMP/E bölgelerinin otomatik yeniden bağlama için ayarlanması ve CSQ8SLDQ işinin çalıştırılması için herhangi bir işlem yapılması gerekmez.

Çizelge 69. Ürünlerden biri, yeni bir SMP/E ortamında ve kitaplıklarında yeni bir yayın düzeyine güncellendi		
<b>Ürün</b>	<b>CALLLIBS ve SMP/E V3r2 ya da üstü kullanılıyorsa işlem</b> <b>Not: Dil Ortamı ve Çağrılabilir hizmetler için üç ayrı iş çalıştırmanız gerekmez. Her iki ürün için de bir iş yeterli olacaktır.</b>	<b>LINK kullanılıyorsa işlem</b>
Dil Ortamı	<ol style="list-style-type: none"> <li>1. SCEELKED ve SCEESPC için DDDEF ' leri yeni kitaplığı gösterecek şekilde değiştirin.</li> <li>2. SMP/E işinizdeki Sınırı Hedef bölgeye ayarlayın.</li> <li>3. SMP/CNTL kartında LINK LMODS CALLLIBS değerini belirleyin. CHECK, RETRY (YES) ve RC gibi diğer parametreleri de belirtebilirsiniz. Daha fazla bilgi için bkz. <u>z/OS SMP/E Komutları</u> .</li> <li>4. SMP/E işini çalıştırın.</li> </ol>	<ol style="list-style-type: none"> <li>1. IBM MQ for z/OS hedef bölgesinde aşağıdaki LMOD girdilerine ilişkin XZMOD alt girdilerini silin: CMQXDCST, CMQXRCTL, CMQXSUPR, CSQCBE00, CSQCBE30, CSQCBP00, CSQCBP10, CSQCBR00, CSQUCVX, CSQUDLQH, CSQVXPCB, CSQVXSPT, CSQXDCST, CSQXRCTL, CSQXSUPR, CSQXTCMI, CSQXTCP, CSQXTNSV, CSQ7DRPS, IMQB23IC, IMQB23IM, IMQB23IR, IMQS23IC, IMQS23IM, IMQS23IR</li> <li>2. IBM MQ bölgeleri ve Dil Ortamı bölgeleri arasında uygun ZONEINDEX 'leri ayarlayın.</li> <li>3. CSQ8SLDQ değerini, LINK komutlarının FROMZONE parametresindeki yeni bölgeye gönderme yapmak üzere uyarlar. CSQ8SLDQ , SCSQINST kitaplığında bulunabilir.</li> <li>4. CSQ8SLDQkomutunu çalıştırın.</li> </ol>

Çizelge 69. Ürünlerden biri, yeni bir SMP/E ortamında ve kitaplıklarında yeni bir yayın düzeyine güncellendi (devamı var)

Ürün	CALLLIBS ve SMP/E V3r2 ya da üstü kullanılıyorsa işlem	LINK kullanılıyorsa işlem
Çağrılabilir hizmetler	<p><b>Not: Dil Ortamı ve Çağrılabilir hizmetler için üç ayrı iş çalıştırmanız gerekmez. Her iki ürün için de bir iş yeterli olacaktır.</b></p> <ol style="list-style-type: none"><li>1. CSSLIB için DDDEF ' yi yeni kitaplığı gösterecek şekilde değiştirin</li><li>2. SMP/E işinizdeki Sınırı Hedef bölgeye ayarlayın.</li><li>3. SMPCTL kartında LINK LMODS CALLLIBS değerini belirleyin. CHECK, RETRY (YES) ve RC gibi diğer parametreleri de belirtebilirsiniz. Daha fazla bilgi için bkz. <a href="#">z/OS SMP/E Komutları</a> .</li><li>4. SMP/E işini çalıştırın.</li></ol>	<ol style="list-style-type: none"><li>1. IBM MQ for z/OS hedef bölgesinde aşağıdaki LMOD girdilerine ilişkin XZMOD alt girdilerini silin: CMQXRCTL, CMQXSUPR, CSQBSRV, CSQILPLM, CSQXJST, CSQXRCTL, CSQXSUPR, CSQ3AMGP, CSQ3EPX, CSQ3REPL</li><li>2. IBM MQ bölgeleri ve Çağrılabilir Hizmetler bölgeleri arasında uygun ZONEINDEX 'leri ayarlayın.</li><li>3. CSQ8SLDQ değerini, LINK komutlarının FROMZONE parametresindeki yeni bölgeye gönderme yapmak üzere uyarlar. CSQ8SLDQ , SCSQINST kitaplığında bulunabilir.</li><li>4. CSQ8SLDQ komutunu çalıştırın.</li></ol>

CALLLIBS kullanırken modülleri yeniden bağlamak için bir iş örneği için bkz. "[LINK CALLLIBS işinin çalıştırılması](#)" sayfa 994.

## z/OS LINK CALLLIBS işinin çalıştırılması

CALLLIBS kullanılırken modülleri yeniden bağlamak için örnek bir iş.

Aşağıda, bir SMP/E V3r2 sisteminde CALLLIB ' ler kullanıldığında modülleri yeniden bağlamak için kullanılan bir iş örneği verilmiştir. IBM MQ for z/OS için bir JOBCARD ve SMP/E CSI veri kümesi adı sağlamanız gerekir.

```
//*****  
//* RUN LINK CALLLIBS.  
//*****  
//CALLLIBS EXEC PGM=GIMSMP,REGION=4096K  
//SMPCSI DD DSN=your.csi  
//      DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SMPCNTL DD *  
SET BDY(TZONE).  
LINK LMODS CALLLIBS .  
/*
```

Şekil 122. Örnek SMP/E LINK CALLLIBS işi

## z/OS IMS içinde OTMA çıkışlarının kullanılması

IMS Open Transaction Manager Access ürününü IBM MQ for z/OS ile birlikte kullanmak istiyorsanız bu konuyu kullanın.

Bir IMS hareketinden IBM MQ' e çıkış göndermek istiyorsanız ve bu işlem IBM MQ'iden kaynaklanmamışsa, bir ya da daha çok IMS OTMA çıkışını kodlemeniz gerekir.

Benzer şekilde, çıkışı OTMA olmayan bir hedefe göndermek istiyorsanız ve işlem IBM MQ kaynaklı ise, bir ya da daha çok IMS OTMA çıkışını kodlamanız gerekir.

IMS ile IBM MQ arasında işlemeyi özelleştirmenizi sağlamak için IMS içinde aşağıdaki çıkışlar kullanılabilir:

- OTMA ön yönlendirme çıkışı
- Hedef çözünürlüklü kullanıcı (DRU) çıkışı

## OTMA çıkış adları

DFSYPRX0 ön yönlendirme çıkışını adlandırmanız gerekir. DRU çıkışını, IMS içinde önceden var olan bir modül adıyla çağırmadığı sürece herhangi bir ad verebilirsiniz.

### Hedef çözünürlük kullanıcı çıkışı adının belirtilmesi

IMStarafından çalıştırılacak OTMA DRU çıkışının adını belirtmek için CSQ6SYSP makrosunun OTMACON anahtar sözcüğünün *Druexit* değerini kullanabilirsiniz.

Nesne tanımlamasını basitleştirmek için DRU0xxxxadlandırma kuralını benimsemeyi düşünün; burada xxxx, IBM MQ kuyruk yöneticinizin adıdır.

OTMACON parametresinde bir DRU çıkışının adını belirlemezseniz, varsayılan değer DFSYDRU0 olur. Daha fazla bilgi için bkz. [DFSYDRU0](#).

### IMS hedefi için adlandırma kuralı

IMS programınızdan çıktıyı gönderdiğiniz hedef için bir adlandırma kuralı gerekir. Bu, IMS uygulamanızın CHNG çağrısında ayarlanan ya da IMS PSB ' de önceden ayarlanan hedeftir.

## OTMA çıkışı için örnek senaryo

IMS için bir ön yönlendirme çıkışı ve hedef yönlendirme çıkışı örneği için aşağıdaki konuları kullanın:

- “Ön yönlendirme çıkışı DFSYPRX0” sayfa 995
- “Hedef çözünürlük kullanıcı çıkışı” sayfa 997

Tanımlamayı kolaylaştırmak için, OTMA hedef adını IBM MQ kuyruk yöneticisi adıyla benzer yapın; örneğin, IBM MQ kuyruk yöneticisi adı yinelenmiştir. Bu durumda, IBM MQ kuyruk yöneticisi adı " **VCPE** ", CHNG çağrısıyla belirlenen hedef: " **VCPEVCPE** ".

### İlgili kavramlar

IBM MQ ve IMS

“IBM MQ ' yi IMS ile kullanma” sayfa 984

IBM MQ -IMS bağdaştırıcısı ve IBM MQ - IMS köprüsü, IBM MQ ' in IMS ile etkileşimde bulunmasına izin veren iki bileşendir.

[IBM MQ for z/OS üzerinde IMS ve IMS köprü uygulamaları](#)

## Ön yönlendirme çıkışı DFSYPRX0

Bu konu, IMS içinde OTMA için örnek bir ön yönlendirme çıkışı içerir.

Önce bir DFSYPRX0 ön yönlendirme çıkışını kodlamanız gerekir. Bu yordama IMStarafından geçirilen parametreler için [OTMA Destination Resolution kullanıcı çıkışı \(DFSYPRX0 ve diğer OTMAYPRX tip çıkışları\)](#) konusuna bakın.

Bu çıkış, iletinin bilinen bir OTMA hedefi (örneğimizdeki VCPEVCPE) için tasarlanıp tasarlanmadığını sınar. Bu durumda, çıkışın iletiyi gönderen işlemin OTMA ' dan kaynaklanıp kaynaklanmadığını denetlemesi gerekir. İleti OTMA ' dan kaynaklandıysa, bir OTMA üstbilgisine sahip olacaktır; bu nedenle, 15 numaralı kayıt dosyası sıfır olarak ayarlanmış olarak DFSYPRX0 ' dan çıkmanız gerekir.

- İletiyi gönderen hareket OTMA ' da kaynaklanmamışsa, istemci adını geçerli bir OTMA istemcisi olarak ayarlamanız gerekir. Bu, iletiyi göndermek istediğiniz IBM MQ kuyruk yöneticisinin XCF üyesinin adıdır. İstemci adınızı ( CSQ6SYSP makrosunun OTMACON parametresinde) kuyruk yöneticisi adına ayarlamalısınız. Bu varsayılandır. Daha sonra, DFSYPRX0 ayar kaydından 15-4 arasında çıkmanız gerekir.
- İletiyi gönderen hareket OTMA ' dan kaynaklandıysa ve hedef OTMA değilse, 15-8 arasında bir kayıt ayarlamalı ve çıkmalısınız.
- Diğer tüm durumlarda, kayıt 15 'i sıfıra ayarlamanız gerekir.

OTMA istemcisi adını IMStarafından bilinmeyen bir ada ayarlarsanız, uygulamanız CHNG ya da ISRT çağırısı A1 durum kodunu döndürür.

Birden çok IBM MQ kuyruk yöneticisiyle iletişim kuran bir IMS sistemi için, her IBM MQ kuyruk yöneticisi için mantığı yinelemeniz gerekir.

Örnek çevirici kodu Şekil 123 sayfa 996 içinde gösterilmektedir:

```

TITLE 'DFSYPRX0: OTMA PRE-ROUTING USER EXIT'
DFSYPRX0 CSECT
DFSYPRX0 AMODE 31
DFSYPRX0 RMODE ANY
*
SAVE (14,12),,DFSYPRX0&SYSDATE&SYSTEMTIME
SPACE 2
LR R12,R15          MODULE ADDRESSABILITY
USING DFSYPRX0,R12
*
L R2,12(,R1)        R2 -> OTMA PREROUTE PARMs
*
LA R3,48(,R2)       R3 AT ORIGINAL OTMA CLIENT (IF ANY)
CLC 0(16,R3),=XL16'00' OTMA ORIG?
BNE OTMAIN          YES, GO TO THAT CODE
*
NOOTMAIN DS 0H      NOT OTMA INPUT
LA R5,8(,R2)        R5 IS AT THE DESTINATION NAME
CLC 0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE EXIT0           NO, NORMAL PROCESSING
*
L R4,80(,R2)        R4 AT ADDR OF OTMA CLIENT
MVC 0(16,R4),=CL16'VCPE' CLIENT OVERRIDE
B EXIT4             AND EXIT
*
OTMAIN DS 0H        OTMA INPUT
LA R5,8(,R2)        R5 IS AT THE DESTINATION NAME
CLC 0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE EXIT8           NO, NORMAL PROCESSING

*
EXIT0 DS 0H
LA R15,0            RC = 0
B BYEBYE
*
EXIT4 DS 0H
LA R15,4            RC = 4
B BYEBYE
*
EXIT8 DS 0H
LA R15,8            RC = 8
B BYEBYE
*
BYEBYE DS 0H
RETURN (14,12),,RC=(15) RETURN WITH RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
END

```

Şekil 123. OTMA ön yönlendirme çıkış çevirici örneği



## Hedef çözünürlük kullanıcı çıkışı

Bu konu, IMSiçin örnek bir hedef çözümleme kullanıcı çıkışı içerir.

DFSYPX0' da 15-4 kayıt ayarladıysanız ya da hareketin kaynağı OTMA **ise ve** Kayıt 15 'i sıfıra ayarladıysanız, DRU çıkışınız çağrılır. Bu örnekte, DRU çıkış adı DRU0VCPE' dir.

DRU çıkışı, hedefin VCPEVCPE olup olmadığını denetler. Eğer öyleyse, OTMA kullanıcı verilerini (OTMA önekinde) aşağıdaki gibi ayarlar:

### Görelî Konum

#### OTMA kullanıcı verileri

#### (ondalık)

**0**

OTMA kullanıcı verileri uzunluğu (bu örnekte, 334)

**2**

Mqmd

**326**

Yanıtın gönderileceği biçim

Bu görelî konumlar, IBM MQ - IMS köprüsünün bu bilgileri bulmayı beklediği yerdir.

DRU çıkışı mümkün olduğunca basit olmalıdır. Bu örnekte, belirli bir IBM MQ kuyruk yöneticisi için IMS ' den kaynaklanan tüm iletiler aynı IBM MQ kuyruğuna konmuştur.

İletin kalıcı olması gerekiyorsa, IMS eşitlenmiş bir işlem veri bağlantısı kullanmalıdır. Bunu yapmak için, DRU çıkışı OUTPUT işaretini ayarlamalıdır. Daha fazla bilgi için [IBM MQiçin eşitlenmiş veri bağlantılarının belirtilmesi](#) başlıklı konuya bakın.

Bu kuyruğu işlemek için bir IBM MQ uygulaması yazın ve her iletiyi hedefine yönlendirmek için MQMD yapısından, MQIIH yapısından (varsa) ya da kullanıcı verilerinden bilgileri kullanın.

[Şekil 124 sayfa 998içinde](#) örnek bir çevirici DRU çıkışı gösterilmektedir.

```

TITLE 'DRU0VCPE: OTMA DESTINATION RESOLUTION USER EXIT'
DRU0VCPE CSECT
DRU0VCPE AMODE 31
DRU0VCPE RMODE ANY
*
SAVE (14,12),,DRU0VCPE&SYSDATE&SYSTEME
SPACE 2
LR R12,R15          MODULE ADDRESSABILITY
USING DRU0VCPE,R12
*
L R2,12(,R1)        R2 -> OTMA DRU PARMS
*
L R5,88(,R2)        R5 ADDR OF OTMA USERDATA
LA R6,2(,R5)        R6 ADDR OF MQMD
USING MQMD,R6       AS A BASE
*
LA R4,MQMD_LENGTH+10 SET THE OTMA USERDATA LEN
STH R4,0(,R5)       = LL + MQMD + 8
*
MVI 0(R6),X'00'     CLEAR REST OF USERDATA
MVC 1(255,R6),0(R6) ..NULL FIRST BYTE
MVC 256(MQMD_LENGTH-256+8,R6),255(R6) ..AND PROPAGATE IT
*
VCPE DS 0H
CLC 44(16,R2),=CL16'VCPE' IS DESTINATION VCPE?
BNE EXIT4           NO, THEN DEST IS NON-OTMA
MVC MQMD_REPLYTOQ,=CL48'IMS.BRIDGE.UNSOLICITED.QUEUE'
MVC MQMD_REPLYTOQMGR,=CL48'VCPE' SET QNAME AND QMGRNAME
MVC MQMD_FORMAT,MQFMT_IMS SET MQMD FORMAT NAME
MVC MQMD_LENGTH(8,R6),MQFMT_IMS_VAR_STRING
*
B EXIT0             SET REPLYTO FORMAT NAME
*
EXIT0 DS 0H
LA R15,0            SET RC TO OTMA PROCESS
B BYEBYE           AND EXIT
*
EXIT4 DS 0H
LA R15,4            SET RC TO NON-OTMA
B BYEBYE           AND EXIT
*
BYEBYE DS 0H
RETURN (14,12),,RC=(15) RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
CMQA EQUONLY=NO
CMQMDA DSECT=YES
SPACE 2
END

```

Şekil 124. Örnek çevirici DRU çıktısı

z/OS

## IBM z/OSMF olanağının otomatikleştirmek için kullanılması IBM

### MQ

IBM z/OS Management Facility (z/OSMF), anabilgisayar z/OS sistemlerinizin günlük işlemlerini ve yönetimini daha kolay yönetebilmeniz için tümleşik kullanıcı yardımı ile görev odaklı, web tarayıcısı tabanlı bir kullanıcı arabiriminde sistem yönetimi işlevleri sağlar.

z/OSMF, bazı geleneksel görevleri düzene sokarak ve diğerlerini otomatikleştirerek z/OS sistem yönetiminin bazı alanlarını basitleştirmeye yardımcı olabilir.

Kaynaklar, kullanıcı tarafından sağlanan bir portaldan, bir düğme tıklatıldığında sağlanabilir ya da kaynak yetkisi kaldırılabilir. z/OSMF, bu göreve yardımcı olmak için REST API 'leri sağlar.

z/OSMF ile birlikte sağlanan örnek pazar portalı, kaynakların yetkilendirilmesinde ve yetkilendirilmesinde de kullanılabilir. Alternatif olarak, daha deneyimli kullanıcılar z/OSMF Web Kullanıcı Arabirimi 'ni (WUI) kullanabilir.

Bu bölümde z/OSMF'i anladığınız varsayılır, ancak z/OSMF ' u tanıımıyorsanız [z/OSMFile çalışmaya başlamak](#) başlıklı konuyu okumalısınız. Alternatif olarak, bu bölüme z/OSMF WUI çevrimiçi yardımından erişebilirsiniz.

z/OS Cloud yapılandırması hakkında bilgi edinmeniz gerekir, yani:

- Bulut Sağlama- [Kaynak yönetimi hizmetleri](#)
- İş Yüklü Yönetimi-daha fazla bilgi için bkz. [IBM z/OS Management Facility Programming Guide](#) .
- Başlarken-bkz. [Başlarken Öğretici Programı-Bulut Bilgi İşlem](#)

z/OSMF 2.2 rol tabanlı etkinlikler ve görevler sunar; bu nedenle, aşağıdakiler gibi kavramları anlamamız önemlidir:

etki alanları  
Administrators  
Onaylayanlar  
kiracılar  
şablonlar  
eşgörünüm  
iş akışları

Ve böyle devam ediyor.

Örnek IBM MQ z/OSMF iş akışları ve ilişkili dosyalar sağlanır ve IBM MQ for z/OS UNIX System Services Components özelliğinin bir parçası olarak kurulabilir. Bu özelliğe ilişkin kuruluş işlemi ve izin ve dosya yapısı IBM MQ for z/OS Program Dizininde Program Dizinlerine ilişkin yükleme bağlantıları için bkz. [IBM MQ for z/OS Program Dizini PDF dosyaları](#). açıklanmaktadır.

Örnek iş akışları XML biçiminde yazılır ve IBM MQ kuyruk yöneticilerinin, kanal başlatıcılarının ve yerel kuyrukların yetkilendirilmesinin (oluşturulmasının) ya da tahsisinin (yok edilmesinin) nasıl otomatikleştirileceğini ve yetkilendirilen IBM MQ kaynaklarına ilişkin işlemlerin nasıl gerçekleştirileceğini gösterir. İş akışları işleri gönderir (JCL), REXX yürütme dosyalarını çalıştırır, Kabuk komut dosyalarını işler ya da REST API çağrılarını yayınlar.

Örnekler, z/OSMF kullanılarak gerçekleştirilebilecek işlev tiplerini göstermek için tasarlanmıştır. z/OSMF iş akışlarının genellikle konan ya da ileti alma gibi kaynakların ve eylemlerin sağlanması için IBM MQ uygulamaları kullanılarak gerçekleştirileceği tahmin edilmektedir.

Örnek iş akışlarını, iş akışı değişkeni özellikleri ayarlanmışsa (aşağıdaki kısımlarda anlatıldığı gibi), sağlandığı şekilde çalıştırabilir ya da bunları gerektiği şekilde uyarlayabilirsiniz. Ek işlev gerçekleştirmek için kendi iş akışlarınızı yazmayı tercih edersiniz. Örnek iş akışlarını çalıştırmadan önce bkz:

- [“z/OSMF için önkoşullar” sayfa 999](#)
- [“Güvenlik ayarları” sayfa 1001](#)
- [“Sınırlamalar” sayfa 1003](#)

Aşağıdakiler için örnek iş akışı uygulamaları sağlanır:

- [“IBM MQ kuyruk yöneticilerinin yetkilendirilmesini ya da yetkilendirmesini otomatikleştirin ve yetkilendirilen kuyruk yöneticilerine karşı işlemler gerçekleştirin” sayfa 1005](#)
- [“IBM MQ yerel kuyruklarının yetkilendirilmesini ya da yetkilendirmesini kaldırma işlemini otomatikleştirin ve sağlanan kuyruklara karşı işlemler gerçekleştirin” sayfa 1006.](#)

## İlgili kavramlar

“ayarlar” IBM MQ for z/OS” sayfa 881

IBM MQ for z/OS sisteminizi uyarlamak için adım adım bu konuyu kullanın.

## z/OSMF için önkoşullar

IBM MQ ile IBM z/OS Management Facility (z/OSMF) programını çalıştırmak için gereksinim duyduğunuz önkoşullar

IBM MQ for z/OS 9.1.0 içinde verilen iş akışları, hem z/OS 2.1 hem 2.2 üzerindeki APAR ' lar aracılığıyla sağlanan z/OSMF içindeki yeni işlevden yararlanır. Daha fazla ayrıntı aşağıdaki metinde sağlanır.

1. IBM z/OS Management Facility 2.2 ürününü doğru biçimde kurup yapılandırdınız. Güvenlik etkinleştirilmiş olarak çalışıyorsanız, z/OSMF tarafından belgelendiği şekilde tüm güvenlik ayarlarının yapılandırıldığından emin olun.
2. Aşağıdakiler için aşağıdaki APAR ' ları kurdunuz:

#### **z/OS 2.1**

- PI71068
- PI71079
- PI71082
- PI71084
- OA50130

#### **z/OS 2.2**

- PI70526
- PI70521
- PI70527
- PI67839
- PI70767
- PI46315
- OA49081
- OA49802
- OA50130

3. z/OSMF melek (gerekliyse) ve sunucu işlemleri yapılandırıldı.
4. z/OS Cloud ortamı yapılandırılmıştır (yukarıda kısaca anlatıldığı ve z/OSMF tarafından belgelendiği şekilde)
5. IBM MQ for z/OS 9.0.1 kuruldu ve ürün yükleme kitaplıkları kullanılabilir.
6. Aşağıdaki IBM MQ kuyruk yöneticisi uyarlama görevleri gerçekleştirildi:

<b>Görev</b>	<b>Açıklama</b>
1	z/OS sistem parametrelerinin tanımlanması
2	APF IBM MQ yükleme kitaplıklarını yetkilendirir
3	z/OS bağlantı listesini ve LPA ' yı güncelle
4	z/OS program özellikleri çizelgesinin güncellenmesi

7. Örnek iş akışları ve ilişkili dosyalar uygun bir z/OS UNIX System Services (z/OS UNIX) dizinine kurulur.
8. provision.xml iş akışı bu dizinde geçici bir dosya yaratabileceğinden, /tmp z/OS UNIX dizini kullanılabilir. Bir dosya yaratılırsa, iş akışı genel olarak kullanıldıktan sonra dosyayı siler.
9. deprovision.xml dosyasında, CSQ4ZWS1.rexx ve CSQ4ZWS2.rexx REXX yürütücülerini çağıran adımlar vardır. Bu yöneticiler, kuyruk yöneticisi ve kanal başlatıcı altsistemlerinin durmasını bekler; yöneticiler z/OS UNIX **SLEEP** komutunu sistem çağırısı olarak çağırır.

z/OS UNIX yapılandırmanıza bağlı olarak, **SLEEP** komutunun kodlanmış olarak çalışmadığını öğrenebilirsiniz. İşleme sırasında **SLEEP** komutunun bulunmadığını belirten bir hatayla karşılaşırsanız, CSQ4ZWS1.rexx ve CSQ4ZWS2.rexx: execs 'te aşağıdaki satırları değiştirmeyi deneyebilirsiniz:

```
CALL SYSCALLS('ON')          /* Enable z/OS UNIX calls */
```

```
ADDRESS SYSCALL
"SLEEP" 10
CALL SYSCALLS 'OFF'                /* Sleep for 10 seconds */
/* Disable z/OS UNIX calls */
```

birlikte

```
'sleep' 10
```

Daha sonra, PATH ortam deęiřkeni ayarınızı denetlemek için Open MVS (OMVS) **env** komutunu verin. **sleep** komutunu içeren dizinin PATH deęiřkenine tanımlandığını doğrulayın. **sleep** komutunun genellikle /bin dizininde bulunduęunu unutmayın.

10. z/OSMF ' in başlatıldığından emin olun.

Hem melek hem de sunucu z/OSMF işlemleri başlatılmalı ve z/OSMF Web Kullanıcı Arabirimi (WUI) çalışır durumda olmalıdır. Daha fazla ayrıntı için bkz. [Liberty profile: Process types on z/OS](#).

REST API kullanarak iş akışlarını kullanmayı planlarsanız da, z/OSMF WUI ' nin başlatılması gerekir. z/OSMF WUI, iş akışlarının oluşturulmasını ve yürütülmesini izlemek için yararlı olabilir.

### İlgili kavramlar


“IBM z/OSMF olanağının otomatikleştirmek için kullanılması IBM MQ” sayfa 998

IBM z/OS Management Facility (z/OSMF), anabilgisayar z/OS sistemlerinizin günlük işlemlerini ve yönetimini daha kolay yönetebilmeniz için tümleşik kullanıcı yardımı ile görev odaklı, web tarayıcısı tabanlı bir kullanıcı arabiriminde sistem yönetimi işlevleri sağlar.

### z/OS Güvenlik ayarları

z/OSMFkomutunu çalıştırmak için gereken güvenlik ayarları.

Özellikler dosyasında aşağıdaki Kullanıcı Kimliği deęiřkeni özellikleri tanımlanır. Daha fazla ayrıntı için bkz. “İş akışlarının çalıştırılması” sayfa 1008.

Kullanıcı kimliği özellięi	Açıklama
CSQ_USERID	İş akışı adımlarını çalıştırmak için kullanılan kullanıcı kimliği. Ancak, seçilen adımların (genellikle yükseltilmiş yetki düzeyi gerektiren), aşağıdaki metinde listelenen <b>CSQ_ADMIN_*</b> kullanıcı kimliklerinin ayarına dayalı olarak farklı kullanıcı kimlikleriyle çalıştırılacağına dikkat edin. Kullanılmakta olan kullanıcı kimliği, iş akışlarındaki ilgili adımda <b>runAsUser</b> özellięi tarafından tanımlanır.
CSQ_ADMIN_APF_USERID	Kuyruk yöneticisi sistem parametresi modülünü içeren yükleme kitaplığına yetki veren APF için kullanılacak kullanıcı kimliği.
CSQ_APF_APPROVAL_ID	Kullanıcıların veri kümesi APF yetkilendirme adımını kullanıcı CSQ_ADMIN_APF_USERID olarak çalıştırmalarına izin vermek için kullanılan onay kimliği.
CSQ_ADMIN_CONSOLE_USERID	z/OS konsol komutlarını veren çalıştırma altındaki adımlar çalıştırılırken kullanılan kullanıcı kimliği.  <b>Uyarı:</b> Bu kullanıcı kimliğinin, başlatılan görev tanıtımı (MVS.START.STC. *) için UPDATE erişimine izin vermesi gerekir. OPERCMDS sınıfında. Ek bilgi için z/OS belgelerinde <a href="#">İşletmen komutlarının kullanımını denetleme</a> başlıklı konuya bakın.
CSQ_CONSOLE_APPROVAL_ID	Kullanıcıların, CSQ_ADMIN_CONSOLE_USERID kullanıcısı olarak çalıştırma altında z/OS konsol komutlarını veren adımları çalıştırmalarına izin vermek için kullanılan onay kimliği.
CSQ_ADMIN_SAF_USERID	SSF komutları verilirken kullanılacak kullanıcı kimliği.

Kullanıcı kimliği özelliği	Açıklama
CSQ_SAF_APPROVAL_ID	Kullanıcıların CSQ_ADMIN_SAF_USERID kullanıcısı olarak çalıştırma altında SAF komut adımlarını çalıştırmalarına izin vermek için kullanılan onay kimliği.
CSQ_ADMIN_SSI_USERID	z/OS' e sağlanmakta olan altsistemi tanımlamak için SETSSI komutu verilirken kullanılacak kullanıcı kimliği.
CSQ_SSI_APPROVAL_ID	Kullanıcıların CSQ_ADMIN_SSI_USERID kullanıcısı olarak çalıştırma altında SETSSI komut adımını çalıştırmalarına izin vermek için kullanılan onay kimliği.

**Not:** Yetkilendirmeyi çalıştırmak ve iş akışlarının yetkilendirmesini kaldırmak için kullanılan kullanıcı kimliğinin aşağıda listelendiği şekilde yeterli yetkisi olması gerekir:

1. Kuyruk Yöneticisi 'nin yetkilendirmesi ve iş akışlarının yetkilendirmesi, veri kümelerini yetkilendirmek için SETPROG komutunu kullanır. Kullanıcı kimliği CSQ\_ADMIN\_APF\_USERID özelliğinde ayarlanmış ya da iş akışlarını çalıştırmak için kullanılan kullanıcı kimliğinin bu komutu vermesine izin verilmeli. Bunu gerçekleştirmek için aşağıdaki komutu verebilirsiniz:

```
PERMIT MVS.SETPROG CLASS(OPERCMD) ID(value of CSQ_ADMIN_APF_USERID) ACCESS(UPDATE)
```

**Not:** SETPROG komutu bir z/OS sisteminin IPL 'sinde kalıcı olmayabilir; bu nedenle, bir IPL ' den sonra aşağıdaki SETPROG komutunu el ile vermek gerekebilir:

```
SETPROG APF,ADD,DSN=value of CSQ_AUTH_LIB_HLQ.value of CSQ_SSID.APF.LOAD,SMS
```

SETPROG komutuyla ilgili daha fazla ayrıntı için [APF listelerini denetlemek için RACF ' nin kullanılması](#) başlıklı konuya bakın.

Ayrıca, hangi kitaplıkların APF yetkisi verilebileceğini denetlemek için FACILITY sınıfını etkinleştirmiş olabilirsiniz; bu nedenle şu komutu vermeniz gerekebilir:

```
PERMIT CSVAPF.libname CLASS(FACILITY) ID(value of CSQ_ADMIN_APF_USERID) ACCESS(UPDATE)
```

2. Kuyruk Yöneticisi sağlama iş akışındaki bir adım, z/OS' e IBM MQ altsistemini tanıtmak için SETSSI komutunu verir. CSQ\_ADMIN\_SSI\_USERID özelliğinde ayarlanan kullanıcı kimliğinin bu komutu kullanmasına izin verilmeli. Bunu gerçekleştirmek için aşağıdaki komutu verebilirsiniz:

```
PERMIT MVS.SETSSI.ADD CLASS(OPERCMD) ID(value of CSQ_ADMIN_SSI_USERID) ACCESS(CONTROL)
```

**Not:** SETSSI komutuyla z/OS olarak tanımlanan altsistemler, z/OS sisteminin IPL 'si boyunca devam etmez. Bu nedenle, IPL ' den sonra aşağıdaki SETSSI komutunu el ile vermek gerekebilir:

```
SETSSI ADD,S='value of CSQ_SSID',I=CSQ3INI, P='CSQ3EPX,value of CSQ_CMD_PFX,S'
```

SETSSI komutuyla ilgili daha fazla ayrıntı için bkz. [SETSSI komutu](#).

3. İş akışları kuyruk yöneticisi komutlarını verir; güvenliği etkinleştirmeyi planlıyorsanız, CSQ\_ADMIN\_RACF\_USERID özelliğinde ayarlanan kullanıcı kimliğinin (ya da iş akışlarını çalıştırmak için kullanılan kullanıcı kimliğinin) MQADMIN ya da MXADMIN sınıfına (hangi sınıfın kullanıldığına bağlı olarak) CLAUTH (istemci kimlik doğrulaması) yetkisi verilmesi gerekir. Bu, bu kullanıcı kimliğinin bu sınıflara ilişkin güvenlik tanımları tanımlanmasına izin vermek için kullanılır. Bunu gerçekleştirmek için aşağıdaki komutu verebilirsiniz:

**CLAUTH** ile ilgili daha fazla ayrıntı için bkz. [CLAUTH \(class authority\) özniteliği](#).

4. deprovision.xml iş akışı sorunları z/OS komutları; örneğin, DISPLAY ACTIVE işleri, CANCEL ya da FORCE altsistemleri; bu nedenle, CSQ\_ADMIN\_CONSOLE\_USERID özelliğinde ayarlanan kullanıcı kimliğinin (ya da iş akışlarını çalıştırmak için kullanılan kullanıcı kimliğinin) bu komutları yayınlamak için uygun yetkiye sahip olması gerekir.
5. Yazılım Hizmetleri görevinin şablonlar tablosunu kullanarak bir kuyruk yöneticisi örneği isteyen kullanıcıların, z/OSMF tarafından tanımlandığı şekilde z/OSMF ve Configuration Assistant 'a erişme izni olmalıdır.
6. Bir kuyruk yöneticisinin tüketici yetkilendirmesinin kullanıcı kimliği, CSQ\_PROC\_LIB değişkeniyle tanımlanan PROCLIB veri kümesine üye ekleme ve silme yetkisi gerektirir.
7. Bir kuyruk yöneticisi, sağlama kuyruklarından önce yetkilendirilmelidir.
8. queueLoad.xml ve queueOffload.xml iş akışlarını kullanmak için, kullanılan veri kümelerinin önceden tanımlanması gerekir. Ayrıca, bu iş akışlarını çalıştırmak için kullanılan kullanıcı kimliğine veri kümeleri için UPDATE yetkisi verilmesi gerekir.
9. Kuyruk yöneticisi provision.xml iş akışındaki bir adım şu anda altsistem güvenliğini devre dışı bırakır. IBM MQ kaynaklarını korumak için uygun güvenlik komutlarını ekleyerek altsistem güvenliğini etkinleştirmek için csq4znse.jc1 işini değiştirebilirsiniz. Ancak, ek komutlar eklerseniz, deprovision.xml iş akışı tarafından gönderilen csq4dse.jc1 dosyasında güvenlik izinlerini silmek için de komutlar eklemeniz gerektiğini unutmayın.

**Not:** Bu adım, RACF güvenlik komutlarını verir. Alternatif bir güvenlik ürünü kullanıyorsanız, güvenlik ürününüze ilişkin uygun komutları yayınlamak için bu adımı değiştirmeniz gerekir.

## Ağ Gereksinimleri

Bir kuyruk yöneticisi şablonu ve şablona ilişkin kaynaklar eklerken **Ağ kaynağı havuzu yarat** ögesini tıklatmanız gerekir. Bu, bu şablon için ağ kaynakları olan bir kaynak havuzu yaratır.

Configuration Assistant olanağını kullanarak, ağ denetimcinizin bu şablon için ayrılacak kapı sayısı için bir sınır tanımlayarak bu ağ kaynağı havuzu tanımlamasını tamamlaması gerekir.

Her şablon eşgörünümü için, provision.xml iş akışı aralıktaki bir kapıyı ayırır ve o kapıda dinleyecek bir dinleyici başlatır.

## IBM Workload Manager ile sınıflandırma

Kuyruk yöneticisi ve kanal başlatıcısı adres alanlarını WLM ile sınıflandırmak istiyorsanız, bir kuyruk yöneticisini yetkilendirmek için bir şablon eklerken bunu belirtmeniz gerekir.

Sınıflandırılıp sınıflandırılmayacağı, workflow\_variables.properties dosyasında ayarlanan **CSQ\_DEFINE\_MSTR\_WLM\_RULE** ve **CSQ\_DEFINE\_CHIN\_WLM\_RULE** işaretleri tarafından denetlenir.

WLM ile sınıflandırma hakkında daha fazla bilgi için bkz. [z/OSMF Yapılandırma Kılavuzu](#).

### İlgili kavramlar

[“z/OSMF için önkoşullar” sayfa 999](#)

IBM MQ ile IBM z/OS Management Facility (z/OSMF) programını çalıştırmak için gereksinim duyduğunuz önkoşullar

## Sınırlamalar

IBM MQ ile z/OSMF kullanılırken sınırlamalar.

1. provision.xml iş akışı, vurgulanan şu kuyruk yöneticisi uyarılma görevlerini otomatikleştirir:

Görev	Açıklama
1	z/OS sistem parametrelerinin tanımlanması
2	APF IBM MQ yükleme kitaplıklarını <b>yetkilendirir (provision.xml , APF ' nin bazı kitaplıkları yetkilendirmesini sağlar)</b>
3	z/OS bağlantı listesini ve LPA ' yı güncelle
4	z/OS program özellikleri çizelgesinin güncellenmesi
5	<b>IBM MQ altsisteminin tanımlanması z/OS</b>
6	<b>IBM MQ kuyruk yöneticisi için yordam yaratılması</b>
7	<b>Kanal başlatıcı için yordamlar yaratılması</b>
8	<b>IBM MQ altsistemini bir z/OS WLM hizmet sınıfına tanımlama</b>
9	Bağlaşım olanağı boşaltma depolama ortamınızı seçin ve ayarlayın
10	Bağlaşım olanağını ayarla
11	ESM güvenlik denetimlerinizi uygulayın
12	SYS1.PARMLIB üyeleri
13	<b>Başlatma girişi veri kümelerinin uyarlanması</b>
14	<b>Önyükeme ve günlük veri kümelerinin yaratılması</b>
15.000	<b>Sayfa kümelerinizi tanımlayın</b>
16	IBM MQ girdilerini Db2 veri paylaşım grubuna ekleme
17	<b>Sistem değiştirgesi modüllerinizi uyarla (bazı)</b>
18	<b>Kanal başlatıcı değiştirgelerini uyarla (bazıları)</b>
19	Toplu İş, TSO ve RRS bağdaştırıcılarını ayarla
20	İşlemleri ve denetim panolarını ayarlama
21	IBM MQ döküm biçimlendirme üyesini dahil et
22	Bilgi iletilerini engelle
23	Advanced Message Security için sistem DIAG üyenizi güncelleyin
24	Advanced Message Security için yordamlar oluşturma
25	Başlatılan görev kullanıcısı Gelişmiş İleti Güvenliğini Ayarla
26	Advanced Message Security için güvenlik yöneticisine RACDCERT izinleri verin
27	Advanced Message Security için kullanıcılara kaynak izinleri verin

2. Koyu metinle vurgulanmayan özelleştirme görevlerinin, gerekirse el ile gerçekleştirilmesi gerekir.
3. Örnek INP1 ve INP2 üyeleri şu anda olduğu gibi kullanılmaktadır. Gerekirse, bu üyeler tarafından tanımlanan kaynakları denetlemek için ek özellikler tanımlanabilir.
4. Özellikler dosyasında listelenen belirli özelliklerle ilgili açıklamalar, bu özelliklerin kullanılmasıyla ilgili sınırlamaları gösterir. Daha fazla ayrıntı için bkz. [“İş akışlarının çalıştırılması”](#) sayfa 1008.

### İlgili kavramlar

“Güvenlik ayarları” sayfa 1001

z/OSMFkomutunu çalıştırmak için gereken güvenlik ayarları.



## z/OS IBM MQ nesnelarının saęlanmasını otomatikleştirin

Kuyruk yöneticilerinin ve yerel kuyrukların saęlanmasını otomatikleştirmek için örnekler saęlanır.

### IBM MQ kuyruk yöneticilerinin yetkilendirilmesini ya da yetkilendirmesini otomatikleştirin ve yetkilendirilen kuyruk yöneticilerine karşı işlemler gerçekleştirin

Aşağıdaki kuyruk yöneticisine özgü örnek z/OSMF iş akışları saęlanır:

İş akışı adı	Açıklama
provision.xml	<p>IBM MQ for z/OS kuyruk yöneticisine erişim saęlama</p> <p>Bu örnek iş akışı:</p> <ul style="list-style-type: none"><li>• Bir kuyruk yöneticisi için gerekli sistem kaynaklarını saęlar.</li><li>• Kanal başlatıcı için gerekli sistem kaynaklarını saęlar.</li><li>• Kuyruk yöneticisini başlatır (kanal başlatıcıyı ve TCP/IP dinleyicisini de başlatır)</li><li>• Örnek kuyruk yöneticisi kuruluş doğrulama programını çalıştırır.</li></ul> <p>Bir ortam özellięi, farklı özelliklere sahip kuyruk yöneticilerinin yetkilendirmesini denetlemek için ayarlanabilir. Daha fazla bilgi için bkz <a href="#">“İş akışlarının çalıştırılması” sayfa 1008</a>.</p> <p><b>Not:</b> Bu iş akışı için şablon eklenmesine yardımcı olacak bir bildirge dosyası (provision.mf) saęlanır. Bu dosya, ek bilgi içeren <b>qaas_readme.pdf</b> dosyasına bir başvuru içerir. Şablon eklendikten sonra dosyaya bir bağlantı aracılığıyla erişebilirsiniz.</p>
deprovision.xml	<p>IBM MQ for z/OS kuyruk yöneticisinin yetkilendirmesi</p> <p>Bu örnek iş akışı:</p> <ul style="list-style-type: none"><li>• Kanal başlatıcıyı (TCP/IP dinleyicisini de durdurur) ve kuyruk yöneticisini durdurur.</li><li>• Altsistemlerin durmasını bekler</li><li>• Tüm kanal başlatıcı ve kuyruk yöneticisi sistem kaynaklarını kullanır.</li></ul>
startQMgr.xml	<p>IBM MQ for z/OS kuyruk yöneticisinin başlatılması</p> <p>Bu örnek iş akışı, (kanal başlatıcıyı ve TCP/IP dinleyicisini de başlatan) kuyruk yöneticisini başlatır.</p>
stopQMgr.xml	<p>IBM MQ for z/OS kuyruk yöneticisinin durdurulması</p> <p>Bu örnek iş akışı, kanal başlatıcıyı (TCP/IP dinleyicisini de durdurur) ve kuyruk yöneticisini durdurur.</p>

Her iş akışı bir ya da daha çok adım gerçekleştirir. İş akışlarındaki açıklamalar, her adım tarafından gerçekleştirilen işlevi açıklar. Bazı adımlar veri girişi isteęinde bulunurken, bazı adımlar JCL 'yi gönderirken, belirtilen işlevi gerçekleştirmek için REXX yürütme dosyalarını, Shell komut dosyalarını ya da REST API çağrılarını çağırır.

JCL ya da REXX exec dosyalarının tam adı için her adıma bakın. İş akışları ve ilişkili JCL ya da REXX exec dosyaları, bir ya da daha fazla deęişken XML dosyasında bildirilen deęişkenlere başvurmaktadır. Daha fazla ayrıntı için bkz. [“İş akışı deęişkeni bildirim dosyaları” sayfa 1008](#).

**deprovision**, **startQMgr** ve **stopQMgr**, yetkilendirilen bir IBM MQ for z/OS kuyruk yöneticisine yönelik işlemler olarak gerçekleştirilebilir.

## IBM MQ yerel kuyruklarının yetkilendirilmesini ya da yetkilendirmesini kaldırma işlemini otomatikleştirin ve sağlanan kuyruklara karşı işlemler gerçekleştirin

Aşağıdaki kuyruğa özgü örnek z/OSMF iş akışları sağlanır:

İş akışı adı	Açıklama
defineQueue.xml	<p>Yerel kuyruk tanımla</p> <p>Bu örnek iş akışı, z/OSMF iş akışlarının özellik ayarlarına dayalı olarak küçük, orta ya da büyük boyutlu kuyruklar tanımlamak için nasıl kullanılabileceğini gösterir.</p> <p><b>Not:</b> Bu iş akışı için şablon eklenmesine yardımcı olacak bir bildirge dosyası (provision.mf) sağlanır. Bu dosya, ek bilgi içeren <b>qaas_readme.pdf</b> dosyasına bir başvuru içerir. Şablon eklendikten sonra dosyaya bir bağlantı aracılığıyla erişebilirsiniz.</p>
displayQueue.xml	<p>Yerel bir kuyruğun seçilen özniteliklerini görüntüle</p> <p>Bu örnek iş akışı, yerel bir kuyruğun seçilen özniteliklerini görüntüler. Öznitelikler bir z/OSMF değişkeninde döndürülür (değişkenin adına ilişkin iş akışındaki adımlara bakın) ve daha sonra görüntülenir. Gerekirse, değişkenin içeriğine bir REST API kullanılarak erişilebilir.</p> <p>Daha fazla ayrıntı için bkz. <a href="#">Cloud provisioning REST API ' lerine</a> ayrıca bkz. <a href="#">z/OSMF iş akışı hizmetleri</a>.</p>
deleteQueue.xml	<p>Yerel kuyruğun silinmesi</p> <p>Bu örnek iş akışı, belirtilen bir kuyruk yöneticisindeki yerel bir kuyruğu siler.</p>
putQueue.xml	<p>Yerel kuyruğa bir ya da daha çok ileti koyun.</p> <p>Bu örnek iş akışı, bir ya da daha çok iletiyi yerel kuyruğa koyar. İleti metni belirlenebilir, ancak aynı anda birden çok ileti yerel kuyruğa yerleştirilirse, aynı ileti metni kullanılır.</p>
getQueue.xml	<p>Yerel bir kuyruktan bir ya da daha çok ileti alın.</p> <p>Bu örnek iş akışı, yerel bir kuyruktan bir ya da daha çok ileti alır. İletiler bir z/OSMF değişkeninde döndürülür (değişkenin adı için iş akışındaki adımlara bakın) ve daha sonra görüntülenir. Gerekirse, bir REST API kullanılarak değişkenin içeriğine erişebilirsiniz.</p> <p>Daha fazla ayrıntı için bkz. <a href="#">Cloud provisioning REST API ' lerine</a> ayrıca bkz. <a href="#">z/OSMF iş akışı hizmetleri</a>.</p>
loadQueue.xml	<p>İletileri bir veri kümesinden yerel kuyruğa yükleyin.</p> <p>Bu örnek iş akışı, bir veri kümesindeki iletileri yerel bir kuyruğa yükler. Veri kümesinin varsayılan adı bir özellik ayarlanarak belirtilir. Daha fazla ayrıntı için bkz. <a href="#">“İş akışlarının çalıştırılması” sayfa 1008</a>.</p>
offloadQueue.xml	<p>İletileri yerel bir kuyruktan bir veri kümesine boşaltın.</p> <p>Bu örnek iş akışı, iletileri yerel bir kuyruktan bir veri kümesine yükler. Veri kümesinin varsayılan adı bir özellik ayarlanarak belirtilir. Daha fazla ayrıntı için bkz. <a href="#">“İş akışlarının çalıştırılması” sayfa 1008</a>.</p>
clearQueue.xml	<p>Yerel kuyruktaki iletileri temizler.</p> <p>Bu örnek iş akışı, yerel bir kuyruktaki tüm iletileri temizler (siler).</p>

## Notlar:

1. **Koyma Kuyruğu** işlemi, bazı ileti verilerini girmenizi ve bir ya da daha çok iletiyi kuyruğa koymanızı sağlar. Belirli bir istek sırasında kuyruğa birden çok ileti yerleştirilirse, aynı ileti verileri kullanılır.
2. loadQueue.xml ve offloadQueue.xml iş akışları, SCSQLOAD kitaplığındaki CSQUDMSG yürütülür modülünü QLOAD diğer adıyla çağırır. Bu, IBM MQ for Multiplatforms ile kullanılabilen **dmpmqmsg** yardımcı programına eşdeğerdir. Bu nedenle, bir veri kümesinden bir kuyruğa ya da bir kuyruktan bir veri kümesine yüklenen iletilerin **dmpmqmsg** biçiminde olması beklenir.

Örnek JCL, SCSQPROC ' de CSQ4QLOD üyesi olarak da sağlanır.

loadQueue ve offloadQueue işlemlerini denemenin en kolay yolu aşağıdaki işlemleri gerçekleştirmektir:

- a. Kuyruğa bazı iletiler koymak için **putQueue** komutunu birkaç kez yayınlayın.
- b. Kuyruktaki iletileri bir veri kümesine boşaltmak için **offloadQueue** seçeneğini kullanın.
- c. Gerekliyse, kuyruktan tüm iletileri kaldırmak için **clearQueue** komutunu verin.
- d. Bir veri kümesindeki iletileri aynı ya da farklı bir kuyruğa yüklemek için **loadQueue** seçeneğini kullanın.

**dmpmqmsg** biçimiyle ilgileniyorsanız, bir Boşaltma isteği yayınladıktan sonra veri kümesinin içeriğine göz atabilirsiniz.

3. Sağlanan IBM MQ for z/OS yerel kuyruğuna ilişkin işlemler olarak **displayQueue**, **deleteQueue**, **putQueue**, **getQueue**, **loadQueue**, **offloadQueue** ve **clearQueue** işlemlerini gerçekleştirebilirsiniz. İşlemler ve işlem dosyalarıyla ilgili daha fazla ayrıntı için *z/OSMF Programming Guide* belgesine bakın.
4. İşlemlerle ilgili tüm iş akışları varsayılan olarak silinir. Bunun nedeni, kullanıcıların iş akışlarını temizleme gereksinimini en aza indirmektir.

Ancak bununla ilgili sorun, bir eylemin bazı çıkışlarla sonuçlandığı yerdir. Örneğin, **displayQueue** ve **getQueue** işlemlerinin her ikisi de çıkış üretir.

İşlem gerçekleştirilir gerçekleştirilmez ilgili iş akışı silindiğinden çıkış görülemiyor. Bu nedenle, iş akışı işlemlerini z/OS WUI 'sinden yürütüyorsanız, çıkışını görmek istediğiniz her işlem için **< workflow>** etiketinde **cleanAfterComplete** işaretini *false* olarak ayarlamanız gerekir.

Örneğin, **displayQueue** çıkışını görmek için işareti aşağıdaki gibi ayarlayın:

```
<action name="displayQueue">
  <workflow cleanAfterComplete="false">
    ...
  </workflow>
</action>
```

Ancak bu, işlemle ilgili iş akışlarını el ile temizlemeniz gerektiği anlamına gelir.

Her örnek z/OSMF iş akışı bir ya da daha fazla adım gerçekleştirir. İş akışlarındaki açıklamalar, her adım tarafından gerçekleştirilen işlevi açıklar. Bazı adımlar veri girişi isteğinde bulunurken, bazı adımlar JCL ' yi gönderirken diğerleri belirtilen işlevi gerçekleştirmek için REXX yürütücülerine başvurmaktadır.

JCL ya da REXX exec dosyalarının tam adı için her adıma bakın. İş akışları ve ilişkili JCL ya da REXX exec dosyaları, bir ya da daha fazla [“İş akışı değişkeni bildirim dosyaları” sayfa 1008](#) içinde bildirilen değişkenlere başvuruda bulunuyor.

## İlgili kavramlar

“Sınırlamalar” sayfa 1003

IBM MQ ile z/OSMF kullanılırken sınırlamalar.

## z/OS İş akışlarının çalıştırılması

Örnek z/OSMF iş akışları tarafından başvuru dosyalarının açıklaması ve bir iş akışını nasıl çalıştırdığımız.

### İş akışı değişkeni bildirim dosyaları

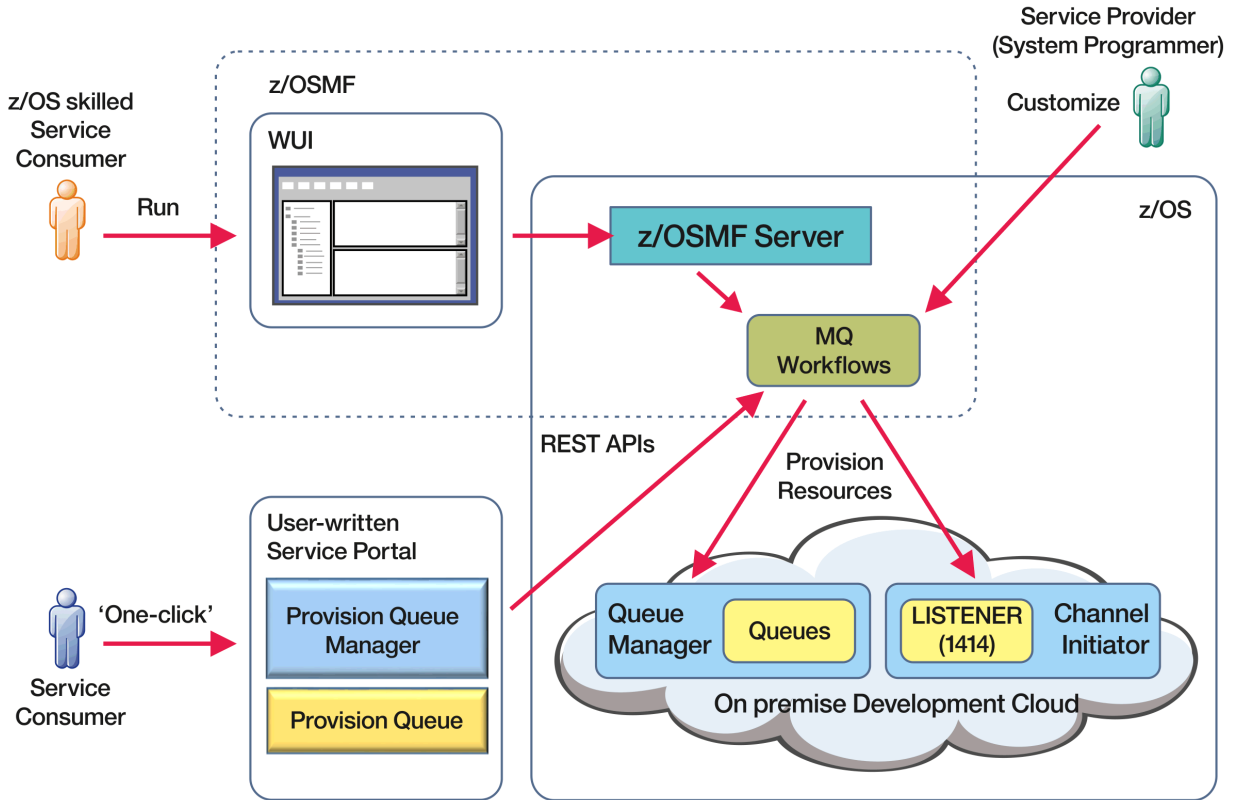
Aşağıdaki dosyalar, örnek z/OSMF iş akışları ve ilişkili JCL ya da REXX exec dosyaları tarafından başvuru değişkenleri bildirir:

İş akışı değişkeni bildirim dosyası adı	Açıklama
common_variables.xml	Hem kuyruk yöneticisi (ve kanal başlatıcısı) hem de kuyruk iş akışları için ortak değişkenler.
qmgr_variables.xml	Kuyruk yöneticisine (ve kanal başlatıcısına) özgü değişkenler.
queue_variables.xml	Kuyruk iş akışlarına özgü değişkenler.
tcPIP_variables.xml	Kuyruk yöneticisine (ve kanal başlatıcısına) özgü ve TCP/IP kaynaklarını tanımlamak için kullanılan değişkenler.

**Not:** Değişkenlerin varsayılan görünürlüğü *özel*' dir. Değişkenlerin z/OSMF REST API kullanılarak sorgulanmasına izin vermek için, seçilen değişkenler *genel* olarak işaretlenmiştir. Ancak, gerekirse belirli bir değişkenin görünürlüğünü değiştirebilirsiniz.

### İş akışlarının çalıştırılması

Şekil 125. IBM MQ for z/OS kaynakları için 'Tek tıklatma' yetkilendirmesi



İş akışlarının çalıştırılabilmesi için önce aşağıdaki dosyada bazı özelliklerin ayarlanması gerekir:

İş akışı değişkeni özellikleri dosyası adı	Açıklama
workflow_variables.properties	<p>İş akışı değişkenlerine ilişkin başlangıç özellikleri. Dosyadaki yorumlar, her özelliğin amacını gösterir.</p> <ul style="list-style-type: none"> <li>• Meta parantezler (&lt; &gt;) içindeki özelliklerin kullanıcıya özgü değerlere ayarlanması gerekir.</li> <li>• Bir ortam özelliği, kuyruk yöneticilerini geliştirme (DEV) ya da test (TEST) ya da kalite güvencesi (QA) ya da üretim (PROD) ortamları için yetkilendirmek üzere ayarlanabilir.</li> </ul> <p>Ek özellik ayarları, her ortam için sağlanacak kuyruk yöneticisinin özelliklerini denetler. Örneğin, her ortam tipi için etkin günlük sayısını ya da sayfa kümesi sayısını değiştirebilirsiniz.</p> <ul style="list-style-type: none"> <li>• Diğer özellikler IBM MQ varsayılan değerlerine ayarlanır, ancak gerekirse yerel kuralları karşılayacak şekilde değiştirilebilir.</li> </ul>

Genel olarak, özellikler ayarlandıktan sonra iş akışları olduğu gibi çalıştırılabilir. Ancak, gerekiyorsa, var olan adımları değiştirmek ya da kaldırmak ya da yeni adımlar eklemek için bir iş akışını özelleştirebilirsiniz.

İş akışları çalıştırılabilir:

- z/OSMF WUI 'den.

WUI 'deki Bulut Sağlama-> Yazılım Hizmetlerinden iş akışları otomatik veya manuel kipte çalıştırılabilir. El ile kip, sınavı sırasında kullanışlıdır ve her iki kipte de iş akışındaki her adımın ilerleyişi izlenebilir.

Daha fazla ayrıntı için bkz. [Bulut sağlama hizmetleri](#) ve [İş akışı oluşturma](#).

- z/OSMF REST Workflow Services kullanılarak.

REST Workflow Services, iş akışlarını bir REST API aracılığıyla çalıştırmak için kullanılabilir. Bu kip, kullanıcı tarafından yazılan bir portaldan tek tıklamayla işlem yaratmak için kullanışlıdır.

Daha fazla ayrıntı için bkz. [Cloud provisioning REST API ' lerine](#) ayrıca bkz. [z/OSMF iş akışı hizmetleri](#).

- z/OSMF ile sağlanan örnek pazar portalı kullanılıyor.

### İlgili kavramlar

“IBM MQ nesnelerinin sağlanmasını otomatikleştirin” sayfa 1005

Kuyruk yöneticilerinin ve yerel kuyrukların sağlanmasını otomatikleştirmek için örnekler sağlanır.

## z/OS MQ Adv. VUE Uzak z/OS kuyruk yöneticilerine MFT aracı bağlanırlığının etkinleştirilmesi

Bazı durumlarda, z/OS üzerindeki Managed File Transfer araçları istemci bağlantısı kullanarak z/OS üzerindeki bir uzak kuyruk yöneticisine bağlanabilir. Bu, daha basit IBM MQ topolojilerine yol açabilir.

Uzak z/OS kuyruk yöneticilerine yönelik istemci bağlantıları aşağıdaki durumlarda desteklenir:

- **LTS** **V 9.3.4** MFT aracı IBM MQ 9.3.4 ya da daha sonraki bir sürümde ya da Long Term Support PH56722 APAR ile uygulanmış ve IBM MQ Advanced for z/OS VUE ya da IBM MQ Advanced for z/OS ürün tanıtıcısıyla (PID) ilişkilendirilmiştir.
- MFT aracı IBM MQ 9.3.0 adresinde yer almakta ve IBM MQ Advanced for z/OS VUEPID 'si ile ilişkilendirilmiştir.

Farklı PID ' lerle ilgili bilgi için [IBM MQ ürün tanıtıcıları ve dışa aktarma bilgileri](#) başlıklı konuya bakın.

Bir MFT kuruluşuyla ilişkili PID ' nin ayarlanmasıyla ilgili bilgi için bkz. [fteSetProductId](#).

Aracının altında çalıştığı PID, aracı başlatıldığında günlükte görüntülenir.

z/OS üzerinde çalışan bir MFT aracı, başka bir PID altında çalışır ve bağ tanımlama kipi bağlantısını kullanarak yalnızca yerel bir kuyruk yöneticisine bağlanabilir.

Bir aracı z/OS üzerinde çalışmayan bir kuyruk yöneticisine bağlanmayı denerse, BFGQM1044E iletisi yayınlanır ve aracı başlatma sona erdirilir.

### İlgili görevler

[z/OS üzerinde bir MFT aracısının başlatılması](#)

## yapılandırma IBM MQ Internet Pass-Thru

Bu bölümde, IBM MQ Internet Pass-Thru (MQIPT) tarafından desteklenen çeşitli özellikler ve bunların nasıl yapılandırılacağı açıklanmaktadır.

`mqipt.conf` yapılandırma dosyasında değişiklik yaparak MQIPT ' i yapılandırın. MQIPT yapılandırma dosyasının yapısı ve belirtilebilir özellikler, [IBM MQ Internet Pass-Thru yapılandırma başvurusunda](#) açıklanır.

**Not:** Yetkisiz kullanıcıların saklanan parolaları görmesini ya da yapılandırmayı değiştirmesini önlemek için, `mqipt.conf` dosyasının bulunduğu dizinde güvenli dosya izinleri ayarlamanız gerekir. "[MQIPT içinde saklanan parolaları şifreleme](#)" sayfa 1048 içindeki yordamı izleyerek yapılandırma dosyasında belirtilen tüm parolaları koruyun.

Yapılandırma dosyasında yapılan değişiklikler, MQIPT başlatıldığında ya da yenilendiğinde yürürlüğe girer. MQIPT ' un etkin bir örneğinin yenilenmesi, yapılandırma değişikliklerini yeniden başlatmadan yürürlüğe sokar MQIPT. MQIPT yenilendiğinde, `mqipt.conf` yapılandırma dosyası yeniden okunur ve MQIPT aşağıdaki işlemleri gerçekleştirir:

- Etkin değil olarak işaretlenen ya da artık konfigürasyon dosyasında belirtilmeyen etkin rotalar kapanır ve artık gelen bağlantıları kabul etmez.
- Yapılandırma dosyasında etkin olarak işaretlenen ve şu anda çalışmayan rotalar başlatılır.
- Etkin rotaların yapılandırma parametrelerinde yapılan değişiklikler uygulanır. Mümkün olduğunda, bu değişiklikler etkin bağlantılarda herhangi bir kesinti olmadan yürürlüğe girer. Rota hedefinde değişiklik yapılması gibi bazı parametre değişiklikleri için, değişiklik uygulanmadan ve rota yeniden başlatılmadan önce tüm bağlantılar kapatılır.

MQIPT' i yenilemek için `mqiptAdmin` komutunu kullanın. `mqiptAdmin` komutunu kullanarak MQIPT komutunu yönetme hakkında daha fazla bilgi için bkz. [MQIPT komutunu komut satırını kullanarak yönetme](#).

## MQIPT içinde HTTP desteği

MQIPT , HTTP tüneli oluşturmayı destekler. MQIPT , iletildiği veri paketlerinin HTTP istekleri olarak kodlanması için yapılandırılabilir.

IBM MQ kanalları HTTP isteklerini kabul etmez. Bu nedenle, HTTP isteklerini almak ve bunları IBM MQ protokol paketlerine dönüştürmek için ikinci bir MQIPT gereklidir. İkinci MQIPT , gelen paketi hedef kuyruk yöneticisine geçirmeden önce standart bir IBM MQ iletişim kuralı paketine dönüştürmek için HTTP üstbilgisini kaldırır.

HTTP , iki MQIPT eşgörünümü arasında kullanıldığında, HTTP istek ve yanıt akışının kalıcı olduğu ve ileti kanalının kullanım ömrü boyunca açık tutulduğu TCP/IP bağlantısı. MQIPT , istek/yanıt çiftleri arasındaki TCP/IP bağlantısını kapatmaz.

İki MQIPT eşgörünümü HTTP aracılığıyla iletişim kuruyorsa, bir HTTP isteği uzun bir süre beklemeye devam edebilir. Sunucu tarafı iletim kuyruğuna yeni iletilerin gelmesini beklerken, istekte bulunan/sunucu kanalında örnek verilebilir. IBM MQ kanal iletişim kuralı "Kalp atışı" mekanizması, ortağına kalp atışı iletileri göndermek için belirli aralıklarla bekleme sona ermesini gerektirir. Varsayılan kanal sağlıklı işletim bildirim süresi 5 dakikadır. MQIPT , HTTP yanıtı olarak bu sağlık sinyalini kullanır. Bazı güvenlik duvarlarında zaman aşımalarıyla ilgili sorunlara neden olmasını önlemek için bu kanal sağlıklı işletim sinyalini devre dışı bırakmayın ya da çok yüksek bir değere ayarlamayın.

MQIPT , bir HTTP yetkili sunucusu ya da sunucusu tarafından oluşturulan, parçalı biçimde HTTP trafiğini kabul eder.

MQIPTiçinde HTTP kullanımı örneği için bkz. [HTTP tünelleme'](#) nin yapılandırılması.

## HTTP yetkili sunucuları

İki MQIPTeşgörünümü arasına bir HTTP yetkili sunucusu yerleştirilebilir. HTTP yetkili sunucusu aşağıdaki gereksinimleri karşılamalıdır:

- Yetkili sunucu HTTP 1.1 iletişim kuralını desteklemelidir.
- MQIPT tarafından ayarlanan **Connection** ya da **Proxy-Connection** HTTP üstbilgileri yetkili sunucu tarafından yerine getirilmelidir. Bu, iki MQIPT örneği arasındaki bağlantıların, ileti kanalının kullanım ömrü boyunca açık tutulmasını sağlar.
- Kalıcı bağlantıların bire bir eşlemesi yetkili sunucuda tutulmalıdır. Bu, yetkili sunucudan hedef MQIPT hedefine TCP/IP bağlantılarının, birden çok ileti kanalına ilişkin verileri iletmek için kullanılmamasını sağlar.

Bazı HTTP yetkili sunucularında kalıcı bağlantıların nasıl yönetileceğini yapılandırmak için özellikleri ayarlayabilirsiniz. Örneğin, kalıcı bir bağlantıda yapılabilecek istek sayısı üst sınırını ayarlayabilirsiniz. Aşağıdaki özellikler ayarlanmalıdır:

- Kalıcı bağlantılar etkinleştirilmelidir.
- Yetkili sunucu üzerinden kalıcı bağlantıların bire bir eşlemesini korumak için, yetkili sunucudan MQIPT birden çok HTTP oturumu tarafından TCP/IP bağlantılarının yeniden kullanımı devre dışı bırakılmalıdır.
- Yetkili sunucu isteklerinde zamanaşımı yüksek bir değere ayarlanmalıdır. Örneğin, 12 saat.
- Kalıcı bir bağlantıda yapılabilecek istek sayısı üst sınırı yüksek bir değere ayarlanmalıdır. Örneğin, 5000.

MQIPT , MQIPT' un iki örneği arasında veri göndermek için HTTP POST isteklerini kullanır. MQIPT yapılandırması **HTTProxy** özelliğini kullanarak yetkili sunucunun anasistem adını belirtiyorsa, MQIPT yetkili sunucuya bağlanır ve HTTP CONNECT yöntemini kullanarak yetkili sunucunun hedefe bir tünel oluşturmasını sağlar MQIPT. Bu, HTTPS bağlantılarının yetkili sunucuda TLS oturumunu sonlandırmadan yetkili sunucudan geçmesini sağlar.

MQIPT yönetim ortamları arasına bir yük dengeleyici yerleştirilirse, her oturuma ilişkin tüm isteklerin aynı hedefe iletilmesini sağlamak için *MQIPTSessionId* HTTP tanımlama bilgisinin değerini kullanacak şekilde yapılandırılmalıdır.

## MQIPT içinde HTTPS

HTTPS , istemci bağlantısını yayınlayan MQIPT ürününde **HTTPS** ve **SSLClient** rota özellikleri etkinleştirilerek bir HTTP bağlantısında kullanılabilir.

MQIPT , hedef HTTP yetkili sunucusunun kimliğini doğrulamak için kullanılacak güvenilir CA sertifikasına erişime sahip olmalıdır. **SSLClientCAKeyring** özelliği, güvenilir CA sertifikasını içeren anahtarlık dosyasını tanımlamak için kullanılabilir.

HTTPS için ortak bir ayar, bir güvenlik duvarı üzerinden tünel açmak ve uzak bir HTTP sunucusuna (ya da başka bir yetkili sunucuya) bağlanmak için yerel bir HTTP yetkili sunucusu kullanır. Bu sunucu, uzak MQIPTsunucusuna bağlanır. Bağlantının sunucu tarafındaki bu MQIPT , bağlantı isteği herhangi bir normal HTTP bağlantısı olarak değerlendirildiğinden belirli bir yapılandırmaya gerek duymaz.

MQIPT , yerel ve uzak yetkili sunucuları ayırt etmek için **HTTProxy** ve **HTTPServer** özelliklerini kullanır. **HTTProxy** özellik ayarlı MQIPT rotası yerel HTTP yetkili sunucusu ve **HTTPServer** özellik ayarlı MQIPT rotası uzak sunucu (ya da yetkili sunucu) olarak görülür.

HTTPS bağlantıları genellikle HTTP proxy/server üzerindeki 443 dinleyici kapısı adresine yapılır, ancak bu varsayılanı geçersiz kılmak için **HTTProxyPort** ve **HTTPServerPort** özellikleri kullanılabilir.



## MQIPT içinde SOCKS desteği

SOCKS yetkili sunucusu, güvenlik duvarı üzerinden denetimli bir çıkış noktası olarak kullanılan bir ağ hizmetidir. Güvenlik duvarının içinde çalışan SOCKS etkin bir uygulama, uzak bir uygulamaya bağlanmak için SOCKS yetkili sunucusunu kullanabilir.

MQIPT , **SocksServer** özelliğini etkinleştirerek SOCKS etkin bir IBM MQ uygulamasının MQIPT üzerinden uzak bir IBM MQ kuyruk yöneticisine bağlanmasına izin vererek SOCKS yetkili sunucusu olarak işlev görür. Bu özellik kullanılırken, hedef hedef ve hedef kapı adresi SOCKS el sıkışması işlemi sırasında alınır ve bu nedenle **Destination** ve **DestinationPort** rota özellikleri geçersiz kılınır. Bu, IBM MQ kümelemeyi desteklemek için temel bir özelliktir.

MQIPT , SOCKS etkinleştirilmemiş yerel bir IBM MQ uygulaması adına SOCKS istemcisi olarak da işlev görür. Bu, yalnızca SOCKS yetkili sunucusu aracılığıyla giden bağlantılara izin veren bir güvenlik duvarı kullanıldığında yararlıdır. Her MQIPT rotasının konfigürasyonu, farklı bir SOCKS yetkili sunucusuyla iletişim kuracak şekilde tanımlanabilir.

SOCKS ' un nasıl kullanılacağına ilişkin bir örnek için [SOCKS yetkili sunucusunun yapılandırılması](#) başlıklı konuya bakın.

## MQIPT içinde kümeleme

IBM MQ kümeleri, MQIPT ile birlikte kullanılabilir; SOCKS, kümede interneti kapsayan her kuyruk yöneticisinin etkinleştirilmesini ve MQIPT ' un SOCKS yetkili sunucusu olarak işlev görmesini sağlar.

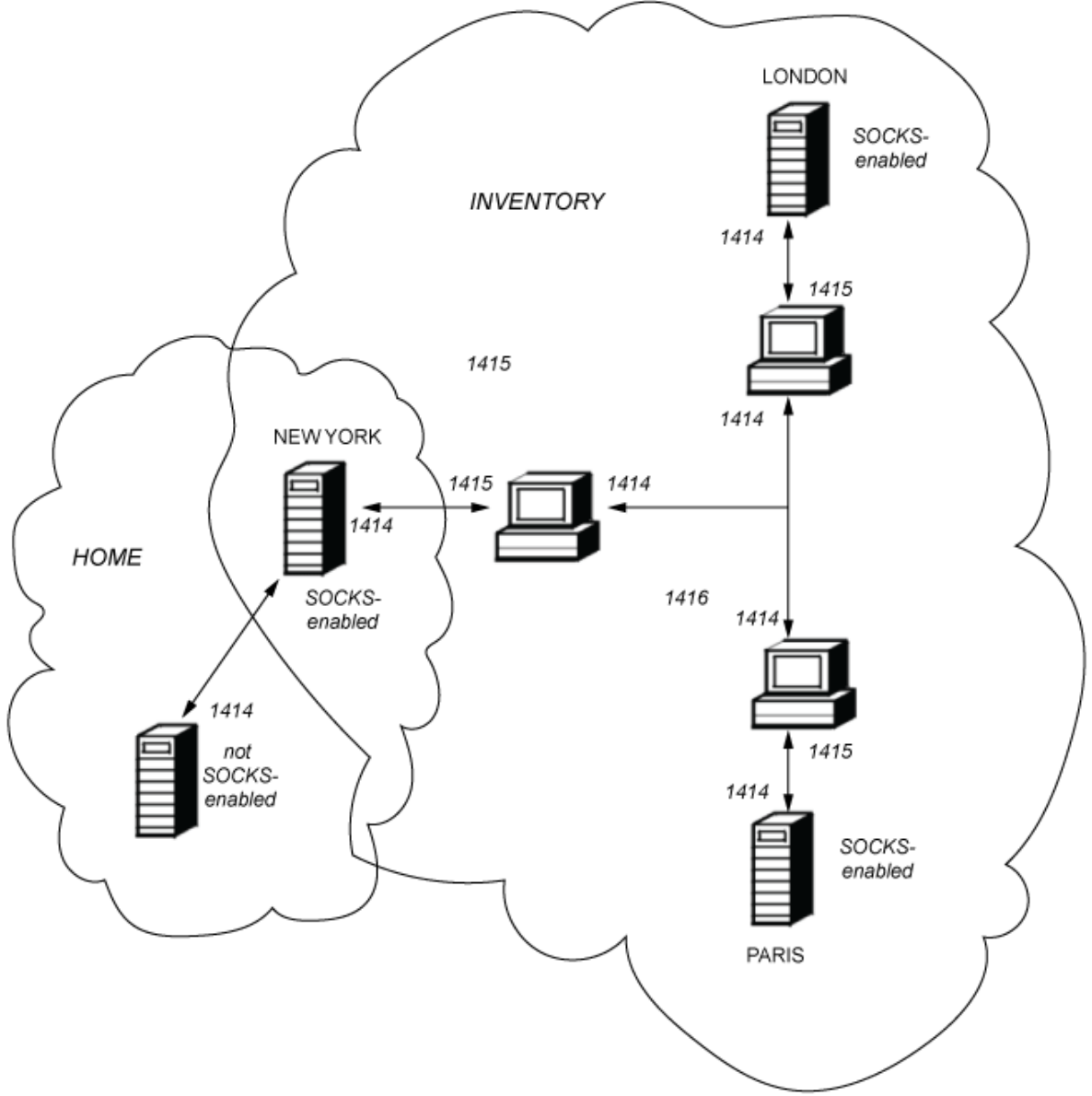
Aşağıdaki şemada, NEWYORK ve CHICAGO HOME adlı bir kümede yer almakta ve her ikisi de tam havuzları tutmaktadır. NEWYORK, LONDON ve PARIS, INVENTORY adlı başka bir kümede. ŞIKAGO ' nun MQIPTgerekirmeyen bir kümede olduğu için SOCKS etkin olması gerekmediğini unutmayın.

INVENTORY kümesindeki her kuyruk yöneticisi etkili bir şekilde bir MQIPTarkasında "gizlenir". Kuyruk yöneticisi SOCKS etkin olduğundan, bir küme gönderen kanalı başlatıldığında istek, MQIPT SOCKS yetkili sunucusu olarak hareket ederek hedefine gönderilir. Olağan durumda, bir küme-alıcı kanalındaki CONNAME yerel kuyruk yöneticisini tanımlamak için kullanılır, ancak MQIPTile kullanıldığında CONNAME, yerel MQIPT ve gelen dinleyici kapısını tanımlıdır. Aşağıdaki şemada, tüm gelen dinleyici kapı adresleri 1414, giden dinleyici kapı adresleri 1415 'tir.

SOCKS etkin bir kuyruk yöneticisini çalıştırmanın iki yolu vardır. Birincisi, kuyruk yöneticisinin çalıştığı tüm bilgisayarı etkinleştirmek için SOCKS kullanılır. İkincisi, SOCKS ' un yalnızca kuyruk yöneticisini etkinleştirmesi. Her iki yöntemi de kullanarak, SOCKS istemcisini, yalnızca SOCKS yetkili sunucusu olarak MQIPT kullanarak uzak bağlantılar kurması ve kullanıcı kimlik doğrulamasını devre dışı bırakması için yapılandırmanız gerekir. SOCKS desteği almak için piyasada bir dizi ürün vardır. SOCKS V5 protokolünü destekleyen bir protokol seçmelisiniz.

Bir küme ağının konfigürasyonunun nasıl tanımlanacağına ilişkin bir örnek için [MQIPT kümeleme desteğinin yapılandırılması](#) başlıklı konuya bakın.





## MQIPT içinde SSL/TLS desteği

İletişim gizliliğini, iletişim bütünlüğünü ve kimlik doğrulamasını sağlamak için güvenli yuvalar kullanılabilir.

### İletişim gizliliği

Bağlantı özel yapılabilir. İstemci ile sunucu arasında değiş tokuş edilecek veriler şifrelenebilir ve yalnızca gönderen ve alıcı verileri anlayabilir. Bu, kredi kartı numaraları gibi özel bilgilerin güvenli bir şekilde aktarılabilceği anlamına gelir.

### İletişim bütünlüğü

Bağlantı güvenilirdir. İletim, güvenli bir hash işlevine dayalı bir ileti bütünlüğü denetimi içerir.

### Kimlik Doğrulaması

İstemci sunucunun kimliğini doğrulayabilir ve kimliği doğrulanmış bir sunucu istemcinin kimliğini doğrulayabilir. Bu, bilgilerin yalnızca amaçlanan taraflar arasında değiş tokuş edilmesinin garanti edildiği anlamına gelir. Kimlik doğrulama mekanizması, dijital sertifikaların (X.509v3 sertifikaları) değiş tokuşuna dayalıdır.

## Güvenli yuva protokolleri

MQIPTiçinde, güvenli yuvalar TLS (Transport Layer Security; İletim Katmanı Güvenliği) ve SSL (Secure Sockets Layer; Güvenli Yuva Katmanı) protokolleri kullanılarak sağlanır. İki güvenli yuva protokolü benzerdir, ancak birlikte çalışmazlar. Bu belgede, belirli bir fark belirtilmedikçe SSL ve TLS terimleri birbirinin yerine kullanılır.

MQIPT , sağlanan Java runtime environment (JRE) tarafından sağlanan SSL 3.0, TLS 1.0, TLS 1.1ve TLS 1.2 ' yi destekler. **V9.3.0** IBM MQ 9.3.0olanağından, MQIPT TLS 1.3' ünü de destekler. Uzak kanalın IBM MQ CipherSpec , MQIPT ' un hangi protokolün kullandığını belirler.

SSL 3.0, TLS 1.0ve TLS 1.1 güvenli değildir ve MQIPTiçinde varsayılan olarak devre dışıdır. Bu geçersiz kılınmış iletişim kurallarından herhangi birini kullanmanız gerekirse, bunlar “MQIPT içinde kullanımdan kaldırılan iletişim kurallarını ve şifreleme takımlarını etkinleştirme” sayfa 1036içindeki yordamı izleyerek yeniden etkinleştirilebilir.

SSL/TLS protokolleri, iletişim taraflarının kimlik doğrulaması için farklı dijital imza algoritmaları kullanabilir. SSL/TLS ' de kullanılan şifreleme işlemleri, veri gizliliği için şifreleme ve ileti bütünlüğü için güvenli hash algoritması, istemci ile sunucu arasında gizli anahtarların paylaşılmasına dayanır. SSL/ TLS, gizli anahtarların paylaşılmasına izin veren çeşitli anahtar değişim mekanizmaları sağlar. SSL/TLS, şifreleme ve hash algoritmaları için çeşitli algoritmalarından yararlanabilir.

## MQIPT içinde FIPS kipinin etkinleştirilmesi

JRE ' nin SSL/TLS şifreleme bileşeni, FIPS 140-2 standardıyla uyumlu sertifikalı IBMJCEPlusFIPS güvenlik sağlayıcısını içerir. MQIPTiçinde yalnızca FIPS onaylı şifreleme kullanmak istiyorsanız, MQIPT başlatıldığında aşağıdaki Java sistem özelliklerini ayarlayarak IBMJSSE2 sağlayıcısında FIPS kipini etkinleştirin:

- `com.ibm.jsse2.usefipsprovider=true`
- **V9.3.0** `com.ibm.jsse2.usefipsProviderName=IBMJCEPlusFIPS`

**MQIPT\_JVM\_OPTIONS** ortam değişkenini kullanarak MQIPT başlatıldığında Java sistem özelliklerini ayarlayabilirsiniz. Örneğin, Linux'ta MQIPT' ı başlatmak üzere komutu vermeden önce ortam değişkenini ayarlamak için aşağıdaki komutu verin:

```
export MQIPT_JVM_OPTIONS="-Dcom.ibm.jsse2.usefipsprovider=true  
-Dcom.ibm.jsse2.usefipsProviderName=IBMJCEPlusFIPS"
```

FIPS kipinin etkinleştirilmesiyle ilgili daha fazla bilgi için bkz. [IBMJSSE2 Sağlayıcısında FIPS Kipinin Etkinleştirilmesi](#).

## SSL/TLS Köprme Kipi

Bir rotanın hem SSLServer hem de SSLClient kümesi varsa, MQIPT gelen bir SSL/TLS güvenli bağlantısını kabul eder ve başka bir MQIPT ya da hedef kuyruk yöneticisine ikinci bir SSL/TLS güvenli bağlantısı kurar. IBM MQ kanal bilgilerinin şifresi çözülür ve bu iki SSL/TLS bağlantısı arasında yeniden şifrelenir. SSL/TLS köprüleme, *SSL/TLS sonlandırma yetkili sunucusu* olarak da adlandırılır.

IBM MQ , MQIPTkullanarak SSL/TLS köprüsünü destekler. Yetkili sunucu, IBM MQtarafından gönderilenden farklı boyutlarda SSL/TLS kayıtlarını birleştirir ya da yeniden yapılandırır, IBM MQ içeren diğer SSL/TLS sonlandırma yetkili sunucularının bozuk bağlantılara neden olduğu gözlemlenmiştir. Bunun nedeni, kuyruk yöneticilerinin gelen IBM MQ ağ verileri için bellek ayırma ve yönetme yöntemi ile IBM MQ ağ verilerinin SSL/TLS kayıtlarına paketleniş şekli arasındaki etkileşimdir.

MQIPT , IBM MQ ağ verilerinin bölünmeden ya da birleştirilmeden SSL/TLS kayıtlarında paketlenmesini korur. Diğer SSL/TLS köprüleri SSL/TLS kayıtlarını tam olarak korumazsa, IBM MQ kanallarının hata iletileriyle başarısız olmasına neden olabilir:

```
AMQ9638: SSL communications error for channel  
AMQ9208: Error on receive from host
```

## SSL/TLS Yetkili Sunucu Kipi

Bir MQIPT rotası, SSL/TLS yetkili sunucu kipinde SSL/TLS köprüsüne alternatif olarak yapılandırılabilir. Bu kipte rota yalnızca iki IBM MQ uç noktası arasındaki SSL/TLS verilerini iletir; SSL/TLS el sıkışmasına katılmaz ve herhangi bir dijital sertifika gerektirmez.

SSL/TLS yetkili sunucu kipini, MQIPT aracılığıyla iletişim kuran IBM MQ kanallarının SSL/TLS iletişimi için önceden yapılandırılmış olduğu ve MQIPT ' i güvenlik duvarları üzerinden bağlantıları yönlendirme ya da bir güvenlik çıkışı aracılığıyla izin verilen bağlantı kümesini kısıtlama gibi başka bir amaçla kullanmak istediğiniz durumlarda kullanabilirsiniz. SSL/TLS yetkili sunucu kipinde çalışırken MQIPT , paketleri hedefe iletmeden önce yeni bir bağlantıdan alınan ilk SSL/TLS paketlerinin geçerli olup olmadığını denetler.

IBM MQ , MQIPT ya da diğer herhangi bir SSL/TLS yetkili sunucusu ile SSL/TLS yetkili sunucu kipini destekler

## IBM MQ MQIPT ile birden çok sertifika desteği

IBM MQ 8.0ve daha sonra, kanal tanımlamasında **CERTLABL** özniteliği kullanılarak belirtilen kanal başına sertifika etiketi kullanılarak, aynı kuyruk yöneticisinde birden çok sertifika kullanılmasını destekler. Kuyruk yöneticisine gelen kanallar (örneğin, sunucu bağlantısı ya da alıcı), kuyruk yöneticisinden doğru sertifikayı sunmak için TLS Sunucu Adı Göstergesi 'ni (SNI) kullanarak kanal adının saptanmasına güvenir. Bir kuyruk yöneticisinde birden çok sertifika kullanma hakkında daha fazla bilgi için bkz. [Nasıl IBM MQ birden çok sertifika yeteneği sağlar.](#)

Bir kanal MQIPTüzerinden hedef kuyruk yöneticisine bağlanıyorsa ve MQIPT rotasında hem **SSLServer** hem de **SSLClient** ayarlanmışsa, uç noktalar arasında iki ayrı TLS oturumu vardır. IBM MQ 9.3.0' den önceki sürümlerde, SNI verileri oturum kesme sırasında akamaz. Bu, MQIPT ile kuyruk yöneticisi arasındaki TLS bağlantısı için hedef kuyruk yöneticisinde kanal başına bir sertifikanın kullanılmasını öner. Hedef kuyruk yöneticisinde kanal başına bir sertifika kullanmak için, MQIPT IBM MQ 9.3.0 sürümünden önceki bir sürümdekinden geçen bir TLS bağlantısı için MQIPT rotası, SNI adı da dahil olmak üzere tüm TLS denetim akışlarını bozulmadan ileden SSL/TLS Yetkili Sunucu Kipini kullanmalıdır.

**V 9.3.0** IBM MQ 9.3.0içinden MQIPT , TLS bağlantıları için SNI 'yi belirli bir değere ayarlamak ya da rotaya gelen bağlantıda alınan SNI' den geçmek için **SSLClientOutboundSNI** rota özelliği kullanılarak yapılandırılabilir. Kanal başına düşen sertifikaların bir hedef kuyruk yöneticisinde kullanılmasına izin vermek için, rotanın konfigürasyonu SNI 'yi IBM MQ kanal adına ayarlamak ya da rotaya gelen bağlantıda alınan SNI' den geçmek için tanımlanmalıdır. MQIPT SNI 'den geçecek şekilde yapılandırıldıysa, MQIPT ' e bağlanan kuyruk yöneticisi ya da istemci SNI ' yi kanal adına ayarlamalıdır.

MQIPT tarafından sonlandırılan ya da başlatılan TLS bağlantıları için kullanılan sertifikalar, her bir rota için ayrı ayrı yapılandırılabilir; örneğin, **SSLServerSiteLabel** ve **SSLClientSiteLabel** rota özellikleri kullanılarak.

## MQIPT tarafından desteklenen CipherSuites

Aşağıdaki tablo, MQIPT tarafından desteklenen ve varsayılan olarak etkinleştirilen CipherSuites ' i göstermektedir.

Varsayılan olarak yalnızca bir CipherSuites alt kümesi etkinleştirilir. Güvenli olmayan olarak kabul edilen birkaç algoritmaya dayalı CipherSuites JRE tarafından geçersiz kılınır. Olası tehlikeleri biliyorsanız, ancak yine de bu CipherSuitesürünlerinden birini kullanmanız gerekiyorsa, “MQIPT içinde kullanımdan kaldırılan iletişim kurallarını ve şifreleme takımlarını etkinleştirme” sayfa 1036başlıklı konudaki yordamı izleyerek devre dışı bırakılmış bir CipherSuite için destek ekleyebilirsiniz.

Çizelge 70. MQIPT ile kullanabileceğiniz CipherSuites	
CipherSuite	Varsayılan olarak etkindir
CipherSuites for TLS 1.3	

Çizelge 70. MQIPT ile kullanabileceğiniz CipherSuites (devamı var)

CipherSuite	Varsayılan olarak etkindir
▶ V9.3.0 ▶ V9.3.0 TLS_AES_128_GCM_SHA256	Evet
▶ V9.3.0 ▶ V9.3.0 TLS_AES_256_GCM_SHA384	Evet
▶ V9.3.0 ▶ V9.3.0 TLS_CHACHA20_POLY1305_SHA256	Evet
<b>CipherSuites for SSL 3.0, TLS 1.0, TLS 1.1 ve TLS 1.2</b>	
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA	
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	
SSL_DH_anon_WITH_AES_128_CBC_SHA	
SSL_DH_anon_WITH_AES_128_CBC_SHA256	
SSL_DH_anon_WITH_AES_128_GCM_SHA256	
SSL_DH_anon_WITH_AES_256_CBC_SHA	
SSL_DH_anon_WITH_AES_256_CBC_SHA256	
SSL_DH_anon_WITH_AES_256_GCM_SHA384	
SSL_DH_anon_WITH_DES_CBC_SHA	
SSL_DH_anon_WITH_RC4_128_MD5	
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	
SSL_DHE_DSS_WITH_AES_128_CBC_SHA	Evet
SSL_DHE_DSS_WITH_AES_128_CBC_SHA256	Evet
SSL_DHE_DSS_WITH_AES_128_GCM_SHA256	Evet
SSL_DHE_DSS_WITH_AES_256_CBC_SHA	Evet
SSL_DHE_DSS_WITH_AES_256_CBC_SHA256	Evet
SSL_DHE_DSS_WITH_AES_256_GCM_SHA384	Evet
SSL_DHE_DSS_WITH_DES_CBC_SHA	
SSL_DHE_DSS_WITH_RC4_128_SHA	
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_DHE_RSA_WITH_AES_128_CBC_SHA	Evet
SSL_DHE_RSA_WITH_AES_128_CBC_SHA256	Evet
SSL_DHE_RSA_WITH_AES_128_GCM_SHA256	Evet
SSL_DHE_RSA_WITH_AES_256_CBC_SHA	Evet
SSL_DHE_RSA_WITH_AES_256_CBC_SHA256	Evet
SSL_DHE_RSA_WITH_AES_256_GCM_SHA384	Evet

Çizelge 70. MQIPT ile kullanabileceğiniz CipherSuites (devamı var)

CipherSuite	Varsayılan olarak etkindir
SSL_DHE_RSA_WITH_DES_CBC_SHA	
SSL_ECDH_anon_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_anon_WITH_AES_128_CBC_SHA	
SSL_ECDH_anon_WITH_AES_256_CBC_SHA	
SSL_ECDH_anon_WITH_NULL_SHA	
SSL_ECDH_anon_WITH_RC4_128_SHA	
SSL_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA	Evet
SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	Evet
SSL_ECDH_ECDSA_WITH_AES_128_GCM_SHA256	Evet
SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA	Evet
SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA384	Evet
SSL_ECDH_ECDSA_WITH_AES_256_GCM_SHA384	Evet
SSL_ECDH_ECDSA_WITH_NULL_SHA	
SSL_ECDH_ECDSA_WITH_RC4_128_SHA	
SSL_ECDH_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_RSA_WITH_AES_128_CBC_SHA	Evet
SSL_ECDH_RSA_WITH_AES_128_CBC_SHA256	Evet
SSL_ECDH_RSA_WITH_AES_128_GCM_SHA256	Evet
SSL_ECDH_RSA_WITH_AES_256_CBC_SHA	Evet
SSL_ECDH_RSA_WITH_AES_256_CBC_SHA384	Evet
SSL_ECDH_RSA_WITH_AES_256_GCM_SHA384	Evet
SSL_ECDH_RSA_WITH_NULL_SHA	
SSL_ECDH_RSA_WITH_RC4_128_SHA	
SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	Evet
SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	Evet
SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	Evet
SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	Evet
SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	Evet
SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	Evet
SSL_ECDHE_ECDSA_WITH_NULL_SHA	
SSL_ECDHE_ECDSA_WITH_RC4_128_SHA	
SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	

Çizelge 70. MQIPT ile kullanabileceğiniz CipherSuites (devamı var)

<b>CipherSuite</b>	<b>Varsayılan olarak etkindir</b>
SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA	Evet
SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256	Evet
SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256	Evet
SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA	Evet
SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384	Evet
SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384	Evet
SSL_ECDHE_RSA_WITH_NULL_SHA	
SSL_ECDHE_RSA_WITH_RC4_128_SHA	
SSL_KRB5_EXPORT_WITH_DES_CBC_40_MD5	
SSL_KRB5_EXPORT_WITH_DES_CBC_40_SHA	
SSL_KRB5_EXPORT_WITH_RC4_40_MD5	
SSL_KRB5_EXPORT_WITH_RC4_40_SHA	
SSL_KRB5_WITH_3DES_EDE_CBC_MD5	
SSL_KRB5_WITH_3DES_EDE_CBC_SHA	
SSL_KRB5_WITH_DES_CBC_MD5	
SSL_KRB5_WITH_DES_CBC_SHA	
SSL_KRB5_WITH_RC4_128_MD5	
SSL_KRB5_WITH_RC4_128_SHA	
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	
SSL_RSA_EXPORT_WITH_RC4_40_MD5	
SSL_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_RSA_WITH_AES_128_CBC_SHA	Evet
SSL_RSA_WITH_AES_128_CBC_SHA256	Evet
SSL_RSA_WITH_AES_128_GCM_SHA256	Evet
SSL_RSA_WITH_AES_256_CBC_SHA	Evet
SSL_RSA_WITH_AES_256_CBC_SHA256	Evet
SSL_RSA_WITH_AES_256_GCM_SHA384	Evet
SSL_RSA_WITH_DES_CBC_SHA	
SSL_RSA_WITH_NULL_MD5	
SSL_RSA_WITH_NULL_SHA	
SSL_RSA_WITH_NULL_SHA256	
SSL_RSA_WITH_RC4_128_MD5	Evet
SSL_RSA_WITH_RC4_128_SHA	

Çizelge 70. MQIPT ile kullanabileceğiniz CipherSuites (devamı var)

CipherSuite	Varsayılan olarak etkindir
> V9.3.0 TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	
> V9.3.0 TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	
> V9.3.0 TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	

## CipherSpecs ve MQIPT CipherSuites

Aşağıdaki tablo, IBM MQ tarafından desteklenen CipherSpecs ile MQIPT tarafından desteklenen CipherSuites arasındaki ilişkiyi göstermektedir.

Çizelge, IBM MQ 'in her bir CipherSpec ' in kullanmasını beklediği protokol sürümünü de gösterir.

IBM MQ CipherSpec , hem şifreleme algoritmasını hem de kullanılacak güvenli yuva iletişim kuralı sürümünü benzersiz olarak belirler. Bazı IBM MQ CipherSpecs yalnızca protokol sürümüne göre farklılık gösterir; bu nedenle, CipherSuite ' i tek başına yapılandırmak yeterli değildir. SSL/TLS anlaşması, her iki taraf tarafından desteklenen en yüksek güvenli yuva iletişim kuralı sürümünü kararlaştırır ve daha sonra, karşılıklı olarak etkinleştirilmiş şifreler kümesinden bir CipherSuite seçer.

Örneğin, SSLClientCipherSuites=SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA ile bir SSLClient rotası, uzak kuyruk yöneticisiyle TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA (TLS 1.0) ya da TRIPLE\_DES\_SHA\_US (SSL 3.0) ile anlaşabilir. Aslında bu CipherSuite ile TLS 1.2 üzerinden anlaşılabilir, ancak IBM MQ bu CipherSuite ' i TLS 1.2 üzerinden desteklemez. Bu nedenle, SSLClient rotaları özellikle kuyruk yöneticisinde AMQ9616 ya da AMQ9631 hatalarına neden olabilir.

SSLClient rotalarında bu tür hataları önlemek için **SSLClientProtocols** rota özelliğini, amaçlanan CipherSpec için uygun değere ayarlayın. Bazı durumlarda, **SSLServerProtocols** rota özelliği kullanılarak ayarlanan sunucu tarafı protokolünün kısıtlanması da gerekebilir. Bu rota özelliklerine ilişkin doğru ayarı belirlemek için çizelgede gösterilen protokol sürümünü kullanın.

Bu sorun özellikle SSLClient rotaları için aşağıdaki CipherSuites ve CipherSpecs öğelerini etkiler:

- SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA:
  - SSL 3.0: MQ CipherSpec TRIPLE\_DES\_SHA\_US
  - TLS 1.0: MQ CipherSpec TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- SSL\_RSA\_WITH\_DES\_CBC\_SHA:
  - SSL 3.0: MQ CipherSpec DES\_SHA\_EXPORT
  - TLS 1.0: MQ CipherSpec TLS\_RSA\_WITH\_DES\_CBC\_SHA
- SSL\_RSA\_WITH\_RC4\_128\_SHA, karşılık gelen:
  - SSL 3.0: MQ CipherSpec RC4\_SHA\_US
  - TLS 1.2: MQ CipherSpec TLS\_RSA\_WITH\_RC4\_128\_SHA256

Farklı CipherSpecskullanan birden çok IBM MQ kanalını tünele almak için tek bir MQIPT SSLClient rotası kullanmak istiyorsanız, tüm kanalların birbiriyle aynı güvenli yuva iletişim kuralı sürümünü kullanan CipherSpecs ' e sahip olduğundan ve **SSLClientProtocols** ' i bu tek iletişim kuralı sürümünü kullanacak şekilde ayarladığınızdan emin olun.

IBM MQ CipherSpecs ile ilgili daha fazla bilgi için CipherSpecs başlıklı konuya bakın.

IBM MQ CipherSpec	MQIPT CipherSuite	Protokol sürümü
DES_SHA_EXPORT (DIŞA AKTARMA)	SSL_RSA_WITH_DES_CBC_SHA	SSLv3

IBM MQ CipherSpec	MQIPT CipherSuite	Protokol sürümü
DES_SHA_EXPORT1024	Yok	Yok
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	TLSv1.2
ECDHE_ECDSA_AES_128_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	TLSv1.2
ECDHE_ECDSA_AES_128_GCM_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	TLSv1.2
ECDHE_ECDSA_AES_256_CBC_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	TLSv1.2
ECDHE_ECDSA_AES_256_GCM_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLSv1.2
ECDHE_ECDSA_NULL_SHA256	SSL_ECDHE_ECDSA_WITH_NULL_SHA	TLSv1.2
ECDHE_ECDSA_RC4_128_SHA256	SSL_ECDHE_ECDSA_WITH_RC4_128_SHA	TLSv1.2
ECDHE_RSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1.2
ECDHE_RSA_AES_128_CBC_SHA256	SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLSv1.2
ECDHE_RSA_AES_128_GCM_SHA256	SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256	TLSv1.2
ECDHE_RSA_AES_256_CBC_SHA384	SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLSv1.2
ECDHE_RSA_AES_256_GCM_SHA384	SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384	TLSv1.2
ECDHE_RSA_NULL_SHA256	SSL_ECDHE_RSA_WITH_NULL_SHA	TLSv1.2
ECDHE_RSA_RC4_128_SHA256	SSL_ECDHE_RSA_WITH_RC4_128_SHA	TLSv1.2
NULL_MD5	SSL_RSA_WITH_NULL_MD5	SSLv3
NULL_SHA	SSL_RSA_WITH_NULL_SHA	SSLv3
RC2_MD5_EXPORT	Yok	Yok
RC4_56_SHA_EXPORT1024	Yok	Yok
RC4_MD5_EXPORT	SSL_RSA_EXPORT_WITH_RC4_40_MD5	SSLv3
RC4_MD5_US	SSL_RSA_WITH_RC4_128_MD5	SSLv3
RC4_SHA_US	SSL_RSA_WITH_RC4_128_SHA	SSLv3
<b>&gt; V9.3.0 &gt; V9.3.0</b> TLS_AES_128_GCM_SHA256	TLS_AES_128_GCM_SHA256	TLSv1.3
<b>&gt; V9.3.0 &gt; V9.3.0</b> TLS_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384	TLSv1.3
<b>&gt; V9.3.0 &gt; V9.3.0</b> TLS_CHACHA20_POLY1305_SHA256	TLS_CHACHA20_POLY1305_SHA256	TLSv1.3
TLS_RSA_WITH_3DES_EDE_CBC_SHA	SSL_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA	SSL_RSA_WITH_AES_128_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA256	SSL_RSA_WITH_AES_128_CBC_SHA256	TLSv1.2
TLS_RSA_WITH_AES_128_GCM_SHA256	SSL_RSA_WITH_AES_128_GCM_SHA256	TLSv1.2



IBM MQ CipherSpec	MQIPT CipherSuite	Protokol sürümü
TLS_RSA_WITH_AES_256_CBC_SHA	SSL_RSA_WITH_AES_256_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_256_CBC_SHA256	SSL_RSA_WITH_AES_256_CBC_SHA256	TLSv1.2
TLS_RSA_WITH_AES_256_GCM_SHA384	SSL_RSA_WITH_AES_256_GCM_SHA384	TLSv1.2
TLS_RSA_WITH_DES_CBC_SHA	SSL_RSA_WITH_DES_CBC_SHA	TLSv1
TLS_RSA_WITH_NULL_NULL (boş değerli)	Yok	Yok
TLS_RSA_WITH_NULL_SHA256	SSL_RSA_WITH_NULL_SHA256	TLSv1.2
TLS_RSA_WITH_RC4_128_SHA256	SSL_RSA_WITH_RC4_128_SHA	TLSv1.2
TRIPLE_DES_SHA_US	SSL_RSA_WITH_3DES_EDE_CBC_SHA	SSLv3

## MQIPT içinde SSL/TLS el sıkışması

SSL/TLS el sıkışması işlemi, CipherSuites kimlik doğrulaması ve anlaşması gerçekleştirildiğinde SSL/TLS istemcisi ve sunucusu arasındaki ilk bağlantı isteği sırasında oluşur.

Desteklenen tüm SSL/TLS CipherSuites (bkz. “MQIPT içinde SSL/TLS desteği” sayfa 1013), anonim CipherSuites dışında, sunucu kimlik doğrulamasını gerektirir ve istemci kimlik doğrulamasına izin verir; sunucu istemci kimlik doğrulaması isteyecek şekilde yapılandırılabilir. Uzak eşin kimliğine ilişkin hiçbir garanti vermedikleri için anonim CipherSuites kullanmaktan kaçınmalısınız. Bir ortadaki adam saldırısının anonim SSL/TLS bağlantılarını sizin bilginiz olmadan kesmesi mümkündür. Yalnızca güvenilir iç ağlarda ve yalnızca veri önleme riskini kabul etmeye hazırsanız anonim CipherSuites kullanın.

SSL/TLS iletişim eşdüzey kimlik doğrulaması, genel anahtar şifrelemesi ve X.509v3 dijital sertifikalarına dayalıdır. SSL/TLS protokolünde kimliği doğrulanması gereken bir site, sertifikanın geçerlilik süresi olan bir özel anahtar ve bir dijital sertifika (sitenin kimliğiyle ilgili bilgilerle birlikte karşılık gelen genel anahtarı içeren) gerektirir. Sertifikalar bir Sertifika Yetkilisi tarafından imzalanır, bu yetkililerin sertifikalarına imzalayıcı sertifikaları denir. Bir veya daha fazla imzalayıcı sertifikasının izlediği bir sertifika, bir sertifika zincirini oluşturur. Bir sertifika zinciri, ilk sertifikadan (site sertifikası) başlayarak, zincirdeki her sertifikanın imzasının bir sonraki imzalayıcı sertifikasında bulunan genel anahtar kullanılarak doğrulanabileceği gerçeğiyle karakterizedir.

Sunucu kimlik doğrulaması gerektiren güvenli bir bağlantı kurulduğunda, sunucu, kimliğini kanıtlamak için istemciye bir sertifika zinciri gönderir. SSL/TLS istemcisi, yalnızca sunucunun kimliğini doğrulayabiliyorsa (örneğin, sunucunun site sertifikasının imzasını doğrulayabiliyorsa) sunucuyla bağlantı kurulmasına devam eder. İmzanın doğrulanması için SSL/TLS istemcisinin sunucu sitesinin kendisine ya da sunucu tarafından sağlanan sertifika zincirindeki imzalayıcılardan en az birine güvenmesi gerekir. Bu doğrulamayı gerçekleştirmek için, güvenilir sitelerin ve imzalayıcıların sertifikalarının istemci tarafında tutulması gerekir.

SSL/TLS istemcisi, site sertifikasıyla başlayarak sunucunun sertifika zincirini inceler. İstemci, aşağıdaki durumlarda site sertifikasının imzasının geçerli olduğunu kabul eder:

- Site sertifikası, güvenilir site ya da imzalayıcı sertifikalarının havuzunda
- Zincirdeki bir imzalayıcı sertifikası, güvenilir imzalayıcı sertifikaları havuzuna dayalı olarak doğrulanabilir

İkinci durumda, SSL/TLS istemcisi, güvenilir imzalayıcı sertifikasından sunucunun site sertifikasına kadar sertifika zincirinin gerçekten doğru şekilde imzalanmış olup olmadığını denetler. Bu süreçte yer alan her sertifika, biçimin ve geçerlilik tarihlerinin doğruluğu için de incelenir. Bu denetimlerden herhangi biri başarısız olursa, sunucuyla bağlantı reddedilecektir. Sunucu sertifikasını doğruladıktan sonra istemci, SSL/TLS iletişim kuralının sonraki adımlarında bu sertifikaya gömülü genel anahtarı kullanır. SSL/TLS bağlantısı yalnızca sunucu gerçekten karşılık gelen özel anahtara sahipse kurulabilir.

İstemci kimlik doğrulaması aynı yordamı izler: bir SSL/TLS sunucusu istemci kimlik doğrulaması gerektiriyorsa istemci, kimliğini kanıtlamak için sunucuya bir sertifika zinciri gönderir. Sunucu, zinciri güvenilir site ve imzalayıcı sertifikaları havuzuna dayalı olarak doğrular. İstemcinin sertifikasını

doğruladıktan sonra sunucu, SSL/TLS iletişim kuralının sonraki adımlarında bu sertifikaya gömülü genel anahtarı kullanır. SSL/TLS bağlantısı yalnızca istemci gerçekten ilgili özel anahtara sahipse kurulabilir.

TLS iletişim kurallarının son sürümleri yüksek güvenli iletişim sağlar (SSL ve eski TLS iletişim kuralları güvensiz olarak kabul edilir). Ancak protokol, uygulama tarafından sağlanan bilgilere dayalı olarak çalışır. Yalnızca bu bilgi tabanı da güvenli bir şekilde korunursa, güvenli iletişimin genel hedefi elde edilebilir. Örneğin, güvenilir site ve imzalayıcı sertifikaları havuzunuz tehlikeye girdiyse, çok güvenli olmayan bir iletişim ortağıyla güvenli bir bağlantı kurabilirsiniz.

## MQIPT SSL/TLS uygulaması

SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2 ve TLS 1.3 , X509.V3 sertifikalarını içeren anahtarlık dosyalarında ( . p12 ya da . pfxdosya tipleriyle) saklanan Genel Anahtar Şifreleme Standartları (PKCS) #12 belirteçleriyle uygulanır. MQIPT , PKCS#11 Şifreleme Simgesi Arabirimi standardını destekleyen şifreleme donanımı anahtar depolarını da kullanabilir. MQIPT , IBM Java Secure Socket Extension (JSSE) paketini kullanır.

MQIPT , bağlantıyı başlatan uca bağlı olarak bir SSL/TLS istemcisi ya da bir SSL/TLS sunucusu olarak işlev görür. İstemci bir bağlantı başlatır ve sunucu bağlantı isteğini kabul eder. Bir MQIPT rotasının hem istemci hem de sunucu olarak işlev görmesi mümkündür. Bu durumda, SSL/TLS Yetkili Sunucu Kipi özelliğinin kullanılması genellikle daha iyi performans sağlar.

MQIPT , SSL/TLS Yetkili Sunucu Kipi için yapılandırıldığında, yalnızca iki uç nokta arasında SSL/TLS verilerini iletir; SSL/TLS el sıkışmasına katılmaz ve herhangi bir dijital sertifika gerektirmez.

IBM MQ 9.3.0' dan önceki sürümlerde MQIPT , bir gelen TLS bağlantısında alınan TLS Sunucu Adı Göstergesi (SNI) verilerini giden TLS bağlantısına geçmez. Bu, **CERTLABL** kanal özneliği kullanılarak belirtilen kanal başına sertifikaların MQIPT ile hedef kuyruk yöneticisi arasındaki TLS bağlantıları için kullanılmayacağı anlamına gelir. Hedef kuyruk yöneticisinde kanal başına bir sertifika kullanmak için, MQIPT IBM MQ 9.3.0 sürümünden önceki bir sürümde içinden geçen bir TLS bağlantısı için MQIPT rotası, SNI adı da dahil olmak üzere tüm TLS denetim akışlarını bozulmadan ileden SSL/TLS Yetkili Sunucu Kipini kullanmalıdır. **V9.3.0** IBM MQ 9.3.0'den MQIPT , TLS bağlantıları için SNI' yi belirli bir değere ayarlamak ya da rotaya gelen bağlantıda alınan SNI ' den geçmek üzere yapılandırılabilir. MQIPT ile bir kuyruk yöneticisinde birden çok sertifika kullanma hakkında daha fazla bilgi için bkz. ["IBM MQ MQIPT ile birden çok sertifika desteği"](#) sayfa 1015.

Her MQIPT rotası, kendi SSL/TLS özellikleri kümesiyle bağımsız olarak yapılandırılabilir. Daha fazla ayrıntı için bkz. [MQIPT rota özellikleri](#) .

## MQIPT içinde bir anahtarlık parolasını şifreleme

Bir anahtarlık dosyasını açmak ya da MQIPT tarafından kullanılan şifreleme donanımına erişmek için kullanılan parolayı **mqiptPW** komutuyla şifreleyin. Şifrelenmiş parola şu özelliklerden herhangi biri tarafından kullanılabilir: **SSLClientKeyRingPW**, **SSLClientCAKeyRingPW**, **SSLServerKeyRingPW**, **SSLServerCAKeyRingPW** ve **SSLCommandPortKeyRingPW**. Bu konuda, MQIPT tarafından kullanılmak üzere bir anahtarlık parolasını saklamanın doğru yolu açıklanmaktadır.

**mqiptkeyman** (iKeyman) saklama dosyası olanağı MQIPT tarafından desteklenmez. Parola saklama dosyası kullanmak yerine, şifrelenmiş parolayı saklamak için **mqiptPW** komutunu kullanmanız gerekir.

IBM MQ 9.1.5 sürümünden önceki sürümlerde, MQIPT tarafından kullanılacak anahtarlık parolaları, **SSL\*KeyRingPW** özelliklerinden herhangi biri tarafından başvuru dosyalarda depolanır.

IBM MQ 9.1.5' den **mqiptPW** komutunu kullanarak MQIPT tarafından kullanılacak anahtarlık parolalarını şifreleyin ve **SSL\*KeyRingPW** özelliklerinin değerini şifrelenmiş parolaya ayarlayın. MQIPT , IBM MQ 9.1.5 öncesinde oluşturulan yapılandırmalarla uyumluluk için özellik değerlerinde şifrelenmiş parolalar ile dosya adlarını ayırt edebilir.

**Deprecated** IBM MQ 9.1.5 sürümünden önceki MQIPT sürümlerinde bulunan anahtar deposu parolalarını şifreleme yöntemi kullanımdan kaldırılmıştır, ancak yine de kullanılabilir. Anahtarlık parolalarının korunmasını geliştirmek için, daha önce şifrelenmiş olan tüm anahtarlık parolalarını en son koruma yöntemini kullanarak yeniden şifreleyin.

Bir anahtarlık parolasını MQIPT tarafından kullanılmak üzere şifrelemek için [“MQIPT içinde saklanan parolaları şifreleme” sayfa 1048](#) içindeki adımları izleyin.

MQIPT kuruluş dizininin `samples/ssl` alt dizininde bulunan örnek anahtarlık dosyalarından birini açmak için `mqiptSample` parolasını kullanmanız gerekir.

## MQIPT içinde bir anahtarlık dosyasından sertifikaları seçme

Aynı anahtarlık dosyasında ya da şifreleme donanım simgesinde birden çok kişisel sertifikanın saklanması mümkündür. **SSLClientSite\*** özellikleri, kimlik doğrulaması için sunucuya gönderilecek sertifikayı seçmek üzere istemci tarafında kullanılabilir ve kimlik doğrulaması için istemciye gönderilecek sertifikayı seçmek için sunucu tarafında **SSLServerSite\*** özellikleri kullanılabilir.

Bu özellikleri kullanarak, sertifika Ayırt Edici Adına (DN) dayalı olarak seçilebilir. Diğer bir seçenek olarak, sertifika etiketi **SSLServerSiteLabel** ve **SSLClientSiteLabel** özelliklerini kullanarak bir sertifika seçmek için kullanılabilir.

TLS komut kapısı tarafından kullanılan sunucu sertifikasını seçmek için sertifikanın etiket adını belirtmek üzere **SSLCommandPortSiteLabel** özelliğini kullanın.

## MQIPT içindeki güven ayarları

Bir anahtarlık, imzalayıcı sertifikasını ya da imzalayıcı sertifikaları zincirini içeren kişisel bir sertifika içerir.

MQIPT tarafından kullanılan iki tip anahtar halkası vardır:

### Sertifika Yetkilisi (CA) anahtar halkası

Bu anahtarlık, uzak bir eşe ait sertifikaları doğrulamak için kullanılan güvenilir CA sertifikalarını içerir. Bu CA sertifikaları, uzak eşin güvenilir olup olmadığını belirlemeye yardımcı olur. MQIPT, CA sertifikalarını saklamak için hem PKCS #12 biçim anahtarlık dosyalarını hem de PKCS #11 arabirimini destekleyen şifreleme donanımı anahtar depolarını destekler. MQIPT CA anahtarlık dosyaları, **SSLClientCAKeyRing** ve **SSLServerCAKeyRing** rota özellikleriyle tanımlanır. CA sertifikalarına erişmek için şifreleme donanımının kullanılması, **SSLClientCAKeyRingUseCryptoHardware** ve **SSLServerCAKeyRingUseCryptoHardware** özellikleri ayarlanarak etkinleştirilir.

SSL/TLS istemci tarafındaki CA anahtar halkası, sunucudan gönderilen sertifikayı doğrulamak için kullanılacak güvenilir CA sertifikalarının bir listesini içermelidir. İstemci kimlik doğrulaması için bir SSL sunucu rotası yapılandırıldıysa, SSL/TLS sunucu tarafındaki CA anahtar halkası, istemciden gönderilen sertifikayı doğrulamak için kullanılacak güvenilir CA sertifikalarının bir listesini içermelidir.

### Kişisel sertifika anahtar halkası

Bu anahtarlık, MQIPT 'in kendisini uzak bir eşe tanıtmak için kullandığı kişisel sertifikaları içerir. Kendinden onaylı bir sertifika oluşturduğunuzda ya da sertifika yetkilisi tarafından imzalanmış bir sertifika istediğinizde, bunu kişisel sertifika anahtar halkasını kullanarak yapmanız gerekir. MQIPT, kişisel sertifikaların saklanması için hem PKCS #12 biçim anahtarlık dosyalarını hem de PKCS #11 arabirimini destekleyen şifreleme donanımı anahtar depolarını destekler. MQIPT içinde, kişisel sertifika anahtarlık dosyaları **SSLClientKeyRing** ve **SSLServerKeyRing** rota özellikleriyle tanımlanır. Kişisel sertifikalara erişmek için şifreleme donanımının kullanılması, **SSLClientKeyRingUseCryptoHardware** ve **SSLServerKeyRingUseCryptoHardware** özellikleri ayarlanarak etkinleştirilir.

SSL/TLS sunucu tarafındaki anahtar halkası, MQIPT sunucusunun kişisel sertifikasını içermelidir. SSL istemci rotasında istemci kimlik doğrulaması gerekiyorsa, SSL/TLS istemci tarafındaki anahtar halkası, istemcinin kişisel sertifikasını içermelidir.

İstemci kimlik doğrulamasına gereksinim duyarsanız, sunucu tarafında **SSLServerAskClientAuth** özelliğini etkinleştirmeniz gerekir. İstemci tarafındaki anahtarlık, istemcinin kişisel sertifikasını içermelidir. Sunucu tarafındaki, **SSLServerCAKeyRing** özelliği tarafından tanımlanan MQIPT anahtarlığı, istemcinin kimliğini doğrulamak için kullanılacak güvenilir CA sertifikalarının bir listesini içermelidir.

Bir rota için CA anahtar halkası yapılandırmazsanız, MQIPT bunun yerine kişisel sertifika anahtar halkasında CA sertifikalarını arar (yapılandırıldıysa). Örneğin, **SSLServerCAKeyRing** için herhangi bir değer ayarlanmazsa, MQIPT **SSLServerKeyRing** ile tanımlanan anahtarlık içinde CA sertifikalarını arar.

Güvenilir bir CA tarafından imzalanmış sertifikaları kullanmaya alternatif olarak, kendinden imzalı sertifikaları kullanabilirsiniz. `samples/ssl` alt dizinindeki MQIPT ile birlikte sağlanan `sslSample.pfx` örnek anahtarlık dosyasında kendinden onaylı bir sertifika örneği bulabilirsiniz. Örnek PKCS#12 anahtarlık dosyalarını açmak için `mqiptSample` parolasını kullanmalısınız.

Kendinden imzalı sertifikalar, bir sertifika için CA ödemedi SSL/TLS bağlantılığını sağlamanız gereken test senaryolarında yararlı olabilir. Ancak, üretim ortamlarında kendinden imzalı sertifikaları kullanmamalısınız. CA imzalı bir sertifika oluşturmak için bkz. [Anahtarlık dosyası oluşturma](#).

Dijital sertifikaları ve anahtar depolarını yönetmek için MQIPT ile birlikte sağlanan **mqiptkeyman** adlı bir yardımcı programı kullanabilirsiniz. Kuruluş yönergeleri ve ek bilgi için bkz. [“MQIPT içinde mqiptKeyman ve mqiptKeycmd” sayfa 1027](#).

Anahtarlık dosyalarını ve parola dosyalarını, bunlara yetkisiz erişimi önlemek için işletim sisteminin güvenlik özelliklerini kullanarak korumanız gerekir.

## MQIPT içinde SSL/TLS test ediliyor

SSL/TLS bağlantısını sınamanıza yardımcı olacak örnekler.

Çeşitli senaryoların açıklaması için bkz. [IBM MQ Internet Pass-Thru ile çalışmaya başlama](#). Özellikle aşağıdaki görevlere bakın:

- [SSL/TLS sunucusunun doğrulanması](#)
- [SSL/TLS istemcisinin kimliğini doğrulama](#)
- [MQIPT SSL/TLS yetkili sunucu kipinde çalıştırılıyor](#)
- [MQIPT ' i bir güvenlik yöneticisiyle SSL/TLS yetkili sunucu kipinde çalıştırma](#)

SSL/TLS yapılandırmanızın doğru çalıştığını test etmek için kendinden onaylı sertifikaları kullanabilirsiniz. Kendinden onaylı sertifikalar, bir sertifika için Sertifika Yetkilisi (CA) ödemedi SSL/TLS bağlantılığını sağlayabilmek için test senaryolarında kullanışlıdır. Ayrıntılar için [Test sertifikalarının yaratılması](#) başlıklı konuya bakın.

`samples/ssl` alt dizinindeki MQIPT ile birlikte sağlanan `sslSample.pfx` örnek anahtarlık dosyasında kendinden onaylı bir sertifika örneği bulabilirsiniz. Örnek PKCS #12 anahtarlık dosyalarını açmak için `mqiptSample` parolasını kullanmalısınız. Örnek sertifika, test sırasında size kolaylık sağlamak için sağlanır. Ancak, örnek sertifikanın özel anahtarları tüm MQIPT kullanıcıları tarafından bilinir. Bu, güvensiz olduğu ve yalnızca bir test ortamında kullanılması gerektiği anlamına gelir.

Örnek sertifikalar olsun ya da olmasın, üretim ortamlarında kendinden imzalı sertifikaları kullanmamalısınız. Bunun yerine, güvenilir bir CA ' dan CA imzalı bir sertifika edinin. CA imzalı bir sertifika oluşturmak için bkz. [Anahtarlık dosyası oluşturma](#).

Bir sertifika oluştururken ya da isterken, güvenlik gereksinimlerinize uygun anahtar tipi, anahtar boyutu ve sayısal imza algoritmasını göz önünde bulundurmanız gerekir. Daha fazla bilgi için bkz. [“MQIPT için sayısal sertifikayla ilgili önemli noktalar” sayfa 1029](#).

Sertifikalar ve sertifika yönetimi teknolojileri bir dizi üçüncü kişi tedarikçiden edinilebilir.

## MQIPT içinde SSL/TLS hata iletileri

Tokalaşma hataları, MQIPT bağlantı günlüğüne JSSE kural dışı durumları biçiminde kaydedilir.

Daha fazla bilgi için bkz. [“MQIPT içinde bağlantı günlükleri” sayfa 1051](#). Aşağıdaki çizelge, farklı kural dışı durumları, olası nedeni ve hatayı çözmek için ilgili işlemi açıklar.

Sertifika kural dışı durumları genellikle bağlantının uzak ucundaki sertifikalarla ilgilidir.

Hatanın bir IBM MQ istemcisi ya da kuyruk yöneticisi sertifikasıyla ilgili olduğu durumlarda, *anahtarlık dosyası* terimi uzak ortağın IBM MQ anahtar havuzunu içerir.

MQIPT içinde CA sertifikaları, **SSLClientCAKeyRing** ve **SSLServerCAKeyRing** rota özellikleriyle tanımlanan CA anahtarlık dosyasında saklanır. CA anahtarlık rotası özellikleri ayarlanmazsa, bunun yerine

CA sertifikaları için ilgili kişisel anahtarlık dosyası ( **SSLClientKeyRing** ya da **SSLServerKeyRing** özelliği tarafından başvuru) aranır.

Kural dışı durum	Neden	Eylem
CertificateException	Sertifika, CA anahtar halkasında olmayan bir CA tarafından imzalandığı için güvenilir değil.	CA anahtarlık dosyasında gerekli tüm CA sertifikalarının olup olmadığını denetleyin. Güvenilir bir kaynaktan her CA sertifikasının bir kopyasını almak üzere eksik CA sertifikalarını eklemek için IBM MQIPT ile birlikte sağlanan Anahtar Yönetimi aracını kullanın.
CertificateExpiredKural Dışı Durumu	1. Sertifikanın süresi doldu: <b>notAfter</b> tarihi geçti. 2. Sistem saati yanlış ayarlandı.	1. Yeni bir sertifika edinin ve anahtarlık dosyasına ekleyin. Sertifika bir Sertifika Yetkilisine aitse, yeni sertifikayı CA anahtarlık dosyasına yerleştirin. 2. UTC sistem saatinin doğru saate ayarlandığını doğrulayın.
CertificateNotYetValidKural Dışı Durumu	1. Sertifika erken kullanılıyor: <b>notBefore</b> tarihi henüz gelmedi. 2. Sistem saati yanlış ayarlandı.	1. Sertifikanın doğru oluşturulup imzalanmadığını denetleyin. Kuruluşunuz kendi CA ' sını çalıştırıyorsa, CA için UTC sistem saati yanlış olabilir. 2. UTC sistem saatinin doğru saate ayarlandığını doğrulayın.
CertificateParsingÖzel Durumu	1. Sertifika geçersiz DER verileri içeriyor. 2. Sertifika desteklenmeyen DER özelliklerini kullanıyor.	Sertifikanın doğru oluşturulduğundan ve MQIPT ile birlikte sağlanan IBM Key Management aracında görüntülenebildiğinden emin olun. Daha az sertifika uzantısına sahip yeni bir sertifika edinmeyi düşünün.
CertificateRevokedKural Dışı Durumu	Sertifika iptal denetimi etkinleştirildi ve sertifikanın iptal edildiği saptandı.	Söz konusu sertifikaya güvenilmemelidir. Yeni bir sertifika edinin ve yeni sertifikanın ve özel anahtarının anahtarlık dosyasında bulunduğundan emin olun.
CertPathBuilderException	Sertifika zinciri, tanınan bir Sertifika Yetkilisi tarafından imzalanmadı.	1. CA imzalı sertifikalar kullanıyorsanız, CA anahtarlık dosyasında tüm kök CA ve ara CA sertifikalarının olup olmadığını denetleyin. 2. Kendinden onaylı sertifikalar kullanıyorsanız, uzak sertifikanın genel kısmının bir kopyasını çıkarıp CA anahtarlık dosyasına eklediğinizden emin olun. Üretim ortamlarında kendinden imzalı sertifikaları kullanmaktan kaçının.

Kural dışı durum	Neden	Eylem
CertStoreKural Dışı Durumu KeyStoreKural Dışı Durumu	<p>Aşağıdaki bir nedenden ötürü bir anahtar halkasından sertifika okunurken hata oluştu:</p> <ol style="list-style-type: none"> <li>1. Anahtarlık dosyası zarar görmüş.</li> <li>2. Anahtarlık dosyası eksik.</li> <li>3. Saklanan parola, anahtarlık dosyası parolasıyla eşleşmiyor.</li> <li>4. Rota, şifreleme donanımını kullanacak şekilde yapılandırıldıysa, MQIPT şifreleme donanımına bağlanamadı.</li> </ol>	<ol style="list-style-type: none"> <li>1. Anahtarlık dosyasının okunabildiğinden ve tüm sertifikaların IBM Key Management (Anahtar Yönetimi) aracılığıyla görüntülenebildiğinden emin olun.</li> <li>2. Tüm anahtarlık rotası özelliklerinin doğru dosya adını işaret edip etmediğini denetleyin.</li> <li>3. Saklanan anahtarlık dosyası parolasının doğru olup olmadığını denetleyin. Doğru parolayı saklamak için <b>mqiptPW</b> aracını kullanın.</li> <li>4. Rota, şifreleme donanımını kullanacak şekilde yapılandırıldıysa, aşağıdakileri denetleyin: <ul style="list-style-type: none"> <li>• Java güvenlik özellikleri dosyası, IBMPKCS11Impl güvenlik sağlayıcısının kurulu olduğunu belirtir.</li> <li>• Java güvenlik özellikleri dosyası, IBMPKCS11Impl güvenlik sağlayıcısını kullanıma hazırlamak için kullanılan yapılanış kütüğünün tam olarak nitelenmiş adını içerir.</li> <li>• IBMPKCS11Impl güvenlik sağlayıcısını kullanıma hazırlamak için kullanılan yapılanış kütüğü geçerli.</li> </ul> </li> </ol>
SSLException: Kullanılabilecek sertifika yok ya da anahtar, etkinleştirilen SSL şifreleme takımlarına karşılık geliyor.	<p>Kullanmakta olduğunuz CipherSuites için doğru anahtar tipine sahip bir kişisel sertifikanız olmalıdır. Örneğin, adları SSL_ECDH_ECDSA_ ile başlayan CipherSuites , Eliptik Eğri ortak anahtarı olan bir sertifika gerektirir. En sık kullanılan CipherSuites , RSA genel anahtarına sahip bir sertifika gerektirir.</p>	<p>Anahtarlık dosyasını IBM Key Management (Anahtar Yönetimi) aracılığıyla açın. Kişisel Sertifikalar görünümü altında, her bir sertifikayı sırayla seçin ve görüntüleyin. Genel anahtar türünü görmek için <b>Ayrıntıları Görüntüle</b> ' yi tıklayın ve Konu Genel Anahtarı bölümüne gidin. Daha sonra, uygun CipherSuites ' in etkinleştirildiğinden emin olmak için MQIPT <b>SSLClientCipherSuites</b> ve <b>SSLServerCipherSuites</b> rota özelliklerini denetleyin.</p>

Kural dışı durum	Neden	Eylem
SSLException: Ortak şifreleme takımı yok SSLHandshakeException: Ortak şifreleme takımı yok	<p>Bağlantının her iki ucunda etkinleştirilmiş CipherSuites kümeleri arasında çakışma olmadığından, anlaşma CipherSuite ' i kabul etmemiştir. Özellikle, bir giden IBM MQ bağlantısı yalnızca tek bir şifrelemeyi etkinleştirir; bu nedenle, SSLServer MQIPT rotaları özellikle bu hatayla karşılaşabilmektedir.</p> <p>Bu hata, aşağıdaki koşulların üçü de doğru olduğunda da oluşabilir:</p> <ul style="list-style-type: none"> <li>• rotada CipherSuite belirtilmedi</li> <li>• rota için yapılandırılan anahtar halkasında uygun bir site sertifikası bulunamadı</li> <li>• anonim CipherSuites devre dışı bırakıldı</li> </ul>	<p>MQIPT <b>SSLClientCipherSuites</b> ve <b>SSLServerCipherSuites</b> rota özelliklerinde etkin CipherSuites listesini denetleyin. Ek CipherSuites' i etkinleştirmeyi düşünün. Her bir IBM MQ kanal CipherSpec değeri için geçerli kılınacak doğru CipherSuites değerini saptamak için sağlanan çizelgeye bakın.</p> <p>Rotada CipherSuite belirtilmezse, anahtarlık rotası özelliklerinin doğru anahtarlık dosyasına başvurup başvurmadığını ve anahtarlık halkasının MQIPT ' in kullanabileceği kişisel bir sertifika içerip içermediğini denetleyin. Rota, şifreleme donanımını kullanacak şekilde yapılandırıldıysa, IBMPKCS11Impl güvenlik sağlayıcısını başlatmak için kullanılan yapılandırma dosyasındaki <b>tokenlabel</b> özniteliğinin doğru şifreleme aygıtı simgesi etiketini belirttiğini doğrulayın.</p>

## MQIPT içinde mqiptKeyman ve mqiptKeycmd

**mqiptKeyman** (iKeyman), IBM MQ kullanıcıları için önceden bilinen bir sertifika ve anahtar yönetimi uygulamasıdır. **mqiptKeyman** ve **mqiptKeycmd** komutları, IBM MQ Internet Pass-Thru tarafından kullanılan anahtarlık dosyalarında simetrik ve asimetrik anahtarları, dijital sertifikaları ve sertifika isteklerini yönetmek için kullanılabilir. Bu dosyalar, anahtarlık dosyalarını yönetmek için de kullanılabilir.

**mqiptKeyman** ve **mqiptKeycmd** komutları, bir anahtarlık dosyasına başvurmak için *anahtar veritabanı* terimini kullanır; bu terimler eşanlamlıdır.

iKeyman , grafik kullanıcı arabirimi (GUI) ve komut satırı arabirimi (CLI) olmak üzere iki kipte çalıştırılabilir. GUI 'yi başlatmak için **mqiptKeyman** komutunu ve CLI 'yi çalıştırmak için **mqiptKeycmd** komutunu kullanın.

IBM MQ içindeki sertifikaları yönetmeye ilişkin eşdeğer komutlar, GUI 'yi başlatmak için **strmqimk** ve CLI 'yi çalıştırmak için **runmqckm** komutlarıdır. IBM MQ komutları **runmqckm**, **runmqakm** ve **strmqimk** dijital sertifikaları yönetmek için kullanılabilecek kısımda açıklanmıştır.

**Not:** **V 9.3.4** **Deprecated** **mqiptKeycmd** ve **mqiptKeyman** araçları, IBM MQ 9.3.4' den kullanımdan kaldırılmıştır. Daha fazla bilgi için bkz. **runmqckm**, **runmqakm** ve **strmqimk** dijital sertifikaları yönetmek için kullanma.

## MQIPT için gerekli anahtarlık dosyası biçimi

MQIPT içinde kullanılmak üzere anahtarlık dosyaları oluştururken PKCS #12 dosya biçimini kullanmalısınız:

- Kullanıcı arabiriminde, anahtarlık dosyasını oluştururken **Anahtar veritabanı tipi** alanında PKCS#12 seçeneğini belirleyin.
- CLI 'de, mqiptKeycmd -keydb -create komutuna -type pkcs12 parametresini ekleyin.



MQIPT , PKCS #11 arabirimini destekleyen şifreleme donanımında saklanan sertifikalara da erişebilir. Arabirim, PKCS #11 donanımında sertifikaları yönetmek için de kullanılabilir. Daha fazla bilgi için bkz [“MQIPT içinde PKCS #11 şifreleme donanımının kullanılması” sayfa 1037.](#)

## MQIPT için anahtarlık parolasını şifreleme

Anahtarlık dosyasını oluşturduktan sonra, anahtarlık parolasını MQIPT ' in dosyaya erişmek için kullanabileceği bir biçimde şifrelemelisiniz. Bununla ilgili bilgi için bkz. [“MQIPT içinde bir anahtarlık parolasını şifreleme” sayfa 1022 .](#)

Saklama dosyası olanağının MQIPT tarafından desteklenmediğini unutmayın. Parola saklama dosyası kullanmak yerine anahtarlık parolasını şifrelemek için **mqiptPW** komutunu kullanmanız gerekir.

## Komut satırı örnekleri

CLI, IBM MQ **runmqckm** komutuyla aynı sözdizimini kullanır. Aşağıdaki örneklerde gösterildiği gibi, gerekli parametreleri **mqiptKeycmd** içine ekleyin:

- PKCS#12 dosyası yaratmak için:

```
mqiptKeycmd -keydb -create -db key.p12 -pw password -type pkcs12
```

- Sınama amacıyla kendinden imzalı bir kişisel sertifika yaratmak için:

```
mqiptKeycmd -cert -create -db key.p12 -pw password -type pkcs12  
-label mqipt -dn "CN=Test Certificate,OU=Sales,O=Example,C=US"  
-sig_alg SHA256WithRSA -size 2048
```

Komut, 2048 bit RSA genel anahtarı ve SHA-256 hash algoritması ile RSA kullanan bir sayısal imza ile bir sayısal sertifika oluşturur. Bir sertifika oluşturduğunuzda, kuruluşunuzun güvenlik gereksinimlerine uygun bir genel anahtar şifreleme algoritması, anahtar boyutu ve dijital imza algoritması seçmeye dikkat edin. Ek bilgi için bkz. [“MQIPT için sayısal sertifikayla ilgili önemli noktalar” sayfa 1029 .](#)

Bu örnek, test amaçlarına uygun kendinden imzalı bir sertifika kullanır. Ancak, üretim ortamında bunun yerine Sertifika Yetkilisi tarafından imzalanmış bir sertifika kullanmanız gerekir.

MQIPT v2.0 ve daha eski sürümlerin SHA-2 dijital imzalarını desteklemediğini, bu nedenle bu sertifikanın önceki MQIPT yayınlarına güvenli yuva bağlantıları kurmak için uygun olmadığını ve SHA1WithRSAGibi daha eski bir imza algoritmasının gerekli olduğunu unutmayın.

- Üretim amacıyla CA imzalı sertifika için bir sertifika isteği oluşturmak için:

```
mqiptKeycmd -certreq -create -db key.p12 -pw password -type pkcs12 -file cert.req  
-label mqipt -dn "CN=Test Certificate,OU=Sales,O=Example,C=US"  
-sig_alg SHA256WithRSA -size 2048
```

Komut, 2048 bitlik RSA ortak anahtarı ve SHA-256 hash algoritmasıyla RSA kullanan bir sayısal imza ile bir sayısal sertifika isteği oluşturur. Bir sertifika oluşturduğunuzda, kuruluşunuzun güvenlik gereksinimlerine uygun bir genel anahtar şifreleme algoritması, anahtar boyutu ve dijital imza algoritması seçmeye dikkat edin. Ek bilgi için bkz. [“MQIPT için sayısal sertifikayla ilgili önemli noktalar” sayfa 1029 .](#)

- CA imzalı kişisel sertifika dosyasını cert . crt anahtarlık dosyasına almak için:

```
mqiptKeycmd -cert -receive -db key.p12 -pw password -type pkcs12 -file cert.crt
```

Kişisel sertifikayı imzalayan CA ' nın CA sertifikasının CA anahtarlık dosyasında bulunduğundan emin olmanız gerekir, örneğin:

```
mqiptKeycmd -cert -add -db key.p12 -pw password -type pkcs12 -file ca.crt -label rootCA
```



## **MQIPT için sayısal sertifikayla ilgili önemli noktalar**

Dikkate alınacak noktalar arasında sertifika anahtarı boyutu, uygun bir sertifika dijital imza algoritması ve dijital sertifika seçimi ve CipherSuite compatibilityDigital sertifikası ve CipherSuite uyumluluğu yer alır.

## **MQIPT için sertifika anahtarı boyutuyla ilgili önemli noktalar**

Genel anahtar boyutu, kuruluşunuzun güvenlik ilkesine bağlıdır ve kullanılan şifreleme algoritmasına bağlıdır. Genel olarak, daha büyük anahtar boyutları daha güvenlidir. Aşağıdaki çizelge, kullanmanız gereken anahtar büyüklüğü alt sınırını listeler:

<b>Algoritma</b>	<b>Anahtar büyüklüğü alt sınırı (bit)</b>
Eliptik Eğri	256
RSA	2048

Bir sertifika ya da sertifika isteği yaratırken sertifikanız için anahtar boyutunu belirtin.

- **mqiptKeycmd** CLI komutu kullanılırken, **-size** parametresi anahtar boyutunu belirtir.
- **mqiptKeyman** GUI 'sini kullanırken Sertifika Oluşturma penceresindeki **Anahtar Boyutu** alanı anahtar boyutunu belirtir.

## **Uygun bir sertifika sayısal imza algoritması seçilmesi**

Dijital sertifikaların sahteciliğini önlemek için güçlü bir dijital imza algoritması kullanılması önemlidir. Bir sertifika yaratırken ya da istekte bulunduğunuzda, iyi bir algoritma seçmeye dikkat edin.

Bu algoritmalar artık modern kullanım için yeterince güvenli olmadığından, MD5 ya da SHA-1 tabanlı eski dijital imza algoritmalarını kullanmaktan kaçınmalısınız. Mümkünse, RSA (SHA256WithRSA) ile SHA-256 gibi daha yeni SHA-2 tabanlı sayısal imza algoritmalarından birini kullanın.

Ancak, MQIPT Sürüm 2.1 SHA-2 dijital imzalarını desteklemez; bu nedenle, önceki MQIPT yayınlarıyla birlikte çalışabilirlik için SHA1WithRSA dijital imza algoritmasını kullanın. Ancak, eski MQIPT sürümlerini yükseltmeyi ve MD5 ve SHA-1 dijital imzalarının aşamalı olarak kullanımını büyütmeyi planlamanız gerekir.

- **mqiptKeycmd** CLI komutu kullanılırken, **-sig\_alg** parametresi dijital imza algoritmasını belirtir.
- **mqiptKeyman** GUI kullanılırken Sertifika Oluşturma penceresinin **İmza Algoritması** alanı sayısal imza algoritmasını belirtir.

## **MQIPT içinde dijital sertifika ve CipherSuite uyumluluğu**

Tüm CipherSuites , tüm dijital sertifikalarla birlikte kullanılamaz. CipherSuite ad öneğine göre gruplanmış çeşitli CipherSuitetipleri vardır. Her CipherSuite tipi, kullanılabilecek sayısal sertifika tipine farklı kısıtlamalar getirir. Bu kısıtlamalar tüm MQIPT SSL/TLS bağlantıları için geçerlidir, ancak özellikle Eliptik Eğri şifreleme kullanıcıları için geçerlidir. Güvenli yuva el sıkışması gerçekleştirilirken MQIPT , kararlaştırılan CipherSuite için uygun olan kendini tanımlamak üzere otomatik olarak bir kişisel sertifika seçer. Çoğu durumda MQIPT , uzak eşle otomatik olarak birlikte çalışır. Ancak, bazı senaryolarda uzak bir IBM MQ sistemiyle birlikte çalışmak için belirli bir MQIPT CipherSuite kullanmanız gerekebilir. MQIPT ile sağlanan **mqiptKeyman** uygulaması, yalnızca DSA ve RSA ortak anahtarlarıyla sertifika ve sertifika istekleri oluşturma yeteneğine sahiptir. Buna ek olarak, IBM MQ **runmqacm** yardımcı programı Elliptic Curve ortak anahtarlarıyla sertifikalar ve sertifika istekleri oluşturabilir. Başka sertifika tipleri yaratılmasına ilişkin öneriler için Sertifika Yetkilisi 'ne başvurun.

Kullanılacak dijital sertifikanın tipi, kullandığınız CipherSuite tipine bağlıdır:

- **SSL\_ECDH\_ECDSA\_** ve **SSL\_ECDHE\_ECDSA\_** değerlerini başlatan CipherSuites , Eliptik Eğri ortak anahtarı olan bir sayısal sertifika gerektirir.
- **anon** içeren adlara sahip CipherSuites anonim olur; uzak eşin tanınması için sayısal sertifika gerekmez. Bu tür CipherSuites , alternatif bir kimlik doğrulama yönteminin kullanıldığı ağlarda sertifika yaşam

çevrimi yönetiminin ek giderlerini önleyebilir, ancak genel olarak kimlik doğrulama eksikliği nedeniyle bunların kullanımını önler.

- Diğer CipherSuites , RSA genel anahtarına sahip bir dijital sertifika gerektirir.

**Not:** `mqiPTKeyman` ve `mqiPTKeycmd` araçları, Eliptik Eğri ortak anahtarıyla sertifika ya da sertifika istekleri yaratamaz. Bu amaçla IBM MQ ile sağlanan `runmqakm` komutunu kullanabilirsiniz. `runmqakm` komutu, `runmqckm`, `runmqakmve` `strmqikm` dijital sertifikaları yönetmek için kullanılabilecek şekilde açıklanmaktadır.

## MQIPT içinde sertifika çıkışı

Bir sertifika çıkışının amacı, MQIPT tarafından alınan bir SSL/TLS eş sertifikasının doğrulanmasını sağlamaktır.

MQIPT rotasını, yeni bir bağlantı yaptığında SSL/TLS istemcisi olarak işlev görecektir ve bir bağlantı isteği aldığı anda SSL/TLS sunucusu olarak işlev görecektir şekilde yapılandırabilirsiniz. SSL/TLS el sıkışması işlemi sırasında, bir SSL/TLS istemcisi sunucudan bir eş sertifika alır ve sertifika, sunucunun kimliğini doğrulamak için kullanılabilir. Bir SSL/TLS sunucusu istemciden eş sertifika da alabilir ve sertifika istemcinin kimliğini doğrulamak için kullanılabilir.

MQIPT bir eş düzey sertifika aldığı anda, daha fazla doğrulama gerçekleştirmenize olanak tanıyan sertifika çıkışı çağırılır. Çıkışın yakaladığı kural dışı durumlar MQIPT tarafından yakalanır ve bağlantı isteği sonlandırılır. Bu nedenle, çıkışın tüm kural dışı durumları yakalaması ve MQIPT' e uygun bir dönüş kodu geçirmesi iyi bir uygulamadır.

Daha fazla bilgi için [SSL/TLS sunucusunu doğrulamak için sertifika çıkışının kullanılması](#) başlıklı konuya bakın.

**Not:** MQIPT tek bir Java Virtual Machine üzerinde çalışır; böylece, kullanıcı tanımlı bir sertifika çıkışı MQIPT ' in olağan çalışmasını aşağıdaki yollardan biriyle tehlikeye atabilir:

- Sistem kaynaklarını etkiler
- Darboğazlar oluşturur
- Performansı düşürebilirsiniz

Bir üretim ortamında uygulamadan önce, sertifika çıkışının etkilerini kapsamlı olarak test etmelisiniz.

### ***MQIPT içinde com.ibm.mq.ipt.exit.CertificateExit sınıfı***

`com.ibm.mq.ipt.exit.CertificateExit` sınıfı, `SSLExitName` özelliğiyle tanımlanan sınıf tarafından gerçekleştirilmesi gereken bir soyut sınıftır.

Sınıf, çıkışı çalıştırmaya ilişkin varsayılan uygulamaları ve isteğe bağlı olarak, gereksinimlerinize göre geçersiz kılabilirsiniz bazı genel yöntemleri içerir. Desteklenen yöntemlerin tam listesi aşağıdaki gibidir:

## yöntemler

### **public int init (IPTTrace)**

Çıkış MQIPT tarafından yüklendiğinde `init` yöntemi MQIPT tarafından çağırılır ve çıkışın herhangi bir başlatılmasını gerçekleştirmek için uygulanabilir; örneğin, doğrulama işlemi sırasında kullanılan verilerin yüklenmesi gibi. Varsayılan somutlama hiçbir şey yapmaz.

### **public int refresh (IPTTrace)**

Yenileme yöntemi, herhangi bir veriyi yenilemek için uygulanır; örneğin, doğrulama işlemi sırasında kullanılan diske ilişkin verilerin yeniden yüklenmesi gibi. Bu yöntem, MQIPT yöneticisi bir `refresh` komutu yayınladığında çağırılır. Varsayılan somutlama hiçbir şey yapmaz.

### **genel geçersiz kapatma (IPTTrace)**

Rota durdurulmak üzereyken ya da MQIPT kapatılırken bakım gerçekleştirmek için `kapatma` yöntemi uygulanır. Varsayılan somutlama hiçbir şey yapmaz.

## genel CertificateExitYanıt doğrulaması (IPTTrace)

Eşdüzey sertifikanın geçerliliğini denetlemek için doğrulama yöntemi çağrılır. Dönüş nesnesi, bilgileri MQIPT' e geri aktarmak için kullanılabilir; örneğin, bir dönüş kodu ve bağlantı günlüğüne eklenebilecek bir metin. Varsayılan uygulama, CertificateExitResponse.OK ile bir CertificateExitYanıtı döndürür.

Özellikleri elde etmek için desteklenen yöntemler:

### public int getListenerPort ()

ListenerPort özelliğinde tanımlandığı şekilde rota dinleyici kapısını alır

### public String getDestination()

hedef adresi alır-Hedef özelliği tarafından tanımlandığı şekilde

### public int getDestinationPort ()

DestinationPort özelliğinde tanımlandığı şekilde hedef dinleyici kapı adresini alır

### public String getClientIPAddress ()

bağlantı isteğinde bulunan istemcinin IP adresini alır

### public int getClientPortAddress()

bağlantı isteğinde bulunan istemci tarafından kullanılan kapı adresini alır

### genel boole isSSLClient()

Çıkışın bir SSL/TLS istemcisi ya da SSL/TLS sunucusu olarak çağrılıp çağrılmadığını belirlemek için kullanılır. Bu, true değerini döndürürse, çıkış bağlantısının istemci tarafında olur ve sunucudan alınan sertifikanın geçerliliğini denetler. Bu seçenek false değerini döndürürse, çıkış bağlantısının sunucu tarafındadır ve istemci tarafından gönderilen sertifikayı doğrular. Bir rotanın hem bir SSL/TLS sunucusu hem de bir SSL/TLS istemcisi olarak hareket etmesi, trafiğin şifresini çözmesi ve yeniden şifrelemesi için geçerlidir. Bu durumda, tek bir çıkış sınıfı olmasına rağmen, sınıfın bazı eşgörünümleri istemci olarak, bazıları sunucu olarak çağrılır. Belirli bir yönetim ortamına ilişkin durumu saptamak için isSSLClient istemcisini kullanabilirsiniz.

### public int getConnThreadID()

Bağlantı isteğini işleyen işçi iş parçacığının tanıtıcısını almak için kullanılır; bu, hata ayıklama için yararlı olabilir.

### public String getChannelName ()

Bağlantı isteğinde kullanılan IBM MQ kanal adını alır. Bu yalnızca gelen istek SSL/TLS kullanmıyorsa ve MQIPT bir SSL/TLS istemcisi olarak hareket ettiğinde kullanılabilir.

### public String getQMName()

Bağlantı isteğinde kullanılan IBM MQ kuyruk yöneticisinin adını alır. Bu yalnızca istemci isteği SSL/TLS kullanmıyorsa ve MQIPT bir SSL/TLS istemcisi olarak hareket ederken kullanılabilir.

### genel Boole getTimedout()

zaman aşımının süresinin dolup dolmadığını saptamak için çıkış tarafından kullanılır.

### genel IPTCertificate getCertificate()

doğrulanması gereken SSL/TLS sertifikasını alır.

### public String getExitData ()

SSLExitData özelliğinde tanımlandığı şekilde çıkış verilerini alır.

### public String getExitName ()

SSLExitName özelliği tarafından tanımlandığı şekilde çıkış adını alır.

## ***MQIPT içinde com.ibm.mq.ipt.exit.CertificateExitResponse sınıfı***

Bu sınıf, bir sertifika doğrulandıktan sonra bilgileri MQIPT ' e geri aktarmak için kullanılır.

## **Oluşturucular**

### public CertificateExitYanıt (int rc, dizgi iletisi)

Bu oluşturucu, bir dönüş kodu ve bir ileti metni iletmek için kullanılabilir. Olası neden kodları şunlardır

- ExitRc.-Tamam.
- ExitRc.GEÇERLID\_HATASI

- ExitRc.VALIDATE\_REDEDILDI

### **public CertificateExitResponse (int rc)**

Bu oluşturucu, ileti metni olmadan bir dönüş kodu iletmek için kullanılabilir. Olası neden kodları şunlardır

- ExitRc.-Tamam.
- ExitRc.GEÇERLID\_HATASI
- ExitRc.VALIDATE\_REDEDILDI

### **public CertificateExitResponse ()**

Bu oluşturucu, ExitRcdönüş kodunu iletmek için kullanılabilir.Tamam, mesaj metni olmadan.

## **yöntemler**

### **public String getVersion()**

Bu yöntem, bu sınıfın sürümünü döndürür.

### **public String toString**

Bu yöntem, yanıtın dizgi gösterimini döndürür; örneğin, " Neden kodu: 4, İleti: Başarısız CRL denetimi.

## ***MQIPT içinde com.ibm.mq.ipc.exit.IPTCertificate sınıfı***

Bu sınıf, doğrulanacak SSL/TLS sertifikasını içerir.

## **yöntemler**

### **public int getVersion()**

Bu yöntem, bu sınıfın sürümünü döndürür.

### **public byte [] getDerEncoding ()**

Bu yöntem, X.509 sertifikasının ASN.1/DER kodlamasını ya da bir hata varsa NULL değerini döndürür.

### **genel bayt [] getPemKodlama ()**

Bu yöntem, X.509 sertifikasının PEM (BASE64) kodlamasını ya da bir hata varsa NULL değerini döndürür.

### **public String getLabel()**

Bu yöntem sertifika etiketini döndürür ya da bir hata varsa NULL değerini döndürür.

### **public String getName()**

Bu yöntem, sertifikanın Ayırt Edici Adını ya da yoksa NULL değerini döndürür. Örneğin:

```
CN=Test Queue Manager,OU=Sales,O=Example,L=London,C=GB
```

### **public String getIssuerName ()**

Bu yöntem, sertifikayı veren kişinin Ayırt Edici Adını ya da kullanılamıyorsa NULL değerini döndürür. Örneğin:

```
CN=Certificate Authority,OU=Security,O=Example,L=New York,C=US
```

### **genel IPTCertificate getSigner()**

Bu yöntem, imzalayıcı sertifikasını ya da kullanılamıyorsa NULL değerini döndürür. Kendinden imzalı bir sertifika için kendisine bir başvuru döndürür.

## **public String toString()**

Bu yöntem, sertifikanın dizgi gösterimini döndürür.

## **MQIPT içinde com.ibm.mq.ipc.exit.IPTTrace sınıfı**

MQIPT izleme işlevleri, bir yöntem girişi ve çıkışı için kullanılabilir. Giriş ve çıkış çağrılarını sağlar. Yararlı bilgileri izlemek için çeşitli veri çağrılarını da vardır.

## **yöntemler**

### **public void entry (String fid)**

Burada *fid* , çağrının yapıldığı yeri (örneğin, sınıf ve yöntem adı) tanımlamak için kullanılır.

Bu yöntem, denetim akışının bir yöntem girdiği noktayı kaydetmek için uygun girinti düzeyiyle izleme çıkış dosyasına bir giriş yazar. Bu çağrı isteğe bağlıdır, ancak kullanılırsa, aynı yöntem içinde eşleşen bir "exit (String)" çağrısı da kullanılmalıdır.

### **public void exit (String fid)**

Burada *fid* , çağrının yapıldığı yeri (örneğin, sınıf ve yöntem adı) tanımlamak için kullanılır.

Bu yöntem, denetim akışının bir yöntemi bıraktığı noktayı kaydetmek için uygun girinti düzeyiyle izleme çıkış dosyasına bir çıkış yazar. Bu yöntem yalnızca, daha önce aynı yöntemde bir "entry (String)" çağrısı kullanıldıysa kullanılır.

### **public void exit (String fid, int rc)**

Burada *fid* , çağrının yapıldığı yeri tanımlamak için kullanılır; örneğin, sınıf ve yöntem adı ve *rc* , yöntemin sayısal dönüş kodudur. Bu izleme yöntemi, tamsayı döndüren yöntemlerden çıkışı kaydetmek için kullanılmalıdır.

Bu yöntem, denetim akışının bir yöntemi ve bu yöntemin sayısal dönüş kodunu bir noktayı kaydetmek için, izleme çıkış dosyasına bir giriş yazar. Bu yöntem yalnızca, daha önce aynı yöntemde bir "entry (String)" çağrısı kullanıldıysa kullanılır.

### **public void exit (String fid, boolean rc)**

Burada *fid* , çağrının yapıldığı yeri (örneğin, sınıf ve yöntem adı) tanımlamak için kullanılır ve *rc* , yöntemin Boole dönüş kodudur. Bu izleme yöntemi, Boole değeri döndüren yöntemlerden çıkışı kaydetmek için kullanılmalıdır.

Bu yöntem, denetim akışının bir yöntemi ve o yöntemin Boole dönüş kodunu bir noktayı kaydetmek için, izleme çıkış dosyasına bir giriş yazar. Bu yöntem yalnızca, daha önce aynı yöntemde bir "entry (String)" çağrısı kullanıldıysa kullanılır.

### **public void data (String fid, String data)**

Burada *fid* , çağrının yapıldığı yeri (örneğin, sınıf ve yöntem adı) tanımlamak için kullanılır.

Bu yöntem, baz dizilim verilerini izleme çıkış dosyasına bir giriş yazar.

### **public void data (String fid, int data)**

Burada *fid* , çağrının yapıldığı yeri (örneğin, sınıf ve yöntem adı) tanımlamak için kullanılır.

Bu yöntem, baz tamsayı verilerini izleme çıkış dosyasına bir giriş yazar.

### **public void data (String fid, byte [])**

Burada *fid* , çağrının yapıldığı yeri (örneğin, sınıf ve yöntem adı) tanımlamak için kullanılır.

Bu yöntem, baz ikili verileri izleme çıkış dosyasına bir giriş yazar.

## **Örnek izleme**

Bir çıkıştaki sorunların tanımlanmasına yardımcı olmak için MQIPT ile aynı izleme olanağını kullanabilir, diğer bir seçenek olarak kendi izleme işlevlerinizi uygulayabilirsiniz. MQIPT izleme işlevlerini kullanmaya karar

verirseniz, bir yöntem giriş ve çıkış için kullanılabilir giriş ve çıkış çağruları vardır. Aşağıdaki örnekte gösterildiği gibi, yararlı bilgileri izlemek için çeşitli veri çağruları da vardır.

```

/**
 * This method is called to initialize the exit (for example, for
 * loading validation information) and place itself in a ready
 * state to validate connection requests.
 */
public int init(IPTTrace t) {
    final String fid = "MyExit.init";

    // Trace entry into this method
    t.entry(fid);

    // Trace useful information
    t.data(fid, "Starting exit - MQIPT version " + getVersion());

    // Perform initialization and load any data
    t.data(fid, "Ready for work");

    // Trace exit from this method
    t.exit(fid);

    return ExitRc.OK;
}

```

Bu yöntem, aşağıdaki örnekte gösterilen biçimde izleme üretir:

```

16:36:48.625    14    5000-1s    -----{ ConnectionThread.setCertificateExit()
16:36:48.625    14    5000-1s    Creating instance of certificate exit
16:36:48.625    14    5000-1s    Calling init() of certificate exit
16:36:48.625    14    5000-1s    -----} MyExit.init()
16:36:48.625    14    5000-1s    Starting exit - MQIPT version 2.1.0.0
16:36:48.625    14    5000-1s    Ready for work
16:36:48.625    14    5000-1s    -----} MyExit.init() rc=0
16:36:48.625    14    5000-1s    -----} ConnectionThread.setCertificateExit() rc=0

```

### **MQIPT içindeki sertifika çıkış dönüş kodları**

Bir dizi farklı durumda sertifika çıkışı ararken MQIPT ' in tanıdığı dönüş kodları

Aşağıdaki dönüş kodları, aşağıdaki durumlarda bir sertifika çıkışı çağrılırken MQIPT tarafından tanınır:

<b>Dönüş kodu</b>	<b>Açıklama</b>	<b>Init</b>	<b>Validat e</b>	<b>yenile</b>
ExitRc.-Tamam.	İstek başarıyla tamamlandı.	evet	evet	evet
ExitRc.GIRIŞ_HATASI	Kullanıma İnit isteği başarısız oldu, rota devre dışı bırakılacak.	evet		
ExitRc.YENILEME_HATASI	Yenileme isteği başarısız oldu, rota devre dışı bırakılacak.			evet
ExitRc.GEÇERLID_HATASI	Doğrulama işlemi başarısız oldu, bağlantı isteği reddedildi.		evet	
ExitRc.VALIDATE_REDDDEDILDI	Doğrulama isteği reddedildi, bağlantı isteği reddedildi.		evet	

### **MQIPT içinde LDAP ve CRL ' ler**

MQIPT , bir sayısal sertifikada Sertifika İptal Listesi (CRL) kimlik doğrulamasını gerçekleştirmek için LDAP (Lightweight Directory Access Protocol; Temel Dizin Erişimi Protokolü) sunucusunun kullanılmasını destekler.

LDAP desteği, aynı LDAP sunucusu hem IBM MQ hem de MQIPT için kullanılabilir için IBM MQ' deki benzer bir şekilde uygulanmıştır.

SSL/TLS el sıkışması sırasında, iletişim ortakları dijital sertifikalarla birbirlerini doğruluyor. Kimlik doğrulaması, alınan sertifikanın hala güvenilir olup olmadığını denetleyebilir. Sertifikasyon Yetkilileri (CA), aşağıdakiler de dahil olmak üzere çeşitli nedenlerle sertifikaları iptal ediyor:

- Sahip farklı bir kuruluşa taşındı.
- Özel anahtar artık gizli değil.

Sertifika kuruluşları, bir CRL 'de (Certificate Revocation List; Sertifika İptal Listesi) iptal edilen kişisel sertifikaları yayınlıyor. İptal edilen CA sertifikaları, bir Yetki İptal Listesinde (ARL) yayınlanır. CRL 'lere yapılan sonraki başvuruların da ARL' lere uygulandığını unutmayın.

IBM MQ ile LDAP sunucularının kullanımı ve CRL 'lerin ve ARL' lerin yönetimi hakkında daha fazla bilgi için bkz. [Sertifika İptal Listeleri ve Yetki İptal Listeleriyle Çalışma](#).

MQIPT , her bir rotada en çok iki LDAP sunucusunu destekleyebilir. İlk LDAP sunucusu, yedek olarak tutulan ikinci LDAP sunucusu ile ana sunucu olarak kabul edilir. İkinci sunucu yalnızca ana sunucuya ulaşılamadıysa kullanılır. Yedek sunucu, ana sunucunun ikiz görüntüsü olmalıdır.

LDAP sunucusunda saklanan bilgilere erişim, LDAP kullanıcı kimliği ve parola özellikleri kullanılarak bir kullanıcı kimliği ve parolayla korunabilir. LDAP sunucusu parolaları, IBM MQ 9.1.5'teki MQIPT yapılandırmasında şifrelenebilir. MQIPT tarafından kullanılacak parolaların şifrelenmesiyle ilgili daha fazla bilgi için bkz. [“MQIPT içinde saklanan parolaları şifreleme” sayfa 1048](#).

MQIPT Bir anahtarlık dosyasından PKCS #12 simgesi yüklediğinde, tüm CA sertifikalarının CRL geçerliliği denetlenir. CA sertifikasının ekli bir CRL 'si varsa, süresi dolup dolmadığını görmek için kontrol edilir ve geçerliyse, LDAP sunucusundan daha yeni bir CRL alınır. Alınan tüm CRL ' ler geçerli belirtece yüklenir ve CA sertifikasına eklenir.

Ana LDAP sunucusuna sorgu gönderildiğinde belirtilen CA ile eşleşen giriş yoksa, bu CA için CRL olmadığı varsayılır ve yedek sunucu kullanılmaz. Ancak, ana LDAP sunucusuna ulaşılamazsa ya da belirli bir zaman çerçevesi içinde dönmezse, yedekleme sunucusu kullanılır. Yedekleme sunucusundaki hatalar, istemci bağlantısının sona erdirilmesine neden olur. Bu işlem, **LDAPIgnoreErrors** özelliği true olarak ayarlanarak geçersiz kılınabilir.

MQIPT tarafından alınan CRL ' ler önbellekte tutulur ve bu rotadaki tüm bağlantılar tarafından paylaşılır. Önbelleğe alınan CRL ' nin süresi dolmuşsa, CRL önbellekten kaldırılır ve LDAP sunucusundan yeni bir CRL alınır. Yeni bir CRL kullanılmıyorsa, bağlantı reddedilmeye devam ediyor.

LDAP sunucusundan alınan bir CRL de süre bitimi için denetlenir ve bir uyarı iletisi görüntülenir (MQCPW001). Süresi dolan CRL sisteme yüklenmeye devam eder ve CRL ' ye atıfta bulunan bağlantı istekleri reddedilir. LDAP sunucusunda süresi dolan CRL ' yi geçerli bir CRL ile değiştirmeniz gerekir.

**LDAPCacheTimeout** özelliği, CRL önbelleğinin ne sıklıkta temizleneceğini denetlemek için kullanılabilir. Varsayılan değer 1 gündür. Bu değer 0 olarak ayarlanması, rota yeniden başlatılıncaya kadar önbellek girişlerinin temizlenmediği anlamına gelir.

Süresi dolan bir CRL, bir anahtarlık dosyasında ya da bir LDAP sunucusunda saklanabilir. Yeni bir CRL yayınlanmamışsa, daha fazla bağlantı isteği reddedilmiştir. **IgnoreExpiredCRLs** özelliğini etkinleştirerek süresi dolan CRL ' leri yoksayabilirsiniz.

**Not:** **LDAPIgnoreErrors** özelliğini ya da **IgnoreExpiredCRLs** özelliğini etkinleştirirseniz, SSL/TLS bağlantısı kurmak için iptal edilmiş bir sertifika kullanılabilir.

## **MQIPT içinde çok değerli sertifika Ayırt Edici Adı Kuruluş Birimi özellikleri**

Sertifika Ayırt Edici Adlarında birden çok kuruluş birimi (OU) değerini eşleyebilirsiniz.

Aşağıdaki rota özellikleri artık birden çok Kuruluş Birimi değerinin eşleştirilmesini desteklemektedir:

- **SSLClientDN\_OU**
- **SSLClientSiteDN\_OU**
- **SSLServerDN\_OU**
- **SSLServerSiteDN\_OU**

Birden çok kuruluş birimi değerini eşleştirmek için, rota özelliği değerinde ayırıcı olarak virgül kullanın. Örneğin:

```
SSLClientDN_OU=Sales, Europe
```

Bu, sertifikaları hem OU=Sales hem de OU=Europe ile eşleştirir. Kuruluş birimi değerleri, IBM MQ SSLPEER süzgeçlerinde birden çok Kuruluş Birimi değeriyle aynı sırayla eşleştirilir.

[route] kısmında aynı rota özelliğini bir kereden fazla belirtmeyin. Birden çok kuruluş birimi değerini eşleştirmenin doğru yolu, önceki örnekte gösterildiği gibi, özelliği bir kez belirtmektir. Aynı özneliği aynı mqipt.conf bölümünde bir kereden fazla girerseniz, son değer geçerli olur. Örneğin, ikinci satır birinci satırı geçersiz kıldığından, aşağıdaki girdiler yalnızca Avrupa ile eşleşmekle sonuçlanır:

```
SSLClientDN_OU=Sales  
SSLClientDN_OU=Europe
```

Bir kuruluş birimi değeri içinde hazır virgül ile eşleşmeniz gerekiyorsa, virgülden hemen önce bir çıkış karakteri olarak ters eğik çizgi (\) ekleyin. Örneğin:

```
SSLClientDN_OU=Sales\, Europe
```

Bu, tek bir değerle eşleşir: OU=Sales, Europe. Hemen ardından virgül gelmeyen ters eğik çizgi, hazır bilgi ters eğik çizgiyle eşleşir.

Önceki bir MQIPT yayın düzeyinden büyütme yapıyorsanız ve OU değerlerindeki virgülleri eşleştirme yeteneğine güveniyorsanız, önceki davranışı korumak için OU rota özelliklerine ters eğik çizgi çıkış karakterleri eklemelisiniz.

## **Deprecated** MQIPT içinde kullanımdan kaldırılan iletişim kurallarını ve şifreleme takımlarını etkinleştirme

Varsayılan olarak, MQIPT ile verilen Java runtime environment (JRE) içinde güvenli yuva iletişim kuralları ve güvenli olmayan olarak kabul edilen şifreleme takımları devre dışı bırakılır. Bu kullanımdan kaldırılan iletişim kuralları ve şifreleme takımları, kullanılabilirlikleri için etkinleştirilmelidir.

### **Bu görev hakkında**

Olası tehlikeleri biliyorsanız, ancak yine de MQIPT içinde güvensiz olarak kabul edilen iletişim kurallarından ya da şifre takımlarından birini kullanmanız gerekiyorsa, kullanmanız gereken iletişim kuralını ya da şifre takımını etkinleştirmek için bu yordamı izleyin.

**Not:** Kullanımdan kaldırılan iletişim kuralları ve şifreleme takımları TLS komut kapağıyla kullanılamaz.

### **Yordam**

1. `mqipt_path/java/jre/lib/security` dizininde bulunan `java.security` dosyasını düzenleyin; burada `mqipt_path`, MQIPT 'in kurulu olduğu konumdur.
2. `jdk.tls.disabledAlgorithms` özelliğindeki geçersiz kılınmış algoritmalar listesinden ilgili girişi kaldırarak, bir iletişim kuralı ya da algoritma için JRE 'ye destek ekleyin.
  - Bir protokole ilişkin destek eklemek için, protokolü geçersiz kılınmış algoritmalar listesinden kaldırın. Örneğin, TLS 1.0 desteğini eklemek için listeden TLSv1 ögesini kaldırın.
  - Bir şifreleme takımına ilişkin destek eklemek için, geçersiz kılınmış algoritmalar listesinden ilgili algoritmaları kaldırın. Örneğin, SSL\_ECDHE\_ECDSA\_WITH\_3DES\_EDE\_CBC\_SHA şifreleme takımı için destek eklemek istiyorsanız, 3DES\_EDE\_CBC ve DESede ögesini listeden kaldırın.
3. JRE 'de SSL 3.0 'ı etkinleştirmek için `com.ibm.jsse2.disableSSLv3=false` sistem özelliğini de ayarlamamız gerekir.



MQIPT komutunu **mqipt** komutunu kullanarak komut satırından başlatıyorsanız, **MQIPT\_JVM\_OPTIONS** ortam değişkenini kullanarak özelliği ayarlayabilirsiniz. Örneğin:

```
set MQIPT_JVM_OPTIONS=-Dcom.ibm.jsse2.disableSSLv3=false
```

**Windows** MQIPT bir Windows hizmeti olarak kuruluysa, özelliği, Windows kaydında HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\MQInternetPassThru anahtarı altında bir dizgi değeri tanımlayarak ayarlayabilirsiniz. Değer aşağıdaki özniteliklere sahip olmalıdır:

#### Ad

MqiptJvmSeçenekleri

#### Değer verileri

-Dcom.ibm.jsse2.disableSSLv3=false

4. Bir MQIPT rotasında SSL 3.0, TLS 1.0 ya da TLS 1.1 ' i etkinleştirmek için **SSLServerProtocols** ya da **SSLClientProtocols** rota özelliğine karşılık gelen iletişim kuralını ekleyin.
5. JRE özelliklerinde yapılan değişikliklerin yürürlüğe girmesi için MQIPT ' yi yeniden başlatın.

## MQIPT içinde PKCS #11 şifreleme donanımının kullanılması

MQIPT , PKCS #11 arabirimini destekleyen şifreleme donanımında saklanan sayısal sertifikalara erişebilir.

### Başlamadan önce

MQIPT ' i şifreleme donanımını kullanacak şekilde yapılandırmaya başlamadan önce, şifreleme kartının, kart sürücüsünün ve ilişkili destek yazılımlarının kurulduğundan ve düzgün çalıştığından emin olun.

IBM Java PKCS11 Şifreleme Sağlayıcısı (IBMPKCS11Impl sağlayıcısı) tarafından MQIPT içindeki PKCS #11 şifreleme donanımı için destek sağlanır. IBMPKCS11Impl sağlayıcısı ve Java 8 tarafından desteklenen şifreleme kartlarının listesi hakkında daha fazla bilgi için bkz. [IBM PKCS11 Şifreleme Sağlayıcısı](#).

### Bu görev hakkında

MQIPT tarafından erişilen kişisel sertifikaları ve CA sertifikalarını bir şifreleme donanım anahtarı deposunda saklayabilirsiniz. Ancak, PKCS #11 aygıtı olağan durumda çok miktarda imzalayıcı sertifikasını depolamak için yeterli alan olmadığı için, CA sertifikaları için ayrı bir dosya tabanlı anahtar deposu kullanmak isteyebilirsiniz.

MQIPT ' yi bir şifreleme donanım anahtar deposundaki sertifikaları kullanacak şekilde yapılandırmak için bu yordamı izleyin.

**Not:** MQIPT ile şifreleme donanımının kullanılması bir IBM MQ Advanced yeteneğidir. Bu yeteneği kullanmak için, MQIPT rotası kullanılarak bağlanan yerel kuyruk yöneticisinin de IBM MQ Advanced, IBM MQ Appliance, IBM MQ Advanced for z/OS VUE ya da IBM MQ Advanced for z/OS yetkisine sahip olması gerekir.

### Yordam

1. IBMPKCS11Impl sağlayıcısını kullanıma hazırlarken kullanılan yapılandırma kütüğünü yaratın.

IBMPKCS11Impl sağlayıcısı tarafından desteklenen donanım şifreleme kartlarının her biri için örnek yapılandırma dosyalarını karşıdan yükleyin ve sisteminiz için bir örnek yapılandırın. Örnekler, Java için IBM Documentation içindeki şu konudan yüklenebilir: [Yapılandırma dosyası](#).

Yapılandırma dosyası bir metin dosyasıdır ve en az aşağıdaki öznitelikleri içermelidir:

#### Ad

Sağlayıcı örneğinin ad soneki.

#### kitaplık

Şifreleme donanımıyla birlikte sağlanan PKCS #11 kitaplığının tam olarak nitelenmiş adı.

## simgeetiketi

PKCS #11 şifreleme aygıtı simgesi etiketi.

Örneğin, yapılandırma dosyası aşağıdaki girişleri içerebilir:

```
name = IPTPKCS11Provider
library = /usr/lib64/pkcs11/PKCS11_API.so
tokenlabel = icatoken
```

2. MQIPT kuruluş dizininin `java/jre/lib/security` alt dizininde bulunan Java güvenlik özellikleri dosyasını (`java.security`) düzenleyin.

a) Dosyada önceden yoksa, `IBMPKCS11Impl` güvenlik sağlayıcısını ekleyin.

Örneğin, aşağıdaki satırı ekleyerek:

```
security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
```

b) Yapılandırma dosyasının tam olarak nitelenmiş adını sağlayıcı adından sonra ekleyin.

Örneğin, "1" sayfa 1037 . adımda oluşturduğunuz yapılandırma dosyasının adı `/opt/mqipt/pkcs11.cfg` ise, bu yolu güvenlik sağlayıcısıyla aynı satıra eklemelisiniz:

```
security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl /opt/mqipt/
pkcs11.cfg
```

3. CA sertifikaları için bir anahtarlık dosyası kullanıyorsanız, CA sertifikalarını şifreleme donanımında saklamak yerine bir PKCS #12 biçimi CA anahtarlık dosyası oluşturun.

**mqiptKeyman** grafik kullanıcı arabirimini (GUI) ya da **mqiptKeycmd** komut satırı arabirimini (CLI) kullanarak bir CA anahtarlık dosyası oluşturabilirsiniz.

• CLI ' yi kullanmak için aşağıdaki komutu girin:

```
mqiptKeycmd -keydb -create -db filename -pw password -type pkcs12
```

Burada *kütükadı* , yaratılacak anahtarlık kütüğünün adı ve *parola* , anahtarlık parolasıdır.

• GUI ' yi kullanmak için aşağıdaki adımları izleyin:

a. **mqiptKeyman** komutunu girerek GUI ' yi başlatın.

b. **Anahtar Veritabanı Dosyası** > **Aç** düğmesini tıklayın.

c. **Anahtar veritabanı tipi** öğesini tıklayın ve **PKCS11Config** öğesini seçin.

d. **Tamam**' ı tıklayın. Open Cryptographic Token (Şifreleme Simgesini Aç) penceresi açılır.

e. Sertifikaları saklamak için kullanmak istediğiniz şifreleme aygıtı simgesi etiketini seçin.

f. **Cryptographic Token Password** (Şifreleme Simgesi Parolası) alanında, şifreleme donanımına erişmek için gereken parolayı yazın.

g. Yeni bir CA anahtarlık kütüğü yaratmak için **Yeni ikincil anahtar veritabanı kütüğü yarat** öğesini seçin.

h. **Anahtar veritabanı tipi** öğesini tıklayın ve **PKCS12** öğesini seçin.

i. **Dosya Adı** alanında CA anahtarlığı dosya adını yazın.

j. **Location** (Yer) alanına CA anahtarlık dosyasının tam yolunu yazın.

k. **Tamam**' ı tıklayın. Password Prompt (Parola İstemi) penceresi açılır.

l. **Password** (Parola) alanına CA anahtarlığı için bir parola yazın ve **Confirm Password** (Parolayı Onayla) alanına parolayı yeniden yazın.

m. **Tamam**' ı tıklayın.

4. **mqiptKeycmd** ya da **mqiptKeyman** kullanarak, şifreleme donanımı için kişisel bir sertifika isteyin.

• CLI ' yi kullanmak için aşağıdaki komutu girin:

```
mqiptKeycmd -certreq -create -crypto module_name -tokenlabel hardware_token
-pw password -label label -size key_size
-sig_alg algorithm -dn distinguished_name -file filename
```

Burada:

**-crypto modülü\_adi**

Şifreleme donanımıyla birlikte sağlanan PKCS #11 kitaplığının tam olarak nitelenmiş adını belirler.

**-tokenlabel token\_label**

PKCS #11 şifreleme aygıtı simgesi etiketini belirtir.

**-pw parola**

Şifreleme donanımına erişmek için kullanılacak parolayı belirler.

**-label etiket**

Sertifika etiketini belirtir.

**-size anahtar\_büyüklüğü**

Anahtar boyutunu belirtir. Değer 512, 1024, 2048ya da 4096 olabilir.

**-sig\_alg algoritma**

Girdinin anahtar çiftinin oluşturulması için kullanılan asimetrik imza algoritmasını belirtir.

Değer MD2\_WITH\_RSA, MD2WithRSA, MD5\_WITH\_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, SHA2/ECDSA, SHA224WithECDSA, SHA256\_WITH\_RSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithECDSA, SHA3/ECDSA, SHA384\_WITH\_RSA, SHA384WithECDSA, SHA384WithRSA, SHA3WithECDSA, SHA5/ECDSA, SHA512\_WITH\_RSA, SHA512WithECDSA, SHA512WithRSA, SHA5WithECDSA, SHA\_WITH\_DSA, SHA\_WITH\_RSAYa da SHAWithDSA. Varsayılan değer SHA256WithRSAdeğeridir.

**-dn ayırt edici\_ad\_adi**

Çift tırnak içine alınmış X.500 ayırt edici adını belirtir.

**-file kütükadı**

Sertifika isteğine ilişkin dosya adını belirtir.

- GUI ' yi kullanmak için aşağıdaki adımları izleyin:

- a. **Oluştur** menüsünden **Yeni Sertifika İsteği** seçeneğini tıklatın.
- b. **Anahtar Etiket**i alanına sertifika etiketini girin.
- c. Gerek duyduğunuz **Anahtar Boyutu** ve **İmza Algoritması** ' nı seçin.
- d. **Ortak Ad** ve **Kuruluş** için değerleri girin ve bir **Ülke** seçin. Geri kalan isteğe bağlı alanlar için varsayılan değerleri kabul edin ya da yeni değerler yazın ya da seçin.
- e. **Sertifika isteğinin saklanacağı dosyanın adını girin** alanına varsayılan değeri kabul edin `certreq.armya` da tam yol ile yeni bir değer yazın.
- f. **Tamam**' ı tıklatın.
- g. **Kişisel Sertifika İstekleri** listesi, yarattığınız yeni kişisel sertifika isteğinin etiketini gösterir. Sertifika isteği, seçtiğiniz dosyada saklanır.

5. CA size kişisel sertifikayı gönderdikten sonra, CA sertifikasını şifreleme anahtarı deposuna ya da CA anahtar halkası dosyasına (yoksa) ekleyin.

- CA sertifikasını CA anahtarlık dosyasına eklemek üzere CLI ' yi kullanmak için aşağıdaki komutu girin:

```
mqiptKeycmd -cert -add -db filename -pw password -type pkcs12  
-label label -file cert_filename
```

Burada *kütükadı* , CA anahtarlık kütüğünün adıdır, *parola* CA anahtarlık parolasıdır, *etiket* sertifikaya iliştilen etikettir ve *cert\_kütükadı* , CA sertifikasını içeren kütüğün adıdır.

- CA sertifikasını şifreleme donanımına eklemek üzere CLI ' yi kullanmak için aşağıdaki komutu girin:

```
mqiptKeycmd -cert -add -crypto module_name -tokenlabel hardware_token  
-pw password -label label -file cert_filename
```

Burada *module\_name* , şifreleme donanımıyla birlikte sağlanan PKCS #11 kitaplığının tam olarak nitelenmiş adıdır, *hardware\_token* PKCS #11 şifreleme aygıtı simgesi etiketidir, *password*

şifreleme donanımına erişim parolasıdır, *label* sertifikaya ekli etikettir ve *cert\_filename* , CA sertifikasını içeren dosyanın adıdır.

- GUI ' yi kullanmak için aşağıdaki adımları izleyin:
  - a. **Anahtar veritabanı içeriği** alanında **İmzalayan Sertifikalar** seçeneğini belirleyin.
  - b. **Ekle**'yi tıklatın. Dosyadan CA Sertifikası Ekle penceresi açılır.
  - c. Sertifikanın saklandığı sertifika dosyası adını ve yerini yazın ya da adı ve yeri seçmek için **Göz At** düğmesini tıklatın.
  - d. **Tamam**'ı tıklatın. Enter a Label (Etiket Girin) penceresi açılır.
  - e. Etiket Girin penceresinde sertifikanın adını yazın.
  - f. **Tamam**'ı tıklatın. Sertifika anahtar veritabanına eklenir.

6. CA tarafından sağlanan kişisel sertifikayı şifreleme donanım anahtarı deposuna alın.

- CLI ' yı kullanmak için aşağıdaki komutu girin:

```
mqiPTKeycmd -cert -receive -file filename -crypto module_name  
-tokenlabel hardware_token -pw password
```

Burada *kütükadı* , alınacak sertifikayı içeren kütüğün adıdır; *modül\_adi* , şifreleme donanımıyla birlikte verilen PKCS #11 kitaplığının tam olarak nitelenmiş adıdır; *donanım\_simgesi* PKCS #11 şifreleme aygıtı simgesi etikettir ve *parola* , şifreleme donanımına erişim parolasıdır.

CA sertifikası şifreleme donanımı yerine bir CA anahtar halkasında saklanıyorsa, **mqiPTKeycmd** komutu kişisel sertifikayı şifreleme anahtarı deposuna alırken CA anahtar halkasına erişemediğinden sertifika zincirinin doğrulanamadığına ilişkin bir uyarı alırsınız.

- GUI ' yi kullanmak için aşağıdaki adımları izleyin:
  - a. **Aldüğmesini** tıklatın. Dosyadan Sertifika Al penceresi açılır.
  - b. Yeni kişisel sertifika için sertifika dosyası adını ve yerini yazın ya da adı ve yeri seçmek için **Göz At** düğmesini tıklatın.
  - c. **Tamam**'ı tıklatın. **Kişisel Sertifikalar** alanı, eklediğiniz yeni kişisel sertifikanın etiketini gösterir.

7. **mqiPTPW** komutunu kullanarak şifreleme donanımına erişmek için parolayı şifreleyin.

Aşağıdaki komutu girin:

```
mqiPTPW -sf encryption_key_file
```

Burada *encryption\_key\_file* , MQIPT kuruluşunuz için parola şifreleme anahtarını içeren bir dosyanın adıdır. MQIPT kuruluşunuz varsayılan parola şifreleme anahtarını kullanıyorsa, **-sf** parametresini belirtmenize gerek yoktur. İstendiğinde şifrelenecek şifreleme donanımına erişmek için parolayı yazın.

Anahtar deposu parolalarını şifreleme hakkında daha fazla bilgi için bkz. [“MQIPT içinde bir anahtarlık parolasını şifreleme” sayfa 1022.](#)

8. [“3” sayfa 1038.](#) adımda bir CA anahtarlık dosyası oluşturduysanız, [“7” sayfa 1040.](#) adımdaki yönergeleri izleyerek CA anahtarlık dosyasının parolasını şifreleyin.

9. **mqiPT.conf** yapılandırma dosyasını düzenleyin.

- a) **EnableAdvancedCapabilities** genel özelliğini **true** olarak ayarlayarak bu IBM MQ Advanced özelliğini kullanmak için uygun yetkiye sahip olduğunuzu onaylayın.
- b) Bir ya da daha çok **SSLServerKeyRingUseCryptoHardware**, **SSLServerCAKeyRingUseCryptoHardware**, **SSLServerKeyRingUseCryptoHardware** ya da **SSLServerKeyRingUseCryptoHardware** özelliğini **true** değerine ayarlayarak rotada şifreleme donanımı anahtar deposunun kullanımını etkinleştirin.

Bir rotada şifreleme donanımının kullanılmasını sağlayan özellikler hakkında daha fazla bilgi için bkz. [MQIPT rota özellikleri.](#)

IBM MQ 9.2.0' den, **SSLCommandPortKeyRingUseCryptoHardware** özelliğini **true** olarak ayarlayarak TLS komut kapısı ile şifreleme donanımını da kullanabilirsiniz.

c) CA sertifikaları için bir anahtarlık dosyası kullanıyorsanız, **SSLServerCAKeyRing** ya da **SSLServerCAKeyRing** özelliklerinden birini ya da birkaçını ayarlayarak CA anahtarlığı konumunu belirtin.

Site sertifikası için şifreleme donanımı kullanacak bir rota yapılandırdıysanız ve bir CA anahtar halkası dosyası belirtmezseniz, şifreleme donanım anahtarı deposu CA anahtar deposu olarak kullanılır.

d) **SSLServerKeyRingPW**, **SSLServerCAKeyRingPW**, **SSLClientKeyRingPW**, **SSLClientCAKeyRingPW** ya da **SSLCommandPortKeyRingPW** özelliğini kullanarak şifreleme donanımına ve CA anahtar halkasına erişmek için şifrelenmiş parolayı belirtin.

**SSL\*KeyRingPW** özelliklerinin değerini, **mqiPTPW** komutuyla şifrelenmiş parola çıkışına ayarlayın.

e) Şifreleme donanımı birden fazla kişisel sertifika içeriyorsa, kimlik doğrulaması için SSL/TLS sunucusuna ya da istemcisine göndermek üzere MQIPT tarafından hangi sertifikanın seçilmesi gerektiğini belirtin.

Bir SSL/TLS istemci rotası için **SSLClientSite\*** özelliklerinden birini ya da SSL/TLS sunucu rotası için **SSLServerSite\*** özelliklerinden birini ayarlayarak hangi sertifikanın seçileceğini belirtebilirsiniz.

Sertifika etiketi adını belirtmek için **SSLCommandPortSiteLabel** özelliğini kullanarak TLS komut kapısı tarafından hangi sertifikanın kullanılacağını belirtebilirsiniz.

Bir anahtarlıktan sertifika seçilmesiyle ilgili daha fazla bilgi için bkz. "[MQIPT içinde bir anahtarlık dosyasından sertifikaları seçme](#)" sayfa 1023. Bir anahtar halkasından sertifika seçme özellikleri MQIPT rota özelliklerinde açıklanmıştır.


Örneğin, TLS sunucu rotasındaki site sertifikası için bir şifreleme donanım anahtarı deposu ve aynı rotaya ilişkin CA sertifikalarını saklamak için bir anahtarlık dosyası kullanmak üzere rota tanımına aşağıdaki özellikleri ekleyin:

```
SSLServerKeyRingUseCryptoHardware=true
SSLServerKeyRingPW=<mqiPTPW>1!g0RdM4wft5d1rCgNMDEGag==!dZxhgQD2A8Ea0yeqawQvPg==
SSLServerCAKeyRing=/opt/mqiPT/ssl/ca.pfx
SSLServerCAKeyRingPW=<mqiPTPW>1!3Vdripiu6kMwn0sWRCVgT5g==!LH1tGLEg30FvN8+02Re0YA==
SSLServerSiteLabel=mqiPTsite
```

10. MQIPT olanağını yeniden başlatın.

## Java security manager içinde MQIPT

Java security manager , başka bir güvenlik düzeyi sağlamak için herhangi bir MQIPT özelliğiyle birlikte kullanılabilir.

**Not:**  Java security manager with MQIPT kullanımı, Java security manager ' in gelecekteki bir Javayayınında kaldırılmak üzere kullanımdan kaldırılmış olması nedeniyle kullanımdan kaldırılmıştır.

MQIPT , `java.lang.SecurityManager` sınıfında tanımlandığı şekilde varsayılan Java security manager değerini kullanır. MQIPT içindeki Java security manager özelliği, **SecurityManager** genel özelliği kullanılarak etkinleştirilebilir ya da devre dışı bırakılabilir. Daha fazla bilgi için bkz. [MQIPT genel özellikler](#) .

Java security manager iki varsayılan ilke dosyası kullanır:

- `$MQIPT_PATH/java/jre/lib/security/java.policy` adlı genel sistem ilkesi kütüğü (burada `$MQIPT_PATH` , MQIPT olanağının kurulu olduğu dizindir), bir anasistemdeki sanal makinenin tüm yönetim ortamları tarafından kullanılır.
- Kullanıcının ana dizininde var olabilen `.java.policy` adlı kullanıcıya özgü bir ilke dosyası.

Ek bir MQIPT ilke dosyası da kullanılabilir. Daha önce açıklanan varsayılan ilke dosyaları yerine MQIPT ilke dosyasını kullanmalısınız. Daha fazla bilgi için [MQIPT genel özellikler](#) içindeki **SecurityManagerPolicy** bölümüne bakın.

İlke dosyasının sözdizimi oldukça karmaşıktır ve bir metin düzenleyicisi kullanılarak değiştirilebilse de, değişiklik yapmak için Java ile birlikte sağlanan İlke Aracı yardımcı programını kullanmak genellikle daha

kolaydır. İlke Aracı yardımcı programını `$MQIPT_PATH/java/jre/bin` dizininde bulabilirsiniz ve Java belgelerinde tam olarak belgelenmiştir.

MQIPT ile birlikte, MQIPT' u çalıştırmak için hangi izinlerin ayarlanması gerektiğini gösteren bir örnek ilke dosyası (`mqiptSample.policy`) sağlanmıştır.

Örnek ilke dosyasını yapılandırmanızla eşleşecek şekilde düzenlemeniz gerekir. Özellikle, `mqipt.conf` yapılandırma dosyasını içeren MQIPT ana dizininin MQIPT kuruluş diziniyle aynı olmayabileceğini unutmayın, bu nedenle güvenlik ilkesinde **FilePermission** girişlerini yapılandırırken doğru dizinleri belirtmeye dikkat edin.

Aşağıdaki girdileri değiştirmeniz gerekir:

- **errors** dizinine okuma ve yazma erişimi veren **java.io.FilePermission** girdisi. Bu girişteki dosya yolu, **errors** dizininin bulunduğu yer olduğundan MQIPT ana dizinine başvurmalıdır. MQIPT , **errors** dizininde FFST Failure Data Capture dosyalarını (`AMQ*.FDC`) ve izleme dosyalarını (`AMQ*.TRC*`) oluşturur. Sorun gidermenin mümkün olması için MQIPT ' in **errors** dizininde izleme ve FFST dosyaları yaratma iznine sahip olduğundan emin olmanız gerekir.
- **logs** dizinine okuma ve yazma erişimi veren **java.io.FilePermission** girdisi. Bu girişteki dosya yolu, **logs** dizininin bulunduğu yer olduğundan MQIPT ana dizinine başvurmalıdır. MQIPT , **ConnectionLog** genel özelliği etkinleştirildiyse, **logs** dizininde bağlantı günlüğü dosyaları (`mqipt*.log`) oluşturur.
- MQIPT kuruluş dizinindeki **bin**, **exits**, **libve ssl** dizinleri gibi dizinlere okuma ve yürütme erişimi veren **java.io.FilePermission** girdileri. Bu girişlerdeki dosya yolları, MQIPT kuruluş dizinine başvurmak için değiştirilmelidir. Gerekli değilse, bu girdilerin bazıları atlanabilir.
- **java.net.SocketPermission** girişleri, her bir dinleme MQIPT rotasındaki bağlantıları denetleyecek şekilde değiştirilmelidir. Her MQIPT rotasına ilişkin dinleyici kapısı ve dinleyici adresi için **dinleme** ve **kabul** izinleri gerekir.
- **java.net.SocketPermission** girişleri, her MQIPT rotasındaki bağlantıları denetlemek için değiştirilmelidir. Bağlantı izni, MQIPT rotasının bağlandığı rota hedefleri, yetkili sunucular ya da LDAP sunucuları için gereklidir. Hedefleri bir IP adresi yerine bir anasistem adı kullanarak belirtirken **çözümleme** izni gereklidir.

Yapılandırmanıza bağlı olarak aşağıdaki girdileri de eklemeniz gerekebilir:

- `mqipt.conf` yapılandırma dosyası ya da `mqipt.conf` içeren MQIPT ana dizini için okuma erişimi vermek üzere bir **java.io.FilePermission** girdisi.
- Güvenlik ilkesi dosyasının kendisine okuma erişimi vermek için bir **java.io.FilePermission** girdisi. MQIPT yenilemesi, güvenlik ilkesi dosyasının yeniden okunmasına neden olursa bu kullanışlıdır.
- SSL/TLS anahtarlık dosyalarına ve anahtarlık parola dosyalarına okuma erişimi vermek için bazı **java.io.FilePermission** girdileri. Bu yalnızca **SSLClient** ya da **SSLServer** özellikleri etkinleştirilmiş bir rota kullanıldığında ya da TLS komut kapısı yapılandırıldığında gereklidir.
- Herhangi bir MQIPT çıkış sınıfına okuma ya da yürütme erişimi vermek için bazı **java.io.FilePermission** girişleri. Bu yalnızca bir MQIPT çıkışı etkinleştirildiğinde gereklidir. Çıkışın gerektirdiği durumlarda ek izinler vermeniz gerekebilir.

**Not:** Windows **java.io.FilePermission** girişleri, yoldaki her ters eğik çizgi için iki ters eğik çizgi karakteri (`\\`) kullanılmalıdır. Bunun nedeni, çıkış karakteri olarak tek bir ters eğik çizgi kullanılmamasıdır.

Örnek dosyada, MQIPT ' un `C:\Program Files\IBM\MQ\Internet Pass-Thru` içindeki bir Windows sistemine kurulduğu varsayılmıştır. Ayrıca, MQIPT ana dizininin (`mqipt.conf` kütüğünün yeri) MQIPT kuruluş diziniyle aynı olduğu da varsayılır.

MQIPT ürününü başka bir yere kurduysanız, **codeBase** tanımlamasındaki dizini MQIPT kuruluş dizininize gönderme olacak şekilde değiştirmeniz gerekir. Doğru öneki (`file:/`) ve doğru dosya sonekini (`/lib/com.ibm.mq.ipt.jar`) eklemeyi göz önüne alın. AIX and Linux sistemlerinde, tipik bir **codeBase** URL `file:/opt/mqipt/lib/com.ibm.mq.ipt.jar` olabilir (MQIPT ' in `/opt/mqipt` içinde kurulu olduğu varsayılarak).

İzinler genellikle üç öznitelikle tanımlanır. Yuva bağlantılarını denetlemek için değerleri şunlardır:

## sınıf izni

java.net.SocketPermission

## denetlemek için ad

Bu, adın her bir bileşeninin bir genel arama karakteriyle belirtilebileceği hostname : port biçimiyle oluşur. Anasistem adı bir etki alanı adı ya da IP adresi olabilir. Anasistem adının en soldaki konumu yıldız işaretiyle (\*) belirlenebilir. Örneğin, harry . company1 . com şu dizgilerin her biri tarafından eşleştirilir:

- harry
- harry . company1 . com
- \* . company1 . com
- \*
- 198 . 51 . 100 . 123 (bunun harry . company1 . com IP adresi olduğu varsayılarak)

Adın kapı bileşeni, tek bir kapı adresi ya da bir kapı adresi aralığı olarak belirtilebilir; örneğin:

### 1414

yalnızca kapı 1414

### 1414 ...

1414 'ten büyük ya da 1414 'e eşit tüm kapı adresleri

### -1414.

1414 'ten küçük ya da 1414 'e eşit tüm kapı adresleri

### 1-1414

1 ile 1414 arasındaki tüm kapı adresleri (bu değerler de içinde olmak üzere)

## izin verilen işlem

java.net.SocketPermission tarafından kullanılan işlemler şunlardır:

### kabul et

Belirtilen hedeften gelen bağlantıların kabul edilmesine izin ver

### bağlan

Belirtilen hedefle bağlantıya izin ver

### Dinle.

Uygulamanın bağlantı istekleri için belirtilen kapıda ya da kapılarda dinleme yapmasına izin ver

### Çözümle

Etki alanı adlarını IP adreslerine çözümlemek için DNS kullanılmasına izin ver

Java security manager denetimi, java . security . manager ve java . security . policy Java sistem özellikleriyle de yapılabilir, ancak MQIPT denetlemek için **SecurityManager** ve **SecurityManagerPolicy** özelliklerini kullanmanız önerilir.

Tanımlama bilgilerini izleme ve FFST kayıtlarına dahil etmek için MQIPT , belirli MQIPT sistem özelliklerine ve ortam değişkenlerine erişmelidir. Java güvenlik ilkesine her zaman aşağıdaki özellikleri eklemelisiniz:

```
permission java.util.PropertyPermission "java.home", "read";
permission java.util.PropertyPermission "java.version", "read";
permission java.util.PropertyPermission "java.runtime.version", "read";
permission java.util.PropertyPermission "java.vm.info", "read";
permission java.util.PropertyPermission "java.vm.vendor", "read";
permission java.util.PropertyPermission "os.arch", "read";
permission java.util.PropertyPermission "os.name", "read";
permission java.util.PropertyPermission "os.version", "read";
permission java.lang.RuntimePermission "getenv.MQIPT_PATH";
permission java.lang.RuntimePermission "getStackTrace";
permission javax.management.MBeanServerPermission "createMBeanServer";
permission javax.management.MBeanPermission "com.ibm.mq.ipt.IPTManager#[com.ibm.mq.ipt:type=IPTManager]", "registerMBean";
permission javax.management.MBeanPermission "com.ibm.mq.ipt.IPTManager#[com.ibm.mq.ipt:type=IPTManager]", "unregisterMBean";
permission javax.management.MBeanTrustPermission "register";
```

Bu özelliklerin tümünü dahil etmezseniz, MQIPT düzgün çalışmaz ve sorun tanımlama bozulur.

## MQIPT içinde güvenlik çıkışları

**Destination** rota özelliği tarafından tanımlandığı şekilde, hedef hedefe erişimi denetlemek için bir güvenlik çıkışı kullanın. MQIPT bir istemciden bir bağlantı isteği aldığı anda, ancak hedef hedefle bağlantı kurmadan önce güvenlik çıkışı çağrılır.

İlk bağlantı özelliklerine dayalı olarak, bağlantının tamamlanmasına izin verilip verilmeyeceğine güvenlik çıkışı karar verir.

Bir rota başlatıldığında, bir bağlantı isteğini başlatmak ve işlemeye hazır hale getirmek için güvenlik çıkışı çağrılır. Kullanıma hazırlama işlemi, herhangi bir kullanıcı verisini yüklemek ve bu verileri hızlı ve kolay erişim için hazırlamak için kullanılmalıdır; böylece, bağlantı isteğini işleme süresi en aza indirilir.

Her rotanın kendi güvenlik çıkışı olabilir.

- **SecurityExit** özelliği, kullanıcı tanımlı güvenlik çıkışını etkinleştirmek/devre dışı bırakmak için kullanılır.
- **SecurityExitName** özelliği, kullanıcı tanımlı güvenlik çıkışının sınıf adını tanımlamak için kullanılır.
- **SecurityExitPath** özelliği, sınıf dosyasını içeren dizin adını tanımlamak için kullanılır. Bu özellik ayarlanmazsa, sınıf dosyasının çıkışlar alt dizininde bulunacağı varsayılır. **SecurityExitPath**, kullanıcı tanımlı güvenlik çıkışını içeren JAR dosyasının adını da tanımlayabilir.
- **SecurityExitTimeout** özelliği, bir bağlantı isteğini doğrularken güvenlik çıkışından yanıt beklenmesini ne kadar süreyle bekleyeceğini belirlemek için MQIPT tarafından kullanılır.

Güvenlik çıkışı özelliklerinin ayrıntıları için bkz. [MQIPT rota özellikleri](#).

MQIPT, kullanıcı tanımlı bir güvenlik çıkışını çağırarak için `SecurityExit` sınıfını kullanır. Bu sınıf, gerekli işlevselliği sağlamak için kullanıcı tanımlı güvenlik çıkışı tarafından genişletilmeli ve yöntemlerinin çoğu geçersiz kılınmalıdır. `SecurityExitResponse` nesnesi MQIPT 'e veri iletmek için kullanılır ve bu veriler MQIPT tarafından bağlantı isteğinin kabul edilip edilmeyeceğine karar vermek için kullanılır. `SecurityExitResponse` nesnesi, güvenlik çıkışı özellikleri tarafından tanımlanan rotayı geçersiz kılmak için kullanılan yeni bir hedef ve hedef kapı adresi de içerebilir.

Bir güvenlik çıkışının nasıl gerçekleştirileceğini göstermek için üç örnek güvenlik çıkışı sağlanır.

- `SampleSecurityExit`, IBM MQ kanalının adına dayalı olarak bir IBM MQ kuyruk yöneticisine erişimin nasıl denetleneceğini gösterir. Yalnızca "MQIPT" dizgisiyle başlayan bir kanal adına sahip bağlantıya izin verir. Ek bilgi için [Güvenlik çıkışını kullanma](#) başlıklı konuya bakın.
- `SampleRoutingExit`, istemci bağlantısı isteklerinin, her biri aynı ada ve aynı özniteliklere sahip bir kuyruk yöneticisini barındıran, tanımlı IBM MQ sunucularından oluşan bir havuza dinamik olarak yönlendirilmesine olanak sağlar. Örnek, sunucu adlarının listesini içeren bir yapılandırma kütüğünü içerir. Ek bilgi için [Güvenlik çıkışlarını kullanarak istemci bağlantısı isteklerini IBM MQ kuyruk yöneticisi sunucularına yönlendirme](#) başlıklı konuya bakın.
- `SampleOneRouteExit`, bağlantı isteğinde kullanılan IBM MQ kanal adından türetilen bir IBM MQ kuyruk yöneticisine dinamik yöneltme sağlar. Örnek, kuyruk yöneticisi adlarının sunucu adlarıyla eşleşmesini içeren bir yapılandırma dosyası içerir. Daha fazla bilgi için bkz. [Dinamik olarak istemci bağlantısı isteklerini yönlendirme](#).

**Not:** MQIPT tek bir JVM 'de çalışır; böylece kullanıcı tanımlı bir güvenlik çıkışı, MQIPT 'in olağan çalışmasını aşağıdaki yollardan biriyle tehlikeye atabilir:

- Sistem kaynaklarını etkiler
- Darboğazlar oluşturur
- Performansı düşürebilirsiniz

Güvenlik çıkışınızı bir üretim ortamında uygulamadan önce, güvenlik çıkışınızın etkilerini kapsamlı olarak test etmelisiniz.



## **MQIPT içinde com.ibm.mq.ipc.exit.SecurityExit sınıfı**

Bu sınıf ve genel yöntemleri, bazı ortak verilere erişim sağlamak ve bazı MQIPT kullanıma hazırlama işlemlerine izin vermek için kullanıcı tanımlı güvenlik çıkışı tarafından genişletilmelidir.

Her yöntem MQIPT tarafından çağrılmadan önce, yöntemin kullanılabilmesi için bazı özellikler kullanılmalıdır. Değerleri, bu sınıfta tanımlanan uygun alma yöntemleri kullanılarak alınabilir.

### **yöntemler**

#### **public int init (IPTTrace)**

Aşağıdaki özellikler kullanılabilir:

- dinleyici kapısı
- Hedef
- Hedef Kapısı
- sürüm

init yöntemi, bir rota başlatıldığında MQIPT tarafından çağrılır. Bu yöntemden döndüğünüzde, güvenlik çıkışı bir bağlantı isteğinin geçerliliğini denetlemeye hazır olmalıdır. Geçerli dönüş kodları şunlardır: ExitRc.OK ya da ExitRc.INIT\_ERROR.

#### **public int refresh (IPTTrace)**

Aşağıdaki özellikler kullanılabilir:

- dinleyici kapısı
- Hedef
- Hedef Kapısı

refresh yöntemi, MQIPT yapılandırması yenilendiğinde MQIPT tarafından çağrılır. Bu işlem genellikle yapılandırma dosyasında bir özellik değiştirildiğinde gerçekleştirilir. MQIPT, hangi özelliklerin değiştirildiğini ve rotanın yeniden başlatılması gerekip gerekmediğini belirlemek için konfigürasyon dosyasındaki tüm özellikleri yeniden yükler.

Bu yöntem, kullandığı herhangi bir dış verinin yeniden yüklenmesini (yani, **init** yöntemiyle yüklenen verilerin) gerçekleştirmelidir. Geçerli dönüş kodları şunlardır: ExitRc.OK ya da ExitRc.REFRESH\_ERROR.

#### **genel geçersiz kapatma (IPTTrace)**

Aşağıdaki özellikler kullanılabilir:

- dinleyici kapısı
- Hedef
- Hedef Kapısı

close yöntemi dururken MQIPT tarafından çağrılır. Bu yöntem, çalışma sırasında çıkışın edindiği tüm sistem kaynaklarını serbest bırakmalıdır. MQIPT, kapanmadan önce bu yöntemin tamamlanmasını bekler.

Bu yöntem, daha önce bir güvenlik çıkışı etkinleştirildiyse, ancak şimdi yapılandırma dosyasında devre dışı bırakıldıysa da çağrılır.

#### **public SecurityExitYanıt doğrulaması (IPTTrace)**

Aşağıdaki özellikler kullanılabilir:

- dinleyici kapısı
- Hedef
- Hedef Kapısı
- zaman aşımı
- İstemci IP adresi

- istemci kapısı adresi
- Kanal Adı
- Kuyruk yöneticisi adı

validate yöntemi, geçerliliğini denetlemek için bir bağlantı isteği aldığında MQIPT tarafından çağrılır. **SSLProxyMode** özelliği etkinleştirildiyse, kanal adı ve kuyruk yöneticisi adı kullanılamaz; bu özellik yalnızca TLS verilerini tünel olarak aktarmak için kullanılır; bu nedenle, ilk veri akışından genellikle elde edilen veriler okunamaz.

Güvenlik çıkışı, aşağıdaki bilgileri içeren bir SecurityExitResponse nesnesi döndürmelidir:

- neden kodu (ayarlanmalıdır)
- yeni hedef adresi (isteğe bağlı)
- yeni hedef dinleyici kapısı adresi (isteğe bağlı)
- ileti (isteğe bağlı)

Neden kodu, bağlantının MQIPT tarafından kabul edilip edilmediğini ya da reddedilip kabul edilmediğini belirler. newDestination ve newDestinationPort alanları isteğe bağlı olarak yeni bir hedef kuyruk yöneticisi tanımlayacak şekilde ayarlanabilir. Bu özellikleri ayarlamazsanız, konfigürasyon dosyasında tanımlanan **Destination** ve **DestinationPort** rotası özellikleri kullanılır. Döndürülen iletiler, bağlantı günlüğü kütüğü girişinin sonuna eklenir.

MQIPT yapılandırma özelliklerinin değerlerini almak için aşağıdaki yöntemler desteklenir:

#### **public int getListenerPort ()**

**ListenerPort** özelliğinde tanımlandığı şekilde rota dinleyicisi kapısını alır

#### **public String getDestination()**

**Destination** özelliği tarafından tanımlandığı şekilde hedef adresi alır

#### **public int getDestinationPort ()**

**DestinationPort** özelliğinde tanımlandığı şekilde hedef dinleyici kapı adresini alır.

#### **public String getClientIPAddress ()**

bağlantı isteğinde bulunan istemcinin IP adresini alır

#### **public int getClientPortAddress()**

bağlantı isteğinde bulunan istemci tarafından kullanılan kapı adresini alır

#### **public int getTimeout()**

zamanasını değeri alır. MQIPT , **SecurityExitTimeout** özelliğinde tanımlandığı gibi, güvenlik çıkışının bir isteği doğrulamasını bekler

#### **public int getConnThreadID()**

Hata ayıklama amacıyla yararlı olan bağlantı isteğini işleyen bağlantı iş parçacığı tanıtıcısını alır

#### **public String getChannelName ()**

bağlantı isteğinde kullanılan IBM MQ kanal adını alır

#### **public String getQMName()**

bağlantı isteğinde kullanılan IBM MQ kuyruk yöneticisi adını alır

#### **genel Boole getTimedout()**

zamanasının süresinin dolup dolmadığını saptamak için güvenlik çıkışı tarafından kullanılabilir

## **MQIPT içinde com.ibm.mq.ipt.exit.SecurityExitResponse sınıfı**

Bu sınıf, kullanıcı tanımlı bir güvenlik çıkışından MQIPT ' e yanıt geçirmek için kullanılır ve bağlantı isteğinin kabul edilip edilmeyeceğini belirlemek için kullanılır.

Bu tipteki nesnelere yalnızca doğrulama yönteminde yaratılır (bkz. "[MQIPT içinde com.ibm.mq.ipt.exit.SecurityExit sınıfı](#)" sayfa 1045). Bu nesnelere yaratmak için uygun oluşturucular vardır ve her özellik için yöntemler vardır. Daha fazla bilgi için örnek güvenlik çıkışlarına bakın.

Varsayılan bir SecurityExitYanıt nesnesi yaratılması bağlantı isteğini reddeder.

## Olusturucular

- **public SecurityExitResponse (String dest, int destPort, int rc, String msg)**

Burada:

- dest yeni hedef
- destPort , yeni hedef kapı adresidir
- rc neden kodudur
- msg , bağlantı günlüğü girişine eklenecek bir iletidir

- **public SecurityExitResponse (String dest, int destPort, int rc)**
- **public SecurityExitYanıt (int rc, String msg)**
- **public SecurityExitResponse (int rc)**

## yöntemler

### **public void setDestination(String dest)**

bağlantı isteği için yeni bir hedef adres belirler

### **public void setDestinationPort (int port) throws IPTException**

Bağlantı isteği için yeni bir hedef dinleyici kapı adresi belirler-geçersiz bir kapı adresi için IPTException yayınla

### **public void setMessage(String msg)**

bağlantı günlüğü kaydına bir ileti ekler

### **public void setReasonCode (int rc)**

Bağlantı isteğine ilişkin neden kodunu belirler.

## MQIPT içindeki güvenlik çıkışı dönüş kodları

MQIPT ' in birçok farklı durumda bir güvenlik çıkışını ararken tanıdığı dönüş kodları.

Aşağıdaki dönüş kodları, aşağıdaki durumlarda bir güvenlik çıkışı çağrılırken MQIPT tarafından tanınır:

Dönüş kodu	Açıklama	Init	Validat e	yenile
ExitRc.-Tamam.	İstek başarıyla tamamlandı.	evet	evet	evet
ExitRc.GIRIŞ_HATASI	Kullanıma İnit isteği başarısız oldu, rota devre dışı bırakılacak.	evet		
ExitRc.YENILEME_HATASI	Yenileme isteği başarısız oldu.			evet
ExitRc.YETKILI DEĞİL_yetkili	Doğrulama işlemi başarısız oldu, bağlantı isteği reddedildi.		evet	
ExitRc.DISABLE_SSL	Doğrulama isteği başarılı oldu, hedefle bağlantı SSL ya da TLS kullanmayacak.		evet	

## MQIPT içinde kapı numarası denetimi

MQIPTkullanılırken, giden bağlantı kurulurken kullanılan yerel kapı numarası aralığını sınırlamak mümkündür.

İlk yerel kapı numarasını belirtmek için rotadaki **OutgoingPort** özelliğini ayarlayın ve kullanılacak kapı sayısını belirtmek için **MaxConnectionThreads** değerini belirleyin. Örneğin, **OutgoingPort** değerini 1600 ve **MaxConnectionThreads** değerini 20olarak ayarlarsanız, bu rotaya ilişkin yerel kapı numaraları aralığı 1600-1619 olur.

Rotalar arasında çakışma olmadığından emin olmak MQIPT yöneticisinin sorumluluğundadır.

**OutgoingPort** tanımlı değilse, varsayılan değer olan 0 , her bağlantı için sistem tarafından ayrılmış bir kapı numarasının kullanıldığı anlamına gelir.

HTTPkullanılırken, giden kapı sayısı, HTTPkullanılmadığında iki kat daha fazladır. Önceki örnekte, rota HTTPkullanıyorsa, sayı aralığı 1600-1639 arasındadır.

Ek bilgi için [Kapı numaralarının ayrılması](#) başlıklı konuya bakın.

## Çok yönlü sistemler

Çok anasistemli bir sistem kullanırken, **LocalAddress** özelliğini kullanarak giden bağlantının hangi IP adresine bağlanacağını belirtebilirsiniz. Bu özellikte anasistem adları desteklenmez.

## MQIPT içinde saklanan parolaları şifreleme

MQIPT yapılandırması, komut kapısını kullanarak MQIPT ' e erişmek için kullanılan parolanın yanı sıra çeşitli kaynaklara erişmek için parolaları da içerebilir. IBM MQ 9.2.0' dan tüm bu parolalar şifrelenerek korunmalıdır.

### Bu görev hakkında

IBM MQ 9.2.0sürümünden önceki sürümlerde, yalnızca anahtar halkalarına ya da şifreleme donanımı anahtar depolarına erişmek için MQIPT tarafından kullanılan parolalar şifrelenebilir. Şifrelenmiş parolalar, **SSL\*KeyRingPW** özelliklerinden herhangi birinin başvurduğu dosyalarda saklanır. LDAP sunucuları ve MQIPT erişim parolası için diğer parolalar, `mqipt.conf` yapılandırma dosyasında düz metin olarak depolanır.

IBM MQ 9.2.0' den MQIPT tarafından kullanılacak tüm saklanmış parolalar, parola **mqiptPW** komutuyla şifrelenerek korunmalıdır. Şifrelenmiş parolalar, `mqipt.conf` yapılandırma dosyasında özellik değerleri olarak depolanır. MQIPT , şifrelenmiş parolaları, düz metin parolalarını ve özellik değerlerindeki dosya adlarını ayırt edebilir. En güvenli koruma yöntemi olduğundan, MQIPT tarafından kullanılmak üzere saklanan tüm parolaları bu şekilde şifrelemelisiniz.

**Deprecated** MQIPT 'dan önce IBM MQ 9.2.0 içinde kullanılan anahtar deposu parolalarını şifreleme yöntemi kullanımdan kaldırılmıştır, ancak IBM MQ 9.2.0' den önce var olan yapılandırma özellikleri için kullanılmaya devam edilebilir. Anahtarlık parolalarının korunmasını geliştirmek için, daha önce şifrelenmiş olan tüm anahtarlık parolalarını en son koruma yöntemini kullanarak yeniden şifreleyin.

**Not:** `mqipt.conf` yapılandırma dosyasındaki **SSLCommandPortKeyRingPW** özelliği ve **mqiptAdmin** özellikler dosyasındaki **SSLClientCAKeyRingPW** özelliği, parola dosyalarına başvuramaz. Bu özelliklerin değerleri, **mqiptPW** komutu tarafından şifrelenmiş parola dizgisi çıkışına ayarlanmalıdır.

MQIPT yapılandırmasında düz metin ya da zayıf korumalı parola varsa, MQIPT başlatıldığında ya da bir rota başlatıldığında bir uyarı iletisi yayınlanır.

En son koruma yöntemini kullanarak MQIPT tarafından kullanılmak üzere saklanacak bir parolayı şifrelemek için bu yordamı kullanın. Bir anahtarlık parolasını MQIPT before IBM MQ 9.2.0içinde şifrelemek için “IBM MQ 9.2.0 içinde MQIPT ' den önce bir anahtarlık parolasını şifreleme” sayfa 1049içindeki adımları izleyin.

### Yordam

1. İsteğe bağlı: Önceden yoksa, parola şifreleme anahtarını içeren bir dosya oluşturun.

MQIPT , parolaları şifrelemek için bir şifreleme anahtarı kullanır. Bir dosyada kendi şifreleme anahtarınızı belirtebilirsiniz. Dosya en az bir karakter ve yalnızca bir metin satırı içermelidir.

MQIPTörneğine ilişkin tüm depolanan parolaları şifrelemek ve parolaların şifresini çözmek için aynı parola şifreleme anahtarı kullanılır. Bu nedenle, her MQIPT kuruluşu için yalnızca tek bir parola şifreleme anahtarı dosyası gerekir.

**mqiPTAdmin** özellikler dosyasında saklanan parolaları şifrelemek için, MQIPT yapılandırmasında parolaları şifrelemek için kullanılan şifreleme anahtarından farklı bir parola şifreleme anahtarı kullanabilirsiniz.

MQIPT ' u otomatik olarak başlatılan bir hizmet olarak çalıştırmayı planlıyorsanız, parola şifreleme anahtarı dosyasını `mqiPT_cred`.key varsayılan adıyla oluşturmanız ve MQIPT ana dizinine yerleştirmeniz gerekir.

Bir parola şifreleme anahtarı belirtmeniz gerekmez, ancak bunu yapmak daha güvenlidir. Kendi şifreleme anahtarınızı belirtmezseniz, varsayılan şifreleme anahtarı kullanılır.

**Not:** Yetkisiz kullanıcıların şifreleme anahtarını okumasını önlemek için parola şifreleme anahtarı dosyasında uygun dosya izinlerinin ayarlandığından emin olmanız gerekir. Yalnızca **mqiPTPW** komutunu çalıştıran kullanıcı ve MQIPT ' un çalıştırıldığı kullanıcı, parola şifreleme anahtarını okumak için yetkiye gereksinim duyar.

2. **mqiPTPW** komutunu kullanarak parolayı şifreleyin.

**mqiPTPW** komutunun sözdizimi `mqiPTPW (saklanmış parolayı şifrele)` içinde açıklanmıştır.

“1” sayfa 1048. adımda bir parola şifreleme anahtarı dosyası oluşturduysanız, **mqiPTPW** için **-sf** parametresini kullanarak dosya adını belirtin. Örneğin, **-sf** parametresiyle belirtilen dosyadaki şifreleme anahtarı kullanılarak bir parolayı şifrelemek için aşağıdaki komut verilebilir:

```
mqiPTPW -sf /opt/mqiPT/mqiPT_password.key
```

3. İstendiğinde şifrelenecek parolayı girin.

Şifrelenmiş parolanın çıkışı **mqiPTPW** tarafından yapılacaktır.

4. Şifrelenmiş parolayı `mqiPT.conf` yapılandırma dosyasında ya da **mqiPTAdmin** özellikler dosyasında uygun özelliğe kopyalayın.

Örneğin, aşağıdaki satır MQIPT erişim parolası için şifrelenmiş bir parola belirtir:

```
AccessPW=<mqiPTPW>1!QL+2Jvj/tigKK1D7Nz80qw==!AMDBef0UımPf5i10uqV5MA==
```

5. MQIPT yazılımını başlatın. “1” sayfa 1048 . adımda varsayılan addan farklı bir adla bir parola şifreleme anahtarı dosyası oluşturduysanız, MQIPT ' u başlatırken şifreleme anahtarı dosyasının adını belirtin.

MQIPT ' u başlatırken **-sf** parametresini kullanarak parola şifreleme anahtarı dosyasının adını belirtebilirsiniz. Örneğin, **-sf** parametresiyle belirtilen dosyadaki şifreleme anahtarını kullanarak MQIPT ' i başlatmak için aşağıdaki komutu verin:

```
mqiPT /opt/mqiPT -sf /opt/mqiPT/mqiPT_password.key
```

MQIPT ' u başlatırken parola şifreleme anahtarı dosya adını belirlemeye ilişkin diğer yöntemler hakkında bilgi için bkz. [Parola şifreleme anahtarının belirtilmesi](#).

**mqiPTAdmin** özellikler dosyasındaki **PasswordProtectionKeyFile** özelliğini kullanarak **mqiPTAdmin** komutu için parola şifreleme anahtarı dosyasının adını belirtebilirsiniz.

## IBM MQ 9.2.0 içinde MQIPT ' den önce bir anahtarlık parolasını şifreleme

IBM MQ 9.2.0' den önce, MQIPT tarafından kullanılan anahtar halkalarına erişmek için kullanılan şifrelenmiş parolalar dosyalarda saklanır.

### Bu görev hakkında

MQIPT before IBM MQ 9.2.0 tarafından kullanılmak üzere bir anahtarlık parolasını şifrelemek için bu görevdeki yordamı izleyin. IBM MQ 9.2.0 for Long Term Support içinde MQIPT içinde, “MQIPT içinde saklanan parolaları şifreleme” sayfa 1048 içinde açıklanan daha güvenli koruma yöntemini kullanın.

### Yordam

1. **mqiPTPW** komutunu kullanarak anahtarlık parolasını şifreleyin.

Parolayı şifrelemek için aşağıdaki komutu girin:

```
mqiPT password filename
```

burada:

### **parola**

anahtarlığa erişmek için gereken açık metin parolasıdır

### **Dosya adı**

yaratılacak parola dosyasının adı

**mqiPT** komutunun sözdizimi **mqiPT** (saklanmış parolayı şifrele) içinde açıklanmıştır.

2. Uygun rota özelliğini, "1" sayfa 1049. adımda oluşturulan şifrelenmiş parolayı içeren dosyanın adına ayarlayın.

Örneğin, MQIPT TLS sunucu sertifikasını içeren anahtarlık için parola dosyasını belirtmek üzere `mqiPT.conf` yapılandırma dosyasına aşağıdaki satırı ekleyin:

```
SSLServerKeyRingPW=filename
```

## **MQIPT ile ilgili diğer güvenlik konuları**

MQIPT , tasarımcının güvenli bir çözüm oluşturmasına yardımcı olan birkaç ek işleve sahiptir.

- Bir iç ağda giden bağlantılar kurmaya çalışan çok sayıda istemci varsa, bunların tümü güvenlik duvarının içinde bulunan bir MQIPT içinden geçebilir. Güvenlik duvarı yöneticisinin daha sonra yalnızca MQIPT bilgisayarına dış erişim vermesi gerekir.
- MQIPT , MQIPT bir SOCKS yetkili sunucusu ya da bir güvenlik çıkışı kullanmıyorsa, yalnızca yapılanış dosyasında belirttik olarak yapılandırılmış olan kuyruk yöneticilerine bağlanabilir.
- MQIPT , aldığı ve ilettiği iletilerin geçerli olduğunu doğrular ve IBM MQ iletişim kuralına uyar. Bu, MQIPT ' in IBM MQ iletişim kuralı dışındaki güvenlik saldırıları için kullanılmasını önlemeye yardımcı olur. MQIPT bir SSL/TLS yetkili sunucusu olarak davranıyorsa, tüm IBM MQ verileri ve iletişim kuralları şifrelendiğinde MQIPT yalnızca ilk SSL/TLS el sıkışmasını garanti edebilir. Bu durumda [Java security manager](#) kullanın.
- MQIPT , kanal çıkışlarının kendi uçtan uca güvenlik iletişim kurallarını çalıştırmasına olanak sağlar.
- `MaxConnectionThreads` özelliğini ayarlayarak gelen bağlantıların toplam sayısını kısıtlayabilirsiniz. Bu, güvenlik açığı bulunan bir iç kuyruk yöneticisinin hizmet reddi saldırılarından korunmasına yardımcı olur.

### **Yapılandırma dosyası**

MQIPT yapılandırma dosyasını (`mqiPT.conf`), yetkisiz kullanıcılar tarafından okunmaya karşı korumanız gerekir; bu dosya, MQIPT' e uzaktan yönetici erişimini denetleyen **AccessPW** parolası gibi hassas bilgiler içerebilir. "MQIPT içinde saklanan parolaları şifreleme" sayfa 1048 içindeki yordamı izleyerek yapılandırma dosyasında belirtilen tüm parolaları koruyun. Ayrıca, `mqiPT.conf` ' in yetkisiz değişikliklere karşı korunduğundan emin olun. `mqiPT.conf` işletim sistemi dosya izinlerini, yalnızca MQIPT işletim sistemini çalıştıran kullanıcı hesabının dosyayı okuyabileceği ya da güncelleyebileceği şekilde ayarlayın.

### **Komut kapısı**

MQIPT komut kapıları, ağ üzerinden **mqiPTAdmin** komutuyla uzak bir MQIPT yönetim ortamına verilen yönetim komutlarını kabul eder.

IBM MQ 9.2.0 içinden MQIPT , güvenli olmayan bir komut kapısı ve TLS ile güvenli kılınan bir komut kapısı ile yapılandırılabilir. Güvenli olmayan komut kapısına yapılan bağlantılar şifrelenmez.

**Not:** Ağ üzerinden MQIPT erişim parolası da dahil olmak üzere güvenli olmayan komut kapısına gönderilen veriler, ağdaki diğer kullanıcılar tarafından görülebilir.

Güvenli olmayan ya da TLS komut kapısını etkinleştirmeden önce bir komut kapısını etkinleştirmeniz gerekip gerekmediğini göz önünde bulundurmanız ve MQIPT tuzak yönetimine izin verme risklerini

değerlendirmeniz gerekir. IBM MQ 9.2.0' den **mqiptAdmin** komutu, komut kapısı kullanmadan **mqiptAdmin** komutuyla aynı kullanıcı altında çalışan MQIPT yerel yönetim ortamlarını yönetebilir. Bu nedenle, MQIPT' in yerel yönetim ortamlarını yönetmek için bir komut kapısını etkinleştirmeniz gerekmez.

Güvenli olmayan ya da TLS komut kapısı etkinleştirildiyse, komut kapısına yetkisiz erişimi önlemelisiniz. Örneğin, komut kapısına erişimi güvenceye almak için aşağıdaki noktaları göz önünde bulundurmalısınız:

- MQIPT komut kapısına bağlanabilen bilgisayar kümesini sınırlamak için bir güvenlik duvarı kullanın.
- **AccessPW** ve **RemoteCommandAuthentication** özelliklerini kullanarak komut kapılarında kimlik doğrulamasını etkinleştirin. Komut kapısı kimlik doğrulamasını etkinleştirme hakkında daha fazla bilgi için bkz. [Komut kapısı kimlik doğrulaması](#).
- **RemoteShutdown** özelliğiyle uzaktan kapatmayı devre dışı bırakmayı düşünün.
- Komut kapılarını belirli bir ağ arabiriminde dinleyecek şekilde yapılandırmak için **CommandPortListenerAddress** ve **SSLCommandPortListenerAddress** özelliklerini kullanmayı düşünün.

**mqiptAdmin** komutunu MQIPT yönetmek için kullanma hakkında daha fazla bilgi için bkz. [MQIPT komutunu komut satırını kullanarak yönetme](#).

## MQIPT içinde bağlantı günlükleri

MQIPT , tüm başarılı ve başarısız bağlantı girişimlerinin listelerini içeren bir bağlantı günlüğü olanağı sağlar.

Bir MQIPT rotası tarafından alınan ya da yapılan her bağlantı için ve MQIPT tarafından alınan her denetim komutu için bağlantı günlüğüne bir giriş yazılır. Bağlantı günlüğü, **ConnectionLog** ve **MaxLogFileSize** özellikleri kullanılarak denetlenir. Daha fazla bilgi için bkz. [MQIPT genel özellikler](#) .

MQIPT ' in her başlatılışında yeni bir bağlantı günlüğü yaratılır. Tanımlama için, dosya adı geçerli zaman damgasını içerir; örneğin:

```
mqiptyyyyymmddhhmmss.log
```

burada:

YYYY yıl  
MM ay  
DD gün  
HH saatleri  
mm dakika  
SS saniyedir

Bir bağlantı günlüğü, **MaxLogFileSize** özelliğinin belirlediği büyüklük üst sınırına ulaştığında, bir yedek dosya (mqipt001.log) yaratılır. En çok iki yedek dosya korunur (mqipt001.log ve mqipt002.log).

Bağlantı günlüğündeki bir giriş, bağlantı isteğinin her bir bölümünü gösterir. MQIPT tarafından alınan bir bağlantı isteği ve MQIPT ' un hedef adrese yaptığı yeni bağlantı, iki günlük girdisi olarak ve her bağlantı sona erdirildiğinde daha sonra iki giriş olarak görünür.

Başarılı bir bağlantı isteğine ilişkin bağlantı günlüğü:

```
Wed May 15 13:13:51 BST 2013 conn accept 127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0  
Wed May 15 13:13:51 BST 2013 conn conn 127.0.0.1(3843) localhost(3500) OK 5000-0  
Wed May 15 13:13:52 BST 2013 conn close 127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0  
Wed May 15 13:13:52 BST 2013 conn close 127.0.0.1(3843) localhost(3500) OK 5000-0
```

Başarısız bir bağlantı isteğine ilişkin bir bağlantı günlüğü:

```
Wed May 15 14:56:40 BST 2013 conn accept 127.0.0.1(4138) 127.0.0.1(7000) OK 7000-0
```

## Bağlantı günlüğü girişleri

Her bağlantı günlüğü girişi aşağıdaki bilgileri içerir:

- Girdinin oluşturulduğu saat.
- Girdi türü. Değer aşağıdaki değerlerden biri olabilir:

### yönetim

Yönetim komutu

### konn

Rota bağlantısı

- Oluşan olay. Değer aşağıdaki değerlerden biri olabilir:

### kabul et

Bağlantı isteği alındı

### kapat

Bağlantı kapatıldı

### konn

Rota hedefine bağlantı isteği

### disipt

Alınan MQIPT komutunu görüntüle

### veri yok

Çağırandan veri alınmadı

### Ping

Ping isteği alındı

### durum

Alınan durum komutunu görüntüle

### yeniden

Yenileme komutu alındı

### Durdur

Durdurma komutu alındı

- Kaynak ağ adresi ve kapı numarası. LOCAL değeri, komut kapısı kullanılmadan yerel olarak verilen yönetim komutları için görüntülenir.
- Hedef ağ adresi ve kapı numarası. Bu, komut kapısı kullanılmadan yerel olarak verilen yönetim komutları için görüntülenmez.
- Tamamlanma kodu. Değer OK ya da ERROR olabilir.
- MQIPT iş parçacığı tanıtıcısı.
- İsteğe bağlı bir hata iletisi.

## IBM MQ Internet Pass-Thru ' nin kapsayıcılar kullanılarak yapılandırılması

IBM MQ Internet Pass-Thru (MQIPT) uygulamasını bir kapsayıcıda çalıştırabilirsiniz. Kapsayıcı görüntü tarafından kullanılan temel görüntü, desteklenen bir Linux işletim sistemini kullanmalıdır.

### Yordam

- Örnek bir MQIPT Docker görüntüsü, mq-container GitHub havuzunda bulunur. Kapsayıcıyı oluşturmak ve çalıştırmak için [IBM MQ Internet Pass-Thru açık Docker](#) içindeki yönergeleri izleyin.



## Sonraki adım

Çalışan kapsayıcıları **docker ps** komutunu kullanarak görüntüleyebilirsiniz. Bir Docker kapsayıcısında çalışan MQIPT konsol çıkışı için **docker logs \${CONTAINER\_ID}** komutunu kullanın.

## V 9.3.0 Akış kuyruklarının yapılandırılması

Akıtmalı kuyruklar özelliği, her iletinin bir kopyasını ikinci bir kuyruğa göndermenizi sağlar. Akış kuyruklarının yapılandırılması, kuyruk temelinde bir kuyrukta gerçekleştirilir.

Yerel ve model kuyruklarının akış kuyruklarıyla ilgili iki yeni özneliği vardır:

### STREAMQ

Bu, akıtılan iletilerin teslim edilmesi gereken kuyruğun adıdır. **STREAMQ** özneliğini başka bir kuyruğun adına ayarlamalısınız.

Kuyrukların, iletileri diğer kuyruklara akıtılabilecek şekilde yapılandırılabilmesine ilişkin kısıtlamalar vardır ve kuyrukların, akıtılan iletiler için hedef olarak belirlenmesine ilişkin kısıtlamalar vardır. İleti akışı kısıtlamalarına ilişkin bilgi için bkz. [Akış kuyruğu kısıtlamaları](#).

### STRMQOS

Bu, akıtılan iletiler teslim edilirken kullanılacak hizmet kalitesi.

**STRMQOS** özneliğini iki değerden birine ayarlayabilirsiniz:

#### BESTEF

Varsayılan değer olan en iyi çalışma.

Kuyruk yöneticisi, **STREAMQ** özneliğinde belirtilen kuyruğa her iletinin bir kopyasını teslim etmeye çalışır. Akıtılan ileti sağlanmasında bir sorun varsa, bu, özgün iletinin teslimini etkilemez.

#### MUSTDUP

Kuyruk yöneticisi, her iletinin bir kopyasını akış kuyruğuna teslim etmeye çalışır.

Akıtılan ileti sağlanmasında bir sorun varsa, özgün ileti kuyruğuna teslim edilmez ve uygulama uygun bir neden koduyla birlikte MQCC\_FAILED ' i alır.

Daha fazla ayrıntı için [ALTER kuyrukları](#), [DEFINE kuyrukları](#) ve [DISPLAY QUEUE MQSC komutları](#) ve [Change, Copy, and Create Queue](#), [Inquire Queue](#) ve [Inquire Queue \(Response\) PCF komutlarına](#) bakın.

Her iletinin birden çok kopyası gerekiyorsa, **STREAMQ** özneliğini, hedefi bir IBM MQ konusuna başvuran IBM MQ diğer ad kuyruğunun adına gönderme yapmak üzere yapılandırabilirsiniz. Bir ileti özgün kuyruğa konduğunda, iletinin bir kopyası belirtilen konuya yayınlanır.

Her abonelik iletinin bir kopyasını aldığından, konu nesnesine ilişkin API ' nizin ya da yönetilen abonelikleriniz olduğundan emin olmanız gerekir. Abonelere gönderilen ileti, diğer yayınlama/abone olma iletileriyle aynı kuralları izler. Örneğin, her iletinin yeni bir ileti tanıtıcısı vardır ve MQMD ' nin bağlam alanları özgün iletidekilerden farklıdır. Özgün ve akış halindeki iletiler arasındaki benzerlikler ve farklılıklar hakkında daha fazla bilgi için bkz. [Akış iletileri](#).

## Örnekler

### En iyi çalışma örneği

Aşağıdaki örnekte, bir yerel kuyruk ORDERS.QUEUE , akıtılan iletileri ikinci bir kuyruğa ANALYTICS.QUEUE. BESTEF hizmet kalitesi, akış iletisinin ANALYTICS.QUEUE, örneğin, ANALYTICS.QUEUE dolu, özgün ileti yine de ORDERS.QUEUE.

Bu tür bir yapılandırma, özgün iletiler sipariş kuyruğuna yerleştirilirken ve işlenirken, akış halindeki iletiler analiz edilerek, alınan siparişler üzerinde analitik gerçekleştirmek için kullanılabilir. Akış kuyruğu özelliğinin bir yararı, akış halindeki iletileri ANALYTICS.QUEUE İş tarafından karşılanan gerçek siparişleri etkilemeden işlenmeyi bekliyor.

```
DEFINE QLOCAL (ANALYTICS.QUEUE)
```

```
ALTER QLOCAL (ORDERS.QUEUE) STRMQOS (BESTEF) STREAMQ (ANALYTICS.QUEUE)
```

**Not:** **STRMQOS** örneğinde BESTEFolarak ayarlanmıştır, ancak BESTEF varsayılan hizmet kalitesi olduğundan bu özneliği **ALTER** komutunun dışında bırakabilirsiniz.

### Örnek yinelenmeli

Bu örnekte, yerel bir kuyruk PAYMENTS.QUEUE , her iletinin akıtılan kopyalarını başka bir yerel kuyruğa AUDIT.QUEUE. Ödeme kuyruğuna konan her iletinin denetim kuyruğuna iletilmesi önemlidir, bu nedenle MUSTDUP hizmet kalitesi kullanılır.

Akıtılan ileti kuyruğuna teslim edilirse, özgün ileti de teslim edilmez ve uygulama uygun bir tamamlanma ve neden kodu alır. Uygulama, yalnızca tek bir kuyruk varsa, koyma işlemini aynı şekilde yeniden denemelidir.

```
DEFINE QLOCAL (AUDIT . QUEUE)
```

```
ALTER QLOCAL (PAYMENTS . QUEUE) STRMQOS (MUSTDUP) STREAMQ (AUDIT . QUEUE)
```

### Notlar:

1. Özgün kuyruk değiştirilirken akış kuyruğunun var olması gerekmez. Ancak, kullanılmakta olan hizmet kalitesi MUSTDUP olduğu için, iletileri özgün kuyruğa koyma girişimlerinin, akış kuyruğunu tanımlayınca kadar başarısız olduğunu unutmayın.
2. Bir konu nesnesinin hedefiyle bir kuyruk diğer adı kullanılırken, herhangi bir abone yoksa, akış iletisinin teslim edilmesi başarılı olarak kabul edilir ve özgün ileti kuyruğuna teslim edilir.
3. Akıtılan bir ileti kuyruğuna teslim edilemezse, kuyruk yöneticisi iletiyi gitmeyen ileti kuyruğuna teslim etme girişiminde bulunmaz. Ancak, akıtılan bir ileti uzak bir kuyruğa gönderilirse, bir kanal üzerinden başka bir kuyruk yöneticisine geçtiğinde, ileti, var olan teslim edilmeyen mektup kurallarına uygun olmayan bir ileti kuyruğuna teslim edilebilir.

### Akış kuyruğunun yapılandırılması

Akış kuyruğunda ek yapılandırma gerçekleştirilmesine gerek yoktur. Bu ileti, kuyruğunu bir akış kuyruğu olarak adlandır herhangi bir kuyruktan ileti alır. Ancak, akış kuyruğunda yapılandırılan öznelik değerlerini dikkate almak mantıklı olabilir.

Örneğin, özgün kuyrukta 100.000 derinlik üst sınırı varsa ve akış kuyruğu yalnızca 5000 derinlik üst sınırına sahipse, STRMQOS BESTEF olarak ayarlanırsa akımlı ileteler kaybolabilir ya da STRMQOS MUSTDUP olarak ayarlandıysa, MQRC\_Q\_FULL hatasıyla, özgün kuyrukta yeterli yer kaldıysa bile, geçiş zamanında başarısız olabilir.

Akış kuyruğundaki hangi özneliklerin, özgün kuyruğun nasıl yapılandırıldığına bağlı olarak, uygun değerlere sahip olması gerekebileceğini göz önünde bulundurun.

### İlgili kavramlar

[Akış kuyrukları](#)

## Özel notlar

Bu belge, ABD'de kullanıma sunulan ürünler ve hizmetler için hazırlanmıştır.

IBM, bu belgede sözü edilen ürün, hizmet ya da özellikleri diğer ülkelerde kullanıma sunmayabilir. Bulduğunuz yerde kullanıma sunulan ürün ve hizmetleri yerel IBM müşteri temsilcisinden ya da çözüm ortağınızdan öğrenebilirsiniz. Bir IBM ürün, program ya da hizmetine gönderme yapılması, açık ya da örtük olarak yalnızca o IBM ürünü, programı ya da hizmetinin kullanılabilirliğini göstermez. Aynı işlevi gören ve IBM'in fikri mülkiyet haklarına zarar vermeyen herhangi bir ürün, program ya da hizmet de kullanılabilir. Ancak, IBM dışı ürün, program ya da hizmetlerle gerçekleştirilen işlemlerin değerlendirilmesi ve doğrulanması kullanıcının sorumluluğundadır.

IBM'in, bu belgedeki konularla ilgili patentleri ya da patent başvuruları olabilir. Bu belgenin size verilmiş olması, patentlerin izinsiz kullanım hakkının da verildiği anlamına gelmez. Lisansla ilgili sorularınızı aşağıdaki adrese yazabilirsiniz:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Çift byte (DBCS) bilgilerle ilgili lisans soruları için, ülkenizdeki IBM'in Fikri Haklar (Intellectual Property) bölümüyle bağlantı kurun ya da sorularınızı aşağıda adrese yazın:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japonya

**İzleyen paragraf, bu tür kayıt ve koşulların, yasalarıyla bağdaşmadığı ülkeler ya da bölgeler için geçerli değildir:** IBM BU YAYINI, "OLDUĞU GİBİ", HİÇBİR KONUDA AÇIK YA DA ÖRTÜK GARANTİ VERMEKSİZİN SAĞLAMAKTADIR; TİCARİ KULLANIMA UYGUNLUK AÇISINDAN HER TÜRLÜ GARANTİ VE BELİRLİ BİR AMACA UYGUNLUK İDDİASI AÇIKÇA REDDEDİLİR. Bazı ülkeler bazı işlemlerde garantinin açık ya da örtük olarak reddedilmesine izin vermez; dolayısıyla, bu bildirim sizin için geçerli olmayabilir.

Bu yayın teknik yanlışlar ya da yazım hataları içerebilir. Buradaki bilgiler üzerinde düzenli olarak değişiklik yapılmaktadır; söz konusu değişiklikler sonraki basımlara yansıtılacaktır. IBM, önceden bildirimde bulunmaksızın, bu yayında açıklanan ürünler ve/ya da programlar üzerinde iyileştirmeler ve/ya da değişiklikler yapabilir.

Bu belgede IBM dışı Web sitelerine yapılan göndermeler kullanıcıya kolaylık sağlamak içindir ve bu Web sitelerinin onaylanması anlamına gelmez. Bu Web sitelerinin içerdiği malzeme, bu IBM ürününe ilişkin malzemenin bir parçası değildir ve bu tür Web sitelerinin kullanılmasının sorumluluğu size aittir.

IBM'e bilgi ilettiğinizde, IBM bu bilgileri size karşı hiçbir yükümlülük almaksızın uygun gördüğü yöntemlerle kullanabilir ya da dağıtabilir.

(i) Bağımsız olarak yaratılan programlarla, bu program da içinde olmak üzere diğer programlar arasında bilgi değiş tokuşuna ve (ii) değiş tokuş edilen bilginin karşılıklı kullanımına olanak sağlamak amacıyla bu program hakkında bilgi sahibi olmak isteyen lisans sahipleri şu adrese yazabilirler:

IBM Corporation  
Yazılım Birlikte Çalışabilirlik Koordinatörü, Bölüm 49XA  
3605 Karayolu 52 N  
Rochester, MN 55901  
U.S.A.

Bu tür bilgiler, ilgili kayıt ve koşullar altında ve bazı durumlarda bedelli olarak edinilebilir.

Bu belgede açıklanan lisanslı program ve bu programla birlikte kullanılacak tüm lisanslı malzeme, IBM tarafından IBM Müşteri Sözleşmesi, IBM Uluslararası Program Lisans Sözleşmesi ya da taraflar arasında yapılan herhangi bir eşdeğer sözleşmenin koşulları kapsamında sağlanır.

Burada belirtilen performans verileri denetimli bir ortamda elde edilmiştir. Bu nedenle, başka işletim ortamlarında çok farklı sonuçlar alınabilir. Bazı ölçümler geliştirilme düzeyindeki sistemlerde yapılmıştır ve bu ölçümlerin genel kullanıma sunulan sistemlerde de aynı olacağı garanti edilemez. Ayrıca, bazı sonuçlar öngörü yöntemiyle elde edilmiş olabilir. Dolayısıyla, gerçek sonuçlar farklı olabilir. Bu belgenin kullanıcıları, kendi ortamları için geçerli verileri kendileri doğrulamalıdır.

IBM dışı ürünlerle ilgili bilgiler, bu ürünleri sağlayan firmalardan, bu firmaların yayın ve belgelerinden ve genel kullanıma açık diğer kaynaklardan alınmıştır. IBM bu ürünleri sinamamıştır ve IBM dışı ürünlerle ilgili performans doğruluğu, uyumluluk gibi iddiaları doğrulayamaz. IBM dışı ürünlerin yeteneklerine ilişkin sorular, bu ürünleri sağlayan firmalara yöneltilmelidir.

IBM'in gelecekteki yönelim ve kararlarına ilişkin tüm bildirimler değişebilir ve herhangi bir duyuruda bulunulmadan bunlardan vazgeçilebilir; bu yönelim ve kararlar yalnızca amaç ve hedefleri gösterir.

Bu belge, günlük iş ortamında kullanılan veri ve raporlara ilişkin örnekler içerir. Örneklerin olabildiğince açıklayıcı olması amacıyla kişi, şirket, marka ve ürün adları belirtilmiş olabilir. Bu adların tümü gerçek dışıdır ve gerçek iş ortamında kullanılan ad ve adreslerle olabilecek herhangi bir benzerlik tümüyle rastlantıdır.

#### YAYIN HAKKI LİSANSI:

Bu belge, çeşitli işletim platformlarında programlama tekniklerini gösteren, kaynak dilde yazılmış örnek uygulama programları içerir. Bu örnek programları, IBM'e herhangi bir ödemede bulunmadan, örnek programların yazıldığı işletim altyapısına ilişkin uygulama programlama arabirimiyle uyumlu uygulama programlarının geliştirilmesi, kullanılması, pazarlanması ya da dağıtılması amacıyla herhangi bir biçimde kopyalayabilir, değiştirebilir ve dağıtabilirsiniz. Bu örnekler her koşul altında tüm ayrıntılarıyla sinanmamıştır. Dolayısıyla, IBM bu programların güvenilirliği, bakım yapılabilirliği ya da işlevleri konusunda açık ya da örtük güvence veremez.

Bu bilgileri elektronik kopya olarak görüntülediyseniz, fotoğraflar ve renkli resimler görünmeyebilir.

## Programlama arabirimi bilgileri

Sağlandıysa, programlama arabirimi bilgileri, bu programla birlikte kullanılmak üzere uygulama yazılımı oluşturmanıza yardımcı olmak amacıyla hazırlanmıştır.

Bu kitapta, müşterinin WebSphere MQ hizmetlerini elde etmek üzere program yazmasına olanak sağlayan amaçlanan programlama arabirimlerine ilişkin bilgiler yer alır.

Ancak, bu bilgiler tanılama, değiştirme ve ayarlama bilgilerini de içerebilir. Tanılama, değiştirme ve ayarlama bilgileri, uygulama yazılımlarınızda hata ayıklamanıza yardımcı olur.

**Önemli:** Bu tanılama, değiştirme ve ayarlama bilgilerini bir programlama arabirimi olarak kullanmayın; bu bilgiler değişebilir.

## Ticari Markalar

IBM, IBM logosu, ibm.com, IBM Corporation 'ın dünya çapında birçok farklı hukuk düzeninde kayıtlı bulunan ticari markalarıdır. IBM ticari markalarının güncel bir listesine Web üzerinde "Copyright and trademark information" [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) (Telif hakkı ve ticari marka bilgileri) başlıklı konudan ulaşılabilir. Diğer ürün ve hizmet adları IBM'in veya diğer şirketlerin ticari markaları olabilir.

Microsoft ve Windows, Microsoft Corporation firmasının ABD'de ve/ya da diğer ülkelerdeki markalarıdır.

UNIX, The Open Group şirketinin ABD ve diğer ülkelerdeki tescilli ticari markasıdır.

Linux, Linus Torvalds'ın ABD ve/ya da diğer ülkelerdeki tescilli ticari markasıdır.

Bu ürün, Eclipse Project (<https://www.eclipse.org/>) tarafından geliştirilen yazılımları içerir.

Java ve Java tabanlı tüm markalar ve logolar, Oracle firmasının ve/ya da iřtiraklerinin markaları ya da tescilli markalarıdır.







Parça numarası:

(1P) P/N: