

9.1

*IBM MQ ' nun Planlanması*

**IBM**

**Not**

Bu bilgileri ve desteklediđi ürünü kullanmadan önce, "[Özel notlar](#)" sayfa 219 bölümündeki bilgileri okuyun.

Bu basım, yeni basımlarında tersi belirtilmediđi sürece, IBM® MQ sürüm 9 yayın düzeyi 1 'i ve sonraki tüm yayın düzeyleri ve deđişiklikler için geçerlidir.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

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

# İçindekiler

<b>Planlama.....</b>	<b>5</b>
IBM MQ yayın tipleri: planlama ile ilgili önemli noktalar.....	5
IBM MQ ve IBM MQ Appliance şirket içinde GDPR ' ye hazır olma ile ilgili önemli noktalar.....	8
Tek bir kuyruk yöneticisine dayalı mimariler.....	17
Birden çok kuyruk yöneticisine dayalı mimariler.....	18
Dağıtılmış kuyruklarınızın ve kümelerinizin planlanması.....	19
Dağıtılmış yayınlama/abone olma ağınızın planlanması.....	84
Çoklu Platformlar üzerinde depolama ve performans gereksinimlerinizin planlanması.....	123
Çoklu Platformlar üzerindeki Disk Alanı Gereksinimleri.....	124
Çoklu Platformlar üzerinde Dosya Sistemi Desteği Planlanması.....	127
Çoklu Platformlar üzerinde MFT için planlama dosya sistemi desteği.....	154
Çoklu platformlar üzerinde dairesel ya da doğrusal oturum açma seçilmesi.....	155
AIXüzerinde paylaşılan bellek.....	155
IBM MQ ve UNIX System V IPC kaynakları.....	156
IBM MQ ve UNIX Process Priority.....	156
z/OSüzerinde IBM MQ ortamınızın planlanması.....	156
Kuyruk yöneticinizin planlanması.....	157
Kanal başlatıcınızın planlanması.....	183
Kuyruk paylaşım grubunuzun planlanması (QSG).....	187
Yedekleme ve kurtarma planlanması.....	200
z/OS UNIX ya da UNIX System Services ortamınızın planlanması.....	209
Advanced Message Securityplanlaması.....	210
Managed File Transferplanlaması.....	211
Planning to use the IBM MQ Console and REST API on z/OS.....	216
<b>Özel notlar.....</b>	<b>219</b>
Programlama arabirimi bilgileri.....	220
Ticari Markalar.....	220



# IBM MQ mimarisinin planlanması

IBM MQ ortamınızı planlarken, IBM MQ ' un tek ve birden çok kuyruk yöneticisi mimarisi ve noktadan noktaya iletişim ve yayınlama/abone olma ileti sistemi stilleri için sağladığı desteği göz önünde bulundurun. Ayrıca, kaynak gereksinimlerinizi ve günlük kaydı ve yedekleme olanaklarını kullanımınızı da planlayın.

## Bu görev hakkında

IBM MQ mimarinizi planlamadan önce, temel IBM MQ kavramlarını tanıyın. Bkz. [IBM MQ Teknik Genel Bakış](#).

IBM MQ mimarileri, tek bir kuyruk yöneticisi kullanan basit mimarilerden birbirine bağlı kuyruk yöneticilerinden oluşan daha karmaşık ağlara kadar değişir. Birden çok kuyruk yöneticisi, dağıtılmış kuyruğa alma teknikleri kullanılarak birbirine bağlanır. Tek kuyruk yöneticisini ve birden çok kuyruk yöneticisi mimarisini planlama hakkında daha fazla bilgi için aşağıdaki konulara bakın:

- [“Tek bir kuyruk yöneticisine dayalı mimariler” sayfa 17](#)
- [“Birden çok kuyruk yöneticisine dayalı mimariler” sayfa 18](#)
  - [“Dağıtılmış kuyruklarınızın ve kümelerinizin planlanması” sayfa 19](#)
  - [“Dağıtılmış yayınlama/abone olma ağınızın planlanması” sayfa 84](#)

**z/OS** IBM MQ for z/OS üzerinde, iş yükü dengelemeyi uygulamanızı ve IBM MQ uygulamalarınızın ölçeklenebilir ve yüksek kullanılabilirlikli olmasını sağlamak için paylaşılan kuyrukları ve kuyruk paylaşım gruplarını kullanabilirsiniz. Paylaşılan kuyruklar ve kuyruk paylaşım gruplarına ilişkin bilgi için [Paylaşılan kuyruklar ve kuyruk paylaşım grupları](#) başlıklı konuya bakın.

IBM MQ iki farklı yayın modeli sağlar:

- LTS (Long Term Support; Uzun Süreli Destek) yayın düzeyi, uzun vadeli devreye alma ve maksimum istikrar gerektiren sistemler için en uygundur
- Continuous Delivery (CD) yayın düzeyi, IBM MQ için en son işlevsel geliştirmelerden hızla yararlanması gereken sistemler için tasarlanmıştır.

Her iki yayın tipi de aynı şekilde kurulur, ancak anlamanız gereken destek ve geçişle ilgili önemli noktalar vardır. Daha fazla bilgi için bkz. [IBM MQ yayın tipleri ve sürüm oluşturma](#).

Birden çok kuruluş planlaması, depolama ve performans gereksinimleri ve istemcilerin kullanımı hakkında bilgi için diğer alt konulara bakın.

## İlgili kavramlar

[“z/OS üzerinde IBM MQ ortamınızın planlanması” sayfa 156](#)

IBM MQ ortamınızı planlarken, veri kümeleri, sayfa kümeleri, Db2, Coupling Faciliti'lerinin ve yedekleme olanaklarının ve yedekleme olanaklarının kaynak gereksinimlerini göz önünde bulundurmanız gerekir. Use this topic to plan the environment where IBM MQ runs.

[Kullanılabilirlik, kurtarma ve yeniden başlatma](#)

## İlgili görevler

[Gereksinimler denetleniyor](#)

[İletilerin kaybolmadığından emin olmak \(günlüğe kaydetme\)](#)

## IBM MQ yayın tipleri: planlama ile ilgili önemli noktalar

IBM MQ 9.0 ' den iki tip yayın düzeyi vardır; bir Long Term Support (LTS) yayın düzeyi ve bir Continuous Delivery (CD) yayın düzeyi. Desteklenen her platform için, seçtiğiniz yayın tipi sıralama, kuruluş, bakım ve geçiş işlemlerini etkiler.

Yayın tipleriyle ilgili ayrıntılı bilgi için bkz. [IBM MQ yayın tipleri ve sürüm oluşturma](#).

## IBM MQ for Multiplatforms ile ilgili dikkat edilecek noktalar

Multi

### Sıralama

Passport Advantage içinde IBM MQ 9.1 için iki ayrı eAssemblies vardır. Biri IBM MQ 9.1.0 Long Term Support yayın düzeyine ilişkin kuruluş görüntülerini, diğeri IBM MQ 9.1.x Continuous Delivery yayın düzeyine ilişkin kuruluş görüntülerini içerir. Yayın düzeyi seçiminize göre eAssembly ' den kuruluş görüntülerini yükleyin.

Alternatif olarak, yalnızca LTS yayın düzeyi için, ürün kuruluş DVD ' lerini içeren bir ortam paketi vardır.

Tüm IBM MQ sürümleri ve IBM MQ 9.1 hem LTS yayınları hem de CD yayınları için aynı Ürün Tanıtıcısına aittir.

IBM MQ kullanım yetkisi, lisanslı bileşenlerin ve fiyatlandırma ölçülerinin kısıtlamalarına tabi olarak ürünün tamamına yayılır. Bu, IBM MQ 9.1 için LTS yayın ve CD yayın kuruluş görüntüleri arasında özgürce seçim yapabileceğiniz anlamına gelir.

### Kuruluş

Passport Advantageolanağından bir kuruluş görüntüsü yükledikten sonra, kuruluş için yalnızca yetki satın aldığınız bileşenleri seçmeniz gerekir. Her bir ücretlendirilebilir bileşen için hangi kurulabilir bileşenlerin dahil edilmesiyle ilgili daha fazla bilgi için bkz. [IBM MQ lisans bilgileri](#) .

IBM MQ 9.1.0 LTS yayın düzeyini ve IBM MQ 9.1.x CD yayın düzeyini aynı işletim sistemi görüntüsüne kurabilirsiniz. Bunu yaparsanız, bileşenler, IBM MQ çoklu sürüm desteği tarafından desteklenen ayrı kuruluşlar olarak görünür. Her bir sürümün, o sürümle ilişkilendirilmiş ayrı kuyruk yöneticisi kümeleri vardır.

Her yeni CD yayını, kuruluş görüntüsü olarak sağlanır. Yeni CD yayın düzeyi, var olan bir yayın düzeyinin yanı sıra kurulabilir ya da kuruluş programı tarafından yerine daha önceki bir CD yayın düzeyi yeni yayın düzeyine güncellenebilir.

CD yayınları, en son hata düzeltmeleri ve güvenlik güncellemeleri kümesinin yanı sıra işlevsel geliştirmeler içerir. Her CD yayın düzeyi kümülatiftir ve bu IBM MQsürümünün tüm önceki yayın düzeylerinin yerine geçer. Bu nedenle, kuruluşunuzla ilgili herhangi bir işlev içermiyorsa belirli bir CD yayını atlayabilirsiniz.

### Bakım

LTS yayınına, hata düzeltmeleri sağlayan düzeltme paketlerinin uygulanması ve güvenlik yamaları sağlayan toplu güvenlik güncellemeleri (CSU) tarafından hizmet verilir. Düzeltme paketleri ve CSU ' lar belirli aralıklarla kullanıma sunulur ve kümülatiftir.

CD için, CSU ' lar yalnızca sonraki bir sürümde olabilecek en son CD yayın düzeyi için üretilir.

Ara düzeltme uygulamak için IBM Destek ekibi tarafından bazen yönlendirilebilirsiniz. Ara düzeltmeler, acil durum ya da test düzeltmeleri olarak da bilinir ve sonraki bakım teslimatı için bekleyemeyecek acil güncellemeleri uygulamak için kullanılır.

### LTS yayın düzeyi ve CD yayın düzeyi arasında geçiş

Kısıtlamalar ve sınırlamalar vardır, ancak genellikle tek bir kuyruk yöneticisi LTS yayın kodunu CD yayın koduna ya da hedef yayın düzeyinin geçişten önce kullanılanlardan daha yüksek olması koşuluyla, CD yayın kodunu LTS yayın koduna kullanmaktan geçirebilir.

İki yaklaşım mümkündür:

- Var olan bir IBM MQ kuruluşunun güncellenmesi için yeni kod yayın düzeyini kurun. Kuruluşla ilişkili kuyruk yöneticileri, başlatıldığında kodun yeni yayın düzeyini kullanır.
- Yeni kod yayını yeni bir kuruluş olarak kurun ve [setmqm](#) komutunu kullanarak tek tek kuyruk yöneticisi yönetim ortamlarını yeni kuruluşa taşıyın.

Bir kuyruk yöneticisi kodun CD yayın düzeyini çalıştırmaya başladığında, kuyruk yöneticisi komut düzeyi yeni yayın düzeyini gösterecek şekilde güncellenir. Bu, yayın düzeyinde sağlanan yeni işlevlerin etkinleştirildiği anlamına gelir ve daha düşük bir VRM numarasıyla bir kod yayını kullanarak kuyruk yöneticisini artık yeniden başlatamayamazsınız.

## IBM MQ for z/OS ile ilgili dikkat edilecek noktalar



### Sıralama

z/OS için IBM MQ 9.1 siparişi verirken, ShopZ' de iki ayrı aksam sunulur. Özellikler, LTS yayın düzeyine ve CD yayın düzeyine karşılık gelir. Her iki özellik de aynı ürün tanıtıcısı (PID) için geçerlidir. Bu, lisanslanan ürün tanıtıcısıdır, bu nedenle bir aksam lisanslandığında, gerekirse alternatif aksamı kullanma yetkisi vardır. Sipariş verirken, LTS yayın düzeyi ya da CD yayın düzeyine karşılık gelen özelliği seçin:

- IBM MQ for z/OS 9.1
- Managed File Transfer for z/OS 9.1

Ürünleri ServerPac' e eklemek üzere seçtiğinizde, ürünler SMP/E tarafından aynı hedef bölgeye kurulamayacağı için, aynı ServerPac siparişindeki hem LTS yayın düzeyini hem de CD yayın düzeyini seçemezsiniz.

Advanced Message Security for z/OS (AMS) 9.1 , LTS yayın düzeyi ve CD yayın düzeyi seçeneklerini sağlar.

Yayın seçenekleri, temel ürün için eşdeğer yayın düzeyi ( IBM MQ for z/OS 9.1) seçilerek ve 5655-AM9ürünüyle birlikte etkinleştirilerek belirlenir.



**Uyarı:** Bir LTS yayın IBM MQ tabanında AMS işlevleri için CD yayın geliştirmeleri seçemezsiniz.

İşletmeniz IBM MQ for z/OS Value Unit Edition (VUE) 9.1 kullanıyorsa, aynı ürün 5655-VU9 , IBM MQ for z/OS 9.1 ürününün LTS yayın düzeyinde ya da CD yayın düzeyinde VUE lisansını etkinleştirir.

### Kuruluş

LTS ve CD yayınları ayrı FMID kümelerinde sağlanır. Bu FMID ' lerin aynı SMP/E hedef bölgesine kurulamayacağını unutmayın. Hem LTS hem de CD yayın düzeylerine gereksinim duyarsanız:

- LTS yayın düzeyini ve CD yayın düzeyini ayrı hedef bölgelere kurma
- İki yayın düzeyi için ayrı hedef ve dağıtım kitaplıkları sağlayın

Kuyruk yöneticiniz bir kuyruk paylaşım grubundaydı, en son CD sürümüne yükseldiğinizde, gruptaki tüm kuyruk yöneticilerini yükseltmeniz gerekir.

Kuyruk yöneticisinin komut düzeyi üç basamaklı VRM düzeyidir. IBM MQ programı, bağlı olduğu kuyruk yöneticisinin komut düzeyini almak için MQIA\_COMMAND\_LEVEL seçicisini geçirerek MQINÇağırabilir.

Yayınlar farklı FMID ' ler kullandığından, bir CD yayın düzeyini LTS yayın düzeyine ilişkin bir bakım ile güncelleyemezsiniz ya da diğer bir şekilde yuvarlayamazsınız. Benzer şekilde, ürün kodunun bir sürümünü LTS yayın düzeyinden CD yayın düzeyine ya da tersi yönde değiştirmenin bir yolu yoktur. Ancak, bir kuyruk yöneticisini serbest bırakma modelleri arasında geçiş yapabilirsiniz. Bkz. [Migration between LTS release and CD release](#).

Kuyruk yöneticisi iş günlüğündeki [CSQY000I](#) iletisine bakarak aynı VRM düzeyine sahip bir LTS ve CD yayını ayırt edebilirsiniz.

### Bakım

IBM MQ for z/OS , Bakım için PTF ' leri kullanır.



PTF ' ler belirli bir yayın düzeyine karşılık gelen belirli bir kitaplık kümesine özgüdür. UNIX System Services özellikleri ( JMS ve WEB UI, Connector Pack ve Managed File Transfer) için z/OS PTF 'ler doğrudan Multiplatforms düzeltme paketleriyle ve birikmeli güvenlik güncellemeleriyle (CSU' lar)hızalanır. Bu düzeltmeler kümülatiftir ve eşdeğer Multiplatforms düzeltme paketiyle ya da CSUayı anda kullanılabilir.

**CD** CD CSU ' lar genellikle CD yayınları arasında kullanılamaz, ancak sonraki IBM MQ for z/OS CD yayınında bulunur. ++ USERMOD istemek için destek ile de iletişim kurabilirsiniz.

IBM MQ for z/OS üzerindeki diğer düzeltmeler, belirli bölümlere ilişkin ayrı düzeltmelerdir. Bu düzeltmeler, belirli sorunları çözebilir, kümülatif değildir ve bunlar üretildikleri zaman ve oluşturuldukları şekilde kullanıma sunulur.

### LTS yayın düzeyi ve CD yayın düzeyi arasında geçiş

Kısıtlamalar ve sınırlamalar vardır, ancak hedef yayın düzeyinin geçişten önce kullanılan daha yüksek olması koşuluyla, genellikle tek bir kuyruk yöneticisinin LTS yayın kodu kullanılarak CD yayın koduna ya da CD yayın kodu kullanılarak LTS yayın koduna geçişi yapılabilir.

IBM MQ for z/OS geleneksel olarak bir geri dönüş yeteneği (geriye doğru geçiş) sağlamıştır; böylece, bir geçişin ardından bir süre çalıştıktan sonra önceki yayına geri dönebilirsiniz. Bu yetenek LTS yayınları için korunur, ancak bir geçişin kaynağı ya da hedefi bir CD yayın düzeyi olduğunda mümkün değildir.

Aşağıda geçerli geçiş senaryoları yer almaktadır ve bu prensibin nasıl çalıştığını göstermektedir:

### V 9.1.0

Kaynak yayın düzeyi	Hedef yayın düzeyi	Notlar
V8.0.0	V9.1.0 LTS	Geriye doğru geçiş her zaman desteklenir, V8.0.0 sürümünde uygulanan geriye doğru geçiş PTF ' leri sağlanır.
V9.0.0	V9.1.0 LTS	Geriye doğru geçiş her zaman desteklenir, V9.0.0'da uygulanan geriye doğru geçiş PTF' leri sağlanmıştır.
V9.0.5 CD 'si	V9.1.0 LTS	Geriye doğru geçiş yok, kaynak yayın düzeyi CD.
<b>V 9.1.1</b> V 9.1.1 V9.0.0	V9.1.1 CD 'si	Geriye doğru geçiş yok, hedef yayın CD. Geçiş onaylamak için Write to operator with reply <a href="#">CSQY041D</a> yayınlandı.

### İlgili görevler

[z/OS üzerinde bakım uygulanması ve kaldırılması](#)

### İlgili bilgiler

[Aşağı yükleniyor IBM MQ 9.1](#)

## IBM MQ ve IBM MQ Appliance şirket içinde GDPR ' ye hazır olma ile ilgili önemli noktalar

### PID ' ler için:

- 5724-H72 IBM MQ
- 5655-AV9 IBM MQ Advanced for z/OS
- 5655-AV1 IBM MQ Advanced for z/OS Value Unit Edition
- 5655-AM9 IBM MQ Advanced Message Security for z/OS
- 5725-S14 IBM MQ Appliance M2000
- 5725-Z09 IBM MQ Appliance M2001



- 5737-H47 IBM MQ Appliance M2002
- 5655-MQ9 IBM MQ for z/OS
- 5655-VU9 IBM MQ for z/OS Value Unit Edition
- 5655-MF9 IBM MQ Managed File Transfer for z/OS
- 5655-ADV IBM WebSphere MQ Advanced for z/OS
- 5655-AMS IBM WebSphere MQ Advanced Message Security for z/OS
- 5724-A39 IBM WebSphere MQ for HP NonStop Server
- 5655-W97 IBM WebSphere MQ for z/OS
- 5655-VU8 IBM WebSphere MQ for z/OS Value Unit Edition
- 5655-VUE IBM WebSphere MQ for z/OS Value Unit Edition
- 5655-MFT IBM WebSphere MQ Managed File Transfer for z/OS

## **Bildirim:**

Bu belgenin amacı, GDPR için hazır olma çalışmalarınıza yardımcı olmaktır. GDPR'ye hazır olma konusunda kuruluşunuza yardımcı olmak için yapılandırabileceğiniz IBM MQ özellikleri hakkında bilgiler ve ürünün kullanımıyla ilgili dikkat etmeniz gereken noktalar sağlar. Müşterilerin özellikleri seçip yapılandırabileceği ve ürünün kendi içinde ya da üçüncü taraf uygulamaları ve sistemleri ile kullanılabileceği birçok yol olduğundan, bu bilgiler çok kapsamlı değildir.

**Müşteriler, Avrupa Birliği Genel Veri Koruma Yönetmeliği de dahil olmak üzere çeşitli yasalara ve yasal düzenlemelere uyumluluklarını sağlamak için kendileri sorumludur. Müşteriler kendi işlerini ve söz konusu yasalara ve yasal düzenlemelere uymak için yapmaları gereken işlemleri etkileyebilecek olan ilgili yasa ve yasal düzenlemelerin belirlenmesi ve yorumlanması ile ilgili olarak yetkili bir hukuk danışmanının önerisini almaktan tek başına sorumludur.**

**Ürünler, hizmetler ve burada açıklanan diğer yeterlilikler, tüm müşteri durumları için uygun olmayabilir ve bunların sağlanabilirliği kısıtlı olabilir. IBM , yasal, muhasebeyle ilgili ya da denetimle ilgili öneriler sağlamaz ya da hizmetlerinin ya da ürünlerinin, müşterilerin herhangi bir yasaya ya da düzenlemeye uymalarını sağlayacağını beyan ya da garanti etmez.**

## **İçindekiler**

1. [GDPR/GVKY](#)
2. [GDPR için Ürün Yapılandırması](#)
3. [Veri Yaşam Çevrimi](#)
4. [Veri Toplama](#)
5. [Veri Depolama](#)
6. [Veri Erişimi](#)
7. [Verilerin İşlenmesi](#)
8. [Verilerin Silinmesi](#)
9. [Verilerin İzlenmesi](#)
10. [Kişisel Verilerin Kullanımını kısıtlama yeteneği](#)
11. [Dosya işleme](#)

## **GDPR/GVKY**

Genel Veri Koruma Yönetmeliği (GDPR; General Data Protection Regulation), Avrupa Birliği ve Avrupa Birliği ("AB") tarafından kabul edilmiş olup 25 Mayıs 2018 tarihinden itibaren yürürlüğe girmiştir.

### **GDPR neden önemli?**

GDPR, bireylerin kişisel verilerinin işlenmesine yönelik daha güçlü bir yasal veri koruma çerçevesi oluşturur. GDPR aşağıdakileri sunar:

- Bireyler için yeni ve geliştirilmiş haklar
- Kişisel verilerin daha geniş bir tanımı
- Veri işleyenler için yeni yükümlülükler
- İhlal halinde ciddi finansal ceza uygulama potansiyeli
- Zorunlu veri ihlali bildirim

#### **GDPR hakkında daha fazla bilgi edinin:**

- [AB GDPR bilgi portalı](#)
- [ibm.com/GDPR](http://ibm.com/GDPR) web sitesi

### **Ürün Yapılandırması - GDPR'ye hazırlıkla ilgili önemli noktalar**

Aşağıdaki bölümlerde, kuruluşunuza GDPR 'ye hazır olma konusunda yardımcı olması için IBM MQ ' in yapılandırılmasına ilişkin önemli noktalar sağlanır.

#### **Veri Yaşam Çevrimi**

IBM MQ , uygulamaların uygulama tarafından sağlanan verileri zamanuyumsuz olarak değiş tokuş etmesini sağlayan, işlemsel ileti odaklı bir ara katman yazılımı ürünüdür. IBM MQ , uygulamaların bağlanması amacıyla çeşitli ileti sistemi API ' lerini, iletişim kurallarını ve köprülerini destekler. Bu nedenle, IBM MQ , bazıları GDPR ' ye tabi olabilecek birçok veri biçimini değiştirmek için kullanılabilir. IBM MQ ' in veri alışverişinde bulunabileceği birkaç üçüncü kişi ürünü vardır. Bunlardan bazıları IBM' e aittir, ancak diğerleri diğer teknoloji tedarikçileri tarafından sağlanır. [Software Product Compatibility Reports](#) (Yazılım Ürünü Uyumluluk Raporları) web sitesi, ilişkili yazılımların listelerini sağlar. Bir üçüncü kişi ürününün GDPR ' ye hazır olup olmamasına ilişkin dikkat edilmesi gereken noktalar için bu ürünün belgelerine bakmanız gerekir. IBM MQ yöneticileri, IBM MQ ' in veri geçişiyle nasıl etkileşimde bulunacağı, kuyrukların, konuların ve aboneliklerin tanımına göre denetler.

#### **IBM MQ aracılığıyla hangi veri akışı tipleri?**

IBM MQ uygulama verileri için zamanuyumsuz ileti sistemi hizmeti sağladığından, kullanım senaryoları uygulama devreye alınmasına göre değiştiğinden bu soruya kesin bir yanıt yoktur. Uygulama iletileri verileri kuyruk dosyalarında ( z/OS üzerinde sayfa kümeleri ya da Coupling Facility), günlüklerde ve arşivlerde saklanır ve iletinin kendisi GDPR tarafından yönetilen veriler içerebilir. Uygulama tarafından sağlanan ileti verileri, hata günlükleri, izleme dosyaları ve FFSTler gibi sorun belirleme amacıyla toplanan dosyalara dahil edilebilir. z/OS uygulaması için sağlanan ileti verileri adres alanına ya da Coupling Facility dökümlerine de eklenebilir.

Aşağıda, IBM MQ kullanılarak değiş tokuş edilebilecek bazı tipik kişisel veri örnekleri verilmiştir:

- Müşterinin çalışanları (örneğin; IBM MQ , müşterinin bordro ya da İK sistemleri arasında bağlantı kurmak için kullanılabilir)
- Müşterinin kendi müşterilerinin kişisel verileri (örneğin; IBM MQ , bir müşteri tarafından, satış fırsatlarını almak ve Müşteri İlişkileri Yönetimi sistemi içinde verileri depolamak gibi, müşterileriyle ilgili uygulamalar arasında veri alışverişi yapmak için kullanılabilir).
- Müşterinin kendi müşterilerinin hassas kişisel verileri (örneğin; IBM MQ , klinik uygulamalarını bütünleştirirken HL7-based sağlık hizmetleri kayıtları gibi kişisel verilerin değiş tokuş alınmasını gerektiren sektör bağlamlarında kullanılabilir).

IBM MQ , uygulama tarafından sağlanan ileti verilerine ek olarak aşağıdaki veri tiplerini de işler:

- Kimlik Doğrulama Kimlik Bilgileri (kullanıcı adı ve parolalar, API anahtarları vb.)
- Teknik Olarak Tanımlanmış Kişisel Bilgiler (örneğin, aygıt tanıtıcıları, kullanıma dayalı tanıtıcılar, IP adresi, vb.) -bir kişiye bağlandığında)

#### **IBM ile çevrimiçi iletişim için kullanılan kişisel veriler**

IBM MQ müşterileri, öncelikle aşağıdakiler olmak üzere çeşitli yollarla IBM MQ konularına ilişkin çevrimiçi yorumlar/geri beslemeler/istekler gönderebilirler:

- [IBM MQ alanı IBM Developer](#) içindeki sayfalarda genel yorumlar alanı
- [IBM MQ ürün bilgileri IBM Documentation](#) sayfalarındaki genel yorumlar alanı
- [IBM Destek Forumları](#) içindeki genel yorumlar
- [IBM Integration Ideas](#) içindeki genel yorumlar

Genellikle iletişim konusu için kişisel yanıtların etkinleştirilmesi amacıyla yalnızca müşteri adı ve e-posta adresi kullanılır ve kişisel veri kullanımı [IBM Çevrimiçi Gizlilik Bildirimi](#)'ne uygun olur.

## Veri Toplama

IBM MQ kişisel verileri toplamak için kullanılabilir. IBM MQ kullanımınızı değerlendirirken ve GDPR taleplerini karşılama gereksinimlerinizi değerlendirirken, IBM MQ'ından geçen kişisel veri türlerini göz önünde bulundurmanız gerekir. Aşağıdakiler gibi yönleri göz önünde bulundurmak isteyebilirsiniz:

- Veriler kuyruk yöneticilerinize nasıl ulaşır? (Hangi protokollerde? Veriler şifreli mi? Veriler imzalandı mı?)
- Veriler kuyruk yöneticilerinizden nasıl gönderilir? (Hangi protokollerde? Veriler şifreli mi? Veriler imzalandı mı?)
- Bir kuyruk yöneticisinden geçerken veriler nasıl depolanır? (Herhangi bir ileti sistemi uygulaması, bir ileti kalıcı olmasa da, ileti verilerini durumlu ortama yazma potansiyeline sahiptir. İleti sistemi özelliklerinin, üründen geçen uygulama iletileri verilerinin çeşitli yönlerini nasıl ortaya çıkarabileceğini biliyor musunuz?)
- IBM MQ 'in üçüncü kişi uygulamalarına erişmek için gereksinim duyduğu yerde kimlik bilgileri nasıl toplanır ve depolanır?

IBM MQ 'in kimlik doğrulaması gerektiren diğer sistemler ve hizmetlerle (örneğin, LDAP) iletişim kurması gerekebilir. Gerekliğinde, kimlik doğrulama verileri (kullanıcı kimlikleri, parolalar) bu tür iletişimlerde kullanılmak üzere IBM MQ tarafından yapılandırılır ve depolanır. Mümkün olduğunda, IBM MQ kimlik doğrulaması için kişisel kimlik bilgilerini kullanmaktan kaçınmalısınız. Kimlik doğrulama verileri için kullanılan depolamanın korunmasını göz önünde bulundurun. (Aşağıdaki Veri Depolama bölümüne bakın.)

## Veri Depolama

İleti verileri kuyruk yöneticileri arasında dolaşırken, IBM MQ bu verileri doğrudan durumlu ortama aktarmaya devam eder (belki birden çok kopya). IBM MQ kullanıcıları, dururken ileti verilerinin güvenliğini sağlamak isteyebilir.

Aşağıdaki öğeler, IBM MQ 'in uygulama tarafından sağlanan ve kullanıcıların GDPR'ye uygunluğu sağlarken göz önünde bulundurmak isteyecekleri verileri belirttiği alanları vurgular.

- Uygulama İleti Kuyrukları:

IBM MQ , uygulamalar arasında zamanuyumsuz veri alışverişine izin vermek için ileti kuyrukları sağlar. Bir kuyrukta saklanan kalıcı olmayan ve kalıcı olmayan iletiler, durumlu ortama yazılır.

- Dosya Aktarma Aracısı Kuyrukları:

IBM MQ Managed File Transfer , dosya verilerinin güvenilir aktarımını koordine etmek için ileti kuyruklarını kullanır; kişisel verileri içeren dosyalar ve aktarma kayıtları bu kuyruklarda saklanır.

- İletim Kuyrukları:

İletileri kuyruk yöneticileri arasında güvenilir bir şekilde aktarmak için, iletiler iletim kuyruklarında geçici olarak saklanır.

- Gönderilmeyen Mektup Kuyrukları:

Kuyruk yöneticisinde böyle bir kuyruk yapılandırıldıysa, iletilerin hedef kuyruğa konamayacağı ve gitmeyen iletiler kuyruğunda saklanamayacağı bazı durumlar vardır.

- Geriletme Kuyrukları:

JMS ve XMS ileti sistemi arabirimleri, diğer geçerli iletilerin işlenmesine izin vermek için bir dizi geriletme gerçekleştirildikten sonra zehirli iletilerin bir geriletme kuyruğuna taşınmasına olanak sağlayan bir yetenek sağlar.

- **AMS Hata Kuyruğu:**

IBM MQ Advanced Message Security , bir güvenlik ilkesine uymayan iletileri SYSTEM.PROTECTION.ERROR.QUEUE hata kuyruğu, hatalı kuyruğa alma yöntemine benzer bir şekilde.

- **Alıkonan Yayınlar:**

IBM MQ , abone olan uygulamaların önceki bir yayını geri çağırmasına izin vermek için korunan bir yayın özelliği sağlar.

- **Ertelenen Teslimat:**

IBM MQ , iletilerin gelecekteki bir zamanda hedeflerine teslim edilmesini sağlayan JMS 2.0 teslim gecikmesi özelliğini destekler. Henüz teslim edilmemiş iletiler, SYSTEM.DDELAY.LOCAL.QUEUE (KUYRUK).

Daha fazla bilgi:

- [Günlüğe kaydetme: İletilerin kaybolmadığından emin olun](#)
- [MFT Agent kuyruk ayarları](#)
- [İleti gönderilmeyen iletiler kuyruğunun kullanılması](#)
- [JMS için IBM MQ sınıflarında zehirli iletilerin işlenmesi](#)
- [AMS hata işleme](#)
- [Alıkonan yayınlar](#)
- [JMS 2.0 teslim gecikmesi](#)

Aşağıdaki öğeler, IBM MQ 'in kullanıcıların GDPR' ye uygunluğu sağlarken dikkate almak isteyecekleri, uygulama tarafından sağlanan verileri dolaylı olarak sürdürebileceği alanları vurgular.

- **İzleme rotası ileti alışverişi:**

IBM MQ , bir iletinin uygulamalar arasında izlediği rotayı kaydeden izleme rotası yetenekleri sağlar. Oluşturulan olay iletileri, IP adresleri gibi teknik olarak tanımlanabilir kişisel bilgileri içerebilir.

- **Uygulama etkinliği izlemesi:**

IBM MQ , uygulamaların ve kanalların ileti sistemi API etkinliklerini kaydeden uygulama etkinliği izlemesini sağlar; uygulama etkinliği izlemesi, uygulama tarafından sağlanan ileti verilerinin içeriğini olay iletilerine kaydedebilir.

- **Hizmet izleme:**

IBM MQ , ileti verilerinin aktığı iç kod yollarını kaydeden hizmet izleme özellikleri sağlar. Bu özelliklerin bir parçası olarak IBM MQ , uygulama tarafından sağlanan ileti verilerinin içeriğini diskte saklanan izleme dosyalarına kaydedebilir.

- **Kuyruk yöneticisi olayları:**

IBM MQ , yetki, komut ve yapılandırma olayları gibi kişisel verileri içerebilecek olay iletileri oluşturabilir.

Daha fazla bilgi:

- [İzleme rotası ileti alışverişi](#)
- [İzleme özelliğinin kullanılması](#)
- [Olay izleme](#)
- [Kuyruk yöneticisi olayları](#)

Uygulamanın sağladığı ileti verilerinin kopyalarına erişimi korumak için aşağıdaki eylemleri göz önünde bulundurun:

- Ayrıcalıklı kullanıcı erişimini dosya sistemindeki IBM MQ verileriyle sınırlandırın; örneğin, UNIX platformlarında 'mqm' grubunun kullanıcı üyeliğini kısıtlayın.

- Özel olarak ayrılmış kuyruklar ve erişim denetimi aracılığıyla IBM MQ verilerine uygulama erişimini kısıtlayın. Uygun olduğunda, kuyruklar gibi kaynakların uygulamalar arasında gereksiz yere paylaşılmasından kaçınin ve kuyruk ve konu kaynakları için ayrıntılı erişim denetimi sağlayın.
- Yüksek kullanılabilirlikli (HA) ya da olağanüstü durumdan kurtarma (DR) yapılandırmalarında IBM MQ verilerinin eşlenmiş kopyalarına erişimi kısıtlayın ve eşleme için kullanılan bağlantıların güvenliğini sağlayın.
- İleti verilerinin uçtan uca imzalanmasını ve/veya şifrenmesini sağlamak için IBM MQ Advanced Message Security olanağını kullanın.
- İzleme günlüklerini saklamak için kullanılan dizinin içeriğini korumak için dosya ya da birim düzeyinde şifreleme kullanın.
- Hizmet izlemesini IBM' e yükledikten sonra, kişisel verileri içerebilecek içerikle ilgili endişeleriniz varsa, hizmet izleme dosyalarınızı ve FFST verilerinizi silebilirsiniz.

Daha fazla bilgi:

- [Ayrıcalıklı kullanıcılar](#)
- [Multiplatforms üzerinde planlama dosya sistemi desteği](#)

IBM MQ yöneticisi, kimlik bilgileriyle (kullanıcı adı ve parola, API anahtarları, vb.) bir kuyruk yöneticisini yapılandırabilir LDAP, Salesforce gibi 3rd kişi hizmetleri için Bu veriler genellikle, dosya sistemi izinleriyle korunan kuyruk yöneticisi veri dizininde saklanır.

Bir IBM MQ kuyruk yöneticisi yaratıldığında, veri dizini, IBM MQ ' un yapılandırma dosyalarını okuyabileceği ve bu sistemlere bağlanmak için kimlik bilgilerini kullanabileceği grup tabanlı erişim denetimiyle ayarlanır. IBM MQ yöneticileri ayrıcalıklı kullanıcılar olarak kabul edilir ve bu grubun üyeleri olduğu için dosyalara okuma erişimine sahip olur. Bazı dosyalar gizlenmiş, ancak şifrenmemiş. Bu nedenle, kimlik bilgilerine erişimi tam olarak korumak için aşağıdaki işlemleri göz önünde bulundurmanız gerekir:

- Ayrıcalıklı kullanıcı erişimini IBM MQ verileriyle sınırlandırın; örneğin, UNIX platformlarında 'mqm' grubunun üyeliğini kısıtlayın.
- Kuyruk yöneticisi veri dizininin içeriğini korumak için dosya ya da birim düzeyinde şifreleme kullanın.
- Üretim yapılandırma dizininin yedeklerini şifreleyin ve bunları uygun erişim denetimleriyle saklayın.
- Güvenlik, komut ve yapılandırma olaylarıyla kimlik doğrulama hatası, erişim denetimi ve yapılandırma değişiklikleri için denetim izleri sağlamayı düşünün.

Daha fazla bilgi:

- [IBM MQ' nun Güvenliğinin Sağlama](#)

## Veri Erişimi

IBM MQ kuyruk yöneticisi verilerine, bazıları uzak bağlantı üzerinden erişim için, bazıları da yerel bağlantı üzerinden erişim için tasarlanmış olan aşağıdaki ürün arabirimleri aracılığıyla erişilebilir.

- IBM MQ Konsol [ Yalnızca Uzak]
- IBM MQ Yönetim REST API 'si [ Yalnızca Uzak]
- IBM MQ Messaging REST API [ Yalnızca Uzak]
- MQI [ Yerel ve Uzak]
- JMS [ Yerel ve Uzak]
- XMS [ Yerel ve Uzak]
- IBM MQ Telemetry (MQTT) [ Yalnızca Uzak]
- IBM MQ Light (AMQP) [ Yalnızca Uzak]
- IBM MQ IMS köprüsü [ Yalnızca Yerel]
- IBM MQ CICS köprüsü [ Yalnızca Yerel]
- IBM MQ MFT İletişim Kuralı köprüleri [ Yalnızca Uzak]

- IBM MQ Connect:Direct köprüleri [ Yalnızca Uzak]
- IBM MQ Köprüden Salesforce ' a [ Yalnızca Uzak]
- IBM MQ Köprüden Blockchain [ Yalnızca Uzak]
- IBM MQ MQAI [ Yerel ve Uzak]
- IBM MQ PCF komutları [ Yerel ve Uzak]
- IBM MQ MQSC komutları [ Yerel ve Uzak]
- IBM MQ Explorer [ Yerel ve Uzak]
- IBM MQ Kullanıcı Çıkışları [ Yalnızca Yerel]
- IBM MQ Internet Pass-Thru [ Yalnızca Uzak]
- IBM MQ Appliance Dizisel Konsol [ Yalnızca Yerel]
- IBM MQ Appliance SSH [ Yalnızca Uzak]
- IBM MQ Appliance REST API [ Yalnızca Uzak]
- IBM MQ Appliance Web UI [ Yalnızca Uzak]

Arabirimler, kullanıcıların bir IBM MQ kuyruk yöneticisinde ve üzerinde saklanan iletilerde değişiklik yapmalarını sağlamak üzere tasarlanmıştır. Yönetim ve ileti sistemi işlemleri, bir istek yapıldığında üç aşamaya dahil olmak üzere güvence altına alınır;

- Kimlik Doğrulaması
- Rol eşleme
- Yetkilendirme

#### **Kimlik denetimi:**

İleti ya da denetim işlemi yerel bir bağlantıdan istendiye, bu bağlantının kaynağı aynı sistemde çalışan bir işlemdir. İşlemi çalıştıran kullanıcı, işletim sistemi tarafından sağlanan kimlik doğrulama adımlarını geçmiş olmalıdır. Bağlantının yapıldığı sürecin sahibinin kullanıcı adı kimlik olarak bildirilir. Örneğin, bu, bir uygulamanın başlatıldığı kabuğu çalıştıran kullanıcının adı olabilir. Yerel bağlantılar için olası kimlik doğrulama biçimleri şunlardır:

1. Bildirili kullanıcı adı (yerel işletim sistemi)
2. İsteğe bağlı kullanıcı adı ve parola (OS, LDAP ya da özel 3rd taraf havuzları)

Denetim işlemi uzak bir bağlantıdan istendiye, IBM MQ ile iletişim bir ağ arabirimi üzerinden kurulur. Ağ bağlantıları aracılığıyla kimlik doğrulaması için aşağıdaki kimlik formları sunulabilir;

1. Bildiri kullanıcı adı (uzak işletim sisteminden)
2. Kullanıcı adı ve parola (OS, LDAP ya da özel 3rd taraf havuzları)
3. Kaynak ağ adresi (IP adresi gibi)
4. X.509 Sayısal Sertifika (karşılıklı SSL/TLS kimlik doğrulaması)
5. Güvenlik belirteçleri ( LTPA2 simgesi gibi)
6. Diğer özel güvenlik ( 3rd kişi çıkışları tarafından sağlanan yetenek)
7. SSH anahtarları

#### **Rol eşleme:**

Rol eşleme aşamasında, kimlik doğrulama aşamasında sağlanan kimlik bilgileri diğer bir kullanıcı kimliğiyle eşlenebilir. Eşlenen kullanıcı kimliğinin devam etmesine izin verildiği sürece (örneğin, yönetimle görevli kullanıcılar kanal kimlik doğrulama kuralları tarafından engellenebilir), eşlenen kullanıcı kimliği, IBM MQ kaynaklarına ilişkin etkinlikler yetkilendirilirken son aşamaya taşınır.

#### **Yetki:**

IBM MQ , farklı kullanıcıların kuyruklar, konular ve diğer kuyruk yöneticisi nesnelere gibi farklı ileti kaynakları için farklı yetkilere sahip olmalarını sağlar.

#### **Günlüğe kaydetme etkinliği:**

Bazı IBM MQ kullanıcılarının MQ kaynaklarına erişim için bir denetim kaydı oluşturmaları gerekebilir. İstenen denetim günlüklerine örnek olarak, değişikliği kimin istediğine ek olarak değişikliğe ilişkin bilgiler de içeren yapılandırma değişiklikleri gösterilebilir.

Bu gereksinimi gerçekleştirmek için aşağıdaki bilgi kaynakları kullanılabilir:

1. Bir yönetici komutu başarıyla çalıştırıldığında, IBM MQ kuyruk yöneticisi komut olayları üretecek şekilde yapılandırılabilir.
2. Bir IBM MQ kuyruk yöneticisi, bir kuyruk yöneticisi kaynağı yaratıldığında, değiştirildiğinde ya da silindiğinde yapılandırma olayları üretecek şekilde yapılandırılabilir.
3. IBM MQ kuyruk yöneticisi, bir kaynak için yetki denetimi başarısız olduğunda bir yetki olayı üretecek şekilde yapılandırılabilir.
4. Başarısız yetki denetimlerinin kuyruk yöneticisi hata günlüklerine yazıldığını gösteren hata iletileri.
5. IBM MQ Console, kimlik doğrulama, yetkilendirme denetimleri başarısız olduğunda ya da kuyruk yöneticileri oluşturulduğunda, başlatıldığında, durdurulduğunda ya da silindiğinde denetleme iletilerini günlüklerine yazar.
6. IBM MQ Appliance , kullanıcı oturum açmaları ve sistem değişikliklerini kaydetmek için denetleme iletilerini günlüklerine yazar.

IBM MQ kullanıcıları, bu tür çözümleri göz önünde bulundurarak aşağıdaki noktaları dikkate almak isteyebilir:

- Olay iletileri kalıcı değildir; bu nedenle, kuyruk yöneticisi bilgileri yeniden başlattığında bilgiler kaybolur. Herhangi bir olay izleyicisi, var olan iletileri sürekli olarak kullanacak ve içeriği kalıcı ortama aktarabilecek şekilde yapılandırılmalıdır.
- IBM MQ ayrıcalıklı kullanıcılarının devre dışı bırakılan olaylar için yeterli ayrıcalıkları vardır, günlükleri temizleyin ya da kuyruk yöneticilerini silin.

IBM MQ verilerine erişimin güvenliğini sağlama ve bir denetim izi sağlama hakkında daha fazla bilgi için aşağıdaki konulara bakın:

- [IBM MQ güvenlik mekanizmaları](#)
- [Yapılandırma olayları](#)
- [Komut olayları](#)
- [Hata Günlükleri](#)

## Verilerin İşlenmesi

### Genel Anahtar Altyapısı kullanılarak şifreleme:

Bağlantıların TLS kullandığını belirterek IBM MQ ile ağ bağlantılarının güvenliğini sağlayabilirsiniz; bu, bağlantının başlangıç tarafının karşılıklı kimlik doğrulamasını da sağlayabilir.

İletim mekanizmaları tarafından sağlanan PKI güvenlik olanaklarının kullanılması, IBM MQ ile veri işleminin güvenli kılınmasına yönelik ilk adımdır. Ancak, daha fazla güvenlik özelliği etkinleştirilmeden, tüketen bir uygulamanın davranışı, iletinin kökenini doğrulamadan veya geçiş sırasında değiştirilip değiştirilmediğini doğrulamadan ona teslim edilen tüm iletileri işlemektir.

Advanced Message Security (AMS) yeteneklerini kullanmak üzere lisanslanan IBM MQ kullanıcıları, güvenlik ilkelerinin tanımlanması ve yapılandırılması aracılığıyla uygulamaların iletilerde tutulan kişisel verileri nasıl işleyeceklerini denetleyebilirler. Güvenlik ilkeleri, uygulamalar arasında ileti verilerine dijital imzalama ve/veya şifreleme uygulanmasını sağlar.

İletilerin gerçek olduğundan emin olmak için iletileri kullanırken dijital imzayı zorunlu tutmak ve doğrulamak için güvenlik ilkeleri kullanılabilir. AMS şifrelemesi, ileti verilerinin okunabilir bir formdan kodlanmış bir sürüme dönüştürülmesi için bir yöntem sağlar; bu yöntem, yalnızca amaçlanan alıcı ya da iletise ve doğru şifre çözme anahtarına erişimi varsa başka bir uygulama tarafından çözülebilir.

Ağ bağlantılarınızın güvenliğini sağlamak için SSL ve sertifikaların kullanılmasıyla ilgili ek bilgi için IBM MQ ürün belgelerinde aşağıdaki konulara bakın:

- [IBM MQ](#)
- [AMS ' ye Genel Bakış](#)

## Verilerin Silinmesi

IBM MQ , ürüne sağlanan verileri silmek için komutlar ve kullanıcı arabirimi işlemleri sağlar. Bu, IBM MQ kullanıcılarının belirli kişilerle ilgili verileri silebileceği anlamına gelir (bu gerekiyorsa).

- GDPR Müşteri Verilerini silmeye uymak için göz önünde bulundurulması gereken IBM MQ davranış alanları
  - Uygulama kuyruğunda saklanan ileti verilerini silmek için:
    - İleti alışverişi API 'sini ya da araçlarını kullanarak ya da ileti süre bitimini kullanarak tek tek iletileri kaldırma.
    - İletilerin kalıcı olmadığını, kalıcı olmayan ileti sınıfının normal olduğu bir kuyrukta tutulduğunu ve kuyruk yöneticisini yeniden başlattığı belirtiliyordu.
    - Yönetim tarafından kuyruk temizleniyor.
    - Kuyruk siliniyor.
  - Bir konuda saklanan alıkonan yayın verilerini silmek için:
    - İletilerin kalıcı olmadığını belirtme ve kuyruk yöneticisini yeniden başlatma.
    - Alıkonan verileri yeni verilerle ya da ileti süre bitimini kullanarak değiştirme.
    - Yönetimsel olarak konu dizesi temizleniyor.
  - Yüksek kullanılabilirlik ya da olağanüstü durumdan kurtarma için tüm kuyruk yöneticisini ve eşlenen kopyaları silerek bir kuyruk yöneticisinde saklanan verileri silin.
  - İzleme dizinindeki dosyaları silerek, Hizmet izleme komutlarıyla saklanan verileri silin.
  - Hatalar dizinindeki dosyaları silerek saklanan FFST verilerini silin.
  - Adres alanını ve Coupling Facility dökümlerini silin ( z/OSüzerinde).
  - Bu tür verilerin arşivini, yedeklemesini ya da diğer kopyalarını silin.
- GDPR Hesap Verilerini silmeye uymak için göz önünde bulundurulması gereken IBM MQ davranış alanları
  - IBM MQ tarafından kuyruk yöneticilerine ve 3rd taraf hizmetlerine bağlanmak için depolanan hesap verilerini ve tercihlerini silmek için bunları silebilirsiniz (arşiv, yedekleme ya da başka bir şekilde eşlenmiş kopyaları da içinde olmak üzere):
    - Kimlik bilgilerini saklayan kuyruk yöneticisi kimlik doğrulama bilgileri nesnelere.
    - Kullanıcı tanıtıcılarına başvuran kuyruk yöneticisi yetkisi kayıtları.
    - Belirli IP adreslerini, sertifika DN ' lerini ya da kullanıcı tanıtıcılarını eşleyen ya da engelleyen kuyruk yöneticisi kanalı kimlik doğrulama kuralları.
    - Kuyruk yöneticisi ve dosya sunucularıyla kimlik doğrulaması için IBM MQ Managed File Transfer Agent, Logger ve MQ Explorer MFT Plugin tarafından kullanılan kimlik bilgileri dosyaları.
    - X.509 , SSL/TLS bağlantıları ya da IBM MQ Advanced Message Security (AMS) tarafından kullanılabilen anahtar depolarından bir kişiyi temsil eden ya da bu kişiye ilişkin bilgileri içeren sayısal sertifikalar.
    - Sistem günlük dosyalarında bu hesaplara başvuru da dahil olmak üzere, IBM MQ Appliance' den tek tek kullanıcı hesapları.
    - IBM MQ Explorer çalışma alanı meta verileri ve Eclipse ayarları.
    - IBM MQ Explorer Parola Deposu, [Parola Tercihleri'](#) nde belirtildiği şekilde.
    - IBM MQ Konsol ve mqweb sunucusu yapılandırma dosyaları.
    - Salesforce bağlantı verileri yapılandırma dosyaları.
    - Blockchain bağlantı verileri yapılandırma dosyaları.



- IBM MQ Internet Pass-Thru yapılandırma dosyaları ve anahtar depoları.

Daha fazla bilgi:

- [IBM MQ Bridge 'in Salesforceolarak yapılandırılması](#)
- [IBM MQ ' nun Blockchain ile kullanılmak üzere yapılandırılması](#)
- [MFT ve IBM MQ bağlantı kimlik doğrulaması](#)
- [ProtocolBridgeCredentials.xml dosyasını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerini eşleme](#)
- [IBM MQ Console kullanıcılarının ve rollerinin yapılandırılması](#)

## Verilerin İzlenmesi

IBM MQ , kullanıcıların uygulamaların ve kuyruk yöneticilerinin nasıl performans gösterdiğini daha iyi anlamak için kullanabilecekleri bir dizi izleme özelliği sağlar.

IBM MQ , kuyruk yöneticisi hata günlüklerinin yönetilmesine yardımcı olan bir dizi özellik de sağlar.

Daha fazla bilgi:

- [IBM MQ ağınızın izlenmesi](#)
- [Tanılama iletisi hizmetleri](#)
- [QMErrorLog hizmeti](#)
- [IBM MQ Appliance izleme ve raporlama](#)

## Kişisel Verilerin Kullanımını kısıtlama yeteneği

IBM MQ , bu belgede özetlenen olanakları kullanarak bir son kullanıcının kişisel verilerinin kullanımını kısıtlamasını sağlar.

IBM MQ ileti kuyrukları, bir veritabanıyla aynı şekilde kalıcı veri deposu olarak kullanılmamalıdır; bu özellikle GDPR ' ye tabi uygulama verileri işlenirken geçerlidir.

Verilerin arama sorgusuyla bulunabileceği bir veritabanından farklı olarak, bir iletinin kuyruğunu, ileti ve ilinti tanıtıcılarını bilmediğiniz sürece ileti verilerini bulmak zor olabilir.

Bir kişinin verilerini içeren iletiler kolayca tanımlanabilir ve bulunabilir; ileti verilerine erişmek ya da ileti verilerini değiştirmek için standart IBM MQ ileti sistemi özellikleri kullanılabilir.

## Dosya işleme

1. IBM MQ Managed File Transfer , aktarılan dosyalarda kötü amaçlı yazılım taraması gerçekleştirmez. Dosyalar olduğu gibi aktarılır ve aktarma sırasında dosya verilerinin değiştirilmediğinden emin olmak için bir bütünlük denetimi gerçekleştirilir. Kaynak ve hedef sağlama toplamları, aktarım durumu yayınının bir parçası olarak yayınlanır. Son kullanıcıların, MFT dosyayı aktarmadan önce ve MFT bir dosyayı uzak uç noktaya teslim ettikten sonra ortamlarına uygun şekilde kötü amaçlı yazılım taraması gerçekleştirmeleri önerilir.
2. IBM MQ Managed File Transfer , MIME tipine ya da dosya uzantısına dayalı olarak işlem yapmaz. MFT , dosyaları okur ve baytları tam olarak giriş dosyasından okunduğu gibi aktarır.

## Tek bir kuyruk yöneticisine dayalı mimariler

En basit IBM MQ mimarileri, tek bir kuyruk yöneticisinin yapılandırılmasını ve kullanılmasını içerir.

IBM MQ mimarinizi planlamadan önce, temel IBM MQ kavramlarını tanıyın. Bkz. [IBM MQ Teknik Genel Bakış](#).

Tek bir kuyruk yöneticisi kullanan olası mimarilerin sayısı aşağıdaki bölümlerde açıklanmıştır:

- [“Bir hizmete erişen yerel uygulamalara sahip tek kuyruk yöneticisi” sayfa 18](#)

- [“Bir hizmete istemci olarak erişen uzak uygulamalara sahip tek kuyruk yöneticisi” sayfa 18](#)
- [“Yayınlama/abone olma yapılandırması içeren tek kuyruk yöneticisi” sayfa 18](#)

## **Bir hizmete erişen yerel uygulamalara sahip tek kuyruk yöneticisi**

Tek bir kuyruk yöneticisine dayalı ilk mimari, hizmeti sağlayan uygulamalarla aynı sistemde bir hizmete erişen uygulamaların çalıştığı yerdir. Bir IBM MQ kuyruk yöneticisi, hizmeti isteyen uygulamalar ile hizmeti sağlayan uygulamalar arasında zamanuyumsuz ara iletişim sağlar. Bu, uygulamalardan biri uzun bir süre çevrimdışı olsa da, uygulamalar arasındaki iletişimin devam edebileceği anlamına gelir.

## **Bir hizmete istemci olarak erişen uzak uygulamalara sahip tek kuyruk yöneticisi**

Tek bir kuyruk yöneticisine dayalı olarak ikinci mimaride, hizmeti sağlayan uygulamalardan uzaktan çalışan uygulamalar vardır. Uzak uygulamalar, hizmetlerde farklı sistemlerde çalışıyor. Uygulamalar, tek kuyruk yöneticisine istemci olarak bağlanır. Bu, bir hizmete erişimin tek bir kuyruk yöneticisi aracılığıyla birden çok sisteme sağlanabileceği anlamına gelir.

Bu mimarinin bir sınırlaması, bir uygulamanın çalışması için ağ bağlantısının kullanılabilir olması gerekir. Ağ bağlantısı üzerinden uygulama ile kuyruk yöneticisi arasındaki etkileşim zamanuyumlu olur.

## **Yayınlama/abone olma yapılandırması içeren tek kuyruk yöneticisi**

Tek bir kuyruk yöneticisi kullanan alternatif bir mimari, yayınlama/abone olma yapılandırması kullanmandır. Yayınlama/abone olma mesajlarında, bilgi sağlayıcısını bu bilgilerin tüketicilerinden çezebilirsiniz. Bu, daha önce açıklanan mimarilerde, uygulamaların hedef uygulama hakkında bilgi sahibi olması gereken, örneğin, iletileri koymak için kuyruk adı gibi, noktadan noktaya ileti sistemi stillerine göre farklılık gösterir. Using IBM MQ publish/subscribe the sending application publishes a message with a specified topic based on the subject of the information. IBM MQ , iletinin dağıtımını, bir abonelik aracılığıyla bu konuya ilgi çeken uygulamalara dağıtacağını sağlar. Alma uygulamalarının, bunları alacak iletilerin kaynağı hakkında herhangi bir bilgi de bilmesi gerekmez. Daha fazla bilgi için bkz. [Yayınlama/abone olma ileti alışverişi](#) ve [Tek bir kuyruk yöneticisi yayınlama/abone olma yapılandırması örneği](#).

### **İlgili kavramlar**

[IBM MQ' a giriş](#)

### **İlgili görevler**

[“IBM MQ mimarisinin planlanması” sayfa 5](#)

IBM MQ ortamınızı planlarken, IBM MQ ' un tek ve birden çok kuyruk yöneticisi mimarisi ve noktadan noktaya iletişim ve yayınlama/abone olma ileti sistemi stilleri için sağladığı desteği göz önünde bulundurun. Ayrıca, kaynak gereksinimlerinizi ve günlük kaydı ve yedekleme olanaklarını kullanımınızı da planlayın.

[Çoklu Platformlar üzerinde kuyruk yöneticilerinin yaratılması ve yönetilmesi](#)

## **Birden çok kuyruk yöneticisine dayalı mimariler**

Birden çok kuyruk yöneticisinin yapılandırmasını ve kullanımını içeren bir IBM MQ mimarisi yaratmak için dağıtılmış ileti kuyruklama tekniklerini kullanabilirsiniz.

IBM MQ mimarinizi planlamadan önce, temel IBM MQ kavramlarını tanıyın. Bkz. [IBM MQ Teknik Genel Bakış](#).

Bir IBM MQ mimarisi, ek kuyruk yöneticileri ekleyerek, hizmet sağlayan uygulamalar için değişiklik yapılmaksızın değiştirilebilir.

Uygulamalar, kuyruk yöneticisiyle aynı makinede barındırılabilir ve başka bir sistemde başka bir kuyruk yöneticisiyle barındırılan bir hizmetle zamanuyumsuz iletişim kazanabilir. Diğer bir seçenek olarak, bir hizmete erişen uygulamalar, istemci olarak başka bir kuyruk yöneticilerindeki hizmete zamanuyumsuz erişim sağlayan bir kuyruk yöneticisine bağlanabilirler.

Farklı kuyruk yöneticilerini bağlayan rotalar ve bunların kuyrukları dağıtımlı kuyruğa alma teknikleri kullanılarak tanımlanır. Mimarinin içindeki kuyruk yöneticileri kanallar kullanılarak bağlanır. İletiler, kuyruk yöneticilerinin yapılandırmasına bağlı olarak, iletileri bir kuyruk yöneticisinden diğerine otomatik olarak başka yöne taşımak için kullanılır.

For a high level overview of planning an IBM MQ network, see [“Dağıtılmış kuyruk yöneticisi ağları tasarlanması” sayfa 20.](#)

For information about how to plan channels for your IBM MQ architecture, see [IBM MQ dağıtılmış kuyrukluama teknikleri.](#)

Dağıtılmış kuyruk yönetimi, kuyruk yöneticileri arasında iletişim yaratmanızı ve izlemenizi sağlar. Dağıtılmış kuyruk yönetimiyle ilgili ek bilgi için [Dağıtımlı kuyruk yönetimine girişbaşlıklı konuya](#) bakın.

### **İlgili görevler**

[“IBM MQ mimarisinin planlanması” sayfa 5](#)

IBM MQ ortamınızı planlarken, IBM MQ ' un tek ve birden çok kuyruk yöneticisi mimarisi ve noktadan noktaya iletişim ve yayınlama/abone olma ileti sistemi stilleri için sağladığı desteği göz önünde bulundurun. Ayrıca, kaynak gereksinimlerinizi ve günlük kaydı ve yedekleme olanaklarını kullanımınızı da planlayın.

[Çoklu Platformlar üzerinde kuyruk yöneticilerinin yaratılması ve yönetilmesi](#)

## **Dağıtılmış kuyruklarınızın ve kümelerinizin planlanması**

Dağıtımlı kuyruk yöneticilerindeki barındırılan kuyrukları el ile bağlayabilir ya da bir kuyruk yöneticisi kümesi yaratabilir ve ürünün kuyruk yöneticilerini sizin için bağlayıp bağlamanıza izin verebilirsiniz. Dağıtılmış ileti sistemi ağınız için uygun bir topoloji seçmek üzere el ile denetim, ağ boyutu, değişiklik sıklığı, kullanılabilirlik ve ölçeklenebilirlik gereksinimlerinizi göz önünde bulundurmanız gerekir.

### **Başlamadan önce**

Bu görev, dağıtılmış ileti sistemi ağlarının ne olduğunu ve bunların nasıl çalıştığını anladığınızı varsayar. Teknik bir genel bakış için bkz. [Dağıtılmış kuyruğa alma ve küme sayısı.](#)

### **Bu görev hakkında**

Dağıtılmış bir ileti sistemi ağı yaratmak için, farklı kuyruk yöneticilerindeki kuyruklar arasında bağlantı kurmak için kanalları el ile yapılandırabilir ya da bir kuyruk yöneticisi kümesi yaratabilirsiniz. Kümeleme, kuyruk yöneticilerinin, ek kanal tanımlamalarını ya da uzak kuyruk tanımlarını ayarlama, yapılandırma ve yönetimlerini basitleştirme gereksinimi olmadan birbirleriyle iletişim kurmalarını sağlar.

Dağıtılmış yayınlama/abone olma ağınız için uygun bir topoloji seçmek üzere aşağıdaki geniş soruları göz önünde bulundurmanız gerekir:

- Ağınızdaki bağlantılar için ne kadar el ile denetime gereksinim duyarsınız?
- Ağınız ne kadar büyük olacak?
- Ne kadar dinamik olacak?
- Kullanılabilirlik gereksinimleriniz ve ölçeklenebilirlik gereksinimleriniz nelerdir?

### **Yordam**

- Ağınızdaki bağlantılarda ne kadar el ile kontrol gerektiğini göz önünde bulundurun.

Yalnızca birkaç bağlantıya gereksinim duyarsanız ya da tek tek bağlantıların çok kesin olarak tanımlanması gerekiyorsa, ağı el ile yaratmalısınız.

Mantıksal olarak ilgili birden çok kuyruk yöneticisine gereksiniminiz varsa ve bu, verileri ve uygulamaları paylamanız gerekiyorsa, bunları bir kuyruk yöneticisi kümesinde birlikte gruplamanız gerekir.

- Ağınızın ne kadar büyük olması gerektiğini tahmin edin.

- a) Kaç kuyruk yöneticisi gerektiğini tahmin edin. Kuyrukların birden çok kuyruk yöneticisinde barındırılabilmesinin akılda olması gerekir.
- b) Bir küme kullanmayı düşünüyorsanız, tam havuz olarak hareket etmek üzere iki fazladan kuyruk yöneticisi ekleyin.

Daha büyük ağlar için, bağlantıların el ile yapılandırılması ve bakımı çok zaman alabilir ve bir küme kullanmayı göz önünde bulundurmanız gerekir.

- Ağ etkinliğinin ne kadar dinamik olacağını değerlendirin.

Performant kuyruk yöneticilerine ev sahipliği yapmak için yoğun kuyruklar planlayın.

Kuyrukların sık sık yaratılıp silinmesini bekliyorsanız, bir küme kullanmayı düşünün.

- Uygunluk ve ölçeklenebilirlik gereksinimlerinizi göz önünde bulundurun.
  - a) Kuyruk yöneticilerinin yüksek düzeyde kullanılabilirliğini garanti altına almak isteyip istemediğinizi belirleyin. Böyle bir durumda, bu gereksiminin kaç kuyruk yöneticisi için geçerli olduğunu tahmin edin.
  - b) Kuyruk yöneticilerinizin bazılarının diğerlerinden daha az yetenekli olup olmadığını göz önünde bulundurun.
  - c) Kuyruk yöneticilerinizin bazılarına yönelik iletişim bağlantılarının diğerlerine göre daha kırılabilir olup olmadığını göz önünde bulundurun.
  - d) Birden çok kuyruk yöneticisi üzerinde barındırma kuyruklarını dikkate alın.

El ile yapılandırılan ağların ve kümelerin hem yüksek düzeyde kullanılabilir hem de ölçeklenebilir olacak şekilde yapılandırılabilir. Bir küme kullanırsanız, tam havuz olarak iki fazladan kuyruk yöneticisi tanımlamanız gerekir. İki tam havuzun olması, tüm havuzlardan biri kullanılamaz duruma gelirse, kümenin çalışmaya devam etmesini sağlar. Tam havuz kuyruğu yöneticilerinin sağlam, performans gösterdiğinden ve iyi ağ bağlantılarına sahip olduğundan emin olun. Diğer herhangi bir iş için tam havuz kuyruğu yöneticilerini kullanmayı planlamayın.

- Bu hesaplamalara dayalı olarak, kuyruk yöneticileri arasında bağlantıların el ile yapılandırılıp yapılandırılmamasına ya da bir küme kullanmanıza karar vermenize yardımcı olması için sağlanan bağlantıları kullanın.

## Sonraki adım

Artık dağıtılmış ileti sistemi ağınızı yapılandırmak için hazırsınız.

### İlgili görevler

[Dağıtılmış kuyruklama yapılandırılması](#)

[Kuyruk yöneticisi kümesinin yapılandırılması](#)

## Dağıtılmış kuyruk yöneticisi ağları tasarlanması

IBM MQ , uygulamalar arasında veri alır ve Kuyruk Yöneticileri ve Kanallar kullanarak ağlar üzerinden veri alır. Ağ planlaması, bu sistemleri bir ağ üzerinden bağlamak üzere bir çerçeve oluşturmak için gereken gereksinimleri tanımlamayı içerir.

Kanallar, sisteminizle ve iletişimde bulunmanız gereken diğer sistemler arasında yaratılabilir. Doğrudan bağlantıların olmadığı sistemlere bağlanmak için çoklu sekme kanalları yaratılabilir. Senaryolarda açıklanan ileti kanalı bağlantıları, [Şekil 1 sayfa 21](#) içinde bir ağ çizgesi olarak gösterilir.

Farklı fiziksel ağlardaki sistemler ya da bir güvenlik duvarı üzerinden iletişim kuran kanallar arasında kanal yaratmanız gerekiyorsa, IBM MQ Internet Pass-Thru komutunu kullanarak yapılandırmayı basitleştirebilirsiniz. Daha fazla bilgi için bkz. [IBM MQ Internet Pass-Thru](#).

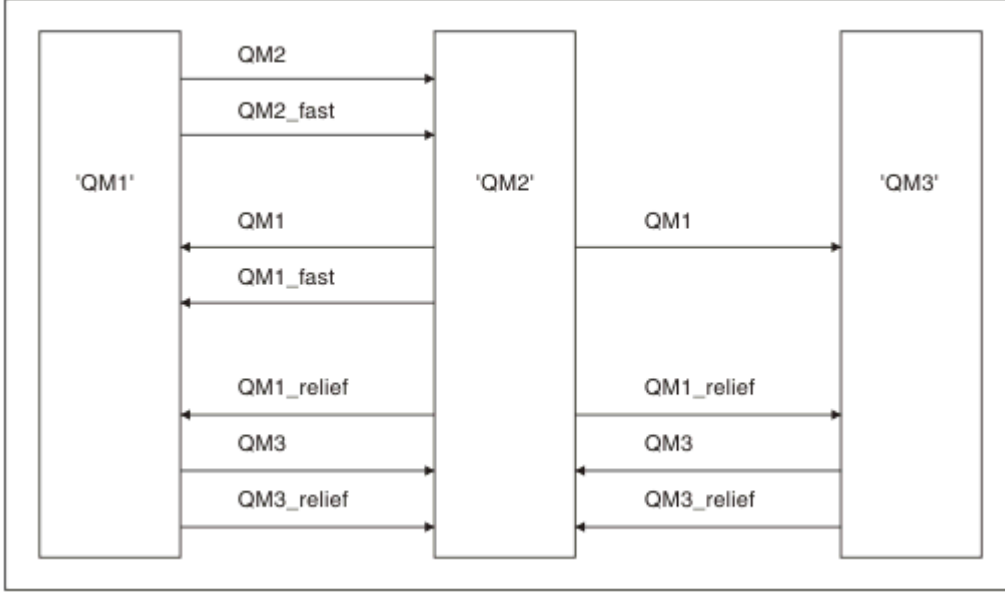
## Kanal ve iletim kuyruğu adları

İletim kuyruklarına herhangi bir ad verilebilir. Ancak karışıklığı önlemek için, hedef kuyruk yöneticisi adlarıyla aynı adları ya da kuyruk yöneticisi diğer ad adlarını uygun şekilde verebilirsiniz. Bu, iletim kuyruğunu kullandıkları rotayla ilişkilendirir ve ara (çok yönlü) kuyruk yöneticileri aracılığıyla yaratılan koşturmalara ilişkin açık bir genel bakış sağlar.

Kanal isimleri için o kadar da açık değil. The channel names in Şekil 1 sayfa 21 for QM2, for example, must be different for incoming and outgoing channels. Tüm kanal adları yine de iletim kuyruğu adlarını içerebilir, ancak bunların benzersiz olması için nitelenmeleri gerekir.

Örneğin, QM2'de QM1' den gelen bir QM3 kanalı ve QM3 kanalına giden bir QM3 kanalı vardır. Adları benzersiz yapmak için, birincinin adı QM3\_from\_QM1 olabilir, ikincisi ise QM3\_to\_QM2 olarak adlandırılabilir. Bu şekilde, kanal adları, adın ilk kısmında iletim kuyruğu adını gösterir. Yön ve bitişik kuyruk yöneticisi adı, adın ikinci kısmında gösterilir.


A table of suggested channel names for Şekil 1 sayfa 21 is given in Çizelge 1 sayfa 21.



Şekil 1. Tüm kanalları gösteren ağ çizgesi

Çizelge 1. Kanal adları örneği			
Rota adı	Kanal barındıran kuyruk yöneticileri	İletim kuyruğu adı	Önerilen kanal adı
QM1	QM1 & QM2	QM1 ( QM2)	QM1.from.QM2
QM1	QM2 & QM3	QM1 ( QM3)	QM1.from.QM3
QM1_fast	QM1 & QM2	QM1_fast ( QM2' de)	QM1_fast.from.QM2
QM1_relief	QM1 & QM2	QM1_relief ( QM2)	QM1_relief.from.QM2
QM1_relief	QM2 & QM3	QM1_relief ( QM3)	QM1_relief.from.QM3
QM2	QM1 & QM2	QM2 ( QM1)	QM2.from.QM1
QM2_fast	QM1 & QM2	QM2_fast ( QM1)	QM2_fast.from.QM1
QM3	QM1 & QM2	QM3 ( QM1' de)	QM3.from.QM1
QM3	QM2 & QM3	QM3 ( QM2)	QM3.from.QM2
QM3_relief	QM1 & QM2	QM3_relief ( QM1)	QM3_relief.from.QM1
QM3_relief	QM2 & QM3	QM3_relief ( QM2)	QM3_relief.from.QM2

**Not:**

1.  IBM MQ for z/OS üzerinde, kuyruk yöneticisi adları dört karakterle sınırlıdır.
2. Ağınızdaki tüm kanalları benzersiz bir şekilde adlayın. As shown in [Çizelge 1 sayfa 21](#), including the source and target queue manager names in the channel name is a good way to do so.

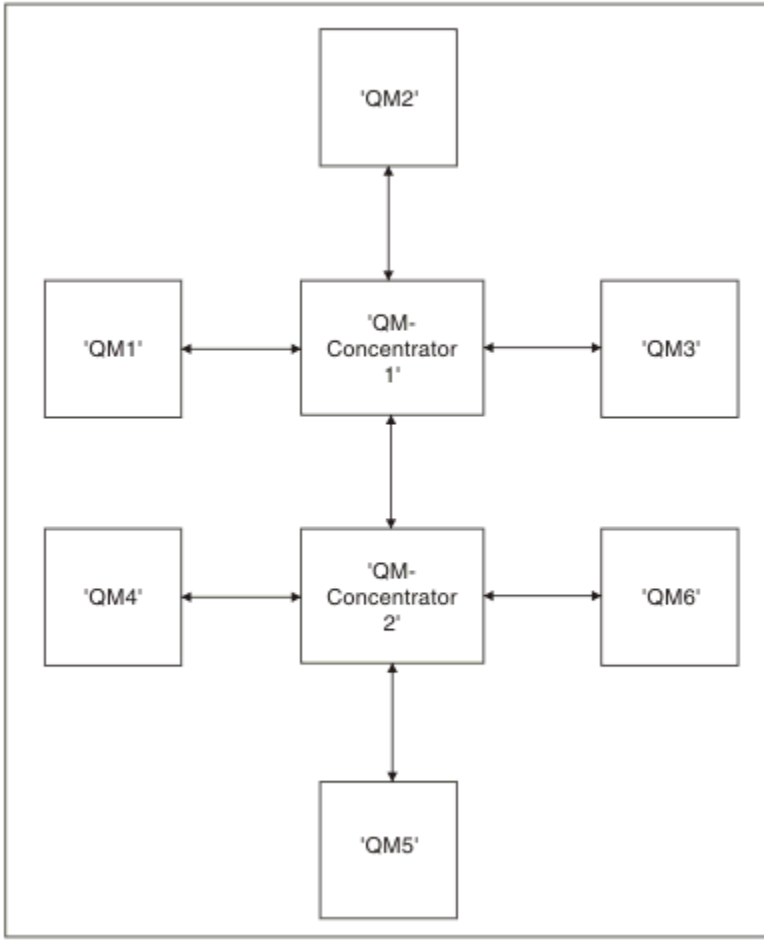
## Ağ planlayıcısı

Ağ oluşturma, planları ekibin diğer üyeleri tarafından uygulanmış olan başka bir *ağ planlayıcısı* düzeyinin daha yüksek düzeyde işlev göreceğini varsayar.

For widely used applications, it is more economical to think in terms of local access sites for the concentration of message traffic, using wide-band links between the local access sites, as shown in [Şekil 2 sayfa 23](#).

Bu örnekte iki ana sistem ve bir dizi uydu sistemi bulunmaktadır. Gerçek yapılandırma, işle ilgili önemli noktalara bağlı olacaktır. Uygun merkezlerde bulunan iki yoğunlaştırıcı kuyruk yöneticisi vardır. Her QM yoğunlaştırıcının, yerel kuyruk yöneticilerine ileti kanalları vardır:

- QM yoğunlaştırıcısı 1 'in, üç yerel kuyruk yöneticisinin ( QM1, QM2ve QM3) her birine ileti kanalları vardır. Bu kuyruk yöneticilerini kullanan uygulamalar, QM-yoğunlaştırıcılar aracılığıyla birbirleriyle iletişim kurabilirler.
- QM yoğunlaştırıcısı 2 'nin, üç yerel kuyruk yöneticisinin ( QM4, QM5ve QM6) her birine ileti kanalları vardır. Bu kuyruk yöneticilerini kullanan uygulamalar, QM-yoğunlaştırıcılar aracılığıyla birbirleriyle iletişim kurabilirler.
- QM yoğunlaştırıcıların, kuyruk yöneticisinde herhangi bir uygulamanın başka bir kuyruk yöneticisinde başka bir uygulama ile ileti alışverişi yapmasına olanak sağlayarak, QM yoğunlaştırıcılarının kendileriyle ileti kanalları vardır.



Şekil 2. QM yoğunlaştırıcılarını gösteren ağ çizgesi

## Kümelerin tasarlanması

Kümeler, hem ilk yapılandırmayı hem de devam eden yönetimi basitleştiren bir şekilde birbirine bağlanan kuyruk yöneticileri için bir mekanizma sağlar. Kümelerin doğru çalıştığından ve gerekli kullanılabilirlik ve yanıt verebilirlik düzeylerine ulaşabildiğinden emin olmak için kümeler dikkatli bir şekilde tasarlanmalıdır.

### Başlamadan önce

Kümeleme kavramlarına giriş için aşağıdaki konulara bakın:

- [Dağıtılmış kuyruğa alma ve kümeler](#)
- [“Kümeleme ve dağıtılmış kuyruklama karşılaştırması” sayfa 29](#)
- [Kümenin bileşenleri](#)

Kuyruk yöneticisi kümenizi tasarlarken bazı kararlar almanız gerekir. Önce kümedeki hangi kuyruk yöneticilerinin küme bilgilerinin tam havuzlarını tutacağına karar vermeniz gerekir. Yaratmış olduğunuz herhangi bir kuyruk yöneticisi bir kümede çalışabilir. Bu amaçla istediğiniz sayıda kuyruk yöneticisi seçebilirsiniz, ancak ideal sayı ikidir. Tam havuzları tutmak üzere kuyruk yöneticilerini seçme hakkında bilgi için bkz. [“Tüm havuzları tutmak için küme kuyruğu yöneticilerinin nasıl seçileceği” sayfa 32.](#)

Kümenizi tasarlamaya ilişkin ek bilgi için aşağıdaki konulara bakın:

- [“Örnek kümeler” sayfa 38](#)
- [“Küme düzenleme” sayfa 33](#)
- [“Küme adlandırma kuralları” sayfa 33](#)

- **z/OS** “Kuyruk paylaşımı grupları ve kümeleri” sayfa 34
- “Çakışan kümeler” sayfa 35

## Sonraki adım

Kümeleri yapılandırma ve kümelerle çalışma hakkında daha fazla bilgi için aşağıdaki konulara bakın:

- [Kümede iletişim kurulması](#)
- [Kuyruk yöneticisi kümesinin yapılandırılması](#)
- [İletileri kümelere/kümelerden yönlendirme](#)
- [İş yükü yönetimi için kümelerin kullanılması](#)

Kümenizi yapılandırmanıza yardımcı olacak daha fazla bilgi için bkz. “Kümeleme ipuçları” sayfa 36.

## **Birden çok küme iletim kuyruklarını nasıl kullandığınızı planlama**

İletim kuyruklarını belirttik olarak tanımlayabilir ya da sistem iletim kuyruklarını sizin için yaratmış olabilir. İletim kuyruklarını kendiniz tanımlarsanız, kuyruk tanımlamalarında daha fazla denetime sahip olursunuz.

**z/OS** z/OS üzerinde, iletilerin tutulduğu sayfa kümesi üzerinde de daha fazla denetime sahip olursunuz.

## İletim kuyruklarının tanımlanması

İletim kuyruklarını tanımlamaya ilişkin iki yöntem vardır:

- Aşağıdaki gibi, DEFCLXQ kuyruk yöneticisi özniteliğini kullanarak otomatik olarak:

```
ALTER QMGR DEFCLXQ(SCTQ | CHANNEL)
```

DEFCLXQ (SCTQ), tüm küme gönderici kanallarına ilişkin varsayılan iletim kuyruğunun SYSTEM.CLUSTER.TRANSMIT.QUEUE. Bu varsayılan değerdir.

DEFCLXQ (KANAL), varsayılan olarak her bir küme gönderici kanalının SYSTEM.CLUSTER.TRANSMIT.kanal adı. Her iletim kuyruğu otomatik olarak kuyruk yöneticisi tarafından tanımlanır. Ek bilgi için “Otomatik olarak tanımlanmış küme iletim kuyrukları” sayfa 25 başlıklı konuya bakın.

- CLCHNAME özniteliği için belirlenen değere sahip bir iletim kuyruğu tanımlayarak el ile. CLCHNAME özniteliği, iletim kuyruğunu hangi küme gönderici kanallarının kullanması gerektiğini belirtir. Ek bilgi için “El ile tanımlanmış küme iletim kuyruklarının planlanması” sayfa 27 başlıklı konuya bakın.

## Neye ihtiyacım var?

Bir anahtar otomatik olarak ya da el ile başlatmak için, bir kanal başlatmak için yetkiniz olması gerekir.

İletim kuyruğu olarak kullanılan kuyruğu tanımlamak için, kuyruğu tanımlamak için standart IBM MQ yetkinizin olması gerekir.

## değişikliği uygulamak için uygun bir zaman ne zaman?

Küme gönderici kanalları tarafından kullanılan iletim kuyruğunu değiştirirken, aşağıdaki noktaları göz önünde bulundurarak, güncellemeyi yapmak için bir süre ayırmanız gerekir:

- Bir kanala geçiş kuyruğunda geçiş için gereken süre, eski iletim kuyruğunda bulunan toplam ileti sayısına, kaç iletinin taşınması gerekmesine ve iletilerin büyüklüğünün ne kadar olduğunu bağlıdır.
- Uygulamalar, ileti değişikliği gerçekleşirken iletim kuyruğuna ileti yerleştirmeye devam edebilir. Bu, geçiş süresinde bir artışa neden olabilir.
- Herhangi bir iletim kuyruğunun ya da DEFCLXQ ' nun CLCHNAME parametresini istediğiniz zaman, tercihen iş yükü düşük olduğunda değiştirebilirsiniz.

Hiçbir şeyin hemen gerçekleşmemesine dikkat edin.



- Değişiklikler yalnızca bir kanal başlatıldığında ya da yeniden başlatıldığında gerçekleşir. Bir kanal başlatıldığında, yürürlükteki yapıları denetler ve gerekiyorsa, yeni bir iletim kuyruğuna geçer.
- İletim kuyruğuyla bir küme gönderici kanalının ilişkisini değiştirebilecek çeşitli değişiklikler vardır:
  - Bir iletim kuyruğunun CLCHNAME özniteliğinin değerini değiştirerek, CLCHNAME ' i daha az özel ya da boş hale getirir.
  - Bir iletim kuyruğunun CLCHNAME özniteliğinin değerini değiştirerek CLCHNAME ' yi daha ayrıntılı hale getirin.
  - Bir kuyruğun CLCHNAME ile silinmesi belirtildi.
  - DEFCLXQ kuyruk yöneticisi özniteliği değiştiriliyor.

## Anahtar ne kadar sürecek?

Geçiş dönemi boyunca, kanala ilişkin iletiler bir iletim kuyruğundan başka bir iletim kuyruğuna taşınır. Bir kanala geçiş kuyruğunda geçiş için gereken süre, eski iletim kuyruğunda bulunan iletilerin toplam sayısına ve kaç iletinin taşınması gerektiğini bağlıdır.

Birkaç bin ileti içeren kuyruklar için, iletileri taşımak için bir saniyenin altına girmelidir. Gerçek saat, iletilerin sayısına ve boyutuna bağlıdır. Kuyruk yöneticinizin iletileri her saniye bir çok megabayt cinsinden taşıyabilmelidir.

Uygulamalar, ileti değişikliği gerçekleşirken iletim kuyruğuna ileti yerleştirmeye devam edebilir. Bu, geçiş süresinde bir artışa neden olabilir.

Değişikliğin yürürlüğe girmesi için, etkilenen her bir küme gönderici kanalı yeniden başlatılmalıdır. Bu nedenle, kuyruk yöneticisi meşgul olmadığı zaman iletim kuyruğu yapılandırmasını değiştirmek en iyileridir ve küme iletim kuyruklarında birkaç ileti saklanır.

**runswch1** komutu, **z/OS** or the CSQUTIL ' te SWITCH CHANNEL (\*) STATUS komutu on z/OS, , küme-gönderici kanallarının durumunu sorgulamak için kullanılabilir ve bekleyen değişikliklerin iletim kuyruğu yapılandırması için ödenmemiş olması gerekir.

## Değişikliğin nasıl gerçekleştirileceği

Birden çok küme iletim kuyruğunda, otomatik olarak ya da el ile nasıl değişiklik yaptığınızla ilişkin ayrıntılar için [Sistemin birden çok küme iletim kuyrukları kullanılarak uygulanması](#) başlıklı konuya bakın.

## Değişikliği geri al

Sorunlarla karşılaşırsanız, değişikliklerin nasıl geri çıktığına ilişkin ayrıntılar için bkz. [Değişiklik yapmaması](#) .

*Otomatik olarak tanımlanmış küme iletim kuyrukları*  
Sistem iletim kuyruklarını sizin için oluşturmasını sağlar.

## Bu görev hakkında

Bir kanalda, kendisiyle ilişkilendirilmiş el ile tanımlanmış bir küme iletim kuyruğu yoksa ve kanal başlatıldığında, kuyruk yöneticisi otomatik olarak kümeli gönderen kanalı için kalıcı dinamik bir kuyruk tanımlıyorsa, DEFCLXQ (KANAL) seçeneğini belirtiyorsunuz. Model queue SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE is used to automatically define the permanent dynamic cluster transmit queue with the name SYSTEM.CLUSTER.TRANSMIT.Channe1Name.

**z/OS** Küme iletim kuyruklarını el ile ayarlamak için “[El ile tanımlanmış küme iletim kuyruklarının planlanması](#)” sayfa 27 başlıklı konuya bakın.

**Önemli:** **z/OS**

Kuyruk yöneticisi IBM MQ 8.0'e geçirildiyse, kuyruk yöneticisinin SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE.

Önce bu kuyruğu tanımlayın, böylece ALTER QMGR DEFCLXQ (KANAL) komutunun yürürlüğe girmesini sağlar.

Aşağıdaki JCL, model kuyruğunu tanımlamak için kullanabileceğiniz bir kod örneğidir:

```
//CLUSMODL JOB MSGCLASS=H,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=(MVCC)
//MQCMD EXEC PGM=CSQUTIL,REGION=4096K,PARM='CDLK'
//STEPLIB DD DISP=SHR,DSN=SCEN.MQ.V000.COM.BASE.SCSQAUTH
// DD DISP=SHR,DSN=SCEN.MQ.V000.COM.BASE.SCSQANLE
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(CMDINP)
/*
//CMDINP DD *
DEFINE QMODEL( 'SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE' ) +
QSGDISP( QMGR ) +

* COMMON QUEUE ATTRIBUTES
DESCR( 'SYSTEM CLUSTERING TRANSMISSION MODEL QUEUE' ) +
PUT( ENABLED ) +
DEFPRTY( 5 ) +
DEFFPSIST( YES ) +

* MODEL QUEUE ATTRIBUTES
DEFTYPE( PERMDYN ) +

* LOCAL QUEUE ATTRIBUTES
GET( ENABLED ) +
SHARE +
DEFSOPT( EXCL ) +
MSGDLVSQ( PRIORITY ) +
RETINTVL( 999999999 ) +
MAXDEPTH( 999999999 ) +
MAXMSGL( 4194304 ) +
NOHARDENBO +
BOTHRESH( 0 ) +
BOONAME( ' ' ) +
STGCLASS( 'REMOTE' ) +
USAGE( XMITQ ) +
INDXTYPE( CORRELID ) +
CFSTRUCT( ' ' ) +
MONQ( OFF ) ACCTQ( OFF ) +

* EVENT CONTROL ATTRIBUTES
QDPMAXEV( ENABLED ) +
QDPHIEV( DISABLED ) +
QDEPTHHI( 80 ) +
QDPLOEV( DISABLED ) +
QDEPTHLO( 40 ) +
QSVCEV( NONE ) +
QVCINT( 999999999 ) +

* TRIGGER ATTRIBUTES
TRIGGER +
TRIGTYPE( FIRST ) +
TRIGPRI( 0 ) +
TRIGPTH( 1 ) +
TRIGDATA( ' ' ) +
PROCESS( ' ' ) +
INITQ( ' ' )
/*
```

## Yordam

1. DEFCLXQ kuyruk yöneticisi özniteliğini kullanın.

Bu özniteliğe ilişkin ek bilgi için [ALTER QMGR](#) başlıklı konuya bakın.

İki seçenek vardır:

### SCTQ

Bu seçenek varsayılan değerdir ve tek bir SYSTEM.CLUSTER.TRANSMIT.QUEUE.

### Kanal

Birden çok küme iletim kuyruğu kullandığınızı gösterir.


2. Yeni ilişkilendirmeye geçmek için:

- Kanalı durdurun ve yeniden başlatın.
- Kanal, yeni iletim kuyruğu tanımlamasını kullanır.
- İletiler, eski kuyruktan yeni iletim kuyruğuna geçiş anahtarı işlemi tarafından aktarılır.

Herhangi bir uygulama iletilerinin eski tanımlamaya konulduğunu unutmayın.

Eski kuyruklardaki ileti sayısı sifıra ulaştığında, yeni iletiler doğrudan yeni iletim kuyruğuna yerleştirilir.

### 3. Anahtarlama işleminin sona erdiği zamanı izlemek için:

- a) Bir kanal tarafından başlatılan bir iletim kuyruğu anahtarı, arka planda çalışır ve sistem yöneticiniz kuyruk yöneticisi iş günlüğünü, ne zaman tamamlayıp tamamlayıp tamamlayacağını belirlemek için izleyebilir.
- b) Anahtardaki ilerlemeyi göstermek için iş günlüğündeki iletileri izleyin.
- c) Yalnızca istediğiniz kanalların bu iletim kuyruğunu kullandığından emin olmak için, DIS CLUSQMGR (\*) komutunu verin; örneğin, iletim kuyruğunu tanımlayan iletim kuyruğu özelliği APPQMGR . CLUSTER1 . XMITQ olur.
- d)  CSQUTIL altında SWITCH CHANNEL (\*) STATUS komutunu kullanın.  
Bu seçenek, bekleyen değişikliklerin ne olduğunu ve kaç iletinin iletim kuyrukları arasında taşınması gerektiğini size bildirir.

## Sonuçlar

Küme iletim kuyruğunuzu ya da kuyruklarınızı ayarladınız.

### İlgili görevler

[“El ile tanımlanmış küme iletim kuyruklarının planlanması” sayfa 27](#)

İletim kuyruklarını kendiniz tanımlıyorsanız, tanımlamalar üzerinde daha fazla denetime sahip olduğunuzu ve iletilerin düzenlendiği sayfa kümesini tanımlırdınız.

### İlgili başvurular

[ALTER QMGR](#)

[CLUSQMGR GÖRÜNTÜLE](#)

*El ile tanımlanmış küme iletim kuyruklarının planlanması*

İletim kuyruklarını kendiniz tanımlıyorsanız, tanımlamalar üzerinde daha fazla denetime sahip olduğunuzu ve iletilerin düzenlendiği sayfa kümesini tanımlırdınız.

## Bu görev hakkında

Yöneticiniz bir iletim kuyruğunu el ile tanımlar ve hangi küme gönderen kanalını ya da kanallarını tanımlamak için yeni bir kuyruk özniteliği CLCHNAME değerini kullanır; bu kuyruğu iletim kuyruğu olarak kullanır.

Tek bir kuyruğun birden çok kanal için kullanılmasına izin vermek için CLCHNAME ' in başında ya da sonunda bir genel arama karakteri içerebileceğini unutmayın.

Küme iletim kuyruklarını otomatik olarak ayarlamak için [“Otomatik olarak tanımlanmış küme iletim kuyrukları” sayfa 25](#) konusuna bakın.

## Yordam

### 1. Örneğin, aşağıdaki bilgileri girin:

```
DEFINE QLOCAL (APPQMGR . CLUSTER1 . XMITQ)
CLCHNAME (CLUSTER1 . TO . APPQMGR)
USAGE (XMITQ) STGCLASS (STG1)
INDXTYPE ( CORRELID ) SHARE

DEFINE STGCLASS (STG1) PSID (3)
DEFINE PSID (3) BUFFERPOOL (4)
```

**İpucu:** İletim kuyruklarınız için hangi sayfa kümesini (ve arabellek havuzunu) kullanabildiğinizi planlamanız gerekir. Farklı kuyruklar için farklı sayfa kümeleri olabilir ve bunlar arasında yalıtma sağlamak için, bir sayfa doldurma işlemi, diğer sayfa kümelerindeki iletim kuyruklarını etkilemez.

Her bir kanalın uygun kuyruğu nasıl seçilmesiyle ilgili bilgi için [Küme iletim kuyruklarıyla ve kümeyle gönderici kanallarıyla çalışma](#) başlıklı konuya bakın.

Kanal başlatıldığında, ilişkilendirmesini yeni iletim kuyruğuna geçirir. İleti kaybolmadığından emin olmak için, kuyruk yöneticisi iletileri eski küme iletim kuyruğundan otomatik olarak yeni iletim kuyruğuna aktarır.

2. Yeni ilişkilendirmeye geçmek için CSQUTIL SWITCH işlevini kullanın.

Ek bilgi için [Kümeyle gönderici kanallarla ilişkili iletim kuyruğunun değiştirilmesi \(ANAHTAR\)](#) başlıklı konuya bakın.

a) İletim kuyrukları değiştirilecek olan kanalı ya da kanalları DURDURULMUŞ durumda durdurmak için durdurun.

Örneğin:

```
STOP CHANNEL (CLUSTER1 .TO .APPQMGR)
```

b) İletim kuyruğunda CLCHNAME (XXXX) özniteliğini değiştirin.

c) İletileri değiştirmek ya da neler olduğunu izlemek için SWITCH işlevini kullanın.

Komutu kullan

```
SWITCH CHANNEL (*) MOVEMSGS (YES)
```

kanalı başlatmaksızın mesajları hareket ettirin.

d) Kanalı ya da kanalları başlatın ve kanalın doğru kuyrukları kullanıp kullanmadığını denetleyin.

Örneğin:

```
DIS CHS (CLUSTER1 .TO .APPQMGR)
DIS CHS (*) where (XMITQ eq APPQMGR .CLUSTER1 .XMITQ)
```

### İpucu:

- Aşağıdaki işlem CSQUTIL SWITCH işlevini kullanır; daha fazla bilgi için bkz. [Kümeyle gönderici kanallarla ilişkili iletim kuyruğunu değiştir \(SWITCH\)](#).

Bu işlevi kullanmak zorunda değilsiniz, ancak bu işlevi kullanmanız daha fazla seçenek sağlar:

- SWITCH CHANNEL (\*) STATUS (ANAHTAR KANALI) DURUMU, Küme Gönderen Kanallarının geçiş durumunu belirlemek için kolay bir yol sağlar. Denetimcinizin hangi kanalların değiştirildiğini görmenizi sağlar ve bu kanallar, sonraki başlatma sırasında geçerli olacak bir anahtar bekliyor (askıda).

Bu yetenek olmadan, denetimcinizin birden çok DISPLAY komutunu kullanması gerekir ve daha sonra, elde edilen çıktıyı bu bilgilerin kesinlemesiyle işler. Sistem yöneticiniz, bir yapılandırma değişikliğinin gerekli sonucu olduğunu da onaylayabilir.

- Anahtarı başlatmak için CSQUTIL kullanılırsa, CSQUTIL bu işlemin ilerleyişini izlemeye devam eder ve yalnızca anahtar tamamlanınca sona erer.

Bu işlem, bu işlemlerin toplu kipte gerçekleştirilmesini çok daha kolay bir hale getirebilir. CCOSTil birden çok kanalı değiştirmek için çalıştırılırsa, CSQUTIL bu işlemleri sırayla gerçekleştirir; bu işlem, işletmeniz için koşut olarak çalışan birden çok anahtardan daha az etkiye sahip olabilir.

## Sonuçlar

Küme iletim kuyruğunuzu ya da kuyruklarınızı ayarladınız.

### **Erişim denetimi ve birden çok küme iletim kuyruğu**

Bir uygulama uzak küme kuyruklarına ileti yerleştirdiğinde üç denetleme kipi arasında seçim yapın. Kipler, küme kuyruğunda uzaktan denetlenir, SYSTEM . CLUSTER . TRANSMIT . QUEUE' a yerel olarak denetlenir ya da küme kuyruğu ya da küme kuyruğu yöneticisi için yerel tanıtlara karşı denetlenir.


IBM MQ , yerel olarak ya da yerel olarak ya da yerel olarak, bir kullanıcının uzak bir kuyruğa ileti koyma iznine sahip olup olmadığını denetleme seçeneğini sunar. Tipik bir IBM MQ uygulaması yalnızca yerel denetimlerini kullanır ve uzak kuyruk yöneticisine dayalı olarak, yerel kuyruk yöneticisinde yapılan erişim denetimlerine güvenerek güvenir. Uzak denetim kullanılmıyorsa, ileti, uzak ileti kanalı işleminin yetkisiyle hedef kuyruğa konabilir. Uzaktan denetimi kullanmak için giriş kanalının koyma yetkisini bağlam güvenliğine ayarlamalısınız.

Yerel denetimler, uygulamanın açıldığını kuyruğa alır. Dağıtım kuyruğa alma sırasında, uygulama genellikle uzak bir kuyruk tanımlaması açar ve uzak kuyruk tanımlamasına karşı erişim denetimlerine erişir. İleti tam bir yöneltme üstbilgisiyle konursa, çekler iletim kuyruğunda yapılır. Bir uygulama yerel kuyruk yöneticisinde olmayan bir küme kuyruğunu açarsa, denetilecek bir yerel nesne yoktur. Erişim denetimi denetimleri, küme iletim kuyruğuna ( SYSTEM . CLUSTER . TRANSMIT . QUEUE ) ilişkin olarak yapılır. Birden çok küme iletim kuyrukları olsa da, SYSTEM . CLUSTER . TRANSMIT . QUEUE ' e karşı uzak küme kuyrukları için yerel erişim denetimi denetimleri yapılır.

Yerel ya da uzaktan denetime ilişkin seçenek, iki uç nokta arasında bir seçimdir. Uzaktan kontrol etmek çok ince bir işarettir. Her kullanıcının herhangi bir küme kuyruğuna konması için, kümedeki her kuyruk yöneticisinde bir erişim denetimi tanıtımı olmalıdır. Yerel olarak kontrol etmek kaba bir şekilde işarettir. Her kullanıcının, bağlı oldukları kuyruk yöneticilikteki küme iletim kuyruğu için tek bir erişim denetimi tanıtımı gerekir. Bu profille, herhangi bir kümedeki herhangi bir kuyruk yöneticisinde herhangi bir küme kuyruğuna ileti yerleştirebilir.

Denetimcilerin, küme kuyrukları için erişim denetimi ayarlamaya ilişkin başka bir yolu vardır. **setmqaut** komutunu kullanarak kümedeki herhangi bir kuyruk yöneticisinde küme kuyruğu için bir güvenlik profili yaratabilirsiniz. Uzak bir küme kuyruğunu yerel olarak açmanızı, yalnızca kuyruk adını belirterek, tanıtıma etki eder. Ayrıca, uzak kuyruk yöneticisi için bir tanıtım da ayarlayabilirsiniz. Bunu yapmazsanız, kuyruk yöneticisi tam olarak nitelenmiş bir ad sağlayarak, küme kuyruğunu açan kullanıcının tanıtımını denetleyebilir.

Yeni tanıtımlar yalnızca, kuyruk yöneticisi stanza 'yı ( **ClusterQueueAccessControl** - RQMAd1) değiştirdiğinizde çalışır. Varsayılan değer Xmq' dur. Varolan tüm küme kuyrukları için tanıtımlar yaratmalısınız, varolan uygulamalar küme kuyruklarını kullanır. Profilleri oluşturmadan stanza 'yı RQMName ' e çevirdiğinizde, uygulamaların başarısız olma olasılığı yüksektir.

**İpucu:** Denetmeye erişen küme kuyruğu, uzak kuyruklama için geçerli değildir. Erişim denetimleri hala yerel tanımlara göre yapılır. Bu değişiklikler, küme kuyruklarında ve küme konularındaki erişim denetimini yapılandırmak için aynı yaklaşımı izleyebileceğiniz anlamına gelir.  Değişiklikler, küme kuyruklarına ilişkin erişim denetimi yaklaşımını da z/OS ile daha yakından uyumlu hale getirmektedir. z/OS ' ta erişim denetimi ayarlanacak komutlar farklı, ancak her ikisi de, nesnenin kendisine karşı değil, profille ilgili erişim denetimi.

### İlgili kavramlar

“Kümeleme: Birden çok küme iletim kuyruğu kullanılarak uygulama yalıtımı” sayfa 46

Bir kümedeki kuyruk yöneticileri arasındaki ileti akışlarını yalıtabilirsiniz. Farklı küme gönderici kanalları tarafından, farklı küme iletim kuyruklarına iletilmekte olan iletileri yerleştirebilirsiniz. Yaklaşımı tek bir kümede ya da çakışan kümelerle kullanabilirsiniz. Bu konuda, kullanabileceğiniz bir yaklaşım seçiminde size yol göstermesi için bazı örnekler ve bazı en iyi uygulamalar sağlanır.

### İlgili görevler

[Ayar ClusterQueueAccessControl](#)

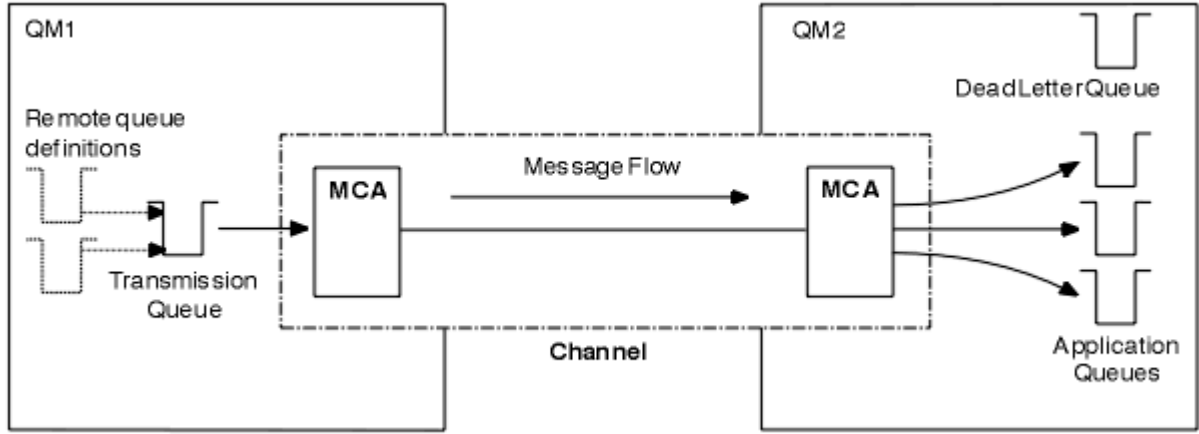
### **Kümeleme ve dağıtılmış kuyruklama karşılaştırması**

Dağıtılmış kuyruğa alma ve kümeleme kullanarak kuyruk yöneticilerini bağlamak için tanımlanması gereken bileşenleri karşılaştırın.

Kümeleri kullanmayarsanız, kuyruk yöneticilerinizin dağıtım kuyruğa alma ile bağımsız ve bağımsız bir şekilde iletişim kurmanızı sağlar. Bir kuyruk yöneticisinin başka bir kuyruk yöneticisine ileti göndermesi gerekiyorsa, şunları tanımlamanız gerekir:

- İletim kuyruğu
- Uzak kuyruk yöneticisine bir kanal

Şekil 3 sayfa 30 , dağıtılmış kuyruklama için gereken bileşenleri gösterir.



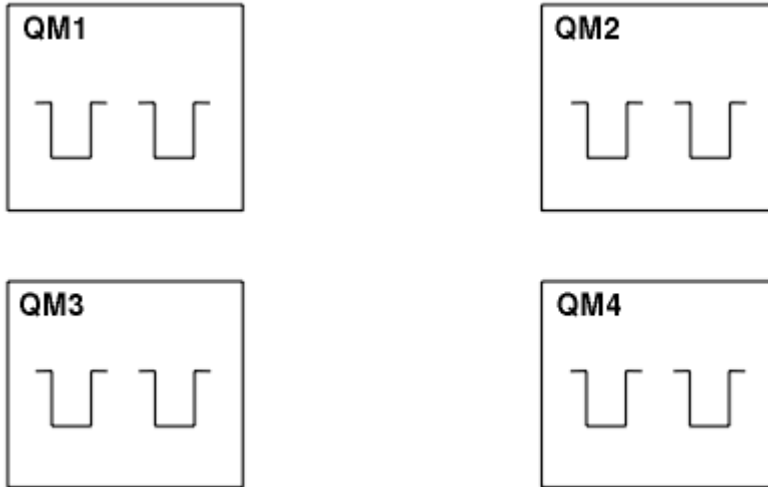
Şekil 3. Dağıtılmış kuyruklama

Bir kümedeki kuyruk yöneticilerini gruplarsanız, herhangi bir kuyruk yöneticisinde bulunan kuyruklar, kümedeki başka bir kuyruk yöneticisinde kullanılabilir. Herhangi bir kuyruk yöneticisi, belirtilik tanımlamalarla aynı kümedeki başka bir kuyruk yöneticisine ileti gönderebilir. Her hedef için kanal tanımları, uzak kuyruk tanımları ya da iletim kuyrukları sağlamıyor musunuz? Bir kümedeki her kuyruk yöneticisinin, iletileri kümedeki diğer bir kuyruk yöneticisine iletebileceği tek bir iletim kuyruğu vardır. Bir küme içindeki her kuyruk yöneticisinin yalnızca tanımlanması gerekir:

- İletilerin alınacağı bir küme-alıcı kanalı
- Kendi kendisini tanıtan ve kümeyi öğrenen bir küme-gönderici kanalı

### Bir küme ve dağıtılmış kuyruklama ayarına ilişkin tanımlar

Her biri iki kuyrukla dört kuyruk yöneticisini gösteren Şekil 4 sayfa 30' a bakın. Dağıtılmış kuyruklama özelliğini kullanarak bu kuyruk yöneticilerini bağlamak için kaç tanım gerektiğini göz önünde bulundurun. Bir kümeyle aynı ağı kurmak için gereken tanımların sayısını karşılaştırın.



Şekil 4. Dört kuyruk yöneticilerinden oluşan bir ağ

### Dağıtılmış kuyruklama kullanarak bir ağ oluşturmak için tanımlar

Dağıtılmış kuyruklama kullanarak Şekil 3 sayfa 30 içinde gösterilen ağı ayarlamak için aşağıdaki tanımlara sahip olabilirsiniz:

<i>Çizelge 2. Dağıtılmış kuyruğa alma tanımlarına ilişkin tanımlar</i>		
<b>Tanım</b>	<b>Kuyruk yöneticisi başına sayı</b>	<b>Toplam sayı</b>
Diğer tüm kuyruk yöneticilerine iletilerin gönderileceği bir kanala ilişkin gönderici kanal tanımlaması	3	12
Diğer tüm kuyruk yöneticilerinden iletilerin alınacağı bir kanala ilişkin bir günlük nesnesi kanal tanımlaması	3	12
Diğer kuyruk yöneticiliklerine iletim kuyruğu için bir iletim kuyruğu tanımlaması	3	12
Her yerel kuyruk için yerel kuyruk tanımlaması	2	8
Bu kuyruk yöneticisinin iletileri koymak istediği her uzak kuyruk için bir uzak kuyruk tanımlaması	6	24

Soysal alıcı kanal tanımlamalarını kullanarak bu tanımlama sayısını azaltabilir. Tanımlama sayısı üst sınırı, bu ağ için toplam 68 olan her bir kuyruk yöneticisinde en çok 17 olabilir.

### **Kümeleri kullanarak bir ağ oluşturmak için tanımlar**

Şekil 3 sayfa 30 ' ta gösterilen ağ, kümeleri kullanarak ayarlamak için aşağıdaki tanımlara gerek duyarsınız:

<i>Çizelge 3. Kümelemeye ilişkin tanımlar</i>		
<b>Tanım</b>	<b>Kuyruk yöneticisi başına sayı</b>	<b>Toplam sayı</b>
Bir havuz kuyruğu yöneticisine ileti göndermek için kullanılan bir kanala ilişkin küme gönderen kanal tanımlaması	1	4
Kümedeki diğer kuyruk yöneticilerinden iletilerin alınacağı kanala ilişkin bir küme alıcı kanal tanımlaması	1	4
Her yerel kuyruk için yerel kuyruk tanımlaması	2	8

Bu kuyruk yöneticisi kümesini (iki tam havuzla) ayarlamak için, her kuyruk yöneticisi için toplam on altı tanım olmak üzere dört tanımlama gerekir. Ayrıca, kuyruk yöneticisi tanımlamalarını, küme için tam havuz kuyruğu yöneticilerine yapmak üzere kuyruk yöneticisi tanımlamalarını değiştirmeniz de gerekir.

Yalnızca bir CLUSTSDR ve bir CLUSTRVR kanal tanımlaması gerekli. Küme tanımlandığında, diğer kuyruk yöneticilerine herhangi bir kesinti olmadan kuyruk yöneticisi (havuz kuyruğu yöneticileri dışında) ekleyebilir ya da var olan bir kuyruk yöneticisini kaldırabilirsiniz.

Bir küme kullanılması, birçok kuyruk yöneticisi içeren bir ağ oluşturmak için gereken tanımların sayısını azaltır.

Daha az tanımın olması, hata riskinin daha az olduğunu ortaya çıkartır:

- Nesne adları her zaman eşleşir; örneğin, bir gönderen alıcı çiftindeki kanal adı.
- Bir kanal tanımlamasında belirtilen iletim kuyruğu adı, her zaman doğru iletim kuyruğu tanımlamasıyla ya da uzak bir kuyruk tanımlamasında belirlenen iletim kuyruğu adıyla eşleşir.
- Bir QREMOTE tanımlaması her zaman uzak kuyruk yöneticisinde doğru kuyruğu işaret eder.

Bir küme ayarlandıktan sonra, başka bir kuyruk yöneticisinden herhangi bir sistem yönetimi çalışması yapmak zorunda kalmadan, küme kuyruklarını küme içinde bir kuyruk yöneticisinden diğerine taşıyabilirsiniz. Kanal, uzak kuyruk ya da iletim kuyruğu tanımlarını silmeyi ya da değiştirmeyi unutmanın

bir şansı yoktur. Var olan ağda herhangi bir kesinti olmadan, bir kümeye yeni kuyruk yöneticileri ekleyebilirsiniz.

### ***Tüm havuzları tutmak için küme kuyruğu yöneticilerinin nasıl seçileceği***

Her bir kümede en az bir adet ve tercihen iki kuyruk yöneticisi tam olarak havuz tutmasını tercih etmelisiniz. İki tam havuz, en istisnai durumlar için yeterlidir. Mümkünse, sağlam ve kalıcı olarak bağlantılı platformlarda barındırılan, çakışan kesintiler olmayan ve coğrafi olarak merkezi bir konumda olan kuyruk yöneticilerini seçin. Ayrıca, sistemleri tam havuz anasistemleri olarak ayırmanın yanı sıra, bu sistemleri başka görevler için kullanmamayı da göz önünde bulundurun.

*Tam havuzlar* , kümenin durumunu tam olarak koruyan kuyruk yöneticileridir. Bu bilgileri paylaşmak için, her tam havuz CLUSSDR kanalları (ve karşılık gelen CLUSRCVR tanımlamaları) tarafından kümedeki diğer tüm tam havuzlarla bağlantılıdır. Bu kanalları el ile tanımlamanız gerekir.



Şekil 5. İki bağlantılı tam havuz.

Kümedeki diğer her kuyruk yöneticisi, bir *kısmi havuz* içindeki kümenin durumu hakkında şu anda ne bilse de bir resmi saklar. Bu kuyruk yöneticileri, kendilerine ilişkin bilgileri yayınlar ve kullanılabilir iki tam havuzu kullanarak diğer kuyruk yöneticilerine ilişkin bilgi ister. Seçilen bir tam havuz kullanılamıyorsa, başka bir havuz kullanılır. Seçilen tam havuz yeniden kullanılabilir duruma geldiğinde, en son yeni ve değiştirilmiş bilgileri diğer kullanıcıların adımlarına devam etmek üzere toplar. Tüm havuzların tümü hizmet dışına çıkıyorsa, diğer kuyruk yöneticileri, kısmi havuzlarındaki bilgileri kullanır. Ancak, bunlar sahip oldukları bilgileri kullanmakta sınırlanmıştır; yeni bilgiler ve güncellemeler için istekler işlenemez. Tüm havuzlar ağa yeniden bağlandığında, tüm havuzları (hem tam hem de kısmi) güncel olarak getirmek için iletiler değiştirilir.

Tüm havuzların tahsisini planlarken aşağıdaki noktaları göz önünde bulundurulması gerekir:

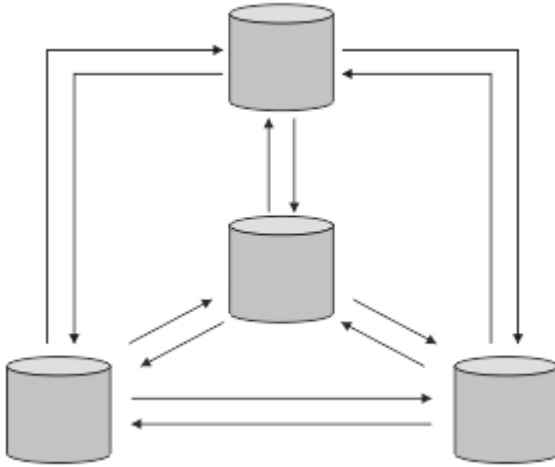
- Tüm havuzları tutmak için seçilen kuyruk yöneticilerinin güvenilir ve yönetimli olması gerekir. Sağlam ve kalıcı olarak bağlantılı bir platformda barındırılan kuyruk yöneticilerini seçin.
- Tüm havuzlarınıza ev sahipliği yapan sistemler için planlanmış kesintileri göz önünde bulundurun ve bu kesintilerin denk gelmesi durumunda olmamalarını sağlayın.
- Ağ performansını göz önünde bulundurun: Coğrafi konumda bulunan ya da kümedeki diğer kuyruk yöneticileriyle aynı sistemi paylaşan kuyruk yöneticilerini seçin.
- Kuyruk yöneticisinin birden çok küme üyesi olup olmadığını dikkate alın. Bu avantajın, kuyruk yöneticisinin ne kadar meşgul olmasını beklediğiniz konusunda dengeli olması koşuluyla, aynı kuyruk yöneticisini birden çok küme için tüm havuzları barınmak üzere kullanması için uygun bir şekilde uygun bir yönetim olabilir.
- Bazı sistemleri yalnızca tam havuzları içerecek şekilde ayırmanın ve bu sistemleri başka herhangi bir görev için kullanılmayalı olarak değerlendirin. Bu, bu sistemlerin yalnızca kuyruk yöneticisi yapılandırması için bakım gerektirmesini ve diğer iş uygulamalarının bakımına ilişkin hizmetten kaldırılmamasını sağlar. Ayrıca, havuzu koruma görevinin sistem kaynakları için uygulamalarla rekabet etmemesini sağlar. Bu özellikle, tam havuzların küme durumunu korumada çok daha yüksek bir iş yüküne sahip olduğu büyük kümelerde (örneğin, binden fazla kuyruk yöneticisi kümelerine) yararlı olabilir.

İki tam havuzdan daha fazla havuz olması mümkün, ancak nadiren önerilir. Nesne tanımlamaları (yani, kuyruklar, konular ve kanallar) kullanılabilir tüm tüm havuzlara akışı olsa da, kısmi bir havuzdan en çok iki tam havuza doğru akış sağlar. Bu, iki tam havuzun tanımlandığı ve iki tam havuzun kullanılamaz duruma geldiği anlamına gelir; bazı kısmi havuzlar bekledikleri güncellemeleri alamayabilir. Bkz. [MQ Cluster ürünleri: Neden yalnızca iki Tam Havuz?](#)

Var olan tam havuzları, yeni donanıma ya da yeni kuyruk yöneticilerine geçirirken, ikiden fazla tam havuz tanımlamayı yararlı bulabileceğiniz bir durum. Bu durumda, önceki tam havuzları kaldırmadan önce, yeni



tam havuzları tanıtmalı ve tam olarak doldurulduklarını onaylamanız gerekir. Tam bir havuz eklediğinizde, bunu CLUSSDR kanallarıyla diğer tüm havuzlara doğrudan bağlamanız gerektiğini unutmayın.



Şekil 6. İki bağlantılı tam havuz sayısı

### İlgili bilgiler

MQ Cluster ürünleri: Neden yalnızca iki Tam Havuz?

MQ Kümesi ne kadar büyük olabilir?


### Küme düzenleme

Hangi kuyruk yöneticilerinin hangi tam havuza bağlanacağı seçin. Performans etkisini, kuyruk yöneticisi sürümünü ve birden çok CLUSSDR kanalının istenip istenmediğini göz önünde bulundurun.

Tam havuzları tutmak için kuyruk yöneticilerini seçmiş olması, hangi kuyruk yöneticilerinin hangi tam havuza bağlanmaya karar vereceğine karar vermeniz gerekir. CLUSSDR kanal tanımlaması, bir kuyruk yöneticisini kümedeki diğer tam havuzlarla ilgili olarak bulduğu tam havuza bağlar. Bundan sonra, kuyruk yöneticisi iki tam havuza ileti gönderir. Her zaman önce bir CLUSSDR kanal tanımlamasına sahip olduğu bir kanal tanımlamasını kullanmayı dener. Bir kuyruk yöneticisini tüm havuzla bağlantıyla ilişkilendirmeyi seçebilirsiniz. Seçim yaparken, yapılandırmanın topolojisini ve kuyruk yöneticilerinin fiziksel ya da coğrafi konumunu göz önünde bulundurun.

Tüm küme bilgileri iki tam havuza gönderildiğinden, ikinci bir CLUSTSDR kanal tanımlaması yapmak istediğiniz durumlar olabilir. Geniş bir alana yayılmış birçok tam havuzu olan bir kümede ikinci bir CLUSSDR kanalı tanımlayabilirsiniz. Daha sonra, bilgilerinizin gönderildiği iki tam havuzun hangilerinin gönderileceğini denetleyebilirsiniz.

### Küme adlandırma kuralları

Kuyruk yöneticisinin ait olduğu kümeyi tanıtan bir adlandırma kuralı kullanarak aynı kümedeki kuyruk yöneticilerini adlandırmayı düşünün. Kanal adları için benzer bir adlandırma kuralı kullanın ve kanal özelliklerini açıklayacak şekilde genişletin.  z/OS üzerinde küme alıcı kanalları tanımlarken sosyal bağlantı kullanmayın.

### MQ Cluster ürünlerini adlandırırken en iyi uygulamalar

Küme adları en çok 48 karakter olabilsede, diğer nesnelere adlandırma kuralları uygulanırken görece olarak kısa küme adları yararlı olur. Bkz. “Küme kanalı adlarını seçerken en iyi uygulamalar” sayfa 34.

Bir küme adı seçerken, genellikle 'içerik' yerine kümenin 'amacını' (uzun ömürlü olması muhtemeldir) temsil etmek yararlı olur. Örneğin, 'QM1\_QM2\_QM3\_CLUS' yerine 'B2BPROD' veya 'ACTTEST'.

## Küme Kuyruğu Yöneticisi adlarını seçerken en iyi uygulamalar

Yeni bir küme ve üyeleri sıfırdan oluşturuyorsanız, kuyruk yöneticileri için küme kullanımını yansıtan bir adlandırma kuralı düşünün. Her kuyruk yöneticisinin farklı bir adı olmalıdır. Ancak, bir kümedeki kuyruk yöneticilerine mantıksal gruplamaların (örneğin, 'ACTTQM1, ACTTQM2) tanımlanmasına ve anımsamasına yardımcı olması için benzer adlar verebilirsiniz.

Görece kısa kuyruk yöneticisi adları (örneğin, 8 karakterden kısa), kanal adları için sonraki bölümde açıklanan kuralı ya da benzeri bir şeyi kullanmayı seçerseniz yardımcı olur.

## Küme kanalı adlarını seçerken en iyi uygulamalar

Kuyruk yöneticileri ve kümelerin adları en çok 48 karakterden oluşabildiğinden ve bir kanal adı 20 karakterle sınırlandırıldığından, bir projenin ortasında adlandırma kuralını değiştirmek zorunda kalmamak için ilk adlandırma nesnelere dikkatli olun (önceki bölüme bakın).

Kanalları tanımlarken, kümedeki herhangi bir kuyruk yöneticisinde otomatik olarak yaratılan küme gönderen kanallarının adlarını, kümedeki alıcı kuyruk yöneticisinde yapılandırılan karşılık gelen küme-alıcı kanalından aldığı ve bu nedenle bu kanalların benzersiz olması ve kümedeki uzak kuyruk yöneticilerine anlamlı olması gerektiğini unutmayın.

Yaygın bir yaklaşım, küme adının önünde bulunan kuyruk yöneticisi adını kullanmaktır. Örneğin, küme adı CLUSTER1 ve kuyruk yöneticileri QM1, QM2 ise, küme alıcı kanalları CLUSTER1.QM1, CLUSTER1.QM2 olur.

Kanalların farklı öncelikleri varsa ya da farklı iletişim kuralları kullanıyorsa, bu kuralı genişletebilirsiniz. Örneğin:

- CLUSTER1.QM1.S1
- CLUSTER1.QM1.N3
- CLUSTER1.QM1.T4

Bu örnekte, S1 ilk SNA kanalı, N3 üç ağ önceliğine sahip NetBIOS kanalı ve T4 IPV4 ağı kullanan TCP/IP olabilir.

## Paylaşılan kanal tanımlarını adlandırma

Tek bir kanal tanımı birden çok küme arasında paylaşılabilir; bu durumda, burada önerilen adlandırma kurallarının değiştirilmesi gerekir. Ancak, [Kanal tanımlarının yönetilmesi](#) konusunda açıklandığı gibi, her durumda her küme için ayrı kanallar tanımlanması tercih edilir.

## Eski kanal adlandırma kuralları

Küme ortamlarının dışında geçmişte 'FROMQM.TO.TARGETQM' adlandırma kuralını kullanmak yaygındır, bu nedenle var olan kümelerin benzer bir şey kullandığını bulabilirsiniz (CLUSTER.TO.TARGET gibi). Yeni bir küme adlandırma şemasının parçası olarak bu önerilmez; kanal adınız içinde 'yararlı' bilgileri iletmek için kullanılacak karakterleri daha da azaltır.

## z/OS IBM MQ for z/OS üzerindeki kanal adları

VTAM sosyal kaynaklarını ya da *Dinamik Etki Alanı Ad Sunucusu* (DDNS) sosyal adlarını tanımlayabilirsiniz. Sosyal adları kullanarak bağlantı adları tanımlayabilirsiniz. Ancak, bir küme-günlük nesnesi tanımlaması yarattığınızda, sosyal bir bağlantı adı kullanmayın.

Küme-günlük nesnesi tanımlamaları için sosyal bağlantı adlarının kullanılmasıyla ilgili sorun şöyledir: Sosyal CONNAME ile bir CLUSRCVR tanımlarsanız, CLUSSDR kanallarınızın amaçladığınız kuyruk yöneticilerini gösterdiğine ilişkin bir garanti yoktur. Başlangıçtaki CLUSSDR, kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisini gösteriyor olabilir, tam havuz bulduran bir kuyruk yöneticisini göstermeyebilir. Bir kanal bağlantıyı yeniden denemeye başlarsa, aynı sosyal ada sahip farklı bir kuyruk yöneticisine yeniden bağlanarak ileti akışını kesintiye uğratabilir.

## z/OS Kuyruk paylaşımı grupları ve kümeleri

Paylaşılan kuyruklar küme kuyrukları ve bir kuyruk paylaşım grubundaki kuyruk yöneticileri de küme kuyruğu yöneticileri olabilir.

IBM MQ for z/OS ' da kuyruk yöneticilerini kuyruk paylaşım grupları içinde gruplayabilirsiniz. Bir kuyruk paylaşım grubundaki bir kuyruk yöneticisi, en çok 32 kuyruk yöneticisi tarafından paylaşılacak bir yerel kuyruk tanımlayabilir.

Paylaşılan kuyruklar küme kuyrukları da olabilir. Ayrıca, bir kuyruk paylaşım grubundaki kuyruk yöneticileri de bir ya da daha fazla kümede olabilir.

VTAM soysal kaynaklarını ya da *Dinamik Etki Alanı Ad Sunucusu* (DDNS) soysal adlarını tanımlayabilirsiniz. Soysal adları kullanarak bağlantı adları tanımlayabilirsiniz. Ancak, bir küme-günlük nesnesi tanımlaması yarattığınızda, soysal bir bağlantı adı kullanmayın.

Küme-günlük nesnesi tanımlamaları için soysal bağlantı adlarının kullanılmasıyla ilgili sorun şöyledir: Soysal CONNAME ile bir CLUSRCVR tanımlarsanız, CLUSSDR kanallarınızın amaçladığınız kuyruk yöneticilerini gösterdiğine ilişkin bir garanti yoktur. Başlangıçtaki CLUSSDR , kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisini gösteriyor olabilir, tam havuz bulduran bir kuyruk yöneticisini göstermeyebilir. Bir kanal bağlantıyı yeniden denemeye başlarsa, aynı soysal ada sahip farklı bir kuyruk yöneticisine yeniden bağlanarak ileti akışını kesintiye uğratabilir.

Grup dinleyici kapısını kullanan bir CLUSRCVR kanalı başlatılamıyor; böyle bir durumda, CLUSRCVR ' nin her seferinde hangi kuyruk yöneticisine bağlanacağı belirlenemez. Kümeyle ilgili bilgilerin paylaşılmadığı küme sistemi kuyrukları. Her kuyruk yöneticisinin kendine ait bir kuyruk yöneticisi vardır.

Küme kanalları yalnızca uygulama iletilerini aktarmak için değil, kümenin kurulumuna ilişkin iç sistem iletilerini aktarmak için de kullanılır. Kümedeki her kuyruk yöneticisinin bu iç sistem iletilerini kümelemenin doğru bir şekilde katılması için alması gerekir; bu nedenle, bunların alınacağı kendi benzersiz CLUSRCVR kanalı gerekir.

Paylaşılan bir CLUSRCVR , kuyruk paylaşım grubundaki (QSG) herhangi bir kuyruk yöneticisinde başlayabilir ve QSG kuyruk yöneticilerine iç sistem iletilerinin tutarsız bir şekilde sağlanmasına neden olabilir; bu da, kümeyle doğru bir şekilde katılmayacaklarını gösterir. Paylaşılan CLUSRCVR kanallarının kullanılmadığından emin olmak için, herhangi bir girişim CSQX502E iletilisiyle başarısız olur.

### **Çakışan kümeler**

Çakışan kümeler, ek yönetim yetenekleri sağlar. Çakışan kümeleri denetlemek için gereken komut sayısını azaltmak için ad listelerini kullanın.

Çakışan kümeleri oluşturabilirsiniz. Çakışan kümeleri tanımlayabileceğiniz bir dizi nedeni vardır; örneğin:

- Farklı organizasyonların kendi yönetimlerine sahip olmasını sağlamak.
- Bağımsız uygulamaların ayrı olarak denetlenmesine izin vermek için.
- Hizmet sınıfları yaratmak için.

Şekil 7 sayfa 36'inde, kuyruk yöneticisi STF2 hem kümelerin bir üyesidir. Kuyruk yöneticisi birden çok kümenin üyesi olduğunda, gereksinim duyduğunuz tanımlama sayısını azaltmak için ad listelerinden yararlanabilirsiniz. Ad listeleri, ad listesi (örneğin, küme adları) içerir. Kümeleri adlandırma adına bir ad listesi oluşturabilirsiniz. Specify the namelist on the ALTER QMGR command for STF2 to make it a full repository queue manager for both clusters.

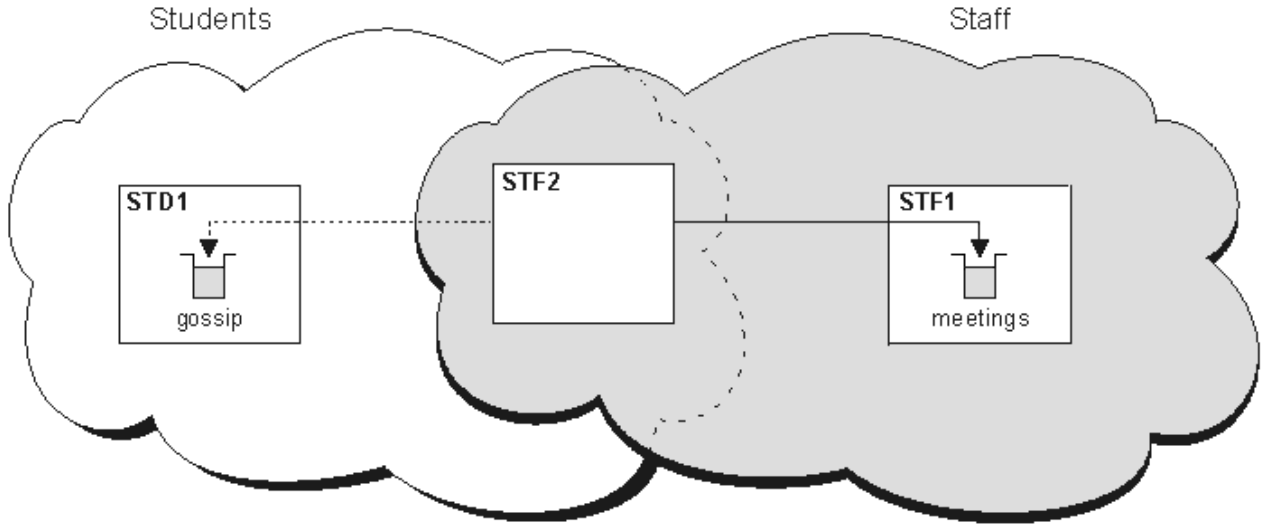
Ağınızda birden çok kümeniz varsa, bu kümeden farklı adlar vermelisiniz. Aynı adı taşıyan iki küme hiç birleştirilirse, bunları yeniden ayırmak mümkün değildir. Ayrıca, kümelere ve kanallara farklı adlar vermek de iyi bir fikir. Çıkışa DISPLAY komutlarından baktığınızda daha kolay ayırt edilirler. Kuyruk yöneticisi adlarının düzgün çalışabilmek için bir kümede benzersiz olması gerekir.

### **Hizmet sınıflarının tanımlanması**

Her bir öğrenci üyesi ve her öğrenci için kuyruk yöneticisi olan bir üniversite düşünün. personel üyeleri arasındaki mesajlar yüksek öncelikli ve yüksek bant genişliğine sahip kanallarda seyahat etmek. öğrenciler arasındaki mesajlar daha ucuz, daha yavaş kanallara seyahat etmek. Bu ağı geleneksel dağıtılmış kuyruklama tekniklerini kullanarak ayarlayabilirsiniz. IBM MQ , hedef kuyruk adı ve kuyruk yöneticisi adına bakarak hangi kanalların kullanılacağını seçer.

Çalışanları ve öğrencileri ayırt etmek için, kuyruk yöneticilerini Şekil 7 sayfa 36' ta gösterildiği gibi iki küme halinde gruplayabilirsiniz. IBM MQ , iletileri yalnızca o kümede tanımlı olan kanalların üzerinden

personel kümesindeki toplantı kuyruğuna taşır. Öğrenci kümesindeki dedikodu kuyruğuna ilişkin iletiler, o kümede tanımlı olan kanalların üzerinden gider ve uygun hizmet sınıfını alır.



Şekil 7. Hizmet sınıfları

### Kümeleme ipuçları

Kümelemeyi kullanmadan önce sistemlerinizde ya da uygulamalarınızda bazı değişiklikler yapmanız gerekebilir. dağıtık kuyruklama davranışından hem benzerlikler hem de farklar var.

- Küme kuyruklarına erişmeleri için, bir kümenin dışındaki kuyruk yöneticilerine el ile yapılandırma tanımlamaları eklemelisiniz.
- İki kümeyi aynı adla birleştirirseniz, bunları yeniden ayıramazsınız. Bu nedenle, tüm kümelere benzersiz bir ad verebilmek önerilir.
- Kuyruk yöneticisinde bir ileti gelirse, ancak bu iletiyi alacak bir kuyruk yoksa, ileti, ölü-mektup kuyruğuna konadır. Herhangi bir ölü harf kuyruğu yoksa, kanal başarısız olur ve yeniden dener. Ölü harf kuyruğun kullanımı, dağıtılmış kuyruğa alma ile aynıdır.
- Kalıcı iletilerin bütünlüğü sürdürülüdür. İletiler, kümelerin kullanılmasının sonucu olarak çoğaltılmaz ya da kaybolmaz.
- Kümelerin kullanılması sistem yönetimini azaltır. Kümeler, dağıtılmış kuyruklama kullanarak tasarlamayacaklarından çok daha fazla kuyruk yöneticisi ile daha büyük ağların birbirine bağlanmasını kolaylaştırır. Bir kümedeki her kuyruk yöneticisi arasında iletişimi etkinleştirmeye çalışırsanız, aşırı ağ kaynakları tüketebileceğiniz bir risk vardır.
- Kuyruk yöneticilerini ağaç yapısında sunan IBM MQ Explorer kullanıyorsanız, büyük kümeler için görünüm hantal olabilir.
- **Multi** Dağıtım listelerinin amacı, birden çok hedefe aynı iletiyi göndermek için tek bir MQPUT komutu kullanmaktır. Dağıtım listeleri IBM MQ for Multiplatforms üzerinde desteklenir. Dağıtım listelerini kuyruk yöneticisi kümelerine sahip olarak kullanabilirsiniz. Bir kümede, tüm iletiler MQPUT saatinde genişletilir. Bu avantaj, ağ trafiği açısından, kümeleme dışı bir ortamda olduğu kadar büyük değildir. Dağıtım listelerinin avantajı, çok sayıda kanal ve iletim kuyruklarının el ile tanımlanması gerekmemesine neden olur.
- İş yükünüzü dengelemek için kümeleri kullanabildiğinizde, uygulamalarınızı inceler. İletilerin belirli bir kuyruk yöneticisi tarafından mı, yoksa belirli bir sırada mı işlenmesini zorunlu kılacağını belirleyin. Bu tür uygulamaların mesaj afetlerine sahip olduğu söyleniyor. Karmaşık kümelerde bunları kullanabilmeniz için uygulamalarınızı değiştirmeniz gerekebilir.
- İletilerin belirli bir hedefe gönderilmesini sağlamak için MQOPEN üzerinde MQ00\_BIND\_ON\_OPEN seçeneğini kullanmayı seçebilirsiniz. Hedef kuyruk yöneticisi kullanılmıyorsa, kuyruk yöneticisi yeniden kullanılabilir duruma gelinceye kadar iletiler teslim edilmez. Yinelenme riski nedeniyle iletiler başka bir kuyruk yöneticisine yönlendirilmedi.

- Kuyruk yöneticisi bir küme havuzunu barıntacaksa, anasistem adını ya da IP adresini bilmeniz gerekir. You have to specify this information in the AD1 parameter when you make the CLUSSDR definition on other queue managers joining the cluster. DHCP kullanıyorsanız, bir sistemi her yeniden başlattığınızda DHCP yeni bir IP adresi ayarabileceği için, IP adresi değişebilir. Bu nedenle, CLUSSDR tanımlamalarında IP adresini belirtmemelisiniz. Tüm CLUSTSDR tanımlarında IP adresi yerine anasistem adı belirtilse bile, tanımlamalar yine de güvenilir olmaz. DHCP, anasisteme ilişkin DNS izin girişini yeni adrese güncelleyemez. Kuyruk yöneticilerini, DHCP kullanan sistemlerde tam havuz olarak aday göstermeniz gerekiyorsa, DNS dizininizi güncel tutmayı garanti eden yazılımı kurun.
- VTAM soysal kaynakları ya da Dynamic Domain Name Server (DDNS) soysal adları, kanallarınıza ilişkin bağlantı adları olarak kullanılır. Bunu yapabiliyorsanız, kanallarınız beklenenden farklı bir kuyruk yöneticisine bağlanabilirler.
- Bir iletiyi yalnızca yerel bir küme kuyruğundan alabilirsiniz, ancak bir kümedeki herhangi bir kuyruğa ileti koyabilirsiniz. If you open a queue to use the MQGET command, the queue manager opens the local queue.
- Yalın bir IBM MQ kümesi ayarladıysanız, uygulamalarınızdan herhangi birini değiştirmenize gerek yoktur. Uygulama, MQOPEN çağrısında hedef kuyruğun adını verebilir ve kuyruk yöneticisinin yerini bilmek zorunda değildir. İş yükü yönetimi için bir küme ayarlarsanız, uygulamalarınızı gözden geçirmeniz ve bunları gerektiği gibi değiştirmelisiniz.
- DISPLAY CHSTATUS ve DISPLAY QSTATUS **runmqsc** komutlarını kullanarak bir kanala ya da kuyruğa ilişkin yürürlükteki izleme ve durum verilerini görüntüleyebilirsiniz. İzleme bilgileri, sistemin performansını ve durumunu göstermenize yardımcı olmak için kullanılabilir. İzleme, kuyruk yöneticisi, kuyruk ve kanal öznitelikleri tarafından denetlenir. Monitoring of auto-defined cluster-sender channels is possible with the MONACLS queue manager attribute.

## İlgili kavramlar

### Kümeler

“Kümeleme ve dağıtılmış kuyruklama karşılaştırması” sayfa 29

Dağıtılmış kuyruğa alma ve kümeleme kullanarak kuyruk yöneticilerini bağlamak için tanımlanması gereken bileşenleri karşılaştırın.

### Bir kümenin bileşenleri

## İlgili görevler

### Kuyruk yöneticisi kümesinin yapılandırılması

### Yeni bir küme ayarlanıyor

## ***Kuyruk yöneticisi havuzları bilgileri ne kadar süreyle alıkoymuyor?***

Kuyruk yöneticisi havuzları bilgileri 30 gün boyunca saklar. Otomatik işlem, kullanılmakta olan bilgileri verimli bir şekilde yeniler.

Bir kuyruk yöneticisi kendisi hakkında bazı bilgiler gönderdiğinde, tam ve kısmi havuz kuyruğu yöneticileri bilgileri 30 gün boyunca saklar. Örneğin, bir kuyruk yöneticisi yeni bir kuyruğun yaratılmasına duyurulduğunda, bilgi dışarı gönderilir. Bu bilgilerin süresi dolmasına engel olmak için kuyruk yöneticileri, 27 gün sonra kendi hakkındaki tüm bilgileri otomatik olarak yeniden göndermektedir. Kısmi bir havuz, 30 günlük yaşam süresi boyunca bilgi için yeni bir istek gönderirse, son 30 gün süre bitimi kalır.

Bilgilerin süresi dolduğunda, havuzdan hemen kaldırılmaz. Bunun yerine 60 günlük bir ek süre için tutulmaktadır. Ek süre içinde güncelleme alınmazsa, bilgiler kaldırılır. Ek süre, bir kuyruk yöneticisinin süre bitimi tarihinde geçici olarak hizmet dışı kalabileceği gerçemesine olanak tanır. Kuyruk yöneticisi 90 günden uzun bir süre kümeden kesilirse, kümenin bir parçası olmayı durdurur. Ancak, ağla yeniden bağlanırsa, yeniden kümenin bir parçası olur. Tüm havuzlar, diğer kuyruk yöneticilerinden gelen yeni istekleri karşılamak için süresi dolmuş bilgileri kullanmaz.

Benzer şekilde, bir kuyruk yöneticisi dolu bir havuzdan güncel bilgiler için bir istek gönderdiğinde, istek 30 gün sürer. After 27 days IBM MQ checks the request. 27 gün içinde başvurulduysa, otomatik olarak yenilenir. Yoksa, kullanım süresi sona erer ve kuyruk yöneticisi tarafından yeniden güncellenir ve yeniden gereklidir. Süresi dolan istekler, atıl kuyruk yöneticilerinden bilgi için isteklerin birikmesini önler.

**Not:** Büyük kümeler için, birçok kuyruk yöneticisi aynı anda kendileriyle ilgili tüm bilgileri otomatik olarak yeniden gönderirse kesintiye yol açar. Bkz. Büyük bir kümede yenilenme, kümenin performansını ve kullanılabilirliğini etkileyebilir.

### İlgili kavramlar

“Kümeleme: REFRESH CLUSTER en iyi uygulamaları kullanma” sayfa 67

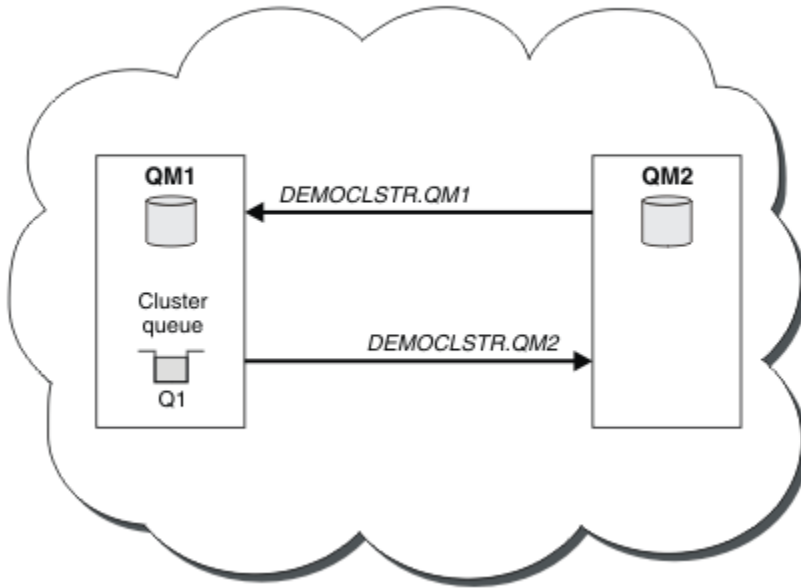
Bir kümeye ilişkin olarak yerel olarak tutulan tüm bilgileri atmak ve bu bilgileri kümedeki tüm havuzlardan yeniden oluşturmak için **REFRESH CLUSTER** komutunu kullanıyorsunuz. Bu komutu, istisnai koşullar dışında, kullanmanıza gerek yoktur. Bunu kullanmanız gerekiyorsa, bunu nasıl kullanmanız gerektiği konusunda dikkat edilmesi gereken noktalar vardır. Bu bilgiler, müşterilerden gelen test ve geri bildirimlere dayalı bir kılavuzdur.

### Örnek kümeler


İlk örnekte, olası en küçük iki kuyruk yöneticisi kümesi gösterilmektedir. İkinci ve üçüncü örneklerde, üç kuyruk yöneticisi kümesinin iki sürümü gösterilir.

Olası en küçük küme, yalnızca iki kuyruk yöneticisi içerir. Bu durumda, her iki kuyruk yöneticisi de tam havuz içerir. Kümeyi ayarlamak için yalnızca birkaç tanımlamaya ihtiyacınız var, ancak her kuyruk yöneticisinde yüksek düzeyde bir özerklik söz sahibi.

### DEMOCLSTR



Şekil 8. İki kuyruk yöneticilerinden oluşan küçük bir küme

- Kuyruk yöneticilerine LONDON ve NEWYORK gibi uzun adlar olabilir.  IBM MQ for z/OS üzerinde, kuyruk yöneticisi adları dört karakterle sınırlıdır.
- Her kuyruk yöneticisi tipik olarak ayrı bir makinede yapılandırılır. Ancak, aynı makinede birden çok kuyruk yöneticisi olabilir.

Benzer bir örnek küme ayarına ilişkin yönergeler için bkz. Yeni bir küme ayarlama.

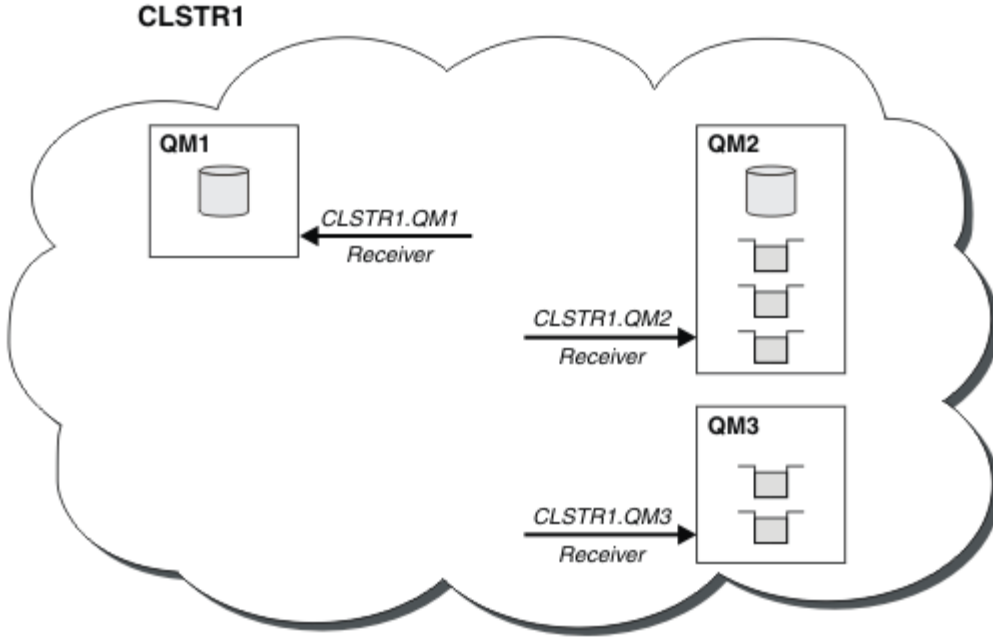
Şekil 9 sayfa 39 , CLSTR1adlı bir kümenin bileşenlerini gösterir.

- Bu kümede, üç kuyruk yöneticisi, QM1, QM2 ve QM3 vardır.
- QM1 ve QM2 anasistem havuzları, kümedeki tüm kuyruk yöneticilerine ve kümeyle ilgili nesnelere ilişkin bilgi içerir. Bunlar, *tam havuz kuyruğu yöneticileri* olarak adlandırılır. Havuzlar, çizgede gölgeli silindirlere tarafından temsil edilir.
- QM2 ve QM3 , kümedeki diğer kuyruk yöneticilerine erişilebilen bazı kuyruklar içerir. Kümedeki başka bir kuyruk yöneticisinde erişilebilir olan kuyruklara *küme kuyrukları* adı verilir. Küme kuyrukları, çizgede

gölgeli kuyruklar tarafından temsil edilir. Küme kuyruklarına kümenin herhangi bir yerinden erişilebilir. IBM MQ kümeleme kodu, küme kuyruklarına ilişkin uzak kuyruk tanımlarının, bunlara gönderme yapan herhangi bir kuyruk yöneticiliğinde yaratılmasını sağlar.

Dağıtılmış kuyruğa alma işlemi sırasında, bir uygulama, kümedeki herhangi bir kuyruk yöneticisinde bir küme kuyruğuna ileti koymak için MQPUT çağrısını kullanır. Bir uygulama, yalnızca kuyruğun bulunduğu kuyruk yöneticisinde bulunan bir küme kuyruğundan iletileri almak için MQGET çağrısını kullanır.

- Her kuyruk yöneticisinin, iletileri alabileceği *cluster\_name . queue\_manager\_name* adlı bir kanalın giriş sonu için el ile yaratılmış bir tanımlama vardır. Alıcı kuyruk yöneticisinde, *cluster\_name . queue\_manager\_name* bir küme-alıcı kanalıdır. Küme-alıcı kanalı, dağıtılmış kuyruksa kullanılan bir alıcı kanalı gibidir; kuyruk yöneticisine ilişkin iletileri alır. Buna ek olarak, küme hakkında da bilgi alır.



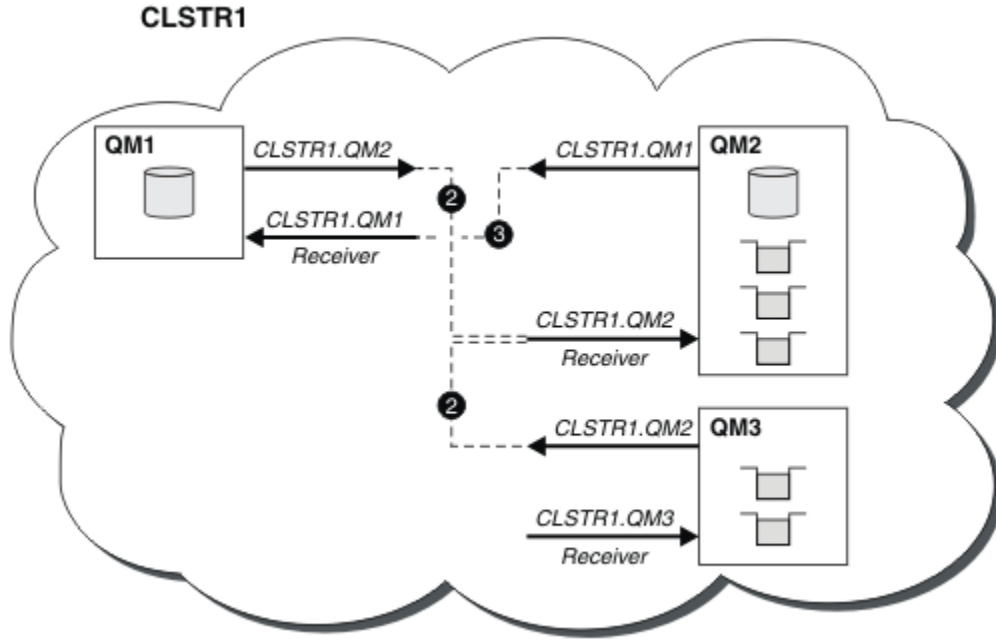
Şekil 9. Bir kuyruk yöneticisi kümesi

- Şekil 10 sayfa 40 ' ta, her kuyruk yöneticisinin bir kanala gönderme bitişi için de bir tanımlaması vardır. Bu, tam havuz kuyruğu yöneticilerinden birinin küme alıcı kanalına bağlanır. Gönderme kuyruk yöneticisinde, *cluster\_name . queue\_manager\_name* bir küme-gönderici kanalıdır. QM1 ve QM3 , CLSTR1 . QM2'a bağlanan küme gönderici kanallarına sahiptir, noktalı çizgi "2"' ye bakın.

QM2 , CLSTR1 . QM1 ile bağlantı kuran bir küme gönderici kanalına sahiptir, noktalı çizgi "3"' e bakın. Küme-gönderici kanalı, dağıtılmış kuyruksa kullanılan bir gönderen kanalı gibidir; bu kanal, alıcı kuyruk yöneticisine ileti gönderir. Buna ek olarak, küme hakkında da bilgi gönderir.

Hem kümeleme alıcısı sonu hem de bir kanalın küme gönderen sonu tanımlandıktan sonra kanal otomatik olarak başlatılır.





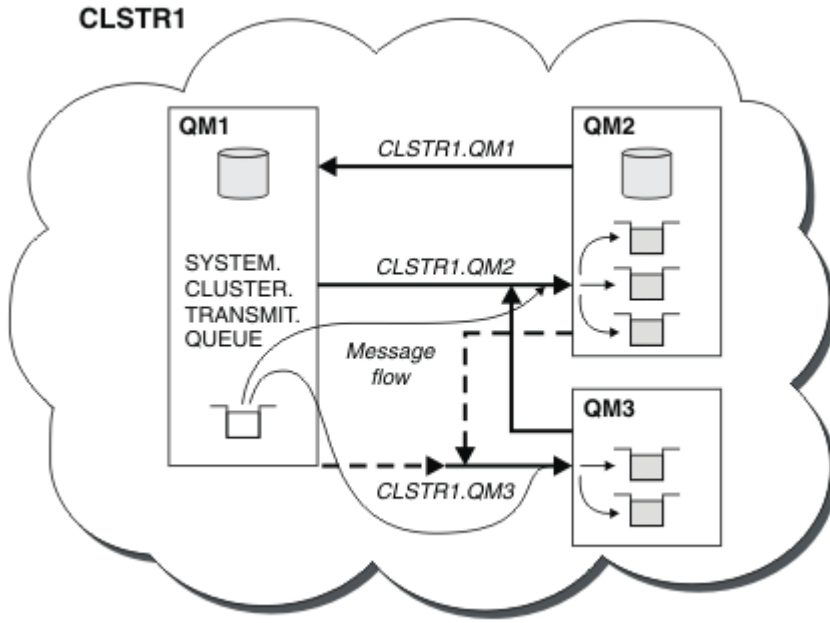
Şekil 10. Gönderen kanallarına sahip bir kuyruk yöneticisi kümesi

Yerel kuyruk yöneticisinde bir küme gönderen kanalı tanımlanırken, kuyruk yöneticisi tam havuz kuyruğu yöneticilerinden birine tanıtılır. Tam havuz kuyruk yöneticisi, bilgileri tam olarak tam havuzunda güncelleştirir. Daha sonra, otomatik olarak özgün kuyruk yöneticisine bir küme gönderici kanalı yaratır ve kümeyle ilgili kuyruk yöneticisi bilgilerini gönderir. Bu nedenle, bir kuyruk yöneticisi bir küme hakkında bilgi alır ve bir küme kuyruk yöneticisi hakkında bilgi alır.

Şekil 9 sayfa 39' da yeniden bakın. Suppose that an application connected to queue manager QM3 wants to send some messages to the queues at QM2. QM3 ' in bu kuyruklara ilk erişmesi gereken ilk kez, tam bir havuza danışılarak bunları keşfeder. The full repository in this case is QM2, which is accessed using the sender channel CLSTR1.QM2. Havuzdaki bilgilerle bu kuyruklar için otomatik olarak uzak tanımlar yaratılabilir. If the queues are on QM1, this mechanism still works, because QM2 is a full repository. Tam havuzun, kümedeki tüm nesnelere tam bir kaydı vardır. Bu ikinci durumda, QM3 otomatik olarak, QM1 üzerindeki küme alıcı kanalına karşılık gelen bir küme gönderici kanalı yaratır ve bu kanal iki arasında doğrudan iletişim sağlar.

Şekil 11 sayfa 41 , otomatik olarak oluşturulan iki kümeli gönderici kanalına sahip aynı kümeyi gösterir. The cluster-sender channels are represented by the two dashed lines that join with the cluster-receiver channel CLSTR1.QM3. It also shows the cluster transmission queue, SYSTEM.CLUSTER.TRANSMIT.QUEUE, which QM1 uses to send its messages. Kümedeki tüm kuyruk yöneticilerinin, aynı kümedeki başka bir kuyruk yöneticisine ileti gönderebildikleri bir küme iletim kuyruğuna sahip olması gerekir.





Şekil 11. Otomatik tanımlı kanalları gösteren bir kuyruk yöneticisi kümesi

**Not:** Diğer çizgeler, yalnızca el ile tanımlamalar yaptığınız kanalların giriş uçlarını gösterir. Gönderme uçları, gerektiğinde otomatik olarak tanımlandığından atlanır. En çok kümeli gönderen kanalların otomatik tanımı, kümelerin işlevi ve verimliliği açısından çok önemlidir.

### İlgili kavramlar

[“Kümeleme ve dağıtılmış kuyruklama karşılaştırması” sayfa 29](#)

Dağıtılmış kuyruğa alma ve kümeleme kullanarak kuyruk yöneticilerini bağlamak için tanımlanması gereken bileşenleri karşılaştırın.

[Bir kümenin bileşenleri](#)

### İlgili görevler

[Kuyruk yöneticisi kümesinin yapılandırılması](#)

[Yeni bir küme ayarlanıyor](#)

### **Kümeleme: En iyi uygulamalar**

Kümeleme, kuyruk yöneticilerine bağlanmak için bir mekanizma sağlar. Bu bölümde açıklanan en iyi uygulamalar, müşterilerden gelen test ve geri bildirimlere dayanır.

Başarılı bir küme kurulumu, iyi bir planlama ve iyi uygulama yönetimi ve ağ tasarımı gibi temel IBM MQ temel kavramlarıyla ilgili ayrıntılı bir anlayışla bağlıdır. Devam etmeden önce ilgili konulardaki bilgileri aşına olduğunuzu doğrulayın.

### İlgili kavramlar

[Dağıtılmış kuyruğa alma ve kümeler](#)

[Kümeler](#)

### İlgili görevler

[“Kümelerin tasarlanması” sayfa 23](#)

Kümeler, hem ilk yapılandırmayı hem de devam eden yönetimi basitleştiren bir şekilde birbirine bağlanan kuyruk yöneticileri için bir mekanizma sağlar. Kümelerin doğru çalıştığından ve gerekli kullanılabilirlik ve yanıt verebilirlik düzeylerine ulaşabildiğinden emin olmak için kümeler dikkatli bir şekilde tasarlanmalıdır.

[İzleme kümeleri](#)

*Kümeleme: çakışan kümeler için dikkat edilmesi gereken özel noktalar*

Bu konuda, IBM MQ kümelerini planlamaya ve yönetmeye ilişkin yönergeler sağlanır. Bu bilgiler, müşterilerden gelen testlere ve geribildirime dayalı bir kılavuzdur.

## Küme sahipliği

Aşağıdaki bilgileri okumadan önce çakışan kümeleri tanıyın. Gerekli bilgiler için [“Çakışan kümeler” sayfa 35 ve Kümeler arasında ileti yollarını yapılandırma başlıklı konuya](#) bakın.

Çakışan kümelerden oluşan bir sistemi yapılandırırken ve yönetirken, en iyisi aşağıdakilere uymaktır:

- IBM MQ kümeleri daha önce açıklandığı gibi 'gevşek bir şekilde eşlenmiş' olsalar da, bir kümeyi tek bir yönetim birimi olarak kabul etmek yararlıdır. Bu kavram, tek tek kuyruk yöneticilerindeki tanımlamalar arasındaki etkileşim, kümenin düzgün çalışması için kritik önem taşıdığından kullanılır. Örneğin: İş yükü dengeli küme kuyrukları kullanılırken, tek bir yöneticinin ya da ekibin, küme genelinde dağıtılan tanımlamalara bağlı olarak iletilere ilişkin olası hedeflerin tam kümesini anlaması önemlidir. Daha önemsiz olarak, küme gönderen/alıcı kanal çiftleri boyunca uyumlu olmalıdır.
- Bu önceki kavram göz önünde bulundurulur; birden çok kümenin (ayrı ekipler/kişiler tarafından yönetileceği) bir araya geldiği durumlarda, ağ geçidi kuyruk yöneticilerinin yerinde denetim yönetiminde net ilkelere sahip olmak önemlidir.
- Çakışan kümeleri tek bir ad alanı olarak kabul etmek yararlıdır: Kanal adları ve kuyruk yöneticisi adları tek bir küme boyunca benzersiz olmalıdır. Yönetim, topolojinin tamamında benzersiz olduğunda çok daha kolaydır. Uygun bir adlandırma kuralına uymanız en iyi yoldur; olası kurallar [“Küme adlandırma kuralları” sayfa 33](#) içinde açıklanmıştır.
- Bazen idari ve sistem yönetimi işbirliği çok önemlidir/önlenebilir: Örneğin, örtüşmesi gereken farklı kümelerle sahip kuruluşlar arasında işbirliği. Kimin neye sahip olduğuna ilişkin net bir anlayış ve uygulanabilir kurallar/kurallar, kümelemenin çakışan kümeler sırasında sorunsuz bir şekilde çalışmasına yardımcı olur.

## Çakışan kümeler: Ağ Geçitleri

Genel olarak, tek bir kümeyi yönetmek birden çok kümeye göre daha kolaydır. Bu nedenle çok sayıda küçük küme oluşturmak (örneğin her uygulama için bir tane) genel olarak kaçınılması gereken bir şeydir.

Ancak, hizmet sınıfları sağlamak için çakışan kümeler uygulayabilirsiniz. Örneğin:

- Daha küçük olanın Yayınlama/Abone Olma için olduğu eş merkezli kümeleriniz varsa. Ek bilgi için [Sistemleri nasıl boyutlandırılır](#) başlıklı konuya bakın.
- Bazı kuyruk yöneticileri farklı ekipler tarafından yönetiliyorsa. Daha fazla bilgi için önceki [“Küme sahipliği” sayfa 42](#) bölümüne bakın.
- Bir organizasyonel ya da coğrafi bakış açısından mantıklı ise.
- Eşdeğer kümeler ad çözümlemesi ile çalışıyorsa, örneğin var olan bir kümede TLS uygulanıyor.

Örtüşen kümelerden güvenlik avantajı yoktur; iki farklı ekip tarafından yönetilen kümelerin örtüşmesine izin verir, topolojinin yanı sıra ekiplere de etkili bir şekilde katılır. Herhangi biri:

- Böyle bir kümede duyurulan ad, diğer küme tarafından erişilebilir.
- Bir kümede duyurulan ad, uygun mesajları çekmek için diğerinde duyurulebilir.
- Ağ geçidinin bitişiğindeki bir kuyruk yöneticisinde duyurulmayan nesne, ağ geçidinin üyesi olduğu kümelerden çözümlenebilir.

Ad alanı, her iki kümenin birleşmesi ve tek bir ad alanı olarak ele alınmalıdır. Bu nedenle, çakışan bir kümenin sahipliği her iki kümenin tüm yöneticileri arasında paylaşılır.

Bir sistem birden çok küme içerdiğinde, iletileri bir kümedeki kuyruk yöneticilerinden başka bir kümedeki kuyruk yöneticilerine yönlendirme gereksinimi olabilir. Bu durumda, birden çok kümenin bir şekilde birbirine bağlanması gerekir: İzlenmesi gereken iyi bir kalıp, kümeler arasında ağ geçidi kuyruk yöneticilerinin kullanılmasıdır. Bu düzenleme, noktadan noktaya iletişim kanallarının yönetilmesi zor bir ağ


oluşturulmasını önler ve güvenlik politikaları gibi sorunları yönetmek için iyi bir yer sağlar. Bu düzenlemeyi gerçekleştirmenin iki farklı yolu vardır:

1. İkinci bir küme alıcı tanımlaması kullanarak her iki kümeye bir (ya da daha çok) kuyruk yöneticisi yerleştirin. Bu düzenleme daha az yönetim tanımı içerir, ancak daha önce belirtildiği gibi, çakışan bir kümenin sahipliğinin her iki kümenin tüm yöneticileri arasında paylaşıldığı anlamına gelir.
2. Küme 1 'deki bir kuyruk yöneticisini, geleneksel noktadan noktaya iletişim kanallarını kullanarak küme 2 'deki bir kuyruk yöneticisiyle eşleştirin.

Bu iki durumda da, trafiği uygun şekilde yönlendirmek için çeşitli araçlar kullanılabilir. Özellikle, kuyruk ya da kuyruk yöneticisi diğer adları diğer kümeye yönelmek için kullanılabilir ve boş **RQMNAME** özelliğine sahip bir kuyruk yöneticisi diğer adı, iş yükü dengelemesini istendiği yerde yeniden yönlendirir.

## İlgili kavramlar

[“Küme adlandırma kuralları” sayfa 33](#)

Kuyruk yöneticisinin ait olduğu kümeyi tanıtan bir adlandırma kuralı kullanarak aynı kümedeki kuyruk yöneticilerini adlandırmayı düşünün. Kanal adları için benzer bir adlandırma kuralı kullanın ve kanal özelliklerini açıklayacak şekilde genişletin.  z/OS üzerinde küme alıcı kanalları tanımlarken sosyal bağlantı kullanmayın.

*Kümeleme: Topoloji tasarımıyla ilgili dikkat edilecek noktalar*

Bu konuda, IBM MQ kümelerini planlamaya ve yönetmeye ilişkin yönergeler sağlanır. Bu bilgiler, müşterilerden gelen testlere ve geribildirime dayalı bir kılavuздur.

Kullanıcı uygulamalarının ve iç idari süreçlerin önceden nerede bulunacağı düşünülüp, birçok sorun ya önlenabilir ya da daha sonraki bir tarihte en aza indirilebilir. Bu konu, performansı artırabilen tasarım kararlarıyla ilgili bilgileri içerir ve küme ölçeklendirdiğiniz bakım görevlerini basitleştirir.

- [“Kümeleme altyapısının performansı” sayfa 43](#)
- [“Tam havuzlar” sayfa 44](#)
- [“Uygulamalar tam havuzlardaki kuyrukları kullanmalı mı?” sayfa 45](#)
- [“Kanal tanımlarının yönetilmesi” sayfa 45](#)
- [“Birden çok kanal üzerinden iş yükü dengeleme” sayfa 45](#)

## Kümeleme altyapısının performansı

Bir uygulama kümedeki bir kuyruk yöneticisinde bir kuyruk açmaya çalışıldığında, kuyruk yöneticisi, kuyruğun kümede nerede bulunduğunu öğrenmesi için ilgili kuyruğun tüm havuzlarına ilgisini kaydeder. Kuyruk konumu ya da yapılandırmasında yapılan güncellemeler, tüm havuzlar tarafından otomatik olarak ilgili kuyruk yöneticisine gönderilir. İlginizi çeken bu kayıt, dahili olarak abonelik olarak bilinir (bu abonelikler, IBM MQ içinde yayınlama/abone olma ileti sistemi için kullanılan IBM MQ abonelikleriyle aynı değildir)

Bir kümeyle ilgili tüm bilgiler her tam havuzdan geçer. Bu nedenle, tam havuzlar her zaman yönetimle ilgili ileti trafiği için bir kümede kullanılır. Bu abonelikleri yönetirken sistem kaynaklarının yüksek kullanımı ve bunların iletimi ve sonuçta ortaya çıkan yapılandırma iletileri, kümeleme altyapısında önemli bir yüke neden olabilir. Bu yükün anlaşılmasını ve mümkün olan yerlerde en aza indirilmesini sağlarken göz önünde bulundurulması gereken birçok şey vardır:

- Küme kuyruğunu kullanan tek tek kuyruk yöneticileri ne kadar çok olursa, sistemde o kadar çok abonelik olur ve bu nedenle, değişiklikler oluştuğunda ve ilgili abonelere, özellikle de tam havuz kuyruğu yöneticilerine ilişkin bildirim gönderilmesi gereken yönetim ek yükü o kadar büyük olur. Gereksiz trafiği ve tam havuz yükünü en aza indirmenin bir yolu, benzer uygulamaları (yani aynı kuyruklarla çalışan uygulamaları) daha az sayıda kuyruk yöneticilerine bağlamaktır.
- Sistemdeki abonelik sayısının başarımı etkilemesine ek olarak, kümelenmiş nesnelere konfigürasyonundaki değişiklik hızı başarımı etkileyebilir; örneğin, kümelenmiş bir kuyruk konfigürasyonunun sık sık değiştirilmesi gibi.

- Bir kuyruk yöneticisi birden çok kümenin üyesiye (yani, çakışan bir küme sisteminin parçasıysa), aynı kuyruk yöneticileri birden çok kümenin tam havuzları olsa bile, bir kuyrukta yapılan herhangi bir ilgi, üyesi olduğu her küme için bir abonelikte sonuçlanır. Bu düzenleme, sistemdeki yükü artırır ve tek bir küme yerine birden çok çakışan kümenin gerekli olup olmadığını düşünmek için bir nedendir.
- Uygulama iletisi trafiği (yani, IBM MQ uygulamaları tarafından küme kuyruklarına gönderilen ileteler), hedef kuyruk yöneticilerine ulaşmak için tam havuzlar üzerinden gitmez. Bu ileti trafiği, doğrudan iletinin kümeye girdiği kuyruk yöneticisi ile küme kuyruğunun bulunduğu kuyruk yöneticisi arasında gönderilir. Bu nedenle, tam havuz kuyruğu yöneticileri söz konusu iki kuyruk yöneticisinden biri olmadığı sürece, tam havuz kuyruğu yöneticilerine göre uygulama ileti trafiğinin yüksek oranlarına uyulması gerekli değildir. Bu nedenle, kümeleme altyapısı yükünün önemli olduğu kümelerdeki uygulama iletisi trafiği için tam havuz kuyruğu yöneticilerinin kullanılmaması önerilir.

## Tam havuzlar

Havuz, bir kümenin üyesi olan kuyruk yöneticilerine ilişkin bilgilerin bir toplamıdır. Kümedeki her kuyruk yöneticisine ilişkin eksiksiz bir bilgi kümesini barındıran bir kuyruk yöneticisi tam bir havuza sahiptir. Tam havuzlar ve kısmi havuzlar hakkında daha fazla bilgi için bkz. [Küme havuzu](#).

Güvenilir ve yüksek düzeyde kullanılabilir olan sunucularda tam havuzlar tutulmalıdır ve tek hata noktalarından kaçınılmalıdır. Küme tasarımının her zaman iki tam havuzu olmalıdır. Tam havuzda bir hata oluşursa, küme çalışmaya devam edebilir.

Kümedeki bir kuyruk yöneticisi tarafından yapılan küme kaynaklarında yapılan güncellemelerin ayrıntıları; örneğin, kümelenecek kuyruklar, o kümedeki en çok iki tam havuza (ya da kümede yalnızca bir tam havuz kuyruk yöneticisi varsa) gönderilir. Bu tam havuzlar, bilgileri tutar ve kümede ilgili olan herhangi bir kuyruk yöneticisine geçirir (yani, ona abone olurlar). Kümenin her üyesinin küme kaynaklarının güncel bir görünümüne sahip olmasını sağlamak için, her kuyruk yöneticisinin bir kerede en az bir tam havuz kuyruk yöneticisiyle iletişim kurabilmesi gerekir.

Herhangi bir nedenle, bir kuyruk yöneticisi herhangi bir tam havuzla iletişim kuramıyorsa, bir dönem için önceden önbelleğe alınmış bilgi düzeyine dayalı olarak kümede çalışmaya devam edebilir, ancak daha önce kullanılmamış küme kaynaklarına yeni güncelleme ya da erişim sağlanmaz.

Bu nedenle, iki tam havuzu her zaman kullanılabilir tutmayı hedeflemelisiniz. Bununla birlikte, bu düzenleme, kümenin tam bir havuz olmadan kısa bir süre için yeterli şekilde işlev göstermesi nedeniyle aşırı önlemlerin alınması gerektiği anlamına gelmez.

Bir kümenin, küme bilgilerinin kullanılabilirliği dışında iki tam havuz kuyruğu yöneticisinin olması gerekmesinin başka bir nedeni vardır: Bu neden, kurtarma amacıyla, tam havuz önbelleğinde tutulan küme bilgilerinin iki yerde bulunduğundan emin olmaktır. Yalnızca bir tam havuz varsa ve kümeyle ilgili bilgilerini kaybederse, kümenin yeniden çalışması için küme içindeki tüm kuyruk yöneticilerine el ile müdahale gerekir. Bununla birlikte, iki tam havuz varsa, bilgiler her zaman iki tam havuza yayınlanır ve bunlara abone olduğundan, başarısız olan tam havuz en az çabayla kurtarılabilir.

- Bu kümenin kullanıcılarını etkilemeden, iki tam havuz kümesi tasarımında tam havuz kuyruğu yöneticilerine bakım gerçekleştirilebilir: Küme, yalnızca bir havuzla çalışmaya devam eder; böylece, mümkün olan yerlerde havuzları aşağı indirir, bakım uygulamaları ve her defasında bir tane yedekler. İkinci tam havuzda bir kesinti olsa bile, çalışan uygulamalar en az üç gün boyunca etkilenmez.
- Coğrafi nedenlerle coğrafi olarak yerel bir tam havuz kullanmak gibi üçüncü bir havuz kullanmak için iyi bir neden yoksa, iki havuz tasarımını kullanın. Üç tam havuz olması, şu anda kullanımda olan ikisinin hangilerinin kullanıldığını hiçbir zaman bilmediğiniz anlamına gelir ve birden çok iş yükü yönetimi parametresi arasındaki etkileşimlerin neden olduğu yönetimle ilgili sorunlar olabilir. İkide fazla tam havuz olması önerilmez.
- Hala daha iyi kullanılabilirliğe gereksiniminiz varsa, tüm havuz kuyruğu yöneticilerini çok eşgörünümlü kuyruk yöneticileri olarak barındırmayı ya da kullanılabilirliği artırmak için platforma özgü yüksek kullanılabilirlik desteğini kullanmayı düşünün.
- El ile tanımlanan küme gönderen kanallarıyla tüm havuz kuyruğu yöneticilerini tam olarak birbirine bağlamanız gerekir. Küme, haklı bir nedenle ikiden fazla tam havuza sahip olduğunda özellikle dikkatli olunmalıdır. Bu durumda genellikle bir veya daha fazla kanalı gözden kaçırmak ve hemen anlaşılması

mümkündür. Tam ara bağlantı gerçekleşmediğinde, sorunların tanınması genellikle zordur. Bazı tam havuzlar tüm havuz verilerini tutmadığından ve bu nedenle bağlı oldukları tam havuzlara bağlı olarak kümedeki kuyruk yöneticilerinin farklı görünümlere sahip olmalarına neden olduğundan, bu havuzların tanınması zordur.

## Uygulamalar tam havuzlardaki kuyrukları kullanmalı mı?

Tam havuz çoğu şekilde diğer kuyruk yöneticilerine benzer ve bu nedenle uygulama kuyruklarını tam havuzda barındırmak ve uygulamaları doğrudan bu kuyruk yöneticilerine bağlamak mümkündür. Uygulamalar tam havuzlardaki kuyrukları kullanmalı mı?

Yaygın olarak kabul edilen cevap "Hayır?" dir. Bu yapılandırma mümkün olsa da, birçok müşteri bu kuyruk yöneticilerini tam havuz kümesi önbelleğinin bakımı için özel olarak ayırmayı tercih eder. Her iki seçeneğe karar verirken dikkate alınması gereken noktalar burada açıklanmıştır, ancak sonuçta küme mimarisi ortamın belirli taleplerine uygun olmalıdır.

- Yükseltmeler: Genellikle, IBM MQ ' un yeni yayınlarında yeni küme özelliklerini kullanmak için öncelikle o kümenin tam havuz kuyruğu yöneticilerinin yükseltilmesi gerekir. Kümedeki bir uygulama yeni özellikler kullanmak istediğinde, birlikte bulunan bir dizi uygulamayı test etmeden tam havuzları (ve kısmi havuzların bazı alt kümelerini) güncellemek yararlı olabilir.
- Bakım: Benzer bir şekilde, tam havuzlara acil bakım uygulamanız gerekiyorsa, bunlar **REFRESH** komutuyla uygulamalara dokunmadan yeniden başlatılabilir ya da yenilenebilir.
- Performans: Kümeler büyüdükçe ve tam havuz kümesi önbelleği bakımına ilişkin talepler büyüdükçe, uygulamaların ayrı tutulması, sistem kaynakları için çekişme yoluyla uygulama performansını etkileyen riski azaltır.
- Donanım gereksinimleri: Tipik olarak, tam havuzların güçlü olması gerekmez; örneğin, iyi bir kullanılabilirlik beklentisi olan basit bir UNIX sunucusu yeterlidir. Alternatif olarak, çok büyük veya sürekli değişen kümeler için, tam havuz bilgisayarının performansı dikkate alınmalıdır.
- Yazılım gereksinimleri: Gereksinimler genellikle uygulama kuyruklarını tam bir havuzda barındırmak için ana nedendir. Küçük bir kümede, iş birliği (collocation), tüm kuyruk yöneticilerinin/sunucularının daha az olması için bir gereksinim anlamına gelebilir.

## Kanal tanımlarının yönetilmesi

Tek bir küme içinde bile, iki kuyruk yöneticisi arasında birden çok rota sağlayan birden çok kanal tanımı olabilir.

Bazen tek bir küme içinde paralel kanallara sahip olmanın bir avantajı vardır, ancak bu tasarım kararı iyice düşünülmelidir; karmaşıklık eklemenin yanı sıra, bu tasarım, performansı azaltan kanalların az kullanılmasıyla sonuçlanabilir. Bu durum, testlerin genellikle sabit bir hızda çok sayıda ileti göndermeyi içermesi nedeniyle ortaya çıkar, bu nedenle paralel kanallar tamamen kullanılır. Ancak sürekli olmayan ileti akışının gerçek dünya koşullarıyla, iş yükü dengeleme algoritması, ileti akışı kanaldan kanala geçildikçe performansın düşmesine neden olur.

Bir kuyruk yöneticisi birden çok kümenin üyesiye, her küme için ayrı bir CLUSRCVR kanalı tanımlamak yerine küme ad listesi olan tek bir kanal tanımı kullanma seçeneği vardır. Ancak, bu kurulum daha sonra yönetim sorunlarına neden olabilir; örneğin, TLS ' nin bir kümeye uygulanacağı, ancak bir saniye uygulanmadığı durumu göz önünde bulundurun. Bu nedenle ayrı tanımlar yaratılması tercih edilir ve "Küme adlandırma kuralları" sayfa 33 içinde önerilen adlandırma kuralı bunu destekler.

## Birden çok kanal üzerinden iş yükü dengeleme

Bu bilgi, konunun gelişmiş bir şekilde anlaşılması amacıyla hazırlanmıştır. Bu konuya ilişkin temel açıklamalar için (buradaki bilgileri kullanmadan önce anlaşılması gereken) bkz. [İş yükü yönetimi için kümeleri kullanma, kümelerde iş yükü dengeleme](#) [Küme iş yükü yönetimi algoritması](#).

Küme iş yükü yönetimi algoritması büyük bir araç kümesi sağlar, ancak bunların tümü nasıl çalıştıklarını ve etkileşimde bulduklarını tam olarak anlamadan birbirleriyle birlikte kullanılmamalıdır. İş yükü dengeleme süreci için kanalların ne kadar önemli olduğu hemen belli olmayabilir: İş yükü yönetimi

çevrimsel sıralı algoritması, kümelenmiş bir kuyruğu olan bir kuyruk yöneticisine birden çok küme kanalı gibi davranır ve bu kuyruğun birden çok eşgörünümü olarak kabul edilir. Bu işlem aşağıdaki örnekte daha ayrıntılı olarak açıklanmıştır:

1. Bir kümede kuyruk barındıran iki kuyruk yöneticisi vardır: QM1 ve QM2.
2. QM1 için beş küme alıcı kanalı vardır.
3. QM2 için yalnızca bir küme alıcı kanalı vardır.
4. QM3 üzerinde **MQPOT** ya da **MQOPEN** bir eşgörünüm seçtiğinde, algoritmanın iletiyi QM1 'e gönderme olasılığı QM2' e göre beş kat daha fazladır.
5. 4 numaralı adımdaki durum, algoritma beş kanaldan QM1 'e ve tek kanaldan QM2' e kadar altı seçeneği (5 + 1) ve yuvarlamalı robinler arasından seçim yapabileceği için ortaya çıkar.

Diğer bir ince davranış, iletileri yerel kuyruk yöneticisinde tek bir yönetim ortamının yapılandırılmış olduğu kümelenmiş bir kuyruğa koyarken bile, IBM MQ iletilerin kuyruğun yerel eşgörünümüne mi, yoksa kuyruğun uzak eşgörünümlerine mi konacağına karar vermek için yerel küme alıcı kanalının durumunu kullanmasıdır. Bu senaryoda:

1. İş yükü yönetimi algoritmasının tek tek küme kuyruklarına bakmadığı iletiler konurken, bu hedeflere ulaşabilen küme kanallarına bakar.
2. Yerel hedeflere ulaşmak için, yerel alıcı kanalları bu listeye eklenir (ileti göndermek için kullanılmamalarına rağmen).
3. Bir yerel alıcı kanal durdurulduğunda, iş yükü yönetimi algoritması, CLUSRCVR durdurulmamışsa varsayılan olarak alternatif bir örneği tercih eder. Hedef için birden çok yerel CLUSRCVR yönetim ortamı varsa ve en az biri durdurulmazsa, yerel yönetim ortamı uygun kalır.

*Kümeleme: Birden çok küme iletim kuyruğu kullanılarak uygulama yalıtımı*

Bir kümedeki kuyruk yöneticileri arasındaki ileti akışlarını yalıtabilirsiniz. Farklı küme gönderici kanalları tarafından, farklı küme iletim kuyruklarına iletilmekte olan iletileri yerleştirebilirsiniz. Yaklaşımı tek bir kümede ya da çakışan kümelerle kullanabilirsiniz. Bu konuda, kullanabileceğiniz bir yaklaşım seçiminde size yol göstermesi için bazı örnekler ve bazı en iyi uygulamalar sağlanır.

Bir uygulamayı konuşlandırdığınızda, hangi IBM MQ kaynaklarının diğer uygulamalarla paylaştığı ve hangi kaynakların paylaşmadığı bir seçeneğiniz vardır. Paylaşılabilecek kaynakların sayısı, sunucunun kendisi, kuyruk yöneticisi, kanal ve kuyruklar olmak üzere birçok kaynak tipi vardır. Uygulamaları daha az paylaşılan kaynakla yapılandırmayı seçebilir; ayrı kuyruklar, kanallar, kuyruk yöneticileri ya da hatta tek tek uygulamalar için sunucular yerleştirebilirsiniz. Bunu yapmazsanız, genel sistem yapılandırması daha büyük ve daha karmaşık hale gelir. IBM MQ kümelerin kullanılması, daha fazla sunucunun, kuyruk yöneticilerinin, kuyrukların ve kanalların yönetilmesine ilişkin karmaşıklığı azaltır, ancak başka bir paylaşılan kaynak, küme iletim kuyruğu, SYSTEM . CLUSTER . TRANSMIT . QUEUE tanıtılır.

Şekil 12 sayfa 47 is a slice out of a large IBM MQ deployment that illustrates the significance of sharing SYSTEM . CLUSTER . TRANSMIT . QUEUE. In the diagram, the application, Client App, is connected to the queue manager QM2 in cluster CL1. A message from Client App is processed by the application, Server App. The message is retrieved by Server App from the cluster queue Q1 on the queue manager QM3 in CLUSTER2. İstemci ve sunucu uygulamaları aynı kümede olmadığından, ileti ağ geçidi kuyruk yöneticisi QM1 tarafından aktarılır.

Bir küme ağ geçidini yapılandırmanın olağan yolu, ağ geçidi kuyruk yöneticisinin tüm kümelerin bir üyesini yapmasıdır. Ağ geçidi kuyruk yöneticisinde, tüm kümelerdeki küme kuyrukları için kümelenmiş diğer ad kuyrukları tanımlanır. Kümelenmiş kuyruk diğer adları tüm kümelerde kullanılabilir. Küme kuyruğu diğer adlarına gönderilen iletiler, ağ geçidi kuyruk yöneticisi aracılığıyla doğru hedefleriyle yönlendirilir. Ağ geçidi kuyruk yöneticisi, kümelenmiş diğer ad kuyruklarına gönderilen iletileri QM1 üzerindeki ortak SYSTEM . CLUSTER . TRANSMIT . QUEUE ' e yerleştirir.

Merkez ve konuşan mimari, kümeler arasındaki tüm iletilerin ağ geçidi kuyruk yöneticisinden geçilmesini gerektirir. Sonuç olarak, tüm iletiler QM1, SYSTEM . CLUSTER . TRANSMIT . QUEUE üzerindeki tek küme iletim kuyruğundan geçer.

Başarım perspektifinden, tek bir kuyruk sorun değildir. Genel olarak yaygın bir iletim kuyruğu, performans darboğazını göstermez. Ağ geçidindeki ileti verimi büyük ölçüde, ona bağlanan kanalların performansı ile



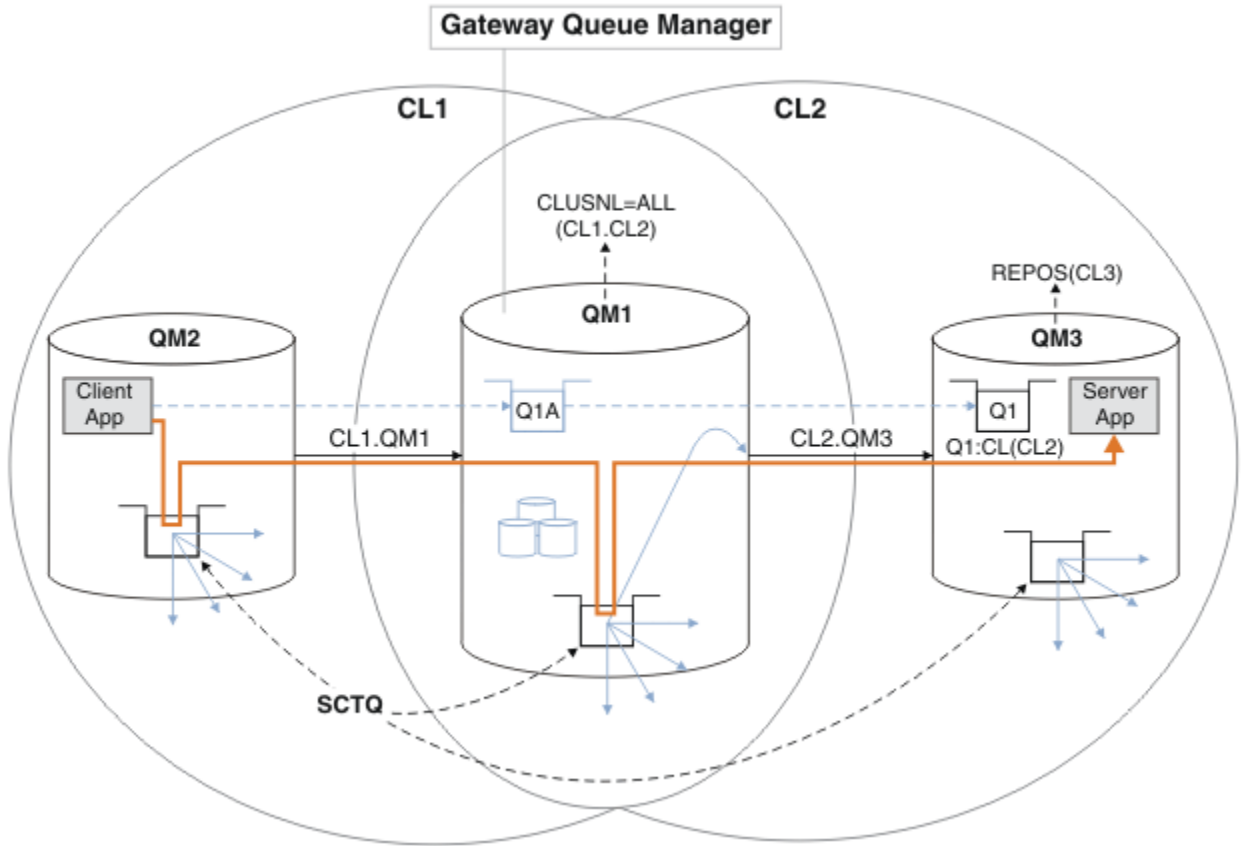
belirlenir. Üretilen iş genellikle kuyruk sayısından ya da kanalların kullandığı kuyruklardaki ileti sayısından etkilenmez.

Diğer bazı perspektiflerden, birden çok uygulama için tek bir iletim kuyruğunu kullanarak sakıncaları geri gelir:

- İletilerin akışını, ileti akışından başka bir hedefe ayıramazsınız. İletilerin farklı kuyruk yöneticilerindeki farklı kümelerde olması durumunda bile, iletilerin iletilmeden önce depolarını ayıramazsınız.

Bir küme hedefi kullanılamaz duruma gelirse, tek iletim kuyruğunda o hedef birikimine ilişkin iletiler ve sonunda iletiler bu bilgileri doldurur. İletim kuyruğu dolunca, iletilerin herhangi bir küme hedefi için iletim kuyruğuna konması durdurulur.

- İletilerin farklı küme hedeflerine aktarılmalarını izlemek kolay değildir. Tüm iletiler tek iletim kuyruğunda yer alıyor. İletim kuyruğunun derinliğinin görüntülenmesi, iletilerin tüm hedeflere aktarılıp aktarılmadığına ilişkin az bilgi verir.



**Not:** Şekil 12 sayfa 47 içindeki oklar ve aşağıdaki rakamlar farklı tiplerdir. Kesiksiz oklar ileti akışlarını gösterir. Kesiksiz oklardaki etiketler, ileti kanalı adlarıdır. The gray solid arrows are potential message flows from the SYSTEM.CLUSTER.TRANSMIT.QUEUE onto cluster-sender channels. Siyah çizgili çizgiler, etiketleri hedefleriyle ilişkilendirir. Gray dashed arrows are references; for example from an MQOPEN call by Client App to the cluster alias queue definition Q1A.

Şekil 12. Client-server application deployed to hub and spoke architecture using IBM MQ clusters

Şekil 12 sayfa 47'ta, Server App'un istemcileri kuyruğu Q1A'yi açar. Messages are put to SYSTEM.CLUSTER.TRANSMIT.QUEUE on QM2, transferred to SYSTEM.CLUSTER.TRANSMIT.QUEUE on QM1, and then transferred to Q1 on QM3, where they are received by the Server App application.

The message from Client App passes through system cluster transmission queues on QM2 and QM1. Şekil 12 sayfa 47'inde, hedef ağ geçidi kuyruk yöneticisinde ileti akışını istemci uygulamasından yalıtmak ve böylece iletilerin SYSTEM.CLUSTER.TRANSMIT.QUEUE' ta saklanmaması gerekir. Diğer kümelenmiş kuyruk yöneticilerinden herhangi birindeki akışları yalıtabilirsiniz. Akışları, diğer yöndeki istemciye geri

giden diğer yönde de yalıtabilirsiniz. Çözümlerin açıklamalarını kısa tutmak için, açıklamalar istemci uygulamasından yalnızca tek bir akış olarak değerlendirir.

## **Küme ileti trafiğini küme ağ geçidi kuyruk yöneticisinde yalıtmanın çözümleri**

Sorunu çözenin bir yolu, kümeler arasında köprü yapmak için kuyruk yöneticisi diğer adlarını ya da uzak kuyruk tanımlamalarını kullanmandır. Ağ geçidi kuyruk yöneticisindeki her bir ileti akışını ayırmak için kümelenmiş bir uzak kuyruk tanımlaması, bir iletim kuyruğu ve bir kanal yaratın; bkz. [Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtma için uzak kuyruk tanımlaması eklenmesi](#).

From IBM WebSphere MQ 7.5 onwards, cluster queue managers are not limited to a single cluster transmission queue. İki seçeneğiniz vardır:

1. Ek küme iletim kuyruklarını el ile tanımlayın ve her bir iletim kuyruğundan hangi küme gönderici kanallarının iletileri aktarmasını tanımlayın; bkz. [Ağ geçidi kuyruk yöneticisinden gönderilen küme ileti trafiğini yalıtma için bir küme iletim kuyruğu eklenmesi](#).
2. Kuyruk yöneticisinin otomatik olarak ek küme iletim kuyrukları yaratmasına ve yönetmesine izin verin. Her bir küme gönderici kanalı için farklı bir küme iletim kuyruğu tanımlar; bkz. [İleti trafiğini yalıtma için varsayılan değeri, küme iletim kuyruklarını ayırmak için değiştirme](#).

Bazı küme gönderici kanalları için el ile tanımlanmış küme iletim kuyruklarını, gerisini kuyruk yöneticisi yöneten kuyruk yöneticisiyle birleştirebilirsiniz. İletim kuyruklarının birleşimi, [Ağ geçidi kuyruk yöneticisinden gönderilen küme ileti trafiğini yalıtma için bir küme iletim kuyruğu eklenmesi](#) başlıklı konu ile ilgili yaklaşımın yaklaşımını sağlar. Bu çözümde, kümeler arasındaki çoğu ileti ortak SYSTEM . CLUSTER . TRANSMIT . QUEUE' ı kullanır. Bir uygulama kritik önem taşıyor ve tüm ileti akışları, el ile tanımlanmış bir küme iletim kuyruğu kullanılarak diğer akışlardan yalıtılıyor.

[Bir ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtma için küme iletim kuyruğu eklenmesi](#) yapılarındaki yapıları sınırlıdır. İleti trafiğini, aynı kümedeki bir küme kuyruğuna, başka bir küme kuyruğunda aynı kuyruk yöneticisinde ayırmaz. Dağıtılmış kuyruğun bir parçası olan uzak kuyruk tanımlamalarını kullanarak, ileti trafiğini tek tek kuyruklara ayırabilirsiniz. Kümelerle, birden çok küme iletim kuyruğu kullanarak, farklı küme gönderen kanallarına giden ileti trafiğini birbirinden ayırabilirsiniz. Aynı kümedeki birden çok küme kuyruğu, aynı kuyruk yöneticisinde, bir küme gönderen kanalı paylaşır. Bu kuyruklara ilişkin iletiler, ağ geçidi kuyruk yöneticisinden iletilmeden önce, aynı iletim kuyruğunda saklanır. [Ağ geçidi kuyruk yöneticisinden gönderilen küme ileti trafiğini yalıtma için bir küme ve bir küme iletim kuyruğu eklenmesi](#) içindeki yapılandırmada, başka bir küme eklenerek ve kuyruk yöneticisini ve küme kuyruğunu yeni kümenin bir üyesi haline getirerek sınırlandırma yan yana olur. Yeni kuyruk yöneticisi, kümedeki tek kuyruk yöneticisi olabilir. Kümeye daha fazla kuyruk yöneticisi ekleyebilir ve aynı kümeyi kullanarak, bu kuyruk yöneticilerindeki küme kuyruklarını yalıtabilirsiniz.

### **İlgili kavramlar**

[“Erişim denetimi ve birden çok küme iletim kuyruğu” sayfa 28](#)

Bir uygulama uzak küme kuyruklarına ileti yerleştirdiğinde üç denetleme kipi arasında seçim yapın. Kipler, küme kuyruğunda uzaktan denetlenir, SYSTEM . CLUSTER . TRANSMIT . QUEUE' a yerel olarak denetlenir ya da küme kuyruğu ya da küme kuyruğu yöneticisi için yerel tanımlara karşı denetlenir.

[Küme iletim kuyruklarıyla ve kümeyle gönderici kanallarla çalışma](#)

[“Çakışan kümeler” sayfa 35](#)

Çakışan kümeler, ek yönetim yetenekleri sağlar. Çakışan kümeleri denetlemek için gereken komut sayısını azaltmak için ad listelerini kullanın.

### **İlgili görevler**

[Uzak küme kuyruklarına ileti koyma yetkisi](#)

[Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtma için uzak kuyruk tanımlaması eklenmesi](#)

[Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtma için bir küme iletim kuyruğu ekleme](#)

[Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtma için bir küme ve bir küme iletim kuyruğu ekleme](#)

[İleti trafiğini yalıtma için varsayılan olarak küme iletim kuyrukları için varsayılan olarak değiştirilmesi](#)

[Ağ geçidi kuyruk yöneticisiyle çakışan iki küme oluşturma](#)



Kümeler arasında ileti yollarının yapılandırılması

güvenlik

**İlgili başvurular**

setmqaut

*Kümeleme: Küme iletim kuyruklarının nasıl yapılandırılacağı planlanması*

Küme iletim kuyrukları seçimleriyle yönlendirildiniz. Ortak bir varsayılan kuyruk, ayrı varsayılan kuyruklar ya da el ile tanımlanmış kuyruklar yapılandırabilirsiniz.

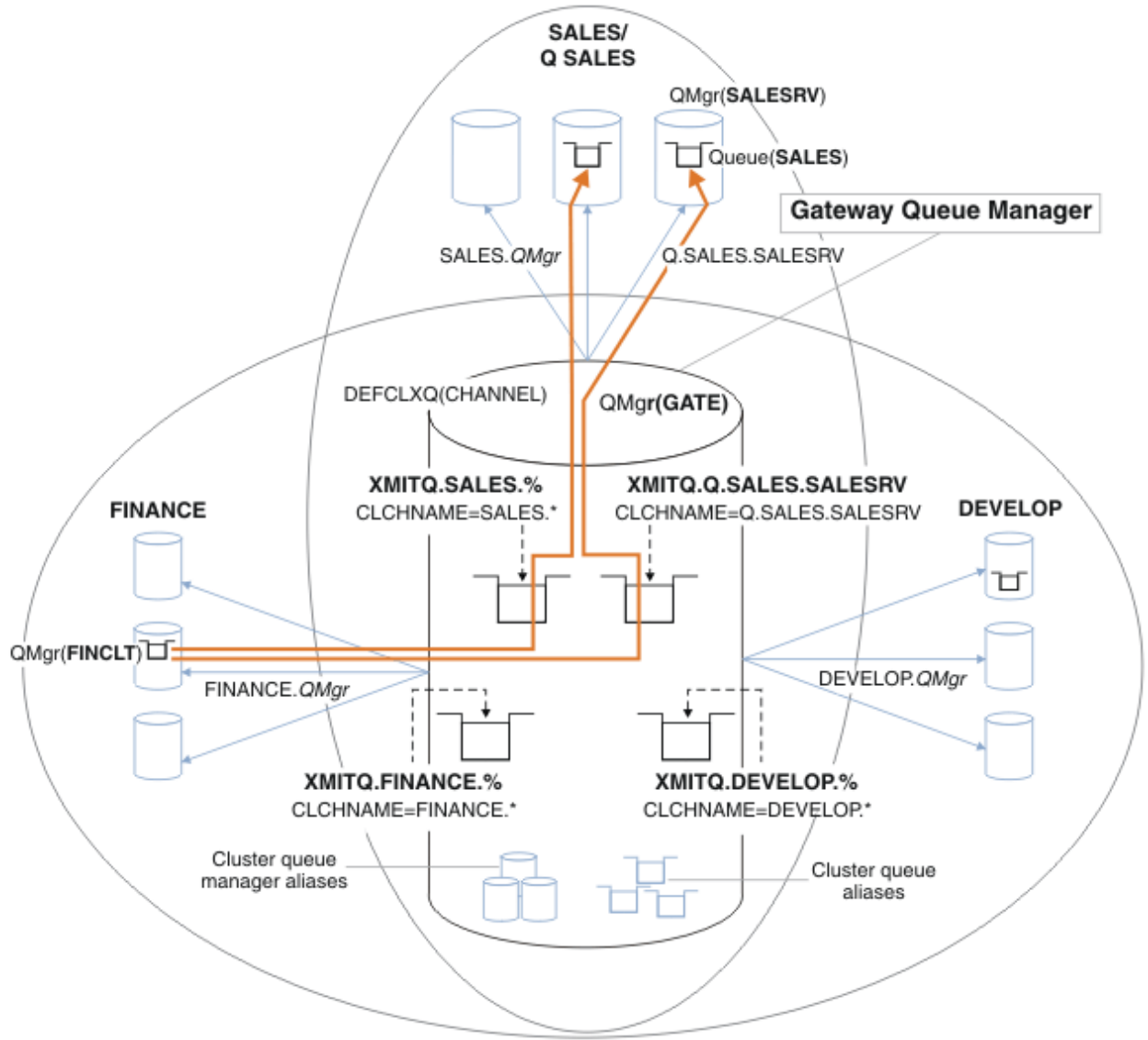
## **Başlamadan önce**

Review "Kullanılacak küme iletim kuyruğunun tipini nasıl seçecek?" sayfa 51.

## **Bu görev hakkında**

Bir kuyruk yöneticisini bir küme iletim kuyruğu seçmek için nasıl yapılandıracağını planlarken yapmanız gereken bazı seçiminiz vardır.

1. Küme iletimi aktarımları için varsayılan küme iletim kuyruğu nedir?
  - a. Ortak bir küme iletim kuyruğu, SYSTEM . CLUSTER . TRANSMIT . QUEUE.
  - b. Ayrı küme iletim kuyrukları. Kuyruk yöneticisi, ayrı küme iletim kuyruklarını yönetir. Bu, bunları model kuyruğundan kalıcı dinamik kuyruklar olarak yaratır, SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE. Bu, kullandığı her bir küme gönderici kanalı için bir küme iletim kuyruğu oluşturur.
2. El ile yaratmaya karar verdiğiniz küme iletim kuyrukları için ek iki seçeneğiniz vardır:
  - a. El ile yapılandırma kararı verdiğiniz her bir küme gönderici kanalı için ayrı bir iletim kuyruğu tanımlayın. Bu durumda, iletim kuyruğunun **CLCHNAME** kuyruk özniteliğini, küme gönderen kanalının adına ayarlayın. Bu iletim kuyruğundan iletileri aktarmak için, küme gönderici kanalını seçin.
  - b. Bir grup kümeli gönderici kanalının ileti trafiğini aynı küme iletim kuyruğunda birleştirin; bkz. Şekil 13 sayfa 50. Bu durumda, her ortak iletim kuyruğunun **CLCHNAME** kuyruk özniteliğini soysal bir küme gönderen kanalı adına ayarlayın. Soysal bir kümeyle gönderici kanal adı, küme gönderen kanal adlarına süzgeç uygulanacak bir süzgeçtir. Örneğin, SALES . \* , adları SALES . ile başlayan tüm küme gönderici kanallarını gruplarla gruplardır. Süzgeç dizesinin herhangi bir yerinde birden çok genel arama karakteri yerleştirebilirsiniz. Genel arama karakteri bir yıldız işaretidir ( "\*"). Sıfır ile herhangi bir sayıda karakter temsil eder.



Şekil 13. Farklı bölüm IBM MQ kümeleri için belirli iletim kuyrukları örneği

## Yordam

1. Kullanmak için varsayılan küme iletim kuyruğu tipini seçin.

- Tek bir küme iletim kuyruğu ya da her küme bağlantısı için ayrı kuyruklar seçin.

Varsayılan ayarı bırakın ya da **MQSC** komutunu çalıştırın:

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. Küme iletim kuyruğunu diğer akışlarla paylaşmaması gereken ileti akışlarını yalıtın.

- Bkz. “Kümeleme: Birden çok küme iletim kuyruklarının yapılandırılması” sayfa 53. Örnek olarak, yalıtılması gereken SALES kuyruğu, SALESRV üzerinde SALES kümesinin bir üyesidir. To isolate the SALES queue, create a new cluster Q.SALES, make the SALESRV queue manager a member, and modify the SALES queue to belong to Q.SALES.
- SALES ' a ileti gönderen kuyruk yöneticileri de yeni kümenin üyesi olmalıdır. Örnekte olduğu gibi, kümelmiş bir kuyruk diğer adı ve bir ağ geçidi kuyruk yöneticisi kullanıyorsanız, birçok durumda, ağ geçidi kuyruk yöneticisini yeni kümenin bir üyesi yapmak için gereken değişiklikleri sınırlandırabilirsiniz.

- Ancak, ağ geçitinden hedefe akan akışlar, akışları kaynak kuyruk yöneticisinden ayrı olarak ayırmaz. ancak bazen ağ geçitten ayrı akışlar ayırmak ve ağ geçidine akmak için yeterli olduğu ortaya çıkıyor. Yeterli değilse, kaynak kuyruk yöneticisini yeni kümeye ekleyin. İletilerin ağ geçidi üzerinden seyahat etmesini istiyorsanız, küme diğer adını yeni kümeye taşıyın ve doğrudan hedef kuyruk yöneticisine değil, ağ geçidindeki küme diğer adlarına ileti göndermeye devam edin.

İleti akışlarını yalıtım için aşağıdaki adımları izleyin:

- a) Akışlar hedeflerini yapılandırın; böylece, her hedef kuyruk belirli bir kümedeki kuyrukta, o kuyruk yöneticisindequeuekuyrukta yer alan tek kuyruk olur.
  - b) Sistematik bir adlandırma kuralından sonra yarattığınız yeni kümeler için küme gönderici ve küme alıcı kanallarını oluşturun.
    - Bkz. [“Kümeleme: çakışan kümeler için dikkat edilmesi gereken özel noktalar” sayfa 42.](#)
  - c) Hedef kuyruğa ileti gönderen her kuyruk yöneticisindeki her yalıtılmış hedef için bir küme iletim kuyruğu tanımlayın.
    - Küme iletim kuyrukları için bir adlandırma kuralı, küme kanalı adı özniteliğinin ( CLCHNAME) değerini XMITQ . ile önekli olarak kullanmaktadır.
3. Yönetişim ya da izleme gereksinimlerini karşılamak için küme iletim kuyrukları yaratın.
- Tipik yönetim ve izleme gereksinimleri, küme başına iletim kuyruğunda ya da kuyruk yöneticisi başına iletim kuyruğu ile sonuçlanır. If you follow the naming convention for cluster channels, *ClusterName* . *QueueManagerName*, it is easy to create generic channel names that select a cluster of queue managers, or all the clusters a queue manager is a member of; see [“Kümeleme: Birden çok küme iletim kuyruklarının yapılandırılması” sayfa 53.](#)
  - Sosyal kanal adları için, küme iletim kuyrukları için adlandırma kuralını, yıldız işaretini bir yüzde işaretiyle değiştirerek, sosyal kanal adları için genişletir. Örneğin,

```
DEFINE QLOCAL(XMITQ.SALES.%) USAGE(XMITQ) CLCHNAME(SALES.*)
```

### İlgili kavramlar

[Küme iletim kuyruklarıyla ve kümeyle gönderici kanallarla çalışma](#)

[“Erişim denetimi ve birden çok küme iletim kuyruğu” sayfa 28](#)

Bir uygulama uzak küme kuyruklarına ileti yerleştirdiğinde üç denetleme kipi arasında seçim yapın. Kipler, küme kuyruğunda uzaktan denetlenir, SYSTEM . CLUSTER . TRANSMIT . QUEUE' a yerel olarak denetlenir ya da küme kuyruğu ya da küme kuyruğu yöneticisi için yerel tanıtlara karşı denetlenir.

[“Çakışan kümeler” sayfa 35](#)

Çakışan kümeler, ek yönetim yetenekleri sağlar. Çakışan kümeleri denetlemek için gereken komut sayısını azaltmak için ad listelerini kullanın.

### İlgili görevler

[Ağ geçidi kuyruk yöneticisinden gönderilen iletileri yalıtım için uzak kuyruk tanımlaması eklenmesi](#)

[Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtım için bir küme iletim kuyruğu ekleme](#)

[Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtım için bir küme ve bir küme iletim kuyruğu ekleme](#)

[İleti trafiğini yalıtım için varsayılan olarak küme iletim kuyrukları için varsayılan olarak değiştirilmesi](#)

[Ağ geçidi kuyruk yöneticisiyle çakışan iki küme oluşturma](#)

[Kümeler arasında ileti yollarının yapılandırılması](#)

*Kullanılacak küme iletim kuyruğunun tipini nasıl seçecek?*

Farklı küme iletim kuyruğu yapılandırma seçenekleri arasında seçim yapabilirsiniz.

IBM WebSphere MQ 7.5 ' den başlayarak, bir küme gönderici kanalıyla hangi küme iletim kuyruğunu ilişkilendirdiğini seçebilirsiniz.

1. Tek varsayılan küme iletim kuyruğu olan SYSTEM . CLUSTER . TRANSMIT . QUEUE ile ilişkili tüm küme gönderici kanallarına sahip olabilirsiniz. This option is the default, and is the only choice for queue managers that run IBM WebSphere MQ 7.1 or earlier.
2. Tüm küme gönderici kanallarını otomatik olarak ayrı bir küme iletim kuyruğuyla ilişkilendirebilirsiniz. Kuyruklar, kuyruk yöneticisi tarafından SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE model kuyruğundan yaratılır ve SYSTEM . CLUSTER . TRANSMIT . ChannelName adlı kuyruğa alır. Kuyruk yöneticisi özniteliği DEFCLXQ KANAL olarak ayarlandıysa, kanallar benzersiz olarak adlandırılan küme iletim kuyruğunu kullanır.



**Uyarı:** If you are using dedicated SYSTEM . CLUSTER . TRANSMIT . QUEUEs with a queue manager that was upgraded from a version of the product earlier than IBM WebSphere MQ 7.5, ensure that the SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE has the SHARE/NOSHARE option set to **SHARE**.

3. Tek bir küme iletim kuyruğu tarafından hizmet vermek için küme gönderici kanalları belirleyebilirsiniz. Bir iletim kuyruğu yaratarak ve **CLCHNAME** özniteliğini, küme gönderen kanalının adına ayarlayarak bu seçeneği belirleyin.
4. Tek bir küme iletim kuyruğu tarafından hizmet vermek için küme gönderici kanallarının gruplarını seçebilirsiniz. Bir iletim kuyruğu yaratarak ve **CLCHNAME** özniteliğini *ClusterName.\** gibi soysal bir kanal adına ayarlayarak bu seçeneği belirleyin. If you name cluster channels by following the naming conventions in “Kümeleme: çakışan kümeler için dikkat edilmesi gereken özel noktalar” sayfa 42, this name selects all cluster channels connected to queue managers in the cluster *ClusterName*.

Bazı küme gönderenler kanallarına ilişkin varsayılan küme iletim kuyruğu seçeneklerinden herhangi birini, herhangi bir sayıda belirli ve soysal küme iletim kuyruğu yapılandırmasıyla birleştirebilirsiniz.

## En iyi uygulamalar

Çoğu durumda, var olan IBM MQ kuruluşları için varsayılan yapılandırma en iyi seçenektir. Küme kuyruk yöneticisi, küme iletilerini tek bir küme iletim kuyruğunda saklar, SYSTEM . CLUSTER . TRANSMIT . QUEUE. Farklı kuyruk yöneticilerine ve ayrı iletim kuyruklarında farklı kümelere ilişkin iletileri saklamak ya da kendi iletim kuyruklarınızı tanımlamak için varsayılan değeri değiştirme seçeneğiniz vardır.

Çoğu durumda, yeni IBM MQ kuruluşları için varsayılan yapılandırma da en iyi seçenektir. Varsayılan yapılandırmadan, her bir küme gönderici kanalı için tek bir iletim kuyruğuna sahip olmanın alternatif varsayılan yapılandırmasından geçiş işlemi otomatik olarak işlenmektedir. Geri geçilmesi de otomatidir. Bir ya da diğerinin seçimi kritik değildir, bunu tersine çevirebilirsiniz.

Farklı bir yapılandırmanın seçilmesinin nedeni, yönetim ve yönetimle ilgili işlevlerden ya da performansa göre daha çok iş yapmasıdır. Birkaç kural dışı durumla, birden çok küme iletim kuyruklarının yapılandırılması, kuyruk yöneticisinin davranışına yararlanmamaktadır. Daha fazla kuyruğa sonuç alır ve önceden ayarladığınız izleme ve yönetim yordamlarıyla tek iletim kuyruğuna başvuruda bulunmanızı gerektirir. Bu nedenle, farklı bir seçim için güçlü bir yönetim veya yönetim nedeniniz yoksa, varsayılan yapılandırma ile birlikte kalan en iyi seçenek budur.

The exceptions are both concerned with what happens if the number of messages stored on SYSTEM . CLUSTER . TRANSMIT . QUEUE increases. Başka bir hedefe ilişkin iletilerden bir hedefe ilişkin iletileri ayırmak için her adımı atsanız, bir hedefle kanal ve teslim sorunları başka bir hedefe teslim edilmeyi etkilemez. However, the number of messages stored on SYSTEM . CLUSTER . TRANSMIT . QUEUE can increase due to not delivering messages fast enough to one destination. Bir hedef için SYSTEM . CLUSTER . TRANSMIT . QUEUE üzerindeki iletilerin sayısı, iletilerin diğer hedeflere teslimi olumsuz etkileyebilir.

Tek bir iletim kuyruğunu doldurmaktan kaynaklanan sorunları önlemek için, yapılandırmanızı kullanarak yeterli kapasite oluşturmayı hedefleyebilirsiniz. Daha sonra, bir hedef başarısız olursa ve bir ileti birikimi oluşturulmaya başlarsa, sorunu çözecek zamanınız olur.

İletiler, küme ağ geçidi gibi bir göbek kuyruk yöneticisiyle yönlendirilmişse, ortak bir iletim kuyruğunu paylaşırlar. SYSTEM . CLUSTER . TRANSMIT . QUEUE. Ağ geçidi kuyruk yöneticisiyle ilgili SYSTEM . CLUSTER . TRANSMIT . QUEUE üzerinde saklanan ileti sayısı derinlik üst sınırına ulaşırsa, kuyruk yöneticisi iletim kuyruğu için derinliği azalana kadar yeni iletileri reddetmeye başlar. Tıkanıklık, ağ

geçidinden yönlendirilen tüm hedeflere ilişkin iletileri etkiler. İletiler, ağ geçidine ileti gönderen diğer kuyruk yöneticilerinin iletim kuyruklarını yedeklemektedir. Bu sorun, kuyruk yöneticisi hata günlüklerine, düşen ileti verisine ve iletiye bir iletinin varış noktasına geldiği süre arasında geçen zaman arasında yazılan iletilerde kendisini bildirmektedir.

Sıkışıklığın tek bir iletim kuyruğunda etkisi, tam olmadan önce bile belirginleşebilir. Bazı büyük ve kalıcı olmayan iletiler ve bazı küçük iletiler ile karışık ileti trafiği varsa, iletim kuyruğu doldururken küçük iletiler teslim etme süresi artar. Gecikme süresi, normalde diske yazılmayan, kalıcı olmayan büyük iletilerin diske yazılması nedeniyle ortaya çıkar. Zaman kritik ileti akışlarına sahipseniz, bir küme iletim kuyruğunu diğer karışık ileti akışlarıyla paylaşırsanız, bu iletiyi diğer ileti akışlarından yalıtım için özel bir ileti yolu tanımlamaya değer olabilir; bkz. [Bir ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtım için küme ve küme iletim kuyruğu eklenmesi.](#)

Ayrı küme iletim kuyruklarının yapılandırılmasına ilişkin diğer nedenler, yönetim gereksinimlerini karşılamakta ya da farklı küme hedeflerine gönderilen izleme iletilerini basitleştirmektedir. Örneğin, bir hedef için iletilerin başka bir hedef için ileti içeren bir iletim kuyruğunu hiçbir zaman paylaşmadığını göstermek zorunda kalabilirsiniz.

Varsayılan küme iletim kuyruğunu denetleyen **DEFCLXQ** kuyruk yöneticisi özniteliğini, her bir küme gönderici kanalı için farklı küme iletim kuyrukları yaratmak üzere değiştirin. Birden çok hedef, bir küme gönderici kanalını paylaşabilir, bu nedenle kümelerin bu amacı tam olarak karşılaması için planlamanız gerekir. [Ağ geçidi kuyruk yöneticisinden gönderilen küme iletileri trafiğini yalıtım için bir küme ve bir küme iletim kuyruğu ekleme yöntemini tüm küme kuyruklarınıza sistematik olarak uygulayın.](#) Hedeflediğiniz sonuç, bir küme gönderici kanalını başka bir küme hedefiyle paylaşmak için herhangi bir küme hedefi olmayışıdır. Sonuç olarak, bir küme hedefine ilişkin hiçbir ileti, başka bir hedef için bir iletiyle küme iletim kuyruğunu paylaşır.

Belirli bir ileti akışı için ayrı bir küme iletim kuyruğu yaratılması, bu hedefe yönelik ileti akışının izlenmesini kolaylaştırır. Yeni bir küme iletim kuyruğu kullanmak için, kuyruğu tanımlayın, bir kümeyle ilişkilendiren bir kanalla ilişkilendirin ve kanalı durdurup başlatın. Değişiklik kalıcı olmak zorunda değil. Bir süre için ileti akışını yalıtıp iletim kuyruğunu izleyebilir ve varsayılan iletim kuyruğunu yeniden kullanmaya geri dönebilirsiniz.

### **İlgili görevler**

**Kümeleme: Birden çok küme iletim kuyruklarının yapılandırılması**

Bu görevde, birden çok küme iletim kuyruklarını üç üst üste çakışan kümeye planlamak için adımları uygulayın. Gereksinimler, diğer tüm ileti akışlarından bir küme kuyruğuna ve farklı küme iletim kuyruklarında farklı kümelere ilişkin iletileri saklamak için, iletileri ayrı ayrı bir küme kuyruğuna yollamak içindir.

**Kümeleme: Küme iletim kuyruklarının değiştirilmesi**

Var olan bir üretim kuyruğu yöneticisinin küme iletim kuyruklarına ilişkin değişikliklerin yürürlüğe nasıl getirileceğini planlayın.

*Kümeleme: Birden çok küme iletim kuyruklarının yapılandırılması*

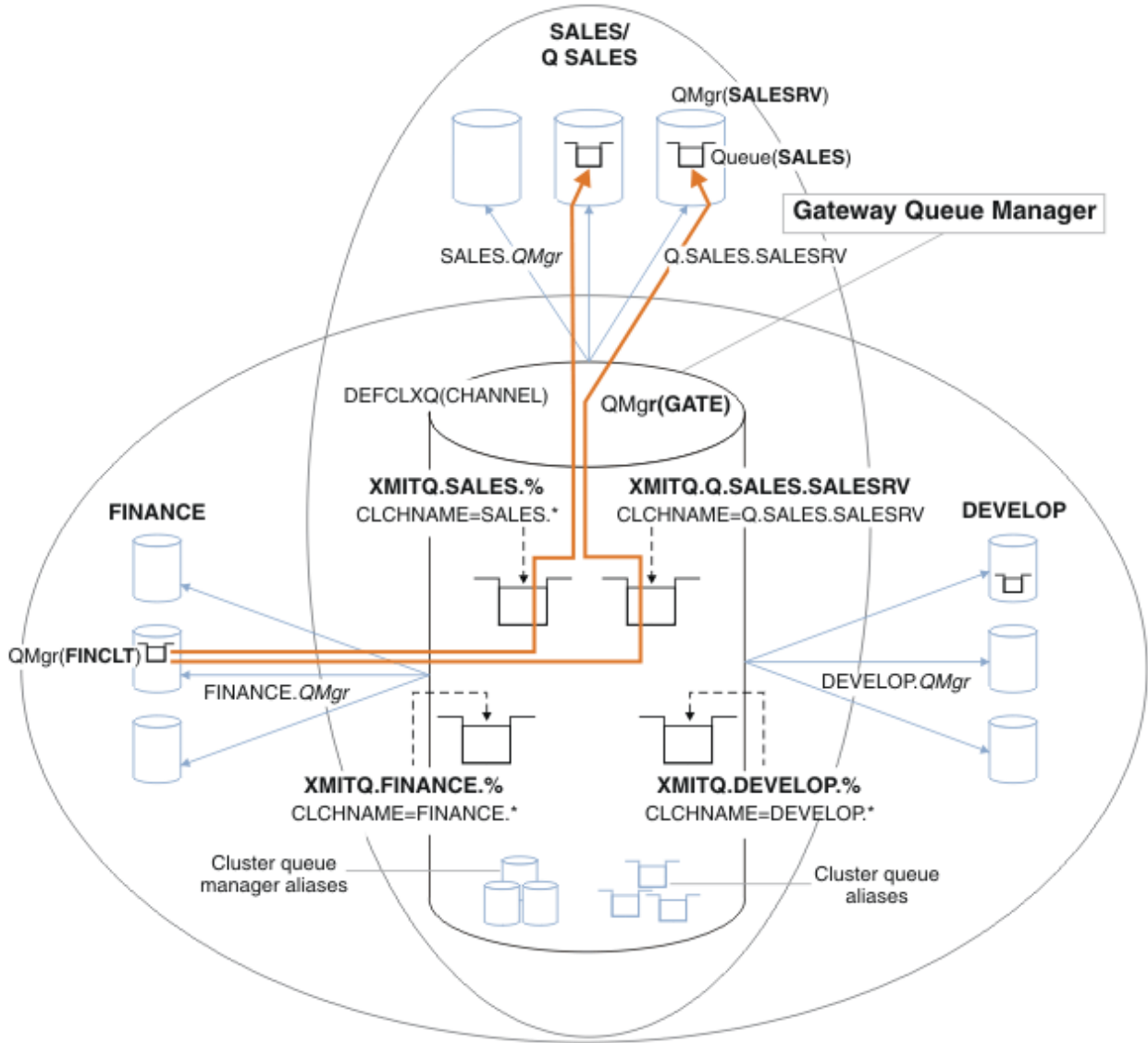
Bu görevde, birden çok küme iletim kuyruklarını üç üst üste çakışan kümeye planlamak için adımları uygulayın. Gereksinimler, diğer tüm ileti akışlarından bir küme kuyruğuna ve farklı küme iletim kuyruklarında farklı kümelere ilişkin iletileri saklamak için, iletileri ayrı ayrı bir küme kuyruğuna yollamak içindir.

### **Bu görev hakkında**

The steps in this task show how to apply the procedure in [“Kümeleme: Küme iletim kuyruklarının nasıl yapılandırılacağı planlanması” sayfa 49](#) and arrive at the configuration shown in [Şekil 14 sayfa 54](#). Bu, ayrı küme iletim kuyruklarıyla yapılandırılmış bir ağ geçidi kuyruk yöneticisi olan çakışan üç kümeden bir örnektir. Kümeleri tanımlamak için kullanılan MQSC komutları [“Örnek kümeler oluşturma” sayfa 56](#) içinde açıklanmıştır.

Örneğin, iki gereksinim vardır. Bunlardan biri, ileti akışını, ağ geçidi kuyruk yöneticisinden satışları günlükte kaydeden satış uygulamasına ayırmak olur. İkincisi, farklı departman alanlarına zaman içinde ne kadar

mesaj gönderileceğini sorgulamak. SALES, FINANCE ve DEVELOP kümeleri önceden tanımlanır. Küme iletileri şu anda SYSTEM.CLUSTER.TRANSMIT.QUEUE' den iletilir.



Şekil 14. Farklı bölüm IBM MQ kümeleri için belirli iletim kuyrukları örneği

Kümeleri değiştirme adımları aşağıdaki gibidir; tanımlamalar için [Yeni bir kümede satış kuyruğunu](#) yaratmak ve ağ geçidi kümesi iletim kuyruklarını ayırmak için yapılan [değişiklikler](#) başlıklı konuya bakın.

## Yordam

1. İlk yapılandırma adımı " [Kullanmak için varsayılan küme iletim kuyruğu tipini seçin](#) " dir.

Bu karar, GATE kuyruk yöneticisine aşağıdaki **MQSC** komutunu çalıştırarak ayrı varsayılan küme iletim kuyrukları yaratılabilir.

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

Bu varsayılan değer, küme iletim kuyruklarını el ile tanımlamakta olduğu için, bu varsayılan değeri seçmenin güçlü bir nedeni yoktur. Seçim zayıf bir tanımlama değerine sahip. El ile tanımlama yanlış yapılırsa ve bir ileti varsayılan küme iletim kuyruğunu aşağı doğru akıyorsa, bu ileti kalıcı dinamik küme iletim kuyruğu oluşturulmasında ortaya çıkar.

2. İkinci yapılandırma adımı " Küme iletim kuyruğunu diğer akışlarla paylaşmaması gereken ileti akışlarını yalıtın " dir.

Bu durumda, SALESRV üzerindeki SALES kuyruğundan ileti alan satış uygulaması yalıtma gerektirir. Yalnızca ağ geçidi kuyruk yöneticisinden gelen iletilerin yalıtılmış olması gerekir. Üç alt adım şunlardır:

- a) " Akışlar hedeflerini yapılandırın; böylece, her hedef kuyruk belirli bir kümedeki kuyrukta, o kuyruk yöneticisindequeuekuyrukta yer alan tek kuyruk olur. " .

The example requires adding the queue manager SALESRV to a new cluster within the sales department. Yalıtma gerektiren birkaç kuyruğunuz varsa, SALES kuyruğu için belirli bir küme yaratmaya karar verebilirsiniz. Küme adı için olası bir adlandırma kuralı, Q . *QueueName* gibi bu tür kümelerin adlandırılması (örneğin, Q . SALES). Yalıtılacak çok sayıda kuyruğunuz olması durumunda daha pratik olabilecek alternatif bir yaklaşım, gerektiğinde ve gerektiğinde yalıtılmış kuyruklar kümelerinin oluşturulabilirdir. Küme adları QUEUES . *not* olabilir.

Örnekte, yeni küme Q . SALESolarak adlandırılır. Yeni kümeyi eklemek için, Yeni bir kümede satış kuyruklarını yalıtma ve ağ geçidi kümesi iletim kuyruklarını ayırmak için yapılan değişiklikler içindeki tanımları görebilirsiniz. Tanımlama değişikliklerinin özeti aşağıdaki gibidir:

- i) Havuz kuyruğu yöneticilerindeki kümelerin ad listesine Q . SALES ekleyin. Ad listesi, kuyruk yöneticisi **REPOSNL** parametresine başvurur.
- ii) Ağ geçidi kuyruk yöneticisinde bulunan kümelerin ad listesine Q . SALES ekleyin. Ad listesi, ağ geçidi kuyruk yöneticisinde tüm küme kuyruğu diğer adı ve küme kuyruğu yöneticisi diğer adı tanımlamalarında gönderme edilir.
- iii) Create a namelist on the queue manager SALESRV, for both the clusters it is a member of, and change the cluster membership of the SALES queue:

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES) REPLACE  
ALTER QLOCAL(SALES) CLUSTER(' ') CLUSNL(SALESRV.CLUSTERS)
```

SALES kuyruğu, geçiş için her iki kümenin de bir üyesidir. Once the new configuration is running, you remove the SALES queue from the SALES cluster; see [Şekil 15 sayfa 59](#).

- b) " Sistematik bir adlandırma kuralından sonra yarattığınız yeni kümeler için küme gönderici ve küme alıcı kanallarını oluşturun " .

- i) Havuz günlük nesnesi kanalını Q . SALES . *RepositoryQMGr* havuz kuyruğu yöneticilerinin her birine ekleyin
- ii) Diğer havuz yöneticisine bağlanmak için, küme gönderici kanalını Q . SALES . *OtherRepositoryQMGr* havuz kuyruğu yöneticilerine ekleyin. Bu kanalları başlatın.
- iii) Q . SALES . SALESRVve Q . SALES . GATE küme alıcı kanallarını çalışmakta olan havuz kuyruğu yöneticilerinden birine ekleyin.
- iv) Add the cluster-sender channels Q . SALES . SALESRV, and Q . SALES . GATE to the SALESRV and GATE queue managers. Küme gönderen kanalını, küme alıcı kanallarını oluşturduğunuz havuz kuyruk yöneticisine bağlayın.

- c) " Hedef kuyruğa ileti gönderen her kuyruk yöneticisindeki her yalıtılmış hedef için bir küme iletim kuyruğu tanımlayın " .

On the gateway queue manager define the cluster transmission queue XMITQ . Q . SALES . SALESRV for the Q . SALES . SALESRV cluster-sender channel:

```
DEFINE QLOCAL(XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
```

3. Üçüncü yapılandırma adımı " Yönetişim ya da izleme gereksinimlerini karşılamak için küme iletim kuyrukları yaratın " dir.

Ağ geçidi kuyruk yöneticisinde, küme iletim kuyruklarını tanımlayın:

```
DEFINE QLOCAL(XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
```



```
DEFINE QLOCAL(XMITQ.DEVELOP) USAGE(XMITQ) CLCHNAME(DEVELOP.*) REPLACE  
DEFINE QLOCAL(XMITQ.FINANCE) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
```

## Sonraki adım

Ağ geçidi kuyruk yöneticisinde yeni yapılandırmaya geçin.

Anahtar, yeni kanallar başlatılarak ve artık farklı iletim kuyruklarıyla ilişkili kanalların yeniden başlatılarak tetiklenir. Diğer bir seçenek olarak, ağ geçidi kuyruk yöneticisini durdurabilir ve başlatabilirsiniz.

1. Ağ geçidi kuyruk yöneticisinde aşağıdaki kanalları durdurun:

```
SALES. Qmgr  
DEVELOP. Qmgr  
FINANCE. Qmgr
```

2. Ağ geçidi kuyruk yöneticisinde aşağıdaki kanalları başlatın:

```
SALES. Qmgr  
DEVELOP. Qmgr  
FINANCE. Qmgr  
Q.SALES.SAVESRV
```

Anahtar tamamlandığında, SALES kuyruğunu SALES kümesinden çıkarın; bkz. [Şekil 15 sayfa 59](#).

## İlgili kavramlar

Kullanılacak küme iletim kuyruğunun tipini nasıl seçecek?

Farklı küme iletim kuyruğu yapılandırma seçenekleri arasında seçim yapabilirsiniz.

## İlgili görevler

Kümeleme: Küme iletim kuyruklarının değiştirilmesi

Var olan bir üretim kuyruğu yöneticisinin küme iletim kuyruklarına ilişkin değişikliklerin yürürlüğe nasıl getirileceğini planlayın.

### Örnek kümeler oluşturma

The definitions and instructions to create the example cluster, and modify it to isolate the SALES queue and separate messages on the gateway queue manager.

## Bu görev hakkında

FINANCE, SALES ve Q.SALES kümelerini oluşturmak için tam **MQSC** komutları, [Temel kümelere ilişkin tanımlar](#), [Yeni bir kümede satış kuyruğunu yalıtma ve ağ geçidi kümesi iletim kuyruklarını ayırma için yapılan değişiklikler](#) ve [Satış kümesinden kuyruk yöneticisi SALESRV kuyruk yöneticisi üzerindeki satış kuyruğunu kaldırılarak sağlanır](#). Tanımları daha kısa tutmak için, DEVELOP kümesi tanımlamalardan çıkarılır.

## Yordam

1. SALES ve FINANCE kümelerini ve ağ geçidi kuyruk yöneticisini yaratın.

- a) Kuyruk yöneticilerini yaratın.

Run the command: `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QmgrName` for each of the queue manager names in [Çizelge 4 sayfa 56](#).

Çizelge 4. Kuyruk yöneticisi adları ve kapı numaraları		
Tanım	Kuyruk yöneticisi adı	Kapı numarası
Finans havuzu	FINR1	1414
Finans havuzu	FINR2	1415



Çizelge 4. Kuyruk yöneticisi adları ve kapı numaraları (devamı var)		
Tanım	Kuyruk yöneticisi adı	Kapı numarası
Finans istemcisi	FINCLT	1418
Satış havuzu	SALER1	1416
Satış havuzu	SALER2	1417
Satış sunucusu	SALESRV	1419
Ağ geçidi	GATE	1420

b) Tüm kuyruk yöneticilerini başlat

Run the command: `strmqm QmgrName` for each of the queue manager names in [Çizelge 4 sayfa 56](#).

c) Kuyruk yöneticilerinin her biri için tanımlamaları yarat

Şu komutu çalıştırın: `runmqsc QmgrName <filename>`, dosyalar [Temel kümelere ilişkin tanımları](#) içinde listelenmekte ve dosya adı kuyruk yöneticisi adıyla eşleşir.

### Temel kümelere ilişkin tanımlar

#### finr1.txt

```
DEFINE LISTENER(1414) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1414) REPLACE
START LISTENER(1414)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
```

#### finr2.txt

```
DEFINE LISTENER(1415) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1415) REPLACE
START LISTENER(1415)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
```

#### finclt.txt

```
DEFINE LISTENER(1418) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1418) REPLACE
START LISTENER(1418)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINCLT) CHLTYPE(CLUSRCVR) CONNAME('localhost(1418)')
CLUSTER(FINANCE) REPLACE
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

#### saler1.txt

```
DEFINE LISTENER(1416) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1416) REPLACE
START LISTENER(1416)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
```

#### saler2.txt

```
DEFINE LISTENER(1417) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1417) REPLACE
START LISTENER(1417)
ALTER QMGR REPOS(SALES)
```

```

DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE

```

### salesrv.txt

```

DEFINE LISTENER(1419) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1419) REPLACE
START LISTENER(1419)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(SALES) REPLACE
DEFINE QLOCAL(SALES) CLUSTER(SALES) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE

```

### gate.txt

```

DEFINE LISTENER(1420) TRPTYPE(TCP) IPADDR(LOCALHOST) CONTROL(QMGR) PORT(1420) REPLACE
START LISTENER(1420)
DEFINE NAMELIST(ALL) NAMES(SALES, FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(SALES) REPLACE
DEFINE QALIAS(A.SALES) CLUSNL(ALL) TARGET(SALES) TARGTYPE(Queue) DEFBIND(NOTFIXED)
REPLACE
DEFINE QREMOTE(FINCLT) RNAME(' ') RQMNAME(FINCLT) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(SALESRV) RNAME(' ') RQMNAME(SALESRV) CLUSNL(ALL) REPLACE

```

2. Örnek istek programını çalıştırarak yapılandırmayı test edin.

a) Start the trigger monitor program on the SALESRV queue manager

Windowsüzerinde bir komut penceresi açın ve `runmqttm -m SALESRV`komutunu çalıştırın.

b) Örnek istek programını çalıştırın ve bir istek gönderin.

Windowsüzerinde bir komut penceresi açın ve `amqsreq A.SALES FINCLT`komutunu çalıştırın.

İstek iletişi yeniden yankılanır ve 15 saniye sonra örnek programın sona ermesi gerekir.

3. Create the definitions to isolate the SALES queue in the Q.SALES cluster and separate cluster messages for the SALES and FINANCE cluster on the gateway queue manager.

Şu komutu çalıştırın: `runmqsc QmgrName < filename` , dosyalar aşağıdaki listede listelenmekte ve dosya adı kuyruk yöneticisi adıyla neredeyse eşleşiyor.

**Satış kuyruğunu yeni bir kümede yalıtma ve ağ geçidi kümesi iletim kuyruklarını ayırmak için yapılan değişiklikler**  
**chgsaler1.txt**

```

DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE

```

### chgsaler2.txt

```

DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE

```

```
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
```

### chgsalesrv.txt

```
DEFINE NAMELIST (CLUSTERS) NAMES(SALES, Q.SALES)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SAVESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(Q.SALES) REPLACE
ALTER QLOCAL (SALES) CLUSTER(' ') CLUSNL(CLUSTERS)
```

### chgate.txt

```
ALTER NAMELIST(ALL) NAMES(SALES, FINANCE, Q.SALES)
ALTER QMGR DEFCLXQ(CHANNEL)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('localhost(1420)')
CLUSTER(Q.SALES) REPLACE
DEFINE QLOCAL (XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
DEFINE QLOCAL (XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
DEFINE QLOCAL (XMITQ.FINANCE) USAGE(XMITQ) CLCHNAME(FINANCE.*) REPLACE
```

4. SALES kuyruğunu SALES kümesinden kaldırın.

Run the **MQSC** command in [Şekil 15 sayfa 59](#):

```
ALTER QLOCAL(SALES) CLUSTER('Q.SALES') CLUSNL(' ')
```

*Şekil 15. Satış kümesinden kuyruk yöneticisi SALESRV 'daki satış kuyruğunu kaldır*

5. Kanalları yeni iletim kuyruklarına değiştirin.

Gereksinim, GATE kuyruk yöneticisinin kullanmakta olduğu tüm kanalları durdurup başlatmadır. Bunu en az sayıda komutla yapmak için, kuyruk yöneticisini durdurup başlatın.

```
endmqm -i GATE
strmqm GATE
```

## Sonraki adım

1. Yeni yapılandırma çalışmalarını doğrulamak için örnek istek programını yeniden çalıştırın; bkz. [“2” sayfa 58](#). adım
2. GATE kuyruk yöneticilerindeki tüm küme iletim kuyruklarından akan iletileri izleyin:
  - a. Kuyruk izlemeyi açmak için, küme iletim kuyruklarının her birinin tanımlamasını değiştirin.

```
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.
name) STATQ(ON)
```

- b. Check queue manager statistics monitoring is OFF, to minimize output, and set the monitoring interval to a lower value to perform multiple tests conveniently.

```
ALTER QMGR STATINT(60) STATCHL(OFF) STATQ(OFF) STATMQI(OFF) STATACLS(OFF)
```

- c. GATE kuyruk yöneticisini yeniden başlatın.
- d. Run the sample request program a few times to verify that an equal number of messages are flowing through SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV and SYSTEM.CLUSTER.TRANSMIT.QUEUE.

İstekler, SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV aracılığıyla akış ve SYSTEM.CLUSTER.TRANSMIT.QUEUE aracılığıyla yanıt verir.

```
amqsmon -m GATE -t statistics
```

e. Birkaç aralık içindeki sonuçlar aşağıdaki gibidir:

```
C:\Documents and Settings\Admin>amqsmon -m GATE -t statistics
MonitoringType: QueueStatistics
QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '14.59.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.00.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
CreateDate: '2012-02-24'
CreateTime: '15.58.15'
...
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [435, 0]
GetCount: [1, 0]
GetBytes: [435, 0]
...
QueueStatistics: 1
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
CreateDate: '2012-02-24'
CreateTime: '16.37.43'
...
PutCount: [1, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [435, 0]
GetCount: [1, 0]
GetBytes: [435, 0]
...
MonitoringType: QueueStatistics
QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '15.00.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.01.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
CreateDate: '2012-02-24'
CreateTime: '15.58.15'
...
PutCount: [2, 0]
PutFailCount: 0
Put1Count: [0, 0]
```

```
Put1FailCount: 0
PutBytes: [863, 0]
GetCount: [2, 0]
GetBytes: [863, 0]
...
QueueStatistics: 1
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
CreateDate: '2012-02-24'
CreateTime: '16.37.43'
...
PutCount: [2, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [863, 0]
GetCount: [2, 0]
GetBytes: [863, 0]
...
2 Records Processed.
```

İlk aralıkta bir istek ve yanıt iletişi, ikincisinde iki kez gönderildi. You can infer that the request messages were placed on SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV, and the reply messages on SYSTEM.CLUSTER.TRANSMIT.QUEUE.

*Kümeleme: Küme iletim kuyruklarının değiştirilmesi*

Var olan bir üretim kuyruğu yöneticisinin küme iletim kuyruklarına ilişkin değişikliklerin yürürlüğe nasıl getirileceğini planlayın.

## Başlamadan önce

Anahtarlama işleminin yeni iletim kuyruğuna aktarılması gereken ileti sayısını azaltıyorsanız, geçiş işlemi daha hızlı tamamlanır. Devam etmeden önce iletim kuyruğunu boşaltma nedenleriyle ilgili [Küme-gönderici kanalını farklı bir iletim kuyruğuna değiştirme işlemi nasıl çalışır](#) ' i okuyun.

## Bu görev hakkında

Küme iletim kuyruklarında yapılan değişikliklerin yürürlüğe girmesi için iki yol seçeneğiniz vardır.

1. Kuyruk yöneticisinin değişiklikleri otomatik olarak gerçekleştirmesine izin verin. Bu varsayılandır. Kuyruk yöneticisi, bir küme gönderici kanalı sonraki başlatıldığında beklemedeki iletim kuyruğu değişikliklerine sahip küme gönderici kanallarını değiştirir.
2. Değişiklikleri el ile yapın. Bir küme gönderici kanalında, durdurulduğunda yapılan değişiklikleri yapabilirsiniz. Kümeyi, küme gönderici kanalı başlamadan önce bir küme iletim kuyruğundan başka bir kümeye geçirebilirsiniz.

Seçilecek iki seçenektan hangisine karar verirken hesaba katmanız gereken etkenler nelerdir ve anahtarı nasıl yönettiyorsunuz?

## Yordam

- Seçenek 1: Kuyruk yöneticisinin değişiklikleri otomatik olarak gerçekleştirmesine izin verin; bkz. [“Etkin küme gönderici kanallarının başka bir küme aktarma kuyruklarına geçişi” sayfa 62.](#)

Kuyruk yöneticisinin anahtarı sizin için gerçekleştirmesini istiyorsanız bu seçeneği belirleyin.

Bu seçeneği açıklamanın alternatif bir yolu da, kuyruk yöneticisinin, kanalı durdurmak için zorlamadan bir küme gönderici kanalını değiştirmesi olduğunu söylemeniz gerekir. Kanalı durdurmak için ve daha sonra kanalı başlatırken, anahtarı daha erken yapmak için kanalı zorlamaya gerek vardır. Anahtar, kanal başlatıldığında başlar ve kanal çalışırken, 2. seçeneğinin farklı olduğu bir sırada çalışır. Seçenek 2 'de, kanal durdurulduğunda anahtar gerçekleşir.

Bu seçeneği, anahtarın otomatik olarak gerçekleşmesine izin vererek seçerseniz, bir kümeden gönderici kanal başlatıldığında geçiş işlemi başlar. Kanal durdurulmamışsa, işlenecek bir ileti varsa, kanal devre dışı duruma geldikten sonra başlar. Kanal durdurulduysa, START CHANNEL komutunu kullanarak başlatın.

Anahtar işlemi, kanal hizmet vermekte olduğu iletim kuyruğunda kümeden gönderici kanal için bir ileti bırakılmaz tamamlanmaz tamamlanır. Bu durumda, kümeli gönderici kanalı için yeni gelen iletiler doğrudan yeni iletim kuyruğunda saklanır. O zamana kadar iletiler eski iletim kuyruğunda saklanır ve geçiş işlemi iletileri eski iletim kuyruğundan yeni iletim kuyruğuna aktarır. Küme gönderici kanalı, tüm anahtarlama işlemi sırasında yeni küme iletim kuyruğundan iletileri iletir. Anahtar işlemi tamamlanınca sistemin durumuna bağlıdır. Değişiklikleri bir bakım penceresinde yapıyorsanız, geçiş sürecinin zamanında tamamlanıp tamamlanmayacağını önceden değerlendirin. Zaman içinde tamamlanıp tamamlanmayacağı, eski iletim kuyruğundan aktarımı bekleyen ileti sayısının sıfıra ulaşmasına bağlıdır.

İlk yöntemin avantajı otomatidir. Bir dezavantaj, yapılandırma değişikliklerinin bir bakım penceresinden sınırlı olması durumunda, sistem, bakım penceresindeki anahtar sürecini tamamlamak için sistemi denetleyebileceğinden emin olmanız gerekir. Emin olamıyorsanız, seçenek 2 daha iyi bir seçim olabilir.

- Seçenek 2: Değişiklikleri el ile yapın; bkz. [“Durdurulmuş bir küme gönderici kanalının başka bir küme iletim kuyruğuna değiştirilmesi” sayfa 64.](#)

Tüm anahtarlama işlemi el ile denetlemek ya da durdurulmuş ya da etkin olmayan bir kanalı değiştirmek istiyorsanız bu seçeneği belirleyin. Birkaç küme gönderici kanalı değiştiriyorsanız ve bir bakım penceresi sırasında anahtarı yapmak istiyorsanız, bu iyi bir seçimdir.

Bu seçeneğin alternatif bir açıklaması, kümeyi gönderen kanalı durdurulurken, kümeyi gönderen kanalı değiştirdiğinizi söylemeniz gerekir.

Bu seçeneği belirlerseniz, anahtar yerine geçtiğinde tam denetime sahip olur.

Bir bakım aralığı içinde, değiştirme işlemi sabit bir süre içinde tamamlamaktan emin olabilirsiniz. Anahtarın bir iletim kuyruğundan diğerine kaç ileti aktarılacağına başlanacağına in saat. İleti gelmeye devam ederse, işlemin tüm iletileri aktararak aktarması biraz zaman alabilir.

Eski iletim kuyruğundan ileti aktarmadan kanalı değiştirme seçeneğiniz vardır. Anahtar "instant"(Anında).

Küme gönderici kanalını yeniden başlattığınızda, bu kanalı yeni atadığınız iletim kuyruğunda iletileri işlemeye başlar.

İkinci yöntemin avantajı, geçiş süreci üzerinde denetime sahip olmanızdır. Dezavantajı ise, anahtarlama olarak küme gönderici kanallarını belirlemeniz, gerekli komutları çalıştırmanız ve küme gönderen kanalının durdurulmasını önleyen tüm belirsiz kanalları çözmeniz gerekir.

## İlgili kavramlar

[Kullanılacak küme iletim kuyruğunun tipini nasıl seçecek?](#)

[Farklı küme iletim kuyruğu yapılandırma seçenekleri arasında seçim yapabilirsiniz.](#)

[Küme-gönderici kanalını farklı bir iletim kuyruğuna değiştirme işlemi nasıl çalışır](#)

## İlgili görevler

[Kümeleme: Birden çok küme iletim kuyruklarının yapılandırılması](#)

Bu görevde, birden çok küme iletim kuyruklarını üç üst üste çakışan kümeye planlamak için adımları uygulayın. Gereksinimler, diğer tüm ileti akışlarından bir küme kuyruğuna ve farklı küme iletim kuyruklarında farklı kümelere ilişkin iletileri saklamak için, iletileri ayrı ayrı bir küme kuyruğuna yollamak içindir.

*Etkin küme gönderici kanallarının başka bir küme aktarma kuyruklarına geçişi*

Bu görev, etkin küme gönderici kanallarını değiştirmek için size üç seçenek sunar. Bir seçenek, kuyruk yöneticisinin anahtarı otomatik olarak gerçekleştirmesine izin vermek; bu da çalışan uygulamaları etkilemez. Diğer seçenekler, kanalları el ile durdurup başlatmak ya da kuyruk yöneticisini yeniden başlatmak için kullanılabilir.

## Başlamadan önce

Küme iletim kuyruğu yapılanışını değiştirin. **DEFCLXQ** kuyruk yöneticisi özniteliğini değiştirebilir ya da iletim kuyruklarının **CLCHNAME** özniteliğini ekleyebilir ya da değiştirebilirsiniz.

Anahtarlama işleminin yeni iletim kuyruğuna aktarılması gereken ileti sayısını azaltıyorsanız, geçiş işlemi daha hızlı tamamlanır. Devam etmeden önce iletim kuyruğunu boşaltma nedenleriyle ilgili [Küme-gönderici kanalını farklı bir iletim kuyruğuna değiştirme işlemi nasıl çalışır](#) ' i okuyun.

## Bu görev hakkında

Küme iletim kuyruğu yapılandırması değişiklikleri yapmak için kendi planınızın üzerinde çalışmak için görevdeki adımları temel olarak kullanın.

## Yordam

### 1. İsteğe bağlı: Geçerli kanal durumunu kaydet

Küme iletim kuyruklarına hizmet veren yürürlükteki ve saklanmış kanalların durumunu kaydedin. Aşağıdaki komutlarda, sistem kümesi iletim kuyruklarıyla ilişkili durum görüntülenir. Tanımladığınız küme iletim kuyruklarıyla ilişkili durumu görüntülemek için kendi komutlarınızı ekleyin. Bu iletim kuyruklarına ilişkin kanal durumunun görüntülenmesini kolaylaştırmak üzere tanımladığınız küme iletim kuyruklarına ad vermek için XMITQ. *ChannelName* gibi bir toplantı kullanın.

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

### 2. Aktarım kuyruklarını değiştir.

- Hiçbir şey yapma. Kuyruk yöneticisi, durdurulduktan ya da devre dışı kaldıktan sonra yeniden başlatıldığında, küme gönderen kanallarını değiştirir.

Bir kuyruk yöneticisi yapılanışını değiştirme konusunda herhangi bir kuralınız ya da endişeniz yoksa bu seçeneği belirleyin. Çalışan uygulamalar değişikliklerden etkilenmez.

- Kuyruk yöneticisini yeniden başlatın. Tüm küme-gönderici kanalları istek üzerine otomatik olarak durdurulur ve yeniden başlatılır.

Tüm değişiklikleri hemen başlatmak için bu seçeneği belirleyin. Çalışmakta olan uygulamalar kuyruk yöneticisi tarafından durdurularak sona erdirilir ve yeniden başlatılır.

- Tek tek küme-gönderici kanallarını durdurun ve yeniden başlatın.

Birkaç kanalı hemen değiştirmek için bu seçeneği belirleyin. Çalışan uygulamalar, durdurulan ve ileti kanalını yeniden başlamanızın arasında ileti aktarımında kısa bir gecikme deneyimi yaşamamanızı sağlar. Küme gönderici kanalı, durdurduğu süre dışında, çalışmakta olmaya devam eder. Anahtar işlemi sırasında iletiler eski iletim kuyruğuna teslim edilir, yeni iletim kuyruğuna geçiş işlemi tarafından aktarılır ve yeni iletim kuyruğundan, küme gönderen kanalına iletilir.

### 3. İsteğe bağlı: Kanalları değişikçe izleyin

Anahtar sırasında kanal durumunu ve iletim kuyruğu derinliğini görüntüleyin. Aşağıdaki örnek, sistem kümesi iletim kuyruklarının durumunu görüntüler.

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

### 4. İsteğe bağlı: Kuyruk yöneticisi hata günlüğüne yazılan "AMQ7341 The transmission queue for channel *ChannelName* switched from queue *QueueName* to *QueueName*" iletilerini izleyin.

*Durdurulmuş bir küme gönderici kanalının başka bir küme iletim kuyruğuna değiştirilmesi*  
Değişiklikleri el ile yapmayı seçerseniz, durdurulduğunda bir küme gönderen kanalına yapılan değişiklikleri yaparsanız, küme gönderici kanalı başlatılmadan önce bir küme iletim kuyruğundan başka bir aktarım kuyruğundan diğerine geçmenizi sağlar.

## Başlamadan önce

Bazı yapılandırma değişiklikleri yapabilirsiniz ve bu değişiklikleri, etkilenen küme gönderen kanallarına başlamaksızın etkili yapmak isteyebilirsiniz. Diğer bir seçenek olarak, gerekli yapılandırma değişikliklerini, görevdeki adımlardan biri olarak gerçekleştirmenizi sağlar.

Anahtarlama işleminin yeni iletim kuyruğuna aktarılması gereken ileti sayısını azaltıyorsanız, geçiş işlemi daha hızlı tamamlanır. Devam etmeden önce iletim kuyruğunu boşaltma nedenleriyle ilgili [Küme-gönderici kanalını farklı bir iletim kuyruğuna değiştirme işlemi nasıl çalışır](#) ' i okuyun.

## Bu görev hakkında

Bu görev, durdurulmuş ya da etkin olmayan küme gönderen kanallarının hizmet verdiği iletim kuyruklarını değiştirir. Bu görevi, bir küme gönderici kanalı durdurulduğundan ve iletim kuyruğunu hemen değiştirmek istiyorsanız yapmanız gerekebilir. Örneğin, bir nedenden dolayı bir küme-gönderen kanalı başlatılmamıştır ya da başka bir yapılandırma sorunu vardır. Sorunu çözmek için, bir küme-gönderici kanalı yaratmaya karar verirsiniz ve tanımladığınız yeni küme gönderici kanalına sahip eski kümeli gönderici kanalına ilişkin iletim kuyruğunu ilişkilendirmeye karar verirsiniz.

Küme iletim kuyruklarının yeniden yapılandırılırken denetlenmesini istediğiniz olası bir senaryo daha olası olur. Yeniden yapılandırmayı tam olarak denetlemek için, kanalları durdurursun, yapılandırmayı değiştirin ve daha sonra iletim kuyruklarını değiştirin.

## Yordam

1. Değiştirmek istediğiniz kanalları durdurun.

- a) Değiştirmek istediğiniz, çalışmakta olan ya da etkin olmayan kanalları durdurun. Etkin olmayan bir küme gönderici kanalı durdurulurken, yapılandırma değişiklikleri yaparken bu kanal başlatılmasına engel olur.

```
STOP CHANNEL(ChannelName) MODE(QUIESCSE) STATUS(STOPPED)
```

2. İsteğe bağlı: Yapılandırma değişikliklerini yapın.

Örneğin, bkz. [“Kümeleme: Birden çok küme iletim kuyruklarının yapılandırılması”](#) sayfa 53.

3. Küme gönderici kanallarını yeni küme iletim kuyruklarına geçirmeniz gerekir.

**Multi** [Multiplatforms'](#) ta şu komutu verin:

```
runcswchl -m QmgrName -c ChannelName
```

**z/OS** [z/OS'](#) ta, iletileri değiştirmek ya da olanları izlemek için CSQUTIL komutu SWITCH işlevini kullanın. Aşağıdaki komutu kullanın.

```
SWITCH CHANNEL(channel_name) MOVEMSGS(YES)
```

Ek bilgi için [SWITCH işlevi](#) başlıklı konuya bakın.

**runcswchl** ya da CSQUTIL SWITCH komutu, eski iletim kuyruğundaki iletileri yeni iletim kuyruğuna aktarır. Bu kanala ilişkin eski iletim kuyruğunda bulunan iletilerin sayısı sifıra ulaştığında, anahtar tamamlanır. Komut zamanuyumlu. Komut değiştirme işlemi sırasında komut, aşama iletilerini pencereye yazar.



Aktarma aşaması sırasında, yeni iletim kuyruğuna, küme gönderen kanalına gönderilen yeni iletiler aktarılır.

Küme gönderen kanalı durdurulduğundan, iletiler yeni iletim kuyruğunda birikir. Durdurulan küme gönderen kanalının tersine, [“Etkin küme gönderici kanallarının başka bir küme aktarma kuyruklarına geçişi” sayfa 62](#) içindeki [“2” sayfa 63](#) adımına bakın. Bu adımda, küme-gönderici kanalı çalışıyor, bu nedenle iletilerin yeni iletim kuyruğunda oluşturulmamış olması gerekmez.

#### 4. İsteğe bağlı: Kanalları değiştikçe izleyin

Farklı bir komut penceresinde, anahtar sırasında iletim kuyruğu derinliğini görüntüleyin. Aşağıdaki örnek, sistem kümesi iletim kuyruklarının durumunu görüntüler.

```
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

#### 5. İsteğe bağlı: Kuyruk yöneticisi hata günlüğüne yazılan "AMQ7341 The transmission queue for channel *ChannelName* switched from queue *QueueName* to *QueueName*" iletilerini izleyin.

#### 6. Durdurduğunuz küme gönderici kanallarını yeniden başlatın.

The channels do not start automatically, as you stopped them, placing them into DURDURULDU status.

```
START CHANNEL(ChannelName)
```

### İlgili başvurular

[runswchl](#)

[KANALIN](#)

[KANAL DURDUR](#)

*Kümeleme: Geçiş ve değişiklik en iyi uygulamaları*

Bu konuda, IBM MQ kümelerini planlama ve yönetme kılavuzu sağlanır. Bu bilgiler, müşterilerden gelen test ve geri bildirimlere dayalı bir kılavuzdur.

1. [“Bir Kümedeki Nesnelerin Taşınması” sayfa 65](#) (Best practices for moving objects around inside a cluster, without installing any fix packs or new versions of IBM MQ ).
2. [“Büyütmeler ve bakım kuruluşları” sayfa 66](#) (Bir çalışan küme mimarisini çalışır ve çalışır durumda tutmak için en iyi uygulamalar, bakım veya büyütmeler uygulanırken ve yeni mimariyi test etmek için).

## Bir Kümedeki Nesnelerin Taşınması

### Uygulamalar ve kuyrukları

Başka bir kuyruk yöneticisine ev sahipliği yapmak üzere bir kuyruk örneğini taşımanız gerekirken, düzgün bir geçiş sağlamak için iş yükü dengeleme parametreleriyle çalışabilirsiniz.

Yeni barındırılacağı bir kuyruk örneği yaratın, ancak uygulamanızın değişmeye hazır oluncaya kadar özgün yönetim ortamına ileti göndermeye devam etmek için küme iş yükü dengeleme ayarlarını kullanın. Bu, aşağıdaki adımlarla gerçekleştirilir:

1. Var olan kuyruğun **CLWL**RANK özelliğini yüksek bir değere (örneğin, beş) ayarlayın.
2. Kuyruğun yeni örneğini oluşturun ve **CLWL**RANK özelliğini sıfır olarak ayarlayın.
3. Yeni sistemin daha fazla yapılandırmasını tamamlayın; örneğin, uygulamaları kuyruğun yeni örneğine karşı konuşlandırmak ve tüketmeye başlayın.
4. Yeni kuyruk örneğinin **CLWL**RANK özelliğini, özgün örneğinden daha yüksek olacak şekilde ayarlayın; örneğin, dokuz.
5. Özgün kuyruk örneğinin sistemde kuyruğa alınan iletileri işlemesini ve kuyruğun silinmesini sağlayın.

### Kuyruk yöneticilerinin tamamının taşınması

Kuyruk yöneticisi aynı anasistemde kalıyorsa, ancak IP adresi değişiyorsa, işlem şu şekildedir:

- DNS, doğru kullanıldığında, süreci basitleştirmeye yardımcı olabilir. Bağlantı adı (CONNNAME) kanal özniteliğini ayarlayarak DNS kullanımıyla ilgili bilgi için ALTER CHANNEL başlıklı konuya bakın.
- Bir tam havuzu taşıyorsanız, değişiklik yapmadan önce sorunsuz bir şekilde çalışan en az bir tam havuza (örneğin, örneğin kanal durumuyla ilgili sorun yok) sahip olduğundan emin olun.
- Trafik oluşturma işlemini önlemek için SUSPEND QMGR komutunu kullanarak kuyruk yöneticisini askıya alın.
- Bilgisayarın IP adresini değiştirin. CLUSTRVR kanal tanımınızda CONNNAME alanında bir IP adresi kullanılıyorsa, bu IP adresi girişini değiştirin. Güncellemelerin her yerde kullanılabilir olduğundan emin olmak için DNS önbelleğinin temizlenmesi gerekebilir.
- Kuyruk yöneticisi tüm havuzlara yeniden bağlandığında, kanal otomatik tanımlamaları otomatik olarak kendini çözer.
- Kuyruk yöneticisi tam bir havuz barındırıyor ve IP adresi değişirse, el ile tanımlanan CLUSSDR kanallarını yeni konuma işaret etmek için tarafların en kısa zamanda bu tarafa geçtiğinden emin olun. Bu anahtar gerçekleştirilinceye kadar, bu kuyruk yöneticileri yalnızca geri kalan (değiştirilmemiş) tam havuzla iletişim kurabilirler ve yanlış kanal tanımlamasıyla ilgili uyarı iletileri görülebilir.
- Resume the queue manager using the QMGR 'YI Sü command.

Kuyruk yöneticisinin yeni bir anasisteme taşınması gerekiyorsa, kuyruk yöneticisi verilerinin kopyalanabileceği ve yedekten geri yüklenmesi mümkündür. Başka seçenek yoksa, bu işlem önerilmez; yeni bir makinede kuyruk yöneticisi yaratmak daha iyi olabilir ve önceki bölümde açıklandığı gibi, kuyrukları ve uygulamaları çoğaltmak daha iyi olabilir. Bu durum pürüzsüz bir baştan başlama/geri alma mekanizması sağlar.

Yedeklemeyi kullanarak tam bir kuyruk yöneticisini taşımaya karar verdiyseniz, aşağıdaki en iyi uygulamaları izleyin:

- Tüm süreci bir kuyruk yöneticisi geri yükleme işlemi olarak, işletim sistemi ortamınız için uygun olan sistem kurtarması için kullanabildiğiniz tüm işlemleri uygulayarak, yedekten geri yükleme işlemini geri yükleyin.
- Yerel olarak tutulan tüm küme bilgilerini (belirsiz olan otomatik tanımlı kanallar da içinde olmak üzere) atmak için geçişten sonra **REFRESH CLUSTER** komutunu kullanın ve bu bilgileri yeniden oluşturulmalıdır.

**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 güncellemeleri gönderdiğinde, 27 günlük aralıklarla kümede kesintiye neden olabilir. Bkz. Büyük bir kümede yenilenmesi, kümenin performansını ve kullanılabilirliğini etkileyebilir.

Bir kuyruk yöneticisi yaratırken kümedeki var olan bir kuyruk yöneticisinden kuruluş eşlenirken (daha önce bu konuda açıklandığı gibi), iki farklı kuyruk yöneticisini hiçbir zaman aynı şekilde davranmayın. Özellikle, yeni bir kuyruk yöneticisi için aynı kuyruk yöneticisi adını ve IP adresini vermeyin. Yeni bir kuyruk yöneticisinde 'bırakma' girişimi, IBM MQ kümelerinde sık rastlanan bir sorun nedenidir. Önbellek, **QMID** özniteliği de dahil olmak üzere güncellemeleri almayı bekler ve durum bozulabilir.

Aynı adı taşıyan iki farklı kuyruk yöneticisi yanlışlıkla oluşturulduysa, yanlış girişi kümeden çıkarmak için **RESET CLUSTER QMID** komutunu kullanmanız önerilir.

## Büyütmeler ve bakım kuruluşları

Sözde büyük patlama senaryosundan kaçının (örneğin, tüm küme ve kuyruk yöneticisi etkinliğini durdurmak, tüm kuyruk yöneticilerine tüm yükseltmeleri ve bakımı uygulamak, daha sonra her şeyi aynı anda başlatmak). Kümeler, kuyruk yöneticisi katile ilişkin birden çok sürümle çalışmaya devam edecek şekilde tasarlanmıştır, bu nedenle iyi planlanmış, aşamalı bakım yaklaşımı önerilir.

Yedek bir planın var mı:

-  z/OS'ta geriye doğru geçiş PTF'leri uyguladınız mı?

- Yedekleri aldın mı?
- Hemen yeni küme işlevselliğini kullanmaktan kaçınin: Tüm kuyruk yöneticilerinin yeni düzeye yükseltildiğinden emin oluncaya kadar bekleyin ve bunların hiçbirini geri almayacağından emin olun. Bazı kuyruk yöneticilerinin daha önceki bir düzeyin bulunduğu bir kümede yeni küme işlevinin kullanılması, tanımsız davranışlara yol açabilir. Örneğin, IBM WebSphere MQ 6.0'dan IBM WebSphere MQ 7.1 ' a yapılan taşıma sırasında, bir kuyruk yöneticisi bir küme konusu tanımlıyorsa, IBM WebSphere MQ 6.0 kuyruk yöneticileri tanımlamayı anlayamaz ya da bu konuda yayınlayamaz.

Önce tam havuzları geçirin. Anlamadıkları bilgileri iletebilirlerse de, bunu sürdüremezler, bu nedenle kesinlikle gerekli olan bir yaklaşım değildir. Ek bilgi için [Kuyruk yöneticisi küme geçiş](#) başlıklı konuya bakın.

*Kümeleme: REFRESH CLUSTER en iyi uygulamaları kullanma*

Bir kümeye ilişkin olarak yerel olarak tutulan tüm bilgileri atmak ve bu bilgileri kümedeki tüm havuzlardan yeniden oluşturmak için **REFRESH CLUSTER** komutunu kullanıyorsunuz. Bu komutu, istisnai koşullar dışında, kullanmanıza gerek yoktur. Bunu kullanmanız gerekiyorsa, bunu nasıl kullanmanız gerektiği konusunda dikkat edilmesi gereken noktalar vardır. Bu bilgiler, müşterilerden gelen test ve geri bildirimlere dayalı bir kılavuzdur.

## **Yalnızca REFRESH CLUSTER ' ı çalıştırın; bu işlemi gerçekten yapmanız gerekir**

IBM MQ küme teknolojisi, kümelenmiş bir kuyruğun değiştirilmesi gibi küme yapılandırmalarında herhangi bir değişikliğin otomatik olarak, bilgileri bilmesi gereken kümenin herhangi bir üyesi tarafından otomatik olarak bilinmesini sağlar. Bu bilgilerin yayılmasını sağlamak için daha fazla idari adıma gerek yoktur.

Bu tür bilgiler, gerekli olduğu kümedeki kuyruk yöneticilerine ulaşmazsa, örneğin kümelenmiş bir kuyruk, bir uygulama ilk kez açmayı denediğinde kümedeki başka bir kuyruk yöneticisi tarafından bilinmez; bu, küme altyapısında bir sorun anlamına gelir. Örneğin, kuyruk yöneticisi ile tam havuz kuyruk yöneticisi arasında bir kanal başlatılamaması mümkündür. bu nedenle tutarsızlıklar gözlenmekte olan her türlü durum araştırılmalıdır. If possible, resolve the situation without using the **REFRESH CLUSTER** command.

Bu ürün belgelerinin başka bir yerinde belgelenmiş nadir durumlarda ya da IBM desteği tarafından istendiğinde, bir kümeye ilişkin olarak yerel olarak tutulan tüm bilgileri atmak ve bu bilgileri kümedeki tüm havuzlardan yeniden oluşturmak için **REFRESH CLUSTER** komutunu kullanabilirsiniz.

## **Büyük bir kümede yenileme, kümenin performansını ve kullanılabilirliğini olumsuz etkileyebilir**

Use of the **REFRESH CLUSTER** command can be disruptive to the cluster while it is in progress, for example by creating a sudden increase in work for the full repositories as they process the repropagation of queue manager cluster resources. Büyük bir kümede (yani, yüzlerce kuyruk yöneticisi) yeniliyorsanız, mümkünse günlük işteki komuttan kaçınmalısınız ve belirli tutarsızlıkları düzeltmek için alternatif yöntemler kullanırsınız. Örneğin, bir küme kuyruğu doğru bir şekilde geçirilmiyorsa, kümelenmiş kuyruk tanımlamasını güncellemeye ilişkin ilk araştırma tekniği (tanımını değiştirmek gibi), kuyruk yapılanışını küme genelinde yeniden yayılır. Bu işlem, sorunun saptanmasına ve geçici tutarsızlığını çözenize yardımcı olabilir.

Alternatif yöntemler kullanılmıyorsa ve **REFRESH CLUSTER** komutunu büyük bir kümede çalıştırmanız gerekiyorsa, kullanıcı iş yüklerinin etkisini önlemek için yoğun zamanlarda ya da bir bakım penceresinde bunu yapmalısınız. Ayrıca, tek bir toplu işte büyük bir kümeyi yenilemekten kaçınmalı ve etkinliği "[Küme nesnelere otomatik güncellemeler gönderildiğinde performans ve kullanılabilirlik sorunlarını önle](#)" sayfa 67'inde açıklandığı gibi basamaklandırmak da gerekir.

## **Küme nesnelere otomatik güncellemeler gönderildiğinde performans ve kullanılabilirlik sorunlarını önle**

Kuyruk yöneticisinde yeni bir küme nesnesi tanımlandıktan sonra, bu nesne için bir güncelleme, tanımlama süresinden her 27 günde bir oluşturulur ve kümedeki her tam havuza gönderilir ve diğer ilgili kuyruk yöneticilerine iletilir. Bir kuyruk yöneticisine **REFRESH CLUSTER** komutunu verdiğinizde, belirtilen kümede yerel olarak tanımlanan tüm nesnelere bu otomatik güncellemeye ilişkin saati sıfırlayın.

Büyük bir kümeyi (yani, yüzlerce kuyruk yöneticisi) tek bir toplu işte ya da bir sistemi yapılandırma yedeklemeden yeniden oluşturma gibi diğer durumlarda, o kuyruk yöneticilerinin 27 gün sonra tüm nesne tanımlamalarını aynı anda tüm nesne tanımlarını tam havuzlara yeniden yayınlayacağı gibi başka bir durumda da yenilemişseniz. Bu yeniden, tüm güncellemeler tamamlanıncaya kadar, sistemin önemli ölçüde daha yavaş çalışmasına ya da kullanılamaz duruma gelmesine neden olabilir. Bu nedenle, büyük bir kümede birden çok kuyruk yöneticisini yenilemeli ya da yeniden oluşturduğunuzda, etkinliği birkaç saat ya da birkaç gün boyunca kaydedemelisiniz; böylece, sonraki otomatik güncellemeler sistem başarımını düzenli olarak etkilemez.

## Sistem kümesi geçmiş kuyruğu

Bir **REFRESH CLUSTER** gerçekleştirildiğinde, kuyruk yöneticisi, yenileme işleminden önce küme durumuna ilişkin bir anlık görüntü alır ve kuyruk yöneticisinde tanımlandıysa, SYSTEM. CLUSTER. HISTORY. QUEUE (SCHQ) üzerinde saklanır. This snapshot is for IBM service purposes only, in case of later problems with the system.

SCHQ, başlatma sırasında dağıtılmış kuyruk yöneticilerindeki varsayılan değer olarak tanımlanır. z/OS geçişi için, SCHQ el ile tanımlanmalıdır.

SCHQ 'daki iletiler üç ay sonra sona erer.

### İlgili kavramlar

[“Yayınlama/abone olma kümeleri için Kümeyi Yenile” sayfa 118](#)

Issuing the **REFRESH CLUSTER** command results in the queue manager temporarily discarding locally held information about a cluster, including any cluster topics and their associated proxy subscriptions.

[REFRESH CLUSTER çalıştırılırken uygulama sorunları görüldü](#)

### İlgili başvurular

[MQSC Commands başvurusu: REFRESH CLUSTER](#)

*Kümeleme: Kullanılabilirlik, çoklu eşgörünüm ve olağanüstü durumdan kurtarma*

Bu konuda, IBM MQ kümelerini planlama ve yönetme kılavuzu sağlanır. Bu bilgiler, müşterilerden gelen test ve geri bildirimlere dayalı bir kılavuzdur.

IBM MQ Clustering 'in kendisi High Availability (Yüksek Kullanılabilirlik) çözümü değildir, ancak bazı durumlarda, örneğin farklı kuyruk yöneticilerinde bir kuyruğun birden çok örneğinin olması gibi, IBM MQkullanan hizmetlerin kullanılabilirliğini artırmak için kullanılabilir. Bu bölümde, IBM MQ altyapısının bu tür bir mimaride kullanılabilmesi için mümkün olduğunca yüksek düzeyde kullanılabilir olmasını sağlama konusunda kılavuz bilgiler verilmektedir.

**Not:** Diğer yüksek kullanılabilirlik ve olağanüstü durumdan kurtarma çözümleri IBM MQiçin kullanılabilir. Bkz. [Configuring high availability, recovery and restart.](#)

### Küme kaynaklarının kullanılabilirliği

İki tam havuzu korumak için olağan önerinin nedeni, birinin kaybının kümenin düzgün bir şekilde çalıştırılması açısından kritik önem göstermemesidir. Her ikisi de kullanılamaz duruma gelse de, kısmi havuzlar tarafından tutulan mevcut bilgi için 60 günlük bir ek süre vardır; ancak, yeni ya da daha önce erişilmemiş kaynaklar (örneğin, örnek) bu etkinlikte kullanılamaz.

### Uygulama kullanılabilirliğini artırmak için kümeleri kullanma

Bir küme, kuyruğun ve uygulamanın birden çok örneğini kullanarak yüksek düzeyde kullanılabilir uygulamalar tasarlamaya (örneğin, bir istek/yanıt tipi sunucu uygulaması) yardımcı olabilir. Gerekiyorsa, öncelik öznitelikleri, örneğin bir kuyruk yöneticisi ya da kanal kullanılamaz hale gelmediği sürece 'canlı' uygulamaya tercih yapabilir. Bu, bir sorun ortaya çıktığında yeni iletileri işlemeye devam etmek için hızla geçiş yapmak için güçlü bir işlemdir.

Ancak, bir kümede belirli bir kuyruk yöneticisine teslim edilen iletiler yalnızca o kuyruk örneğinde tutulur ve kuyruk yöneticisi kurtarıncaya kadar işlenemez. Bu nedenle, gerçek veri yüksek kullanılabilirliği için, çok eşgörünümlü kuyruk yöneticileri gibi diğer teknolojileri de göz önünde bulundurmanız gerekebilir.


## Çok eşgörünümli kuyruk yöneticileri

Software High Availability ("Software High Availability"), var olan iletilerinizi sağlamak için yerleşik bir olanaktır. Daha fazla bilgi için bkz. Yüksek kullanılabilirlikli yapılandırmalarla IBM MQ kullanılması, Çok eşgörünümli kuyruk yöneticisi yaratılması ve aşağıdaki kısım ile ilgili bilgi edinin. Kümedeki tüm kuyruk yöneticilerinin en az IBM WebSphere MQ 7.0.1 çalıştığı sürece, kümedeki herhangi bir kuyruk yöneticisi bu teknik kullanılarak yüksek kullanılabilirlikli kılınabilir. Kümedeki kuyruk yöneticileri önceki düzeylerde varsa, ikincil bir IP 'ye geçemezlerse, çok eşgörünümli kuyruk yöneticileriyle bağlantı kaybedebilir.

Bu konuda daha önce ele alındığı gibi, iki tam havuz yapılandırıldığı sürece, bunlar hemen hemen nitelikleri tarafından kullanılabilir durumda olur. Gerekliyse, IBM MQ yazılımı High Availability/Multi-instance kuyruk yöneticileri tüm havuzlar için kullanılabilir. Bu yöntemleri kullanmak için güçlü bir neden yoktur ve geçici kesintiler için bu yöntemler, hata durumunda yedek sisteme geçiş sırasında ek performans maliyetine neden olabilir. Örneğin, tek bir kanal kesintisi durumunda, iki tam havuz çalıştırmak yerine yazılım HA 'nın kullanılması önerilmez; ancak, bu bir kanal kesintisi durumunda başarısız olmaz, ancak kısmi havuzların küme kaynakları için sorgulayamaması durumunda bırakılabilir.

## Olağanüstü durumdan kurtarma


Olağanüstü durumdan kurtarma, örneğin, bir kuyruk yöneticisinin verilerini saklayan disklerin bozulması nedeniyle kurtarma işlemi bozuk olduğu için, iyi olması zor; IBM MQ yardım edebilir, ancak bunu otomatik olarak yapamayabilir. IBM MQ içindeki tek 'true' olağanüstü durumdan kurtarma seçeneği (herhangi bir işletim sistemi ya da diğer temel kopyalama teknolojileri dışında) bir yedekten geri yüklenir. Bu durumlarda göz önünde bulundurulması gereken bazı kümeleme noktaları vardır:

- Olağanüstü durumdan kurtarma senaryolarını test ederken dikkatli olun. Örneğin, yedek kuyruk yöneticilerinin çalışmasını test etmek, yanlışlıkla canlı kümeye katılmak ve canlı küme kuyruğu yöneticilerindeki aynı adı taşıyan kuyrukları barındırarak 'çalma' iletileri başlatmak mümkün olduğu için, bunları çevrimiçi olarak aynı ağda çevrimiçi olarak getirirken dikkatli olun.
- Olağanüstü durumdan kurtarma testi, çalışmakta olan bir canlı kümeye müdahale etmemelidir. Paraziti önlemek için kullanılan teknikler şunlardır:
  - Ağ ayrılığı ya da ayırma güvenlik duvarı düzeyinde tamamdır.
  -  Kanal kullanıma hazırlama ya da z/OS **chinit** adres alanı başlatılmaz.
  - Olağanüstü durumdan kurtarma sistemine kadar ya da gerçek bir olağanüstü durumdan kurtarma senaryosu ortaya çıkıncaya kadar canlı TLS sertifikası yayınlamamak.
- Bir kuyruk yöneticisinin yedeğini kümede geri yüklerken, yedeğin kümenin geri kalanıyla eşitlenmemiş olması mümkün olur. **REFRESH CLUSTER** komutu, güncellemeleri çözebilir ve kümeyle uyumlulaştırabilir; ancak, **REFRESH CLUSTER** komutu son çare olarak kullanılmalıdır. Bkz. "Kümeleme: REFRESH CLUSTER en iyi uygulamaları kullanma" sayfa 67. Komutu kullanmaya başvurmadan önce basit bir adımın kaçırılıp kaçırılmadığını görmek için kurum içi süreç belgelerini ve IBM MQ belgelerini gözden geçirin.
- Herhangi bir kurtarma işlemi için, uygulamaların yeniden yürütülmesi ve veri kaybı ile ilgilenmesi gerekir. Kuyrukların bilinen bir duruma doğru temizlenip temizlenmeyeceğine ya da yeniden oynamaları yönetmek için başka bir yerde yeterli bilgi olup olmadığına karar verilmelidir.

V 9.1.2

Multi

## Tek biçimli kümeler

Tek bir küme devreye almanın amacı, uygulamaların ölçekleme ve kullanılabilirlik için tasarlanabileceği ve tek tip küme içindeki kuyruk yöneticilerine bağlanabilmesi. Bu, belirli bir kuyruk yöneticisinde bağımlılığı kaldırır ve ileti alışverişi trafiğindeki daha iyi kullanılabilirlik ve iş yükü dengelemesi sağlar.  Uniform kümeleri IBM MQ for z/OS üzerinde kullanılamaz; kuyruk paylaşım grupları, tek bir kümenin yeteneklerinden pek çok özellik sağlar.

Tek biçimli kümeler, yüksek düzeyde kullanılabilir ve yatay olarak ölçeklenen bir kuyruk yöneticisi topluluğu sağlayan belirli bir IBM MQ kümesi kalıdır. Bu kuyruk yöneticileri, bir uygulamanın tek bir grup olarak etkileşimde bulunabilmesi için neredeyse aynı şekilde yapılandırılır. Bu, uygulama

eşgörünümlerinin kuyruk yöneticilerine eşit bir şekilde dağıtılmasını otomatik olarak sağlayarak, kümedeki her kuyruk yöneticisinin kullanılmasının kolaylaştırılmasını sağlar.

Tek biçimli kümeler, yöneticinin birbiriyle bağlantılı bağımsız, birbiriyle bağlantılı kuyruk yöneticilerini oluşturmak ve denetlemek için sahip olduğu el ile ilgili bazı adımları kaldırır. Bazı istemci bağlantısı mantığını istemciden kuyruk yöneticisine taşırlar; burada uygulama etkinliği düzeylerine ilişkin bilgiler, istemcilerde hangi kuyruk yöneticilerinin bağlanmaları gerektiği konusunda kararları bildirebilir.

**V 9.1.4** **Multi** Otomatik yapılandırma ve otomatik kümeleme desteğini kullanarak, tek bir küme kümesini ilk kez oluşturmayı basitleştirebilir ve daha sonra, yapılandırmayı tek tek tek tek küme üyeleri arasında saklayabilirsiniz. Bu yeteneği kullanırken, bir yapılandırma dosyası kümeyi açıklar ve başka bir tane de MQSC yapılandırmasını tek tip kümedeki tüm kuyruk yöneticilerine uygulanacak şekilde gösterir. Her kuyruk yöneticisi yeniden başlatılırsa, yapılandırma yeniden uygulanır ve küme otomatik olarak biçimlendirilir. Bu özelliğin kullanılmasına ilişkin ek bilgi için "[IBM MQ 9.1.4' dan tek biçimli bir küme oluşturma](#)" sayfa 76 ' e bakın.

Bir tekstilden tam olarak yararlanmak için, her uygulama aynı zamanda birden çok eşleşen birden çok eşgörünümle birden çok eşleşen eşgörünümlere de ölçeklenmelidir. Tercihen çok sayıda eşgörünüm varsa, en azından kuyruk yöneticisi olduğu kadar çok sayıda eşgörünüm olmalıdır.

Bir IBM MQ kümesi, her ne boyutta olursa olsun, birden çok yetenek sağlar:

- Bir kümedeki herhangi bir üye tarafından keşfedilebilen tüm kümeleme kaynaklarının bir dizini
- Otomatik kanal oluşturma ve bağlantılık
- İleti iş yükü dengeleme özelliğini kullanarak birden çok eşleşen kuyrukta yatay ölçekleme
- Dinamik ileti yöneltmesi, kullanılabilirliğe dayalı olarak

Tek biçimli kümeler, kuyruk yöneticileri arasındaki iletişim için IBM MQ kümelemesini ve kuyruklar arasındaki iş yükünün dengelenmesini sağlar. Ancak bunlar tipik IBM MQ kümelerinden aşağıdaki şekillerde farklılık gösterir:

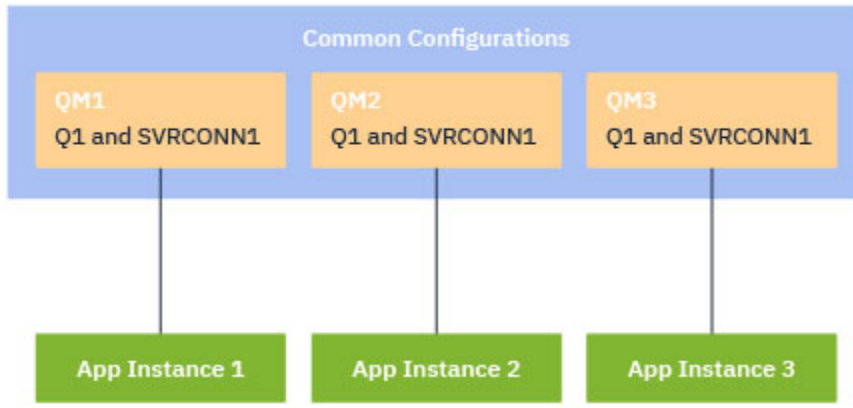
- Bir örnek kümelerin tipik olarak kümede daha az sayıda kuyruk yöneticisi vardır. 10 kuyruk yöneticisinden daha büyük bir tek tip küme oluşturmamalısınız.
- Kümenin her üyesinin aynı yapılandırmaya yakın bir yapılandırması vardır.
- Küme genellikle tek bir uygulama ya da ilgili uygulama grubu tarafından kullanılır.
- Kümeye bağlanan uygulama eşgörünümlerinin sayısı, kuyruk yöneticisi sayısı kadar, daha büyük ya da buna eşit olmalıdır.

Uygulama alma ve alma arasında iyi bir denge olmalıdır.

Bir tekbiçimli küme kalıbında, kümedeki tüm kuyruk yöneticileri aynı ileti alışverişi hizmetlerini sunar. Örneğin, tüm küme üyelerini tanımlanmış aynı yerel kuyruklara sahip olacak şekilde yapılandırabilir ve istemci uygulamalarının kümenin herhangi bir üyeye bağlanmasına olanak tanıyabilirsiniz. Ayrıca, aynı sunucu bağlantısı kanallarının da tanımlanmasını ve aynı yetki kayıtlarının, kanal doğrulama kurallarının vb. olmasını da isteyebilirsiniz. Ancak, küme üyelerinin nesnelere ve yapılandırmalarda bazı farklılıklar olabilir. Örneğin, bazı uygulamalar bir kuyruk yöneticisine bağlıyken geçici dinamik kuyruklar yaratabilirler. Ayrıca, bazı yapılandırma güncellemeleri, üyeler arasında bir süre içinde kullanıma sunulabilir; örneğin, yeni ya da güncellenmiş sertifikalar. Normal IBM MQ kümelerinde olduğu gibi, kuyruk yöneticilerinden ikisi, bunları tam havuz kuyruğu yöneticilerine yapmak için ek yapılandırma gerektirecektir.

Aşağıdaki çizge, kuyruk yöneticilerinin benzer yapılandırmalara sahip olduğunu göstermektedir. Bunlar Q1 adlı aynı kuyruğu ve SVRCONN1sunucu bağlantı kanalını tanımlarlar.





Aynı sunucu bağlantısı kanal adlarını tek bir istemci-kanal tanımlama çizelgesiyle (CCDT) çalışan birden çok kuyruk yöneticisi için, IBM MQ 9.1.2' ta tanımlanan güncellenmiş CCDT biçimini kullanmanız gerektiğini unutmayın. Bkz. [JSON biçimindeki istemci kanal tanımlama çizelgesi](#).

## Uygulama adları ve uygulama örnekleri

Bir uygulama adı, **DISPLAY CONN(\*) TYPE CONN** komutunun APPLTAG özniteliği olarak görüntülenir. IBM MQ 9.1.2' tan, uygulama adının belirlenme şeklinde bir değişiklik var.

**V 9.1.3** IBM MQ 9.1.3' tan, uygulamanın bir eşgörünümlü, o uygulama için bir *yürütme birimi* sağlayan, yakından ilişkili bağlantılar kümesidir. Tipik olarak, bu işlem tek bir işletim sistemi işletidir ve bir dizi iş parçacığını ve ilişkili IBM MQ bağlantılarını içerir.

Uygulama adı ve uygulama örneklerihakkında daha fazla bilgi için [Uygulama geliştirme kavramları](#) konusuna bakın.

## Yeniden bağlantısız istemciler

Tanımlamaya göre, yeniden bağlanamayan bir istemci farklı bir kuyruk yöneticisine yeniden bağlanamaz. Bir iş yükü dağılımına ulaşmak için yalnızca yeniden bağlanabilir istemciler taşınabilirler. Ancak, yeniden bağlanabilir olmayan bir istemciyi tek bir kümeye bağlamak için iyi bir neden olabilir: Örneğin, istemci bir tür kalıcı durum oluşturduğundan ve bazı başka bir mekanizma, uygulamanın her bir kuyruk yöneticisinde çalışmakta olan eşgörünümlerinin olmasını güvenirmek için kullanılır.

## Yerel olarak bağlı uygulamalar

Tek biçimli kümelerin, yerel olarak bağlı uygulamalar yerine, istemci uygulamaları olarak bağlanan IBM MQ uygulamalarına sahip olması beklenir. Yerel olarak bağlı uygulamaların tek tip küme üyelerine bağlanması önlenmez, ancak tek tip kümeler, kümenin başka herhangi bir üyesiyle bağlantı kuramadığından, yerel olarak bağlı uygulamalarla iş yükü dağılımını bile gerçekleştiremez.

### İlgili görevler

[Desteklenen programlama dillerinde uygulama adının belirtilmesi](#)

**V 9.1.2** **Multi** *Otomatik uygulama dengeleme*

IBM MQ 9.1.2' tan, otomatik uygulama dengelemesi, bir IBM MQ tekstili kümesinin, rasgele uygulamaya güvenmek yerine, küme genelinde uygulama dağıtımını yakından yönetmesini sağlayarak uygulama dağıtımını ve kullanılabilirliğini büyük ölçüde geliştirir ya da uygulamaların belirli kuyruk yöneticilerine el ile sabitlenmesini sağlar.

IBM MQ 9.1.2' tan, C dili uygulamaları, kümelenmiş kuyruk yöneticileri kümesi genelinde otomatik olarak dengelenir.

**V 9.1.3** IBM MQ 9.1.3' tan JMS uygulamaları, kümelenmiş kuyruk yöneticilerindeki bağlantıları otomatik olarak dengelemek için de kullanılabilir.

## V 9.1.4

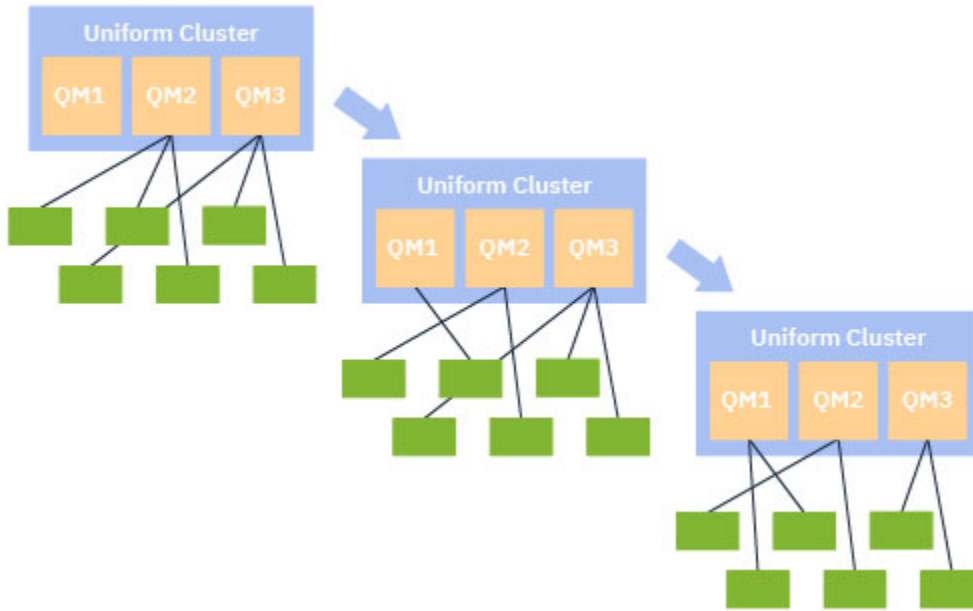
IBM MQ 9.1.4, IBM MQ .NET ve XMS .NET tarafından yönetilen uygulamalar, kümelendirilmiş kuyruk yöneticileri arasındaki bağlantıları otomatik olarak dengelemek için kullanılabilir.

Kuyruk yöneticisi olduğu gibi, aynı uygulamanın en az bir çok örneği olduğunda, tek biçimli küme sürekli olarak her kuyruk yöneticisinin uygulamanın en az bir eşgörünümünün bağlı olmasını güvenceye alır.

Uygulamalar, belirli bir benzeşimi kuyruk yöneticisine kaldırabilir ve bunun yerine, güvenli bir şekilde tek tip kümedeki kuyruk yöneticilerine bağlantılılığı rasgele hale getirebilmek için bir istemci kanal tanımlama çözümleri (CCDT) kullanabilirsiniz. Uygulamalar, aşağıdaki nedenlerden dolayı bunu yapabilir:

- Yeterli sayıda uygulama eşgörünümü varsa, uygulama işleme iletilerinin her zaman bir eşgörünümü vardır.
- Bir kuyruk yöneticisini durdurduğunuzda, bağlı olan tüm uygulama örnekleri, kümedeki kalan kuyruk yöneticilerine eşit olarak dağıtılır.
- Bir kuyruk yöneticisi başlattığınızda, kümedeki diğer kuyruk yöneticilerine bağlı olan uygulama örnekleri, yeni başlatılan kuyruk yöneticisini içerecek şekilde otomatik olarak yeniden dengelenir.

Bu, tek tip kümenin sürekli olarak uygulamaların en iyi şekilde dağıtılmasını, ileti işleme işlemini en üst düzeye çıkarmasını, planlanmış ve planlanmamış kesintiler durumunda bile uygulamaların en üst düzeye çıkarılmasını sağlar.



Otomatik dengeleme gerçekleştirmek için, tekbiçimli kümedeki kuyruk yöneticileri belirli aralıklarla kendi aralarında bilgileri paylaşır. Bunu, konu ağacının ayrılmış \$SYS/MQ dalı altındaki sistem konularındaki meta verileri yayınlayarak yapar. Tek biçimli küme içindeki her kuyruk yöneticisi, diğer kuyruk yöneticileri tarafından yayınlanan iletilere abone olur ve tek tip kümedeki uygulamaların bir resmini oluşturur.

Kuyruk yöneticileri, istemci uygulamalarının tüm küme boyunca dağıtımını izler. Belirli bir kuyruk yöneticisine bağlı uygulamaların sayısı, kümenin dengesiz olduğunu belirlediği kadar düşükse, o kuyruk yöneticisi bir sistem konusunda bir isteği yayımlar ve kümedeki diğer kuyruk yöneticilerinden birine yayımlar.

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 yöneticisiyle yeniden bağlantı kurar. Bu otomatik dengeleme düzeneği, uygulama için saydamdır. Daha fazla bilgi için bkz [“Otomatik dengeleme nasıl çalışır” sayfa 73.](#)

Bağlı uygulamalar hakkında meta veriler düzenli olarak dağıtılarak, tek biçimli küme, istemci uygulamalarının zaman içinde kuyruk yöneticilerine dengeli bir şekilde dengeli bir şekilde oranı elde



edebilir. Art arda gelen yeniden yönlendirme olaylarının oluşmasını önlemek için, otomatik dengeleme algoritması, yeniden yönlendirme isteklerinin yapıldığı hızı sınırlar.

**V 9.1.3** IBM MQ 9.1.3' tan, bir kümedeki kuyruk yöneticilerindeki uygulamaların yürürlükteki durumunu izleyebilir ve uygulama örneklerini izleyebilirsiniz. Daha fazla bilgi için bkz. [Uygulama dengelemesini izleme](#). Ayrıca, [Uygulama dengeleme sorunlarını gidermek](#) konusunda açıklandığı gibi, uygulama dengeleme ile çeşitli sorunları da çözebilirsiniz.

Yeniden dengeleme, uzun bağlantı süresi olan uygulamalar için yalnızca yararlı olur. Kısa bağlantı süreleri olan istemci uygulamalarınız varsa, örneğin, düzenli olarak farklı kuyruk yöneticilerine bağlanmak ve bağlantıyı kesmek için yazılan istemci uygulamaları, bunları yeniden bağlanmamaları için yapılandırmanızdır. Bu, bunları kuyruk yöneticilerinin dengelemeye çalışacağı uygulama kümesinden kaldırır.

### İlgili kavramlar

“Otomatik yeniden bağlanmaya ilişkin geliştirmeler” sayfa 75

IBM MQ 9.1.2'tan, tek biçimli küme otomatik dengelemesi, IBM MQ' un var olan otomatik yeniden bağlanma özelinde yapılan geliştirmelerin kullanımını sağlar.

**V 9.1.2** **Multi** *Otomatik dengeleme nasıl çalışır*

Tek biçimli kümede, istemci bağlantıları, uygulama adına dayalı olarak gruplanır. Aynı uygulama adını kullanan tek tip bir kümenin herhangi bir üyeye bağlanan uygulamalar, aynı uygulama adını kullanan diğer uygulamalara eşdeğeri olarak kabul edilir.

**V 9.1.3** Otomatik dengeleme, uygulama eşgörünümlerinin küme üyeleri arasında bile yayılmasını sağlar; daha fazla bilgi için [“Uygulama adları ve uygulama örnekleri” sayfa 71](#) konusuna bakın. Bir ya da daha çok uygulamanın ve uygulama eşgörünümlerinin durumunu, bir kuyruk yöneticisine ya da tek biçimli kümeye bağlı olarak görüntülemek için [DSP APSTATUS](#) komutunu kullanın.

**V 9.1.3** Örneğin, bir sigorta isteği uygulamasının tüm eşgörünümlerini bir "INSURANCE.REQUESTS". Bu uygulamadaki ilgili bağlantılar, her yönetim ortamı temelinde tüm dengeleme işleminin gerçekleştirildiği durumlarda, otomatik olarak, uygun olan durumlarda gruplanacaktır.

Uygulamanın yeni eşgörünümleri tek biçimli kümenin bir üyeye bağlandığında, otomatik dengeleme algoritması hangi kuyruk yöneticilerinin en az INSURANCE.REQUESTS ve bazı bağlantıları bu kuyruk yöneticilerine yeniden yönlendirir.

Otomatik dengeleme yalnızca aşağıdaki koşullarda etkinleştirilir:

- **V 9.1.3** Kanalın SHARECNV değeri sıfırdan büyük.
- Aşağıdakilerden biri doğrudur:
  - İstemci uygulaması MQCNO\_RECONNECT ' yi belirtiyor
  - mqclient.ini dosyası, **Defrecon**=YES değerini belirtir.

**Not:** **V 9.1.3** Kuyruk yöneticisi benzerliği olan uygulamalar-örneğin, kalıcı bir abonelik ya da kuyruğun dinamik yanıtı nedeniyle-güvenli bir şekilde yeniden dengelenemez ve MQCNO\_RECONNECT\_QMGR kullanmalı ya da hiçbir yeniden bağlanma seçeneği kullanılmamalıdır.

Bir istemci diğer bir kuyruk yöneticisine yeniden yönlendirildiğinde, her zamanki gibi, yeni hedefe ilişkin bağlantı bilgilerini bulmak için yerel istemci kanal tanımlama çizelgelerini (CCDTs) kullanın. Bu nedenle, otomatik dengeleme işleminin sorunsuz ve verimli çalışması için, istemcilerin tek tek kümenin her bir üyesi için bir giriş içeren CCDT ' yi ve ilk bağlantıları dengelemek için kullanılan kuyruk yöneticisi grubunu kullanmaktadır.

Bir JSON biçimi CCDT' nin kullanılması, aynı sunucu bağlantısı adını kullanan birden çok bağlantıya izin verdiği için bunu basitleştirir. Ek bilgi için [JSON biçimindeki istemci kanal tanımlama çizelgesi](#) başlıklı konuya bakın.

### İlgili kavramlar

“Otomatik yeniden bağlanmaya ilişkin geliştirmeler” sayfa 75

IBM MQ 9.1.2'tan, tek biçimli küme otomatik dengelemesi, IBM MQ' un var olan otomatik yeniden bağlanma özelinde yapılan geliştirmelerin kullanımını sağlar.

V 9.1.3

ULW

### JMS uygulamalarının otomatik olarak dengelenmesi

Java Message Service (JMS) uygulamaları otomatik olarak dengelendiğinde, JMS uygulamalarının oluşturduğu temel IBM MQ bağlantıları grupları birlikte taşınır.

For handling JMS connections, uniform clusters have the concept of an *uygulama eşgörünümlü*. JMS için bir *uygulama örneği*, JMS Bağlantısı olarak tanımlanır ve tüm ilişkili JMS Oturumları olarak tanımlanır.

A unique connection tag is allocated on the client connection that corresponds to the JMS Connection, and the same tag is then applied to the client connections that correspond to JMS Sessions that are created by that JMS Connection.

Örneğin, bir istemci uygulaması çifti, tek bir etkin kuyruk yöneticisi (Kuyruk Yöneticisi 1) olan bir tekbiçimli kümeye yönelik JMS uygulamalarını çalıştırıyorsa:

- İstemci 1, üzerinde "App1" uygulama adını belirlediği bir bağlantı üreticisi yaratır ve bir JMS Connection ve üç JMS Oturumu yaratır. İstemci 1, aynı bağlantı etiketini paylaşan her biri Kuyruk Yöneticisi 1 'de dört istemci bağlantısı yaratır ve bu, tek bir "App1" eşgörünümlü olarak ele alınır.
- İstemci 2 ayrıca, üzerinde "App1" uygulama adını belirlediği bir bağlantı üreticisi yaratır ve bir JMS Connection ve iki JMS Oturumu yaratır. İstemci 2, aynı bağlantı etiketini (ayrı olarak Client 1 'e atandan ayrı) paylaşan her biri üç istemci bağlantısı yaratır ve bu, "App1" için tek, ayrı bir örnek olarak kabul edilir.
- Kuyruk yöneticisi, "App1" nin iki eşgörünümlünü görür.

Otomatik dengeleme gerçekleştirildiğinde, uygulama örnekleri taşınır. Kuyruk yöneticisi bir uygulama yönetim ortamını (aynı bağlantı etiketini paylaşan istemci bağlantıları grubu) seçer ve yönetim ortamının farklı bir kuyruk yöneticisine taşınmasını ister. İstemci kodu isteği alır ve ilgili tüm bağlantıların ( JMS Connection ve ilişkili JMS Oturumlarına karşılık gelen) yeni kuyruk yöneticisine taşınmasını sağlar.

Örneğin, önceden sıralanan uygulama örnekleri kümesini ele alın ve yeni bir kuyruk yöneticisinin (Kuyruk Yöneticisi 2) tek tip kümede başlatıldığını varsayın.

Kuyruk Yöneticisi 2 'nin çalışmıyor ancak Kuyruk Yöneticisi 1 'in "App1" 2 eşgörünümlü var, bu nedenle Kuyruk Yöneticisi 2, Kuyruk Yöneticisi 1 'in bir "App1" eşgörünümlünü Kuyruk Yöneticisi 2 'ye aktarıp aktardığını ister.

Kuyruk Yöneticisi 1, taşınmak için bir "App1" örneğini seçer. Örneğin, bu örnek için, istemci 1 'in yarattığı yönetim ortamını seçeceğini varsayın.

- Kuyruk Yöneticisi 1, "App1" yönetim ortamını QM2' ye taşımak için Müşteriye 1 'e bir istek gönderir.
- İstemci, var olan dört istemci bağlantısını Kuyruk Yöneticisi 1 'e kapatır ve Kuyruk Yöneticisi 2 'ye dört yeni bağlantı oluşturur.
- The JMS Connection and its JMS Sessions, except for a short pause in processing, should not normally be disturbed.

#### **Not:**

Bir uygulama eşgörünümlünün taşındığı sırada devam etmekte olan belirli işlemler varsa, bir uygulama JMS kural dışı durumunu alabilir.

JMS kural dışı durumu, hatanın nedenini belirlemek için neden kodunun alınabileceği bağlantılı bir IBM MQ kural dışı durumuna sahip olacaktır.

Beklenen neden kodları şunlardır:

#### **MQR\_CALL\_INTERRUPT**

Bu durum, örneğin, kalıcı (JMS ' deki varsayılan) bir eşitleme noktasının dışına konduğunda, ancak işlem yeniden bağlantı ile kesintiye uğradığında oluşur.

#### **MQR\_BACKED\_OUT**

Bu durum, örneğin, bir eşitleme noktasının içine bir ileti koyma girişimi yeniden bağlantıyla kesintiye uğradığında oluşur.

## İlgili kavramlar

“Otomatik dengeleme nasıl çalışır” sayfa 73

Tek biçimli kümede, istemci bağlantıları, uygulama adına dayalı olarak gruplanır. Aynı uygulama adını kullanan tek tip bir kümenin herhangi bir üyeye bağlanan uygulamalar, aynı uygulama adını kullanan diğer uygulamalara eşdeğeri olarak kabul edilir.

“Otomatik yeniden bağlanmaya ilişkin geliştirmeler” sayfa 75

IBM MQ 9.1.2'tan, tek biçimli küme otomatik dengelemesi, IBM MQ' un var olan otomatik yeniden bağlanma özelinde yapılan geliştirmelerin kullanımını sağlar.

V 9.1.2

Multi

*Otomatik yeniden bağlanmaya ilişkin geliştirmeler*

IBM MQ 9.1.2'tan, tek biçimli küme otomatik dengelemesi, IBM MQ' un var olan otomatik yeniden bağlanma özelinde yapılan geliştirmelerin kullanımını sağlar.

IBM MQ 9.1.2'öncesindeki IBM MQ sürümlerinde, otomatik yeniden bağlanma özelliği otomatik olarak bir kuyruk yöneticisinin yedek yönetim ortamına ya da sağlanan bağlantı ayrıntılarına dayalı olarak farklı bir kuyruk yöneticisine yeniden bağlanır; bu özellik genellikle bir bağlantı adı listesi ya da bir İstemci Kanal Tanımlama Çizelgesi (CCDT) olabilir.

The IBM MQ client, in some circumstances silently performs the reconnection without the application being aware it has happened. Hangi kuyruk yöneticisinin yeniden bağlanacağı karar, bir bağlantı adı listesindeki bağlantı adları sırasına ya da CCDT 'deki iş yükü dengelemesi yapılarındaki kararlara tümüyle inmektedir.

IBM MQ 9.1.2 'tan, istemcinin yeniden bağlanması gereken kuyruk yöneticisinin ipucu içeren bir istemciye yeniden bağlanma isteği gönderilmesini olanaklı. Kuyruk yöneticisi hatası gibi birçok yeniden bağlanma senaryolarında ya da **endmqm -x**komutunu veren yönetici, ipucu bilgilerine bir kuyruk yöneticisi adı eklenmez ve otomatik yeniden bağlanma davranışı şu anda olduğu gibi çalışır.

Ancak, tek tip bir küme yapılandırdıysanız, otomatik uygulama dengeleme düzenli olarak dengeli bir küme elde etmek için istemcilere yeniden bağlanma istekleri gönderir. Bu durumda, tek tip küme, istemci bağlantılarının en az bağlantıya sahip kuyruk yöneticilerine taşınmasını sağlamak için yeniden bağlantı ipucunda bir kuyruk yöneticisi adı belirtir.

Otomatik dengelemenin çalışması için aşağıdakilerin önemli olması önemlidir:

- IBM MQ uygulamaları, bağlantı bilgilerini almak için CCDT 'leri kullanır.
- CCDTs, tek tip kümedeki her kuyruk yöneticisi için bir giriş içerir

Bu durumda, kümenin, uygulamaları kümenin tüm üyeleri arasında otomatik olarak dengelemesi olanaklı değildir.

If an application is using a version of the IBM MQ client that is earlier than IBM MQ 9.1.2, and it is configured to support automatic client reconnection, then it might be sent a request by the uniform cluster to go through its reconnect steps.

İstemcinin belirli bir kuyruk yöneticisine yeniden bağlanması istenmez, ancak bunun yerine, diğer yeniden bağlanma olayları için olduğu yeniden bağlanma mantığı sırası aynı olur. İstemcilerin, kümenin her üyesi için eşit ağırlıklı girdiler içeren CCDT 'leri kullanacak şekilde yapılandırıldığından emin olarak, istemci uygulamalarının tek biçimli küme üzerinden IBM MQ 9.1.2 'dan önce dağıtılması mümkündür.

Uygulamalar, ek yönetim ortamının gereksinim duyduğu bir kuyruk yöneticisine bağlanmadan önce, birkaç yeniden bağlanma girişimi gerçekleştirebilirler ve bu girişim, uygulamaların küme boyunca eşit olarak dağıtılması açısından daha verimli bir yöntemdir. Otomatik dengeleme, bu ortamlarda daha uzun sürebilir.

## IBM MQ istemcileri otomatik istemci yeniden bağlantı özelliğini desteklemiyor

Bir uygulama, otomatik istemci yeniden bağlantısını desteklemeyen bir IBM MQ istemcisi sürümünü kullanıyorsa, uygulama bir MQI çağrısından başarısızlık dönüş kodu alabilir.

Uygulamanız hataları işlemek ve el ile yeniden bağlantı gerçekleştirmek için tasarlanmadıysa, bu uygulamalar için otomatik dengelemeyi devre dışı bırakmak gerekebilir.

**Not:** Yeniden bağlanabilir (reconnectable) olarak tanımlanan herhangi bir uygulama için otomatik dengeleme etkinleştirilir; bu, etkin bağlantı seçeneklerinde uygulamanın MQCNO\_RECONNECT ' ye sahip olduğunu sağlar.

### İlgili görevler

“IBM MQ 9.1.2 ya da IBM MQ 9.1.3 içinde tek biçimli bir küme oluşturma” sayfa 80

IBM MQ 9.1.2' ta, belirli bir IBM MQ kümesinin qm.ini dosyasında bir ayarlama parametresi ayarlayarak tek tip bir küme olarak değerlendirileceğini belirtmiş olmanız gerekir. Bir kuyruk yöneticisinin herhangi bir sayıda standart IBM MQ kümesinin üyesi olmasına rağmen, en çok bir tek tip kümeye katılabileceği unutulmamalı.

### ULW V 9.1.4 IBM MQ 9.1.4' dan tek biçimli bir küme oluşturma

IBM MQ 9.1.4' tan, basit bir küme kurulumu ve yapılandırması biçimi kullanabilirsiniz. Otomatik yapılandırma ve otomatik kümeleme desteğini kullanarak, hem tek bir küme kümesinin ilk yaratılışının hem de konfigürasyonunun aynı şekilde tek tek aynı küme üyeleri arasında kalmasını basitleştirebilirsiniz.

## Başlamadan önce

Tek tip bir küme oluşturmadan önce “Tek tip kümeler için sınırlamalar ve dikkat edilecek noktalar” sayfa 83' u okumalısınız.

## Bu görev hakkında

From IBM MQ 9.1.4, you indicate that a particular IBM MQ cluster is to be treated as a uniform cluster by supplying in the qm. ini file a section for AutoCluster with at least **Type=Tekbiçimli** and **ClusterName=< tek biçimli küme adı >**.

İsteğe bağlı olarak, temeldeki IBM MQ kümesini, *otomatik küme yaratma* özelliğini kullanarak aynı . ini stanza aracılığıyla yapılandırabilirsiniz. Kümeyi ayarlamak için bu otomatik küme desteğini kullanırken, kümeyi ve tam havuzlarını açıklayan bir yapılandırma dosyası sağlıyoruz.

Başlatılmakta olan kuyruk yöneticisi, tam havuzlardan biri olarak listeleniyorsa, otomatik olarak tam bir havuz yapılır. Benzer şekilde, küme alıcı kanalı tanımlandığında, küme gönderen kanalları tam havuza ya da havuzlara otomatik olarak veri tanımlıdır.

## Yordam

Tek tip bir küme gerektiren ek işlevlerden yararlanmak için aşağıdaki adımlardan birini tamamlamanız gerekir:

- Var olan bir kümeyi tek bir kümeye dönüştür, which meets the pattern described in “Tek biçimli kümeler” sayfa 69.
- Bu amaçlayeni bir tek tip küme oluşturun .

### ULW V 9.1.4 Yeni bir tekbiçimli küme oluşturma

IBM MQ 9.1.4' tan yeni bir uniform kümesi nasıl yaratıyorsunuz?

## Yordam

1. Kümenin, tam havuzların terimlerini nasıl göreceğini açıklayan bir dosya oluşturun.

Herhangi bir küme için, iki tam havuz, kümeyle ilgili bilgi merkezi depoları olarak işlev görmektedir. Özellikle, bu kümedeki iki tam havuz için ad ve bağlantı adlarını tanımlamanız gerekir.

**Not:** Bu, yaratılmakta olan herhangi bir şeyden (kuyruk yöneticileri dahil) önceden yapılıyor ve aşağıdaki işlem, bu kuyruk yöneticilerinin yaratılmasını içerir.

Örneğin, QMA, QMB, QMC ve QMD kuyruk yöneticisi üyeleriyle birlikte, UNICLUS adında bir tek tip bir küme kurduğınızı hayal edin. Bu örnekte, QMA ve QMB, kısmi havuzlar olarak QMC ve QMD ' ler ile tam havuz 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 gönderenleri (CLUSDR) tanımlamak için diğer küme üyeleri için *conname* özniteliği olarak kullanılır ve çok eşgörünümlü bir kuyruk yöneticisine ilişkin bir bağlantı listesi olabilir ve isteğe bağlı olarak bağlantı noktasını içerebilir.

2. Tüm küme üyelerine uygulanmasını istediğiniz MQSC tanımlarını içeren bir örnek yapılandırma kütüğü uniclus . mqsc yaratın.

Bu dosyada, otomatik küme adının (genellikle + AUTOCL + insert ile) CLUSTER özniteliği ve + QMNAME + insert ' i içeren bir kanal adı ile bir küme alıcı kanalı (CLUSTVVR) tanımı olan bu dosyada zorunlu bir hat gerekir.

Bu, tek tip küme üyelerinin her bir kuyruk yöneticisine nasıl bağlanacağını ve diğer kuyruk yöneticilerine nasıl bağlanacağını gösteren bir şablon olarak nasıl kullanıldığını gösterir. 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 tip kümedeki tüm kuyruk yöneticilerinde aynı olmasını sağlamak için CLUSTER, CONNAME ve KANAL alanlarındaki bazı ek eklemeler kullanabilir. Bu bilgiler şunları içerir:

**+ AUTOCL +**

Otomatik küme adı

**+ QMNAME +**

Yaratılmakta olan kuyruk yöneticisinin adı

**+ KONADI +**

Kuyruk yöneticisi yaratma sırasında, bağlantı adı değiştirilmesi dizisinde kullanılmak üzere **-iv** değiştirilmesini kullanarak ya da Variables qm.ini stanza içinde tanımlanmış bir değişken. Değişkenin adı herhangi bir değer olabilir.

Kanal adlarının 20 karakterle sınırlı olduğunu unutmayın; bu nedenle, hem içindeki eklemeler, hem de eklemeler değiştirildiğinde, bu sınırlamaya uyması gereken bir değer de vardır. Örnek bir dosya aşağıdaki gibi olabilir:

```
*#####
* 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(CLIENTCHL) chltype(svrconn) trptype(tcp) replace
```

3. Bu iki dosyayı, tek bir küme üyesi barınacak her bir makinede kullanılabilir duruma getiriniz. Örneğin, /shared/uniclus . ini ve /shared/uniclus . mqsc.
4. Bu makinelerin her birinde kuyruk yöneticisini yaratın. Komut satırında şunu belirtin:

- Beklenen kapıda bir dinleyici başlatma isteği
- Otomatik INI yapılandırması (**-ii**) için otomatik küme ayarları dosyasını (uniclus.ini) işaret eden bir istek
- Tek tip küme için CLUSRCVR tanımlamasını içeren MQSC yapılandırma dosyasını gösteren bir otomatik MQSC yapılandırması isteği (**-ic**).
- Bu kuyruk yöneticisine ilişkin bir CONNAME.

QMA anasisteminde:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNNAME=QMA.dnsname(1414) QMA
strmqm QMA
```

Tek biçimli kümedeki her kuyruk yöneticisi, neredeyse aynı bir komut satırı ile yaratılır-tam ile kısmi havuz arasındaki tüm farklar, tek bir küme için otomatik olarak ele alınır.

QMB 'nin anasisteminde:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNNAME=QMB.dnsname(1414) QMB
strmqm QMB
```

QMC 'nin anasisteminde:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNNAME=QMC.dnsname(1414) QMC
strmqm QMC
```

QMD anasisteminde:

```
crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNNAME=QMD.dnsname(1414) QMD
strmqm QMD
```

Otomatik olarak ne olur:

Kuyruk yöneticisi başlatıldığı için, unicus.ini dosyasındaki tanımlamalar qm.ini dosyasına uygulanır. Daha fazla bilgi için bkz. [Başlangıçta qm.ini 'nin otomatik yapılandırması](#). Bu, **AutoCluster** tanımlamasını qm.ini dosyasına ekler.

Kuyruk yöneticisi **AutoCluster** stanza içinde tam havuzlardan biri olarak adlandırıldıysa, otomatik olarak tam bir havuz olacak şekilde dönüştürülür. Bu durumda, MQSC komutu ALTER QMGR REPOS (**ClusterName**) çalıştırılır; tersi durumda, ALTER QMGR REPOS ("") MQSC komutu verilmesine benzer şekilde kısmi bir havuza dönüştürülür.

Otomatik küme için küme günlük nesnesi kanalı tanımlandığında, küme gönderen kanalları bu kuyruk yöneticisinden **AutoCluster** stanza içindeki tüm tüm havuzlara tanımlanır (bu, tam havuzlardan biriye, yerel kuyruk yöneticisi hariç). Bu gönderen kanalları, tanımlanmış olan yerel küme alıcısından tüm ortak kanal özniteliklerini devralır.



**Uyarı:** Kanallar daha fazla manuel müdahale olmadan yaratılsa da, bunlar diğer kanal tanımlamalarında görüntülenebilen ve yönetilebilen yönetimsel kanal nesnelidir. İleti trafiğini yönlendirmek için, bu nesnelere 'otomatik olarak tanımlanmış' küme gönderen kanallarıyla karıştırmamanız gerekir.

## Sonraki adım

### Tek tip küme ayarının doğrulanması

**ClusterName** parametresi doğru şekilde ayarlandığında ve kuyruk yöneticisi adı belirtilen kümenin bir üyesi ise, kümenin artık tek bir küme olarak tanımlandığını onaylamak için AMQ9883 iletisi yayınlanır.

Daha sonra, otomatik uygulama dengeleme gibi tek tip küme yeteneklerini kullanabilirsiniz. During queue manager startup, if this parameter has been set, but the name is not a valid IBM MQ cluster name, the name is ignored and error message AMQ9882 issued.

Ad geçerli bir küme adıysa, ancak tanımlanan küme için herhangi bir küme kanalı yoksa, yöneticinizin bu durumu saptamasına ve düzeltilmesine izin vermek için kuyruk yöneticisi hata günlüğüne uyarı iletisi AMQ9881 yayınlanır.

### Otomatik küme ayarının doğrulanması

Tek tip kümeyi ayarlamak için otomatik küme desteği kullandıysanız, tam havuz olarak belirtilen kuyruk yöneticilerinin şu anda 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 yapılandırılmasa da:

```
QMC:
  1 : dis qmgr repos
AMQ8408I: Display Queue Manager details.
  QMNAME(QMC)                REPOS( )
```

Ayrıca, küme gönderen kanallarını (CLUSDR) her bir kuyruk yöneticisinden diğer tam havuzlara, kanal adını yapılandırma MQSC dosyasından kullanarak da görebilmemiz gerekir:


```
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(CLUSDR)
  CONNAME(QMB.dnsname(1414))

QMC:
  1 : dis chl(UNICLUS*) conname
AMQ8414I: Display Channel details.
  CHANNEL(UNICLUS_QMA)        CHLTYPE(CLUSDR)
  CONNAME(QMA.dnsname(1414))
AMQ8414I: Display Channel details.
  CHANNEL(UNICLUS_QMB)        CHLTYPE(CLUSDR)
  CONNAME(QMB.dnsname(1414))
AMQ8414I: Display Channel details.
  CHANNEL(UNICLUS_QMC)        CHLTYPE(CLUSRCVR)
  CONNAME(QMC.dnsname(1414))
```

### İlgili kavramlar

[“Tek biçimli kümeler” sayfa 69](#)



Tek bir küme devreye alımın amacı, uygulamaların ölçekleme ve kullanılabilirlik için tasarlanabileceği ve tek tip küme içindeki kuyruk yöneticilerine bağlanabilmesi. Bu, belirli bir kuyruk yöneticisinde bağımlılığı

kaldırır ve ileti alışverişi trafiğindeki daha iyi kullanılabilirlik ve iş yükü dengelemesi sağlar. 

Uniform kümeleri IBM MQ for z/OS üzerinde kullanılamaz; kuyruk paylaşım grupları, tek bir kümenin yeteneklerinden pek çok özellik sağlar.

[“Tek tip kümeler için sınırlamalar ve dikkat edilecek noktalar” sayfa 83](#)

Tek tip kümeleri yapılandırırken dikkate alınacak sınırlamalar ve diğer noktalar.

  *Var olan bir kümenin tek bir kümeye dönüştürülmesi*

IBM MQ 9.1.4' tan, var olan bir kümeyi tek bir kümeye dönüştürmek için bu yordamı kullanabilirsiniz.

## Bu görev hakkında

Var olan bir kümeyi tek bir kümeye dönüştürseniz, kuyruk yöneticileri arasında uygulamaların dengelenmesini desteklemek için gerekli herhangi bir tanımın tüm küme üyesinde var olduğundan emin olmanız gerekir.

## Yordam

1. Tüm kuyruk yöneticilerine uzak (kümelenmiş) yayınlama da dahil olmak üzere IBM MQ yayınlama aboneliğini etkinleştirin.

Bu, tek tip küme işlevselliği için bir önkoşuldur; bu nedenle, kuyruk yöneticisinin PSMODE ve PSCLUS özneliklerinin her ikisi de ENABLED varsayılan değerine ayarlandığından emin olmanız gerekir.

2. Add an **AutoCluster** section to the qm.ini file to the name of the IBM MQ cluster, as used in your MQSC object definitions, such as cluster channels.


Örneğin, kümenin adı UNICILUS ise, qm.ini dosyalarınızda AutoCluster stanza ekleyin ya da aşağıdaki gibi değiştirin:

```
AutoCluster:  
ClusterName=UNICLUS  
Type=Uniform
```

3. Yeni ayarı uygulamak için kuyruk yöneticilerini yeniden başlatın.
4. Tüm tek biçimli küme üyelerinin başlatma işlemi sırasında aynı yapılandırmaya sahip olmasını sağlamak için otomatik yapılandırmayı bir mekanizma olarak kullanmayı düşünün.  
Ayrıntılı bilgi için [Başlangıçta bir MQSC komut dosyasından otomatik yapılandırma başlıklı konuya](#) bakın.

## İlgili kavramlar

[“Tek biçimli kümeler” sayfa 69](#)

Tek bir küme devreye almanın amacı, uygulamaların ölçekleme ve kullanılabilirlik için tasarlanabileceği ve tek tip küme içindeki kuyruk yöneticilerine bağlanabilmesi. Bu, belirli bir kuyruk yöneticisinde bağımlılığı kaldırır ve ileti alışverişi trafiğindeki daha iyi kullanılabilirlik ve iş yükü dengelemesi sağlar.  Uniform kümeleri IBM MQ for z/OS üzerinde kullanılamaz; kuyruk paylaşım grupları, tek bir kümenin yeteneklerinden pek çok özellik sağlar.


[“Tek tip kümeler için sınırlamalar ve dikkat edilecek noktalar” sayfa 83](#)

Tek tip kümeleri yapılandırırken dikkate alınacak sınırlamalar ve diğer noktalar.

 *IBM MQ 9.1.2 ya da IBM MQ 9.1.3 içinde tek biçimli bir küme oluşturma*

IBM MQ 9.1.2' ta, belirli bir IBM MQ kümesinin qm.ini dosyasında bir ayarlama parametresi ayarlayarak tek tip bir küme olarak değerlendirileceğini belirtmiş olmanız gerekir. Bir kuyruk yöneticisinin herhangi bir sayıda standart IBM MQ kümesinin üyesi olmasına rağmen, en çok bir tek tip kümeye katılabileceği unutulmamalı.

## Başlamadan önce

 *IBM MQ 9.1.4' tan bir tek tip küme oluşturmak için bkz. [“Yeni bir tekbiçimli küme oluşturma” sayfa 76.](#)*

Tek tip bir küme oluşturmadan önce [“Tek tip kümeler için sınırlamalar ve dikkat edilecek noktalar” sayfa 83' u](#) okumalısınız.

## Bu görev hakkında

Tek tip bir küme gerektiren ek işlevlerden yararlanmak için şunları yapmak gerekir:

- [“Tek biçimli kümeler” sayfa 69' ta](#) açıklanan örüntüye uygun olan bir kümeyi tanımlayın ya da bu amaçla yeni bir küme yaratın.



- Tek biçimli kümeye katılan tüm kuyruk yöneticilerinin bir düzey destekleyen tek bir kümeyi destekleyen düzeye yükseltildiğinden emin olun.

## Yordam


- Tüm kuyruk yöneticilerine uzak (kümelenmiş) yayınlama da dahil olmak üzere IBM MQ yayınlama aboneliğini etkinleştirin.  
Bu, tek tip küme işlevselliği için bir önkoşuldur; bu nedenle, kuyruk yöneticisinin PSMODE ve PSCLUS özniteliklerinin her ikisi de *ETKIN* varsayılan değerine ayarlandığından emin olmanız gerekir.
- qm.ini dosyasında, küme kanalları gibi MQSC nesne tanımlarınızda kullanılan IBM MQ kümesinin adına UniformClusterAd özniteliğini ayarlayın.  
Örneğin, kümenin adı *UNICLUS* ise, qm.ini dosyalarınıza *TuningParameters* stanza adını aşağıdaki gibi ekleyin ya da değiştirin:

```
TuningParameters:  
  UniformClusterName=UNICLUS
```

- Yeni ayarı uygulamak için kuyruk yöneticilerini yeniden başlatın.  
UniformClusterAd parametresi doğru olarak ayarlandığında ve kuyruk yöneticisi adlandırılmış kümenin bir üyesi ise, kümenin artık tek bir küme olarak tanımlandığından emin olmak için AMQ9883 iletisi yayınlanır. Daha sonra, otomatik uygulama dengeleme gibi tek tip küme yeteneklerini kullanabilirsiniz.  
During queue manager startup, if this parameter has been set, but the name is not a valid IBM MQ cluster name, the name is ignored and error message AMQ9882 issued.  
Ad geçerli bir küme adıysa, ancak tanımlanan küme için herhangi bir küme kanalı yoksa, yöneticinizin bu durumu saptamasına ve düzeltmesine izin vermek için kuyruk yöneticisi hata günlüğüne uyarı iletisi AMQ9881 yayınlanır.

## İlgili kavramlar

“Tek biçimli kümeler” sayfa 69

Tek bir küme devreye almanın amacı, uygulamaların ölçekleme ve kullanılabilirlik için tasarlanabileceği ve tek tip küme içindeki kuyruk yöneticilerine bağlanabilmesi. Bu, belirli bir kuyruk yöneticisinde bağımlılığı kaldırır ve ileti alışverişi trafiğindeki daha iyi kullanılabilirlik ve iş yükü dengelemesi sağlar. 

Uniform kümeleri IBM MQ for z/OS üzerinde kullanılamaz; kuyruk paylaşım grupları, tek bir kümenin yeteneklerinden pek çok özellik sağlar.

“Tek tip kümeler için sınırlamalar ve dikkat edilecek noktalar” sayfa 83

Tek tip kümeleri yapılandırırken dikkate alınacak sınırlamalar ve diğer noktalar.

  Otomatik küme yapılandırmasını kullanma

From IBM MQ 9.1.4, you configure IBM MQ to enable automatic configuration by changing the qm.ini configuration information.

**Not:** Tek tip kümeler için yalnızca AutoCluster stanza 'yı kullanabilirsiniz.

## Yapılandırılacak stanzas

Aşağıdaki stanzaları değiştirebilirsiniz:

### AutoConfig

qm.ini dosyasında tanımlanır. Kuyruk yöneticisi başlatıldığında, hangi otomatik yapılandırma dosyalarının uygulanmaya başlanacak olduğunu tanımlar.

Tek tip kümeler kullanıldığında aynı küme yapılandırmasını dağıtmak için bu mekanizmayı kullanmanız gerekir.

### AutoCluster

qm.ini dosyasında tanımlanır. Kuyruk yöneticisi, kümenin otomatik bir kümenin üyesi olup olmadığını belirlemeye ve kümenin tam havuzlarını tanımlayıp tanımlayamadığında kullanılır.

## Değişkenler

qm.ini dosyasında tanımlanır. Bazı kuyruk yöneticisi değişkenleri içerir.

## AutoConfig stanza öznelikleri

AutoConfig stanza içinde aşağıdaki iki özneliğe izin verilir:

### **MQSCConfig=< Path>**

The path is either a full file path, or path to a directory, where all files \*.mqsc are applied to the queue manager, on every queue manager start.

Daha fazla bilgi için bakınız: [Automatic configuration from an MQSC script at startup.](#)

### **IniConfig=< Path>**

The path is either a full file path, or path to a directory, where all files \*.ini are applied to the qm.ini file, on every queue manager start.

Daha fazla bilgi için bkz. [Başlangıçta qm.ini 'nin otomatik yapılandırması.](#)

Bu öznelikler, tek tip kümelerin ayarlanması kapsamında sıklıkla kullanılır. Daha fazla bilgi için bakınız: [Creating a uniform cluster from IBM MQ 9.1.4.](#)

Örnek Stanza:

```
AutoConfig:
MQSCConfig=C:\MQ_Configuration\uniclus.mqsc
IniConfig=C:\MQ_Configuration\uniclus.ini
```

## AutoCluster stanza için öznelikler

AutoCluster kısmı için aşağıdaki öznelikler zorunlu olur:

### **Tip =*Birimform***

Otomatik küme tipini belirtir ve geçerli tek seçenek, tek tip bir kümeyi temsil eden *Uniform*' dur.

### **ClusterName=< String>**

Otomatik küme adı olan kümenin adı.

Yukarıdaki özneliklerin varlığı, tek tip kümeler için uygulama dengelemesini etkinleştirir. Ek ayrıntılar için [Otomatik uygulama dengelemesi](#) konusuna bakın.

Buna ek olarak, küme bu stanza içinde açıklansa, bir kümenin basitleştirilmiş olarak ayarlanması gerçekleştirilebilir. Daha fazla bilgi için bakınız: [Creating a uniform cluster from IBM MQ 9.1.4.](#) Bu kullanılırsa, iki kuyruk yöneticisi adını verebilir ve bu otomatik küme için tam havuzlara ilişkin bağlantı adlarını verebilirsiniz.

Aşağıdaki öznelikler, AutoCluster stanza için isteğe bağlıdır, ancak bunları çift olarak sağlamalısınız:

### **HavuzAdı1 =< Dizi>**

Bu, otomatik kümeden ilk tam havuzun kuyruk yöneticisi adıdır. Bu, bu kuyruk yöneticisinin adı olabilir ya da başka bir kuyruk yöneticisi olabilir.

### **Repository1Conname=< Bağlantı Adı Dizisi >**

Bu, otomatik kümenin üyelerinin bu kuyruk yöneticisine nasıl bağlanacağını içeren bağlantı adı (CONNAME) değeridir.

Buna ek olarak, küme için ikinci bir tam havuz tanımlayabilirsiniz:

### **Repository2Name=< Dizi>**

### **Repository2Conname=< Bağlantı adı dizisi >**

Örnek Stanza:

```
AutoCluster:
Repository2Conname=myFR1.hostname(1414)
Repository2Name=QMFR1
```

```
Repository1Conname= myFR2.hostname(1414)
Repository1Name=QMFR2
ClusterName=UNICLUS
Type=Uniform
```

## Değişkenlere ilişkin öznitelikler kısmı

Öznitelik alanında bir `attribute=value` çifti geçerli olur. Bunlar, bir kuyruk yöneticisi yarattığınızda, `crtmqm` komutundaki `-iv` komut satırı seçeneği kullanılarak sağlanabilir.

CONAD ' in otomatik küme yapılandırması sırasında ve bir küme alıcı kanalının kanal adı MQSC alanlarının otomatik küme yapılandırması sırasında Değişkenler kısmında listelenen öznitelikleri kullanabilirsiniz.

**V 9.1.2** **Multi** *Tek tip kümeler için sınırlamalar ve dikkat edilecek noktalar*

Tek tip kümeleri yapılandırırken dikkate alınacak sınırlamalar ve diğer noktalar.

**Not:** Tek tip kümeleri yapılandırırken genel gereksinimler için bkz. [“IBM MQ 9.1.2 ya da IBM MQ 9.1.3 içinde tek biçimli bir küme oluşturma” sayfa 80.](#)

Burada , kuyruk yöneticisi yapılandırmasının tutarlılığını içeren bir asayiş uygulaması yoktur. Benzer bir yapılandırmaya sahip olması için kümenin üyelerini yapılandırmak sistem yöneticinize bağlı olur. Bu, herhangi bir kaynağın (örneğin, kuyruk, konu ya da yetki kaydı), yeniden dengelenebilecek bir uygulamanın gerektirdiği herhangi bir kaynağın, tek biçimli kümedeki tüm kuyruk yöneticilerine bildirilmesi gerekir.

**V 9.1.5** If you are running a uniform cluster with an earlier Continuous Delivery release, during migration from IBM MQ 9.1.4 to IBM MQ 9.1.5, while some queue managers are running the older, and some the newer release, applications will not necessarily be evenly balanced around the cluster. Tüm kuyruk yöneticileri yeni düzeye geçirildikten sonra olağan dengeleme işlemi sürdürülür.

**V 9.1.5** IBM MQ 9.1.5 kuyruk yöneticilerini içeren bir tek tip kümeyi çalıştırıyorsanız, bir yönetici, kuyruk yöneticilerini daha eski bir yayın düzeyinden, tek biçimli kümeye tanıtmamalıdır. Yani, tüm kuyruk yöneticilerinin IBM MQ 9.1.5 konumunda olması gerekir. Bir denetimci daha önceki bir yayından bir kuyruk yöneticisi tanıyorsa, MQRCCF\_CLUSTER\_TOPIC\_CONFLICTER hata kodunu içeren bir FDC yayınlanır.

**V 9.1.4** IBM MQ 9.1.4' ta, otomatik küme yeteneğiyle birlikte otomatik yapılandırma yeteneğini kullanarak tutarlılığı yardımcı olabilirsiniz. Küme tam havuzlarını önceden yapılandırarak, her bir kuyruk yöneticisi başlangıcındaki yeniden uygulanan tek bir MQSC komut dosyası olması mümkündür ve bunların tümünün aynı tanımlara sahip olmasını sağlayarak, tek bir MQSC komut dosyası, tek bir MQSC komut dosyası başlatıp yeniden kullanılabilir. Bununla ilgili ayrıntılar için [“IBM MQ 9.1.2 ya da IBM MQ 9.1.3 içinde tek biçimli bir küme oluşturma” sayfa 80](#) ' e bakın. aksadı.

**V 9.1.4** Tek biçimli küme kuyruk yöneticisi en çok bir tek tip kümeye katılabilir ve bu küme, standart IBM MQ kümesinin herhangi bir sayıdaki üyesi de olabilir.

**V 9.1.4** Ancak, tek bir küme kuyruk yöneticisi, tek tip bir kümenin kendisinden başka bir şey için tam havuz olmamalıdır. Bunun nedeni, iki örtük olarak yapılandırılmış birden çok yapılandırılmış kuyruk yöneticisi sayısının artırılması, dış küme için önerilen iki tam havuzdan daha fazlasını tanımladığından, [“Tüm havuzları tutmak için küme kuyruğu yöneticilerinin nasıl seçileceği” sayfa 32](#) ' e bakın. Tek bir tam havuz kuyruk yöneticisini tek biçimli bir kümeyle değiştirmeye çalışıyorsanız, üzerinde çalışan uygulama çalışmasından tüm havuzu ayırmalısınız ve yalnızca uygulama işini tek tek kümeyle taşımanız gerekir.

**V 9.1.4** Tek biçimli kümeler için otomatik tanımlamalar kullanılırken, küme kanalları başka kümelerde kullanılmak üzere paylaşılabilir; yani, CLUSTER öznitelikli otomatik kümeyle ayarlayıp CLUSNL öznitelikli boş olması gerekir.

Otomatik dengeleme, istemci uygulamalarının kendilerini, tek biçimli kümenin üyeleri arasında eşit olarak dengelemesine neden olur. **V 9.1.3** IBM MQ 9.1.3' tan, uygulama eşgörünümlerinin ayrıntılarını ve bunların dengeli olup olmadığını, [DISPLAY APSTATUS](#) runmqsc komutunu kullanarak görüntüleyebilirsiniz.

IBM MQ 9.1.3' tan, bir kümedeki kuyruk yöneticilerindeki uygulamaların yürürlükteki durumunu izleyebilir ve uygulama örneklerini izleyebilirsiniz. Daha fazla bilgi için bkz. [Uygulama dengelemesini izleme](#).

Durumlar, özellikle aşağıdaki durumlarda eşit şekilde dengeli bir şekilde dengeli geçmediklerinde bazen de olabilir:

- Kümede kuyruk yöneticilerine göre daha az sayıda uygulama örneği var.

- İstemci uygulamalarının kümeye ya da sola, kümeye bağlandıktan kısa bir süre sonra.

İstemci uygulamalarının çok sık yeniden dengelenmesini önlemek için, özellikle de uygulama bağlantıları geldiğinde ve devam ederken, bu sınırlar, tek bir kümenin istemci uygulamalarının yeniden yönlendirilmesini istediğinde ne sıklıkta bir sınır kümesinin sormasını sağlar. Bir yüksek bağlantı ya da bağlantı kesme etkinleştirmesinden sonra, kalan uygulama eşgörünümlerinin tek biçimli küme boyunca eşit olarak dengeli olması birkaç dakika sürebilir.

- Uygulamaların belirli bir kuyruk yöneticisine benzeşimi varsa.

Yalnızca MQCNO\_RECONNECT değerini belirten uygulamalar yeniden dengelenebilir. Belirli bir kuyruk yöneticisine ilişkin bir benzeşme olan uygulamalar yeniden bağlanma ya da MQCNO\_RECONNECT\_Q\_MGRdeğerini belirtmemelidir.

Örneğin, kalıcı bir abonelik yaratan bir uygulama, kümenin başka bir üyeye yeniden yönlendirilebilir. Yaratılan kalıcı abonelik için ileti almak için, uygulama yönetim ortamının MQCNO\_RECONNECT\_Q\_MGR komutunu kullanarak kümedeki aynı kuyruk yöneticisine bağlanması gerekir.

Tek biçimli kümenin her bir üyesi, küme yöneticisinin başarısız olması nedeniyle, küme kuyruğuna bir küme kuyruğuna yöneltilmiş iletiler gibi, kümenin üyesine özgü tüm durumların çok uzun süre kullanılmaması için, bu kümenin her üyesi yüksek düzeyde kullanılabilir olacak şekilde yapılandırılmalıdır. Örneğin, tek biçimli küme içindeki her kuyruk yöneticisi, başarısız olursa kapsayıcı bir düzenleme hizmeti tarafından yeniden başlatılacak bir taşıyıcı içinde çalışabilir.

Diğer bir seçenek olarak, her bir kuyruk yöneticisi bir IBM MQ çok eşgörünümlü kuyruk yöneticisi olarak yapılandırılabilir ya da başka bir yüksek kullanılabilirlik çözümlerinden yararlanabilir. Hata durumunda yedek sisteme geçiş sırasında belirli bir kuyruk yöneticisinde iletilerin kaybedilmediğinden emin olmak için güvenilir bir depolama seçeneği kullanmanız gerekir.



**Uyarı:** Özellikle önerilen özelliklere sahip olmayan bir kümede tek biçimli küme davranışının etkinleştirilmesi, özellikle çok sayıda kuyruk yöneticisi içeren kümelerin ciddi bir performans etkisine sahip olması olasılığının yüksek olması durumunda.

### İlgili kavramlar

[“Otomatik uygulama dengeleme” sayfa 71](#)

IBM MQ 9.1.2' tan, otomatik uygulama dengelemesi, bir IBM MQ tekstili kümesinin, rasgele uygulamaya güvenmek yerine, küme genelinde uygulama dağıtımını yakından yönetmesini sağlayarak uygulama dağıtımını ve kullanılabilirliğini büyük ölçüde geliştirir ya da uygulamaların belirli kuyruk yöneticilerine el ile sabitlenmesini sağlar.

## Dağıtılmış yayınlama/abone olma ağınızın planlanması

Bir kuyruk yöneticisinde yaratılan aboneliklerin ağ üzerindeki başka bir kuyruk yöneticisine bağlı bir uygulama tarafından yayınlanan eşleşen iletileri alacağı bir kuyruk yöneticisi ağı yaratabilirsiniz. Uygun bir topoloji seçmek için, el ile denetim, ağ boyutu, değişiklik sıklığı, kullanılabilirlik ve ölçeklenebilirlik gereksinimlerinizi göz önünde bulundurmanız gerekir.

### Başlamadan önce

Bu görev, dağıtılmış yayınlama/abone olma ağlarının ne olduğunu ve bunların nasıl çalıştığını anladığınızı varsayar. Teknik bilgiler için bkz. [Distributed publish/subscreen networks](#).

### Bu görev hakkında

Bir yayınlama/abone olma ağı için üç temel topoloji vardır:

- Doğrudan yönlendirilen küme
- Konu anasistem yönlendirmesi
- Sıradüzen

İlk iki topoloji için, başlangıç noktası bir IBM MQ kümesi yapılandırmasıdır. Üçüncü topoloji, bir küme ile ya da bir küme olmadan oluşturulabilir. Temeldeki kuyruk yöneticisi ağını planlamaya ilişkin bilgi için bkz. [“Dağıtılmış kuyruklarınızın ve kümelerinizin planlanması” sayfa 19.](#)

*Doğrudan yönlendirilen küme*, bir küme zaten varsa, yapılandırılacak en basit topolojidir. Herhangi bir kuyruk yöneticisinde tanımladığınız herhangi bir konu, kümedeki her kuyruk yöneticisinde otomatik olarak kullanıma sunulur ve yayınlar doğrudan bir yayınlama uygulamasının bağlandığı herhangi bir kuyruk yöneticisinden, eşleşen aboneliklerin bulunduğu kuyruk yöneticilerine yönlendirilir. Bu yapılandırmanın basitliği, IBM MQ ' un kümedeki her kuyruk yöneticisi arasında bilgi ve bağlantılılığın yüksek düzeyde paylaşılmasını sağlar. Küçük ve basit ağlar için (yani, az sayıda kuyruk yöneticisi ve oldukça statik bir yayıncı ve abone grubu) bu kabul edilebilir. Ancak, daha büyük ya da daha dinamik ortamlarda kullanıldığında genel giderler engelleyici olabilir. Bkz. [“Yayınlama/abone olma kümelerinde doğrudan yönlendirme” sayfa 90.](#)

Bir *Konu anasistem yönlendirilmiş küme*, kümedeki her kuyruk yöneticisinde otomatik olarak kullanılabilir olan herhangi bir kuyruk yöneticisinde tanımladığınız herhangi bir konuyu, doğrudan yönlendirilen bir küme ile aynı yararını verir. Ancak, konu anasistem yönlendirmesi kümeleri, her konuyu barındıran kuyruk yöneticilerini dikkatli bir şekilde seçmenizi gerektirir; çünkü, o konuya ilişkin tüm bilgiler ve yayınlar bu konu anasistem kuyruk yöneticilerinden geçer. Bu, sistemin, tüm kuyruk yöneticileri arasında kanalları ve bilgi akışlarını korumak zorunda kalmadığı anlamına gelir. Bununla birlikte, yayınların doğrudan abonelere gönderilmediği, ancak bir konu ana makine kuyruk yöneticisiyle yönlendirilmiş olabileceği anlamına da gelir. Bu nedenle, özellikle konuları barındıran kuyruk yöneticilerine ek yük bindirilebilir. Bu nedenle, topolojinin dikkatli bir şekilde planlanması gereklidir. Bu topoloji, özellikle çok sayıda kuyruk yöneticisi içeren ya da dinamik bir yayıncı ve abone kümesi (yani, sık eklenen ya da kaldırılmış olan yayıncılar ya da aboneler) barındıran ağlar için etkilidir. Rotaların kullanılabilirliğini artırmak ve yayın iş yükünü yatay olarak ölçeklemek için ek konu anasistemleri tanımlanabilir. Bkz. [“Yayınlama/abone olma kümelerinde konu anasistem yönlendirmesi” sayfa 95.](#)

Bir *Sıradüzen*, en çok el ile yapılandırmanın ayarlanmasını gerektirir ve değiştirilecek en zor topolojidir. Sıradüzendeki her bir kuyruk yöneticisi arasındaki ilişkileri ve doğrudan ilişkileri el ile yapılandırmalısınız. İlişkiler yapılandırdıktan sonra, yayınlar, sıradüzendeki diğer kuyruk yöneticilerindeki aboneliklere yönetilir (önceki iki topolojiye göre). Yayınlar, sıradüzen ilişkileri kullanılarak yönlendirilir. Bu, çok özel topolojilerin farklı gereksinimlere uyacak şekilde yapılandırılmasını sağlar; ancak, aboneliklere ulaşmak için ara kuyruk yöneticileri aracılığıyla birçok "sekme" gerektiren yayınlarla da sonuçlanabilir. Bir yayın sıradüzeni boyunca her zaman yalnızca bir rota vardır; bu nedenle, her kuyruk yöneticisinin kullanılabilirliği kritik önem göstermektedir. Sıradüzenler genellikle tek bir kümenin yapılandırılmadığı durumlarda tercih edilir; örneğin, birden çok kuruluş için yayılma. Bkz. [“Yayınlama/abone olma sıradüzenlerinde yönlendirme” sayfa 119.](#)

Gerekli olduğunda, yukarıda belirtilen üç topoloji belirli topografik gereksinimleri çözmek için birleştirilebilir. Örneğin, [Birden çok kümenin konu alanlarının birleştirilmesibaşlıklı konuya](#) bakın.

Dağıtılmış yayınlama/abone olma ağınıza için uygun bir topoloji seçmek üzere aşağıdaki geniş soruları göz önünde bulundurmanız gerekir:

- Aığınız ne kadar büyük olacak?
- Yapılandırması için ne kadar el ile denetime gereksinim duyarsınız?
- Sistem, hem konular açısından, hem de abonelikler açısından, hem de kuyruk yöneticisi açısından dinamik bir sistem olacak mı?
- Kullanılabilirlik gereksinimleriniz ve ölçeklenebilirlik gereksinimleriniz nelerdir?
- Tüm kuyruk yöneticileri birbirleriyle doğrudan bağlantı kurabilirler mi?

## Yordam

- Aığınızın ne kadar büyük olması gerektiğini tahmin edin.

- a) Kaç konu gerektiğini tahmin edin.
- b) Ne kadar yayıncı ve abone beklediğinizi tahmin edin.
- c) Yayınlama/abone olma etkinliklerine kaç kuyruk yöneticisinin katılacağını tahmin edin.

Ayrıca bkz. "Kümelemeyi yayınlama/abone olma: En iyi uygulamalar" sayfa 104, özellikle de aşağıdaki kısımlar:

- Sisteminizin büyüklüğü
- Yayınlama/abone olma etkinleştirmede yer alan küme kuyruğu yöneticilerinin sayısını sınırlamak için nedenler
- Hangi konuların kümelemeceğine nasıl karar vereceğine

Ağınızda çok sayıda kuyruk yöneticisi varsa ve birçok yayıncı ve aboneyi ele geçirirse, büyük olasılıkla bir konu anasistemi kümesini ya da bir sıradüzenini kullanmanız gerekir. Doğrudan yönlendirilen kümeler, neredeyse el ile yapılandırma gerektirmez ve küçük ya da statik ağlar için iyi bir çözüm olabilir.

- Hangi kuyruk yöneticisinin her konuyu, yayınlayıcıyı ya da aboneyi barındırması için gereken el ile denetimi göz önünde bulundurun.
  - a) Kuyruk yöneticilerinizin bazılarının diğerlerinden daha az yetenekli olup olmadığını göz önünde bulundurun.
  - b) Kuyruk yöneticilerinizin bazılarına yönelik iletişim bağlantılarının diğerlerine göre daha kırılabilir olup olmadığını göz önünde bulundurun.
  - c) Bir konunun birçok yayın ve birkaç aboneye sahip olmasını beklediğiniz durumları tanımlayın.
  - d) Bir konunun birçok aboneye ve birkaç yayınına sahip olmasını beklediğiniz durumları tanımlayın.

Tüm topolojilerde, yayınlar diğer kuyruk yöneticilerindeki aboneliklere teslim edilir. Doğrudan yönlendirilen bir kümede bu yayınlar, aboneliklerin en kısa yolunu alır. Bir konu ana makinesinde ya da bir sıradüzeninde, yayınların üstlendiği rotayı kontrol edebilirsiniz. Kuyruk yöneticilerinizin yetenekleri farklıysa ya da farklı düzeylerde kullanılabilirlik ve bağlantırlık düzeyleri varsa, belirli iş yükleri için belirli kuyruk yöneticilerine atamak isteyebilirsiniz. Bu işlemi, bir konu anasistemi kümesini ya da bir sıradüzenini kullanarak yapabilirsiniz.

Tüm topolojilerde, mümkün olan her yerde abonelikler olarak aynı kuyruk yöneticisinde yayınlama uygulamalarını birlikte bulma ve performansı en üst düzeye çıkarma işlemi. Konu anasistem yönlendirmeleri için, yayıncıları ya da aboneleri, konuyu barındıran kuyruk yöneticilerine koymayı deneyin. Bu işlem, bir yayını bir aboneye iletmek için kuyruk yöneticileri arasında fazladan "sekme" 'ı kaldırır. bu yaklaşım özellikle bir konunun birçok yayıncısı ve az sayıda abonesi veya çok sayıda abonesi ve birkaç yayıncısı olduğu durumlarda etkilidir. Örneğin, Topic anasistem yöneltmesi merkezileştirilmiş yayıncılar ya da aboneler kullanılarak kullanılabilir.

Ayrıca bkz. "Kümelemeyi yayınlama/abone olma: En iyi uygulamalar" sayfa 104, özellikle de aşağıdaki kısımlar:

- Hangi konuların kümelemeceğine nasıl karar vereceğine
- Yayınlayıcı ve abonelik yeri
- Ağ etkinliğinin ne kadar dinamik olacağını değerlendirin.
  - a) Abonelerin farklı konulara nasıl ekleneceğini ve kaldırılacağını tahmin edin.

Bir abonelik eklendiğinde ya da bir kuyruk yöneticisinden bir abonelik kaldırıldığında ve bu, ilgili konu dizisine ilişkin ilk ya da son abonelikse, bu bilgiler topolojideki diğer kuyruk yöneticilerine iletilir. Doğrudan yönlendirilen bir kümede ve bir sıradüzeninde, bu abonelik bilgileri topoloji içindeki her kuyruk yöneticisine yayınlanır ya da konu üzerinde yayıncı olup olmadıkları yayılır. Topoloji birçok kuyruk yöneticisinden oluşuyorsa, bu, önemli bir performans yükü olabilir. Bir konu anasistem kümesindeki küme içinde, bu bilgi yalnızca, aboneliğin konu dizisiyle eşlenen, kümelenmiş bir konuyu barındıran kuyruk yöneticilerine yayılır.

Ayrıca, "Kümelemeyi yayınlama/abone olma: En iyi uygulamalar" sayfa 104' un Abonelik değişikliği ve dinamik konu dizileri bölümüne de bakın.

**Not:** Çok sayıda benzersiz konu dizgisinin hızla ve sürekli olarak değiştiği çok dinamik sistemlerde, modeli bir "her yerde yayınla" kipine geçirmek en iyi yol olabilir. Bkz. [Yayınlama/abone olma ağlarındaki abonelik performansı](#).

b) Kuyruk yöneticilerinin topolojide ne kadar dinamik olduğunu göz önünde bulundurun.

Bir hiyerarşi, topolojideki kuyruk yöneticisindeki her değişikliğin, hiyerarşide daha yüksek düzeylerde kuyruk yöneticileri değiştirilirken dikkatli bir şekilde el ile eklenmesine ya da sıradüzenden kaldırılmasını gerektirir. Bir sıradüzendeki kuyruk yöneticileri genellikle el ile yapılandırılmış kanal bağlantılarını kullanır. Kuyruk yöneticisi eklenip sıradüzenden kaldırılıp kaldırıldığı için, bu bağlantıları korumalı, kanalları ekleme ve kaldırma işlemi gerekir.

Bir yayınlama/abone olma kümesinde, kuyruk yöneticileri otomatik olarak, kümeye ilk katıldıklarında gerekli olan başka bir kuyruk yöneticisine bağlanır ve konular ve aboneliklerden otomatik olarak haberdar olur.

- Rota uygunluğunuzu ve yayın trafiği ölçeklenebilirlik gereksinimlerini göz önünde bulundurun.
  - a) Bir kuyruk yöneticisi kullanılmadığında bile, bir yayınlama kuyruk yöneticisinden abone olan bir kuyruk yöneticisinden her zaman kullanılabilir bir rotaya sahip olmanız gerekir gerekmediğine karar verin.
  - b) Ağ için ne kadar ölçeklenebilir olduğunu göz önünde bulundurun. Yayın trafiği düzeyinin tek bir kuyruk yöneticisi ya da kanal üzerinden yönlendirilmesinin çok yüksek olup olmadığına ve bu yayın trafiğindeki düzeyin tek bir konu dalı tarafından mı işleneceğini, yoksa birden çok konu dalına yayılıp yayılmayacağı konusunda karar verin.
  - c) İleti sıralamasını korumanın gerekir gerekmediğini dikkate alın.

Doğrudan yönlendirilen bir küme, kuyruk yöneticilerine abone olmak üzere doğrudan yayınlama kuyruğu yöneticilerinden iletiler gönderdiğinden, yol boyunca ara kuyruk yöneticilerinin kullanılabilirliğini göz önünde bulundurmanıza gerek yoktur. Benzer şekilde, ara kuyruk yöneticilerine ölçeklendirme de dikkate alınmamaktadır. Ancak, daha önce de belirtildiği gibi, kanalların ve bilgi akışlarının kümedeki tüm kuyruk yöneticileri arasında otomatik olarak sürdürülmesi, özellikle büyük ya da dinamik bir ortamda performansı önemli ölçüde etkileyebiliyor.

Bir konu anasistem yönlendirmesi, tek tek konular için ayarlanabiliyor. Önemli bir yayın iş yükü olan konu ağacının her bir dalının farklı bir kuyruk yöneticisine tanımlandığından ve her kuyruk yöneticisinin yeterli bir performans gösterdiğinden ve konu ağacının o dalı için beklenen iş yükü için kullanılabilir olduğunu doğrulayabilirsiniz. Ayrıca, birden çok kuyruk yöneticisi üzerinde her konuyu tanımlayarak, kullanılabilirliği ve yatay ölçeklemeyi daha da geliştirebilirsiniz. Bu, sistemin, kullanılmayan konu anasistem kuyruk yöneticilerine ve bunların genelinde iş yükü dengelemesi yayınına yol göstermesine olanak tanır. Ancak, birden çok kuyruk yöneticisine bir konu tanımlarken, aşağıdaki koşulları da tanıtırınız:

- Yayınlarda ileti sıralaması kaybedersiniz.
- Alikonan yayınları kullanamazsınız. Bkz. "[Yayınlama/abone olma kümelerinde tutulan yayınlarda dikkat edilmesi gereken noktalar](#)" sayfa 117.

Birden çok rota aracılığıyla bir sıradüzende yöneltme üst düzeyde kullanılabilirliğini ya da ölçeklenebilirliğini yapılandıramazsınız.

Ayrıca, "[Kümelemeyi yayınlama/abone olma: En iyi uygulamalar](#)" sayfa 104' un [Yayın trafiği](#) bölümüne de bakın.

- Bu hesaplamalara dayalı olarak, bir konu anasistemi kümesini, doğrudan yönlendirilmiş bir kümeyi, bir sıradüzeni ya da bu topolojilerin bir karışımı kullanıp kullanmayacağınıza karar vermenize yardımcı olması için sağlanan bağlantıları kullanın.

## Sonraki adım

Artık dağıtılmış yayınlama/abone olma ağını yapılandırmak için hazırsınız.

### İlgili görevler

[Kuyruk yöneticisi kümesinin yapılandırılması](#)

[Dağıtılmış kuyruklama yapılandırılması](#)



## **Yayınlama/abone olma kümelerini tasarlama**

İki temel yayınlama/abone olma kümesi topolojisi vardır: *doğrudan yönlendirme* ve *konu ana makine yönlendirmesi*. Her birinin farklı yararları vardır. Yayınlama/abone olma kümenizi tasarladığınızda, beklenen ağ gereksinimlerinize en uygun topolojiyi seçin.

İki yayınlama/abone olma kümesi topolojisine genel bakış için [kümeleri yayınla/abone olmabaşlıklı](#) konuya bakın. Ağ gereksinimlerinizi değerlendirmenize yardımcı olmak için bkz. "[Dağıtılmış yayınlama/abone olma ağınızın planlanması](#)" sayfa 84 ve "[Kümelenmeyi yayınlama/abone olma: En iyi uygulamalar](#)" sayfa 104.

Genel olarak, her iki küme topolojisi de aşağıdaki avantajları sağlar:

- Noktadan noktaya küme topolojisinin üstündeki basit yapılandırma.
- Kuyruk yöneticilerinin kümeyi otomatik olarak işleme ve kümeyi bırakması.
- Ek kuyruk yöneticileri ekleyerek ve ek abonelikler ve yayıncılar arasında dağıtılarak ek abonelikler ve yayınlayıcılar için ölçeklendirme kolaylığı.

Ancak, iki topoloji de gereksinimlerin daha belirgin hale gelmesi açısından farklı avantajlar elde eder.

## **Doğrudan yönlendirilen yayınlama/abone olma kümeleri**

Doğrudan yöneltme ile, kümedeki herhangi bir kuyruk yöneticisi bağlı uygulamalardan yayınlar gönderir; bu yayınlar, eşleşen bir abonelikle kümedeki diğer bir kuyruk yöneticisine doğrudan yönlendirir.

Doğrudan yönlendirilen bir yayınlama/abone olma kümesi aşağıdaki yararları sağlar:

- Aynı kümedeki belirli bir kuyruk yöneticisinde bulunan bir aboneliğin gönderileceği iletiler doğrudan o kuyruk yöneticisine taşınır ve ara kuyruk yöneticisinde geçişe gerek yoktur. Bu, bir konu anasistemi yönlendirilmiş topolojisiyle ya da sıradüzensel bir topolojiyle karşılaştırmada başarıyı artırabilir.
- Tüm kuyruk yöneticileri doğrudan birbirine bağlandığından, bu topolojinin yöneltme altyapısında tek bir hata noktası yoktur. Bir kuyruk yöneticisi yoksa, kümedeki diğer kuyruk yöneticilerindeki abonelikler hala var olan kuyruk yöneticilerindeki yayıncılardan ileti alabilirler.
- Özellikle var olan bir kümede olmak üzere yapılandırılacak çok basittir.

Doğrudan yönlendirilen yayınlama/abone olma kümesi kullanılırken dikkate alınacak şeyler:

- Kümedeki tüm kuyruk yöneticileri, kümedeki diğer tüm kuyruk yöneticilerinden haberdar olur.
- Kümelenmiş bir konuya bir ya da daha fazla aboneliği barındıran bir kümedeki kuyruk yöneticileri, küme yöneticileri kümelenmiş herhangi bir konu üzerinde ileti yayınlamasa da, otomatik olarak küme gönderen kanalları kümedeki tüm diğer kuyruk yöneticilerine otomatik olarak yaratılır.
- Kümelenmiş bir konu altındaki bir konu dizgisine kuyruk yöneticisinin ilk aboneliği, kümedeki diğer tüm kuyruk yöneticisine gönderilen bir iletiyle sonuçlanır. Benzer şekilde, bir konu dizgisinin de silineceği son abonelik, bir iletiyle sonuçlanabilir. Kümelenmiş bir konu altında kullanılmakta olan tek tek konu dizgileri ve aboneliklerin değişiklik oranı ne kadar yüksekse, kuyruk yöneticisi iletişiminin daha fazla olması gerekir.
- Kümedeki her kuyruk yöneticisi, abone olunan konu dizgilerinin bilgi sahibi olduğunu ve kuyruk yöneticisi bu konulara abone olmasanız da, bu dizgilerle ilgili bilgi edinir.

Yukarıdaki nedenlerden dolayı, doğrudan yönlendirilmiş bir konuya sahip bir kümedeki tüm kuyruk yöneticileri ek bir ek yüke yol açacaktır. Daha fazla kuyruk yöneticisi kümede var, ek yük ne kadar büyük. Aynı şekilde, abone olunan konu dizgileri de, ne kadar büyükse o kadar çok değişim, genel giderler de o kadar fazladır. Bu, büyük ya da dinamik bir doğrudan yönlendirilen yayınlama/abone olma kümesinde küçük sistemlerde çalışan kuyruk yöneticilerine çok fazla yük getirilebilir. Ek bilgi için [Doğrudan yönlendirilen yayınlama/abone olma performansı](#) başlıklı konuya bakın.



Bir kümenin doğrudan yönlendirilen kümelenmiş yayınlama/abone olma genel başkanlarını barındıramamasını biliyorsanız, bunun yerine `topic host routed publish/subscrbseçeneğini` kullanabilirsiniz. Alternately, in extreme situations, you can completely disable clustered publish/subscribe functionality by setting the queue manager attribute **PSCLUS** to DEVRE Dışı on every queue manager in the cluster. Bkz. "[Kümelenmiş yayınlama/abone olma engelini engelle](#)" sayfa 115. Bu, kümelenmiş herhangi bir konunun yaratılmasını önler ve bu nedenle, ağınızın kümelenmiş yayınlama/abone olma ile ilişkili herhangi bir üst tura maruz kalmamasını sağlar.

## Konu anasistem yayınlama/abone olma kümelerini yönlendirdi

Konu anasistem yönlendirmesiyle, kümelenmiş konuların yönetsel olarak tanımlandığı kuyruk yöneticileri yayınlar için yönlendiriciler olur. Kümedeki barındırma dışı kuyruk yöneticilerinden gelen yayınlar, barındırma kuyruk yöneticisi aracılığıyla, eşleşen bir abonelikte kümedeki herhangi bir kuyruk yöneticisine yöneltilmektedir.

Bir konu ana makinesi yayınlama/abone olma kümesi, doğrudan yönlendirilen bir yayınlama/abone olma kümesi üzerinden aşağıdaki ek avantajları sağlar:

- Kümedeki diğer tüm kuyruk yöneticilerinden haberdar olmak üzere, yalnızca konu anasistem yöneltmesi konuları üzerinde bulunan kuyruk yöneticileri vardır.
- Yalnızca, ana makine kuyruğu yöneticilerinin kümedeki diğer tüm kuyruk yöneticilerine bağlanabilmesi ve genellikle yalnızca aboneliklerin var olduğu kişilere bağlanabilmesi gerekir. Bu nedenle, kuyruk yöneticileri arasında çok daha az kanal çalıştırılıyor.
- Kümelenmiş bir konuya bir ya da daha fazla aboneliği barındıran küme kuyruğu yöneticileri, otomatik olarak küme gönderen kanalları oluşturarak yalnızca, aboneliğin konu dizisiyle eşlenen bir küme konusuna sahip olan kuyruk yöneticilerine otomatik olarak küme göndericisi kanalları oluşturur.
- Kümelenmiş bir konu altındaki bir konu dizisine bir kuyruk yöneticisinin ilk aboneliği, kümelenmiş konuyu barındıran kümedeki bir kuyruk yöneticisine gönderilen bir iletiyle sonuçlanıyor. Benzer şekilde, bir konu dizisinin de silineceği son abonelik, bir iletiyle sonuçlanabilir. Kümelenmiş bir konu altında kullanılmakta olan tek tek konu dizileri ve aboneliklerin değişiklik oranı ne kadar yüksekse, kuyruk yöneticisi iletişimi daha fazla olur, ancak yalnızca abonelik anasistemleri ve konu anasistemleri arasında gerçekleşir.
- Fiziksel yapılandırmayla ilgili daha fazla denetim. Doğrudan yöneltme ile, tüm kuyruk yöneticileri yayınlama/abone olma kümesine katılmak zorundaylar ve bunların baş başlıklarını artırmalarını sağlar. Konu anasistem yönlendirmesiyle, yalnızca konu anasistem kuyruk yöneticileri diğer kuyruk yöneticilerinin ve aboneliklerinin farkındadır. Konu anasistem kuyruk yöneticilerini belirttik olarak seçebildiğinizden, bu kuyruk yöneticilerinin yeterli ekipman üzerinde çalıştığından emin olabilirsiniz ve diğer kuyruk yöneticileri için daha az güçlü sistemler kullanabilirsiniz.

Bir konu ana makine tarafından yönlendirilen yayınlama/abone olma kümesi kullanılırken dikkate alınacak şeyler:

- Yayıncı ya da abone, kuyruk yöneticisini barındıran bir konu üzerinde bulunmadığında, yayınlama kuyruğu yöneticisi ile abone olan bir kuyruk yöneticisi arasında ek bir "sekme noktası" tanımlanır. Ek "sekme" 'in neden olduğu gecikme süresi, konu anasistem yöneltmesinin doğrudan yönlendirmenin daha az verimli olması anlamına gelebilir.
- Büyük kümelere, konu ana makine yönlendirmesi, doğrudan yöneltme ile alabileceğiniz önemli performans ve ölçkleme sorunlarını hafifler.
- Tüm konularınızı tek bir kuyruk yöneticisiyle ya da çok az sayıda kuyruk yöneticisi üzerinde tanımlamayı seçebilirsiniz. Bunu yaparsanız, ana makine kuyruk yöneticilerinin güçlü sistemlerde iyi bağlanabilirlik ile barındırıldığından emin olun.
- Aynı konuyu birden çok kuyruk yöneticisinden tanımlayabilirsiniz. Bu, konunun kullanılabilirliğini artırır ve ayrıca, IBM MQ iş yükü, bir konuya ilişkin yayınları o konu için tüm anasistemlerde bulunan bir konuya ilişkin olarak bakiylediğinden, ölçeklenebilirliği artırır. Ancak, aynı konuyu birden çok kuyruk yöneticisinden tanımlayan, bu konuya ilişkin ileti sırasını kaybeder.
- Farklı kuyruk yöneticilerine farklı konuları barındırarak, ileti sırasını kaybetmeden ölçeklenebilirliği geliştirebilirsiniz.

## **İlgili görevler**

Yayınlama/abone olma kümesi yapılandırılması

Dağıtılmış yayınlama/abone olma ağlarının ayarlanması

Dağıtılmış yayınlama/abone olma sorun giderme

## **İlgili başvurular**

Küme senaryolarını yayınla/abone ol

## ***Yayınlama/abone olma kümelerinde doğrudan yönlendirme***

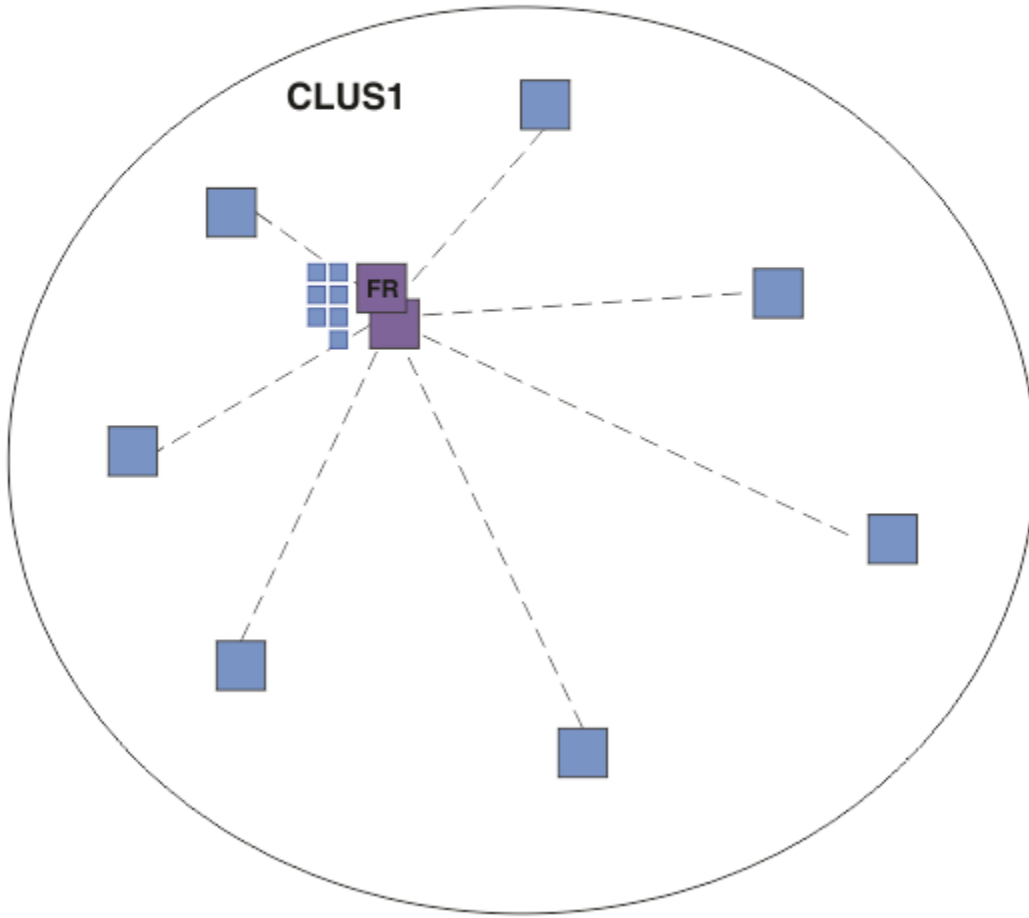
Herhangi bir yayınlama kuyruğu yöneticisinden gelen yayınlar, eşleşen bir aboneliğe sahip kümedeki diğer herhangi bir kuyruk yöneticisine doğrudan yönlendirilir.

Yayınlama/abone olma sıradüzenlerindeki ve kümelerdeki kuyruk yöneticileri arasında iletilerin nasıl yönlendirildiğine ilişkin bir giriş için bkz. Dağıtılmış yayınlama/abone olma ağları.

Doğrudan yöneltilebilir bir yayınlama/abone olma kümesi aşağıdaki gibi davranır:

- Tüm kuyruk yöneticileri diğer tüm kuyruk yöneticilerini otomatik olarak bilir.
- Kümelenmiş konulara abonelikleri olan tüm kuyruk yöneticileri, kümedeki diğer tüm kuyruk yöneticilerine kanallar oluşturur ve onlara abonelikleri hakkında bilgi verir.
- Bir uygulama tarafından yayınlanan iletiler, bağlı olduğu kuyruk yöneticisinden, doğrudan eşleşen bir aboneliğin bulunduğu her kuyruk yöneticisine yöneltilir.

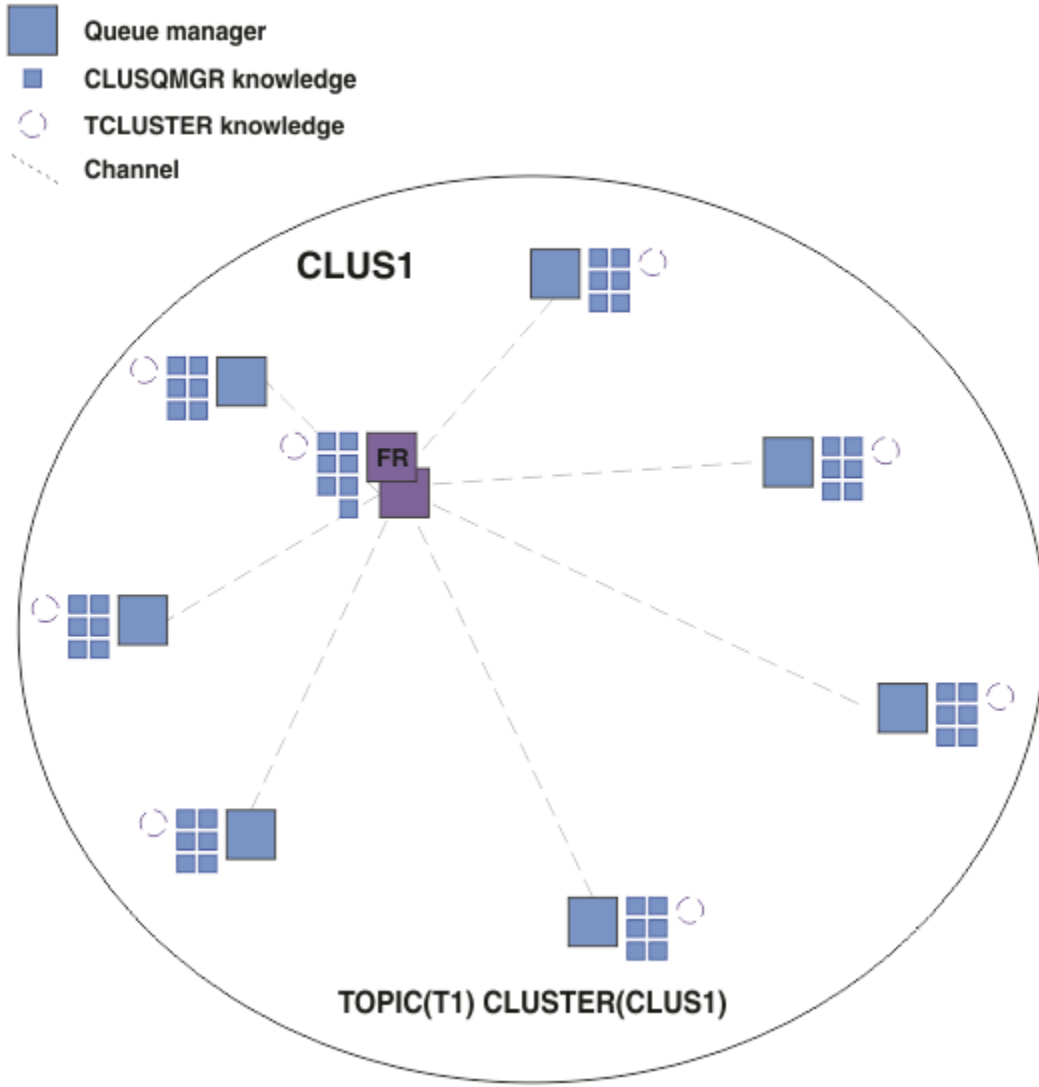
Aşağıdaki çizge, yayınlama/abone olma ya da noktadan noktaya iletişim etkinlikleri için kullanılmayan bir kuyruk yöneticisi kümesini göstermektedir. Kümedeki her kuyruk yöneticisinin yalnızca tam havuz kuyruğu yöneticilerine/yöneticilerinden bağlandığını unutmayın.



Şekil 16. Bir kuyruk yöneticisi kümesi

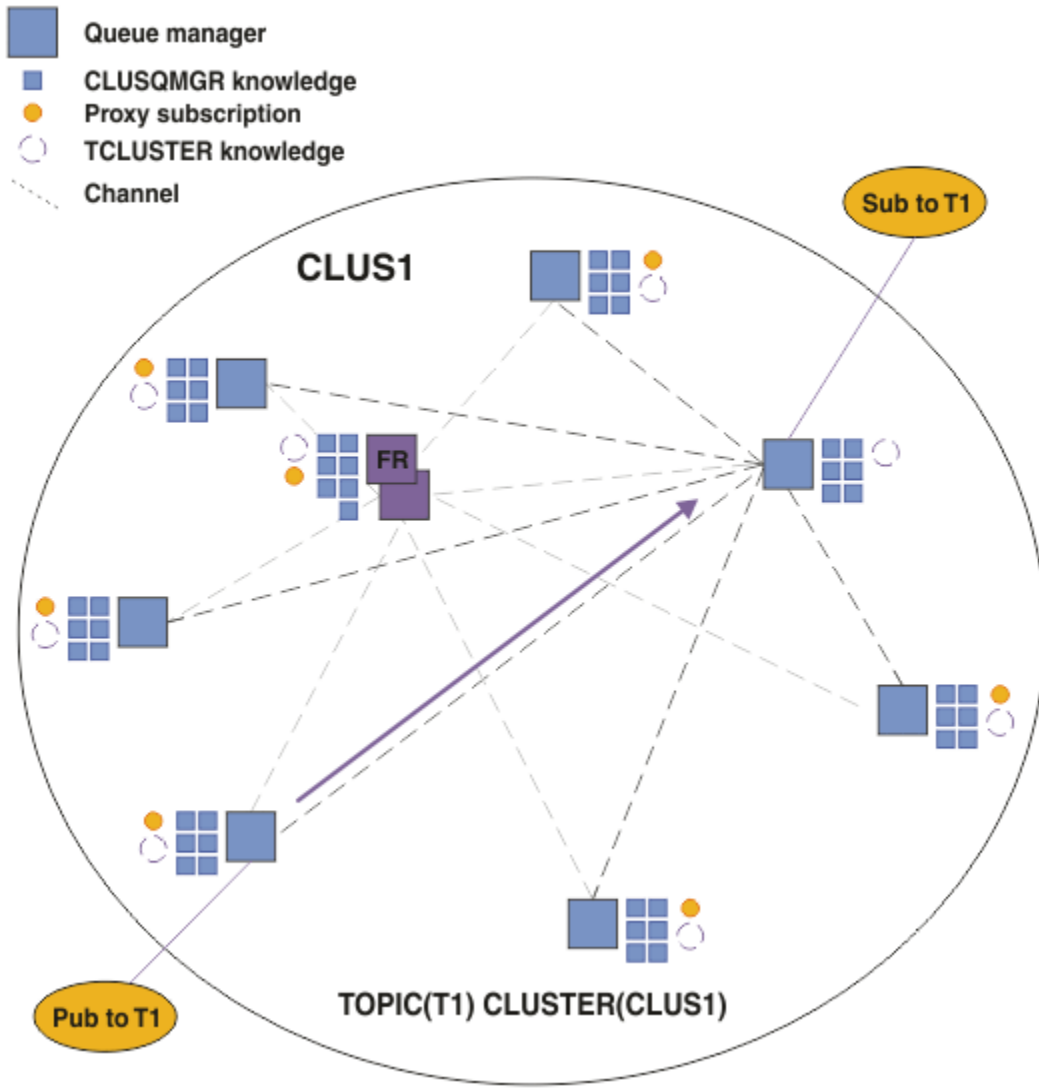
Doğrudan yöneltilen bir kümedeki kuyruk yöneticileri arasında yayınların akması için, konu ağacının bir dalını Bir yayınlama/abone olma kümesinin yapılandırılması konusunda açıkladığı gibi kümelersiniz ve *doğrudan yöneltilme* seçeneğini (varsayılan) belirtirsiniz.

Doğrudan yöneltilen bir yayınlama/abone olma kümesinde, konu nesnesini kümedeki herhangi bir kuyruk yöneticisinde tanımlarsınız. Bunu yaptığınızda, nesnenin bilgisi ve kümedeki diğer tüm kuyruk yöneticilerinin bilgisi, otomatik olarak tam havuz kuyruk yöneticileri tarafından kümedeki tüm kuyruk yöneticilerine gönderilir. Bu, herhangi bir kuyruk yöneticisi konuya başvurmadan önce gerçekleşir:



Şekil 17. Doğrudan yöneltilen yayınlama/abone olma kümesi

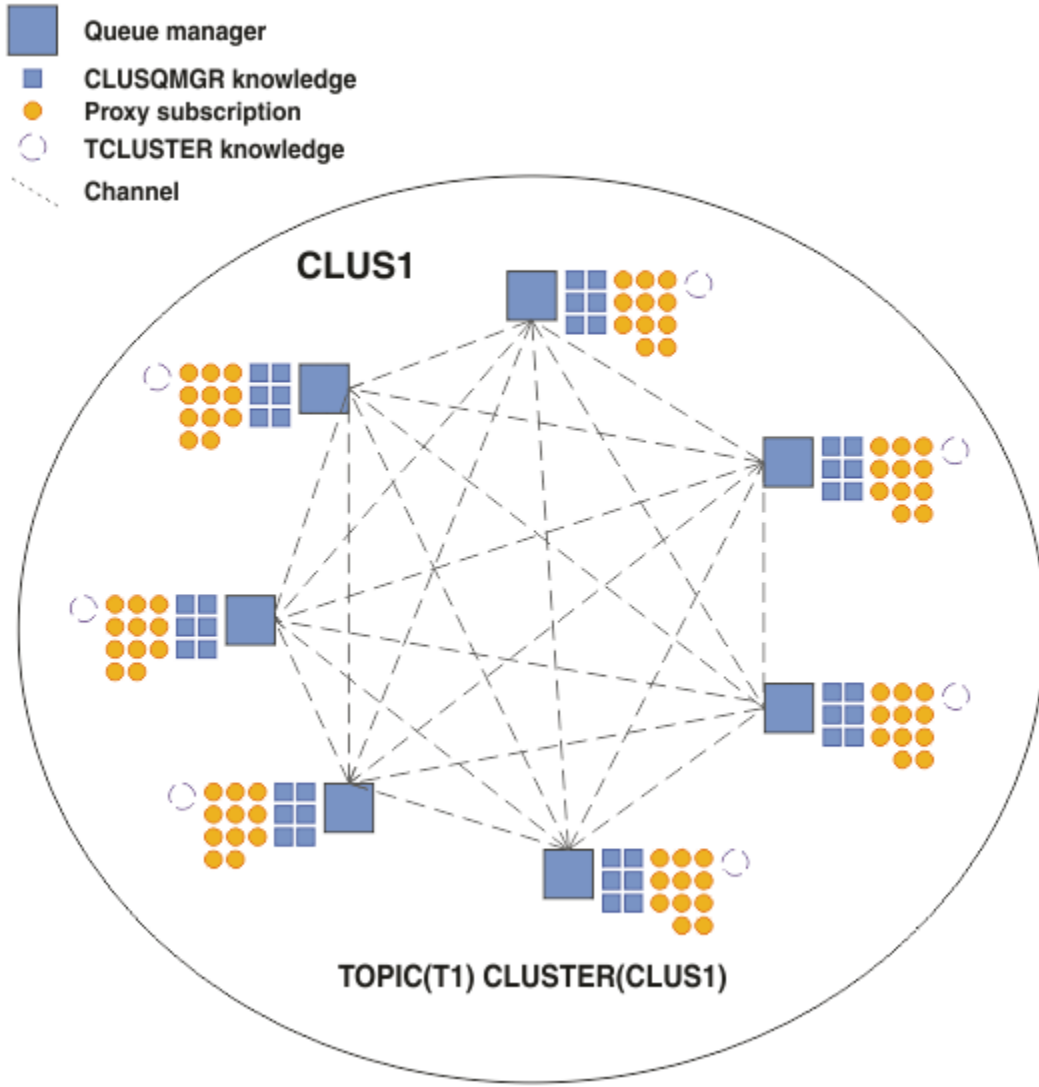
Abonelik yaratıldığında, aboneliği barındıran kuyruk yöneticisi, kümedeki her kuyruk yöneticisine bir kanal oluşturur ve aboneliğin ayrıntılarını gönderir. Bu dağıtılmış abonelik bilgisi, her kuyruk yöneticisinde bir yetkili sunucu aboneliği ile temsil edilir. Kümedeki yetkili aboneliğin konu dizgisiyle eşleşen herhangi bir kuyruk yöneticisinde bir yayın üretildiğinde, yayınlayıcı kuyruk yöneticisinden abonelik barındıran her kuyruk yöneticisine bir küme kanalı oluşturulur ve ileti her birine gönderilir.



Şekil 18. Bir yayıncı ve kümelenmiş bir konuya abone olan doğrudan yönlendirilmiş yayınlama/abone olma kümesi

Yayınlara abonelik barındırma kuyruk yöneticilerine doğrudan yönlendirilmesi, yapılandırmayı basitleştirir ve yayınlara aboneliklere teslim edilmesinde gecikme süresini en aza indirir.

Ancak, aboneliklerin ve yayıncıların konumuna bağlı olarak, kümeniz hızlı bir şekilde tam olarak birbirine bağlanabilir ve her kuyruk yöneticisinin diğer kuyruk yöneticileriyle doğrudan bağlantısı olabilir. Bu, ortamınızda kabul edilebilir ya da edilmeyebilir. Benzer şekilde, abone olunan konu dizgileri kümesi sık sık değişiyorsa, bu bilgilerin tüm kuyruk yöneticileri arasında yayılmasının ek yükü de önemli olabilir. Doğrudan yönlendirilen bir yayınlama/abone olma kümesindeki tüm kuyruk yöneticileri bu ek giderlerle başa çıkabilmelidir.



Şekil 19. Tam olarak birbirine bağlı, doğrudan yönltilen bir yayınlama/abone olma kümesi

## Özet ve diğer önemli noktalar

Doğrudan yönlendirilmiş bir yayınlama/abone olma kümesi, oluşturmak veya yönetmek için çok az manuel müdahaleye ihtiyaç duyar ve yayıncılar ile aboneler arasında doğrudan yönlendirme sağlar. Belirli yapılandırmalar için bu genellikle en uygun topoloji, özellikle birkaç kuyruk yöneticisine sahip kümeler ya da yüksek kuyruk yöneticisi bağlantılılığının kabul edilebilir olduğu ve aboneliklerin seyrek olarak değiştiği yerdir. Ancak, sisteminize bazı kısıtlamalar da getirir:

- Her kuyruk yöneticisindeki yük, kümedeki toplam kuyruk yöneticisi sayısı ile orantılıdır. Bu nedenle, daha büyük kümelerde, tek tek kuyruk yöneticileri ve bir bütün olarak sistem performans sorunlarıyla karşılaşılabilir.
- Varsayılan olarak, abone olunan tüm kümelenebilir konu dizgileri küme boyunca yayılır ve yayınlar yalnızca ilişkili konuya aboneliği olan uzak kuyruk yöneticilerine yayılır. Bu nedenle abonelik kümesinde yapılan hızlı değişiklikler sınırlayıcı bir faktör olabilir. Bu varsayılan davranışı değiştirebilir ve bunun yerine tüm yayınları tüm kuyruk yöneticilerine yayarak yetkili sunucu abonelikleri gereksinimini ortadan kaldırebilirsiniz. Bu, abonelik bilgi trafiğini azaltır, ancak büyük olasılıkla yayın trafiğini ve her kuyruk yöneticisinin oluşturduğu kanal sayısını artırır. Bkz. [Yayınlama/abone olma ağlarında abonelik performansı](#).

**Not:** Benzer bir kısıtlama sıradüzenler için de geçerlidir.

- Yayınlama/abone olma kuyruğu yöneticilerinin birbiriyle bağlantılı niteliği nedeniyle, yetkili aboneliklerin ağdaki tüm düğümlere yayılması zaman alır. Uzak yayınların hemen hemen abone olunması gerekmez, bu nedenle yeni bir konu dizgisine aboneliğin ardından yayınların başında yayınlanmayabilir. Tüm yayınların tüm kuyruk yöneticilerine yayılmasını sağlamak için, abonelik gecikmesinin neden olduğu sorunları kaldırabilirsiniz. Bu durumda, yetkili aboneliklere ilişkin gereksinme ortadan kalkar. Bkz. [Yayınlama/abone olma ağlarındaki abonelik performansı](#).

**Not:** Bu kısıtlama sıradüzenler için de geçerlidir.

Doğrudan yönlendirmeyi kullanmadan önce, [“Yayınlama/abone olma kümelerinde konu anasistem yönlendirmesi” sayfa 95](#)ve [“Yayınlama/abone olma sıradüzenlerinde yönlendirme” sayfa 119](#) içinde ayrıntılı olarak açıklanan alternatif yaklaşımları keşfedin.

### ***Yayınlama/abone olma kümelerinde konu anasistem yönlendirmesi***

Kümedeki barındırma dışı kuyruk yöneticilerinden gelen yayınlar, barındırma kuyruk yöneticisi aracılığıyla, eşleşen bir abonelikle kümedeki herhangi bir kuyruk yöneticisine yöneltilmektedir.

İletilerin yayınlama/abone olma sıradüzenleri ve kümelerinde kuyruk yöneticileri arasında nasıl yönlendirileceği bir giriş için bkz. [Dağıtılmış yayınlama/abone olma ağları](#).

Konu ana makine yönlendirmesinin davranışını ve avantajlarını anlamak için öncelikle [“Yayınlama/abone olma kümelerinde doğrudan yönlendirme” sayfa 90](#)' u anlamanız en iyi olur.

Bir konu anasistemi yayınlama/abone olma kümesi aşağıdaki gibi davranır:

- Kümelenmiş olarak yönetilen konu nesnelere, kümedeki tek tek kuyruk yöneticilerinde el ile tanımlanır. Bunlara *konu anasistem kuyruk yöneticisi* adı verilir.
- Bir küme kuyruk yöneticisinden bir abonelik yapıldığında, abonelik anasistem kuyruk yöneticisinden konu anasistem kuyruğu yöneticilerine kanallar yaratılır ve yetkili sunucu abonelikleri yalnızca konuyu barındıran kuyruk yöneticilerinde yaratılır.
- Bir uygulama bir konuya ilişkin bilgileri yayınlarken, bağlı kuyruk yöneticisi yayını her zaman, konuyu barındıran bir kuyruk yöneticisine iletir; bu da, bu yayını, kümedeki tüm kuyruk yöneticilerine, konu ile eşleşen aboneliklere sahip tüm kuyruk yöneticilerine iletir.

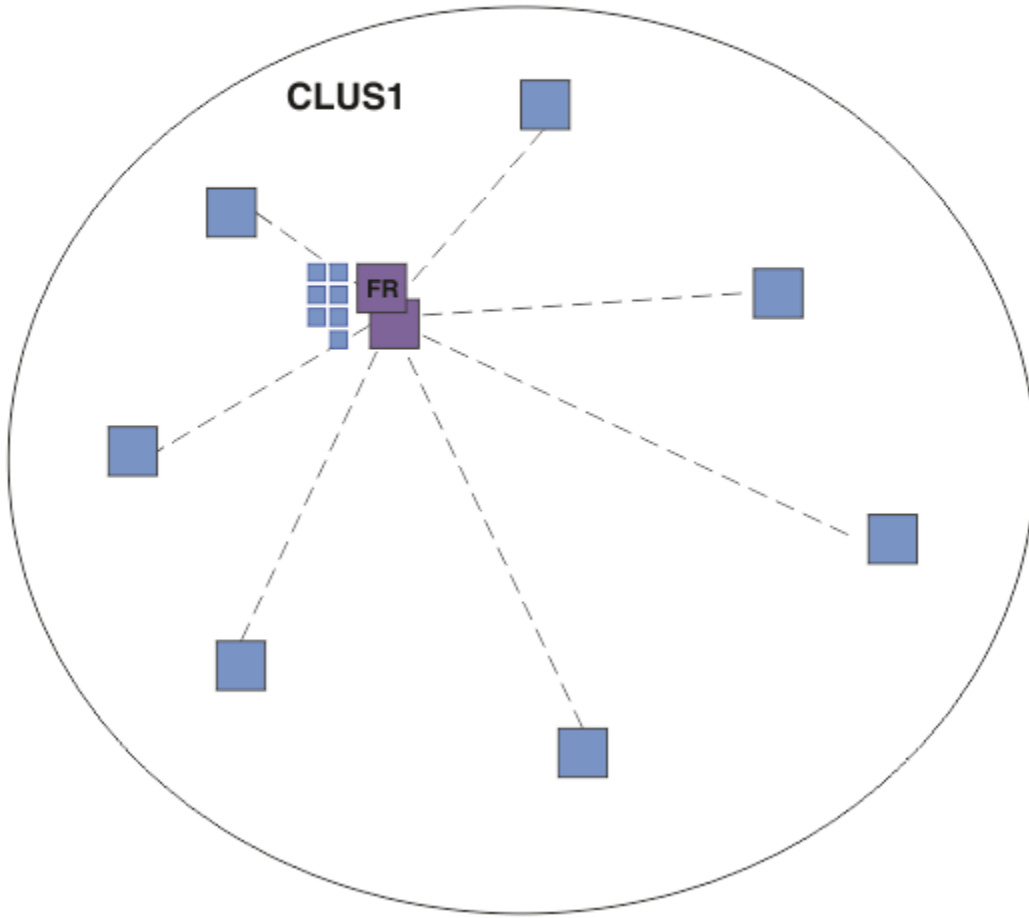
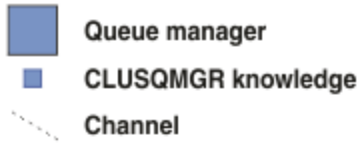
Bu süreç, aşağıdaki örneklerde daha ayrıntılı olarak açıklanmıştır.

### **Konu anasistem yöneltmesi tek bir konu anasistemi kullanılarak yöneltmesi**

Yayınlar için, bir konu ana makinesindeki kuyruk yöneticileri arasında akıllık yayınlar için, konu ağının bir dalını [Yayınlama/abone olma kümesi yapılandırılması](#)' ta açıklandığı gibi kümelediniz ve *konu anasistem yöneltmesi* ögesini belirtin.

Bir kümedeki birden çok kuyruk yöneticisinde bir konu nesnesini yöneltilecek bir konu nesnesini tanımlamak için bir dizi neden vardır. Ancak, basitlik için tek bir konu anasistemiyle başlarız.

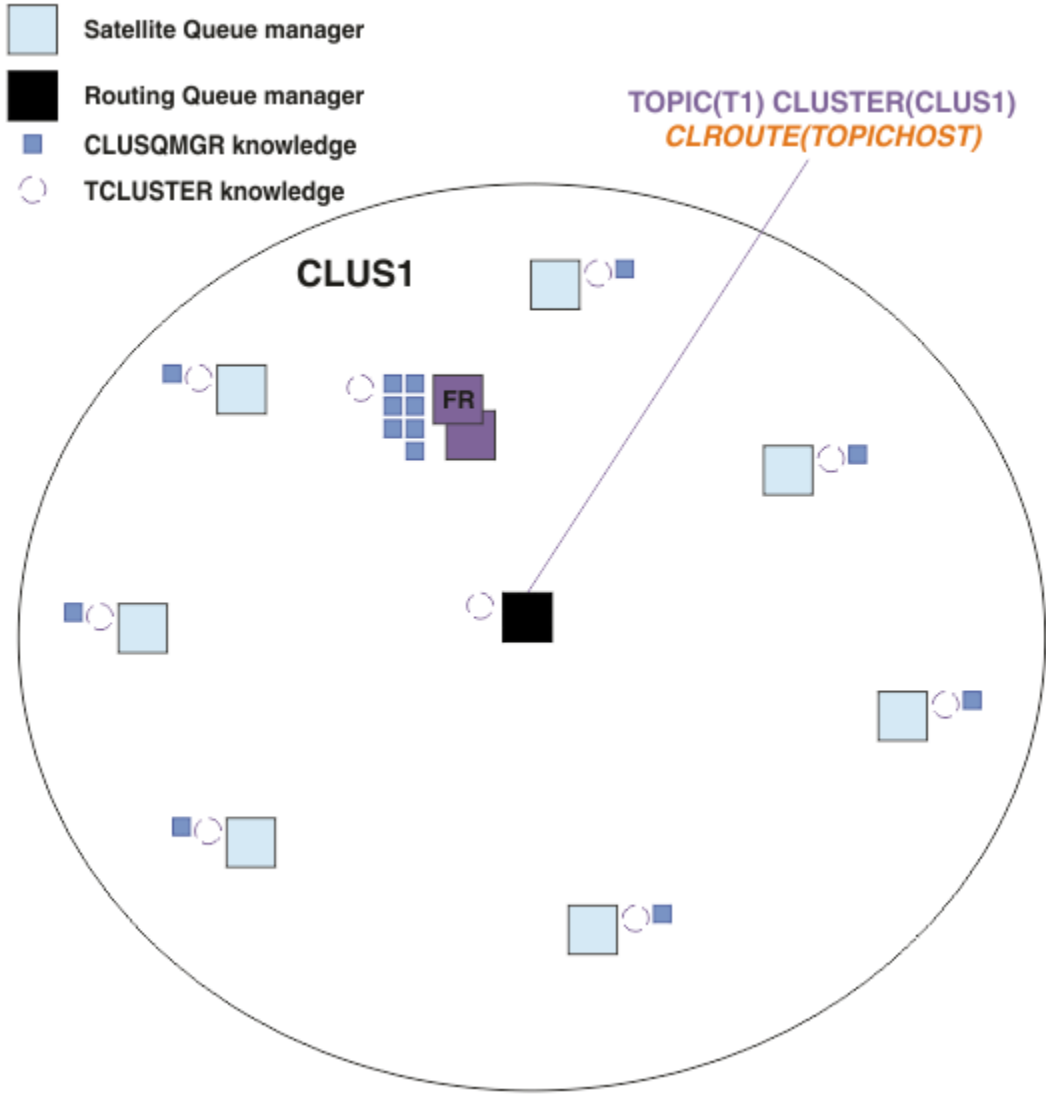
Aşağıdaki çizge, yayınlama/abone olma ya da noktadan noktaya iletişim etkinlikleri için kullanılmayan bir kuyruk yöneticisi kümesini göstermektedir. Kümedeki her kuyruk yöneticisinin yalnızca tam havuz kuyruğu yöneticilerine/yöneticilerinden bağlandığını unutmayın.



Şekil 20. Bir kuyruk yöneticisi kümesi

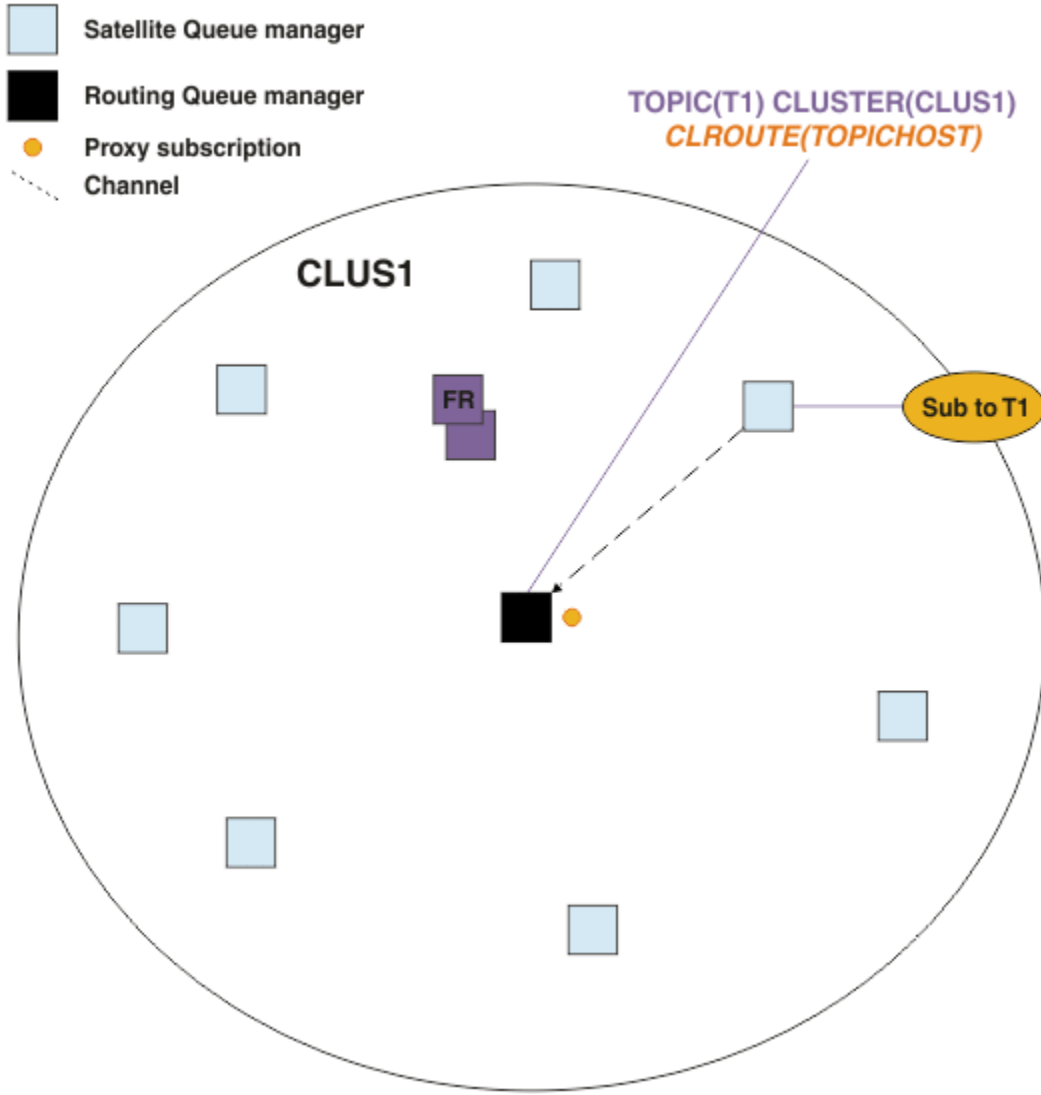
Bir konu ana makinesinde yayınlama/abone olma kümesi içinde, konu nesnesini kümedeki belirli bir kuyruk yöneticisinde tanımlayabilirsiniz. Yayınlama/abone olma trafiği o kuyruk yöneticisiyle akar, bunu kümede kritik bir kuyruk yöneticisi haline getirmekte ve iş yükünü artırmış olur. Bu nedenlerden dolayı, tam havuz kuyruk yöneticisi kullanılması önerilmez; ancak, kümede başka bir kuyruk yöneticisi kullanılması önerilmez. Anasistem kuyruk yöneticisinde konu nesnesini tanımladığınızda, nesnenin ve anasisteminin bilgisi otomatik olarak, tam havuz kuyruğu yöneticileri tarafından kümedeki diğer tüm kuyruk yöneticilerine yönlendirilir. *Doğrudan yöneltme*'nin tersine, her kuyruk yöneticisinin kümedeki diğer tüm kuyruk yöneticilikleriyle ilgili olarak anlatılmadığına dikkat edin.





Şekil 21. Tek bir konu anasisteminde tanımlı bir konu içeren yayınlama/abone olma kümesini yönlendiren bir konu

Bir kuyruk yöneticisine abonelik yaratıldığında, abone olunan kuyruk yöneticisi ile konu anasistem kuyruk yöneticisi arasında bir kanal yaratılır. Abone olunan kuyruk yöneticisi yalnızca konu anasistem kuyruk yöneticisine bağlanır ve aboneliğin ayrıntılarını ( *yetkili sunucu aboneliği* biçiminde gönderir). Konu anasistem kuyruk yöneticisi, bu abonelik bilgilerini kümedeki diğer kuyruk yöneticilerine iletmez.

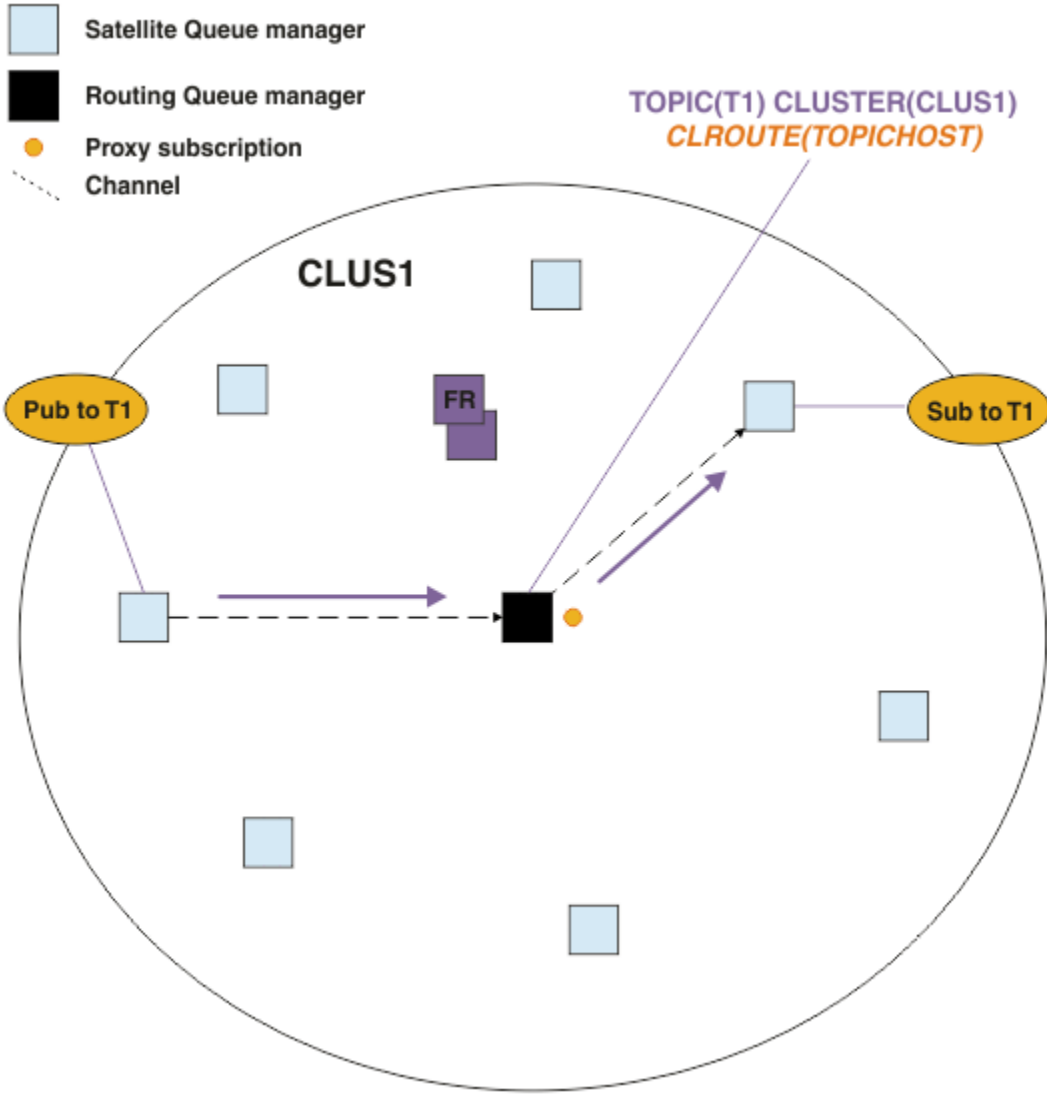


Şekil 22. Bir konu anasistemi ve bir konu anasisteminde tanımlı bir konu içeren yayınlama/abone olma kümesini bir abone olarak yönlendirdi

Bir yayınlama uygulaması başka bir kuyruk yöneticisine bağlandığında ve bir ileti yayınlandığında, yayınlama kuyruk yöneticisi ile konu anasistem kuyruk yöneticisi arasında bir kanal yaratılır ve ileti o kuyruk yöneticisine iletilir. Yayınlama kuyruk yöneticisinin kümedeki diğer kuyruk yöneticilerine ilişkin herhangi bir bilgisi yok. Bu nedenle, kümede o konuya abone olmasalar da ileti, konu anasistem kuyruk yöneticisine iletilir. Yayınlama kuyruğu yöneticisi yalnızca konu anasistem kuyruk yöneticisine bağlanır. Yayınlar, varsa, konu anasisteminden abone olan kuyruk yöneticilerine yöneltilir.

Yayınlayıcıyla aynı kuyruk yöneticisine ilişkin abonelikler, önce iletileri bir konu anasistem kuyruk yöneticisine göndermeden doğrudan karşılanır.

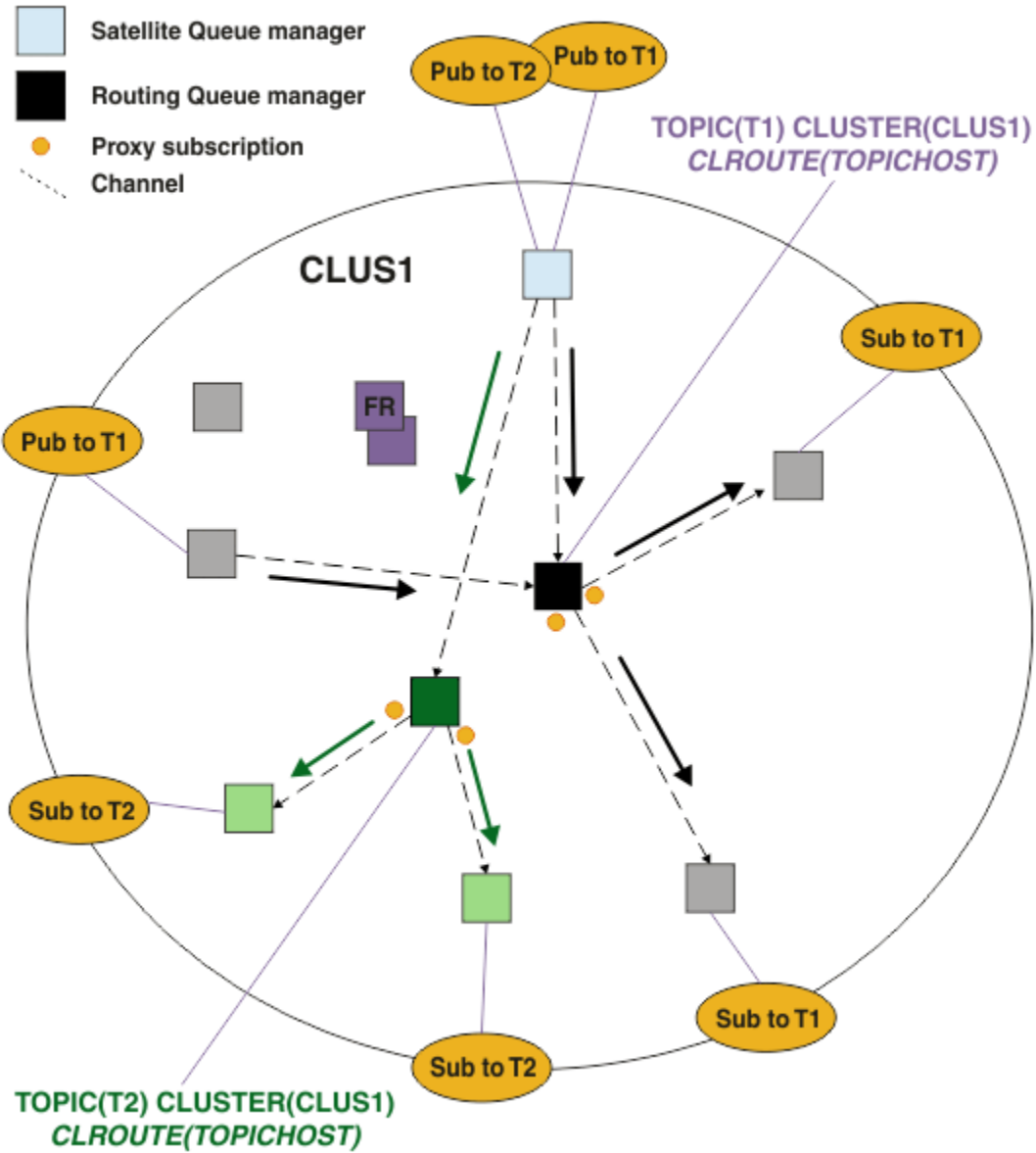
Her konu anasistem kuyruk yöneticisinin oynadığı kritik rol nedeniyle, konu barındırma konularının yüklemeye, kullanılabilirlik ve bağlantı gereksinimlerini işleyebilecek kuyruk yöneticilerini seçmeniz gerekir.



Şekil 23. Tek bir konu, bir abone ve bir yayıncı ile bir konu ana sistemi yayınlama/abone olma kümesini yönlendirdi

### Konu ağacının birden çok kuyruk yöneticisi arasında bölünmesi

Yönlendirilmiş bir konu kuyruk yöneticisi, yalnızca yönetilen konu nesnesinin yapılandırıldığı konu ağacının dalına ilişkin olan abonelik bilgisi ve yayın iletilerinden sorumludur. Farklı konular kümede farklı yayınlama/abone olma uygulamaları tarafından kullanılıyorsa, farklı kuyruk yöneticilerini konu ağacının farklı kümelenmiş dallarına ev sahipliği yapmak için yapılandırabilirsiniz. Bu, kümede yayın trafiğini, abonelik bilgilerini ve kanallarını her bir konu ana makine kuyruk yöneticisinde azaltarak ölçekleme sağlar. Bu yöntemi, konu ağacının farklı yüksek birim dalları için kullanmanız gerekir:



Şekil 24. Her biri bir konu anasisteminde tanımlı olan iki konu içeren yayınlama/abone olma kümesini yönlendiren bir konu

For example, using the topics described in [Konu ağaçları](#), if topic T1 was configured with a topic string of /USA/Alabama, and topic T2 was configured with a topic string of /USA/Alaska, then a message published to /USA/Alabama/Mobile would be routed through the queue manager hosting T1, and a message published to /USA/Alaska/Juneau would be routed through the queue manager hosting T2.

**Not:** Konu ağacında, kümelenmiş noktalardan daha yüksek bir genel arama karakteri kullanarak, tek bir abonelik, konu ağacının birden çok kümelenmiş dalını kapsayamaz. Bkz. [Joker abonelikler](#).

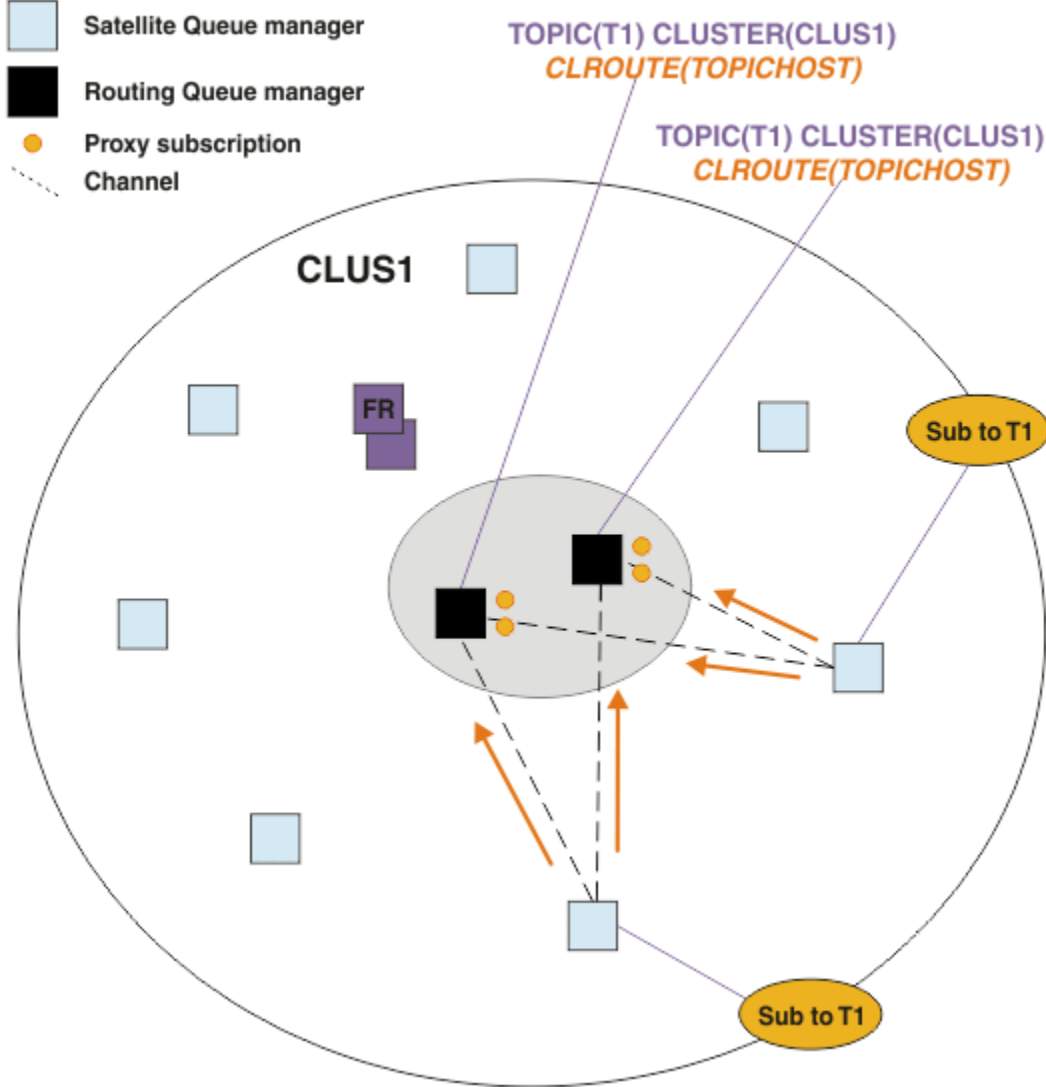
### Tek bir konu için birden çok konu anasistemini kullanarak anasistem yönlendirmesi

Tek bir kuyruk yöneticisinin bir konunun yönlendirilmesi için sorumluluğu varsa ve o kuyruk yöneticisi kullanılamaz duruma gelirse ya da iş yükünü işleyemez duruma gelirse, yayınlar abonelikleri hemen yerine getirmez.

Tek bir kuyruk yöneticisinden bir konu tanımladığınızda elde edebileceğimizden daha fazla esneklik, ölçeklenebilirlik ve iş yükü dengelemesi gerekirse, birden çok kuyruk yöneticisinden bir konu tanımlayabilirsiniz. Yayınlanan her ileti, tek bir konu anasistemiyle yönlendirilir. Birden çok eşleşen konu anasistem tanımlaması varsa, konu anasistemlerinden biri seçilir. Seçim, kümelenmiş kuyruklar

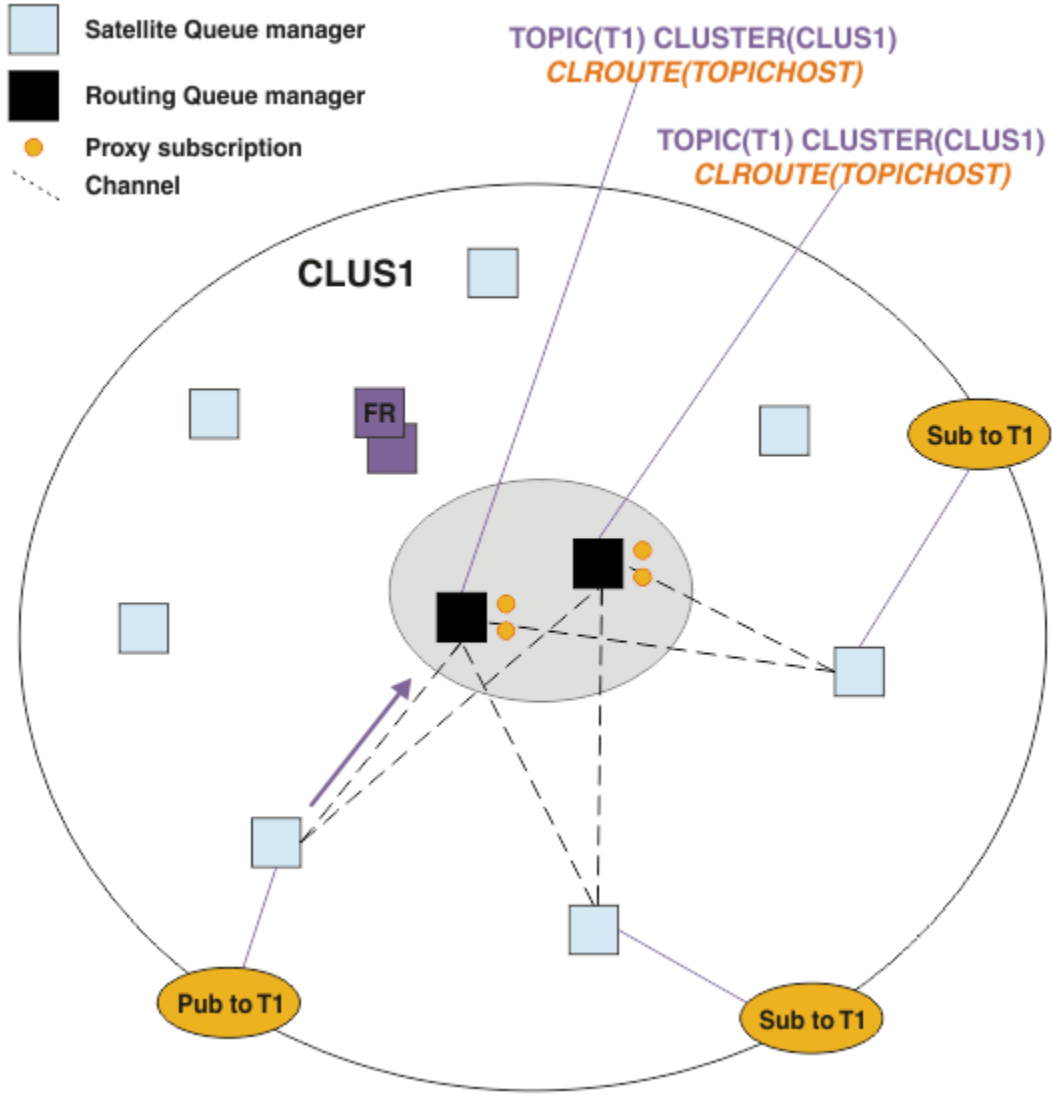
için aynı şekilde yapılır. Bu, iletilerin kullanılabilir konu anasistemlerine yönlendirilmesini, bu iletilerin kullanılmamasını önleyerek, ileti yüklemesini birden çok konu anasistem kuyruğu yöneticisi ve kanallarında dengeli bir şekilde dengelemeye olanak sağlar. Ancak, kümede aynı konu için birden çok konu anasistemi kullandığınızda birden çok ileti için sıralama yapılmamaktadır.

Aşağıdaki şekilde, iki kuyruk yöneticisi üzerinde aynı konunun tanımlandığı bir konu anasistemi yönlendirilmiş küme gösterilir. Bu örnekte, abone olunan kuyruk yöneticileri, abone olunan konuyla ilgili bilgileri her iki konu ana makine yöneticisine yetkili bir abonelik biçiminde gönderir:



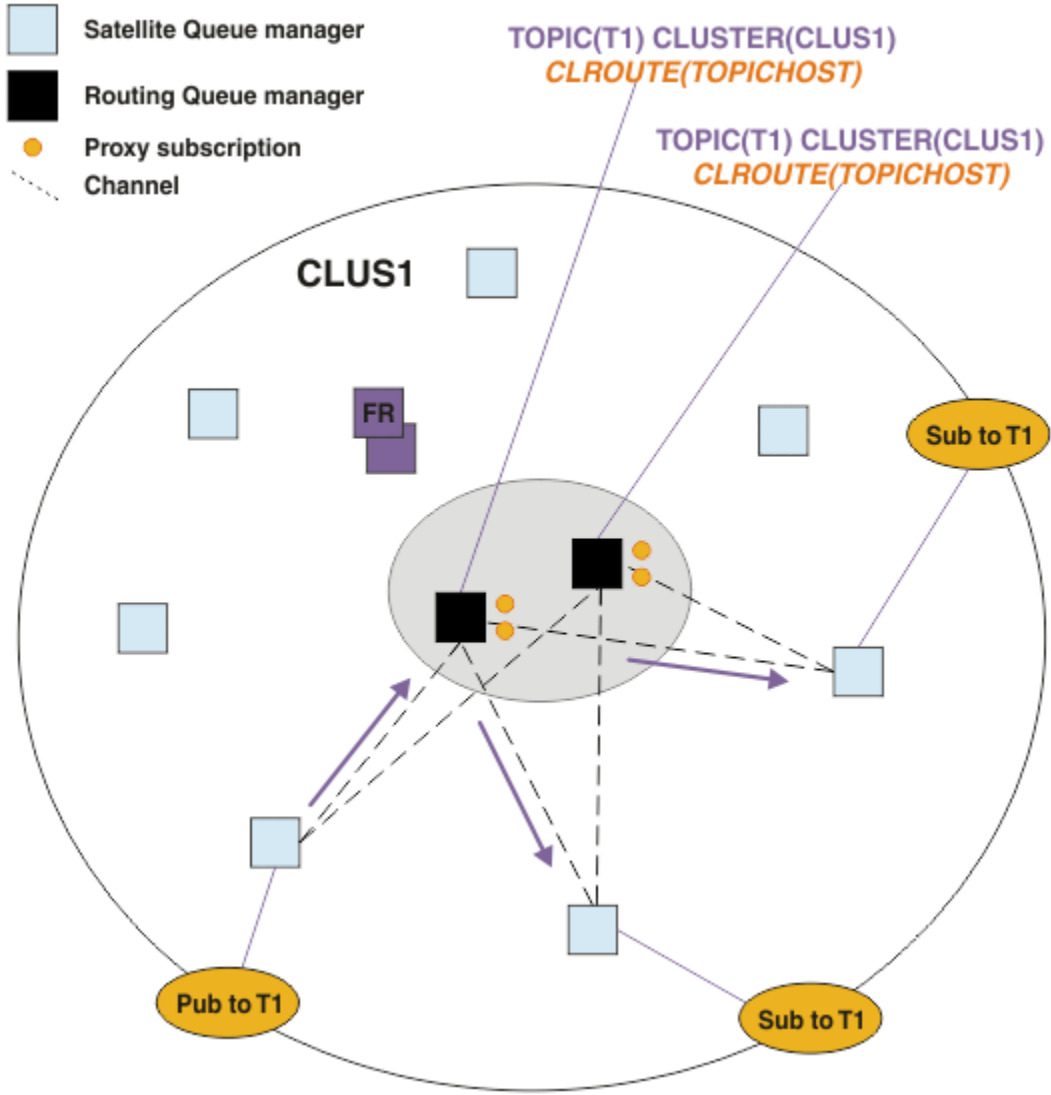
Şekil 25. Birden çok konu ana makine yayınlaması/abone olma kümesinde yetkili sunucu abonelikleri oluşturma

Barındırma olmayan bir kuyruk yöneticisinden bir yayın yapıldığında, kuyruk yöneticisi, bu konuya ilişkin anasistem kuyruk yöneticilerininintopic'e yayınının bir kopyasını *bir* ' a gönderir. The system chooses the host based on the default behavior of küme iş yükü yönetimi algoritması. Tipik bir sistemde, bu durum her bir konu ana makine kuyruk yöneticisinde dönüşümlü olarak dönüşümlü bir dağılıma yol alır. Aynı yayınlama uygulamasından gelen iletiler arasında bir benzeşim yok; bu, NOTFIXEDküme bağ tanımlama tipini kullanmaya eşitler.



Şekil 26. Birden çok konu ana makine yayınlaması/abone olma kümesinde yayınları alma

Daha sonra, seçilen konuya ilişkin anasistem kuyruk yöneticisine yönelik gelen yayınlar, eşleşen bir yetkili sunucu aboneliği kaydettirmiş olan tüm kuyruk yöneticilerine iletilir:



Şekil 27. Yayınları birden çok konu ana makine yayınlaması/abone olma kümesinde abonelere yönlendirme

### Abonelikleri ve yayıncıları bir konu anasistem kuyruk yöneticisiyle yerel olarak yapma

Yukarıdaki örnekler, yönetimler ve aboneler arasında yönetilmiş yönlendirilmiş konu nesnelere olmayan kuyruk yöneticilerindeki yönlendirmeyi gösterir. Bu topolojilerde, iletiler aboneliklere ulaşmak için birden çok sekme gerektirir.

Ek sekmenin istenmediği durumlarda, önemli yayıncıların yer alan kuyruk yöneticilerine bağlanması uygun olabilir. Ancak, bir konu için birden çok konu anasistemi varsa ve yalnızca bir yayıncı varsa, tüm yayın trafiği, yayıncının bağlı olduğu konu anasistem kuyruk yöneticisi aracılığıyla yönlendirilir.

Benzer şekilde, anahtar abonelikleri varsa, bunlar bir konu anasistem kuyruk yöneticisinde yer alır. Ancak, yönlendirilen konunun birden çok anasistemi varsa, yalnızca yayınların bir kısmı ek sekme önlerinden kaçınacaktır; geri kalan kısmı ise önce diğer konu anasistem kuyruk yöneticilerine yönlendirilecektir.

Bu tür topolojiler burada daha ayrıntılı olarak açıklanmaktadır: [Merkezi yayıncılar ya da aboneler kullanılarak konu ana makine yönlendirmesi](#).

**Not:** Yayıncıları ya da abonelikleri yönlendirilmiş konu anasistemleriyle birlikte yerleştirirken yönlendirilen konu yapısının değiştirilmesi için özel planlama gereklidir. Örneğin, [Ek konu anasistemlerinin bir konuya yönlendirilmiş küme için eklenmesi](#) başlıklı konuya bakın.

## Özet ve ek konular

Bir konu ana makinesi yayınlama/abone olma kümesi, her konuda hangi kuyruk yöneticilerinin anasistem üzerinde yer aldığı ve o kuyruk yöneticilerinin konu ağacının o dalı için *yöneltilme* kuyruk yöneticisi olmaları konusunda size kesin denetim sağlar. Ayrıca, abonelikler ya da yayıncılarda bulunan kuyruk yöneticilerinin konu anasistem kuyruk yöneticileriyle bağlanmaya gerek yoktur ve abonelikleri olan kuyruk yöneticilerinin, konu sahibi olmayan kuyruk yöneticilerine bağlanmaya gerek yoktur. Bu yapılandırma, kümedeki kuyruk yöneticileri arasındaki bağlantı sayısını ve kuyruk yöneticileri arasında aktarılmakta olan bilgi miktarını önemli ölçüde azaltabilir. Bu durum özellikle, kuyruk yöneticilerinin yalnızca bir alt kümesinin yayınlama/abone olma çalışmalarını gerçekleştirdikleri büyük kümelerde özellikle geçerlidir. Bu yapılandırma aynı zamanda, kümedeki tek tek kuyruk yöneticilerine yüklemeye üzerinde bazı denetim sağlar, bu nedenle (örneğin, yüksek düzeyde etkin konuları daha güçlü ve daha dayanıklı sistemler üzerinde barınmayı seçebilirsiniz. Bazı yapılandırmalarda-özellikle daha büyük kümeler-genellikle *doğrudan yönlendirme*<sup>1</sup> dan daha uygun bir topolojidir.

Ancak, konu anasistemi yönlendirmesi sisteminize bazı kısıtlamalar da getirir:

- Sistem yapılandırması ve bakımı, doğrudan yönlendirmeye göre daha fazla planlama gerektirir. Konu ağacında hangi kümeyi göstereceğine ve kümedeki konu tanımlamalarının konumuna karar vermeniz gerekir.
- Yöneltilen doğrudan konular için olduğu gibi, yeni bir konu anasistem tarafından yönlendirilen konu tanımlandığında, bilgiler tam havuz kuyruğu yöneticilerine ve oradan doğrudan kümenin tüm üyelerine gönderilir. Bu olay, başlatılmadıysa, kümenin her üyesi için tam havuzlardan kanalların başlatılmasına neden olur.
- Yayınları, kümede abonelik olmasa da, anasistem kuyruk yöneticisinden her zaman bir anasistem kuyruk yöneticisine gönderilir. Bu nedenle, aboneliklerin genellikle var olması beklenirken ya da genel bağlantı ve bilgi yükü ek yayın trafiği riskinden daha fazla yüklenen yönlendirilen konuları kullanmanız gerekir.

**Not:** Daha önce anlatıldığı gibi, yayıncıların bir konu sahibi tarafından yerel olarak yapılması bu riski en aza indirebilir.

- Anasistem dışı kuyruk yöneticisinde yayınlanan iletiler, aboneliği barındıran kuyruk yöneticisine doğrudan gitmez, her zaman bir konu anasistem kuyruk yöneticisi aracılığıyla yönlendirilir. Bu yaklaşım, kümenin toplam ek yükünü artırabilir ve ileti gecikme süresini artırabilir ve performansı azaltabilir.

**Not:** Daha önce açıklandığı gibi, bir konu anasisteminin yerel olarak abonelikleri ya da yayıncıları bu riski en aza indirebilir.

- Tek bir konu anasistem kuyruk yöneticisinin kullanılması, bir konuda yayınlanan tüm iletiler için tek bir hata noktası sağlar. Birden çok konu anasistemi tanımlayarak bu tek hata noktasını kaldırabilirsiniz. Ancak, birden çok anasisteme sahip olmak, abonelikler tarafından alınan yayınlanan iletilerin sırasını etkiler.
- Birden çok kuyruk yöneticisinden gelen yayınlama trafiğinin bu yöneticiler tarafından işlenmesi gerektiğinden, konu anasistem kuyruk yöneticileri fazladan ileti yüklemesini gerçekleştirmektedir. Bu yük azaltılabilir: Tek bir konu için birden çok konu anasistemi kullanın (bu durumda ileti sıralaması korunmaz) ya da konu ağacının farklı dalları için yönlendirilen konuları barındırmak üzere farklı kuyruk yöneticileri kullanın.

Konu ana makine yönlendirmesini kullanmadan önce, "[Yayınlama/abone olma kümelerinde doğrudan yönlendirme](#)" sayfa 90 ve "[Yayınlama/abone olma sıradüzenlerinde yönlendirme](#)" sayfa 119' da ayrıntılı olarak açıklanan diğer yaklaşımları keşfedin.

### ***Kümelemeyi yayınlama/abone olma: En iyi uygulamalar***

Kümelenebilir konuların kullanılması, yayınlama/abone olma etki alanının kuyruk yöneticileri arasında basit olmasına neden olur, ancak mekanik ve çıkarımlar tam olarak anlaşılmalıysa sorunlara yol açabilir. Bilgi paylaşımı ve yayın yönlendirmesi için iki model vardır. Tek tek iş gereksinimlerinizi en iyi karşılayan modeli uygulayın ve seçilen kümeden en iyi şekilde performans gösterin.

Aşağıdaki bölümlerdeki en iyi uygulama bilgileri, tek bir boyutta tüm çözüme uygun bir boyut sağlamaz, daha çok ortak sorunları çözmek için ortak yaklaşımları paylaşır. It assumes that you have a basic



understanding of IBM MQ clusters, and of publish/subscribe messaging, and that you are familiar with the information in [Dağıtılmış yayınlama/abone olma ağları](#) and [“Yayınlama/abone olma kümelerini tasarlama” sayfa 88](#).

Noktadan noktaya ileti alışverişi için bir küme kullandığınızda, kümedeki her kuyruk yöneticisi, bilmesi gereken bir temel üzerinde çalışır. Yani, bu yalnızca kümedeki diğer kuyruk yöneticileri ve kümelenmiş kuyruklardaki diğer kuyruk yöneticileri gibi, bunları kullanmak için istekte bulunan uygulamalar gibi diğer küme kaynaklarını da bulur. Bir kümeye yayınlama/abone olma ileti sistemi eklediğinizde, küme kuyruğu yöneticileri arasında bilgi ve bağlantırlık düzeyinin yükseltilmesi artar. Yayınlama/abone olma kümelerinde en iyi uygulamaları izleyebilmek için, bu değişikliğin davranışlarda etkilerini tam olarak anlamamız gerekir.

Kesin gereksinimlerinize göre en iyi mimariyi oluşturmanıza olanak sağlamak için, yayınlama/abone olma kümelerinde bilgi paylaşımı ve yayın yönlendirmesi için iki model vardır: *doğrudan yönlendirme* ve *konu ana makine yönlendirmesi*. Doğru seçimi yapmak için, her iki modeli de ve her modelin karşıması gereken farklı gereksinimleri de anlamamız gerekir. Bu gereksinimler, [“Dağıtılmış yayınlama/abone olma ağının planlanması” sayfa 84](#) ile bağlantılı olarak aşağıdaki bölümlerde ele alınmıştır:

- [“Yayınlama/abone olma etkinlemede yer alan küme kuyruğu yöneticilerinin sayısını sınırlandırmak için nedenler” sayfa 105](#)
- [“Hangi konuların kümeleceğine nasıl karar vereceğine” sayfa 106](#)
- [“Sisteminizi nasıl boyutlatacağını” sayfa 106](#)
- [“Yayınlama ve abonelik yeri” sayfa 107](#)
- [“Yayın trafiği” sayfa 107](#)
- [“Abonelik değişikliği ve dinamik konu dizgileri” sayfa 108](#)

## **Yayınlama/abone olma etkinlemede yer alan küme kuyruğu yöneticilerinin sayısını sınırlandırmak için nedenler**

Bir kümede yayınlama/abone olma ileti alışverişi kullandığınızda, kapasite ve başarımları bilgileri vardır. Bu nedenle, kuyruk yöneticilerindeki yayınlama/abone olma etkinliği gereksinimini dikkatli bir şekilde göz önünde bulundurmanız ve bu etkinliği, yalnızca onu gerektiren kuyruk yöneticisi sayısı ile sınırlandırmak en iyi uygulamadır. Konuları yayınlama ve abone olma gereksinimi olan kuyruk yöneticisi sayısı alt sınırı belirlendikten sonra, bunlar yalnızca bunları içeren bir kümenin üyesi ve başka kuyruk yöneticisi bulunmuyorlar.

Bu yaklaşım, özellikle noktadan noktaya ileti sistemi için düzgün çalışan bir küme oluşturursanız varsa yararlı olur. Var olan büyük bir kümeyi bir yayınlama/abone olma kümesine çevirirken, yürürlükteki kümeyi kullanmak yerine, uygulamaların denenebileceği yayınlama/abone olma çalışmaları için başlangıçta ayrı bir küme yaratmak daha iyi bir uygulamadır. Önceden bir ya da daha fazla noktadan noktaya küme oluşturan var olan kuyruk yöneticilerinin bir alt kümesini kullanabilir ve bu alt küme üyelerini yeni yayınlama/abone olma kümesinin üyeleri haline getirebilirsiniz. Ancak, yeni kümeniz için tam havuz kuyruğu yöneticileri başka herhangi bir kümenin üyesi olmamalıdır; bu, var olan küme tam havuzlarından ek yükü yalıtmalıdır.

Yeni bir küme oluşturamıyorsanız ve var olan büyük bir kümeyi bir yayınlama/abone olma kümesine çevirmek zorunda kalmıyorsanız, doğrudan yönlendirilmiş bir model kullanmayın. Konu anasistem yöneticisi modeli genellikle daha büyük kümelerde daha iyi performans gösterir, çünkü genellikle yayınlama/abone olma bilgileri paylaşımını ve yayınlama/abone olma çalışmalarını etkin bir şekilde gerçekleştiren kuyruk yöneticilerine bağlantırlığı kısıtladığından, konulara ev sahipliği yapan kuyruk yöneticilerine yoğunlaşılır. Bu kural dışı durum, abonelik bilgilerinin el ile yenilenmesi, bir konu tanımlamasının bulunduğu bir kuyruk yöneticisinde çağrılırsa, bu noktada konu anasistem kuyruk yöneticisi kümedeki her kuyruk yöneticisine bağlanacaktır. Bkz. [Yetkili sunucu aboneliklerinin yeniden eşzamanlanması](#).

Bir kümenin boyutu ya da yürürlükteki yükü nedeniyle yayınlamak/abone olmak için bir küme kullanılmadığını oluşturursanız, bu kümenin beklenmeyen bir şekilde bir yayınlama/abone olma kümesi haline getirilmesini önlemek iyi bir uygulamadır. Kümedeki herhangi bir kuyruk yöneticisinde kümelenmiş bir konu eklemeyi durdurmak için **PSCLUS** kuyruk yöneticisi özelliğini kullanın. Bkz. [“Kümelenmiş yayınlama/abone olma engelini engelle” sayfa 115](#).

## Hangi konuların kümeleceğine nasıl karar vereceğine

Kümeye hangi konuların eklendiğini dikkatli bir şekilde seçmek önemlidir: Konu ağacında bu konuların daha yüksek olması, kullanımlarının daha yaygın hale gelmesinden daha yaygın hale gelir. Bu, daha fazla abonelik bilgisi ve yayınlara ilgili yayınların gerekenden kaynaklanabileceğini sağlar. Bazı gereksinimlerin kümelenmesi ve bazılarının yapılmaması gereken, konu ağacının birden çok, ayrı dalları varsa, kümelenmesi gereken her dalın kökünde denetlenen konu nesnelere yaratın ve bunları kümeye ekleyin. Örneğin, /A, /B ve /C dallarında kümeleme gerekiyorsa, her bir dal için ayrı kümelenmiş bir konu nesnelere tanımlayın.

**Not:** Sistem, konu ağacındaki kümelenmiş konu tanımlarını içiçe yerleştirmenizi engelliyor. Her bir alt dal için konu ağacındaki bir noktada yalnızca küme başlıklarına izin verilir. Örneğin, /A için ve /A/B için kümelenmiş konu nesnelere tanımlayamazsınız. Kümelenmiş konuları içiçe yerleştirme, özellikle abonelikler joker karakterler kullandığında, kümelenmiş nesnenin hangi abonelik için geçerli olduğu karışıklığa neden olabilir. Bu durum, konu anasistemi yöneltmesi kullanılırken, yöneltme kararlarının konu anasistemlerinin tahsisi tarafından tam olarak tanımlandığı durumlarda daha da önemlidir.

Kümelenmiş konuların konu ağacına yüksek bir şekilde eklenmesi gerekiyorsa, ancak kümelenmiş noktenin altındaki ağacın bazı dalları kümelenmiş davranışı gerektirmiyorsa, daha fazla konu için abonelik düzeyini ve yayın paylaşımını azaltmak için abonelik ve yayın kapsamı özniteliklerini kullanabilirsiniz.

Görülen davranışı dikkate almadan, konu kök düğümünü kümenin içine koymamalısınız. Make global topics obvious where possible, for example by using a high-level qualifier in the topic string: /global or /cluster.

Kök konu düğümünü kümelenmiş yapmak istememeniz için başka bir neden vardır. Bunun nedeni, her kuyruk yöneticisinin kök düğüm için bir yerel tanımlaması olması, SYSTEM.BASE.TOPIC konu nesnesi olması gerekir. Bu nesne, kümedeki bir kuyruk yöneticisinde kümelenirse, diğer tüm kuyruk yöneticilerine bunun farkında olarak gelir. Ancak, aynı nesnenin yerel tanımlaması varsa, bu nesnenin özellikleri küme nesnesini geçersiz kılar. Bu, bu kuyruk yöneticilerinin konu kümelenmemiş gibi hareket etmelerine neden olur. Bunu çözmek için, SYSTEM.BASE.TOPIC' un her tanımlamasını kümelemeniz gerekir. Bunu doğrudan yönlendirilen tanımlar için de yapabilirsiniz, ancak konu anasistem yönlendirmeleri için değil, her kuyruk yöneticisinin bir konu anasistemi olmasına neden olur.

## Sisteminizi nasıl boyutlatacağını

Yayınlama/abone olma kümeleri tipik olarak, küme kanallarının bir kümedeki noktadan noktaya ileti kanallarına farklı bir örüntüye neden olur. Noktadan noktaya iletişim modeli bir 'seçenek' dir, ancak yayınlama/abone olma kümelerinde özellikle doğrudan yönlendirilen konular kullanıldığında abonelik fan çıkışıyla daha fazla ayrılmaz bir doğa söz konusu olur. Bu nedenle, yayınlama/abone olma kümelerindeki hangi kuyruk yöneticilerinin diğer kuyruk yöneticilerine bağlanmak için küme kanallarını kullanacağını ve hangi koşullar altında kullanılacağını tanımlamak önemlidir.

Aşağıdaki çizelge, yayınlama/abone olma kümesindeki kuyruk yöneticisi rolüne bağımlı olarak, olağan çalışma altındaki bir yayınlama/abone olma kümesindeki her kuyruk yöneticisi için beklenen tipik küme göndereni ve alıcı kanallarını listeler.

<i>Çizelge 5. Her yöneltme yöntemi için küme gönderen ve alıcı kanalları.</i>				
<b>Kuyruk yöneticisi rolü</b>	<b>Doğrudan küme alıcıları</b>	<b>Doğrudan küme gönderenleri</b>	<b>Konu küme alıcıları</b>	<b>Konu küme göndericileri</b>
Tam havuz	AllQmgrs	AllQmgrs	AllQmgrs	AllQMgrs
Konu tanımlaması anasistemi	yok	yok	AllSubs+AllPubs (1)	AllSubs (1)
Abonelikler yaratıldı	AllPubs (1)	AllQMgrs	AllHosts	AllHosts
Bağlı yayıncılar	AllSubs (1)	AllSubs (1)	AllHosts	AllHosts
Yayıncı ya da abone yok	AllSubs (1)	Yok (1)	Yok (2)	Yok (2)

**Anahtar:****AllQmgrs**

Kümedeki her kuyruk yöneticisinden ve her kuyruk yöneticisinden bir kanal.

**AllSubs**

Bir aboneliğin oluşturulduğu her kuyruk yöneticisinden ve her kuyruk yöneticisinden bir kanal.

**AllPubs**

Bir yayınlama uygulamasının bağlı olduğu her kuyruk yöneticisinden ve her kuyruk yöneticisinden bir kanal.

**AllHosts**

Kümelenmiş konu nesnesinin tanımının yapılandırıldığı her kuyruk yöneticisinden ve her kuyruk yöneticisinden bir kanal.

**Yok**

Tek bir yayınlama/abone olma ileti alışverişi amacıyla kümedeki diğer kuyruk yöneticilerine ya da diğer kuyruk yöneticilerine kanal yok.

**Notlar:**

1. Bu kuyruk yöneticisinden yetkili sunucu abonelikleri için kuyruk yöneticisi yenilemesi yapılırsa, kümedeki diğer tüm kuyruk yöneticilerine bir kanal otomatik olarak yaratılabilir.
2. Bu kuyruk yöneticisinden yetkili sunucu abonelikleri için bir kuyruk yöneticisi yenilemesi yapılırsa, kümelenmiş bir konunun tanımını barındıran kümedeki diğer kuyruk yöneticilerine ve diğer kuyruk yöneticilerine bir kanal otomatik olarak yaratılabilir.

Önceki çizelge, konu anasistem yöneltmenin genellikle doğrudan yöneltmeden çok daha az küme gönderen ve alıcı kanallarını kullandığını gösterir. Kanal bağlantılığı, bir kümedeki belirli kuyruk yöneticileri için, kapasite ya da belirli kanalları (örneğin, güvenlik duvarları yoluyla) oluşturma nedenleriyle ilgileniyorsa, konu ana makine yönlendirmesi bu nedenle tercih edilen bir çözümdür.

**Yayınlayıcı ve abonelik yeri**

Kümelenmiş yayınlama/abone olma, bir kuyruk yöneticisinde yayınlanan iletilerin, kümedeki diğer herhangi bir kuyruk yöneticisinde aboneliklere teslim edilmesini sağlar. Noktadan noktaya ileti sistemine ilişkin olarak, kuyruk yöneticileri arasında iletilerin iletilmesinin maliyeti performansa zarar verebilir. Bu nedenle, iletilerin yayınlanmakta olduğu kuyruk yöneticilerindeki konulara abone olmayı da göz önünde bulunmanız gerekir.

Bir küme içindeki konu anasistem yönlendirmesini kullanırken, aboneliklerin ve yayıncıların yerini, yer alan konu kuyruk yöneticilerine göre de dikkate almak önemlidir. Yayınlayıcı, kümelenmiş bir konunun anasistemi olan bir kuyruk yöneticisine bağlı değilse, yayınlanan iletiler her zaman bir konu barındırma kuyruk yöneticisine gönderilir. Benzer şekilde, kümelenmiş bir konuya ilişkin bir konu anasistemi olmayan bir kuyruk yöneticisinde abonelik yaratıldığında, kümedeki diğer kuyruk yöneticilerinden yayınlanan iletiler her zaman önce kuyruk yöneticisine bir konu barındıran bir konu için gönderilir. Daha özel olarak, abonelik, konuyu barındıran bir kuyruk yöneticisiniyse, ancak aynı konuya da ev sahipliği yapan bir ya da daha çok kuyruk yöneticisi varsa, diğer kuyruk yöneticilerinden gelen yayınların bir kısmı, diğer konu barındıran kuyruk yöneticilerine yöneltilir. Yayıncılar ve abonelikler arasındaki uzaklığı en aza indirmek için bir konu anasistem yönlendirmesi yayınlama/abone olma kümesi tasarlamaya ilişkin ek bilgi için [Topic host roads using merkezleştirilmiş yayıncılar ya da aboneler](#) konusuna bakın

**Yayın trafiği**

Bir uygulamanın bir kümedeki bir kuyruk yöneticisine bağlı olarak yayınlanan iletiler, küme gönderen kanallarını kullanan diğer kuyruk yöneticilerindeki aboneliklere iletilir.

Doğrudan yönlendirme kullandığınızda, yayınlanan iletiler kuyruk yöneticileri arasındaki en kısa yolu alır. Bu, abonelikleri olan kuyruk yöneticilerinin her birine yayınlama kuyruk yöneticisinden doğrudan yönlendirilir. İletiler, konu için abonelikleri olmayan kuyruk yöneticilerine iletilmez. Bkz. [Yayınlama/abone olma](#) ağındaki yetkili sunucu abonelikleri.

Herhangi bir kuyruk yöneticisi ile kümeden başka bir kuyruk yöneticisi arasındaki yayın iletilerinin hızının yüksek olduğu durumlarda, bu iki nokta arasındaki küme kanalı altyapısının hızı korumaları gerekir. Bu işlem, kullanılmakta olan kanalların ve iletim kuyruğunun ayarlanmasını içerebilir.

Konu anasistem yöneltmesini kullanırken, bir olay anasistemi olmayan bir kuyruk yöneticisinde yayınlanan her ileti, bir konu anasistem kuyruk yöneticisine iletilir. Bu, kümedeki bir ya da daha fazla aboneliğin başka bir yerde olup olmadığını bağımsız olarak sağlar. Bu, planlamada dikkate alınacak başka etkenler sunar:

- Her yayını ilk olarak bir konu anasistem kuyruk yöneticisine göndermenin ek gecikme süresi mi kabul edilir?
- Her konu anasistem kuyruk yöneticisi gelen ve giden yayın oranını sürdürebilir mi? Birçok farklı kuyruk yöneticisiyle ilgili yayıncıları olan bir sistemi düşünün. Eğer tüm mesajlarını, çok küçük bir konu barındırma kuyruk yöneticilerine gönderirlerse, bu konu anasistemleri, bu iletilerin işlenmesinde bir darboğaz olabilir ve kuyruk yöneticilerine abone olmak için bunları yönlendirebilir.
- Yayınlanan mesajların önemli bir oranının eşleşen bir aboneye sahip olmadığı mı beklenir? Bu durumda, bu tür iletilerin yayınlanma oranı yüksekse, yayınlayıcının kuyruk yöneticisi bir konu anasistemi yapmak en iyi yol olabilir. Bu durumda, kümede herhangi bir aboneliğin bulunmadığı herhangi bir yayınlanan ileti diğer kuyruk yöneticilerine iletilmeyecektir.

Bu sorunlar, birden çok konu anasistemi tanıtarak, yayın yükünü bunların arasında yaymak için de hafifletebilir:

- Her biri yayın trafiğiyle orantılı olarak birden çok farklı konu varsa, bunları farklı kuyruk yöneticilerine barındırıyorsunuz.
- Konular farklı konu anasistemlerine ayrılamazsa, birden çok kuyruk yöneticisine aynı konu nesnesini tanımlamayı düşünün. Bu, yayınlardaki yayınlarda, yöneltme için her biri için dengeli bir şekilde dengeli bir şekilde sonuçlanıyor. Ancak, bu yalnızca, yayın ileti siparişi gerekli değilse uygun olur.

## Abonelik değişikliği ve dinamik konu dizgileri

Başka bir dikkat edilmesi de, yetkili abonelikleri yaymak için sistemin performansına etkisi olur. Tipik olarak, kuyruk yöneticisi belirli bir kümelenmiş konu dizgisine ilişkin ilk abonelik (yalnızca yapılandırılmış bir konu nesnesi değil) için ilk abonelik yaratıldığında, kümedeki bazı diğer kuyruk yöneticilerine bir yetkili abonelik ileti gönderir. Benzer şekilde, belirli bir kümelenmiş konu dizisine ilişkin son abonelik silindiğinde, yetkili abonelik silme ileti gönderilir.

Doğrudan yöneltme için, abonelikleri olan her kuyruk yöneticisi, bu yetkili sunucu aboneliklerini kümedeki diğer tüm kuyruk yöneticileriyle gönderir. Konu anasistem yöneltmesi için, abonelikleri olan her kuyruk yöneticisi yalnızca, bu kümelenmiş konuya ilişkin bir tanımlı barındıran her kuyruk yöneticisine yetkili sunucu aboneliklerini gönderir. Bu nedenle, doğrudan yönlendirme ile kümede daha fazla kuyruk yöneticisi varsa, bunlar arasında vekil aboneliklerin bakımının ek yükü daha yüksek olur. Ancak, konu anasistem yönlendirmesiyle, kümedeki kuyruk yöneticisi sayısı bir faktör değildir.

Her iki yöneltme modelinde de, bir yayınlama/abone olma çözümü abone olunmakta olan birçok benzersiz konu dizisinden oluşuyorsa ya da kümedeki bir kuyruk yöneticisiyle ilgili konulara sık sık abone olunması ya da abone olunmamış olması durumunda, o kuyruk yöneticisinde önemli bir ek yük görülmeye başlanır ve bunun nedeni, sürekli olarak yetkili aboneliklerin dağıtılması ve silinmesine neden olur. Doğrudan yöneltme ile, bu iletilerin kümedeki her kuyruk yöneticisine gönderilmesi gerekenlerle sıkıştırılır.

Bir konu anasistemi yönlendirilmiş sistem içinde bile aboneliklerin değişiklik hızı çok yüksek olursa, yetkili abonelik ek yükünü azaltma yolları hakkında bilgi için [Yayınlama/abone olma ağlarındaki abonelik performansı](#) başlıklı konuya bakın.

## Küme konularını tanımlama

Küme konuları, **cluster** özneliği tanımlanmış yönetimle ilgili konulardır. Küme konularına ilişkin bilgiler, bir kümenin tüm üyelerine gönderilir ve birden çok kuyruk yöneticilerini kapsayan bir konu alanının bölümlerini oluşturmak için yerel konularla birleştirilir. Bu, bir kuyruk yöneticisinde bir konuda yayınlanan iletilerin kümedeki diğer kuyruk yöneticilerinin aboneliklerine teslim edilmesini sağlar.

Bir kuyruk yöneticisinde küme konusu tanımladığınızda, küme konusu tanımlaması tam havuz kuyruğu yöneticilerine gönderilir. Tam havuzlar daha sonra küme konusu tanımlamasını kümedeki tüm kuyruk yöneticilerine geçirerek, aynı küme konusunu kümedeki herhangi bir kuyruk yöneticisindeki yayıncılar ve aboneler için kullanılabilir hale getirir. Küme konusu yarattığınız kuyruk yöneticisi, küme konusu anasistemi olarak bilinir. Küme konusu kümedeki herhangi bir kuyruk yöneticisi tarafından kullanılabilir, ancak bir küme konusunda yapılan değişiklikler, ilgili konunun tanımlandığı (anasistem) kuyruk yöneticisinde yapılmalıdır; bu noktada değişiklik, tam havuzlar aracılığıyla kümenin tüm üyelerine yayılır.

Doğrudan yönlendirme kullandığınızda, kümelenmiş konu tanımlamasının konumu sistemin davranışını doğrudan etkilemez; çünkü kümedeki tüm kuyruk yöneticileri konu tanımını aynı şekilde kullanır. Bu nedenle, konu gerektiği sürece kümenin bir üyesi olacak herhangi bir kuyruk yöneticisinde konuyu tanımlamalısınız ve bu, düzenli olarak tüm havuz kuyruğu yöneticileriyle iletişim halinde olacak kadar güvenilir bir sistem üzerinde yer alıyor.

Konu anasistem yöneltmesi kullandığınızda, kümelenmiş konu tanımlamasının yeri çok önemlidir; çünkü kümedeki diğer kuyruk yöneticileri bu kuyruk yöneticisine kanal yaratır ve abonelik bilgilerini ve yayınlarını bu yöneticiye gönderir. Konu tanımlamasını barınamak için en iyi kuyruk yöneticisini seçmek için, konu anasistem yöneltmesini anlamanız gerekir. Bkz. [“Yayınlama/abone olma kümelerinde konu anasistem yönlendirmesi” sayfa 95.](#)

Kümelenmiş bir konusunuz ve yerel bir konu nesnesiniz varsa, yerel konu öncelik kazanır. Bkz. [“Aynı adı içeren birden çok küme konu tanımlaması” sayfa 111.](#)

Küme konularını görüntülemek için kullanılacak komutlara ilişkin bilgi için ilgili bilgilere bakın.

## Kümelenmiş konu kalıtımı

Tipik olarak, kümelenmiş bir yayınlama/abone olma topolojisindeki uygulamaların yayınlanması ve abone olunması, bağlı oldukları kümede hangi kuyruk yöneticisi olursa olsun, aynı işlemi beklemez. Bu, kümelenmiş olarak kümelenmiş konu nesnelerinin, kümedeki her kuyruk yöneticisine dağıtılması nedenidir.

Denetlenen bir konu nesnesi, konu ağacındaki diğer yönetilen konu nesnelerinden davranışını devralır. Bu kalıtım, bir konu değiştirgesi için belirtik bir değer ayarlanmadığında oluşur.

Kümelenmiş yayınlama/abone olma durumunda, yayıncıların ve abonelerin, bağlantı kurdukları kuyruk yöneticisine bağlı olarak farklı davranacakları olasılığı tanıtıldığı için bu tür bir mirası göz önünde bulundurmakta önemlidir. Kümelenmiş bir konu nesnesi, daha yüksek konu nesnelerinden edinilecek parametreleri bıraktıysa, bu konu kümedeki farklı kuyruk yöneticilerinde farklı şekilde davranabilir. Benzer şekilde, konu ağacındaki kümelenmiş bir konu nesnesinin altında tanımlanan yerel olarak tanımlanmış konu nesnelere, bu alt konuların hala kümelenmiş olduğu anlamına gelir; ancak, yerel nesne, davranışını kümedeki diğer kuyruk yöneticilerinden farklı olan bir şekilde değiştirebilir.

## Joker abonelikler

Yetkili sunucu abonelikleri, yerel abonelikler, kümelenmiş bir konu nesnesinin üzerinde ya da altında çözülen bir konu dizgisine yapıldığında oluşturulur. Konu sıradüzeninde herhangi bir küme konusundan daha yüksek bir genel arama aboneliği varsa, bu abonelik, eşleşen küme konusu için kümenin etrafında yetkili sunucu abonelikleri gönderilmez ve bu nedenle kümenin diğer üyelerinden herhangi bir yayın almaz. Ancak, yerel kuyruk yöneticisinden yayın alma işlemi gerçekleştirir.

Ancak, başka bir uygulama küme konusuna ya da küme konusuna çözülen bir konu dizgisine abone olursa, yetkili sunucu abonelikleri oluşturulur ve yayınlar bu kuyruk yöneticisine dağıtılır. Özgün, daha yüksek genel arama karakteri aboneliği, bu yayınların geçerli bir alıcısı olarak kabul edilir ve bir kopya alır. Bu davranış gerekli değilse, kümelenmiş konu üzerinde **WILDCARD (BLOCK)** seçeneğini ayarlayın. Bu, özgün genel arama karakterinin geçerli bir abonelik olarak kabul edilmemesine neden olur ve küme konusuna ya da alt konularında herhangi bir yayın (yerel, ya da kümenin başka bir yerinden) almasını durdurur.

## İlgili kavramlar

[Yönetimle ilgili konularla çalışma](#)

[Aboneliklerle çalışma](#)

## İlgili başvurular

### KONU YU GÖRÜNTÜLE

#### TANITIM

#### GÖRÜNTÜLE

##### *Küme konusu öznitelikleri*

Bir konu nesnesi küme adı özniteliği kümesine sahip olduğunda, konu tanımlaması kümedeki tüm kuyruk yöneticilerine dağıtılır. Her kuyruk yöneticisi, yayınlama/abone olma uygulamalarının davranışını denetlemek için yayılan konu özniteliklerini kullanır.

Bir konu nesnesi, yayınlama/abone olma kümeleri için geçerli olan özniteliklere sahip bir sayıdır. Bazıları, yayınlama ve abone olma uygulamalarının genel davranışını ve bazı denetimi, bu konunun küme genelinde nasıl kullanıldığını denetler.

Kümelenmiş bir konu nesnesi tanımlaması, kümedeki tüm kuyruk yöneticilerinin bunu doğru şekilde kullanabilmesi için yapılandırılmalıdır.

Örneğin, yönetilen abonelikler için kullanılacak model kuyrukları ( MDURMDL ve MNDURMDL ) varsayılan olmayan bir kuyruk adına ayarlanır; bu adlandırılmış model kuyruğu, yönetilen aboneliklerin yaratılacağı tüm kuyruk yöneticilerinde tanımlanmalıdır.

Benzer şekilde, herhangi bir öznitelik ASPARENT olarak ayarlanırsa, konunun davranışı, konu ağacındaki daha üst düğümlere bağımlı olur (bkz. [Denetim konusu nesnelere](#) ) kümedeki her bir kuyruk yöneticisinde yer alan her bir kuyruk yöneticisinde Bu, farklı kuyruk yöneticilerinden yayınlama ya da abone olma sırasında farklı davranışa neden olabilir.

Küme genelinde yayınlama/abone olma davranışlarıyla doğrudan ilişkili olan ana öznitelikler aşağıdaki gibidir:

#### **CLROUTE**

Bu değiştirge, yayıncıların bağlı olduğu kuyruk yöneticileri ile eşleşen aboneliklerin bulunduğu kuyruk yöneticileri arasındaki iletilerin yönlendirmesini denetler.

- Rotayı, bu kuyruk yöneticileri arasında doğrudan ya da kümelenmiş konunun tanımını barındıran bir kuyruk yöneticisi arasında doğrudan olacak şekilde yapılandırıyorsunuz. Daha fazla ayrıntı için bkz. [Kümeleri yayınlama/abone olma](#) .
- **CLUSTER** parametresi ayarlıyken **CLROUTE** parametresini değiştiremezsiniz. **CLROUTE** ' ı değiştirmek için, önce **CLUSTER** özelliğini boş olacak şekilde ayarlayın. Bu, konuyu kümelenmiş bir şekilde hareket etmek için kullanan uygulamaları durdurur. Bu, aboneliklere teslim edilmekte olan yayınlardaki bir kesme sonucu ortaya çıktı; bu nedenle, değişikliği yaparken yayınlama/abone olma ileti sistemini de susturmalısınız.

#### **PROXYSUB**

Bu parametre, yetkili aboneliklerin ne zaman yapıldığını denetler.

- **FIRSTUSE** varsayılan değerdir ve yetkili sunucu aboneliklerinin, dağıtılmış yayınlama/abone olma topolojisindeki bir kuyruk yöneticisindeki yerel aboneliklere yanıt olarak gönderilmesini sağlar ve artık gerekli olmayan bir iptal işlemi iptal edilir. For details about why you might want to change this attribute from the default value of **FIRSTUSE**, see [Bireysel yetkili sunucu aboneliği iletimi ve her yerde yayınlama](#) .
- *her yerde yayınlama* ' ı etkinleştirmek için, üst düzey bir konu nesnesi için **PROXYSUB** parametresini **FORCE** olarak ayarladınız. Bu, konu ağacındaki bu konu nesnesinin altındaki tüm konularla eşleşen tek bir genel arama yetkili aboneliği sonuçlarıyla sonuçlanır.

**Not:** Büyük ya da meşgul bir yayınlama/abone olma kümesinde **PROXYSUB (FORCE)** özniteliğinin ayarlanması sistem kaynaklarında aşırı yüklenmeye neden olabilir. **PROXYSUB (FORCE)** özniteliği, yalnızca konunun tanımlandığı kuyruk yöneticisi değil, her kuyruk yöneticisine dağıtılır. Bu, kümedeki her kuyruk yöneticisinin genel bir yetkili sunucu aboneliği oluşturmasına neden olur.

Kümedeki herhangi bir kuyruk yöneticisinde yayınlanan bu konuya ileti kopyası, **CLROUTE** ayarına bağlı olarak, kümedeki her kuyruk yöneticisine doğrudan ya da bir konu anasistem kuyruk yöneticisinden gönderilir.

Konu doğrudan yöneltildiğinde, her kuyruk yöneticisi diğer kuyruk yöneticisine küme gönderen kanalları yaratır. Konu anasistem yöneltildiğinde, her bir konu anasistem kuyruk yöneticisini içeren kanallar kümedeki her kuyruk yöneticisinden yaratılır.

Kümelerde kullanıldığında **PROXYSUB** parametresiyle ilgili daha fazla bilgi için bkz. [Doğrudan yönlendirilen yayınlama/abone olma performansı](#).

### **PUBCOBE ve SUBSCOPE**

Bu parametreler, bu kuyruk yöneticisinin topolojideki kuyruk yöneticilerine (yayınlama/abone olma kümesi ya da sıradüzeni) yayınların yayılıp yayılmayacağını ya da kapsamı yalnızca yerel kuyruk yöneticisiyle sınırlayıp geçirmedini belirler. You can do the equivalent job programmatically using MQPMO\_SCOPE\_QMGR and MQSO\_SCOPE\_QMGR.

#### **PUBSCOPE**

Bir küme konusu nesnesi **PUBSCOPE (QMGR)** ile tanımlandıysa, tanım kümeyle paylaşılır, ancak bu konuya dayalı yayınların kapsamı yalnızca yereldir ve kümedeki diğer kuyruk yöneticilerine gönderilmez.

#### **SUBSCOPE**

Bir küme konusu nesnesi **SUBSCOPE (QMGR)** ile tanımlandıysa, tanım kümeyle paylaşılır, ancak o konuya dayalı aboneliklerin kapsamı yalnızca yereldir; dolayısıyla, kümedeki diğer kuyruk yöneticilerine hiçbir yetkili abonelik gönderilmez.

Bu iki öznelik, bir kuyruk yöneticisini belirli konu başlıklarına ilişkin kümenin diğer üyeleriyle etkileşimde bulunan bir kuyruk yöneticisini yalıtım için genellikle birlikte kullanılır. Kuyruk yöneticisi, kümedeki diğer üyelerden ve diğer üyelerden yayınların yayınlarını ya da yayınlarını yayınlamayı ya da almalarını sağlar. Konu nesnelere alt konularda tanımlandıysa, bu durum yayınlanmasını ya da aboneliği önlememektedir.

Setting **SUBSCOPE** to MMGR on a local definition of a topic does not prevent other queue managers in the cluster from propagating their proxy subscriptions to the queue manager if they are using a clustered version of the topic, with **SUBSCOPE (ALL)**. However, if the local definition also sets **PUBSCOPE** to MMGR those proxy subscriptions are not sent publications from this queue manager.

### **İlgili kavramlar**

[Yayın kapsamı](#)

[Abonelik kapsamı](#)

*Aynı adı içeren birden çok küme konu tanımlaması*

Kümedeki birden çok kuyruk yöneticisinde aynı adlandırılmış küme konusu nesnesini tanımlayabilir ve belirli senaryolarda bu belirli davranışı etkinleştirebilir. Aynı adda birden çok küme konusu tanımı varsa, özelliklerin büyük çoğunluğu eşleşmelidir. Aksi takdirde, hatalar ya da uyarılar, uyumsuzluğa bağlı olarak raporlanır.

Genel olarak, birden çok küme konu tanımının özelliklerinde bir uyumsuzluk varsa, uyarılar yayınlanır ve konu nesnesi tanımlamalarından biri kümedeki her kuyruk yöneticisi tarafından kullanılır. Her kuyruk yöneticisi tarafından kullanılan tanımlama belirlenimci değildir ya da kümedeki kuyruk yöneticilerinde tutarlı değildir. Bu tür yanlış eşleşmeler mümkün olduğunca çabuk çözülmelidir.

Küme kurulumu ya da bakımı sırasında bazen aynı olmayan birden çok küme konusu tanımlaması yaratmanız gerekir. Ancak, bu yalnızca geçici bir ölçüm olarak yararlı olur ve bu nedenle olası bir hata durumu olarak değerlendirilir.

Uyumsuzluk saptandığında, her kuyruk yöneticisinin hata günlüğüne aşağıdaki uyarı iletileri yazılır:

- **Multi** Multiplatform sistemlerinde [AMQ9465](#) ve [AMQ9466](#).
- **z/OS** z/OS sistemlerinde [CSQX465I](#) ve [CSQX466I](#).



Her kuyruk yöneticisinden herhangi bir konu dizisine ilişkin seçilen özellikler, örneğin **DISPLAY TPSTATUS** kullanılarak konu nesne tanımlamalarından ziyade konu durumu görüntüleyerek saptanabilir.

Bazı durumlarda, yapılandırma özelliklerinde bir çakışma, yaratılmakta olan nesne nesnesinin yaratılmasına ya da eşleşmeyen nesnelere geçersiz olarak imlenmesine ve küme boyunca yayılmamasına



neden olur ( [KONUYU GÖRÜNTÜLE](#) içinde **CLSTATE** konusuna bakın). Bu durumlar, küme yöneltme özelinde bir çakışma olduğunda oluşur ( **CLROUTE** ) (konu tanımları). Ek olarak, konu ana makine yönlendirmeleri genelinde tutarlılığın önemi nedeniyle, bu makalenin sonraki bölümlerinde ayrıntılı olarak tutarsızlıklar da reddedilir.

Çakışma, nesnenin tanımlandığı sırada saptanırsa, yapılandırma değişikliği reddedilir. Daha sonra tam havuz kuyruğu yöneticileri tarafından saptanırsa, kuyruk yöneticileri hata günlüklerine aşağıdaki uyarı iletileri yazılır:

-  Multiplatformsüzerinde: [AMQ9879](#)
-  z/OSüzerinde: [CSQX879E](#).

Kümede aynı konu nesnesinin birden çok tanımı tanımlandığında, yerel olarak tanımlanmış bir tanımlama uzaktan tanımlanan herhangi bir tanımdan öncelikli olarak uygulanır. Bu nedenle, tanımlarda herhangi bir farklılık varsa, birden çok tanımı barındıran kuyruk yöneticileri birbirlerinden farklı davranır.

## **Başka bir kuyruk yöneticisinden küme konularıyla aynı adı taşıyan, kümeli olmayan bir konu tanımlanmasını sağlar.**

Bir kümede bulunan bir kuyruk yöneticisinde kümelenmemiş, denetlenen bir konu nesnesini tanımlamak ve farklı bir kuyruk yöneticisinde kümelenmiş bir konu tanımlamasıyla aynı adı taşıyan konu nesnesini aynı anda tanımlamak mümkündür. Bu durumda, yerel olarak tanımlanmış konu nesnesi, aynı addaki tüm uzak tanımlamalara göre öncelik kazanır.

Bu, bu kuyruk yöneticisinden kullanıldığında konunun kümeleme davranışını önlememe etkisine sahiptir. Bu, abonelikler uzak yayıncılardan yayınlar almayabilir ve yayıncılardan gelen iletiler kümedeki uzak aboneliklere yayılamayabilir.

Böyle bir sistemi yapılandırmadan önce dikkatli olunması gerekir, çünkü bu durum kafa karıştırıcı davranışlara yol açabilir.

**Not:** Tek bir kuyruk yöneticisinin, yayınların ve aboneliklerin kümenin etrafında yayılmasını önlememesi gerekiyorsa, konu başka bir yerde kümelense bile, yayını ve abonelik kapsamlarını yalnızca yerel kuyruk yöneticisine ayarlamak için alternatif bir yaklaşım gerekir. Bkz. [“Küme konusu öznelikleri” sayfa 110.](#)

## **Doğrudan yönlendirilen bir kümede birden çok küme konu tanımlaması**

Doğrudan yöneltme için, genellikle aynı küme konusunu birden çok küme kuyruk yöneticisinden tanımlamanız. Bunun nedeni, doğrudan yöneltme, hangi kuyruk yöneticisi üzerinde tanımlanmakta olursa olsun, bu konunun kümedeki tüm kuyruk yöneticilerinde kullanılabilir olmasını sağlar. Ayrıca, birden çok küme konusu tanımı eklenmesi, sistem etkinliklerini ve yönetimsel karmaşıklığı önemli ölçüde artırır ve daha fazla karmaşıklık, insan hatasının daha yüksek bir şansa sahip olduğunu sağlar:

- Her tanım, diğer küme konusu anasistem kuyruk yöneticileri de dahil olmak üzere kümedeki diğer kuyruk yöneticilerine itilmekte olan ek bir küme konusu nesnesiyle sonuçlanır.
- Bir kümedeki belirli bir konuya ilişkin tüm tanımların aynı olması gerekir; tersi durumda, bir kuyruk yöneticisi tarafından hangi konu tanımlamasının kullanıldığını çözmesi zordur.

Küme konu tanımlaması, tam havuz kuyruğu yöneticileri tarafından ve kısmi küme havuzlarındaki diğer tüm kuyruk yöneticileri tarafından ön belleğe alındığından, tek anasistem kuyruk yöneticisinin küme boyunca doğru bir şekilde çalışması için sürekli olarak kullanılabilir olması da şart değildir. Daha fazla bilgi için bakınız: [Availability of topic host queue manager that use direct roady.](#)

İkinci bir kuyruk yöneticisinde geçici olarak bir küme konusunu geçici olarak tanımlamanız gereken bir durum için, örneğin, konunun var olan anasistemi kümeden kaldırılabilirken, [Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması](#) başlıklı konuya bakın.

Bir küme konusu tanımlamasını değiştirmeniz gerekirse, bu tanımı, tanımlanmış olduğu kuyruk yöneticisinde değiştirmeye özen gösteriniz. Başka bir kuyruk yöneticisinden değiştirme girişiminde bulunulması, çakışan konu öznelikleri ile konunun ikinci bir tanımlamasını yanlışlıkla yaratabilir.



## Bir konu anasistem tarafından yönltilen birden çok küme konu tanımlaması

Bir küme konusu *konu anasistem* küme rotasıyla tanımlandığında, konu, *doğrudan* yönltilen konular için olduğu gibi kümedeki tüm kuyruk yöneticilerine dağıtılır. Buna ek olarak, ilgili konuya ilişkin tüm yayınlama/abone olma ileti alışverişi, o konunun tanımlandığı kuyruk yöneticilerine yönltilmektedir. Bu nedenle, kümedeki konuya ilişkin tanımların yeri ve sayısı önemli olur (bkz. [“Yayınlama/abone olma kümelerinde konu anasistem yönlendirmesi” sayfa 95](#)).

Yeterli düzeyde kullanılabilirliği ve ölçeklenebilirliği sağlamak için, mümkünse birden çok konu tanımlarına sahip olmak yararlı olur. Bkz. [Konu anasistem yönlendirmesini kullanan anasistem kuyruğu yöneticilerinin kullanılabilirliği](#).

Bir kümedeki *konu anasistemi* routed konusuna ilişkin ek tanımlamaları eklerken ya da eklerken, yapılanış değişikliği sırasında ileti akışını göz önünde bulundurmanız gerekir. İleti, değişiklik sırasında kümede yayınlanmakta ise, bir konu tanımlaması eklemek ya da kaldırmak için aşamalandırılmış bir işlem gerekir. Bkz. [Küme konusu tanımlamasının farklı bir kuyruk yöneticisine taşınması](#) ve [Bir konu ana makinelerin yönlendirilmiş kümesine ek konu anasistemleri eklenmesi](#).

As previously explained, the properties of the multiple definitions should match, with the possible exception of the **PUB** parameter, as described in the next section. Yayınlar, konu anasistem kuyruk yöneticilerine yönltildiğinde, birden çok tanımlama için daha da önemli olur. Bu nedenle, konu ana makine kümesi yönlendirmesi için bir ya da daha fazla konu tanımlaması yapılandırıldıysa, konu dizisinde ya da küme adında bir tutarsızlık reddedilir.

**Not:** Küme konu tanımları, konu ağacındaki başka bir konunun yukarısında ya da altında yapılandırılmaya çalışıldığında, var olan kümelenmiş konu tanımlamasının konuta anasistem yönltilmesi için yapılandırıldığı durumlarda da reddedilir. Bu, genel yayın aboneliklerine göre yayınların yönltilmesinde belirsizliğini önler.

## PUB parametresine ilişkin özel işleme

The **PUB** parameter is used to control when applications can publish to a topic. Bir kümede ana program yönltilmesi durumunda, yayınları yönltilmek için hangi konu anasistem kuyruk yöneticilerinin kullanıldığını da denetleyebilir. For this reason it is permitted to have multiple definitions of the same topic object in the cluster, with different settings for the PUB parameter.

Bir konunun birden çok uzak kümelenmiş tanımlaması bu parametre için farklı ayarlara sahip olursa, aşağıdaki koşullar karşılanırsa, konu yayınların aboneliklere gönderilmesine ve teslim edilmesine olanak sağlar:

- There is not a matching topic object defined on the queue manager that the publisher is connected to that is set to PUB (DISABLED).
- Kümedeki birden çok konu tanımlamalarından biri ya da daha fazlası PUB (ENABLED) olarak ayarlandı ya da birden çok konu tanımlamasının bir ya da daha çoğu PUB (ASPARENT) olarak ayarlandı ve yayınlayıcının bağlı olduğu yerel kuyruk yöneticileri ve tanımlanan abonelik, konu ağacındaki daha yüksek bir noktada PUB (ENABLED) olarak ayarlandı.

Konu anasistem yönltilmesi için, iletiler, konu anasistemleri olmayan kuyruk yöneticilerine bağlı uygulamalar tarafından yayınlandığında, iletiler yalnızca **PUB** parametresinin belirttik olarak DISABLE olarak ayarlanmamış olduğu ana kuyruk yöneticilerine yönltilir. Bu nedenle, belirli konu anasistemleri aracılığıyla ileti trafiğini susturma için PUB (DISABLE) ayarını kullanabilirsiniz. Bunu, bir kuyruk yöneticisini bakım ya da kaldırma işlemi için ya da [Ek konu anasistemlerinin bir konuya yönlendirilmiş küme olarak eklenmesibaşlıklı](#) konuda açıklanan nedenlerle hazırlanması isteyebilirsiniz.

### *Küme konusu anasistem kuyruk yöneticilerinin kullanılabilirliği*

Yayınlama/abone olma kümenizi, bir konu anasistem kuyruk yöneticisinin kullanılamaz duruma gelmesi riskini en aza indirmek için, küme artık konu için trafiği işleyecek şekilde tasarlayamaz. Bir konu anasistem kuyruk yöneticisinin kullanılamaz duruma gelmesi, kümenin konu anasistem yönltilmesi mi, yoksa doğrudan yönltilme mi kullanıyor olduğuna bağlıdır.

## **Doğrudan yönlendirmeyi kullanan ana makine kuyruk yöneticilerinin kullanılabilirliği**

Doğrudan yönlendirme için, genellikle aynı küme konusunu birden çok küme kuyruk yöneticisinden tanımlamamızdır. Bunun nedeni, doğrudan yönlendirme, hangi kuyruk yöneticisi üzerinde tanımlanmakta olursa olsun, bu konunun kümedeki tüm kuyruk yöneticilerinde kullanılabilir olmasını sağlar. Bkz. [Doğrudan yönlendirilen bir kümede birden çok küme konusu tanımı](#).

Bir kümede, kümelenmiş bir nesnenin (örneğin, kümelenmiş bir kuyruk ya da kümelenmiş bir konu gibi) her anasistem uzun süre kullanılamaz duruma geldiğinde, kümenin diğer üyeleri sonunda bu nesnelerin bilgilerini sona erdirir. Kümelenmiş bir konu durumunda, küme konusu anasistem kuyruk yöneticisi kullanılamaz hale gelirse, diğer kuyruk yöneticileri konu için yayınlama/abone olma isteklerini doğrudan kümelenmiş bir şekilde (uzak kuyruk yöneticilerindeki aboneliklere gönderme), en az 60 gün sonra, tam havuz kuyruğu yöneticileriyle iletişimde en son olduğu zaman bu konuya ilişkin yayınlama/abone olma isteklerini işlemeye devam eder. Küme konusu nesnesini tanımladığınız kuyruk yöneticisi bir daha kullanılabilir kılınmadıysa, diğer kuyruk yöneticilerindeki önbelleğe alınan konu nesnelere silinir ve konu, yerel bir konuya geri döner. Bu durumda abonelikler, uzak kuyruk yöneticilerine bağlı uygulamalardan yayınları almayı sona erdirir.

Bir küme konusu nesnesi tanımladığınız kuyruk yöneticisini kurtarmak için 60 günlük süre ile, bir küme konusu anasisteminin kullanılabilir durumda kalacağını garanti etmek için özel ölçümler almaya gerek vardır (ancak, kullanılmayan küme konusu anasistemde tanımlı olan aboneliklerin kullanılabilir kalmaması). 60 günlük süre teknik sorunlar için yeterli olacaktır ve yalnızca yönetici hataları nedeniyle aşılması muhtemel. Bu olasılığı en aza indirmek için, küme konusu anasistemi kullanılamıyorsa, önbelleğe alınan küme konusu nesnesinin yenilenmediğini belirten, kümenin tüm üyeleri saatlik olarak hata günlüğü iletileri yazar. Küme konusu nesnesinin tanımlı olduğu kuyruk yöneticisinin çalışmakta olduğundan emin olarak bu iletilere yanıt verin. Küme konusu anasistem kuyruk yöneticisini yeniden kullanılabilir duruma getirmek mümkün değilse, aynı kümelenmiş konu tanımlamasını, kümedeki başka bir kuyruk yöneticisinde tam olarak aynı özniteliklerle tanımlayın.

## **Konu anasistem yönlendirmesini kullanan anasistem kuyruğu yöneticilerinin kullanılabilirliği**

Konu anasistem yönlendirmesi için, bir konuya ilişkin tüm yayınlama/abone olma ileti alışverişi, o konunun tanımlandığı kuyruk yöneticilerine yöneltilir. Bu nedenle, kümedeki bu kuyruk yöneticilerinin sürekli olarak kullanılabilirliğini göz önünde bulundurmanız çok önemlidir. Bir konu anasistemi kullanılamaz hale gelirse ve konu için başka bir anasistem yoksa, yayıncılardan farklı kuyruk yöneticilerindeki abonelere giden trafik, konuya ilişkin hemen hemen haller oluşturur. Ek konu anasistemleri varsa, küme kuyruğu yöneticileri bu konu anasistemleri aracılığıyla yeni yayın trafiğini yönlendirir, ileti rotalarının sürekli olarak kullanılabilirliğini sağlar.

Doğrudan konu başlıklarına gelince, 60 gün sonra, ilk konu anasistemi hala kullanılamazsa, bu konu anasisteminin konusu, kümeden kaldırılır. Bu, kümede bu konu için kalan son tanımsa, diğer tüm kuyruk yöneticileri, yönlendirme için herhangi bir konu ana makinesinde yayınlarını ileriye doğru bırakmaktan vazgeçer.

Uygun kullanılabilirliği ve ölçeklenebilirliği sağlamak için, mümkünse, her konuyu en az iki küme kuyruğu yöneticisinde tanımlamak için yararlı olacak şekilde yararlı olur. Bu, belirtilen herhangi bir konu anasistem kuyruk yöneticisinin kullanılamaz duruma gelmesine karşı koruma sağlar. Ayrıca bkz. [Bir konu ana makinesi kümesinde birden çok küme konusu tanımı](#).

Birden çok konu anasistemi yapılandıramıyorsanız (örneğin, ileti sıralamasını korumak için) ve tek bir konu anasisteminin yapılandıramıyorsanız (tek bir kuyruk yöneticisinin kullanılabilir olması, kümedeki tüm kuyruk yöneticilerindeki aboneliklere ilişkin akışı etkilememesi gerektiğinden), konuyu doğrudan yönlendirilmiş bir konu olarak tanımlamayı düşünmenizi sağlar. Bu, tüm küme için tek bir kuyruk yöneticisine bağımlılığı önler, ancak yine de yerel olarak barındırılan abonelikleri ve yayıncıları işlemesi için her bir kuyruk yöneticisinin kullanılabilir olmasını gerektirir.

## ***Kümelenmiş yayınlama/abone olma engelini engelle***

Doğrudan yönlendirilen ilk kümelenmiş konuyu bir küme içinde tanıtarak, kümedeki her kuyruk yöneticisi diğer tüm kuyruk yöneticilerinden haberdar olmak üzere kümedeki her kuyruk yöneticisinde ve potansiyel olarak birbirleri için kanallar oluşturmalarına neden olur. Bu istenmiyorsa, topun yönlendirilmiş yayınlama/abone olma özelliğini yapılandırmak yerine bu işlemi yapılandırmanız gerekir. Doğrudan yönlendirilen kümelenmiş bir konunun varlığı, her kuyruk yöneticisinin ölçekleme endişeleri nedeniyle kümenin istikrarını tehlikeye atsa da, kümedeki her kuyruk yöneticisinde **PSCLUS** ayarını DISABLE değerine ayarlayarak kümelenmiş yayınlama/abone olma işlevselliğini tamamen geçersiz kılabilirsiniz.

“Yayınlama/abone olma kümelerinde doğrudan yönlendirme” sayfa 90' ta açıklandığı gibi, bir kümeye doğrudan yönlendirilmiş kümelenmiş bir konuyu tanıtdığınızda, tüm kısmi havuzlar kümenin diğer tüm üyeleri tarafından otomatik olarak bilgilendirilir. Kümelenmiş konu diğer tüm düğümlerde (örneğin, **PROXYSUB (FORCE)** belirtildiğinde) abonelikler de yaratabilir ve yerel abonelikler olmadığında da, kuyruk yöneticisinden çok sayıda kanal başlatılmasına neden olabilir. Bu, kümedeki her kuyruk yöneticisine anında ek yük koyar. Birçok kuyruk yöneticisi içeren bir küme için bu, performansın önemli ölçüde azaltılmasına neden olabilir. Bu nedenle, bir kümeye doğrudan yönlendirilmiş yayınlama/abone olma olanağının tanıtımı dikkatli bir şekilde planlanmalıdır.

Bir kümenin doğrudan yönlendirilen yayınlama/abone olma genel başkanlarını barındıramamasını biliyorsanız, bunun yerine, konu sahibi yayınlama/abone olma olanağını kullanabilirsiniz. Farklılıklara genel bakış için bkz. “Yayınlama/abone olma kümelerini tasarlama” sayfa 88.

If you prefer to completely disable publish/subscribe functionality for the cluster, you can do so by setting the queue manager attribute **PSCLUS** to DEVRE Dışı on every queue manager in the cluster. Bu ayar, kuyruk yöneticisi işlevinin üç özelliğini değiştirerek, kümedeki doğrudan yönlendirilmiş ve konu ana makinelerinin yayınlama/abone olma özelliğini devre dışı bırakır:

- Bu kuyruk yöneticisinin yöneticisi artık, kümelenmiş olarak bir Topic nesnesi tanımlayamaz.
- Diğer kuyruk yöneticilerinden gelen konu tanımları ya da yetkili sunucu abonelikleri reddedilir ve yanlış yapılandırmayı denetimciye bildirmek için bir uyarı iletisi günlüğe kaydedilir.
- Tam havuzlar, bir konu tanımlaması aldıklarında, diğer tüm kısmi havuzlarla ilgili bilgileri artık otomatik olarak paylaşmaz.

**PSCLUS** , bir kümedeki her bir kuyruk yöneticisinin bir parametresi olsa da, kümedeki bir kuyruk yöneticisi alt kümesinde yayınlama/abone olma işlemi seçmeli olarak devre dışı bırakması amaçlanmaz. Bu şekilde geçersiz bir şekilde devre dışı bırakırsanız, sık sık hata iletileri görürsünüz. This is because proxy subscriptions and topic definitions are constantly seen and rejected if a topic is clustered on a queue manager where **PSCLUS** is enabled.

Bu nedenle, kümedeki her kuyruk yöneticisinde **PSCLUS** ' i DISABLE olarak ayarlamayı hedeflemeniz gerekir. Ancak, uygulamada bu durumun elde edilmesi ve bakımı zor olabilir; örneğin, kuyruk yöneticileri her an kümeden katılabilir ve ayrılabilirler. En azından, tüm havuz kuyruğu yöneticilerindeki **PSCLUS** ' un DISABLE olarak ayarlandığından emin olmanız gerekir. Bunu yaparsanız ve kümelenmiş bir konu daha sonra kümedeki bir ETKİN kuyruk yöneticisinde tanımlandıysa, bu durum tüm havuzların diğer kuyruk yöneticilerinin her kuyruk yöneticisini bilgilendirmesine neden olmaz ve bu nedenle kümeniz tüm kuyruk yöneticilerindeki olası ölçekleme sorunlarından korunur. Bu senaryoda, kümelenmiş konunun kökeni, tam havuz kuyruğu yöneticilerinin hata günlüklerinde bildirilir.

Bir kuyruk yöneticisi bir ya da daha fazla yayınlama/abone olma kümesine ve ayrıca bir ya da daha çok noktadan noktaya iletişim kümesine katılırsa, o kuyruk yöneticisinde **PSCLUS** ögesini ETKİNLEŞTİRİLDİ olarak ayarlamalısınız. Bu nedenle, yayınlama abone olma kümesiyle bir noktadan noktaya iletişim kümesi çakışırken, her kümede ayrı bir tam havuz kümesi kullanmalısınız. Bu yaklaşım, her kuyruk yöneticisine ilişkin konu tanımlarının ve bilgilerin yalnızca yayınlama/abone olma kümesinde akmasına olanak sağlar.

**PSCLUS** 'u ENABLE ' den DISABLEolarak değiştirdiğinizde tutarsız yapılandırmaları önlemek için, bu kuyruk yöneticisinin üyesi olduğu herhangi bir kümede kümelenmiş konu nesnesi bulundurmaz. Any such topics, even remotely defined ones, must be deleted before changing **PSCLUS** to DEVRE Dışı.

**PSCLUS** ile ilgili daha fazla bilgi için bkz. [ALTER QMGR \(PSCLUS\)](#).

## İlgili kavramlar

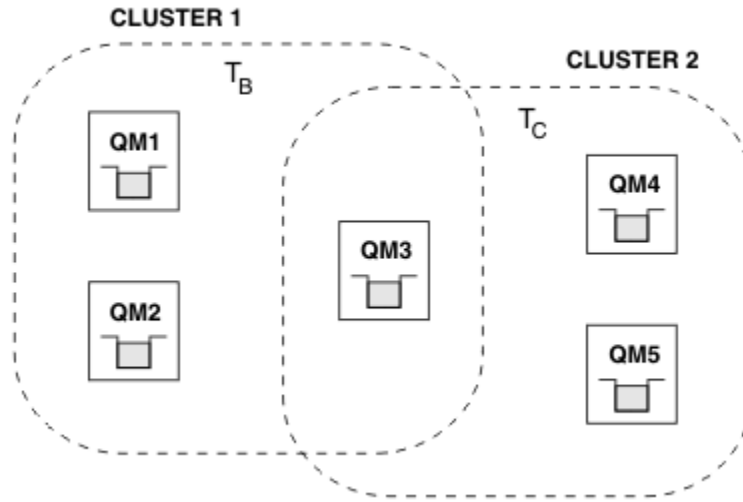
Doğrudan yönlendirilen yayınlama/abone olma kümesi performansı

### **Yayınlama/abone olma ve birden çok küme**

Tek bir kuyruk yöneticisi birden çok kümenin üyesi olabilir. Bu düzenleme bazen *üst üste binen kümeler* olarak bilinir. Bu tür bir çakışma ile, kümelenmiş kuyruklar birden çok kümeden erişilebilir hale getirilebilir ve noktadan noktaya ileti trafiği bir kümedeki kuyruk yöneticilerinden başka bir kümedeki kuyruk yöneticilerine yönlendirilebilir. Yayınlama/abone olma kümelerindeki kümelenmiş konular aynı yeteneği sağlamaz. Bu nedenle, birden çok küme kullanılırken davranışlarının açıkça anlaşılması gerekir.

Bir kuyruktan farklı olarak, bir konu tanımlamasını birden çok kümeyle ilişkilendiremezsiniz. Kümelenmiş bir konunun kapsamı, konunun tanımlandığı kümede yer alan kuyruk yöneticileriyle sınırlıdır. Bu, yayınların yalnızca aynı kümedeki kuyruk yöneticilerindeki aboneliklere yayılmasına izin verir.

### **Bir kuyruk yöneticisinin konu ağacı**



Şekil 28. Çakışan kümeler: Her biri farklı konulara abone olan iki küme

Bir kuyruk yöneticisi birden çok kümenin üyesi ise, bu kümelerin her birinde tanımlanan tüm kümelenmiş konulardan haberdar olur. Örneğin, önceki şekilde QM3, hem T<sub>B</sub> hem de T<sub>C</sub> denetimli kümelenmiş konu nesnelere sahiptir; oysa QM1 yalnızca T<sub>B</sub>'yi bilir. QM3, her iki konu tanımını da yerel konusuna uygular ve bu nedenle, belirli konular için QM1'e farklı bir işlev uygular. Bu nedenle, farklı kümelerdeki kümelenmiş konuların birbirine karışmaması önemlidir. Parazit, farklı bir kümede (örneğin, /Sport ve /Sport/Football konu dizilerine sahip) ya da her ikisinde de aynı konu dizisi için başka bir kümelenmiş konunun üstünde ya da altında tanımlandığında ortaya çıkabilir. Başka bir müdahale biçimi, denetlenen kümelenmiş konu nesnelere farklı kümelerde aynı nesne adıyla, ancak farklı konu dizileri için tanımlanmasıdır.

Böyle bir yapılandırma yapılırsa, yayınların eşleşen aboneliklere teslim edilmesi, yayıncıların ve abonelerin kümeye ilişkin görece konumlarına çok bağlıdır. Bu nedenle, böyle bir yapılandırmaya güvenemezsiniz ve bunu, müdahale eden konuları kaldırmak için değiştirmeniz gerekir.

Yayınlama/abone olma ileti sistemiyle çakışan bir küme topolojisi planlarken, konu ağacını ve kümelenmiş konu nesnesi adlarını topolojideki tüm çakışan kümelere yayılıyor gibi işleyerek herhangi bir parazit oluşmasını önleyebilirsiniz.

### **Birden çok yayınlama/abone olma kümesinin bütünleştirilmesi**

Farklı kümelerdeki kuyruk yöneticilerini kapsamak için yayınlama/abone olma ileti sistemi gereksinimi varsa, iki seçenek vardır:

- Bir yayınlama/abone olma sıradüzeni yapılandırması kullanarak kümeleri birbirine bağlayın. Bkz. [Birden çok kümenin konu alanlarını birleştirme](#).
- Var olan kümelerle örtüşen ve belirli bir konuyu yayınlaması ya da bu konuya abone olması gereken tüm kuyruk yöneticilerini içeren ek bir küme oluşturun.

İkinci seçenikle, kümenin boyutunu ve en etkili küme yönlendirme mekanizmasını dikkatli bir şekilde göz önünde bulundurmalısınız. Bkz. “Yayınlama/abone olma kümelerini tasarlama” sayfa 88.

## ***Yayınlama/abone olma kümelerinde tutulan yayınlarda dikkat edilmesi gereken noktalar***

Alıkonan yayınlarla çalışmak için bir yayınlama/abone olma kümesi tasarlarken göz önünde bulundurulması gereken birkaç kısıtlama vardır.

### **Dikkat Edilecekler**

*Dikkat Edilmesi 1:* Aşağıdaki küme kuyruğu yöneticileri, her zaman alıkonan bir yayının en son sürümünü saklar:

- Yayıncının kuyruk yöneticisi
- Bir konu ana makinesinde, konu ana makinesi (bu makalenin sonraki bölümünde açıklandığı gibi, konu için yalnızca bir konu ana makinesi sağlanır)
- Abonelikleri olan tüm kuyruk yöneticileri, alıkonan yayının konu dizisiyle eşleşiyor

*Dikkat Edilecek Noktalar 2:* Kuyruk yöneticileri, abonelikleri olmadığında güncelleştirilen alıkonan yayınları almaz. Bu nedenle, artık konu için abone olmayan bir kuyruk yöneticisine saklanan tüm alıkonan yayın eski haline gelir.

*Dikkate değer 3:* Herhangi bir abonelik oluştururken, konu dizisine ilişkin alıkonan bir yayının yerel kopyası varsa, yerel kopya aboneliğe teslim edilir. Herhangi bir konu dizisinin ilk abonesiyeniz, eşleşen bir alıkonan yayın aşağıdaki küme üyelerinden birinden de sağlanır:

- Doğrudan yönlendirilen bir kümede, yayıncının kuyruk yöneticisi
- Bir konu ana makinesinde, belirtilen konuya ilişkin konu anasistemleri

Alıkonan bir yayının bir konu anasisteminden ya da yayınlama kuyruk yöneticisinden abone olunan kuyruk yöneticisine gönderilmesi, MQSUB çağrılarının zamanuyumsuz olması olabilir. Bu nedenle, MQSUBRQ çağrısını kullanırsanız, daha sonra MQSUBRQ' a yapılan bir çağrıya kadar alıkonan en son yayın kaçırılabilir.

### **Çıkarımlar**

Herhangi bir yayınlama/abone olma kümesi için, ilk abonelik yapıldığında, yerel kuyruk yöneticisi alıkonan bir yayının eski bir kopyasını saklayabilir ve bu da yeni aboneliğe teslim edilen kopyadır. Yerel kuyruk yöneticisine ilişkin bir aboneliğin varlığı, saklanan yayının bir sonraki güncellenmesinin çözümüne anlamına gelir.

Bir konu anasistemi yayınlama/abone olma kümesinde, belirli bir konu için birden çok konu anasistemi yapılandırırırsanız, yeni aboneler en son tutulan yayını bir konu anasisteminden alabilir ya da başka bir konu anasisteminden (en son kaybedilen) eski bir alıkonan yayını alabilirler. Konu anasistem yöneltmesi için, belirli bir konu için birden çok konu anasisteminin yapılandırılması olağandır. Ancak, uygulamaların alıkonan yayınlardan kullanılmasını bekliyorsanız, her konu için tek bir konu anasistemi yapılandırmanızdır.

Herhangi bir konu dizisi için, yalnızca tek bir yayıncı kullanmalısınız ve yayıncının her zaman aynı kuyruk yöneticisini kullandığından emin olun. Bunu yapmazsanız, aynı konu için farklı kuyruk yöneticilerindeki farklı tutulan yayınlar etkin olabilir ve beklenmeyen davranışlara yol açabilir. Birden çok yetkili sunucu aboneliği dağıtıldığı için, alıkonan birden çok yayın alınabilir.

Eski yayınları kullanan aboneler için hala ilgileniyorsanız, her bir alıkonan yayını oluşturduğunuzda bir ileti süre bitimi ayarlamayı düşünün.

Alıkonan bir yayını bir yayınlama/abone olma kümesinden kaldırmak için **CLEAR TOPICSTR** komutunu kullanabilirsiniz. In certain circumstances you might need to issue the command on multiple members of the publish/subscribe cluster, as described in **CLEAR TOPICSTR**.

## Genel arama abonelikleri ve alıkonan yayınlar

Genel arama karakteri abonelikleri kullanıyorsanız, yayınlama/abone olma kümesinin diğer üyelerine teslim edilen karşılık gelen yetkili abonelik abonelikleri, ilk genel arama karakterinden hemen önce konu ayırıcısından genel arama karakteri olarak gönderilir. Bkz. [Vahşi kartlar ve küme konuları](#).

Bu nedenle, kullanılan genel arama karakteri, abone olan uygulamayla eşleşen daha fazla konu dizgileriyle ve daha fazla tutulan yayınlarla eşleşebilir.

Bu, alıkonan yayınlar için gereken depolama miktarını artırır ve bu nedenle, barındırma kuyruğu yöneticilerinin yeterli depolama kapasitesine sahip olduğundan emin olmanız gerekir.

### İlgili kavramlar

[Alıkonan yayınlar](#)

[Her yerde tek tek yetkili sunucu aboneliği iletme ve yayınlama](#)

## Yayınlama/abone olma kümeleri için Kümeyi Yenile

Issuing the **REFRESH CLUSTER** command results in the queue manager temporarily discarding locally held information about a cluster, including any cluster topics and their associated proxy subscriptions.

The time taken from issuing the **REFRESH CLUSTER** command to the point that the queue manager regains a full knowledge of the necessary information for clustered publish/subscribe depends on the size of the cluster, the availability, and the responsiveness of the full repository queue managers.

Yenileme işlemi sırasında, yayınlama/abone olma kümesinde yayınlama/abone olma trafiğindeki kesinti gerçekleşir. Büyük kümeler için, **REFRESH CLUSTER** komutunun kullanımı devam ederken kümeyi bozabilir ve daha sonra, küme nesnelere tüm ilgili kuyruk yöneticilerine otomatik olarak durum güncellemeleri gönderdiğinde bundan sonra 27 gün aralıklarla küme bozabilir. Bkz. [Büyük bir kümede yenilenme, kümenin performansını ve kullanılabilirliğini etkileyebilir](#). Bu nedenlerle, **REFRESH CLUSTER** komutunun yalnızca IBM Destek Merkezi 'nin rehberliğinde bir yayınlama/abone olma kümesinde kullanılması gerekir.

Kümede kesinti, aşağıdaki belirtiler olarak dışarıdan görülebilir:

- Bu kuyruk yöneticisinde küme konularına ilişkin abonelikler, kümedeki diğer kuyruk yöneticilerine bağlı yayıncılardan gelen yayınları almamakta.
- Bu kuyruk yöneticisindeki küme konularına yayınlanan iletiler, diğer kuyruk yöneticilerindeki aboneliklere geçirilmiyor.
- Bu dönem içinde yaratılan bu kuyruk yöneticisinde bulunan küme konularına ilişkin abonelikler, kümenin diğer üyelerine tutarlı bir şekilde yetkili sunucu aboneliği göndermiyor.
- Bu süre içinde silinen bu kuyruk yöneticisinde bulunan küme konularına ilişkin abonelikler, yedek sunucu aboneliklerini kümenin diğer üyelerinden tutarlı bir şekilde kaldırılmaz.
- İleti tesliminde 10 saniyelik duraksamalar ya da daha uzun süre.
- **MQRPUT** hataları; örneğin, **MQR\_C\_PUBLICATION\_FAILURE**.
- Publications placed on the dead-letter queue with a reason of **MQR\_UNKNOWN\_REMOTE\_Q\_MGR**

For these reasons publish/subscribe applications need to be quiesced before issuing the **REFRESH CLUSTER** command.

Yayınlama/abone olma kümesindeki bir kuyruk yöneticisinde bir **REFRESH CLUSTER** komutu verildikten sonra, tüm küme kuyruğu yöneticileri ve küme konuları başarıyla yenileninceye kadar bekleyin, daha sonra yetkili sunucu aboneliklerini [Yetkili sunucu aboneliklerinin yeniden eşzamanlanması](#) içinde açıklandığı biçimde yeniden eşitleyin. Tüm yetkili abonelikler doğru şekilde yeniden eşitlenmiş olduğunda, yayınlama/abone olma uygulamalarınızı yeniden başlatın.



If a **REFRESH CLUSTER** command is taking a long time to complete, monitor it by looking at the CURDEPTH of SYSTEM . CLUSTER . COMMAND . QUEUE.

### İlgili kavramlar

“Kümeleme: REFRESH CLUSTER en iyi uygulamaları kullanma” sayfa 67

Bir kümeye ilişkin olarak yerel olarak tutulan tüm bilgileri atmak ve bu bilgileri kümedeki tüm havuzlardan yeniden oluşturmak için **REFRESH CLUSTER** komutunu kullanıyorsunuz. Bu komutu, istisnai koşullar dışında, kullanmanıza gerek yoktur. Bunu kullanmanız gerekiyorsa, bunu nasıl kullanmanız gerektiği konusunda dikkat edilmesi gereken noktalar vardır. Bu bilgiler, müşterilerden gelen test ve geri bildirimlere dayalı bir kılavuzdur.

REFRESH CLUSTER çalıştırılırken uygulama sorunları görüldü

### İlgili başvurular

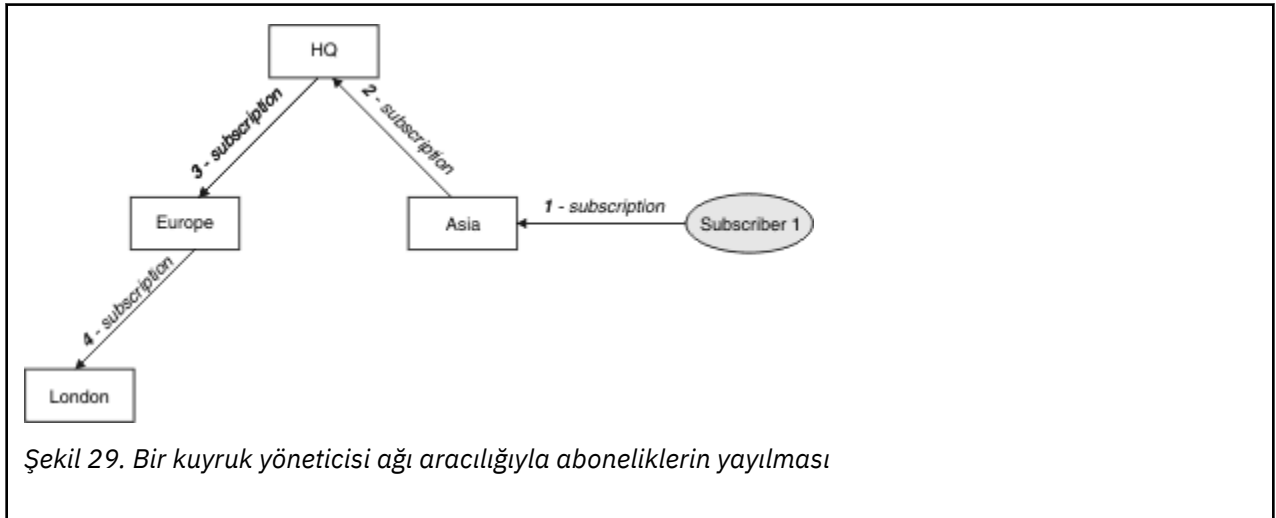
MQSC Commands başvurusu: REFRESH CLUSTER

## Yayınlama/abone olma sıradüzenlerinde yönlendirme

Dağıtımli kuyruk yöneticisi topolojiniz bir yayınlama/abone olma sıradüzeniyse ve bir kuyruk yöneticisinde bir abonelik yapıldıysa, varsayılan olarak, sıradüzendeki her kuyruk yöneticisinde bir yetkili sunucu aboneliği yaratılır. Herhangi bir kuyruk yöneticisine alınan yayınlar, eşleşen bir abonelik barındıran her kuyruk yöneticisine sıradüzenden yönlendirilir.

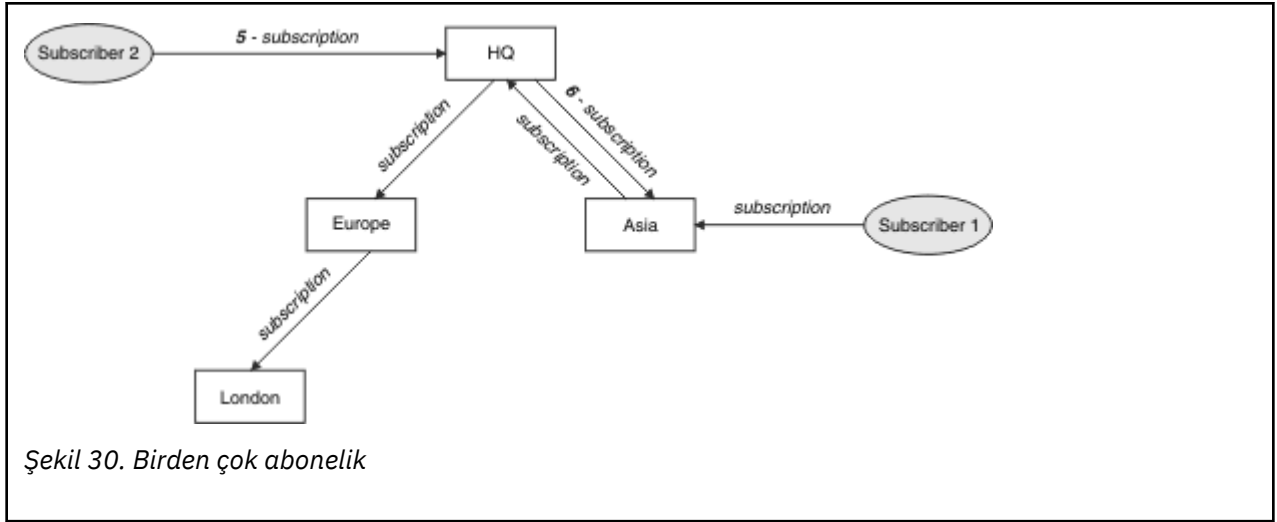
İletilerin yayınlama/abone olma sıradüzenleri ve kümelerinde kuyruk yöneticileri arasında nasıl yönlendirileceği bir giriş için bkz. Dağıtılmış yayınlama/abone olma ağları.

Bir konuya ilişkin abonelik, dağıtımli yayınlama/abone olma sıradüzeninde bir kuyruk yöneticisinde yapıldığında, kuyruk yöneticisi, aboneliğin bağlı kuyruk yöneticilerine dağıtıldığı işlemi yönetir. *Yetkili abonelikler* ağındaki tüm kuyruk yöneticilerine akış. Yetkili sunucu aboneliği, bir kuyruk yöneticisine, bu konuya ilişkin abonelikleri barındıran kuyruk yöneticilerine bir yayını iletmesi için gereken bilgileri verir. Yayınlama/abone olma sıradüzenindeki her kuyruk yöneticisi, yalnızca doğrudan ilişkilerinden haberdar olur. Bir kuyruk yöneticisine gönderilen yayınlar, doğrudan ilişkileri aracılığıyla, abonelikleri olan kuyruk yöneticilerine gönderilir. Bu, *Abone 1* ' un *Asya* kuyruk yöneticisinde (1) belirli bir konuya ilişkin bir aboneliği kaydettirdiği aşağıdaki şekilde gösterilmektedir. *Asya* kuyruk yöneticisinde bu abonelik için yetkili sunucu abonelikleri ağdaki diğer tüm kuyruk yöneticilerine iletilir (2,3, 4).

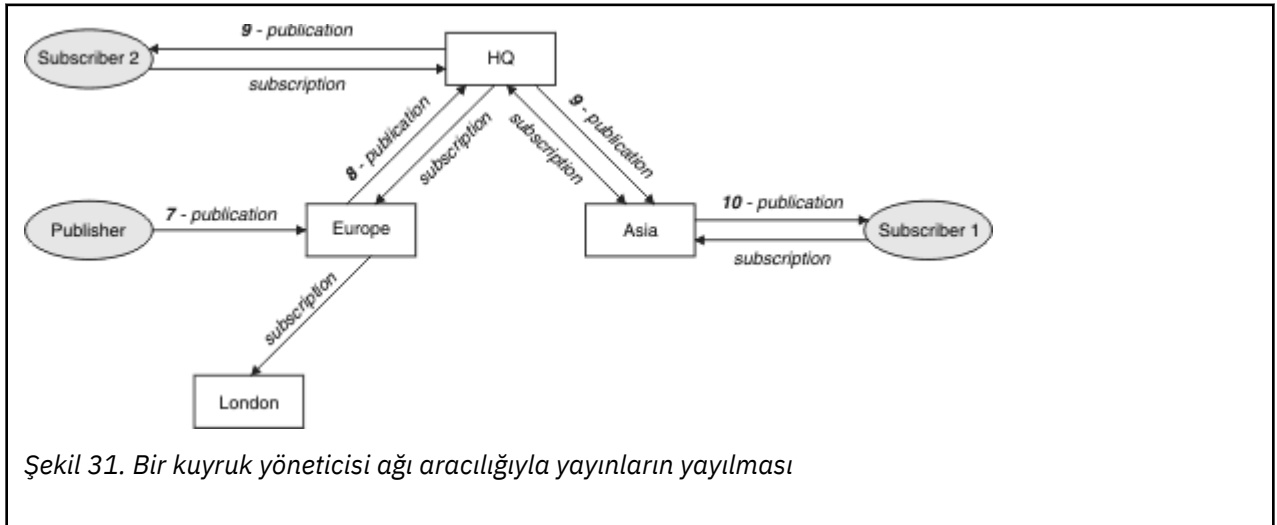


Şekil 29. Bir kuyruk yöneticisi ağı aracılığıyla aboneliklerin yayılması

Kuyruk yöneticisi, yerel uygulamalardan ya da uzak kuyruk yöneticilerinden gelen tüm abonelikleri birleştiriyor. Bir yetkili abonelik zaten var olmadığı sürece, abonelerin konularının komşuları ile ilgili konuları için yetkili abonelikler yaratır. Bu, *Abone 2* ' un bir aboneliği kaydettirdiği aşağıdaki şekilde, *HQ* kuyruk yöneticisinde (5) Şekil 29 sayfa 119 ile aynı konuya kayıt yaptır. The subscription for this topic is forwarded to the *Asya* queue manager, so that it is aware that subscriptions exist elsewhere on the network (6). Bu konuya ilişkin bir abonelik zaten kayıtlı olduğundan, abonelik *Avrupa* kuyruk yöneticisine iletilmedi; Şekil 29 sayfa 119 içindeki 3. adımı bakın.



Bir uygulama bir konuya ilişkin bilgileri yayınlarken, alıcı kuyruk yöneticisi varsayılan olarak konuyu, konuya ilişkin geçerli abonelikleri olan tüm kuyruk yöneticilerine iletir. Bir ya da daha çok ara kuyruk yöneticisi aracılığıyla bunu iletebilirler. Bu, bir yayıncının Şekil 30 sayfa 120' de olduğu gibi aynı konuya ilişkin bir yayını *Avrupa* kuyruk yöneticisine (7) gönderdiği aşağıdaki şekilde gösterilmektedir. Bu konuya ilişkin bir abonelik *HQ* ile *Avrupa* arasında bulunur; bu nedenle yayın, *HQ* kuyruk yöneticisine (8) iletilir. Ancak, *Londra* 'dan *Avrupa* ' a (yalnızca *Avrupa* 'tan *Londra* ' a) abonelik yok, bu nedenle yayın, *Londra* kuyruk yöneticisine iletilmiyor. *HQ* kuyruk yöneticisi yayını doğrudan *Abone 2* ' ye ve *Asya* kuyruk yöneticisine (9) gönderir. Yayın, *Asya* 'dan (10) *Abone 1* ' e iletilir.



Bir kuyruk yöneticisi başka bir kuyruk yöneticisine herhangi bir yayın ya da abonelik gönderdiğinde, ileti içinde kendi kullanıcı kimliğini ayarlar. Bir yayınlama/abone olma sıradüzeni kullanıyorsanız ve gelen kanal, iletilerde kullanıcı kimliği yetkisi olan iletileri koymak için ayarlandıysa, gönderen kuyruk yöneticisinin kullanıcı kimliğini yetkilendirmeniz gerekir. Bkz. [Kuyruk yöneticisi sıradüzeniyle varsayılan kullanıcı kimliklerini kullanma](#).

**Not:** Bunun yerine yayınlama/abone olma kümeleri kullanmak için, yetkilendirme küme tarafından işlenir.

## Özet ve ek konular

Bir yayınlama/abone olma sıradüzeni, kuyruk yöneticileri arasındaki ilişki üzerinde size kesin denetim sağlar. Oluşturulduktan sonra, yönetmek için küçük el müdahaleye gerek vardır. Bununla birlikte, sisteminizde bazı kısıtlamalar da ortaya çıktı:



- Hiyerarşide, özellikle de kök düğümdeki üst düğümler, güçlü, yüksek kullanılabilirlikli ve performant donatıda barındırılmalıdır. Bunun nedeni, bu düğümlerden daha fazla yayın trafiğinin akması beklenmektedir.
- Hiyerarşi içindeki her yaprak olmayan kuyruk yöneticisinin kullanılabilirliği, ağıncılardan gelen iletileri diğer kuyruk yöneticilerindeki abonelere akış yeteneğini etkiler.
- Varsayılan olarak, abone olunan tüm konu dizgileri sıradüzeni boyunca yayılır ve yayınlar yalnızca ilişkili konuya ilişkin aboneliği olan uzak kuyruk yöneticilerine dağıtılır. Dolayısıyla, abonelikler kümesinde yapılan hızlı değişiklikler sınırlayıcı bir faktör haline gelebilir. Bu varsayılan davranışı değiştirebilir ve bunun yerine tüm yayınlar, yetkili sunucu aboneliklerine olan gereksinimi ortadan kaldıran tüm kuyruk yöneticilerine yayılır. Bkz. [Yayınlama/abone olma ağlarındaki abonelik performansı](#).

**Not:** Benzer bir kısıtlama doğrudan yönlendirilen kümeler için de geçerlidir.

- Yayınlama/abone olma kuyruğu yöneticilerinin birbiriyle bağlantılı niteliği nedeniyle, yetkili aboneliklerin ağdaki tüm düğümlere yayılması zaman alır. Uzak yayınların hemen hemen abone olunması gerekmez, bu nedenle yeni bir konu dizgisine aboneliğin ardından yayınların başında yayınlanmayabilir. Tüm yayınların tüm kuyruk yöneticilerine yayılmasını sağlamak için, abonelik gecikmesinin neden olduğu sorunları kaldırabilirsiniz. Bu durumda, yetkili aboneliklere ilişkin gereksinme ortadan kalkar. Bkz. [Yayınlama/abone olma ağlarındaki abonelik performansı](#).

**Not:** Bu kısıtlama doğrudan yönlendirilen kümeler için de geçerlidir.

- Bir yayınlama/abone olma sıradüzeni için, kuyruk yöneticilerinin eklenmesi ya da kaldırılması, hiyerarşiye el ile yapılandırma gerektirir; bu sırada, kuyruk yöneticilerinin konumu ve diğer kuyruk yöneticilerine olan güvenlikleri dikkate alınmak üzere, bu sıradüzenin hiyerarşiye dikkat edilmesi gerekir. Sıradüzenin alt kısmında bulunan kuyruk yöneticilerini eklemeyerseniz ya da kaldırmıyorsanız ve bu nedenle bunların altında başka dallar yoksa, sıradüzendeki diğer kuyruk yöneticilerini de yapılandırmanız gerekir.

Yönlendirme düzeneğiniz olarak bir yayınlama/abone olma sıradüzenini kullanmadan önce, [“Yayınlama/abone olma kümelerinde doğrudan yönlendirme” sayfa 90](#) ve [“Yayınlama/abone olma kümelerinde konu anasistem yönlendirmesi” sayfa 95](#) da ayrıntılı olarak açıklanan diğer yaklaşımları keşfedin.

## Dağıtılmış yayınlama/abone olma sistem kuyrukları

Yayınlama/abone olma ileti sistemi için kuyruk yöneticileri tarafından dört sistem kuyruğu kullanılır. Yalnızca sorun belirleme ve kapasite planlama amaçlarıyla bunların varlığından haberdar olmak gerekir.

Bu kuyrukların nasıl izleneceğine ilişkin yönergeler için [Yayınlama/abone olma ağlarında üreticilerin ve tüketicilerin dengelenmesi](#) başlıklı konuya bakın.

<i>Çizelge 6. Çoklu platformlarda sistem kuyruklarını yayınlama/abone ol</i>	
<b>Sistem kuyruğu</b>	<b>Amaç</b>
SYSTEM.INTER.QMGR.CONTROL	IBM MQ dağıtılmış yayınlama/abone olma denetim kuyruğu
SYSTEM.INTER.QMGR.FANREQ	IBM MQ dağıtılmış yayınlama/abone olma iç yetkili sunucu aboneliği çıkış işlemi giriş kuyruğu
SYSTEM.INTER.QMGR.PUBS	IBM MQ dağıtılmış yayınlama/abone olma yayınları
SYSTEM.HIERARCHY.STATE	IBM MQ dağıtılmış yayınlama/abone olma sıradüzeni ilişkisi durumu

► **z/OS** z/OS sistemlerinde, kuyruk yöneticisini yaratırken gereken sistem nesnelerini, CSQINP2 kullanıma hazırlama giriş verileri kümesine CSQ4INSX, CSQ4INSR ve CSQ4INSG örneklerini ekleyerek ayarlarsanız. Daha fazla bilgi için bkz. [Görev 13: Kullanıma hazırlama giriş veri kümelerinin uyarlanması](#).

Yayınlama/abone olma sistem kuyruklarının öznelikleri [Çizelge 7 sayfa 122](#) içinde gösterilir.

Çizelge 7. Yayınlama/abone olma sistem kuyruklarının öznitelikleri	
Öznitelik	Varsayılan değer
DEFPSIST	Evet
DEFSOPT	PAYLAŞILAN
MAXMSGL	<div style="background-color: #d9ead3; padding: 2px;">Multi</div> Multiplatformsüzerinde: ALTER QMGR komutunun MAXMSGL parametresinin değeri <div style="background-color: #d9534f; color: white; padding: 2px;">z/OS</div> z/OSüzerinde: 4194304 (yani, 4 MB)
DERINLIKS	999999999
PAYLAŞ	Yok
<div style="background-color: #d9534f; color: white; padding: 2px;">z/OS</div> <div style="background-color: #d9534f; color: white; padding: 2px;">z/OS</div> STGCLASS	Bu öznitelik yalnızca z/OS platformlarında kullanılır

**Not:** Uygulamalar tarafından konan iletileri içeren tek kuyruk SYSTEM . INTER . QMGR . PUBS. **MAXDEPTH** , kesintiler ya da aşırı yükleme süreleri sırasında yayınlanan iletilerin geçici olarak oluşturulmasına izin vermek için bu kuyruk için maksimum değerine ayarlanır. Kuyruk yöneticisi, o kuyruk derinliğinin bulunmadığı bir sistemde çalışıyorsa, bu ayarlanmalıdır.

### İlgili görevler

[Dağıtılmış yayınlama/abone olma sorun giderme](#)

### **Dağıtılmış yayınlama/abone olma sistem kuyruğu hataları**

Dağıtılmış yayınlama/abone olma kuyruk yöneticisi kuyrukları kullanılmadığında hatalar oluşabilir. Bu, yayınlama/abone olma ağı genelinde abonelik bilgisi yayılmasını etkiler ve uzak kuyruk yöneticilerindeki aboneliklere yayınlanmasını sağlar.

If the fan-out request queue SYSTEM . INTER . QMGR . FANREQ is unavailable, the creation of a subscription might generate an error, and error messages will be written to the queue manager error log when proxy subscriptions need to be delivered to directly connected queue managers.

Sıradüzen ilişkisi durum kuyruğu SYSTEM . HIERARCHY . STATE kullanılmıyorsa, kuyruk yöneticisi hata günlüğüne bir hata iletileri yazılır ve yayınlama/abone olma motoru COMPAT kipine geçilir. Yayınlama/abone olma kipini görüntülemek için DISPLAY QMGR PSMODEkomutunu kullanın.

SYSTEM . INTER . QMGR kuyruklarından herhangi biri kullanılmıyorsa, kuyruk yöneticisi hata günlüğüne bir hata iletileri yazılır ve işlem geçersiz kılınmasa da, yayınlama/abone olma iletilerinin bu ya da uzak kuyruk yöneticilerindeki kuyruklarda oluşturulacağı olasılığı vardır.

Yayınlama/abone olma sistem kuyruğu ya da gerekli iletiler kuyruğu bir üst, alt ya da yayınlama/abone olma kümesi kuyruk yöneticisine uygun değilse, aşağıdaki sonuçlar oluşur:

- Yayınlar teslim edilmez ve yayınlama uygulaması bir hata alabilir. Yayınlama uygulamasının bir hata aldığı zaman ayrıntıları için, **DEFINE TOPIC** komutunun şu değiştiricilerine bakın: **PMSGDLV** , **NPMMSGDLV** ve **USEDLQ** .
- Alınan kuyruk yöneticisi yayınları, giriş kuyruğuna yedeklenir ve daha sonra yeniden denir. Geriletme eşliğine ulaşırsa, teslim edilmemiş yayınlar gönderilen mektup kuyruğuna yerleştirilir. Kuyruk yöneticisi hata günlüğünde sorunun ayrıntıları yer alır.
- Teslim edilmeyen bir yetkili abonelik, sürekli olarak gönderilen istek kuyruğuna yedeklenir ve daha sonra yeniden denir. Geriletme eşliğine ulaşırsa, teslim edilmeyen yetkili sunucu aboneliği bağlı herhangi bir kuyruk yöneticisine teslim edilmez ve gönderilen mektup kuyruğuna yerleştirilir. Kuyruk yöneticisi hata günlüğünde, gerekli düzeltme denetimi işlemine ilişkin ayrıntılar da içinde olmak üzere, sorunun ayrıntıları yer alır.

- Sıradüzen ilişkisi iletişim kuralı iletileri başarısız olur ve bağlantı durumu ERROR olarak işaretlenir. Bağlantı durumunu görüntülemek için **DISPLAY PUBSUB** komutunu kullanın.

### İlgili görevler

[Dağıtımli yayınlama/abone olma sorun giderme](#)

## Multi Çoklu Platformlar üzerinde depolama ve performans gereksinimlerinin planlanması

IBM MQ sisteminiz için gerçekçi ve ulaşılabilir depolama ve performans hedefleri ayarlamanız gerekir. Altyapınızdaki depolama ve performansı etkileyen etmenler hakkında bilgi almak için bağlantıları kullanın.

Gereksinimler, IBM MQ üzerinde kullandığınız sistemlere ve hangi bileşenlere kullanmak istediğinizde değişir.

Desteklenen donanım ve yazılım ortamlarıyla ilgili en son bilgiler için bkz. [IBM MQ](#).

IBM MQ , kuyruk yöneticisi verilerini dosya sistemindeki depolar. IBM MQ ile kullanmak üzere dizin yapılarını planlama ve yapılandırma hakkında bilgi edinmek için aşağıdaki bağlantıları kullanın:

- [“Çoklu Platformlar üzerinde Dosya Sistemi Desteği Planlanması” sayfa 127](#)
- [“Çoklu Platformlar üzerinde Paylaşılan Dosya Sistemleri Gereksinimleri” sayfa 128](#)
- [“Sharing IBM MQ files on Multiplatforms” sayfa 137](#)
- [Linux > UNIX “UNIX and Linux sistemlerindeki dizin yapısı” sayfa 139](#)
- [Windows “Windows sistemlerindeki dizin yapısı” sayfa 148](#)
- [IBM i “IBM üzerindeki dizin yapısı” sayfa 152](#)

[Linux > UNIX](#) Use the following links for information about system resources, shared memory, and process priority on UNIX and Linux:

- [Linux > UNIX “IBM MQ ve UNIX System V IPC kaynakları” sayfa 156](#)
- [AIX “AIX üzerinde paylaşılan bellek” sayfa 155](#)
- [Linux > UNIX “IBM MQ ve UNIX Process Priority” sayfa 156](#)

Günlük dosyalarıyla ilgili bilgi için aşağıdaki bağlantıları kullanın:

- [“Çoklu platformlar üzerinde dairesel ya da doğrusal oturum açma seçilmesi” sayfa 155](#)
- [Günlüğün büyüklüğünün hesaplanması](#)

### İlgili kavramlar

[“z/OS üzerinde IBM MQ ortamınızın planlanması” sayfa 156](#)

IBM MQ ortamınızı planlarken, veri kümeleri, sayfa kümeleri, Db2, Coupling Facility'lerinin ve yedekleme olanaklarının ve yedekleme olanaklarının kaynak gereksinimlerini göz önünde bulundurmanız gerekir. Use this topic to plan the environment where IBM MQ runs.

### İlgili görevler

[“IBM MQ mimarisinin planlanması” sayfa 5](#)

IBM MQ ortamınızı planlarken, IBM MQ ' un tek ve birden çok kuyruk yöneticisi mimarisi ve noktadan noktaya iletişim ve yayınlama/abone olma ileti sistemi stilleri için sağladığı desteği göz önünde bulundurun. Ayrıca, kaynak gereksinimlerinizi ve günlük kaydı ve yedekleme olanaklarını kullanımınızı da planlayın.

### İlgili başvurular

[UNIX and Linux üzerindeki donanım ve yazılım gereksinimleri](#)

[Windows üzerindeki donanım ve yazılım gereksinimleri](#)

## Çoklu Platformlar üzerindeki Disk Alanı Gereksinimleri

IBM MQ için depolama gereksinimleri, hangi bileşenlere kurduğunuz ve ne kadar çalışma alanı gereksiniminiz olduğunu bağıdır.

Kurulmasını istediğiniz önkoşul bileşenleri de içinde olmak üzere, kurmak üzere seçtiğiniz isteğe bağlı bileşenler için disk saklama alanı gereklidir. Toplam depolama gereksinimi, kullandığınız kuyruk sayısına, kuyruklardaki iletilerin sayısını ve boyutunu ve iletilerin kalıcı olup olmadığını da bağıdır. Ayrıca, kendi uygulama programlarınız için, disk, manyetik bant ya da diğer ortamlardaki arşivleme kapasitesinin yanı sıra, alan için de yer alan bir alan olmasını da gerektirir.

Aşağıdaki çizelgelerde, ürünün çeşitli birleşimlerini farklı platformlarda kurduğunuzda gereken yaklaşık disk alanı gösterilir. (Değerler en yakın 5 MB 'ye kadar yuvarlanır; burada bir MB 1.048.576 bayttır.)

- ▶ **LTS** “Long Term Support için disk alanı gereksinimleri” sayfa 124
- ▶ **CD** “Continuous Delivery için disk alanı gereksinimleri” sayfa 125

### Long Term Support için disk alanı gereksinimleri

LTS

Çizelge 8. IBM MQ disk space requirements for Multiplatforms for Long Term Support			
Platform	İstemci kuruluşu <sup>1</sup>	Sunucu kuruluşu <sup>2</sup>	Tam kuruluş <sup>3</sup>
▶ <b>AIX</b> AIX	150 MB	205 MB	1410 MB
▶ <b>IBM i</b> IBM i (bkz. IBM i Additional için ek notlar)	460 MB	805 MB	1815 MB
▶ <b>Linux</b> Linux for x86-64	130 MB	185 MB	1645 MB
▶ <b>Linux</b> Linux on POWER Systems - Little Endian	95 MB	135 MB	1100 MB
▶ <b>Linux</b> Linux for IBM Z	130 MB	175 MB	1130 MB
▶ <b>Solaris</b> Solaris x86-64, AMD64, EM64T ve uyumlu işlemciler	110 MB <sup>4</sup>	160 MB <sup>4</sup>	1185 MB
▶ <b>Solaris</b> Solaris SPARC	100 MB <sup>4</sup>	150 MB <sup>4</sup>	1160 MB
▶ <b>Windows</b> Windows (64 bit kuruluş) <sup>5</sup>	230 MB	365 MB	1565 MB

#### Notlar:

- İstemci kuruluşu aşağıdaki bileşenleri içerir:
  - Yürütme Ortamı
  - İstemci
- Sunucu kuruluşu aşağıdaki bileşenleri içerir:

- Yürütme Ortamı
  - Sunucu
3. Tam kuruluş, kullanılabilir tüm bileşenleri içerir.
  4. **Solaris** Solaris platformlarında, bu bileşen bileşimini almak için sessiz bir şekilde kurmanız gerekir.
  5. **Windows** Burada listelenen bileşenlerin tümü Windows sistemlerindeki kurulabilir özelliklerdir; işlevleri bazen diğer özelliklerde de yer alır. Bkz. [Windows sistemleri için IBM MQ özellikleri](#).

#### IBM için ek notlar: **IBM i**

1. IBM i üzerinde yerel istemciyi sunucudan ayıramazsınız. Çizelgedeki sunucu şekli, Java olmadan 5724H72\*BASE İngilizce Dil Yükünün (2924) ile birlikte kullanılmasıdır. 22 olası benzersiz dil yükü vardır.
2. Çizelgedeki şekil, yerel istemci 5725A49 \*BASE ' nin Java olmadan yapılması içindir.
3. Java ve JMS sınıfları hem sunucuya, hem de istemci bağ tanımlarına eklenebilir. Bu özelliklerin içermesini istiyorsanız, 110 MB ' lik bir değer ekleyin.
4. Örnek kaynağın istemciye ya da sunucuya eklenmesi fazladan 10 MB ekler.
5. Adding samples to Java and JMS classes adds an extra 5 MB.

#### Continuous Delivery için disk alanı gereksinimleri

CD

Çizelge 9. IBM MQ disk space requirements for Multiplatforms for Continuous Delivery			
Platform/CD yayın düzeyi	İstemci kuruluşu <sup>1</sup>	Sunucu kuruluşu <sup>2</sup>	Tam kuruluş <sup>3</sup>
<b>AIX</b>			
<b>V 9.1.1</b> IBM MQ 9.1.1	270 MB	400 MB	1725 MB
<b>V 9.1.2</b> IBM MQ 9.1.2	275 MB	385 MB	1745 MB
<b>V 9.1.3</b> IBM MQ 9.1.3	150 MB	195 MB	1520 MB
<b>V 9.1.4</b> IBM MQ 9.1.4	150 MB	195 MB	1520 MB
<b>V 9.1.5</b> IBM MQ 9.1.5	325 MB	370 MB	1690 MB
<b>Linux x86-64 için Linux (64 bit)</b>			
<b>V 9.1.1</b> IBM MQ 9.1.1	130 MB	185 MB	1725 MB
<b>V 9.1.2</b> IBM MQ 9.1.2	140 MB	185 MB	1720 MB
<b>V 9.1.3</b> IBM MQ 9.1.3	140 MB	185 MB	1725 MB

Çizelge 9. IBM MQ disk space requirements for Multiplatforms for Continuous Delivery (devamı var)

Platform/CD yayın düzeyi	İstemci kuruluđu 1	Sunucu kuruluđu 2	Tam kuruluđu 3
> V 9.1.4 IBM MQ 9.1.4	140 MB	185 MB	1730 MB
> V 9.1.5 IBM MQ 9.1.5	155 MB	200 MB	1740 MB
<b>Linux Linux on POWER Systems - Little Endian</b>			
> V 9.1.1 IBM MQ 9.1.1	95 MB	135 MB	1170 MB
> V 9.1.2 IBM MQ 9.1.2	95 MB	135 MB	1160 MB
> V 9.1.3 IBM MQ 9.1.3	95 MB	135 MB	1165 MB
> V 9.1.4 IBM MQ 9.1.4	95 MB	135 MB	1195 MB
> V 9.1.5 IBM MQ 9.1.5	150 MB	170 MB	1205 MB
<b>Linux IBM Z için Linux</b>			
> V 9.1.1 IBM MQ 9.1.1	130 MB	175 MB	1190 MB
> V 9.1.2 IBM MQ 9.1.2	130 MB	175 MB	1190 MB
> V 9.1.3 IBM MQ 9.1.3	130 MB	175 MB	1190 MB
> V 9.1.4 IBM MQ 9.1.4	130 MB	175 MB	1190 MB
> V 9.1.5 IBM MQ 9.1.5	235 MB	260 MB	1230 MB
<b>Windows Windows (64 bit kuruluđu) 4</b>			
> V 9.1.1 IBM MQ 9.1.1	245 MB	375 MB	1727 MB
> V 9.1.2 IBM MQ 9.1.2	260 MB	370 MB	1727 MB
> V 9.1.3 IBM MQ 9.1.3	260 MB	370 MB	1727 MB

Çizelge 9. IBM MQ disk space requirements for Multiplatforms for Continuous Delivery (devamı var)

Platform/CD yayın düzeyi	İstemci kuruluđu 1	Sunucu kuruluđu 2	Tam kuruluđu 3
V 9.1.4 IBM MQ 9.1.4	265 MB	380 MB	1745 MB
V 9.1.5 IBM MQ 9.1.5	265 MB	380 MB	1745 MB

#### Notlar:

- İstemci kuruluđu aşağıdaki bileşenleri içerir:
  - Yürütme Ortamı
  - İstemci
- Sunucu kuruluđu aşağıdaki bileşenleri içerir:
  - Yürütme Ortamı
  - Sunucu
- Tam kuruluđu, kullanılabilir tüm bileşenleri içerir.
- Windows** Burada listelenen bileşenlerin tümü Windows sistemlerindeki kurulabilir özelliklerdir; işlevleri bazen diğer özelliklerde de yer alır. Bkz. [Windows sistemleri için IBM MQ özellikleri](#).

#### İlgili kavramlar

[IBM MQ bileşenleri ve özellikleri](#)

Multi

## Çoklu Platformlar üzerinde Dosya Sistemi Desteği Planlanması

Kuyruk yöneticisi verileri dosya sisteminde depolanır. Kuyruk yöneticisi, çok eşgörünömlü bir kuyruk yöneticisinin birden çok örneğinin aynı anda etkin olmasını önlemek için dosya sistemi kilitlemesini kullanır.

### Paylaşılan kütük sistemleri

Paylaşılan kütük sistemleri, birden çok sistemin aynı anda aynı fiziksel depolama aygıtına erişmesine olanak sağlar. Birden çok sistem aynı fiziksel depolama aygıtına doğrudan kilitleme ve eşzamanlılık denetimi araçları olmaksızın doğrudan erişirse, bozulma meydana gelir. İşletim sistemleri, yerel süreçler için kilitleme ve eşzamanlılık denetimi ile yerel dosya sistemleri sağlar; ağ dosya sistemleri dağıtılmış sistemler için kilitleme ve eşzamanlılık denetimi sağlar.

Geçmişteki ağ dosya sistemleri, iletileri günlüğe kaydetme gereksinimlerini karşılamak için yeterince hızlı bir şekilde gerçekleştirilmemiş veya yeterli kilitleme ve eşzamanlılık denetimi sağlamış olmalıdır. Günümüzde, ağ üzerinde çalışan dosya sistemleri iyi performans sağlayabilir ve *RFC 3530, Network File System (NFS) sürüm 4 iletişim kuralı* gibi güvenilir ağ dosya sistemi iletişim kuralları uygulamaları, iletileri güvenilir bir şekilde günlüğe kaydetmeye ilişkin gereksinimleri karşılayabilir.

### Paylaşılan kütük sistemleri ve IBM MQ

Çok eşgörünömlü bir kuyruk yöneticisine ilişkin kuyruk yöneticisi verileri paylaşılan bir ağ dosyası sisteminde saklanır. UNIX, Linux, and Windows sistemlerinde, kuyruk yöneticisinin veri dosyaları ve günlük dosyaları paylaşılan ağ dosya sistemine yerleştirilmelidir. **IBM i** IBM üzerinde günlük dosyaları yerine günlükler kullanılır ve günlükler paylaşamaz. Multi-instance queue managers on IBM i use journal replication, or switchable journals, to make journals available between different queue manager instances.

IBM MQ , aynı anda aynı çok eşgörünümli kuyruk yöneticisinin birden çok örneğinin aynı anda etkin olmasını önlemek için kilitlemeyi kullanır. Aynı kilitleme, iki ayrı kuyruk yöneticisinin yanlışlıkla aynı kuyruk yöneticisi veri dosyaları kümesini kullanmasını da sağlar. Bir kuyruk yöneticisinin yalnızca bir eşgörünümünün kilidine bir kerede sahip olabilir. Sonuç olarak IBM MQ , paylaşılan bir dosya sistemi olarak erişilen ağ üzerinde depolanan depolamaya depolanan kuyruk yöneticisi verilerini destekler.

Ağ dosyası sistemlerinin tüm kilitleme protokolleri sağlam olmadığından ve bir dosya sistemi veri bütünlüğü yerine performans için yapılandırılabilirliğinden, bir ağ dosya sisteminin kuyruk yöneticisi verilerine erişimi denetleyip denetlemediğini ve doğru günlüklere erişip denetmeyeceğini test etmek için **amqmfsc** komutunu çalıştırmanız gerekir. Bu komut yalnızca UNIX, Linux ve IBM i sistemleri için geçerlidir. Windows üzerinde, desteklenen tek bir ağ dosya sistemi vardır ve **amqmfsc** komutu gerekli değildir.

### İlgili görevler

“Çoklu Platformlar üzerinde Paylaşılan Dosya Sistemi Davranış Doğruluyor” sayfa 130

UNIX **IBM i** ve IBM i sistemlerinde paylaşılan bir dosya sisteminin, çok eşgörünümli bir kuyruk yöneticisinin kuyruk yöneticisi verilerini saklamak için gereken gereksinimleri karşılayıp karşılamadığını denetlemek için **amqmfsc** komutunu çalıştırın. Run the IBM MQ MQI client sample program **amqsfhac** in parallel with **amqmfsc** to demonstrate that a queue manager maintains message integrity during a failure.

## Multi Çoklu Platformlar üzerinde Paylaşılan Dosya Sistemleri Gereksinimleri

Paylaşılan dosya sistemleri, IBM MQ ile güvenilir bir şekilde çalışmaması üzerine, veri yazma bütünlüğü, dosyalara özel erişim garantili ve kilitleme serbest bırakma olanağı sağlamalıdır.

### Paylaşılan bir dosya sisteminin karşılaması gereken gereksinimler

Paylaşılan bir dosya sisteminin IBM MQ ile güvenilir bir şekilde çalışmak için yerine getirmeleri gereken üç temel gereksinim vardır:

#### 1. Veri yazma bütünlüğü

Veri yazma bütünlüğüne bazen *Flush 'ta Diske Yazadı* verilir. Kuyruk yöneticisi, fiziksel aygıtla başarıyla kesinleştirilmekte olan verilerle eşitlenebilir. İşlemsel bir sistemde, diğer işleme devam etmeden önce bazı yazma işlemlerinin güvenli bir şekilde kesinleştirildiğine emin olmanız gerekir.

More specifically, IBM MQ on UNIX platforms uses the *O\_SYNC* open option and the *fsync()* system call to explicitly force writes to recoverable media, and the write operation is dependent upon these options operating correctly.



**Uyarı:** **Linux** Kütük sistemini *async* seçeneğiyle sisteme bağlamalısınız; bu seçenek, zaman uyumlu yazma seçenekleri seçeneğini de destekler ve senkronize seçeneğinden daha iyi başarımlar sağlar.

Ancak, dosya sistemi Linux' den dışa aktarıldıysa, dosya sistemini eşitleme seçeneğini kullanarak dışa aktarmanız gerektiğini unutmayın.

#### 2. Dosyalara özel erişim garantili

Birden çok kuyruk yöneticisini uyumlulaştırmak için, kuyruk yöneticisi için bir dosyada dışlayıcı kilit elde etmek için bir mekanizma olması gerekir.

#### 3. Hata üzerindeki kilitleme serbest bırak

Bir kuyruk yöneticisi başarısız olursa ya da dosya sistemiyle bir iletişim hatası varsa, kuyruk yöneticisinin kilitlemiş kütüklerinin kilidinin açılması ve kuyruk yöneticisinin kütük sistemine yeniden bağlanmasını beklemeden diğer işlemler tarafından kullanılabilir kılınması gerekir.

IBM MQ ' in güvenilir bir şekilde çalışması için, paylaşılan bir dosya sisteminin bu gereksinimleri karşılaması gerekir. Böyle bir işlem yoksa, çok eşgörünümli bir kuyruk yöneticisi yapılandırışındaki paylaşılan kütük sistemi kullanılırken kuyruk yöneticisi verileri ve günlükleri bozulabilir.



Microsoft Windows' ta çok eşgörünümlü kuyruk yöneticileri için ağa bağlı depolama alanına Microsoft Windows ağları tarafından kullanılan CIFS (Common Internet File System; Ortak İnternet Dosya Sistemi) iletişim kuralı tarafından erişilmelidir. The Common Internet File System (CIFS) client does not meet the IBM MQ requirements for locking semantics on platforms other than Microsoft Windows, so multi-instance queue managers running on platforms other than Microsoft Windows must not use Common Internet File System (CIFS) as their shared file system.

Desteklenen diğer platformlardaki çok eşgörünümlü kuyruk yöneticileri için depolama alanına, Posix uyumlu bir ağ dosya sistemi iletişim kuralıyla erişilmesi ve kiralama tabanlı kilitlemeyi desteklenmesi gerekir. Ağ Dosya Sistemi Sürüm 4 bu gereksinimi karşılar. Hata sonrası kilitleri serbest bırakmak için güvenilir bir mekanizmaya sahip olmayan Network File System Version 3 gibi eski dosya sistemleri çok eşgörünümlü kuyruk yöneticileriyle birlikte kullanılmamalıdır.

## Paylaşılan dosya sisteminin gereksinimleri karşılayıp karşılamadığını denetler.




Kullanmayı planladığınız paylaşılan dosya sisteminin bu gereksinimleri karşılayıp karşılamadığını denetlemeniz gerekir. Ayrıca, dosya sisteminin güvenilirlik için doğru yapılandırılıp yapılandırılmadığını da denetlemeniz gerekir. Paylaşılan dosya sistemleri bazen, güvenilirliğin giderileceği şekilde performansı artırmak için yapılandırma seçenekleri sağlar.

Daha fazla bilgi için bkz. [IBM MQ çok eşgörünümlü kuyruk yöneticisi dosya sistemlerine ilişkin sınama açıklaması](#).

Olağan koşullarda, IBM MQ öznitelik önbelleğe alma özelliği ile doğru biçimde çalışır ve önbelleğe almayı geçersiz kılmak gerekmez; örneğin, NFS montajı için NOAC ayarını geçersiz kılarak. Öznitelik önbelleğe alma işlemi, birden çok dosya sistemi istemcisi dosya sistemi sunucusunda aynı dosyaya yazma erişimi için çekişmeye devam ederken, her istemci tarafından kullanılan önbelleğe alınan öznitelikler sunucudaki özniteliklerle aynı olmayabilir. Bu şekilde erişilen dosyalara örnek olarak, çok eşgörünümlü bir kuyruk yöneticisi için kuyruk yöneticisi hata günlükleridir. Kuyruk yöneticisi hata günlükleri hem etkin hem de beklemedeki bir kuyruk yöneticisi yönetim ortamı tarafından yazılabilir ve önbelleğe alınan dosya öznitelikleri, dosyaların baştan başlama işlemi gerçekleşmeden önce hata günlüklerinin beklenenden daha büyük bir şekilde büyümesine neden olabilir.

Dosya sisteminin denetlemenize yardımcı olması için [Paylaşılan dosya sistemi davranışı doğrulanıyorgörevini](#) çalıştırın. This task checks if your shared file system meets requirements [2](#) and [3](#). You need to verify requirement [1](#) in your shared file system documentation, or by experimenting with logging data to the disk.

Disk faults can cause errors when writing to disk, which IBM MQ reports as First Failure Data Capture errors. Herhangi bir disk hatasına ilişkin paylaşılan kütük sisteminin denetlemek için işletim sisteminize ilişkin dosya sistemi denetleyicisini çalıştırabilirsiniz. Örneğin:

-   On UNIX and Linux the file system checker is called fsck.
-  Windows altyapılarında, kütük sistemi denetimcisine CHKDSK ya da SCANDISK adı verilir.

## NFS sunucusu güvenliği

### Notlar:

- IBM MQ kuruluş dizinini tutmak için kullanılan bir bağlama noktası için **nosuid** ya da **noexec** seçeneklerini kullanamazsınız. Bunun nedeni, IBM MQ ' un setuid/setgid yürütülür programlarının içerdiği ve bunların düzgün bir şekilde çalıştırılmaması gerektiğinden kaynaklanır.
- Kuyruk yöneticisi verilerini yalnızca bir Network File System (NFS) sunucusuna yerleştirdiğinizde, sistem güvenliğini sağlamak için, kuyruk yöneticisinin çalışmasında herhangi bir zarar görmeyecek şekilde, sisteme bağlama komutuyla aşağıdaki üç seçeneği kullanabilirsiniz:

#### **noexec**

Bu seçeneği kullanarak, ikili dosyaların NFSüzerinde çalıştırılmasını durdurarak, uzak kullanıcının sistemde istenmeyen kod çalıştırmasını önler.

## nosuid


Bu seçeneği kullanarak, uzak kullanıcının daha yüksek ayrıcalıklar kazanmasını önleyen, set-user-identifier ve set-group-identifier bitlerinin kullanılmasını engelliyorsunuz.

## düğüm

Bu seçeneği kullanarak, karakter ve öbek özel aygıtlarının kullanılmasını ya da tanımlanmasını engelleyerek uzak kullanıcının bir chroot hapishanesinden çıkmasını önlemenizi sağlar.

## Multi Çoklu Platformlar üzerinde Paylaşılan Dosya Sistemi Davranış

### Doğrulanıyor

UNIX  ve IBM i sistemlerinde paylaşılan bir dosya sisteminin, çok eşgörünümlü bir kuyruk yöneticisinin kuyruk yöneticisi verilerini saklamak için gereken gereksinimleri karşılayıp karşılamadığını denetlemek için **amqmfscck** komutunu çalıştırın. Run the IBM MQ MQI client sample program **amqsfhac** in parallel with **amqmfscck** to demonstrate that a queue manager maintains message integrity during a failure.

### Başlamadan önce

Ağ üzerinde çalışan depolamaya sahip bir sunucuya ve IBM MQ ' a bağlı olan diğer iki sunucuya gerek duyarsınız. Dosya sistemini yapılandırmak için yönetici (kök) yetkinizin olması ve **amqmfscck** çalıştırmak için bir IBM MQ Yöneticisi olmanız gerekir.


### Bu görev hakkında

“Çoklu Platformlar üzerinde Paylaşılan Dosya Sistemleri Gereksinimleri” sayfa 128 , çok eşgörünümlü kuyruk yöneticileriyle paylaşılan bir dosya sistemini kullanmaya ilişkin dosya sistemi gereksinimlerini açıklar. IBM MQ teknik notu [IBM MQ çok eşgörünümlü kuyruk yöneticisi kütük sistemlerine ilişkin sına](#) [deyimi](#) , IBM ' in önceden sınanmış olduğu paylaşılan kütük sistemlerini listeler. Bu görevdeki yordam, listelenmemiş bir dosya sisteminin veri bütünlüğü sağlayıp sağlamadığını değerlendirmenize yardımcı olmak için bir dosya sisteminin nasıl sınavacağına açıklar.

Çok eşgörünümlü bir kuyruk yöneticisinin yedek sisteme geçişi, kuyruk yöneticisinin verilerine ya da günlük dosyalarına yazılmasını önleyen ağ sorunları da içinde olmak üzere, donanım ya da yazılım hatalarıyla tetiklenebilir. Genellikle, dosya sunucusunda hatalara neden olmak ilginizi çekmektedir. Ancak, kilitleri sınamak için IBM MQ sunucularının başarısız olmasına, başarıyla serbest bırakılmasına da neden olmanız gerekir. Paylaşılan bir dosya sisteminde emin olmak için, aşağıdaki başarısızlıkları ve ortamınıza özgü diğer hataları test edin:

1. Disklerin eşitlenmesi de dahil olmak üzere dosya sunucusundaki işletim sistemi sona erdiriliyor.
2. Disklerin eşitlenmesi olmadan dosya sunucusunda işletim sistemi durduruluyor.
3. Sunucuların her birinde sıfırlama düğmesine basılması.
4. Ağ kablosunu sunucuların her birinden çekerek çıkarın.
5. Sunucuların her birinden güç kablosu çıkarılıyor.
6. Sunucuların her biri kapatılıyor.

Kuyruk yöneticisi verilerini ve günlüklerini paylaşmak için kullanacağınız ağ üzerinde çalışan depolamadaki dizini yaratın. Dizin sahibi, UNIX üzerinde mqm grubunun bir üyesi olan bir IBM MQ Yöneticisi olmalı ya da başka bir deyişle Administrator Yöneticisi olmalıdır. Sınamaları çalıştıran kullanıcının IBM MQ Administrator yetkisi olmalıdır.

Dosya sistemini yapılandırmanıza yardımcı olması için [Linux üzerinde çok eşgörünümlü bir kuyruk yöneticisi yarat](#)  ya da ADDMQMARN kullanan bir ASP ' deki ikizlenmiş günlük yapılandırması içindeki bir dosya sistemini dışa aktarma ve sisteme bağlama örneğini kullanın. Farklı dosya sistemleri, farklı yapılandırma adımları gerektirir. Dosya sistemi belgelerini okuyun.

## Yordam

Her denetimde, dosya sistemi denetleyicisi çalışırken önceki listedeki tüm hatalara neden olur. If you intend to run **amqsfhac** at the same time as **amqmfscck**, do the task, "[İleti bütünlüğünü sınamak için amqsfhac çalıştırılıyor](#)" sayfa 135 in parallel with this task.

1. Dışa aktarılan dizini iki IBM MQ sunucusuna bağlayın.

Dosya sistemi sunucusunda paylaşılan bir dizin ( shared ) ve çok eşgörunümlü kuyruk yöneticilerine ilişkin verileri saklamak için bir alt dizin ( qmdata ) yaratın. Linux üzerinde çok eşgörunümlü kuyruk yöneticileri için paylaşılan bir dizin ayarlama örneği için, [Linux üzerinde çok eşgörunümlü kuyruk yöneticisi yaratıçindeki Örnek](#) başlıklı konuya bakın.

2. Temel dosya sistemi davranışını denetleyin.

Bir IBM MQ sunucusunda, dosya sistemi denetleyicisini hiçbir parametre olmadan çalıştırın.

IBM MQ Server 1 'de:

```
amqmfscck /shared/qmdata
```

3. Her iki IBM MQ sunucusundan aynı dizine eşzamanlı olarak yazma işlemi olup olmadığını denetleyin. Her iki IBM MQ sunucusunda da, dosya sistemi denetleyicisini -c seçeneğiyle aynı anda çalıştırın.

IBM MQ Server 1 'de:

```
amqmfscck -c /shared/qmdata
```

IBM MQ Server 2 'de:

```
amqmfscck -c /shared/qmdata
```

4. Her iki IBM MQ sunucusunda kilit bekleyen ve serbest bırakılan kilitlerin olup olmadığını denetleyin. Her iki IBM MQ sunucusunda da, dosya sistemi denetleyicisini -w seçeneğiyle aynı anda çalıştırın.

IBM MQ Server 1 'de:

```
amqmfscck -w /shared/qmdata
```

IBM MQ Server 2 'de:

```
amqmfscck -w /shared/qmdata
```

5. Veri bütünlüğü olup olmadığını denetleyin.

- a) Test dosyasını biçimlendirin.

Sınamakta olan dizinde büyük bir dosya yaratın. Dosya, sonraki aşamaların başarıyla tamamlanabilmesi için biçimlendirilir. Dosya, hata durumunda yedek sisteme geçmenin benzetimini yapmak üzere ikinci aşamayı kesmek için yeterli süre olacak kadar büyük olmalıdır. 262144 sayfa (1 GB) varsayılan değerini deneyin. Program yavaş dosya sistemlerinde bu varsayılan değeri otomatik olarak azaltır, böylece biçimlendirme yaklaşık 60 saniye içinde tamamlanır.

IBM MQ Server 1 'de:

```
amqmfscck -f /shared/qmdata
```

Sunucu aşağıdaki iletilerle yanıt verir:

```
Formatting test file for data integrity test.
```

Test file formatted with 262144 pages of data.

b) Bir hataya neden olurken, dosya sistemi denetleyicisini kullanarak test dosyasına veri yazın.

Sınama programını aynı anda iki sunucu üzerinde çalıştırın. Hatayı deneyecek olan sunucuda test programını başlatın, ardından başarısızlığı atlatacağı sunucu üzerinde sınama programını başlatın. Çünkü soruşturmaya devam etmiyordun.

İlk sınama programı bir hata iletişiyle durur. İkinci test programı, test dosyasındaki kilidi alır ve ilk test programının bıraktığı yerden başlayarak test dosyasına veri yazar. İkinci test programının tamamlanmasına izin verin.

<i>Çizelge 10. Veri bütünlüğü denetiminin aynı anda iki sunucu üzerinde çalıştırılması</i>	
<b>IBM MQ Sunucu 1</b>	<b>IBM MQ Sunucu 2</b>
amqmfscck -a /shared/qmdata	
Please start this program on a second machine with the same parameters.  File lock acquired.  Start a second copy of this program with the same parameters on another server.  Writing data into test file.  To increase the effectiveness of the test, interrupt the writing by ending the process, temporarily breaking the network connection to the networked storage, rebooting the server or turning off the power.	amqmfscck -a /shared/qmdata  Waiting for lock...  Waiting for lock...  Waiting for lock...  Waiting for lock...  Waiting for lock...
Turn the power off here.	
	File lock acquired.  Reading test file  Checking the integrity of the data read.  Appending data into the test file after data already found.  The test file is full of data. It is ready to be inspected for data integrity.

Testin zamanlaması, dosya sisteminin davranışına bağlıdır. Örneğin, bir dosya sisteminin, bir güç kesintisini izleyen ilk program tarafından elde edilen dosya kilitlerini serbest bırakması için genellikle 30-90 saniye sürer. İlk test programı dosyayı doldurmadan önce başarısızlığı tanıtmak

için çok az zamanınız varsa, test dosyasını silmek için **amqmfscck** ' un -x seçeneğini kullanın. Daha büyük bir test dosyası ile baştan testi deneyin.

c) Test dosyasındaki verilerin bütünlüğünü doğrulayın.

IBM MQ Server 2 'de:

```
amqmfscck -i /shared/qmdata
```

Sunucu aşağıdaki iletilerle yanıt verir:

```
File lock acquired

Reading test file checking the integrity of the data read.

The data read was consistent.

The tests on the directory completed successfully.
```

6. Test dosyalarını silin.

IBM MQ Server 2 'de:

```
amqmfscck -x /shared/qmdata
Test files deleted.
```

Sunucu, şu iletiyle yanıt verir:

```
Test files deleted.
```

## Sonuçlar

Sınamalar başarıyla tamamlanırsa, sıfır dışında bir çıkış kodu döndürülür; tersi durumda sıfır dışında bir değer döndürülür.

## Örnekler

İlk üç örnek kümesi, minimum çıkış üreten komutu gösterir.

### Bir sunucuda temel dosya kilitlemeyi başarılı bir şekilde test edin

```
> amqmfscck /shared/qmdata
The tests on the directory completed successfully.
```

### Bir sunucuda temel dosya kilitleme sınaması başarısız oldu

```
> amqmfscck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck' error '2'.
```

### İki sunucu üzerinde kilitleme sınaması başarıyla tamamlandı

Çizelge 11. İki sunucu üzerinde başarılı kilitleme

IBM MQ Sunucu 1	IBM MQ Sunucu 2
<pre>&gt; amqmfscck -w /shared/qmdata Please start this program on a second machine with the same parameters. Lock acquired. Press Return or terminate the program to release the lock.</pre>	

Çizelge 11. İki sunucu üzerinde başarılı kilitleme (devamı var)

IBM MQ Sunucu 1	IBM MQ Sunucu 2
	> amqmfscck -w /shared/qmdata Waiting for lock...
[ Return pressed ] Lock released.	
	Lock acquired. The tests on the directory completed successfully

İkinci üç örnek kümesi, ayrıntılı kipi kullanarak aynı komutları gösterir.

### Bir sunucuda temel dosya kilitlemeyi başarılı bir şekilde test edin

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd1 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd1, F_SETLK, F_RDLCK)
System call: fd2 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd2, F_SETLK, F_RDLCK)
System call: close(fd2)
System call: write(fd1)
System call: close(fd1)
The tests on the directory completed successfully.
```

### Bir sunucuda temel dosya kilitleme sınaması başarısız oldu

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_RDLCK)
System call: fdSameFile = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fdSameFile, F_SETLK, F_RDLCK)
System call: close(fdSameFile)
System call: write(fd)
AMQxxxx: Error calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck', errno 2
(Permission denied).
```

## İki sunucu üzerinde kilitleme sınaması başarıyla tamamlandı

Çizelge 12. İki sunucu üzerinde başarılı kilitleme-verbose kipi

IBM MQ Sunucu 1	IBM MQ Sunucu 2
<pre>&gt; amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL   O_CREAT   O_RDWR, 0666)' Calling 'fchmod(fd, 0666)' Calling 'fstat(fd)' Please start this program on a second machine with the same parameters. Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. Press Return or terminate the program to release the lock.</pre>	
	<pre>&gt; amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL   O_CREAT   O_RDWR,0666)' Calling 'fd = open("/shared/qmdata/amqmfscck.lkw, O_RDWR, 0666)' Calling 'fcntl(fd, F_SETLK, F_WRLCK) 'Waiting for lock...</pre>
<pre>[ Return pressed ] Calling 'close(fd)' Lock released.</pre>	
	<pre>Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. The tests on the directory completed successfully</pre>

### Multi

### İleti bütünlüğünü sınamak için amqsfhac çalıştırılıyor

**amqsfhac** , ağ üzerinde çalışan depolamayı kullanan bir kuyruk yöneticisinin, bir arızmanın ardından veri bütünlüğünü koruduğunu denetler.

### Başlamadan önce

Bu sınamaya için dört sunucuya gereksinim duyarsınız. Two servers for the multi-instance queue manager, one for the file system, and one for running **amqsfhac** as an IBM MQ MQI client application.

Çok eşgörünümlü bir kuyruk yöneticisine ilişkin dosya sistemini ayarlamak için [Yordam](#) 'daki "1" sayfa 131 adımını izleyin.

### Bu görev hakkında

#### Yordam

1. Create a multi-instance queue manager on another server, QM1, using the file system you created in step "1" sayfa 131 in [Yordam](#).

Bkz. [Çok eşgörünümlü kuyruk yöneticisi yaratılması](#).

2. Kuyruk yöneticisini, yüksek kullanılabilirlikli kılabilir her iki sunucuda da başlatın.

Sunucu 1 'de:

```
strmqm -x QM1
```

Sunucu 2 'de:

```
strmqm -x QM1
```

3. Set up the client connection to run **amqsfhac**.

- Use the procedure in *IBM MQ kuruluşunun doğrulanması* for the platform, or platforms, that your enterprise use to set up a client connection, or the example scripts in [Yeniden bağlanabilir istemci örnekleri](#).
- İstemci kanalını, QM1çalıştıran iki sunucuya karşılık gelen iki IP adresine sahip olacak şekilde değiştirin.

Örnek komut dosyasında şunu değiştirin:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
```

Hedef:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
```

Burada `server1` ve `server2` , iki sunucunun anasistem adlarıdır ve 2345 , kanal dinleyicisinin dinlediği kapıdır. Genellikle bu varsayılan değer 1414değerine ayarlanır. 1414 olanağını varsayılan dinleyici yapılanışıyla kullanabilirsiniz.

4. Sınamaya ilişkin QM1 üzerinde iki yerel kuyruk yaratın.

Aşağıdaki MQSC komut dosyasını çalıştırın:

```
DEFINE QLOCAL(TARGETQ) REPLACE  
DEFINE QLOCAL(SIDEQ) REPLACE
```

5. Test the configuration with **amqsfhac**

```
amqsfhac QM1 TARGETQ SIDEQ 2 2 2
```

6. Dosya sistemi bütünlüğünü test ederken ileti bütünlüğünü test edin.

[Yordam](#)' ın "[5](#)" [sayfa 131](#) adımı sırasında **amqsfhac** komutunu çalıştırın.

```
amqsfhac QM1 TARGETQ SIDEQ 10 20 0
```

Etkin kuyruk yöneticisi yönetim ortamını durdurursanız, **amqsfhac** etkinleştikten sonra diğer kuyruk yöneticisi eşgörünümüye yeniden bağlanır. Durdurulan kuyruk yöneticisi yönetim ortamını yeniden başlatın; böylece, sonraki testinizde başarısızlığı tersine çevirebilirsiniz. Test programının, hata durumunda yedek sisteme geçiş işlemi için yeterli süre çalışması için ortamınızla denemelere dayalı yinelemelerin sayısını artırmanız gerekebilir.

## Sonuçlar

An example of running **amqsfhac** in step "[6](#)" [sayfa 136](#) is shown in the following example. Bu örnekte, test bir başarıdır.

```
Sample AMQSFHAC start  
qmname = QM1  
qname = TARGETQ  
sidename = SIDEQ
```



```

transize = 10
iterations = 20
verbose = 0
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Resolving MQRC_CALL_INTERRUPTED
MQGET browse side tranid=14 pSideinfo->tranid=14
Resolving to committed
Iteration 7
Iteration 8
Iteration 9
Iteration 10
Iteration 11
Iteration 12
Iteration 13
Iteration 14
Iteration 15
Iteration 16
Iteration 17
Iteration 18
Iteration 19
Sample AMQSFHAC end

```

Sinema bir sorun algıladıysa, çıkış hatayı bildirecektir. Bazı testlerde MQRC\_CALL\_INTERRUPTED , "Resolving to backed out" raporunu raporlayabilir. Sonuç için bir fark yok. Sonuç, diske yazma işleminin, hata gerçekleşmeden önce ya da sonra ağa bağlı dosya depolaması tarafından işlenip işlenmediğine bağlıdır.

### İlgili başvurular

[Yüksek kullanılabilirlik örnek programları](#)

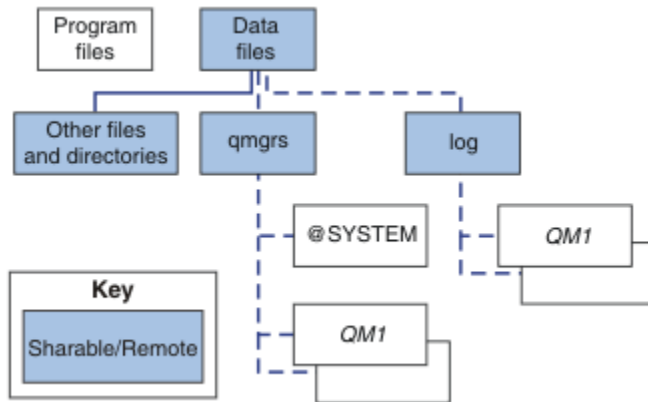
[amqmfsc \(dosya sistemi denetimi\)](#)

## Multi Sharing IBM MQ files on Multiplatforms

Bazı IBM MQ dosyalarına yalnızca etkin bir kuyruk yöneticisi tarafından erişilir, diğer dosyalar paylaşılır.

IBM MQ dosyaları, program dosyalarına ve veri dosyalarına bölünmektedir. Program dosyaları genellikle IBM MQ çalıştıran her bir sunucuya yerel olarak kurulur. Kuyruk yöneticileri, varsayılan veri dizinindeki veri dosyalarına ve dizinlere erişimi paylaşır. Bunlar, Şekil 32 sayfa 137 içinde gösterilen qmgrs ve log dizinlerinin her birinde bulunan kendi kuyruk yöneticisi dizini ağaçlarına dışlayıcı erişim gerektirir.

Şekil 32 sayfa 137 , IBM MQ dizin yapısının üst düzey bir görünümüdür. Kuyruk yöneticileri arasında paylaşılabilen ve uzak kılınabilen dizinleri gösterir. Ayrıntılar altyapıya göre değişir. Noktalı çizgiler, yapılandırılabilir yolları gösterir.



Şekil 32. IBM MQ dizin yapısının genel görünümü

## Program dosyaları

Program dosyaları dizini genellikle varsayılan konumda bırakılır, yerel olur ve sunucudaki tüm kuyruk yöneticileriyle paylaşılır.

## Veri dosyaları

The data files directory is typically local in the default location, /var/mqm on UNIX and Linux systems and configurable on installation on Windows. Kuyruk yöneticileri arasında paylaşılır. Varsayılan konum uzakını yapabilirsiniz, ancak bunu IBM MQ' un farklı kuruluşları arasında paylaşmayın. IBM MQ yapılandırma noktasındaki DefaultPrefix özniteliği bu yola işaret eder.

## qmgrs

Kuyruk yöneticisi verilerinin yerini belirlemenin iki farklı yolu vardır.

### ÖnekKullanma

Prefix özniteliği, qmgrs dizininin yerini belirtir. IBM MQ , kuyruk yöneticisi dizini adını kuyruk yöneticisi adından oluşturur ve bunu, qmgrs dizininin bir alt dizini olarak yaratır.

Prefix özniteliği, QueueManager stanza içinde bulunur ve DefaultPrefix özniteliğinden edinilmiş olan değerden devralınır. Yönetim basitliği için varsayılan olarak, kuyruk yöneticileri genellikle aynı qmgrs dizinini paylaşır.

QueueManager kısmı mqsc . ini dosyasında yer alan bir dosyadır.

Herhangi bir kuyruk yöneticisi için qmgrs dizininin konumunu değiştirirseniz, Prefix özniteliğinin değerini değiştirmeniz gerekir.

Bir UNIX and Linux platformu için [Şekil 32 sayfa 137](#) içindeki QM1 dizinine ilişkin Prefix özniteliği şudur:

```
Prefix=/var/mqm
```

### DataPathanağının kullanılması

DataPath özniteliği, kuyruk yöneticisi veri dizininin yerini belirtir.

DataPath özniteliği, kuyruk yöneticisi veri dizininin adı da içinde olmak üzere, tam yolu belirtir. DataPath özniteliği, kuyruk yöneticisi verileri dizininin tamamlanmamış bir yolu belirten Prefix özniteliğinden farklıdır.

DataPath özniteliği belirtilirse, QueueManager stanza içinde yer alır. Belirtilmişse, Prefix (Önek) özniteindeki herhangi bir değere göre öncelik kazanır.

QueueManager kısmı mqsc . ini dosyasında yer alan bir dosyadır.

Herhangi bir kuyruk yöneticisine ilişkin kuyruk yöneticisi veri dizininin yerini değiştirirseniz, DataPath özniteliğinin değerini değiştirmelisiniz.

Bir UNIX ya da Linux platformu için [Şekil 32 sayfa 137](#) içindeki QM1 dizinine ilişkin DataPath özniteliği şudur:

```
DataPath=/var/mqm/qmgrs/QM1
```

## log

Günlük dizini, kuyruk yöneticisi yapılandırmasındaki Log kısmı içindeki her kuyruk yöneticisi için ayrı olarak belirtilir. The queue manager configuration is in qm . ini.

### DataPath/QmgrName/@IPCC alt dizinleri

DataPath/QmgrName/@IPCC alt dizinleri paylaşılan dizin yolunda yer alıyor. Bunlar, IPC dosya sistemi nesnelere için dizin yolu oluşturmak üzere kullanılır. Bir kuyruk yöneticisi sistemler arasında paylaşıldığında, kuyruk yöneticisinin ad alanını ayırt edebilmeleri gerekir.

IPC dosya sistemi nesnelere sistem tarafından ayırt edilmesi gerekir. Kuyruk yöneticisinin çalıştığı her sistem için bir alt dizin, dizin yoluna eklenerek [Şekil 33 sayfa 139'](#) e bakın.

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

### Şekil 33. Örnek IPC altdizini

*myHostName* , işletim sistemi tarafından döndürülen anasistem adının ilk 20 karakterine uymaktadır. Bazı sistemlerde, kesme işleminden önce anasistem adı en çok 64 karakter uzunluğunda olabilir. The generated value of *myHostName* might cause a problem for two reasons:

1. İlk 20 karakter benzersiz değil.
2. Anasistem adı, her zaman bir sisteme aynı anasistem adını ayırmayan bir DHCP algoritması tarafından oluşturulur.

In these cases, set *myHostName* using the environment variable, MQS\_IPC\_HOST ; see [Şekil 34 sayfa 139.](#)

```
export MQS_IPC_HOST= myHostName
```

### Şekil 34. Örnek: ayar MQS\_IPC\_HOST

## Diğer dosyalar ve dizinler

İzleme dosyalarını içeren dizin ve ortak hata günlüğü gibi diğer dosyalar ve dizinler, olağan durumda yerel dosya sisteminde paylaşılır ve saklanır.

Paylaşılan dosya sistemleri desteği ile IBM MQ , dosya sistemi kilitlerini kullanarak bu dosyalara özel erişimi yönetir. Bir dosya sistemi kilidi, belirli bir kuyruk yöneticisinin yalnızca bir eşgörünümünün bir kerede etkin olmasını sağlar.

Belirli bir kuyruk yöneticisinin ilk eşgörünümünü başlattığınızda, kuyruk yöneticisi dizininin sahipliğini alır. İkinci bir yönetim ortamı başladığınızda, yalnızca ilk yönetim ortamı durduysa, bu yönetim ortamı sahiplik alabilir. İlk kuyruk yöneticisi çalışmaya devam ederse, ikinci eşgörünüm başlatılamaz ve kuyruk yöneticisi başka bir yerde çalıştırıyor demektir. İlk kuyruk yöneticisi durduysa, ikinci kuyruk yöneticisi kuyruk yöneticisi kütüklerinin sahipliğini devralır ve çalışmakta olan kuyruk yöneticisi olur.

İkinci kuyruk yöneticisinin ilkinden devralan prosedürü otomatikleştirebilirsiniz. İlk kuyruk yöneticisini, başka bir kuyruk yöneticisinin devralmasına izin veren `strmqm -x` seçeneği ile başlatın. İkinci kuyruk yöneticisi, kuyruk yöneticisi kütüklerinin sahipliğini alma girişiminde bulunmadan önce kuyruk yöneticisi kütüklerinin kilidi açılıncaya kadar bekler ve başlar.

## Linux > UNIX **UNIX and Linux sistemlerindeki dizin yapısı**

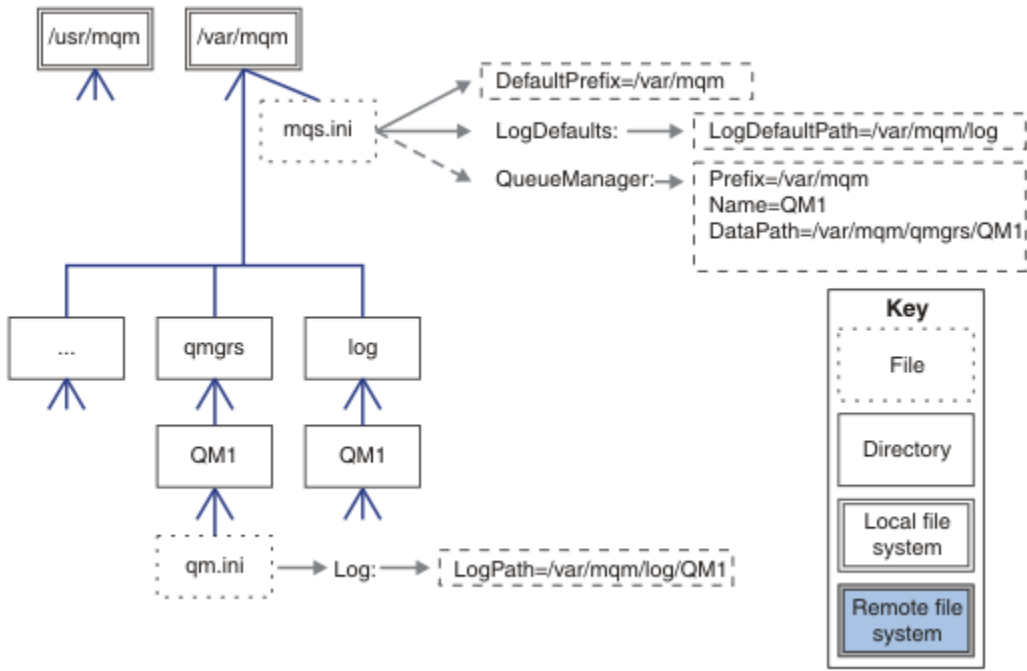
UNIX and Linux sistemlerinde IBM MQ dizin yapısı, daha kolay yönetim, daha iyi performans ve daha iyi güvenilirlik için farklı dosya sistemlerine eşlenebilir.

Çok eşgörünümlü kuyruk yöneticileri çalıştırmak için paylaşılan dosya sistemlerinden yararlanmak için IBM MQ ' in esnek dizin yapısını kullanın.

Use the command `crtmqm QM1` to create the directory structure shown in [Şekil 35 sayfa 140](#) where R is the release of the product. Bu, bir IBM MQ sisteminde yaratılmış bir kuyruk yöneticisi için tipik bir dizin yapısıdır. Bazı dizinler, dosyalar ve .ini öznitelik ayarları netlik için atlanır ve başka bir kuyruk yöneticisi adı mangging tarafından değiştirilebilir. Dosya sistemlerinin adları farklı sistemlere göre değişir.

Tipik kurulumda, yarattığınız her kuyruk yöneticisi yerel dosya sisteminde ortak log ve qmgrs dizinlerine işaret eder. Çok eşgörünümlü bir yapılandırmada, log ve qmgrs dizinleri, başka bir IBM MQ kuruluşuyla paylaşılan bir ağ dosya sisteminde yer alıyor.

[Şekil 35 sayfa 140](#) , IBM MQ v7.R on AIX (burada R , ürünün yayınının yayınıdır). Diğer çok eşgörünümlü yapılandırmalara ilişkin örnekler için bkz. "[UNIX and Linux sistemlerindeki örnek dizin yapılandırmaları](#)" [sayfa 145.](#)



Şekil 35. UNIX and Linux sistemleri için örnek varsayılan IBM MQ dizin yapısı

The product is installed into /usr/mqm on AIX and /opt/mqm on the other systems, by default. Çalışma dizinleri /var/mqm dizinine kurulur.

**Not:** IBM MQ kurulmadan önce /var/mqm kütük sistemini yarattıysanız, mqm kullanıcısının tam izinlerine sahip olduğundan emin olun; örneğin, dosya kipi 755.

**Not:** The /var/mqm/errors directory should be a separate filesystem to prevent FFDCs produced by the queue manager from filling the filesystem that contains /var/mqm.

Ek bilgi için [Creating file systems on UNIX and Linux systems](#) başlıklı konuya bakın.

Log ve qmgrs dizinleri varsayılan konumlarında, mqs.ini dosyasındaki LogDefaultPath ve DefaultPrefix özniteliklerinin varsayılan değerleri tarafından tanımlandığı şekilde gösterilir. Bir kuyruk yöneticisi yaratıldığında, varsayılan olarak kuyruk yöneticisi veri dizini DefaultPrefix/qmgrsdizinde yaratılır ve günlük kütüğü dizini LogDefaultPath/logdizinde yaratılır. LogDefaultYolu ve DefaultPrefix yalnızca kuyruk yöneticilerinin ve günlük dosyalarının varsayılan olarak oluşturulduğu etkiye neden olur. Bir kuyruk yöneticisi dizininin gerçek konumu mqs.ini dosyasına kaydedilir ve günlük dosyası dizininin konumu qm.ini dosyasına kaydedilir.

Bir kuyruk yöneticisine ilişkin günlük dosyası dizini, LogPath özniteindeki qm.ini dosyasında tanımlanır. Bir kuyruk yöneticisi için LogPath özniteliğini ayarlamak için **crtmqm** komutu üzerinde -ld seçeneğini kullanın; örneğin, **crtmqm -ld LogPath QM1**. If you omit the ld parameter the value of LogDefaultYolu is used instead.

Kuyruk yöneticisi verileri dizini, mqs.ini dosyasındaki QueueManager stanza içindeki DataPath özniteisinde tanımlanır. Bir kuyruk yöneticisi için DataPath ayarlamak için **crtmqm** komutundaki -md seçeneğini kullanın; örneğin, **crtmqm -md DataPath QM1**. md parametresini çıkarırsanız, bunun yerine DefaultPrefix ya da Prefix özniteliğinin değerini kullanın. Önek, DefaultPrefix'ten öncelikli olarak uygulanır.

Tipik olarak, tek bir komutta hem günlük hem de veri dizinlerini belirten QM1 oluşturur.

```
crtmqm
-md DataPath -ld
LogPath QM1
```

Kuyruk yöneticisi durdurulduğunda, qm. ini dosyasındaki DataPath ve LogPath özniteliklerini düzenleyerek, var olan bir kuyruk yöneticisinin kuyruk yöneticisi günlüğünün ve veri dizinlerinin yerini değiştirebilirsiniz.

The path to the errors directory, like the paths to all the other directories in /var/mqm, is not modifiable. Ancak, dizinlerin farklı dosya sistemlerine ya da sembolik olarak farklı dizinlere bağlanabileceği.

Linux

UNIX

### **UNIX and Linux sistemlerindeki dizin içeriği**

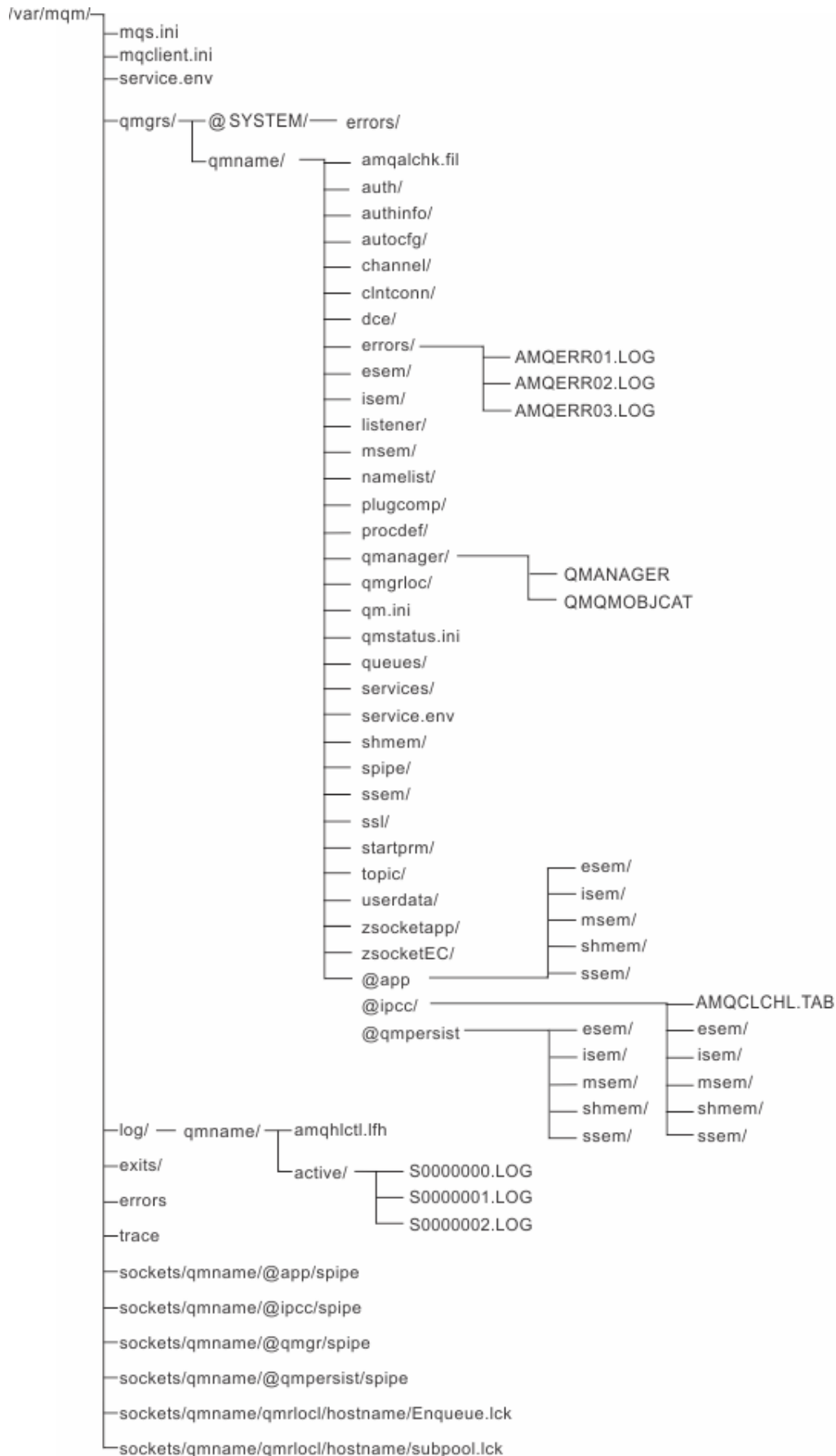
Kuyruk yöneticisiyle ilişkili dizinlerin içeriği.

Ürün dosyalarının konumu hakkında bilgi için bkz. [Kuruluş konumunu seçme](#)

Diğer dizin yapılandırmalarıyla ilgili bilgi için bkz. [“Çoklu Platformlar üzerinde Dosya Sistemi Desteği Planlanması”](#) sayfa 127.

V 9.1.4

Aşağıdaki dizin yapısı, bir kuyruk yöneticisi bir süredir kullanımda olduktan sonra IBM MQ temsilcisini temsil eder. Sahip olduğunuz gerçek yapı, kuyruk yöneticiliklerinde hangi işlemlerin gerçekleştirildiğine bağlıdır.



## **/var/mqm/**

/var/mqm dizini, tek bir kuyruk yöneticisine değil, bir bütün olarak IBM MQ kuruluşu için geçerli olan yapılandırma dosyalarını ve çıkış dizinlerini içerir.

<i>Çizelge 13. Documented content of the /var/mqm directory on UNIX</i>	
<b>Dizin ya da dosya adı</b>	<b>İçindekiler</b>
<b><u>mqs.ini</u></b>	IBM MQ installation wide configuration file, read when a queue manager starts. Dosya yolu, <b>AMQ_MQS_INI_LOCATION</b> ortam değişkeni kullanılarak değiştirilebilir. Bunun ayarlandığından ve <b>stirmqm</b> komutunun çalıştırıldığı kabukta dışa aktarıldığından emin olun.
<b><u>mqclient.ini</u></b>	IBM MQ MQI client programları tarafından okunan varsayılan istemci yapılandırma dosyası. Dosya yolu, <b>MQCLNTCF</b> ortam değişkeni kullanılarak değiştirilebilir.
<b><u>service.env</u></b>	Bir hizmet süreci için makine kapsamı ortam değişkenlerini içerir. Dosya yolu düzeltti.
<b><u>hatalar/</u></b>	Makine kapsamı hata günlükleri ve FFST dosyaları. Dizin yolu düzeltti. Ayrıca bkz. <a href="#">FFST: IBM MQ for UNIX ve Linux sistemleri</a> .
<b><u>yuvalar/</u></b>	Her kuyruk yöneticisine ilişkin bilgileri yalnızca sistem kullanımı için içerir.
<b><u>izleme/</u></b>	İzleme dosyaları. Dizin yolu düzeltti.
<b><u>web/</u></b>	mqweb sunucusu dizini.
<b><u>çıkışlar/</u></b>	Kullanıcı kanalı çıkış programlarını içeren varsayılan dizin. mqs.ini dosyasındaki ApiExit kısmında yer değiştirebilen yer.
<b><u>exits64/</u></b>	

## **/var/mqm/qmgrs/qmname/**

/var/mqm/qmgrs/qmname/ , kuyruk yöneticisine ilişkin dizinleri ve dosyaları içerir. Dizin, etkin kuyruk yöneticisi yönetim ortamı tarafından dışlayıcı erişim için kilitlenir. Dizin yolu doğrudan mqs.ini dosyasında değiştirilebilir ya da **crtmqm** komutunun **md** seçeneği kullanılarak değiştirilebilir.

<i>Çizelge 14. Documented contents of the /var/mqm/qmgrs/qmname directory on UNIX</i>	
<b>Dizin ya da dosya adı</b>	<b>İçindekiler</b>
<b><u>qm.ini</u></b>	Kuyruk yöneticisi yapılanış kütüğü, kuyruk yöneticisi başlatıldığında okunur.
<b><u>hatalar/</u></b>	Kuyruk yöneticisi kapsam hata günlükleri. <b>qmname</b> = @system , bilinmeyen ya da kullanılmayan bir kuyruk yöneticisine ilişkin kanalla ilgili iletileri içerir.

<i>Çizelge 14. Documented contents of the /var/mqm/qmgrs/qmname directory on UNIX (devamı var)</i>		
<b>Dizin ya da dosya adı</b>	<b>İçindekiler</b>	
<b>@ipcc/ AMQCLCHL.TAB</b>	Default client channel control table, created by the IBM MQ server, and read by IBM MQ MQI client programs. <b>MQCHLLIB</b> ve <b>MQCHLTAB</b> ortam değişkenleri kullanılarak dosya yolu değiştirilebilir.	
<b>qmanager</b>	Kuyruk yöneticisi nesne dosyası: QMANAGER Kuyruk yöneticisi nesne kataloğu: QMQMOBJCAT	
<b>authinfo/ kanal/ clntconn/ dinleyici/ ad listesi/ procdef/ kuyruklar/ hizmetler/ konular/</b>	Kuyruk yöneticisi içinde tanımlanan her nesne, bu dizinlerdeki bir dosyayla ilişkilendirilir. Dosya adı yaklaşık olarak tanımlama adıyla eşleşir; bkz. <a href="#">IBM MQ dosya adlarını anlama</a> .	
...		
<b>V 9.1.5 kullanıcıverileri/</b>		Uygulamaların kalıcı durumunu saklamak için kullanılabilir (kuyruk yöneticileri farklı düğümlere taşınırken RDQM tarafından kullanılabilir-bkz. <a href="#">Kalıcı uygulama durumunun saklanması</a> .)
<b>V 9.1.4 DataPath\autocfg</b>		Otomatik yapılandırma için kullanılır

### **/var/mqm/log/qmname/**

/var/mqm/log/qmname/ , kuyruk yöneticisi günlük dosyalarını içerir. Dizin, etkin kuyruk yöneticisi yönetim ortamı tarafından dışlayıcı erişim için kilitletir. Dizin yolu, qm.ini dosyasında değiştirilebilir ya da **crtmqm** komutunun **ld** seçeneği kullanılarak.

<i>Çizelge 15. Documented contents of the /var/mqm/log/qmname directory on UNIX</i>	
<b>Dizin ya da dosya adı</b>	<b>İçindekiler</b>
<b>amqhlctl.lfh</b>	Denetim dosyasını günlüğe kaydet.
<b>etkin/</b>	Bu dizin, S0000000.LOG(LOG), S0000001.LOG, S0000002.LOG(LOG), vb.

### **/opt/mqm**

/opt/mqm , varsayılan olarak, çoğu platformdaki kuruluş dizini olur. Altyapının ya da şirketinizin kullandığı platformlardaki kuruluş dizini için gereken alan miktarına ilişkin daha fazla bilgi edinmek için "[Çoklu Platformlar üzerindeki Disk Alanı Gereksinimleri](#)" sayfa 124 konusuna bakın.



UNIX and Linux sistemlerinde diğer dosya sistemi yapılandırmalarına örnekler.

Çeşitli amaçlara ulaşmak için IBM MQ dizin yapısını çeşitli şekillerde özelleştirebilirsiniz.

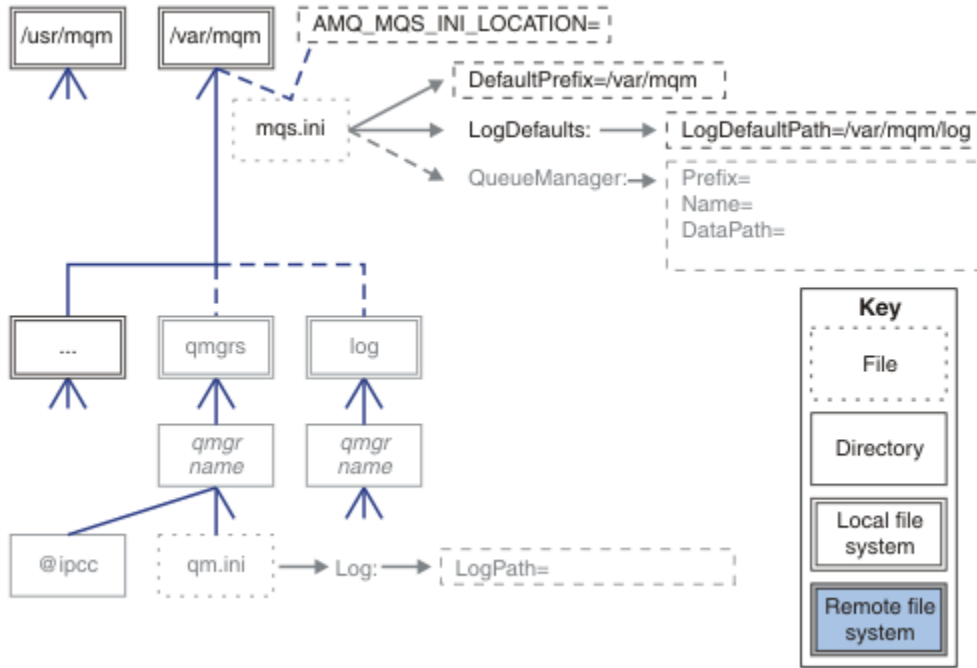
- Çok eşgörunümlü bir kuyruk yöneticisi yapılandırmak için, uzak paylaşılan kütük sistemlerine qmgrs ve log dizinlerini yerleştirin.
- G/Ç çekişmesini azaltarak başarıyı artırmak için, veri ve günlük dizinleri için ayrı dosya sistemleri kullanın ve dizinleri farklı disklere ayırın.
- Daha yüksek performans üzerinde etkisi olan dizinler için daha hızlı depolama aygıtları kullanın. Fiziksel aygıt gecikme süresi, bir aygıtın yerel olarak mı, yoksa uzaktan mı monte edildiğinden, sürekli ileti sisteminin performansında sık sık daha önemli bir etkidir. Aşağıdaki liste, en çok hangi dizinlerin en düşük düzeyde duyarlı olduğunu gösterir.

1. log
2. qmgrs
3. /usr/mqm içinde olmak üzere diğer dizinler

- Örneğin, yedek disk dizisi gibi iyi dayanıklılığa sahip depolama için ayrılan dosya sistemlerinde qmgrs ve log dizinlerini oluşturun.
- Ortak hata günlüklerini bir ağ dosya sisteminde değil, yerel olarak var/mqm/errors' ta saklamak daha iyidir. Böylece, ağ dosya sistemiyle ilgili hata günlüğe kaydedilebilir.

Şekil 36 sayfa 145 , alternatif IBM MQ dizin yapılarının türetildiği bir şablondur. Şablonda noktalı çizgiler, yapılandırılabilir yolları temsil eder. Örneklerde, noktalı çizgiler, AMQ\_MQS\_INI\_LOCATION ortam değişkeninde saklanan yapılandırma bilgilerine karşılık gelen katı çizgileriyle değiştirilir ve mqs.ini ve qm.ini dosyalarında yer alır.

**Not:** Yol bilgileri, mqs.ini ya da qm.ini dosyalarında görüldüğü şekilde gösterilir. If you supply path parameters in the **crtmqm** command, omit the name of the queue manager directory: the queue manager name is added to the path by IBM MQ.



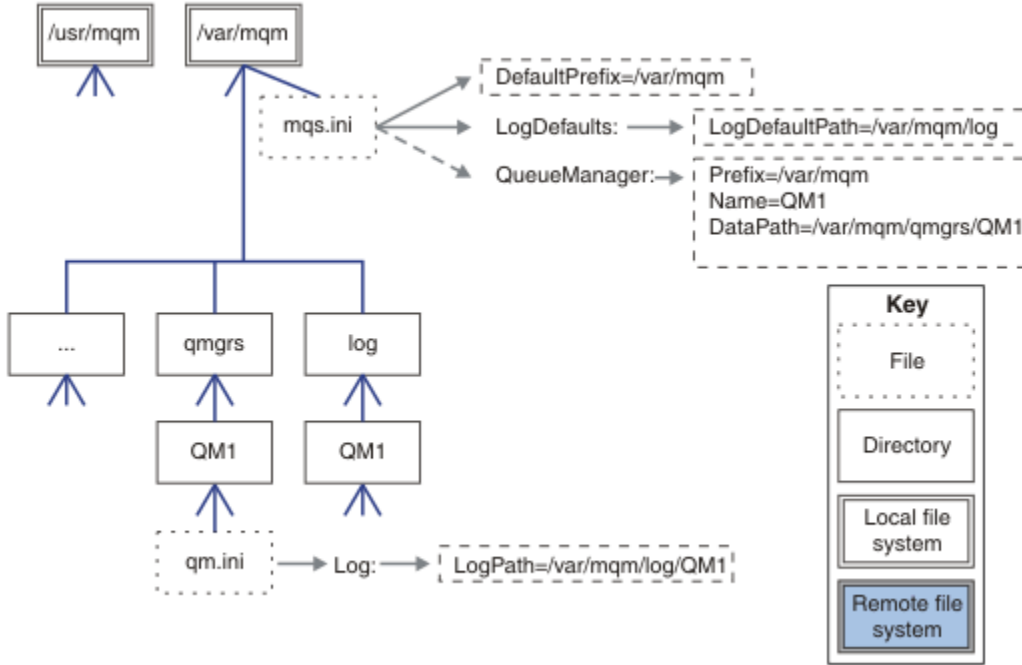
Şekil 36. Dizin yapısı örüntü şablonu

## IBM MQ için tipik dizin yapısı

Şekil 37 sayfa 146 is the default directory structure created in IBM MQ by issuing the command **crtmqm QM1**.

The `mqc.ini` file has a stanza for the QM1 queue manager, created by referring to the value of `DefaultPrefix`. The `qm.ini` file has a value for `LogPath`, set by reference to `LogDefaultYolu` in `mqc.ini`.

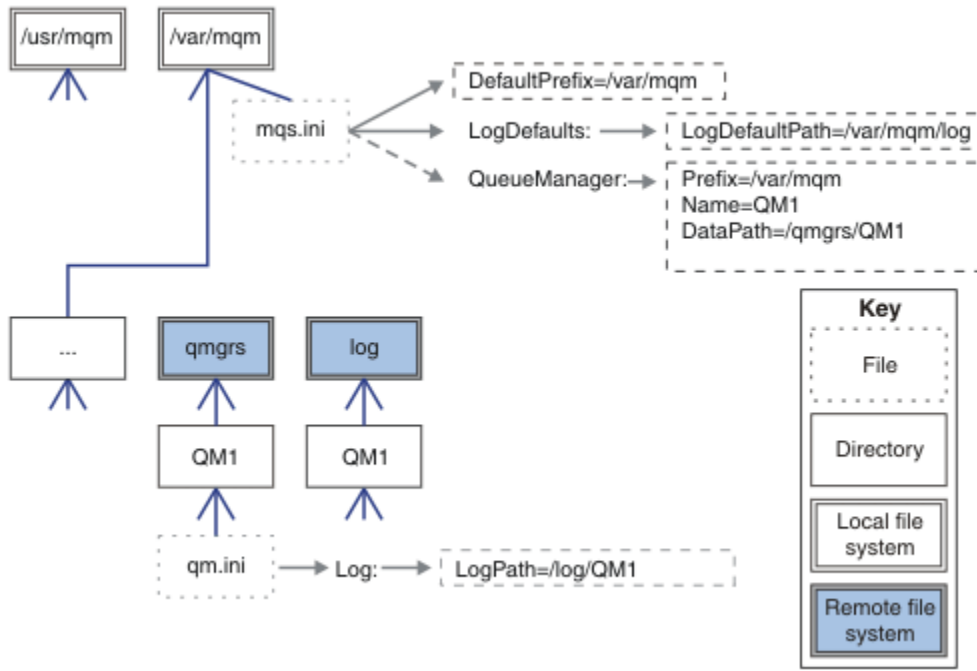
Use the optional **crtmqm** parameters to override the default values of `DataPath` and `LogPath`.



Şekil 37. UNIX and Linux sistemleri için örnek varsayılan IBM MQ dizin yapısı

## Varsayılan qmgrs ve log dizinlerini paylaşma

“Her şeyi paylaşın” sayfa 148' in alternatifi, qmgrs ve log dizinlerini ayrı olarak paylaşmasıdır ( Şekil 38 sayfa 147 ). Bu yapılandırmada, varsayılan `mqc.ini` yerel `/var/mqm` dosya sisteminde saklandığı için `AMQ_MQS_INI_LOCATION` ayarlamaya gerek yoktur. `mqclient.ini` ve `mqserver.ini` gibi dosyalar ve dizinler de paylaşılmaz.

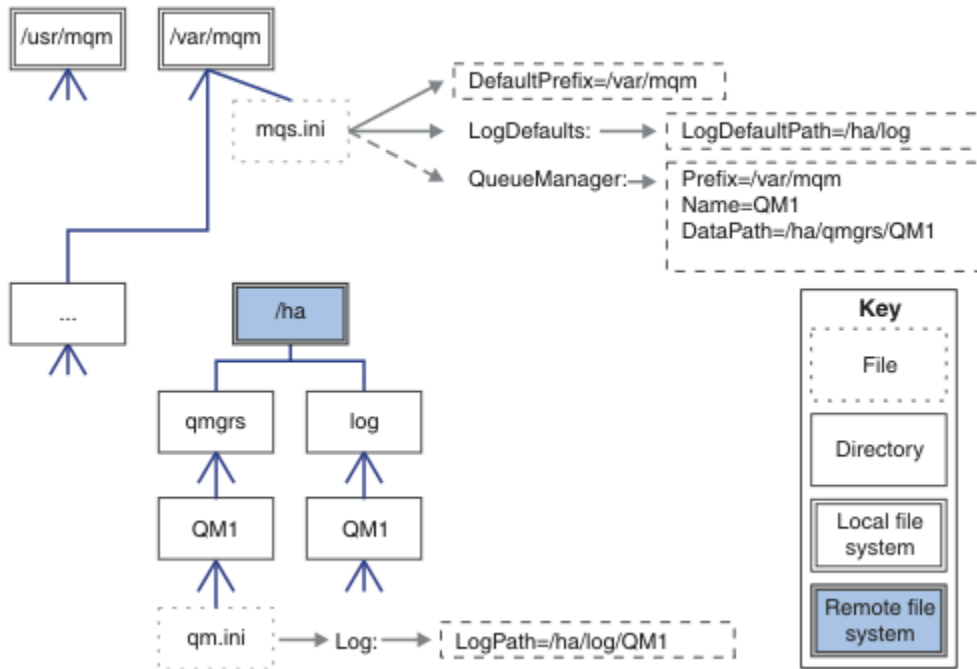


Şekil 38. Share qmgrs and log directories

### Share named qmgrs and log directories

The configuration in Şekil 39 sayfa 147 places the log and qmgrs in a common named remote shared file system called /ha. Aynı fiziksel yapılandırma iki farklı şekilde yaratılabilir.

1. `LogDefaultPath=/ha` seçeneğini ayarlayın ve daha sonra, komutu çalıştırın; `crtmqm - md /ha/qmgrs QM1`. Sonuç, Şekil 39 sayfa 147' ta tam olarak gösterildiği gibidir.
2. Leave the default paths unchanged and then run the command, `crtmqm - ld /ha/log - md /ha/qmgrs QM1`.



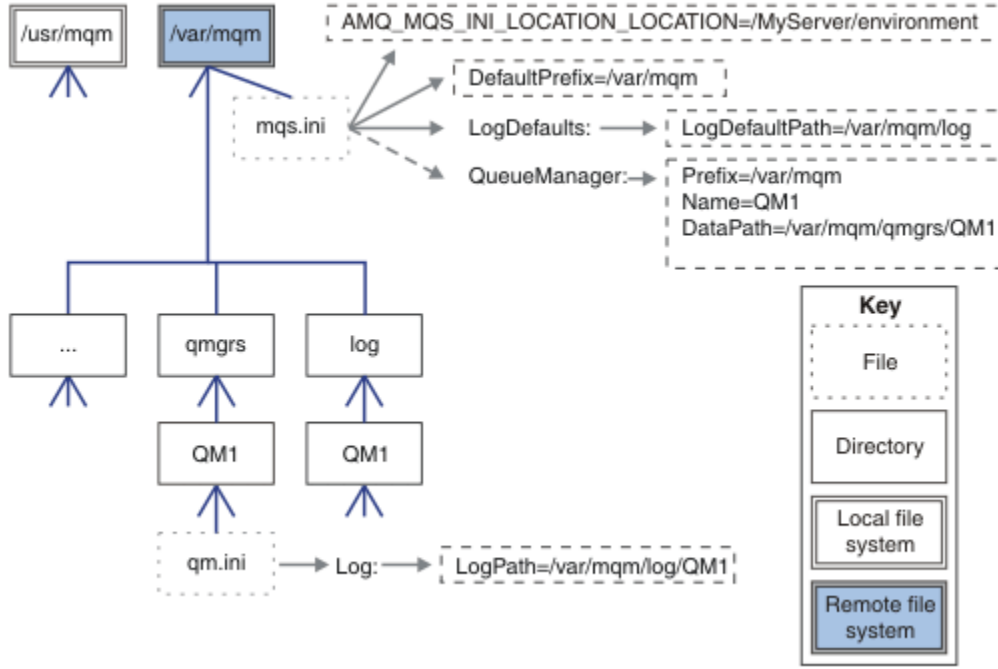
Şekil 39. Share named qmgrs and log directories

## Her şeyi paylaşın

Şekil 40 sayfa 148 , hızlı ağ üzerinde dosya depolaması olan sistem için basit bir yapılandırma.

/var/mqm bağlantısını uzak paylaşılan bir dosya sistemi olarak sisteme bağlayın. Varsayılan olarak, QM1'u başlattığınızda /var/mqm' u arar, paylaşılan dosya sisteminde bulur ve /var/mqmdosyasında mqs . ini dosyasını okur. Tüm sunucularınızda kuyruk yöneticileri için tek bir /var/mqm/mqs . ini dosyası kullanmak yerine, her sunucuda AMQ\_QMS\_INI\_LOCATION ortam değişkenini farklı mqs . ini dosyalarına işaret edecek şekilde ayarlayabilirsiniz.

**Not:** The contents of the generic error file in /var/mqm/errors/ are shared between queue managers on different servers.



Şekil 40. Her şeyi paylaşın

Çok eşgörünümlü kuyruk yöneticileri için bunu kullanamayacağınızı göz önünde bulundurun. Bunun nedeni, çok eşgörünümlü bir kuyruk yöneticisinde bulunan her anasistemin, semaforlar ve paylaşılan bellek gibi yerel verileri izlemek için /var/mqm ' un kendi yerel kopyasına sahip olması gerekir. Bu varlıklar anasistemler arasında paylaşılamaz.

## Windows Windows sistemlerindeki dizin yapısı

Windows üzerindeki kuyruk yöneticisi yapısını ve dizinlerini bulma.

IBM MQ for Windows kurulumu için varsayılan dizinler şunlardır:

### Program dizini

C:\Program Files\IBM\MQ

### Veri Dizini

C:\ProgramData\IBM\MQ

**Önemli:** Windows Windows kurulumlarında, ürünün kayıt girişlerini ve/ya da kuyruk yöneticilerini içeren önceki bir kurulumu yoksa, dizinler belirtildiği gibi olur. Bu durumda, yeni kurulum eski veri dizini konumunu kullanır. Daha fazla bilgi için bkz. [Program and data directory locations](#).

Hangi kurulum dizininin ve hangi veri dizininin kullanıldığını öğrenmek istiyorsanız, `dspmqr` komutunu çalıştırın.

Kurulum dizini, **InstPath** alanında listelenir ve veri dizini, **DataPath** alanında listelenir.

Örneğin, **dspmqrver** komutunun çalıştırılması aşağıdaki bilgileri içerir:

```
>dspmqrver
Name: IBM MQ
Version: 9.0.0.0
Level: p900-L160512.4
BuildType: IKAP - (Production)
Platform: IBM MQ for Windows (x64 platform)
Mode: 64-bit
O/S: Windows 7 Professional x64 Edition, Build 7601: SP1
InstName: Installation1
InstDesc:
Primary: Yes
InstPath: C:\Program Files\IBM\MQ
DataPath: C:\ProgramData\IBM\MQ
MaxCmdLevel: 900
LicenseType: Production
```

## Çok eşgörünümli kuyruk yöneticileri

Çok eşgörünümli bir kuyruk yöneticisini yapılandırmak için, günlük ve veri dizinlerinin ağ üzerinde çalışan depolama alanına yerleştirilmeleri gerekir. Tercihen, tercihen kuyruk yöneticisinin eşgörünümlerini çalıştıran sunuculardan herhangi birine farklı bir sunucu üzerinde olmalıdır.

Kuyruk yöneticisi verilerinin ve günlük dizinlerinin konumunu daha kolay belirtmek için **crtmqm** komutu **-md** ve **-ld**' de iki parametre sağlanır. **-md** parametresini belirtmenin etkisi dört katlıdır:

1. mqs.ini stanza QueueManager\QmgrName , kuyruk yöneticisi verileri dizinini gösteren yeni bir değişken ( DataPath) içerir. Unlike the Önek variable, the path includes the name of the queue manager directory.
2. mqs.ini dosyasında saklanan kuyruk yöneticisi yapılanış bilgileri Ad, Önek, Dizin ve DataPatholarak azaltılır.

## Windows Dizin içeriği

IBM MQ dizinlerinin konumunu ve içeriğini listeler.

Bir IBM MQ yapılanışında üç ana dosya kümesi ve izin vardır:

1. Yürütülebilir dosya ve yalnızca bakım uygulandığında güncelleştirilen diğer salt okunur dosyalar. Örneğin:

- Beni Oku dosyası
- IBM MQ Explorer eklentisi ve yardım dosyaları
- Lisans dosyaları

Bu dosyalar [Çizelge 16 sayfa 149](#) içinde açıklanmıştır.

2. Belirli bir kuyruk yöneticisine özgü olmayan değiştirilebilir, değiştirilebilir dosyalar ve dizinler. Bu dosyalar ve dizinler [Çizelge 17 sayfa 150](#) içinde anlatılır.
3. Bir sunucudaki her kuyruk yöneticisine özgü dosyalar ve dizinler. Bu dosyalar ve dizinler [Çizelge 18 sayfa 151](#) içinde anlatılır.

## Kaynak dizinleri ve dosyaları

Kaynak dizinleri ve dosyaları, bir kuyruk yöneticisini çalıştırmak için tüm yürütülebilir kodları ve kaynakları içerir. The variable, FilePath, in the installation specific IBM MQ configuration registry key, contains the path to the resource directories.

Çizelge 16. FilePath dizinindeki dizinler ve dosyalar	
Dosya yolu	İçindekiler
FilePath\bin	Komutlar ve DLL ' ler
FilePath\bin64	Komutlar ve DLL ' ler (64 bit)

<i>Çizelge 16. FilePath dizinindeki dizinler ve dosyalar (devamı var)</i>	
<b>Dosya yolu</b>	<b>İçindekiler</b>
<i>FilePath\conv</i>	Veri dönüştürme çizelgeleri
<i>FilePath\doc</i>	Sihirbaz yardım dosyaları
<i>FilePath\MQExplorer</i>	Explorer ve Explorer yardımı Eclipse eklentileri
<i>FilePath\gskit8</i>	Genel güvenlik seti
<i>FilePath\java</i>	JRE de içinde olmak üzere Java kaynakları
<i>FilePath\licenses</i>	Lisans bilgisi
<i>FilePath\Non_IBM_License</i>	Lisans bilgisi
<i>FilePath\properties</i>	Dahili olarak kullanılır
<i>FilePath\Tivoli</i>	
<i>FilePath\tools</i>	Geliştirme kaynakları ve örnekleri
<i>FilePath\web</i>	IBM MQ Console ve REST API installation component file structure for non-editable files (Düzenlenebilir olmayan dosyalar için)
<i>FilePath\Uninst</i>	Dahili olarak kullanılır
<i>FilePath\README.TXT</i>	Readme dosyası

## Dizin, kuyruk yöneticisine özgü değil

Bazı dizinler, belirli bir kuyruk yöneticisine özgü olmayan izleme dosyaları ve hata günlükleri gibi dosyaları içerir. *DefaultPrefix* değişkeni, bu dizinlerin yolunu içerir. *DefaultPrefix*, *AllQueueManagers* stanza'nın bir parçasıdır.

<i>Çizelge 17. DefaultPrefix dizinindeki dizinler ve dosyalar</i>	
<b>Dosya yolu</b>	<b>İçindekiler</b>
<i>DefaultPrefix\config</i>	Dahili olarak kullanılır
<i>DefaultPrefix\conv</i>	ccsid_part2.tbl and ccid.tbl data conversion control file, described in <a href="#">Veri dönüştürme</a>
<i>DefaultPrefix\errors</i>	Kuyruk yöneticisi olmayan hata günlükleri, AMQERR nn.LOG
<i>DefaultPrefix\exits</i>	Kanal çıkış programları
<i>DefaultPrefix\exits64</i>	Kanal çıkış programları (64 bit)
<i>DefaultPrefix\ipc</i>	Kullanılmadı
<i>DefaultPrefix\qmgrs</i>	<a href="#">Çizelge 18 sayfa 151</a> içinde anlatılan
<i>DefaultPrefix\trace</i>	iz dosyaları
<i>DefaultPrefix\web</i>	Kullanıcı tarafından düzenlenebilir dosyalar için IBM MQ Console ve REST API kuruluş bileşeni dosya yapısı içinde anlatılan
<i>DefaultPrefix\amqmjpse.txt</i>	Dahili olarak kullanılır

## Kuyruk yöneticisi dizinleri

Bir kuyruk yöneticisi yarattığınızda, kuyruk yöneticisine özgü yeni bir dizin kümesi yaratılır.

**-md filepath** parametresiyle bir kuyruk yöneticisi oluşturursanız, yol, mqsc.ini dosyasının kuyruk yöneticisi stanza içindeki *DataPath* değişkeninde saklanır. If you create a queue manager without setting the **-md filepath** parameter, the queue manager directories are created in the path stored in *DefaultPrefix*, and the path is copied into the *Önek* variable in the queue manager stanza of the mqsc.ini file.

Çizelge 18. <i>DataPath</i> ve <i>Prefix\qmgrs\QmgrName</i> dizinlerindeki dizinler ve dosyalar	
Dosya yolu	İçindekiler
<i>DataPath\@ipcc</i>	AMQCLCHL . TABiçin varsayılan konum, istemci bağlantı tablosu.
<i>DataPath\authinfo</i>	Dahili olarak kullanılır.
<i>DataPath\channel</i>	
<i>DataPath\clntconn</i>	
<i>DataPath\errors</i>	Hata günlükleri, AMQERR nn.LOG
<i>DataPath\listener</i>	Dahili olarak kullanılır.
<i>DataPath\namelist</i>	
<i>DataPath\plugcomp</i>	
<i>DataPath\procdef</i>	
<i>DataPath\qmanager</i>	
<i>DataPath\queues</i>	
<i>DataPath\services</i>	
<i>DataPath\ssl</i>	
<i>DataPath\startpim</i>	
<i>DataPath\topic</i>	
<i>DataPath\active</i>	
<i>DataPath\active.dat</i>	
<i>DataPath\amqalchk.fil</i>	
<i>DataPath\master</i>	
<i>DataPath\master.dat</i>	
<i>DataPath\qm.ini</i>	Kuyruk yöneticisi yapılandırması
<i>DataPath\qmstatus.ini</i>	Kuyruk yöneticisi durumu
<b>V 9.1.5</b> <i>DataPath\userdata</i>	Uygulamaların kalıcı durumunu saklamak için kullanılabilir.
<i>Prefix\qmgrs\QmgrName</i>	Dahili olarak kullanılır
<i>Prefix\qmgrs\@SYSTEM</i>	Kullanılmadı
<i>Prefix\qmgrs\@SYSTEM\errors</i>	
<b>V 9.1.4</b> <i>DataPath\autocfg</i>	Otomatik yapılandırma için kullanılır

## IBM i IBM üzerindeki dizin yapısı

A description of the IFS is given, and the IBM MQ IFS directory structure is described for server, client, and Java.

The integrated file system (IFS) is a part of IBM i that supports stream input/output and storage management similar to personal computer, UNIX and Linux operating systems, while providing an integrating structure over all information stored in the server.

IBM i dizin adları, @ (at) karakterinin yerine & (ampersand) karakteriyle başlar. Örneğin, IBM i üzerinde @system, &system olur.

### IBM MQ Server için IFS kök dosya sistemi

When you install IBM MQ Server for IBM i, the following directories are created in the IFS root file system.

ProdData:

#### Genel Bakış

##### QIBM

```
'-- ProdData
    '-- mqm
    '-- doc
    '-- inc
    '-- lib
    '-- samp
    '-- licenses
    '-- LicenseDoc
    '-- 5724H72_V8R0M0
```

#### /QIBM/ProdData/mqm

Aşağıdaki alt dizinler, tüm ürün verilerini (örneğin, C++ sınıfları, izleme biçimi dosyaları ve lisans kütükleri) içerir. Bu dizindeki veriler silinir ve ürünün her kuruşunda değiştirilir.

#### /QIBM/ProdData/mqm/doc

CL komutlarına ilişkin bir Command Reference, burada HTML biçiminde sağlanır ve burada kurulur.

#### /QIBM/ProdData/mqm/inc

C ya da C++ programlarınızı derlemek için kullanılan üstbilgi dosyaları.

#### /QIBM/ProdData/mqm/lib

MQ tarafından kullanılan yardımcı dosyalar.

#### /QIBM/ProdData/mqm/samp

Ek örnekler.

#### /QIBM/ProdData/mqm/licenses

Lisans dosyaları. Her dil için iki dosya, LA\_ *xx* ve LI\_ *xx* gibi adlandırılır; burada *xx* , sağlanan her bir dil için 2 karakterlik dil tanıtıcısıdır.

Ayrıca, aşağıdaki dizin lisans sözleşmeleri dosyalarını depolar:

#### /QIBM/ProdData/LicenseDoc/5724H72\_V8R0M0

Lisans dosyaları. Dosyalar 5724H72\_V8R0M0\_ *xx* gibi adlandırılır; burada *xx* , sağlanan her bir dil için 2 ya da 5 karakterlik dil tanıtıcısıdır.

UserData:

#### Genel Bakış

##### QIBM

```
'-- UserData
    '-- mqm
    '-- errors
```



```
'-- trace
'-- qmgrs
'-- &system
'-- qmgrname1
'-- qmgrname2
'-- and so on
```

### **/QIBM/UserData/mqm**

Aşağıdaki alt dizinler, kuyruk yöneticilerine ilişkin tüm kullanıcı verilerini içerir.

Ürünü kurduğunuzda, /QIBM/UserData/mqm/ dizininde bir mqm.ini kütüğü yaratılır (önceden kuruluştan önceden yoksa).

Kuyruk yöneticisi yarattığınızda, /QIBM/UserData/mqm/qmgrs/ QMGRNAME /dizininde qm.ini kütüğü yaratılır (burada QMGRNAME , kuyruk yöneticisinin adıdır).

Ürün silindiğinde dizinlerdeki veriler korunur.

## **IFS root file system for IBM MQ MQI client**

IBM MQ MQI client for IBM i' u kurduğunuzda, IFS kök dosya sisteminde yaratılan şu dizinler:

ProdData:

### **Genel Bakış**

#### **QIBM**

```
'-- ProdData
'-- mqm
'-- lib
```

### **/QIBM/ProdData/mqm**

Bu dizinin altındaki alt dizinler tüm ürün verilerini içerir. Bu dizindeki veriler silinir ve ürünün her yerine konması değiştirilir.

UserData:

### **Genel Bakış**

#### **QIBM**

```
'-- UserData
'-- mqm
'-- errors
'-- trace
```

### **/QIBM/UserData/mqm**

Bu dizinin altındaki alt dizinler tüm kullanıcı verilerini içerir.

## **IBM MQ Java için IFS kök dosya sistemi**

When you install IBM MQ Java on IBM i, the following directories are created in the IFS root file system:

ProdData:

### **Genel Bakış**

#### **QIBM**

```
'-- ProdData
'-- mqm
'-- java
'-- samples
'-- bin
```

'-- lib

### /QIBM/ProdData/mqm/java

Aşağıdaki alt dizinler, Java sınıfları da içinde olmak üzere tüm ürün verilerini içerir. Bu dizindeki veriler silinir ve ürünün her yerine konması değiştirilir.

### /QIBM/ProdData/mqm/java/samples

Aşağıdaki alt dizinler tüm örnek Java sınıflarını ve verilerini içerir.

## Sunucu ve istemci kuruluşları tarafından yaratılan kitaplıklar

IBM MQ sunucusu ya da istemcisinin kuruluşu aşağıdaki kitaplıkları yaratır:

- QMQM  
Ürün kitablığı.
- QMQMSAMP  
Örnekler kitablığı (örnekleri kurmayı seçerseniz).
- QMxxxx  
Yalnızca sunucu.

Kuyruk yöneticisi her yarattığınızda, IBM MQ otomatik olarak, QMxxxx gibi bir adla ilişkilendirilmiş bir kitaplık yaratır; burada xxxx , kuyruk yöneticisi adından türetilir. Bu kitaplık, günlükler ve ilişkili günlük nesnelere de içinde olmak üzere, kuyruk yöneticisine özgü nesnelere içerir. Varsayılan olarak, bu kitablığın adı, QM karakteriyle önek olarak başına kuyruk yöneticisi adından türetilir. Örneğin, TEST adlı bir kuyruk yöneticisi için, kitablığa QMTEST adı verilir.

**Not:** Bir kuyruk yöneticisi yarattığınızda, istediğiniz kitablığın adını belirtebilirsiniz. Örneğin:

```
CRTMQM MQMNAME(TEST) MQMLIB(TESTLIB)
```

You can use the WRKLIB command to list all the libraries that IBM MQ for IBM i has created. Kuyruk yöneticisi kitablıklarına karşı, QMGR: QMGRNAMEmetnini görürsünüz. Komutun biçimi şöyledir:

```
WRKLIB LIB(QM*)
```

Bu kuyruk yöneticisiyle ilişkili kitablıklar, ürün silindiğinde korunur.

## Multi Çoklu Platformlar üzerinde MFT için planlama dosya sistemi desteği

IBM MQ Managed File Transfer MFT araçları, bir dosya sistemindeki dosyalara/dosyalardan veri aktarmak için kullanılabilir. Buna ek olarak, bir aracı içinde çalışan kaynak izleyicilerinin konfigürasyonu bir dosya sistemindeki dosyaları izleyecek şekilde tanımlanabilir.

MFT ' in bu dosyaların kilitlemeyi destekleyen bir dosya sisteminde saklanması gereksinimi vardır. Bunun iki nedeni vardır:

- Aracı, bir dosyadan veri okumaya ya da dosyaya veri yazmaya başladıktan sonra dosyanın değişmediğinden emin olmak için dosyayı kilitlet.
- Kaynak izleyicileri, başka bir sürecin bunları kullanmadığını denetlemek için dosyaları kilitlet.

Aracılar ve kaynak izleme programları kilitleme gerçekleştirmek için **FileChannel.tryLock()** Java yöntemini kullanır ve dosya sistemi bu çağrıyı kullanarak dosyaları kilitleyebilmelidir.

**Önemli:** MFT' in teknik gereksinimlerini karşılamadıkları için aşağıdaki dosya sistemleri desteklenmez:

- GlusterFS
- NFS sürüm 3

## Çoklu platformlar üzerinde dairesel ya da doğrusal oturum açma seçilmesi

IBM MQ' ta, dairesel ya da doğrusal günlüğe kaydetme seçebilirsiniz. Aşağıdaki bilgiler, her iki tipe ilişkin genel bir bakış sunar.

### Dairesel günlük kaydının avantajları

Dairesel günlüğe kaydetmenin başlıca yararları döngüsel günlüğe kaydetme:

- Yönetmek daha kolay.

İş yükünüz için dairesel günlük kaydını doğru bir şekilde yapılandırdıktan sonra, başka bir yönetim gerekmez. Ancak, doğrusal günlüğe kaydetme için, ortam görüntülerinin kaydedilmesi ve artık gerekli olmayan günlük kapsamlarının arşivlenmesi ya da silinmesi gerekmesi gerekir.

- Daha iyi performans

Çevrimsel günlüğe kaydetme, önceden biçimlendirilmiş günlük kapsamlarını yeniden kullanabildiği için, çevrimsel günlük kaydı doğrusal günlüğe kaydetmekten daha iyi performans gösterir. Oysa doğrusal günlüğe kaydetme, yeni günlük kesimleri ayırmak ve bunları biçimlendirmek zorundadır.

Ek bilgi için [Günlerin yönetilmesi](#) başlıklı konuya bakın.

### Doğrusal günlüğe kaydetme yararları

Doğrusal günlüğe kaydetmenin asıl avantajı, doğrusal günlüğe kaydetmenin daha fazla başarısızlığa karşı koruma sağlamasıdır.

Bozuk ya da silinmiş bir günlüğe ya da uygulamalara ya da yöneticiye göre silinen iletilere ya da kuyruklara karşı, çevrimsel ya da doğrusal günlüğe kaydetme koruması da yoktur.

Doğrusal günlüğe kaydetme (ancak dairesel değil), zarar görmüş nesnelere kurtarılmasını sağlar. Bu nedenle, doğrusal günlüğe kaydetme işlemi, bozuk ya da silinmiş olan kuyruk dosyalarına karşı koruma sağlar; bu hasar görmüş kuyruklar doğrusal bir günlüğünden kurtarılabilir.

Güç kaybı ya da iletişim hatalarının kurtarıldığı başlıklı konu anlatıldığı gibi, güç kaybı ve iletişim hatasına karşı hem çevrimsel hem de doğrusal koruma sağlar.

### Diğer önemli noktalar

Doğrusal ya da dairesel olarak seçerseniz de, ne kadar yedekliliğe gereksinim duymanıza bağlı olarak değişir.

Daha fazla yedeklilik seçmenin maliyeti vardır; bu, performans maliyetinden ve yönetim maliyetinden kaynaklanan doğrusal günlüğe kaydetmedir.

Ek bilgi için [Günlüğe kaydetme tipleri](#) başlıklı konuya bakın.

## AIX üzerinde paylaşılan bellek

Bazı uygulama tipleri AIX bellek sınırlaması nedeniyle bağlanmayı başaramazsa, çoğu durumda bu durum, EXTSHM=ON ortam değişkeni tanımlanarak çözülebilir.

Some 32-bit processes on AIX might encounter an operating system limitation that affects their ability to connect to IBM MQ queue managers. IBM MQ ' a yapılan her standart bağlantı, paylaşılan belleği kullanır; ancak, diğer UNIX and Linux platformlarının aksine, AIX , 32 bit süreçlerin yalnızca 11 paylaşılan bellek kümesini bağlamasını sağlar.

Most 32-bit processes will not encounter this limit, but applications with high memory requirements might fail to connect to IBM MQ with reason code 2102: MQRC\_RESOURCE\_PROBLEM. Aşağıdaki uygulama tipleri bu hatayı görebilir:

- 32 bit Java sanal makinelerinde çalışan programlar

- Büyük ya da çok büyük bellek modellerini kullanan programlar
- Birden çok kuyruk yöneticisine ya da veritabanına bağlanan programlar
- Kendi başına paylaşılan bellek kümelerine bağlanan programlar

AIX , daha fazla paylaşılan bellek eklemelerine olanak sağlayan 32 bitlik işlemler için genişletilmiş bir paylaşılan bellek özelliği sunar. Bu özelliğe sahip bir uygulamayı çalıştırmak için, kuyruk yöneticilerinize ve programınıza başlamadan önce EXTSHM=ON ortam değişkenini dışa aktarın. EXTSHM=ON özelliği, bu hatanın çoğu durumda olmasını önler, ancak shmctl işlevinin SHM\_SIZE seçeneğini kullanan programlarla uyumsuz.

IBM MQ MQI client uygulamaları ve 64 bit tüm süreçler bu sınırlamadan etkilenmez. EXTSHM ' nin ayarlanıp ayarlanmamasından bağımsız olarak IBM MQ kuyruk yöneticilerine bağlanabilirler.

Linux

UNIX

## IBM MQ ve UNIX System V IPC kaynakları

Kuyruk yöneticisi bazı IPC kaynaklarını kullanır. Kullanılan kaynakların ne olduğunu öğrenmek için **ipcs -a** ' i kullanın.

**Bu bilgiler, yalnızca UNIX and Linux sistemlerinde çalışan IBM MQ için geçerlidir.**

IBM MQ , System V işlem arası iletişim (IPC) kaynaklarını kullanır ( *semaforlar* ve *paylaşılan bellek kesimleri* ) Sistem bileşenleri arasında veri aktarmak ve veri aktarmak için. Bu kaynaklar kuyruk yöneticisi işlemleri ve kuyruk yöneticisine bağlanan uygulamalar tarafından kullanılır. IBM MQ MQI clients , IBM MQ izleme denetimi dışında, IPC kaynaklarını kullanmaz. Use the UNIX command **ipcs -a** to get full information on the number and size of the IPC resources currently in use on the machine.

Linux

UNIX

## IBM MQ ve UNIX Process Priority

Süreç önceliği *güzel* değerlerini ayarlarken iyi uygulamalar.

**Bu bilgiler, yalnızca UNIX and Linux sistemlerinde çalışan IBM MQ için geçerlidir.**

Arka planda bir işlem çalıştırırsanız, bu işlem, çağırın kabuk tarafından daha yüksek bir *güzel* değer (ve dolayısıyla daha düşük öncelik) verilebilir. Bu, genel IBM MQ performans etkisinerine sahip olabilir. Yüksek düzeyde stresli durumlarda, daha yüksek bir öncelikte ve daha düşük öncelikte çok sayıda hazır çalıştırma iş parçacığı varsa, işletim sistemi zamanlama özellikleri, daha düşük öncelikli iş parçacıklarını işlemci süresinde mahrum edebilir.

It is good practice that independently started processes associated with queue managers, such as **runmqtsr**, have the same *güzel* values as the queue manager they are associated with. Kabuğun bu arka plan süreçlerine daha yüksek bir *güzel* değer atamadığından emin olun. Örneğin, ksh 'de, arka plan süreçlerinin *güzel* değerini yükseltmekten ksh ' yi durdurmak için "set +o bgnice" ayarını kullanın. Bir "ps -efl" listesinin *NI* sütununu inceleyerek, çalışan süreçlerin *güzel* değerlerini doğrulayabilirsiniz.

Also, start IBM MQ application processes with the same *güzel* value as the queue manager. Farklı *güzel* değerler ile çalıştırılırsa, bir uygulama iş parçacığı kuyruk yöneticisi iş parçacıklarını engelleyebilir ya da tam tersi de performansın düşmesine neden olabilir.

z/OS

## z/OSüzerinde IBM MQ ortamınızın planlanması

IBM MQ ortamınızı planlarken, veri kümeleri, sayfa kümeleri, Db2, Coupling Faciliti'lerinin ve yedekleme olanaklarının ve yedekleme olanaklarının kaynak gereksinmelerini göz önünde bulundurmanız gerekir. Use this topic to plan the environment where IBM MQ runs.

IBM MQ mimarinizi planlamadan önce, kendinizi temel IBM MQ for z/OS kavramlarıyla tanıyın, [IBM MQ for z/OS kavramları](#) ' nda konulara bakın.

Kuyruk yöneticinizi planlarken, kuruluşunuzda farklı kişilerle çalışmanız gerekebilir. Değişiklik kontrol prosedürlerinin uzun sürebileceği gibi, bu kişileri erken saatlerde de dahil etmek genellikle iyi bir fikirdir. Ayrıca, IBM MQ for z/OS' u yapılandırmak için hangi parametrelere ihtiyacınız olduğunu da söyleyebilirler.

Örneğin, aşağıdakilerle çalışmanız gerekebilir:

- Kuyruk yöneticisi veri kümelerinin üst düzey niteleyicisini belirlemek ve kuyruk yöneticisi veri kümeleri için yeterli alan ayırmak için depolama yöneticisi.
- z/OS system programmer to define the IBM MQ subsystem to z/OS and APF authorize the IBM MQ for z/OS libraries.
- IBM MQ for z/OS için hangi TCP/IP yığınının ve kapılarının kullanılacağını belirlemek üzere ağ yöneticisi.
- Kuyruk yöneticisi veri kümelerine, IBM MQ for z/OS kaynaklarına ilişkin güvenlik profillerine ve TLS sertifikalarına erişimi ayarlamaya ilişkin güvenlik yöneticisi.
- Db2 administrator to set up Db2 tables when configuring a queue sharing group.

### İlgili kavramlar

[IBM MQ Teknik genel bakış](#)

### İlgili görevler

[“IBM MQ mimarisinin planlanması” sayfa 5](#)

IBM MQ ortamınızı planlarken, IBM MQ ' un tek ve birden çok kuyruk yöneticisi mimarisi ve noktadan noktaya iletişim ve yayınlama/abone olma ileti sistemi stilleri için sağladığı desteği göz önünde bulundurun. Ayrıca, kaynak gereksinimlerinizi ve günlük kaydı ve yedekleme olanaklarını kullanımınızı da planlayın.

[yapılandırma/z/OS](#)

[Yönetme IBM MQ for z/OS](#)

► z/OS

## Kuyruk yöneticinizin planlanması

Bir kuyruk yöneticisi ayarlarken, kuyruk yöneticisinin kuruluşunuzun gereksinimlerini karşıladığı için, planlama yöneticisi kuyruk yöneticisinin büyümesine izin vermelidir.

Kuyruk yöneticisi yapılandırmanın en iyi yolu adımlarda:

1. Temel kuyruk yöneticisini yapılandır
2. Kuyruk yöneticisi iletişimini kuyruk yöneticisi yapan kanal başlatıcısını ve uzak istemci uygulaması iletişimini yapılandırın
3. If you want to encrypt and protect messages, configure [Advanced Message Security](#)
4. If you want to use File Transfer over IBM MQ, configure [z/OS için Managed File Transfer](#).
5. If you want to use the administrative or messaging REST API, or the MQ Console to manage IBM MQ from a web browser, configure the mqweb server.

Bazı kuruluşlarda, çevrelerinde binlerce kuyruk yöneticisi vardır. IBM MQ ağını şimdi göz önünde bulundurmanız ve beş yıl içinde göz önünde bulundurmanız gerekir.

z/OS'ta, bazı kuyruk yöneticileri saniyede binlerce iletiyi işlerler ve saniyede 100 MB' nin üzerinde oturum açardır. Çok yüksek birim olmasını bekliyorsanız, birden çok kuyruk yöneticisinden birini bulundurmanız gerekebilir.

On z/OS, IBM MQ can run as part of a queue sharing group (QSG) where messages are stored in the Coupling Facility, and any queue manager in the queue sharing group can access the messages. Bir kuyruk paylaşım grubunda çalışmak istiyorsanız, kaç kuyruk yöneticisi gerektiğini göz önünde bulundurmanız gerekir. Genellikle, her bir LPAR için bir kuyruk yöneticisi vardır. CF yapılarını düzenli olarak yedeklemek için bir kuyruk yöneticiniz de olabilir.

Yapılanışlarda yapılan bazı değişiklikler, yeni bir kuyruk tanımlama gibi kolay bir şekilde yapılır. Günlükler ve sayfa kümeleri daha büyük olması gibi bazıları daha zordur ve bir kuyruk yöneticisinin adı ya da kuyruk paylaşım grubu adı gibi bazı yapılandırma değiştirilemez.

[MP16 performansı SupportPac ' ta kullanılabilir performans ve ayarlama bilgileri vardır.](#)

### Adlandırma kuralları

Kuyruk yöneticisi veri kümeleri için bir adlandırma kuralına sahip olması gerekir.

Birçok kuruluş, yükleme kitaplıkları adı altında yayın numarasını kullanır ve bu şekilde devam eder. You might want to consider having an alias of MQM . SCSQAUTH pointing to the version currently in use, such as MQM . V900 . SCSQAUTH, so you do not have to change CICS, Batch, and IMS JCL when you migrate to a new version of IBM MQ.

Şu anda kullanılmakta olan IBM MQ sürümüne ilişkin kuruluş dizinine başvurmak için UNIX System Services olanağında simgesel bir bağlantı kullanabilirsiniz.

Kuyruk yöneticisi tarafından kullanılan veri kümelerinin (günlükler, sayfa kümeleri, JCL kitaplıkları), güvenlik profillerinin oluşturulmasını basitleştirmek için bir adlandırma kuralına ve veri kümelerinin disk üzerine hangi veri kümelerinin yerleştirildiğini ve sahip oldukları öznitelikleri denetleyen SMS depolama sınıflarına eşlemesini sağlar.

IBM MQ sürümünü sayfa kümelerinin ya da günlüklerin adına yerleştirmenin iyi bir fikir olmadığını unutmayın. Bir gün yeni bir sürüme geçilebilir ve veri kümesi "yanlış" adlara sahip olacaktır.

## Uygulamalar

You need to understand the business applications and the best way to configure IBM MQ. Örneğin, uygulamaların kurtarma ve yineleme yeteneği sağlama mantığı varsa, kalıcı olmayan iletiler yeterince iyi olabilir. IBM MQ ' in kurtarma işlemini işlemesini istiyorsanız, kalıcı iletileri kullanmanız ve eşitleme noktasına ileti koymak ve iletileri almak için kullanmanız gerekir.

Farklı iş hareketlerinden kuyrukları yalıtmanız gerekir. Bir iş uygulamasına ilişkin bir kuyruk doldurulsa, bu uygulamanın diğer iş uygulamalarını etkilmesini istemezseniz, bu işlemi gerçekleştirmeyin. Olanaklıysa, farklı sayfa kümelerindeki ve arabellek havuzlarındaki kuyrukları ya da yapıları yalıtın.

İletilerin profilini anlamanız gerekir. Birçok uygulama için kuyrukların yalnızca birkaç iletilisi vardır. Diğer uygulamalar, gün içinde kuyruklar oluşturabilir ve bir gece boyunca işlenebilir. Olağan durumda yalnızca birkaç iletiye sahip olan bir kuyruğun, bir sorun varsa ve iletiler işlenmezse, birçok saat değerinde ileti tutması gerekebilir. Beklenen en yüksek kapasitenize izin vermek için CF yapılarını ve sayfa kümelerini boyutlamanız gerekir.

## Yapılandırma sonrası

Kuyruk yöneticinizi (ve bileşenlerinizi) yapılandırdıktan sonra aşağıdakini planlamanız gerekir:

- Sayfa kümeleri yedekleniyor.
- Nesne tanımlamaları yedekleniyor.
- CF yapılarının yedeğini otomatikleştirme.
- IBM MQ iletilerini izleme ve bir sorun algılandığında harekete geçme.
- IBM MQ istatistik verileri toplanıyor.
- Sanal saklama alanı gibi kaynak kullanımını ve günlüğe kaydedilen veri miktarını izleme. Bununla birlikte, kaynak kullanımınızın arttığını ve yeni bir kuyruk yöneticisi kurma gibi işlemleri de üstlenmeniz gerektiğini görebilirsiniz.

## ► z/OS Planning your storage and performance requirements on z/OS

IBM MQ sisteminiz için gerçekçi ve ulaşılabilir depolama ve performans hedefleri ayarlamanız gerekir. Bu konuyu, depolamayı ve performansı etkileyen faktörleri anlamanıza yardımcı olur.

Bu konuda, IBM MQ for z/OS depolama ve performans gereksinimleriyle ilgili bilgiler yer almaktadır. Bu belge aşağıdaki bölümleri içerir:

- [IBM MQ için z/OS performans seçenekleri](#)
- [z/OS iş yükü yönetimi önem ve hız hedeflerinin belirlenmesi](#)
- [“Kitaplık depolama alanı” sayfa 159](#)
- [“System LX kullanımı” sayfa 159](#)
- [“Adres alanı depolama alanı” sayfa 160](#)

- “Disk depolama alanı” sayfa 164

Daha fazla bilgi için bkz. “Depolama ve performans gereksinimleriyle ilgili daha fazla bilgi bulabileceğiniz yerler” sayfa 164 .

## IBM MQ için/OS performans seçenekleri

İş yükü yönetimiyle, performans hedeflerini tanımlayabilir ve her bir hedefe bir iş önemi atayabilirsiniz. İş terimlerindeki iş hedeflerini tanımlırsınız ve amacı, işlemci ve saklama alanı gibi bir kaynağın amacına uygun olarak ne kadar kaynak göstereceğine karar verir. İş yükü yönetimi, sağladığınız hedeflere göre dağıtım öncelik tanımayı denetler. İş yükü yönetimi, belirtilen hedefi karşılamak için gerektiği şekilde önceliği yükseltir ya da düşürür. Bu nedenle, sistemdeki her çalışma parçasının tam önceliklerini ayarlamamanız ve iş hedefleri yerine odaklanabilmenize gerek yoktur.

Üç tür hedefler şunlardır:

### Yanıt süresi

İşin ne kadar hızlı işlenmesini istiyorsunuz

### Yürütme hızı

İşlemci, depolama, G/Ç erişimi ve kuyruk gecikmesi için gecikmeden, hazır olduğunda işin ne kadar hızlı çalıştırılması gerektiği

### kullanıcı tarafından oluşturulan

Performans hedefleri olmayan düşük öncelikli işler için bir kategori

Yanıt süresi hedefleri son kullanıcı uygulamaları için uygundur. Örneğin, CICS kullanıcıları, iş yükü hedeflerini yanıt süresi hedefleri olarak ayarlayabilir. IBM MQ adres alanları için hız hedefleri daha uygundur. Kuyruk yöneticisinde yapılan işin küçük bir kısmı bu hız hedefine doğru sayıldı, ancak bu iş performans açısından kritik önem derecesinde. Kuyruk yöneticisi tarafından yapılan çalışmaların çoğu, son kullanıcı uygulamasının performans hedefine doğru sayılıyor. Kanal başlatıcı adres alanı tarafından yapılan çalışmaların çoğu kendi hız hedefine doğru sayılıyor. Kanal başlatıcı tarafından yapılan IBM MQ iletilerinin alınması ve gönderilmesi genellikle, bu iletileri kullanan iş uygulamalarının başarımı açısından önemlidir.

## z/OS iş yükü yönetimi önem ve hız hedeflerinin belirlenmesi

Ek bilgi için “z/OS iş yükü yönetimi öneminin belirlenmesi” sayfa 160 başlıklı konuya bakın.

## Kitaplık depolama alanı

Ürün kitaplıkları için disk saklama alanı ayırmalısınız. Tam rakamlar, yapılandırmanızı temel alıyor, ancak dağıtım kitaplıklarının gerektirdiği alan tahmini değeri 700 MB 'dir. Hedef kitaplıklar yaklaşık 200 MB gerektirir. Ek olarak, SMP/E kitaplıkları için yer gereklidir.

IBM MQ for z/OS tarafından kullanılan hedef kitaplıklar, PDS ya da PDSE biçimlerini kullanır. Herhangi bir PDSE hedef kitaplıklarının bir sistem birleşimi (sysplex) dışında paylaşılmadığından emin olun. Gerekli kitaplıklara ve bunların büyüklüklerine ve gereken biçime ilişkin ek bilgi için Program Dizinine bakın. IBM MQ for z/OS Program Dizini, IBM Publications Center adresinden yüklenebilir (bkz. [IBM MQ for z/OS Program Dizini PDF dosyaları](#)).

## System LX kullanımı

Tanımlanan her IBM MQ altsistemi, IPL saatinde bir sistem bağlantısı dizinini (LX) ayırır ve kuyruk yöneticisi başlatıldığında sistem dışı bağlantı dizinlerinin sayısını ayırır. Kuyruk yöneticisi durdurulduğunda ve yeniden başlatılırsa, sistem bağ dizini yeniden kullanılır. Benzer şekilde, dağıtılmış kuyruklama, sistem dışı bir bağ olmayan bir dizin ayırır. Sisteminizde yetersiz sistem LX'leriniz olan z/OS sisteminizde beklenmedik bir durumda, bu ayrılmış sistem LX'lerini dikkate almanıza gerek vardır.

Gerekliyse, SYS1.PARMLIB parametresi IEASYSxx içindeki NSYSLX parametresi ayarlanarak sistem LX sayısı artırılabilir.

## z/OS z/OS iş yükü yönetimi öneminin belirlenmesi

İş yükü yönetimi ve hizmet tanımlaması aracılığıyla hedefler tanımlama hakkında tam bilgi edinmek için [z/OS MVS Planning: Workload Management](#) başlıklı konuya bakın.

This topic suggests how to set the z/OS workload management importance and velocity goals relative to other important work in your system.

Kuyruk yöneticisi adres alanının altsistem hizmetleri sağladıkça yüksek önceliğe sahip olarak tanımlanması gerekir. Kanal başlatıcı bir uygulama adresi alanıdır, ancak genellikle uzak bir kuyruk yöneticisine gönderilen iletilerin gecikmemesi için yüksek önceliğe sahip olur. Advanced Message Security (AMS), altsistem hizmetlerini de sağlar ve yüksek önceliğe sahip olarak tanımlanması gerekir.

Aşağıdaki hizmet sınıflarını kullan:

### Varsayılan SYSSTC hizmet sınıfı

- VTAM ve TCP/IP adres alanları
- IRLM adres alanı (IRLMPROC)

**Not:** VTAM, TCP/IP ve IRLM adres alanları, tüm DBMS adres alanlarının, bağlı adres alanlarının ve bunların alt adres alanlarının daha yüksek bir dağıtım önceliğine sahip olmalıdır. İş yükü yönetimine, diğer DBMS adres alanlarından VTAM, TCP/IP ya da IRLM 'nin (ya da altında) önceliğini azaltmasına izin verme

### PRODREGN gibi tanımladığınız ada sahip bir hizmet sınıfı için 1 'in yüksek hızlı hedefi ve önemi aşağıdaki gibidir:

- IBM MQ kuyruk yöneticisi, kanal başlatıcı ve AMS adres alanları
- Db2 ( Db2-kurulu saklanmış yordamlar adres alanı dışında tüm adres alanları)
- CICS (tüm bölge tipleri)
- IMS (BMP dışındaki tüm bölge tipleri)

Yüksek bir hız hedefi, tüm bu adres alanları için yeni kurulan ve yeniden başlatma işlemlerinin mümkün olan en hızlı şekilde gerçekleştirilmesini sağlamak için iyidir.

CICS ve IMS bölgeleri için hız hedefleri, yalnızca başlatma ya da yeniden başlatma sırasında önemlidir. İşlemler çalışmaya başladığında, iş yükü yönetimi CICS ya da IMS hız hedeflerini yoksayar ve bölgelerde çalışmakta olan hareketlerin yanıt süresi hedeflerine dayalı olarak öncelikler atar. Bu işlem hedefleri, uyguladıkları iş uygulamalarının görece önceliğini yansıtmalıdır. Bunlar genellikle 2 'nin önem değerine sahip olabilir. IBM MQ kullanan her türlü toplu iş uygulaması, uyguladıkları iş uygulamalarının görece önceliğini yansıtan hız hedeflerine ve önemine benzer şekilde olmalıdır. Genellikle önem ve hız hedefleri, PRODREGN için bu hedeflerden daha az olacaktır.

## z/OS Adres alanı depolama alanı

IBM MQ bileşenlerine ilişkin adres alanı gereksinimleriyle ilgili temel kılavuz bilgileri için bu konuyu kullanın.

Depolama gereksinimleri aşağıdaki kategorilere bölünebilir:

- Ortak depolama
- [Kuyruk yöneticisi özel bölge depolama kullanımı](#)
- [Kanal başlatıcısı depolama kullanımı](#)

64 bit adres alanında, 2GB adresini işaretleyen "çubuk" adı verilen sanal bir hat vardır. The bar separates storage below the 2GB address, called "barın altında", from storage above the 2GB address, called "çubuğun üstünde". Çubuğun altındaki depolama 31 bit adreslenebilirliği kullanır, çubuğun üstünde depolama 64 bit adreslenebilirliği kullanır.

JCL üzerindeki REGORON parametresini kullanarak 31 bit depolama sınırını ve MEMLIMITIT parametresini kullanarak çubuk depolama alanının üst sınırını belirleyebilirsiniz. Bu belirtilen değerler MVS çıkışlarıyla geçersiz kılınabilir.





**Uyarı:** Sistem çalışmalarının nasıl tanıtıldığı ile ilgili bir değişiklik. Şimdi, Cross-system Extended Services (XES), her bir bağlantı için yüksek sanal depolama alanına 4GB depolama alanı ayırır ya da bir kilit yapısına her bağlantı için 36GB ' lik bir depolama alanı ayırır.

Bu değişiklik öncesinde, bu saklama alanı veri alanlarında ayrılmıştı. Bu APAR 'ın uygulanmasından sonra, IBM MQ ' in depolama kullanımını hesaplarına dayalı olarak, [CSQY225E](#) ve [CSQY224I](#) iletileri yayınlanabilir, bu da Kuyruk yöneticisi, çubuğun üstündeki yerel depolamanın kısaltırifadesini belirtir.

You will also see an increase to the above bar values in message [CSQY220I](#)

Daha fazla bilgi için IBM destek belgesine [2017139](#) bakın.

## Önerilen bölge boyutları

Aşağıdaki tabloda, bölge boyutlarına ilişkin önerilen değerler gösterilmektedir.

Çizelge 19. JCL bölgesi büyüklükleri için önerilen tanımlamalar	
Tanımlama ayarı	Sistem
Kuyruk yöneticisi	REGION=0M, MEMLIMIT=3G
Kanal başlatıcı	REGION=0M

## Ortak depolama

Her IBM MQ for z/OS altsistemi, aşağıdaki yaklaşık depolama gereksinimlerine sahiptir:

- CSA 4KB
- ECSA 800KB, artı CSQ6SYSP sistem parametre makrosu TRACTBL parametresinde belirlenen izleme çizelgesinin büyüklüğü. Daha fazla bilgi için bakınız: [Using CSQ6SYSP](#).

Ayrıca, her eşzamanlı IBM MQ mantıksal bağlantısı için 5 KB ' lik ECSA gerekir. Bir görev sona erdiğinde, diğer IBM MQ görevleri bu depolamayı yeniden kullanabilir. IBM MQ does not release the storage until the queue manager is shut down, so you can calculate the maximum amount of ECSA required by multiplying the maximum number of concurrent logical connections by 5KB. Koşut zamanlı mantıksal bağlantı sayısı, aşağıdaki sayıların toplamını içerir:

- Küme, TSO, z/OS UNIX and Linux System Services, IMSve Db2 saklanmış yordam adres alanı (SPAS) bölgelerindeki görevler (TCB 'ler), IBM MQ' a bağlı, ancak bağlantısı kesilmemiş olan bölgelerdir.
- IBM MQ isteği veren, ancak sonlandırılmamış CICS işlemleri
- Oluşturulan JMS Connections, Oturumlar, TopicSessions ya da QueueSessions (bağ tanımları bağlantısı için), ancak henüz yok edilmemiş ya da atık toplama işlemi değil.
- Etkin IBM MQ kanalları.

Kuyruk yöneticisiyle mantıksal bağlantılar tarafından kullanılan ortak saklama alanı için ACELIM yapılış değişikliğiyle bir sınır ayarlayabilirsiniz. ACELIM denetimi, öncelikli olarak Db2 saklanmış yordamlarının IBM MQ kuyrukları üzerindeki işlemlere neden olduğu sitelere ilgi göstermektedir.

Bir saklanmış yordamdan çalıştırıldığında, her bir IBM MQ işlemi kuyruk yöneticisinde yeni bir mantıksal bağlantı ile sonuçlanabilir. Büyük Db2 çalışma birimleri (örneğin, çizelge yükü nedeniyle), sık kullanılan depolama için aşırı talebe neden olabilir.

ACELIM, sistemdeki bağlantı sayısını sınırlayarak, ortak depolama kullanımını sınırlandırmak ve z/OS sistemini korumak için tasarlanmıştır. Bu yalnızca, aşırı miktarda ECSA depolaması kullanan kuyruk yöneticilerindeki ayarlanmalıdır. Daha fazla bilgi için [CSQ6SYSPkomutunu kullanma](#) içindeki [ACELIM](#) bölümüne bakın.

ACELIM için bir değer ayarlamak üzere, öncelikle ACELIM değeri tarafından denetlenen alt havuzdaki depolama miktarını belirleyin. Bu bilgiler, STATISTICS CLASS (3) izleme tarafından üretilen SMF 115 alt tipi 5 kayıtlarında yer alıyor.

IBM MQ SMF verileri [SupportPac MP1B](#) kullanılarak biçimlendirilebilir. ACELIM tarafından denetlenen alt havuzda kullanılan bayt sayısı, *ATE/PEB* başlıklı satırda STGPOOL DD ' de görüntülenir.

SMF 115 istatistik kayıtlarıyla ilgili ek bilgi için [IBM MQ performans istatistiklerini yorumlamaya](#) başlıklı konuya bakın.

Büyüme ve iş yükü çivileri için alan sağlamak için, normal değeri yeterli bir kenar boşluğuna göre artırın. Yeni değeri, ACELIM yapılandırmasında kullanmak üzere KB cinsinden maksimum depolama boyutu elde etmek için 1024 ile bölün.

Kanal başlatıcı genellikle, 160KB' a kadar ECSA kullanımını gerektirir.

## Kuyruk yöneticisi özel bölge depolama kullanımı

IBM MQ for z/OS , bazı iç denetim blokları için 2GB çubuğunun üstünde depolamayı kullanabilir.

Bu depolama alanında arabellek havuzlarına sahip olabilirsiniz; bu, yeterli depolama alanı varsa, çok daha büyük arabellek havuzlarını yapılandırma potansiyeliniz sağlar. Genellikle arabellek havuzları, 2GB çubuğunun üstünde depolamayı kullanan ana iç denetim bloklarıdır.

Her arabellek havuzu büyüklüğü kuyruk yöneticisi kullanıma hazırlama sırasında saptanır ve arabellek havuzu bu arabellek havuzunu kullanan bir sayfa kümesi bağlandığında arabellek havuzu için saklama alanı ayrılır. Arabelleklerin nerede ayrılacağı belirlemek için yeni bir LOCATION (YUKART | ALT) parametresi kullanılır. Arabellek havuzlarının büyüklüğünü devingen olarak değiştirmek için [ALTER BUFPOOL](#) komutunu kullanabilirsiniz.

Çubuk (64 bit) depolama alanı üzerinde kullanmak için, MEMLIMIT değiştirgesi için (örneğin, MEMLIMIT=3G), kuyruk yöneticisi JCL ' deki **EXEC PGM=CSQYASCP** değiştirgesinde bir değer belirtebilirsiniz. Kuruluşunuzda varsayılan bir değer ayarlanmış olabilir.

Bir MEMLIMIT belirtmeli ve olası sorunları önlemek için MEMLIMIT = NOLIMIT yerine duyarlı bir depolama büyüklüğü belirtmelisiniz. NOLIMIT ya da çok büyük bir değer belirlerseniz, büyük boyutlu bir ALTER BUFPOOL komutu, tüm kullanılabilir z/OS sanal saklama alanını kullanabilir ve bu da sisteminizde sayfalamaya neden olur. Kullanılabilecek saklama yeri miktarında sistem çapında bir sınır olması durumunda, MEMLIMIT değerini z/OS sistem programcısı ile birlikte tartışmanız gerekebilir.

Bir MEMLIMIT=3G ile başlayın ve arabellek havuzlarınızın büyüklüğünü artırmanız gerektiğinde bu büyüklüğü artırın.

MEMLIMIT değerini 2GB olarak hesaplatın ve çubuğun üst kısmındaki arabellek havuzlarının büyüklüğünü, en yakın GB ' ye yuvarlayın. MEMLIMIT değerini en az 3GB değerine ayarlayın ve arabellek havuzlarınızın büyüklüğünü artırmanız gerektiğinde bu değeri artırın.

Örneğin, YUKARTION LOCATION ile yapılandırılan 2 arabellek havuzu için, arabellek havuzu 1 'in 10.000 arabelleği var, arabellek havuzu 2 50.000 arabelleğe sahiptir. Çubuktan yukarıda bellek kullanımı 60 bin (toplam arabellek sayısı) \* 4096 = 245,760.000 bayt = 234.375MB' dir. LOCATION ne olursa olsun tüm arabellek havuzları, denetim yapıları için 64 bit depolamanın kullanılmasını sağlar. Bu havuzlardaki arabellek havuzları ve arabelleklerin sayısı arttıkça, bu değer önemli olabilir. Her arabellek için ek 200 bayt 64 bit depolama alanı gerekir. 10 arabellek içeren bir yapılandırma için, her biri 20.000 arabellek gerektirir: 200 \* 10 \* 20,000 = 40,000,000, 40MB değerine eşit. You can specify 3GB for the MEMLIMIT size, which will allow scope for growth (40MB + 200MB + 2GB which rounds up to 3GB).

Bazı yapılanışlarda, arabelleklerinin gerçek saklama alanı tarafından kalıcı olarak desteklendiği arabellek havuzlarının kullanılması önemli bir başarımlar yararı olabilir. Bu değeri, arabellek havuzunun PAGECLAS özneliği için FIXED4KB değerini belirterek elde edebilirsiniz. Ancak, bu işlemi yalnızca, LPAR üzerinde yeterli miktarda gerçek saklama alanı varsa yapmanız gerekir; tersi durumda, diğer adres alanları etkilenebilir. PAGECLAS için FIXED4KB değerini ne zaman kullanmanız gerektiği hakkında bilgi için bkz. IBM MQ Support Pac MP16: [IBM MQ for z/OS -Capacity planning & tuning](#)

Çubuğun üst kısmındaki depolama alanını kullanmadan önce, en yoğun zaman kullanımı için yeterli yardımcı depolama ve sayfalamayı önlemek için yeterli sayıda gerçek depolama gereksinmesi olduğundan emin olmak için z/OS sistem programcınızla görüşmelisiniz.

**Not:** Artırılan sanal saklama alanını işlemek için bellek dökümü veri kümelerinin büyüklüğü artırılmış olabilir.

Arabellek havuzlarının, MVS sayfalama olması başarımı olumsuz yönde etkileyebileceğinden çok büyük bir değer sağlar. You might consider using a smaller buffer pool that does not page, with IBM MQ moving the message to and from the page set.

You can monitor the address space storage usage from the CSQY220I message that indicates the amount of private region storage in use above and below the 2GB bar, and the remaining amount.

## Kanal başlatıcı depolama kullanımı

Aşağıdakileri göz önünde bulundurmanız gereken iki kanal başlatıcı depolama kullanımı alanı vardır:

- Özel bölge
- Muhasebe ve istatistikler

### Özel bölge depolama kullanımı

CHINIT için REGION=0M , çubuk saklama alanının alt üst sınırını kullanmasına izin verecek şekilde belirtmelisiniz. Kanal başlatıcısının kullanabileceği saklama alanı, CHINIT ' in sahip olduğu koşut zamanlı bağlantı sayısını sınırlar.

Her kanal, kanal başlatıcı adres alanında yaklaşık 170KB uzatılmış özel bölge kullanır. 32KB ' den büyük iletiler iletilirse, depolama ileti boyutuna göre artırılır. Bu artırılmış depolama, aşağıdaki durumlarda serbest bırakılır:

- Bir gönderme ya da istemci kanalı, art arda 10 ileti için geçerli arabellek büyüklüğünün yarıdan az olmasını gerektirir.
- Bir sağlıklı işletim bildirim gönderilir ya da gönderilir.

Depolama, Dil Ortamı içinde yeniden kullanım için serbest bırakılır; ancak, z/OS sanal depolama yöneticisi tarafından ücretsiz olarak görülmez. Diğer bir deyişle, kanal sayısı için üst sınır, ileti büyüklüğü ve geliş örüntülerine ve genişletilmiş özel bölge boyutundaki tek tek kullanıcı sistemlerine ilişkin sınırlamalara bağlıdır. The upper limit on the number of channels is likely to be approximately 9000 on many systems because the extended region size is unlikely to exceed 1.6GB. The use of message sizes larger than 32KB reduces the maximum number of channels in the system. For example, if messages that are 100MB long are transmitted, and an extended region size of 1.6GB is assumed, the maximum number of channels is 15.

Kanal başlatıcı izlemesi bir veri alanına yazılıdır. Veri alanı depolamanın boyutu, **TRAXTBL** parametresiyle denetlenir. Bkz. [ALTER QMGR](#).

### Muhasebe ve istatistik depolama kullanımı

Kanal başlatıcı erişiminin, çubuğun üzerinde en az 256MB sanal depolama alanına erişmesine izin vermelisiniz. Bunu MEMLIMIT=256M belirterek yapabilirsiniz.

If you do not set the MEMLIMIT parameter in the channel initiator JCL, you can set the amount of virtual storage above the bar using the MEMLIMIT parameter in the SMFPRMxx member of SYS1.PARMLIB, or from the IEFUSI exit.

MEMLIMIT değeri, yukarıdaki çubuk depolamayı gerekli düzeyin altında sınırlamak için ayarlandıysa, kanal başlatıcı [CSQX124E](#) ileti ve sınıf 4 muhasebe ve istatistik izleme bilgilerini kullanılamaz.

## MEMLIMIT ve REGION boyutlarının yönetilmesi

Diğer mekanizmalar; örneğin, SYS1.PARMLIB ' un SMFPRMxx üyesindeki **MEMLIMIT** parametresi ya da IEFUSI çıkışı, z/OS adres alanları için çubuğun üst kısmında varsayılan bir sanal saklama alanı sağlamak üzere kuruluşunuzda kullanılabilir. Çubuğunun üst kısmındaki depolama alanının sınırlanmasına ilişkin ayrıntılı bilgi için [çubuğun üzerindeki bellek yönetimi](#) başlıklı konuya bakın.

## **z/OS Disk depolama alanı**

Günlük veri kümeleri, Db2 depolama, bağlaşım tesisi depolaması ve sayfa veri kümeleri için disk depolama gereksinimlerinizi planlarken bu konuyu kullanın.

Kuyruk yöneticisi veri kümelerini nereye koyaca şınızı belirlemek için depolama denetimcinizle birlikte çalışın. Örneğin, depolama yöneticiniz farklı veri kümesi tipleri için size belirli DASD birimleri ya da SMS depolama sınıfları, veri sınıfları ve yönetim sınıfları verebilir.

- Günlük veri kümelerinin DASD üzerinde olması gerekir. Bu günlükler, küçük bir yanıt süresiyle yüksek G/Ç etkinliğine sahip olabilir ve yedeklenmeleri gerekmez.
- Arşiv günlükleri DASD ya da manyetik bant üzerinde olabilir. Yaratıldıktan sonra, bir yedekleme işleminden sayfa kümesinin kurtarılması gibi olağandışı bir durum dışında, bir daha okunmayabilirler. Uzun bir alıkoyma süresiye sahip olmalıdı.
- Sayfa kümeleri orta düzey etkinliğe sahip olabilir ve düzenli olarak yedeklenmelidir. Yüksek bir kullanım sisteminde günde iki kez yedeklenmeleri gerekir.
- BSDS veri kümeleri her gün yedeklenmelidir; yüksek G/Ç etkinliğine sahip değillerdir.

Tüm veri kümeleri Db2tarafından kullanılanlarla benzerdir ve benzer bakım yordamları IBM MQiçin kullanılabilir.

Veri depolamanızı planlamaya ilişkin ayrıntılar için aşağıdaki bölümlere bakın:

### • **Günlükler ve arşiv depolaması**

“Arşiv günlüklerini ne kadar süre tutmam gerekiyor?” sayfa 181 , IBM MQ sisteminizin işleyeceği iletilerin hacmindeki ve etkin günlüklerin arşiv veri kümelerine ne kadar sıklıkla yüklendiğine bağlı olarak, etkin günlük ve arşiv veri kümelerinizin ne kadar depolama gerektirdiğini nasıl belirleyeceğini açıklar.

### • **Db2 depolama alanı**

“Db2 depolama” sayfa 198 , IBM MQ verileri için Db2 depolama alanının ne kadar depolama gerektirdiğini nasıl belirleyeceğini açıklar.

### • **bağlaşım olanağı depolama alanı**

“Bağlaşım olanağı kaynaklarının tanımlanması” sayfa 189 , bağlaşım tesisi yapılarınızın nasıl belirleneceğini nasıl belirleyeceğini açıklar.

### • **Sayfa kümesi ve ileti saklama alanı**

“Sayfa kümelerinizin ve arabellek havuzlarınızın planlanması” sayfa 165 , başvuru yaptığınız iletilerin boyutlarına, bu iletilerin rakamlarına ve bunların oluşturulma ya da değiştirilme hızına bağlı olarak, sayfa veri kümelerinizin ne kadar depolama gerektirdiğini nasıl belirleyeceğini açıklar.

## **z/OS Depolama ve performans gereksinimleriyle ilgili daha fazla bilgi bulabileceğiniz yerler**

Depolama ve performans gereksinimleriyle ilgili daha fazla bilgi bulmak için bu konuyu başvuru olarak kullanın.

Aşağıdaki kaynaklardan daha fazla bilgi bulabilirsiniz:

Çizelge 20. Depolama gereksinimleriyle ilgili daha fazla bilgi bulabileceğiniz yerler	
Konu	Nereye bakılacağı
Sistem değıştirmeleri	<a href="#">CSQ6SYSP</a> ' nin ve <a href="#">Kuyruk yöneticilerinizin uyarlanması</a>

Çizelge 20. Depolama gereksinimleriyle ilgili daha fazla bilgi bulabileceğiniz yerler (devamı var)

Konu	Nereye bakılacağı
IBM MQürününü kurmak için gereken depolama alanı	Program Dizini. IBM MQ for z/OS Program Dizini, <a href="#">IBM Publications Center</a> adresinden yüklenebilir (bkz. <a href="#">IBM MQ for z/OS Program Dizini PDF dosyaları</a> ).
IEALIMIT ve IEFUSI çıktıları	<i>MVS Kuruluş Çıktıları</i> , available from the zSeries website <a href="#">z/OS Internet Library</a> .
En son bilgiler	IBM MQ SupportPac Web sitesi <a href="#">SupportPacs for IBM MQ ve diğer proje alanları</a> .
Hizmet tanımı aracılığıyla iş yükü yönetimi ve hedefleri tanımlama	<i>z/OS MVS Planning: Workload Management</i>

## z/OS Sayfa kümelerinizin ve arabellek havuzlarınızın planlanması

İlk sayıyı planlama konusunda size yardımcı olacak bilgiler ve sayfa veri kümelerinizin büyüklükleri ve arabellek havuzları.

Bu konu aşağıdaki bölümleri içerir:

- [“Sayfa kümelerinizi planlayın” sayfa 165](#)
  - [Sayfa kümesi kullanımı](#)
  - [Sayfa kümelerinin sayısı](#)
  - [Sayfa kümelerinin boyutu](#)
  - [V9.1.4 z/OS veri kümesi şifrelemesi planlaması](#)
- [“Sayfa kümelerinizin boyutunu hesapla” sayfa 166](#)
  - [Sayfa kümesi sıfır](#)
  - [Sayfa kümesi 01-99](#)
  - [İletiler için depolama gereksiniminin hesaplanması](#)
- [“Dinamik sayfa kümesi genişletmesinin etkinleştirilmesi” sayfa 168](#)
- [“Arabellek Havuzlarınızın Tanımlanması” sayfa 170](#)

## Sayfa kümelerinizi planlayın

### Sayfa kümesi kullanımı

Kısa ömürlü iletiler için, sayfa kümesinde genellikle birkaç sayfa kullanılır ve başlatma sırasında, denetim noktası sırasında ya da kapanma dışında veri kümelerinde G/Ç ya da G/Ç yoktur.

Uzun ömürlü iletiler için, ileti içeren sayfalar genellikle diske yazılır. Bu işlem, kuyruk yöneticisi tarafından yeniden başlatma süresini kısaltabilmek için gerçekleştirilir.

Uzun ömürlü iletilerden kısa ömürlü iletileri, farklı sayfa kümeleri ve farklı arabellek havuzlarına yerleştirerek ayırın.

### Sayfa kümelerinin sayısı

Çok sayıda büyük sayfa kümesi kullanılması, daha az sayfa kümesine gereksinim dumanız, kuyrukların sayfa kümelerine eşlenmesini daha basit hale getirmek anlamına geldiği için IBM MQ yöneticisinin rolünü daha kolay hale getirebilmesini sağlar.

Çok sayıda, daha küçük sayfa kümelerinin kullanılması bir dizi avantajlara sahiptir. Örneğin, yedeklemek için daha az zaman alır ve yedekleme ve yeniden başlatma işlemleri sırasında G/Ç koşut olarak gerçekleştirilebilir. Ancak, bunun IBM MQ yöneticisinin rolüne önemli bir performans maliyeti

eklediğini göz önünde bulundurun. Bu, her bir kuyruğu çok daha fazla sayıda sayfa kümesinin biriyle eşlenmesi gerekir.

En az beş sayfa kümesi tanımlayın:

- Nesne tanımlamaları için ayrılmış bir sayfa kümesi (sayfa kümesi sıfır)
- Sistem ile ilgili iletiler için bir sayfa kümesi
- Performans açısından kritik uzun ömürlü iletiler için bir sayfa kümesi
- Performans açısından kritik kısa ömürlü iletiler için bir sayfa kümesi
- Tüm diğer iletiler için bir sayfa kümesi

“Arabellek Havuzlarınızın Tanımlanması” sayfa 170 , iletilerinizi sayfa kümeleri üzerinde bu şekilde dağıtmanın performans avantajlarını açıklar.

### Sayfa kümelerinin boyutu

Sayfa kümelerinizde, beklenen en yüksek ileti kapasitesi için yeterli yer tanımlayın. Bir kuyruk sunucusu programı çalışmadığı için iletilerin bir oluşturma işlemi geliştiğinde olduğu gibi, beklenmeyen en yüksek kapasite için göz önünde bulundurun. Bunu, dinamik sayfa kümesi genişletmeyi etkinleştirerek, ikincil kapsamlar ya da diğer bir seçenek olarak da sayfa kümesini ayırarak yapabilirsiniz. Daha fazla bilgi için bkz “Dinamik sayfa kümesi genişletmesinin etkinleştirilmesi” sayfa 168. Sayfa kümesi küçültmesi zordur, bu nedenle genellikle daha küçük bir sayfa kümesi ayırmak ve gerektiğinde genişletmesine izin vermek daha iyidir.

Sayfa kümesi büyüklüklerini planlarken, uygulama dışı ileti verileri de içinde olmak üzere, oluşturulabilecek tüm iletileri göz önünde bulundurun. Örneğin, tetikleme iletileri, olay iletileri ve uygulamanızın istendiği herhangi bir rapor iletisi.

Sayfa kümesinin büyüklüğü, büyük bir sayfa kümesinin geri yüklenmesine daha uzun süreceği için, bir yedekten geri yükleme işlemi sırasında bir sayfa kümesinin kurtarılması için gereken süreyi belirler.

**Not:** Bir sayfa kümesinin kurtarılması, yedekleme işleminin alındığı tarihten bu yana yazılan günlük kayıtlarını işlemek için kuyruk yöneticisinin ne kadar süreceği zamana bağlıdır; bu süre, yedekleme sıklığına göre belirlenir. Daha fazla bilgi için bkz “Yedekleme ve kurtarma planlanması” sayfa 200.

**Not:** 4 GB ' den büyük sayfa kümeleri SMS ek adreslenirliği kullanılmasını gerektirir.

### V 9.1.4 z/OS veri kümesi şifrelemesini planlama

z/OS veri kümesi şifreleme özelliğini, IBM MQ for z/OS 9.1.3 ya da sonraki bir yayın düzeyinde çalışan kuyruk yöneticilerine ilişkin sayfa kümelerine uygulayabilirsiniz.

Bu sayfa kümelerini EXTENDED öznitelikleriyle ayırmalı ve verilerin AES şifrelenmesini sağlayan bir veri kümesi anahtarı etiketiniz olmalıdır.

confidentiality for data at rest on IBM MQ for z/OS with data set encryption. bölümüne bakın. daha fazla bilgi için.

### Sayfa kümelerinizin boyutunu hesapla

Kuyruk yöneticisi nesne tanımlamaları (örneğin, kuyruklar ve işlemler) için, saklama yeri gereksiniminin hesaplanması basittir; bu nesnelere sabit büyüklükler ve kalıcı olur. Ancak, iletiler için, aşağıdaki nedenlerden dolayı hesaplama daha karmaşıktır:

- İletiler büyüklük olarak değişir.
- Mesajlar transitidir.
- Alınan iletiler tarafından işgal edilen alan, zamanuysuz bir süreç tarafından düzenli aralıklarla geri istenilir.

Ağ duraklarsa, iletiler için ek kapasite sağlayan 4 GB ' den büyük sayfa kümeleri, gerekirse oluşturulabiliyor. Var olan sayfa kümeleri değiştirilemez. Bunun yerine, ek adreslenebilirlik ve genişletilmiş biçim öznitelikleriyle yeni sayfa kümeleri yaratılmalıdır. Yeni sayfa kümeleri eskileriyle aynı fiziksel boyutta

olmalı ve eski sayfa kümeleri yeni sayfa kümelerinin kopyalanması gerekir. Geriye doğru geçiş gerekli ise, sayfa kümesi sıfır değiştirilmemelidir. 4 GB ' den küçük sayfa kümeleri yeterli olursa, herhangi bir işlem gerekmez.

### Sayfa kümesi sıfır

Sayfa kümesi sıfır, nesne tanımlamaları için ayrılmıştır.

Sayfa kümesi sıfır için, gereken saklama alanı:

```
(maximum number of local queue definitions x 1010)
(excluding shared queues)
+ (maximum number of model queue definitions x 746)
+ (maximum number of alias queue definitions x 338)
+ (maximum number of remote queue definitions x 434)
+ (maximum number of permanent dynamic queue definitions x 1010)
+ (maximum number of process definitions x 674)
+ (maximum number of namelist definitions x 12320)
+ (maximum number of message channel definitions x 2026)
+ (maximum number of client-connection channel definitions x 5170)
+ (maximum number of server-connection channel definitions x 2026)
+ (maximum number of storage class definitions x 266)
+ (maximum number of authentication information definitions x 1010)
+ (maximum number of administrative topic definitions x 15000)
+ (total length of topic strings defined in administrative topic definitions)
```

Sayfa kümesi veri kümesi için kümede belirtilecek kayıt sayısını belirlemek için bu değeri 4096 'ya bölün.

Paylaşılan havuzda saklanan nesnelere için izin vermenize gerek yoktur; ancak, sıfır (GROUP ya da QMGR biçiminde yok etme) olarak saklanan ya da sayfa kümesi sıfır olan nesnelere kopyalanmış nesnelere için izin vermelisiniz.

Yaratabileceğiniz nesnelere toplam sayısı, sayfa kümesi sıfır olan kapasiteleriyle sınırlıdır. Tanımlayabileceğiniz yerel kuyrukların sayısı 524 287 ile sınırlıdır.

### Sayfa kümeleri 01-99

Sayfa kümeleri 01-99 için, her sayfa kümesi için gerekli olan saklama alanı, o sayfa kümesinde saklanan iletilerin sayısı ve boyutuna göre belirlenir. (Paylaşılan kuyruklardaki iletiler sayfa kümelerinde saklanmaz.)

Sayfa kümesi veri kümesi için kümede belirtilecek kayıt sayısını belirlemek için bu değeri 4096 'ya bölün.

### İletilere ilişkin depolama gereksiniminin hesaplanması

Bu bölümde, iletilerin sayfalarda nasıl saklanmakta olduğu açıklanmaktadır. Bunun anlaşılması, mesajlarınız için ne kadar sayfa kümesi depolaması tanımlamanız gerektiğini hesaplamaya yardımcı olabilir. Bir sayfadaki tüm iletiler için gereken yaklaşık alanı hesaplamak için, sayfa kümesiyle eşlenen tüm kuyrukların kuyruk derinliği üst sınırını ve bu kuyruklardaki iletilerin ortalama boyutunu göz önünde bulundurmanız gerekir.

**Not:** Bu bölümde verilen yapıların büyüklükleri ve kontrol bilgileri, ana yayın düzeyleri arasında değişmekle yükümlü. IBM MQ sürümünüze özgü ayrıntılar için [SupportPac MP16 - WebSphere MQ for z/OS Kapasite planlaması ve ayarlama](#) ve [IBM MQ Family-Performance Reports](#) belgesine bakın.

You must allow for the possibility that message "gets" might be delayed for reasons outside the control of IBM MQ (for example, because of a problem with your communications protocol). bu durumda, mesajların "put" oranı "get" oranını çok aşabilir. Bu, sayfa kümelerinde saklanan ileti sayısında büyük bir artışa ve talep edilen depolama büyüklüğünün sonraki artışına neden olabilir.

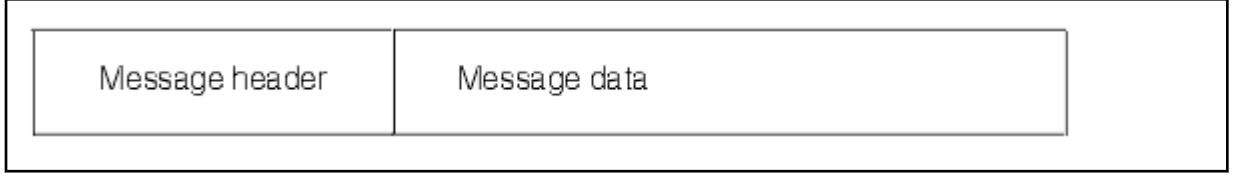
Sayfa kümesindeki her sayfa 4096 bayt uzunluğundadır. Sabit üstbilgi bilgilerinin izin verilmesi, her sayfada iletilerin saklanmasına ilişkin kullanılabilir 4057 bayt yer almıştır.

Her ileti için gereken alan hesaplanırken dikkate alınması gereken ilk şey, iletinin bir sayfaya sığması (kısa bir ileti) ya da iki ya da daha fazla sayfa (uzun bir ileti) üzerinde bölünmesi gerekip gerekmediğini



göz önünde bulundurmanız gerekir. İletiler bu şekilde bölündüğünde, alan hesaplamalarınızda ek denetim bilgileri için izin vermelisiniz.

Alan hesaplamasının amaçları açısından, bir ileti aşağıdaki gibi gösterilebilir:



İleti üstbilgisi kısmı, ileti tanımlayıcısını ve diğer denetim bilgilerini içerir. Bunun büyüklüğü, iletinin büyüklüğüne bağlı olarak değişir. İleti verileri kısmı tüm gerçek ileti verilerini ve diğer tüm üstbilgileri (örneğin, iletim üstbilgisi ya da IMS köprüsü üstbilgisinde) içerir.

Sayfa kümesi denetim bilgileri için en az iki sayfa gereklidir; bu bilgiler genellikle iletiler için gereken toplam alanın %1 değerinden küçüktür.

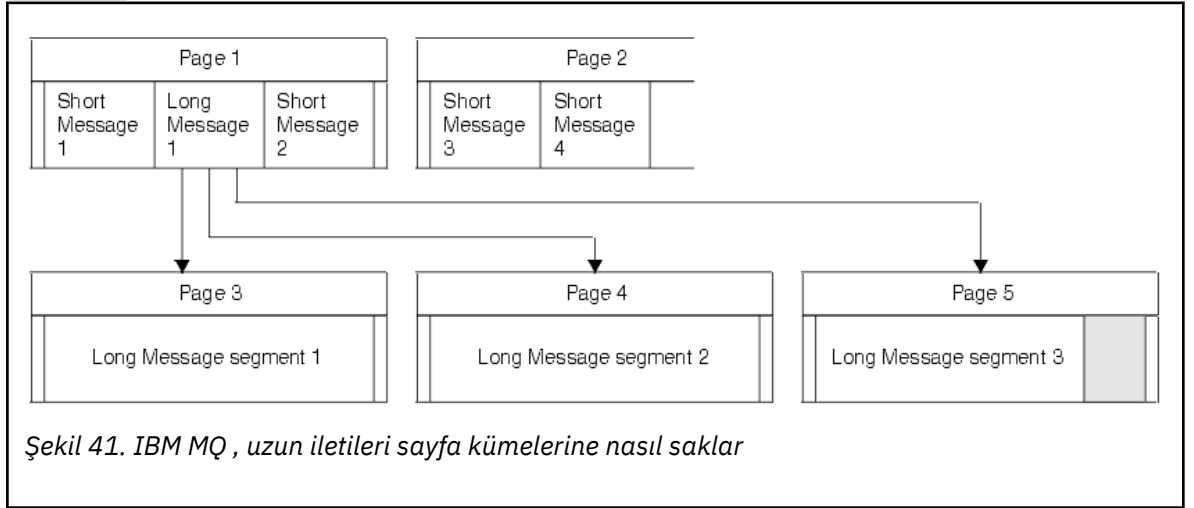
### Kısa iletiler

Kısa bir ileti, bir sayfaya sığan bir ileti olarak tanımlanır.

IBM WebSphere MQ 7.0.1' tan küçük iletiler her sayfada bir tane depolanır.

### Uzun iletiler

İleti verilerinin büyüklüğü 3596 byte 'tan büyükse, ancak 4 MB ' den büyük değilse, ileti uzun bir ileti olarak sınıflanır. IBM MQ , uzun bir iletiyle sunulduğunda, iletiyi bir sayfa serisine depolar ve bu sayfaları gösteren denetim bilgilerini kısa bir iletiyi depolayacağı şekilde saklar. Bu, [Şekil 41](#) sayfa 168'inde gösterilir:



### Çok uzun iletiler

Çok uzun iletiler, 4 MB ' den büyük boyutlara sahip iletilerdir. Bunlar, her 4 MB ' lik 1037 sayfa kullansın diye depolanır. Herhangi bir kalan kısmı, yukarıda açıklandığı gibi, uzun bir iletiyle aynı şekilde saklanır.

## z/OS **Dinamik sayfa kümesi genişletmesinin etkinleştirilmesi**

Kuyruk yöneticisi çalışırken, sayfa kümeleri devingen olarak uzatılabilir. Bir sayfa kümesinin 123 kapsamı olabilir ve birden çok disk birimine dağıtılabilir.

Bir sayfa kümesinin her genişlemesi, yeni bir veri kümesi kapsamı kullanılır. Kuyruk yöneticisi, kapsam sayısı üst sınırına ulaşıncaya ya da uygun birimlerdeki ayırma için kullanılabilir bir depolama alanı kalmayınca kadar, gerektiğinde bir sayfa kümesini genişletmeye devam eder.

Yukarıdaki nedenlerden biri için sayfa kümesi genişletme işlemi başarısız olursa, kuyruk yöneticisi daha fazla genişletme girişiminde bulunmaması için sayfa kümesini işaretler. Bu işaret, sayfa kümesinin EXPAND (SYSTEM) olarak değiştirilmesiyle ilk durumuna getirilebilir.



Sayfa kümesi genişletmesi, sayfa kümesindeki var olan alanın %90 'ının ayrıldığı diğer tüm sayfa kümesi etkinliğiyle zamanuyumsuz bir şekilde gerçekleşir.

Sayfa kümesi genişletme işlemi, yeni ayrılan kapsamı biçimlendirir ve kuyruk yöneticisi tarafından kullanılabilir duruma getirir. Ancak, tüm kapsam biçimleninceye kadar, alanın hiçbiri kullanılabilir değildir. Bu, büyük ölçüde genişlemenin bir süre daha uzun süreceği anlamına gelir ve genişletme işlemi tamamlanmadan önce sayfanın kalan %10 'unu doldururlarsa, uygulamaları 'engelleyebilir' anlamına gelir. Örnek thlqual.SCSQPROC(CSQ4PAGE), ikincil kapsamların nasıl tanımlanacak olduğunu gösterir.

Yeni kapsamların büyüklüğünü denetlemek için, DEFine PSID ve ALTER PSID komutlarının EXPAND anahtar sözcüğünün aşağıdaki seçeneklerinden birini kullanın:

- KULLANICI
- SYSTEM
- YOK

### **KULLANICI**

Sayfa kümesi ayrıldığında belirlenen ikincil kapsam büyüklüğünü kullanır. Bir değer belirtilmediyse ya da sıfır değeri belirtilmişse, dinamik sayfa kümesi genişletme işlemi gerçekleşemez.

Sayfa kümesi genişletmesi, sayfadaki alan %90 kullanıldığında oluşur ve diğer sayfa kümesi etkinliğiyle zamanuyumsuz olarak gerçekleştirilir.

Bu, bir kerede tek bir kapsamdan daha fazla genişlemeye yol açabilir.

Aşağıdaki örneği göz önünde bulundurun: birincil kapsam 100 bin sayfa ve ikincil kapsam 5000 sayfa olan bir sayfa kümesi ayırıyorsunuz. 9999 sayfa gerektiren bir ileti ortaya konmaktadır. Sayfa kümesi şimdiden 85.000 sayfa kullanıyorsa, iletiyi yazmak %90 'ın tam sınırını geçer (90 bin sayfa). Bu noktada, ikincil kapsam olarak 100.000 sayfanın birincil kapsamına ayrıldığında, sayfa kümesi büyüklüğü 105.000 sayfa olarak ayarlanır. İletinin geri kalan 4999 sayfası yazıya devam ediyor. Kullanılan sayfa alanı 94.500 sayfaya ulaştığında, güncellenen sayfa kümesinin %90 'ı olan 105.000 sayfa, başka bir 5000 sayfa kapsamı ayrılır ve sayfa kümesi boyutu 110.000 sayfaya kadar ayrılır. MQPUT ' un sonunda, sayfa kümesi iki kez genişletildi ve 94.500 sayfa kullanıldı. İkinci sayfa kümesi genişletme içindeki sayfaların hiçbiri ayrılmamış olsa da, hiçbir sayfa kullanılmadı.

Yeniden başlatma sırasında, önceden kullanılan bir sayfa kümesi daha küçük bir veri kümesiyle değiştirildiyse, daha önce kullanılan veri kümesinin boyutuna ulaşıncaya kadar genişletilir. Bu boyuta ulaşmak için yalnızca bir kapsam gereklidir.

### **SYSTEM**

Sayfa kümesi tanımlandığında belirtilen ikincil kapsam büyüklüğünü yoksayar. Bunun yerine, kuyruk yöneticisi yürürlükteki sayfa kümesi büyüklüğünün yaklaşık %10 'u bir değer ayarlar. Değer, DASD ' nin en yakın silindirine yuvarlanır.

Bir değer belirtilmediyse ya da sıfır değeri belirtilmişse, dinamik sayfa kümesi genişletme işlemi gerçekleşebilir. Kuyruk yöneticisi, yürürlükteki sayfa kümesi büyüklüğünün yaklaşık %10 'u olan bir değer ayarlar. Yeni değer, DASD ' nin özelliklerine bağlı olarak yukarı yuvarlanır.

Sayfa kümesi genişletmesi, sayfa kümesindeki alan yaklaşık %90 kullanılırsa ve diğer sayfa kümesi etkinliğiyle zamanuyumsuz olarak gerçekleştirildiğinde oluşur.

Yeniden başlatma sırasında, önceden kullanılan bir sayfa kümesi daha küçük bir veri kümesiyle değiştirildiyse, daha önce kullanılan veri kümesinin boyutuna ulaşıncaya kadar genişletilir.

### **YOK**

Başka bir sayfa kümesi genişletme işlemi gerçekleşmez.

### **İlgili başvurular**

ALTER PSID

DEĞERLERİ

KULLANIM KULLANI

## **Arabellek Havuzlarının Tanımlanması**

Tanımlamanız gereken arabellek havuzlarının sayısını ve bunların ayarlarını planlamaya yardımcı olmak için bu konuyu kullanın.

Bu konu aşağıdaki kısımlara bölünmektedir:

1. [“Tanımlanacak arabellek havuzlarının sayısına karar verin” sayfa 170](#)
2. [“Her arabellek havuzu için ayara karar verin” sayfa 171](#)
3. [“Beklenen yükleme altındaki arabellek havuzlarının başarımını izleme” sayfa 171](#)
4. [“Arabellek havuzu özelliklerini ayarla” sayfa 171](#)

## **Tanımlanacak arabellek havuzlarının sayısına karar verin**

Başlangıçta dört arabellek havuzu tanımlamanız gerekir:

### **Arabellek havuzu 0**

Use for object definitions (in page set zero) and performance critical, system related message queues, such as the SYSTEM.CHANNEL.SYNCQ queue and the SYSTEM.CLUSTER.COMMAND.QUEUE and SYSTEM.CLUSTER.REPOSITORY.QUEUE queues.

However it is important to consider point “7” sayfa 172 in *Arabellek havuzu özelliklerini ayarla* if a large number of channels, or clustering, is to be used.

Kullanıcı iletileri için kalan üç arabellek havuzunu kullanın.

### **Arabellek havuzu 1**

Uzun ömürlü iletiler için kullanın.

Uzun ömürlü iletiler, sistemde yer alan iki denetim noktasından daha uzun süre kalan ve bu iletilerin sayfa kümesinden önce yazıldığı iletiler olur. Uzun ömürlü birçok iletiniz varsa, bu arabellek havuzu görece olarak küçük olmalıdır; böylece, sayfa kümesi G/Ç eşit olarak dağıtılır (arabellek havuzu her %85 dolunca daha eski iletiler DASD ' ye yazılır).

Arabellek havuzu çok büyükse ve arabellek havuzu hiçbir zaman %85 'e ulaşmazsa, sayfa kümesi G/Ç denetim noktası işleninceye kadar geciktirilir. Bu işlem, sistem genelinde yanıt sürelerini etkileyebilir.

Yalnızca birkaç uzun ömürlü ileti bekliyorsanız, bu arabellek havuzunu, tüm bu iletileri tutmak için yeterince büyük olacak şekilde tanımlayın.

### **Arabellek havuzu 2**

Performans açısından kritik, kısa ömürlü iletiler için kullanın.

Genellikle birkaç arabelleği kullanarak yüksek bir arabellek kullanımı derecesi vardır. Ancak, bir sunucu uygulaması başarısız olduğunda, beklenmeyen ileti birikimine izin vermek için bu arabellek havuzunu büyük hale getirmeniz gerekir.

### **Arabellek havuzu 3**

Diğer tüm (tipik, performans açısından kritik önem derecesi olmayan) iletiler için kullanın.

Ölü-harfli kuyruk gibi kuyruklar, SYSTEM.COMMAND.\* kuyruklar ve SYSTEM.ADMIN.\* kuyruklar, arabellek havuzu 3 ile de eşlenebilir.

Sanal saklama koşullarının var olduğu ve arabellek havuzlarının daha küçük olması gerektiğinde, arabellek havuzu 3, büyüklük azaltma için ilk adaydır.

Aşağıdaki durumlarda ek arabellek havuzları tanımlamanız gerekebilir:

- Belirli bir kuyruğun yalıtma gerektireceği biliniyorsa, çeşitli zamanlarda farklı davranış sergilediği için olabilir.
  - Bu tür bir kuyruk, değişen koşullar altında en iyi başarımı gerektirebilir ya da bir arabellek havuzundaki diğer kuyrukları olumsuz etkilememesi için yalıtılması gerekir.
  - Bu kuyruğun her biri, kendi arabellek havuzu ve sayfa kümesi içinde yalıtılabilir.
- Hizmet sınıfı nedenleri için birbirlerinden farklı kuyruk kümelerini yalıtma istiyorsunuz.

- Her bir kuyruk kümesi, Arabellek havuzu ayarları için önerilen tanımlamaları içinde açıklandığı gibi, 1 ya da 2 numaralı arabellek havuzlarının iki tipinin birini ya da her ikisini birden gerektirebilir. Bu, belirli bir tipteki arabellek havuzlarının birkaç arabelleği havuzunu gerektirir.

## Her arabellek havuzu için ayara karar verin

“Tanımlanacak arabellek havuzlarının sayısına karar verin” sayfa 170 içinde açıklanan dört arabellek havuzunu kullanıyorsanız, Arabellek havuzu ayarlarına ilişkin önerilen tanımlamalar , arabellek havuzlarının büyüklüğü için iki değer kümesi sağlar.

**V 9.1.0** İlk set bir test sistemi için uygun, diğeri bir üretim sistemi ya da sonunda üretim sistemi olacak bir sistem için uygundur. Tüm durumlarda, arabellek havuzlarınızı **LOCATION**(YUKARIDA) öznitelikle tanımlar

Tanımlama ayarı	Test sistemi	Üretim sistemi
ARABELLEK ALANI 0	1 050 arabellekler	50 000 arabellek
ARABELLEK HAVUZU 1	1 050 arabellekler	20 000 arabellek
ARABELLEK HAVUZU 2	1 050 arabellekler	50 000 arabellek
ARABELLEK HAVUZU 3	1 050 arabellekler	20 000 arabellek

Önerilen dört arabellek havuzundan daha fazlasına gereksinim duyarsanız, arabellek havuzundaki kuyrukların beklenen davranışını en doğru şekilde tanımlayan arabellek havuzunu (1 ya da 2) seçin ve Arabellek havuzu ayarları için önerilen tanımlamaları içindeki bilgileri kullanarak bu havuzu kullanarak bu havuzu boyutlayın.

**V 9.1.0** MEMLIMIT ayarınızın yeterince yüksek olduğundan emin olun, böylece tüm arabellek havuzları çubuğun üst kısmında bulunabilir.

## Beklenen yüklemeye altındaki arabellek havuzlarının başarımını izleme

Arabellek havuzu başarım istatistiklerini çözümlenerek arabellek havuzlarının kullanımını izleyebilirsiniz. Özellikle, arabellek havuzlarının yeterince büyük olduğundan, QPSTSOS, QPSTSTLA ve QPSTDMC değerlerinin sıfır olduğu kadar büyük olduğundan emin olmalısınız.

Ek bilgi için Arabellek yöneticisi veri kayıtları başlıklı konuya bakın.

## Arabellek havuzu özelliklerini ayarla

Gerekliyse, arabellek havuzu ayarlarını “Her arabellek havuzu için ayara karar verin” sayfa 171' dan ayarlamak için aşağıdaki noktaları kullanın.

Use the performance statistics from “Beklenen yüklemeye altındaki arabellek havuzlarının başarımını izleme” sayfa 171 as guidance.

1. Daha önceki bir IBM MQ sürümünden geçiş yapıyorsanız, yalnızca daha fazla gerçek depolamaya sahipseniz, var olan ayarlarınızı değiştirin.
2. Genel olarak, daha büyük arabellek havuzları başarım için daha iyi olur ve arabellek havuzları, çubuğun yukarısında yer aldıklarında çok daha büyük olabilir.

Ancak, her zaman, arabellek havuzlarının gerçek depolamada yerleşik olması için yeterli miktarda gerçek saklama alanına sahip olmamanız gerekir. Sayfalama ile sonuçlanmamış daha küçük arabellek havuzlarının olması, büyük olanlarından daha iyidir.

Buna ek olarak, bir arabellek havuzunun, bu değeri kullanan sayfa kümelerinin toplam büyüklüğünden daha büyük olmasına gerek yoktur; ancak, sayfa kümesi genişlemesini büyük olasılıkla da hesaba kasanız da, bu değeri kullanan bir değer yoktur.

3. Her arabellek havuzu için bir sayfa kümesi hedefinde, bunun yanı sıra daha iyi uygulama yalıtımı sağlar.
4. Arabellek havuzlarının işletim sistemi tarafından hiçbir zaman çağrılmaması gibi, yeterli gerçek saklama yeriniz varsa, arabellek havuzunuzdaki sayfa sabitleyici arabellekleri kullanmayı düşünün.  
Bu özellikle, arabellek havuzunun, G/Ç 'den önceki sayfa düzeltmelerini düzeltmeyle ilişkili CPU maliyetini kaydetmesi ve daha sonra yeniden ayarlamaması durumunda, büyük olasılıkla çok G/Ç' ye geçecekse önemlidir.
5. Çubuğun altına sığabilmek için yeterince küçük olsalar da, arabellek havuzlarını çubuğun yukarısına yerleştirmenin birkaç yararı vardır. Bu bilgiler şunlardır:
  - 31 bit sanal depolama kısıtlaması yardımı-örneğin, ortak depolama alanı için daha fazla alan.
  - Bir arabellek havuzunun büyüklüğü, yoğun olarak kullanıldığında beklenmeyen bir şekilde artırılmalıdır; kuyruk yöneticisi için daha az etki ve risk vardır ve iş yükü, arabellek havuzunu çubuğun üstüne taşıyıp daha sonra arabellekler ekleyerek, zaten çubuğun üzerinde olan bir arabellek havuzuna daha fazla arabellek ekleyerek, kuyruk yöneticisiyle ilgili iş yükünü artırır.
6. 15% boş eşğin hiçbir zaman aşılabilmesi için, arabellek havuzunu sıfır ve kısa ömürlü iletiler için arabellek havuzunu (arabellek havuzu 2) ayarlayın (bu durumda QPSTCBSL, QPSTNBUF ' a bölü her zaman %15 'ten büyük olur). Arabelleklerin %15 'ten fazlası serbest kalıyorsa, bu arabellek havuzlarını kullanan sayfa kümelerinde G/Ç, normal çalışma sırasında büyük ölçüde önlenemez, ancak iki denetim noktasından daha eski iletiler sayfa kümelerine yazılır.



**Uyarı:** Bu parametrelere ilişkin en iyi değer, tek tek sistemin özelliklerine bağlıdır. Verilen değerler yalnızca bir kılavuz olarak amaçlanır ve sisteminiz için uygun olmayabilir.

7. SISTEM.\* çok derin kuyruklar, örneğin, SYSTEM.CHANNEL.SYNCQ(CHANNEL.SYNCQ), yeterli depolama alanı varsa, kendi arabellek havuzlarına konmaktan yararlanabilir.

IBM MQ SupportPac MP16 - WebSphere MQ for z/OS Kapasite planlaması ve ayarlama , ayarlama arabelleği havuzlarına ilişkin ek bilgi sağlar.

## **Günlüğe kaydetme ortamınızın planlanması**

Use this topic to plan the number, size and placement of the logs, and log archives used by IBM MQ.

Günlükler şu şekilde kullanılır:

- Kalıcı iletilere ilişkin kurtarma bilgilerini yazma
- Kalıcı iletileri kullanarak iş birimleri ile ilgili bilgileri kaydet
- Nesne tanımlama gibi nesnelere ilişkin değişikliklere ilişkin bilgileri kaydet
- CF yapılarını yedekle

ve diğer iç bilgiler için.

IBM MQ günlüğe kaydetme ortamı, seçenekler belirtmek için sistem parametre makroları kullanılarak kurulur; örneğin, tek ya da ikili etkin günlüklerin, arşiv günlüğü birimleri için hangi ortamın kullanılacağını ve kaç günlük arabelleğinin sahip olduğunu belirtir.

Bu makrolar [Önyükleme ve günlük veri kümelerinin yaratılması](#) ve [Sistem parametre biriminizi uyarlayın](#) de açıklanmıştır.

**Not:** Kuyruk paylaşım gruplarını kullanıyorsanız, önyükleme ve günlük verileri setlerini SHAREOPTIONS (2 3) ile tanımladığınızdan emin olun.

Bu bölümde aşağıdaki konulara ilişkin bilgiler yer almaktadır:

## **Veri kümesi tanımlamalarını günlüğe kaydet**

Günlük veri kümeleriniz için en uygun yapılandırmaya karar vermek için bu konuyu kullanın.

Bu konu, aşağıdaki soruları yanıtlamanıza yardımcı olacak bilgiler içerir:

- [Kuruluşunuz tek ya da çift günlük kaydı kullanmalı mı?](#)

- Kaç etkin günlük veri kümesi gerekiyor?
- “Etkin günlükler ne kadar büyük olmalıdır?” sayfa 174
- Etkin günlük yerleşimi
- **V 9.1.4** “z/OS veri kümesi şifrelemesi ile etkin günlük şifrelemesi” sayfa 175

## Kuruluşunuz tek ya da çift günlüğe kaydetmeyi kullanmalı mı?

Genel olarak, veri kaybetme riskini en aza indirmek için üretim için ikili günlük kaydı kullanmanız gerekir. Test sisteminizin üretimi yansıtmamasını istiyorsanız, her ikisi de çift günlük kaydı kullanmalıdır; tersi durumda, test sistemleriniz tek günlüğe kaydetme olanağını kullanabilir.

Tek günlük verileri, bir günlük veri kümesi kümesiyle yazılır. Çift günlük verileri, iki günlük veri kümesi kümesine yazılır; bu nedenle, yanlışlıkla silinen veri kümesi gibi bir günlük veri kümesiyle ilgili bir sorun durumunda, verileri kurtarmak için diğer günlükler kümesinde eşdeğer veri kümesi kullanılabilir.

Çift günlüğe kaydetmeyle, tek günlüğe kaydetmeyle birlikte iki kat daha fazla DASD gerekir.

İkili günlük kaydı kullanıyorsanız, veri kurtarma için yeterli miktarda sağlanması sağlamak için çift BSDSS ve çift arşivleme da kullanın.

İkili etkin günlük kaydı, küçük bir başarımlı maliyeti ekler.



**Uyarı:** Use of disk mirroring technologies, such as Metro Mirror, are not necessarily a replacement for dual logging and dual BSDS. İkizlenmiş bir veri kümesi yanlışlıkla silinirse, her iki kopya da kaybedilir.

Kalıcı iletiler kullanıyorsanız, tek günlüğe kaydetme, %10-30 oranında maksimum kapasiteyi artırabilir ve yanıt sürelerini de iyileştirebilir.

Tek günlüğe kaydetme, 2-310 etkin günlük veri kümesi kullanır; oysa ikili günlüğe kaydetme işlemi, aynı sayıda etkin günlük sağlamak için 4-620 etkin günlük veri kümesi kullanır. Bu nedenle, tek günlüğe kaydetme işlemi, günlüğe kaydedilen veri miktarını azaltır. Bu işlem, kuruluşunuz G/Ç kısıtlıysa önemli olabilir.

## Kaç etkin günlük veri kümesi gerekiyor?

Günlüklerin sayısı, kuyruk yöneticinizin etkinliklerine bağlıdır. Düşük verimi olan bir test sistemi için, üç etkin günlük veri kümesi uygun olabilir. Yüksek çıkan bir üretim sistemi için, kullanılabilir günlük sayısı üst sınırını isteyebilirsiniz. Bu nedenle, günlük boşaltma günlükleriyle ilgili bir sorun varsa, sorunları çözmek için daha fazla zamanınız vardır.

En az üç etkin günlük veri kümeniz olmalıdır, ancak daha fazla tanımlamayı tercih edilebilir. Örneğin, bir günlüğü doldurmak için alınan süre büyük olasılıkla, doruk yükleme sırasında bir günlüğü arşivlemek için alınan zamana yaklaşırsa, daha fazla günlük tanımlayın.

**Not:** Sayfa kümeleri ve etkin günlük veri kümeleri, genişletilmiş adres birimlerinin (EAV) genişletilmiş adresleme alanında (EAV) yer almaya hak kazanır ve z/OS V1.12'den bir arşiv günlüğü veri kümesi EAS' de de bulunabilir.

Ayrıca, günlük arşivlemede olası gecikmeleri dengelemek için ek günlükler de tanımlamanız gerekir. Manyetik bantta arşiv günlüklerini kullanırsanız, manyetik bantı monte etmek için gereken süre boyunca izin verin.

DASD eksikliği nedeniyle ya da manyetik bantta yazamadığından, sistemin arşivlenememesi durumunda, gününün veri almaya yetecek kadar etkin günlük alanına sahip olmayı düşünün. Tüm etkin günlükler doldurulduysa, IBM MQ kalıcı iletileri ya da işlemleri işleyemez. Yeterli sayıda etkin günlük alanına sahip olmak çok önemlidir.

Arşiv gecikmelerinin ya da sorunlarının etkisini en aza indirmenin bir yolu olarak yeni etkin günlük veri kümelerini dinamik olarak tanımlamak mümkündür. Yeni veri kümeleri, etkin günlük içinde yer olmaması

nedeniyle kuyruk yöneticisinin 'stall' olmasını önlemek için **DEFINE LOG** komutu kullanılarak hızla çevrimiçi olarak getirilebilir.

31 'den fazla etkin günlük veri kümesi tanımlamak istiyorsanız, günlüğe kaydetme ortamınızı bir sürüm 2 biçimi BSDS kullanacak şekilde yapılandırmanız gerekir. Sürüm 2 biçimi BSDS kullanımda olduğunda, her bir günlük kopyası halkası için en fazla 310 etkin günlük veri kümesi tanımlanabilir. Sürüm 2 biçimli BSDS 'ye nasıl dönüştürülebilmenize ilişkin bilgi için bkz. "[Adreslenebilir günlük aralığı üst sınırını artırmayı planlama](#)" sayfa 182 .

Kuyruk yöneticinizin, yazdırma günlüğü eşlemi yardımcı programını (CSQJU004) çalıştırarak ya da kuyruk yöneticisi başlatma işlemi sırasında yayınlanan CSQJ034I iletisinden 2 ya da daha yüksek bir BSDS sürümünü kullanıp kullanmadığını da söyleyebilirsiniz. An end of log RBA range of FFFFFFFFFFFFFFFF, in the CSQJ034I message, indicates that a version 2, or higher, format BSDS is in use. An end of log RBA range of 0000FFFFFFFFFFFFFF, in the CSQJ034I message, indicates that a version 1 format BSDS is in use.

Bir kuyruk yöneticisi sürüm 2 ya da daha yüksek bir sürümü kullanırken BSDS 'yi biçimlendirirken, 31 'den fazla etkin günlük veri kümesini bir günlük kopyalama halkasına dinamik olarak eklemek için **DEFINE LOG** komutunun kullanılması mümkündür.

## Etkin günlükler ne kadar büyük olmalıdır?

IBM MQ 8.0' den, desteklenen etkin günlük büyüklüğü üst sınırı diskten diske arşivlenirken 4 GB 'dir. Ürünün önceki yayınlarında, diske arşivlenirken desteklenen etkin günlük büyüklüğü üst sınırı 3 GB ' dir.

Manyetik banda arşivlenirken, etkin günlük büyüklüğü üst sınırı 4 GB 'dir.

Üretim ve test sistemleri için en az 1 GB ' lik etkin günlükleri oluşturmalısınız.

**Önemli:** Veri kümelerini ayırırken dikkatli olmanız gerekir, çünkü IDCAMS, ayırdığınız boyutu yuvarlar.

3 GB ' lik bir günlük ayırmak için aşağıdaki seçeneklerden birini belirleyin:

- Silindirler (4369)
- Megabayt (3071)
- TRACKS (65535)
- KAYIT (786420)

Bu ayırmalardan herhangi biri 2.99995 GB ' yi bulur.

4GB günlüğü ayırmak için aşağıdaki seçeneklerden birini belirleyin:

- Silindir sayısı (5825)
- Megabayt (4095)
- TRACKS (87375)
- KAYIT (1048500)

Bu ayırmalardan herhangi biri 3.9997 GB ' yi bulur.

Veri kümesinin birden çok birime dağıtıldığı çizgili veri kümelerini kullanırken, belirlenen boyut değeri, paylaşırma için kullanılan her bir DASD birimi üzerinde ayrılır. Bu nedenle, 4 GB ' lik günlükler ve dört birim paylaşırma için kullanmak istiyorsanız şunları belirtmeniz gerekir:

- CYLinders (1456)
- Megabayt (1023)

Bu özniteliklerin ayarlanması  $4 * 1456 = 5824$  silindirleri ya da  $4 * 1023 = 4092$  Megabayt 'ı ayırır.

**Not:** Paylaşırma, genişletilmiş biçim veri kümeleri kullanılırken desteklenir. Bu genellikle depolama yöneticisi tarafından ayarlanır.

Yordamı gerçekleştirmeye ilişkin bilgi için bkz. [Etkin günlük büyüklüğünün artırılması](#) .

## Etkin günlük yerleşimi

Kuyruk yöneticileri için depolama havuzları ayarlamak üzere depolama yönetimi ekibinizle birlikte çalışmanız gerekir. Şunları göz önünde bulundurmanız gerekir:

- Bir adlandırma kuralı, kuyruk yöneticileri doğru SMS tanımlarını kullanır.
- Etkin ve arşiv günlükleri için gereken alan. Depolama havuzunuzun tüm bir gündeki etkin günlükler için yeterli alanı olmalıdır.
- Başarıya dayanıklılık ve performans.

Performans nedenlerinden dolayı, etkin günlük veri kümelerinizi paylaşırma dikkate almanız gerekir. G/Ç, birden çok birime dağıtılır ve G/Ç yanıt sürelerini kısaltır ve daha yüksek verim elde eder. Paylaşırma kullanırken etkin günlüklerin büyüklüğünün ayrılmasına ilişkin bilgi için önceki metne bakın.

RMF ' den ya da benzeri bir ürünle gelen raporları kullanarak G/Ç istatistiklerini gözden geçirmelisiniz. Veri kümelerinin konumu nedeniyle herhangi bir gecikme olmadığından emin olmak için IBM MQ veri kümeleri için aylık (veya daha sık sık) bu istatistiklerin gözden geçirilmesini gerçekleştirin.

Bazı durumlarda, çok fazla IBM MQ sayfa kümesi G/Ç olabilir ve bu, aynı DASD ' de bulunuyorsa IBM MQ günlük başarımını etkileyebilir.

İkili günlük kaydı kullanıyorsanız, her bir etkin ve arşiv günlüğü kümesinin ayrı tutulduğundan emin olun. Örneğin, bunları ayrı DASD altsistemlerinde ya da farklı aygıtlarda ayırın.

Bu, birimlerden biri bozulursa ya da yok edildiyse, bunların her ikisi de kaybolma riskini azaltır. Günlüğün her iki kopyası da kaybolursa, veri kaybının olasılığı yüksektir.

Yeni bir etkin günlük verisi yarattığınızda, bu verileri [CSQJUFMT](#) komutunu kullanarak önceden biçimlendirmelisiniz. Günlük önceden biçimlenmediyse, kuyruk yöneticisi günlüğü ilk kullanımında biçimlendirir ve bu da performansı etkiler.

Büyük dönen disklere sahip daha eski bir DASD ile, en iyi performansı elde etmek için hangi birimlerin kullanıldığına dikkat edin.

Verilerin birçok PC boyutlu disklerin üzerine yayıldığı modern DASD ile, hangi birimlerin kullanıldığı hakkında bu kadar endişelenmenize gerek yoktur.

Depolama yöneticiniz, performans sorunlarını gözden geçirmek ve çözmek için kurumsal DASD ' yi denetlemelidir. Kullanılabilirlik için, bir DASD altsisteminde bir dizi oturum ve farklı bir DASD altsistemindeki ikili günlükler kullanmak isteyebilirsiniz.

## z/OS veri kümesi şifrelemesi ile etkin günlük şifrelemesi

V 9.1.4

z/OS veri kümesi şifreleme özelliğini, IBM MQ for z/OS 9.1.3 ya da sonraki bir yayın düzeyinde çalışan kuyruk yöneticilerine ilişkin etkin günlük veri kümelerine uygulayabilirsiniz.

Bu etkin günlük veri kümelerini EXTENDED öznitelikleriyle ayırmalı ve verilerin AES ' in şifrelenmesini sağlayan bir veri kümesi anahtarı etiketlemeniz gerekir.

[confidentiality for data at rest on IBM MQ for z/OS with data set encryption](#). bölümüne bakın. daha fazla bilgi için.

V 9.1.2

z/OS

### IBM MQ ile MetroMirror ' in kullanılması

IBM Metro Mirror, previously known as Synchronous Peer to Peer Remote Copy (PPRC), is a synchronous replication solution between two storage subsystems, where write operations are completed on both the primary and secondary volumes before the write operation is considered to be complete. Metro Mirror , depolama altsistemi hatası durumunda veri kaybı olmaması gereken ortamlarda kullanılabilir.



## Desteklenen veri kümesi tipleri

Aşağıdaki IBM MQ veri kümesi tiplerinin tümü, Metro Mirror kullanılarak çoğaltılabilir. Ancak, tam olarak hangilerinin eşlendiği kuruluşunuzun uygunluk gereksinimlerine bağlıdır:

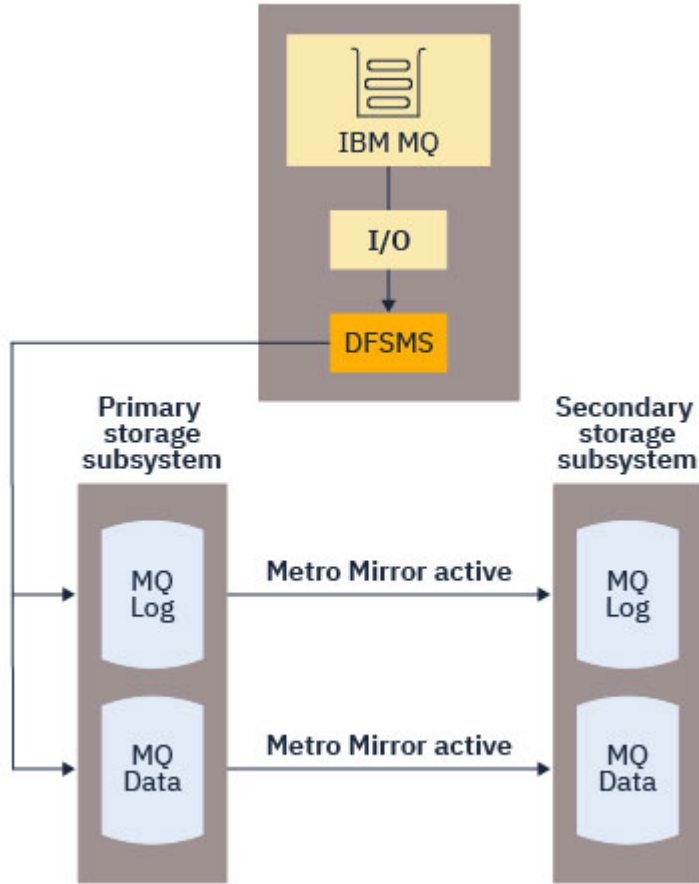
- Etkin günlükler
- Günlükleri arşivle
- Önyükleme veri kümesi (BSDS)
- Sayfa kümeleri
- Paylaşılan ileti veri kümesi (SMDS)
- Yapılandırma için kullanılan veri kümeleri; örneğin, MSTR JCL üzerindeki CSQINP\* DD kartlarında

## zHyperWrite with IBM MQ etkin günlükleriyle yazma

Metro Mirror kullanılarak eşlenen bir veri kümesine bir yazma yapıldığında, yazma ilk olarak birincil birime yapılır ve sonra ikincil birime eşlenir. This replication is done by the storage subsystem and is transparent to the application that issued the write, for example IBM MQ.

Bu işlem aşağıdaki şemada gösterilmektedir.

Bir veri kümesini eşlemek için kullanılan Metro Mirror 'ı



gösteren

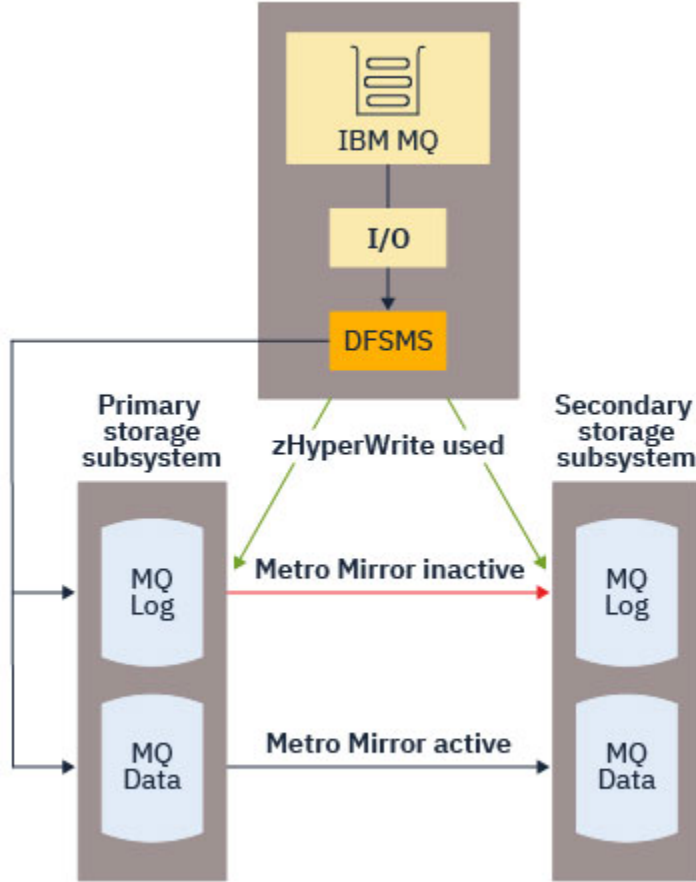
Her iki yazma işleminin birincil ve ikincil depolama alt sistemlerine yazma işlemi IBM MQ' e geri dönmeden önce tamamlanabilmesi için Metro Mirror işlevinin bir başarıım etkisi olabilir. Bu performans etkisini, Metro Mirror' ın kullanılabilirlik avantajlarına karşı dengelemeniz gerekir.

IBM MQ etkin günlüklerinin çoğu Metro Mirror kullanılarak yapılan performans etkisine karşı duyarlıdır. IBM MQ 9.1.2, bu performans etkisini azaltmaya yardımcı olmak için etkin günlüklerle zHyperYazma desteği ekler.



zHyperWrite, Metro Mirror ile eşlenen veri kümelerinde yapılan yazma işlemlerinin performans etkisini azaltmak için z/OS ile birlikte çalışan bir depolama altsistemi teknolojidir. zHyperYazma kullanıldığında, birincil ve ikincil birimlere yazma işlemi, depolama altsistemi düzeyinde sırayla olmak yerine Data Facility Storage Management Subsystem (DFSMS (Veri Olanağı Depolama Yönetimi Altsistemi) (DataData Facility Storage Management Subsystem; Veri Olanağı Depolama Yönetimi Altsistemi) (Data Facility Storage Management Subsystem; Veri Olanağı Depolama Yönetimi Altsistemi) düzeyine paralel olarak

Aşağıdaki şemada, etkin günlükler için kullanılan zHyperYazma ve diğer IBM MQ veri kümesi tipleri için kullanılan Metro Mirror yazıcılarıyla gösterilir. Note that if a zHyperWrite write fails, DFSMS will transparently reissue the write using Metro Mirror.



zHyper IBM MQ üzerine yazma, yalnızca etkin günlük veri kümelerinde desteklenir.

zHyper' un etkin günlüklerle yazma olanağını kullanabilmek için aşağıdakileri yapmak gerekir:

- Configure IBM MQ to use zHyperWrite, and
- Etkin günlüklerin zHyperYazma özellikli birimler üzerinde olması gerekir.

Bu koşulların her ikisi de karşılanırsa, etkin günlüklere yazma işlemi zHyperYazma özelliği için etkinleştirilir.

IBM MQ konfigürasyonunu, aşağıdaki yöntemlerden birini kullanarak zHyperYazma işlevini kullanacak şekilde yapılandırabilirsiniz:

- Sistem parametre modülünde ZHYWRITE(YES) değerini belirleyin.
- SET LOG ZHYWRITE(YES) komut SET komutunu verin.

Etkin günlük veri kümelerinin zHyperWrite yetenekli birimlerinde olacak şekilde ayarlanması için aşağıdaki koşulları ayarlayın:

- Metro Mirror için birimleri etkinleştirin ve birimler zHyperWrite (Yazma) seçeneğini destekler.

- Birimlerin HyperSwap etkin olduğundan emin olun
- IECIOSxx parametresinde HYPERWRITE=YES değerini belirtin

Önceki tüm koşullar karşılanırsa, etkin günlüklere yazma işlemi zHyperYazma özelliği için etkinleştirilir.

Bu koşullardan biri ya da daha fazlası karşılanmazsa, IBM MQ etkin günlüklere normal olarak yazar ve Metro Mirror yapılandırıldıysa yazma işlemlerini çoğaltır.

#### Notlar:

- IBM MQ , tüm etkin günlük veri kümelerinin zHyperYazma özellikli birimler üzerinde olmasını gerektirmez.

IBM MQ , bazı etkin günlük veri kümelerinin zHyperWrite yetenekli birimlerde olduğunu saptarsa ve diğerleri değilse, CSQJ166E iletisi yayınlar ve işleme devam eder.

- IBM MQ , veri kümeleri ilk açıldığında etkin günlük veri kümelerinin zHyperYazma yetenekli olup olmadığını denetler.

Günlük veri kümeleri, kuyruk yöneticisinde açılır ya da DEFINE LOG komutu kullanılarak dinamik bir şekilde ekleniyor. If the log data sets are made zHyperWrite capable while a queue manager has them open, the queue manager will not detect this until it has been restarted.

Yürürlükteki etkin günlük veri kümelerinin zHyperWrite yetenekli olup olmadığını belirtmek için DISPLAY LOG komutunun çıkışını kullanabilirsiniz. Aşağıdaki örnek, veri kümelerinin her ikisinin de zHyperYazma yeteneğine sahip olduğunu göstermektedir. If the queue manager has been configured with ZHYWRITE(YES), writes to these logs would be enabled for zHyperWrite:

```
Copy %Full zHyperWrite DSName
1 4 CAPABLE MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
2 4 CAPABLE MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
```

### **Günlük arşivi depolamanızın planlanması**

Arşiv günlüğü veri kümelerinizi korumanın farklı yollarını anlamak için bu konuyu kullanın.

Arşiv günlüğü veri kümelerini standart etiket manyetik bantlarına ya da DASD ' ye yerleştirebilir ve bunları, veri olanağı hiyerarşik depolama yöneticisi (DFHSM) ile yönetebilirsiniz. Bir arşiv günlüğü veri kümesindeki her z/OS mantıksal kaydı, etkin günlük veri kümesinden bir VSAM denetim arasındadır. Blok büyüklüğü 4 KB ' lik bir tiptir.

Arşiv günlüğü veri kümeleri, IBM MQ tarafından seçilen adlarla dinamik olarak ayrılır. Bu tür ayırmalar için gerekli olan veri kümesi adı öneki, blok boyutu, birim adı ve DASD boyutları sistem parametre modülünde belirtilir. IBM MQ ' un arşiv günlüğü veri kümesi adına bir tarih ve saat eklemesi için kuruluş sırasında da seçebilirsiniz.

Yeni arşiv günlükleri için IBM MQ ile belirtilmek mümkün değildir, ancak bunu yönetmek için Storage Management yordamlarını kullanabilirsiniz. Ayırma hataları oluşursa, boşaltma işlemi bir sonraki boşaltma tetikleninceye kadar ertelenir.

Kuruluş sırasında ikili arşiv günlükleri belirtirseniz, etkin günlüğünden alınan her bir günlük denetim aralığı iki arşiv günlüğü veri kümesine yazılır. Arşiv günlüğü veri kümelerinin çiftinde yer alan günlük kayıtları aynıdır, ancak birim sonu noktaları, çoklu birim veri kümeleri için eşitlenmez.

### **Arşiv günlükleriniz manyetik bant ya da DASD üzerinde mi olmalıdır?**

Arşiv günlükleriniz için manyetik bant mı, yoksa DASD mi kullanacağınıza karar verilirken dikkate almanız gereken bir dizi faktör vardır:

- Manyetik bant ya da diske karar vermeden önce işletim yordamlarınızı gözden geçirin. Örneğin, manyetik bantı arşivlemek seçerseniz, gerekli olduğunda manyetik bant sürücüsü yeterli olmalıdır. Bir olağanüstü durumdan sonra tüm altsistemler manyetik bant sürücülerini istiyor olabilir ve beklediğiniz kadar çok sayıda boş manyetik bant sürücüsü bulunmayabilir.

- Kurtarma işlemi sırasında manyetik bant üzerindeki arşiv günlükleri, manyetik bant takılı olduğu anda kullanılabilir. DASD arşivleri kullanıldıysa ve sıradüzensel depolama yöneticisi (HSM) kullanılarak veri kümeleri manyetik banda geçirildiyse, HSM her bir veri kümesini diske anımsadığında bir gecikme olur. Arşiv günlüğü kullanılmadan önce veri kümelerini geri çağırabilirsiniz. Ancak, her zaman için gerekli olan doğru sıralamayı tahmin etmek mümkün değildir.
- DASD ' de arşiv günlükleri kullanılırken, birçok günlük gereklidir (yedekten geri yükleme işleminden sonra bir sayfa kümesini kurtarılan durum bu olabilir), tüm arşiv günlüklerini tutmak için önemli bir DASD miktarı gerekebilir.
- Düşük kullanım sisteminde ya da test sisteminde, manyetik bant bağlamaları gereksinmesini ortadan kaldırmak için arşiv günlüklerinin DASD ' de olması daha uygun olabilir.
- Her ikisi de RECOVER CFSTRUCT komutu yayınlayarak ve kalıcı bir iş birimini yedekledikten sonra, günlüğün geriye doğru okunmasına neden olur. Donanım sıkıştırması olan manyetik bant sürücüleri, geriye doğru okunan işlemlerde kötü performans sergiler. Manyetik banttıan geriye doğru okumaktan kaçınmak için, DASD üzerinde yeterli günlük verileri planlayın.

DASD ' ye arşivleme daha hızlı kurtarılabirlik sunar, ancak manyetik banta arşivlemeden daha karmaşıktır. İkili günlük kaydı kullanıyorsanız, arşiv günlüğünün birincil kopyasının DASD ' ye ve ikincil kopyada manyetik banda geçeceğini belirtebilirsiniz. Bu işlem, çok fazla DASD kullanmadan kurtarma hızını artırır ve manyetik bandı yedek olarak kullanabilirsiniz.

### Manyetik banda arşivleme

Bir manyetik bant aygıtına arşivlemek için seçerseniz, IBM MQ en fazla 20 birim uzatabilir.

Etkin günlük verilerinin büyüklüğünü, belirlenen bir manyetik bant biriminde olacak şekilde değiştirmeyi düşünüyorsanız, BSDS ' nin bir kopyasının, etkin günlük veri kümesinin kopyasıyla aynı manyetik bant birimine yerleştirildiği unutulmamasını unutmayın. Etkin günlük verilerinin büyüklüğünü, manyetik bant birimindeki BSDS için gereken alanı görelî konuma getirmek için aşağı doğru ayarlayın.

Manyetik bant üzerinde çift arşiv günlüğü kullanırsanız, bu, bir kopyasının yerel olarak tutulmasını ve diğer kopyanın olağanüstü durumdan kurtarma sırasında kullanılmak üzere açık bir yerde tutulabilmesini sağlar.

### DASD birimlerinde arşivleme

IBM MQ , manyetik bant dışı aygıtlarda (DASD) ayrılmış tüm arşiv günlüğü veri kümelerinin kataloğunu kataloğunuz için kataloğa almanızı gerektirir. DASD olarak arşivlenmeyi seçerseniz, CSQ6ARVP makrosu CATALOG parametresi YES (Evet) olmalıdır. Bu parametre NO ise ve arşiv günlüğü veri kümelerini DASD üzerine yerleştirmeye karar verdiyseniz, IBM MQ yine de veri kümesini kataloglasa da, arşiv günlüğü veri kümesi her ayrıldığında CSQJ072E iletisini alırsınız.

Arşiv günlüğü veri kümesi DASD üzerinde tutulursa, arşiv günlüğü veri kümeleri başka bir birime uzayabilir; çoklu birim desteklenmektedir.

DASD kullanmayı seçerseniz, birincil alan ayırmasının (hem miktar hem de blok büyüklüğü), etkin günlük veri kümesinden gelen verileri içerecek kadar büyük olduğundan ya da ikisini de karşılık gelen BSDS ' den (hangisi daha büyükse) içermeye yetecek büyüklükte olduğundan emin olun.

Bu işlem, boşaltma işlemi sırasında istenmeyen z/OS X ' B37 ' ya da X ' E37 ' olağandışı bitiş kodlarının olasılığına en aza indirir. Birincil alan ayırma işlemi, CSQ6ARVP makrosu için PRIQTY (birincil miktar) parametresiyle ayarlanır.

IBM MQ 8.0' tan, arşiv günlüğü veri kümeleri büyük ya da genişletilmiş biçimde sıralı veri kümelerinde bulunabilir. SMS ACS rutinleri artık DSNTYPE (büyük) ya da DSNTYPE (EXT) işlemlerini kullanabilir. Bunlar IBM MQ 8.0öncesinde desteklenmemişti.

IBM MQ , arşiv günlüklerinin genişletilmiş biçim veri kümeleri olarak ayrılmasını destekler. When extended format is used, the maximum archive log size is increased from 65535 tracks to the maximum active log size of 4GB. Arşiv günlükleri, genişletilmiş adres hacimlerindeki (EAV) genişletilmiş adresleme alanında (EAS) ayırma için uygun olur.

Gerekli donanım ve yazılım düzeylerinin olduğu yerlerde, arşiv günlükleri zEDC kullanılarak COMPACTION ile tanımlanan bir veri sınıfına ayrılıyor; arşiv günlüklerini tutmak için gereken disk saklama alanı azaltılabilir. Daha fazla bilgi için bkz. [IBM MQ for z/OS: IBM zEnterprise Data Compression ile depolama doluluğu azaltma \(zEDC\)](#).

Donanım ve yazılım düzeylerine ilişkin ayrıntılar için bkz. [zEnterprise Data Compression \(zEDC\) geliştirmelerinin kullanılması](#) , örneğin, RACF profili değişiklikleri.

z/OS veri kümesi şifreleme özelliği, IBM MQ 8.0 ya da sonraki bir yayın düzeyinde çalışan kuyruk yöneticilerine ilişkin arşiv günlüklerine uygulanabilir. Bu arşiv günlükleri, Otomatik Sınıf Seçimi (ACS) yordamlarıyla, EXTENDED öznitelikleriyle tanımlanmış bir veri sınıfına ve verilerin AES şifreli olmasını sağlayan bir veri kümesi anahtarı etiketiyle ayrılmalıdır.

## Arşiv günlüğü veri kümeleriyle SMS ' in kullanılması

MVS/DFP depolama yönetimi altsisteminiz ( DFSMS) kuruluysa, arşiv günlüğü veri kümeleriniz için Otomatik Sınıf Seçimi (ACS) kullanıcı çıkışı süzgeci yazabilirsiniz; bu süzgeç, SMS ortamı için bunları dönüştürmenize yardımcı olur.

Such a filter, for example, can route your output to a DASD data set, which DFSMS can manage. Bu şekilde bir ACS süzgeci kullanıyorsanız dikkatli olmanız gerekir. SMS, DASD veri kümelerinin kataloğa alınmasını gerektirdiğinden, CSQ6ARVP makrosu için CATALOG veri alanının YES (EVET) değerini içerdiğinden emin olun. Değilse, CSQJ072E iletisi döndürülür; ancak, veri kümesi IBM MQ tarafından kataloğa alınmaya devam eder.

ACS süzgeçleri hakkında daha fazla bilgi için bkz. [DFSMSshsm 'in dinamik olarak ayırdığı veri kümeleri](#) .

## Arşiv günlüklerinin depolama ortamını değiştirme

Arşiv günlükleri tarafından kullanılan depolama ortamının değiştirilmesine ilişkin yordam.

## Bu görev hakkında

Bu kısımda, arşiv günlükleri için kullanılan depolama ortamının nasıl değiştirileceği, örneğin arşivden manyetik banta arşivlemenin DASD ' ye nasıl arşivleneceği açıklanmaktadır.

Değişiklikleri nasıl yapacağınıza ilişkin bir seçiminiz var:

1. Değişiklikleri yalnızca CSQ6ARVP makrosunu kullanarak, kuyruk yöneticisinin sonraki yeniden başlatma zamanından itibaren uygulananlarla birlikte yapın.
2. CSQ6ARVP makrosunu kullanarak değişiklikleri yapın ve ARŞIV komutunu dinamik olarak kullanın. Bu, kuyruk yöneticisinin bir günlük dosyasını arşivlemesinin bir sonraki saatinden itibaren geçerli olduğu ve kuyruk yöneticisi yeniden başlatıldıktan sonra kalıcı olarak devam ettirdiği anlamına gelir.

## Yordam

1. Bu nedenle, arşiv günlüklerinin değiştirilmesi manyetik bant yerine DASD ' de saklanır:
  - a) "[Günlük arşivi depolamanızın planlanması](#)" sayfa 178 ' yi okuyun ve [CSQ6ARVP](#) parametrelerini gözden geçirin.
  - b) CSQ6ARVP içinde aşağıdaki parametrelerde değişiklik yapın
    - UNIT ' yi ve gerekiyorsa, UNIT2 parametrelerini güncelleyin.
    - DASD ' nin en iyi ayarı manyetik banttaki farklı olduğu için BLKSIZE parametresini güncelleyin.
    - PRIQTY ve SECQTY parametrelerini, etkin günlük ya da BSDS ' nin en büyüğünün tutabilmek için yeterince büyük olacak şekilde ayarlayın.
    - CATALOG parametresini YES (Evet) olarak ayarlayın.
    - ALCUNIT ayarını istediğiniz gibi onaylayın. BLK ' yi kullanmanız gerekir, çünkü bu, aygıt tipinden bağımsızdır.
    - Zaten değilse, ARCWTOR parametresini NO olarak ayarlayın.

2. Bu nedenle, arşiv günlüklerinin değiştirilmesi DASD yerine manyetik bant üzerinde saklanır:
- "Günlük arşivi depolamanızın planlanması" sayfa 178' yi okuyun ve CSQ6ARVP parametrelerini gözden geçirin.
  - CSQ6ARVP: içinde aşağıdaki deęiřtirgelerde deęiřiklik yapın.
    - UNIT ' yi ve gerekiyorsa, UNIT2 parametrelerini güncelleyin.
    - Manyetik bant için en iyi ayar DASD ' den farklı olduęu için BLKSIZE parametresini güncelleyin.
    - ALCUNIT ayarını istedięiniz gibi onaylayın. BLK ' yi kullanmanız gerekir, çünkü bu, aygıt tipinden bağımsızdır.
    - ARCWTOR parametresine ilişkin ayarı gözden geçirin.

### **z/OS Arşiv günlüklerini ne kadar süre tutmam gerekiyor?**

Yedekleme stratejinizi planlamanıza yardımcı olması için bu bölümdeki bilgileri kullanın.

You specify how long archive logs are kept in days , using the ARCRETN parameter in USING CSQ6ARVP or the SISTEM AYARLA command. Bu sürenin sonunda veri kümeleri z/OS tarafından silinebilir.

Artık gerek kalmadığında arşiv günlüğü veri kümelerini el ile silebilirsiniz.

- Kuyruk yöneticisi, kurtarma için arşiv günlüklerine gereksinim duyabilir.

Kuyruk yöneticisi en son 1000 arşivini yalnızca BSDS 'de tutabilir; arşiv günlükleri BSDS' de bulunmuyorsa, kurtarma için kullanılamaz ve yalnızca denetleme, çözümleme ya da yeniden yürütme tipi amacıyla kullanılır.

- Günlüklerden bilgi ayıklamak için arşiv günlüklerini tutmak isteyebilirsiniz. Örneğin, günlükten iletilerin alınması ve iletinin hangi kullanıcı kimliğinin konulması ya da g " nderileceęi gözden geçirilmesi.

BSDS, günlükler ve dięer kurtarma bilgileri hakkında bilgi içerir. Bu veri kümesi sabit bir boyuttur. Arşiv günlüklerinin sayısı CSQ6LOGP' da MAXARCH deęerine ulařtıęında ya da BSDS doldurulduęunda, en eski arşiv günlüğü bilgilerinin üzerine yazılır.

Arşiv günlüğü girişlerini BSDS ' den kaldırmak için yardımcı programlar vardır, ancak genel olarak BSDS en eski arşiv günlüğü kaydına sarılır ve bu kayıtle örtüşmektedir.

### **Arşiv günlüğü ne zaman gerekli**

Sayfa setlerinizi düzenli olarak yedeklemeniz gerekir. Yedekleme sıklıęı, bir sayfa kümesinin kaybolması durumunda hangi arşiv günlüklerinin gerekli olduęunu belirler.

CF yapılarınızı düzenli olarak yedeklemeniz gerekir. Yedeklerin sıklıęı, CF yapısındaki verileri kaybetmek durumunda hangi arşiv günlüklerinin gerektięini belirler.

Kurtarma için arşiv günlüğü gerekli olabilir. Ařaęıdaki bilgiler, farklı IBM MQ kaynaklarında sorunların olduęu arşiv günlüęünün ne zaman gerekebileceęini açıklar.

#### **Sayfa kümesi kaybı**

Sisteminizi yedeęinizden kurtarmalı ve kuyruk yöneticisini yeniden başlatmalısınız.

Yedekleme işleminden önce, yedeklemenin yanı sıra, yedeklemeden önce en çok üç günlük veri kümesini de içeren günlüklere gereksiniminiz vardır.

#### **Tüm LPAR ' lar bir CF yapısıyla baęlanırlıęı kaybeder ya da yapı kullanılamaz**

Yapıyı kurtarmak için RECOVER CFSTRUCT komutunu kullanın.

Yapı kurtarma işlemi, son yedeklemeden bu yana yapıya eriřmiř olan tüm kuyruk yöneticilerinden günlükleri gerektirir (yedeklemenin alındıęı zamana geri) artı, yedeklemeyi alan kuyruk yöneticisinin kendisinde bulunan yapı yedeklenmektedir.

CF yapılarının sık sık yedeklenmesini gerçekleřtirdiyse, veriler etkin günlüklerde olmalı ve arşiv günlüklerine gerek olmamanız gerekir.

CF yapısının yeni bir yedeęi yoksa, arşiv günlüklerine gereksinim duyabilirsiniz.

**Not:** Kalıcı olmayan tüm iletiler kaybedilir; tüm kalıcı iletiler aşağıdaki görevleri yerine getirilerek yeniden yaratılacaktır:

1. Günlüğüten son CF yapısı yedeklemesi okunuyor
2. Yapıyı kullanan tüm kuyruk yöneticilerinden günlükler okunuyor
3. Yedeklemeden bu yana güncellemelerin birleştirilmesi

### **Yönetim yapısı yeniden oluşturuluyor**

Denetim yapısını yeniden oluşturmanız gerekirse, bilgiler, QSG ' deki her kuyruk yöneticisi için günlüğün son denetim noktasından okunur.

Kuyruk yöneticisi etkin değilse, QSG ' deki başka bir kuyruk yöneticisi günlüğü okur.

Arşiv günlüklerine gereksinim olmamanız gerekir.

### **SMDS veri kümesinin kaybı**

Bir SMDS veri kümesini kaybederseniz ya da veri kümesi bozulursa, veri kümesi kullanılamaz duruma gelir ve bu durum için durum BAŞARILI olarak ayarlanır. CF yapısı değiştirilmedi.

SSMDS veri kümesini geri yüklemek için aşağıdaki başlara gereksinim duyarsınız:

1. SMDS veri kümesini yeniden tanımlayın ve
2. CFSTRUCT ' U KURTAR komutunu vererek CF yapısını kurtarın.

**Not:** CF yapısındaki tüm kalıcı olmayan iletiler kaybedilir; tüm kalıcı iletiler geri yüklenir.

Kuyruk yöneticisi günlüklerinin gereksinmesi, kullanılamayan bir yapıdan kurtarmanın aynısıdır.

### **z/OS Adreslenebilir günlük aralığı üst sınırını artırmayı planlama**

Daha büyük bir günlük görelî byte adresi (RBA) kullanmak için kuyruk yöneticinizi yapılandırarak, adreslenebilir günlük aralığını üst sınırını artırabilirsiniz.

IBM MQ 8.0 için log RBA değişikliğine ilişkin genel bakış için bkz. [Larger log relative Byte Address](#).

**V 9.1.0** Kuyruk yöneticisi bir kuyruk paylaşım grubunda değilse, bunu istediğiniz zaman 8 baytlık günlük RBA değerlerini kullanmak üzere dönüştürebilirsiniz. Daha sonra IBM MQ 8.0.0 ya da 9.0.0' a geri dönseniz, bu yayınlarda **OPMODE=NEWFUNC** komutunu kullandığınızdan emin olun, tersi durumda kuyruk yöneticisi başlatılmaz.

Kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisinin 8 baytlık günlük RBA kullanacak şekilde dönüştürülebilmesi için, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinin aşağıdaki düzeylerden birinde olması gerekir:

- At IBM MQ 9.0.n CD, IBM MQ 9.1.0 LTS, or later
- IBM MQ 9.0.0 ve **OPMODE= (NEWFUNC,800)** ya da **OPMODE= (NEWFUNC,900)** ile başlanmıştır.
- IBM MQ 8.0.0 ' ta ve **OPMODE= (NEWFUNC,800)** ile başlanmıştır.
- Ya da IBM MQ 8.0.0 ya da 9.0.0saatinde kuyruk paylaşım grubuna eklendi

Daha sonra, her kuyruk yöneticisini, 8 baytlık günlük RBA değerlerini kullanmak üzere değiştirebilirsiniz. Tüm kuyruk yöneticilerini aynı anda değiştirmek şart değildir.

Bir kuyruk paylaşım grubundaki bir kuyruk yöneticisi 8 baytlık günlük RBA değerlerini kullanmak üzere dönüştürüldüğünde, kuyruk paylaşım grubundaki diğer kuyruk yöneticileri, henüz 8 baytlık günlük RBA değerlerini kullanmak üzere dönüştürülmemiş olsa da, dönüştürülen kuyruk yöneticisinin günlüklerini kullanabilir. Bu, örneğin, eş kurtarma için yararlı olur.

### **Değişikliği geri al**

Değişiklik geri verilemez.

## -Ne kadar sürer?

Değişiklik, kuyruk yöneticisi yeniden başlatılmasını gerektiriyor. Kuyruk yöneticisini durdurun, yeni veri kümeleri yaratmak için, önyükleme veri kümesi (BSDS) ya da veri kümeleri için CSQJUCNV yardımcı programını çalıştırın, bu önyükleme veri kümelerini yeniden adlandırın ve kuyruk yöneticisini yeniden başlatın. CSQJUCNV yardımcı programı çalıştırılması için genellikle birkaç saniye sürer.

## Bunun etkisi ne?

- 8 baytlık günlük RBA kullanımında, günlük veri kümelerine ilişkin her veri yazma işlemi ek baytlara sahiptir. Bu nedenle, kalıcı iletilerden oluşan bir iş yükü için, günlüklere yazılan veri miktarında küçük bir artış olur.
- Bir sayfa kümesine ya da bağlaşım tesisine (CF) yazılan veriler etkilenmez.

## İlgili görevler

[Daha büyük günlük Görelî Bayt Adresini uygulama](#)

z/OS

## Kanal başlatıcınızın planlanması

Kanal başlatıcı, kuyruk yöneticileri arasında iletişim sağlar ve kendi adres alanında çalışır.

İki tip bağlantı vardır:

1. Ağ üzerinden kuyruk yöneticisine uygulama bağlantıları. Bunlar istemci kanalları olarak bilinir.
2. Kuyruk yöneticisi için kuyruk yöneticisi bağlantıları. Bunlar MCA kanalları olarak bilinir.

## Dinleyiciler

Bir kanal dinleyici programı, gelen ağ isteklerini dinler ve kanal gerektiğinde uygun kanalı başlatır. Gelen bağlantıları işlemek için, kanal başlatıcısının yapılandırılan en az bir IBM MQ dinleyici görevi olması gerekir. Dinleyici, bir TCP dinleyicisi ya da LU 6.2 dinleyicisi olabilir.

Her dinleyici bir TCP kapısı ya da LU adı gerektirir. IBM MQ for Multiplatforms genellikle 1414 numaralı TCP/IP kapısını kullanır (varsayılan değer).

Her kanal başlatıcısı için birden çok dinleyiciye sahip olabilirsiniz.

## TCP/IP

Bir kanal başlatıcı, aynı z/OS görüntüsünde birden çok TCP yığını kullanarak çalışabilir. Örneğin, bir TCP yığını iç bağlantılar için olabilir ve dış bağlantılar için başka bir TCP yığını olabilir.

Bir çıkış kanalı tanımladığınızda:

1. Bağlantının hedef anasistemini ve bağlantı noktasını ayarladınız. Bu işlem aşağıdakilerden biri olabilir:

- bir IP adresi; örneğin, 10.20.4.6
- bir anasistem adı; örneğin, mvs-prod.myorg.com

Hedefi belirtmek için bir anasistem adı kullanırsanız, IBM MQ hedefin IP adresini çözümlenmek için Etki Alanı Ad Sistemi 'ni (DNS) kullanır.

2. Birden çok TCP yığını kullanıyorsanız, kullanılacak IP Yığını adresini belirten kanal tanımındaki **LOCLADDR** parametresini belirtebilirsiniz.

Yüksek düzeyde kullanılabilir bir DNS sunucusuna ya da DNS sunucularına sahip olmak için plan yapmak gerekir. DNS kullanılmıyorsa, giden kanallar başlatılamayabilir ve bir anasistem adı kullanılarak gelen bir bağlantıyı eşleyen kanal doğrulama kuralları işlenemez.

## APPC ve LU 6.2

APPC kullanıyorsanız, kanal başlatıcısında bir LU adı ve APPC ' de yapılanış gereklidir.



## Kuyruk paylaşım grupları

Tek bir sistem görüntüsü sağlamak ve gelen bir IBM MQ bağlantısı isteğinin kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisine gitmesine izin vermek için bazı yapılandırlara gerek vardır. Örneğin:

1. Bir donanım ağı yönlendiricisi. Bu yönlendirici, kuruluş tarafından görülen bir IP adresine sahiptir ve ilk isteği, bu donanıma bağlı olan herhangi bir kuyruk yöneticisine yönlendirebilir.
2. Sanal IP adresi (VIPA). Kurumsal geniş bir IP adresi belirtilir ve bu adres, bir sistem şebekesinde bulunan TCP yığınlarından herhangi birine yönlendirilebilir. Daha sonra, TCP yığını, bunu sistem şebekesi içindeki herhangi bir dinleme kuyruğu yöneticisine yönlendirebilir.

## IBM MQ trafiğini koruma

You can configure IBM MQ to use TLS (or SSL) connections to protect data on the wire. TLS ' yi kullanmak için dijital sertifikalar ve anahtar halkaları kullanmanız gerekir.

You also need to work with the personnel at the remote end of the channel, to ensure that you have compatible IBM MQ definitions and compatible certificates.

You can control which connections can connect to IBM MQ and the user ID, based on

- IP adresi
- İstemci kullanıcı kimliği
- Uzak kuyruk yöneticisi ya da
- Dijital sertifika (bkz. [Kanal Kimlik Doğrulama Kayıtları](#))

Ayrıca, geçerli bir kullanıcı kimliği ve parola belirtmelerini sağlayarak istemci uygulamalarını kısıtlamak da mümkündür (bkz. [Bağlantı Kimlik Doğrulaması](#)).

Kanal başlatıcı çalışmaya devam edebilir ve her bir kanalı bir kerede bir tane TLS kullanacak şekilde yapılandırabilirsiniz.

## Kanal başlatıcının izlenmesi

Kanal başlatıcı ve kanallarıyla ilgili bilgi veren MQSC komutları vardır:

- [DISPLAY CHINIT](#) komutu, kanal başlatıcısı ve etkin dinleyicilere ilişkin bilgi verir.
- [DISPLAY CHSTATUS](#) komutu, bir kanala ilişkin etkinliği ve durumu görüntüler.

Kanal başlatıcı, kanal başlatıcı görevleri ve kanal etkinlikleriyle ilgili bilgilerle SMF kayıtları da üretebilir. Ek bilgi için "[Kanal başlatıcı SMF verilerinin planlanması](#)" sayfa 185 başlıklı konuya bakın.

Kanal başlatıcısı, kanalları başlatma ve durdurma sırasında iş günlüğüne ileti iletir. İşletmeinizdeki otomasyon, durumu yakalamak için bu iletileri kullanabilir. Bazı kanallar yalnızca birkaç saniye etkinken, birçok ileti üretilebilir. z/OS ileti işleme olanağını kullanarak ya da [SET SYSTEM](#) komutuyla **EXCLMSG** ayarını kullanarak bu iletileri gizleyebilirsiniz.

## IBM MQ kanal tanımlarınızın yapılandırılması

Birden çok kuyruk yöneticisi birbirine bağlandığında, tüm nesne tanımlamalarını yönetmek zor olabilir. IBM MQ kümelemenin kullanılması bunu basitleştirebilir.

Tam havuz olarak iki kuyruk yöneticisi belirtmenizi sağlar. Diğer kuyruk yöneticilerinin, havuzlardan biri ile bir bağlantısı arasında bir bağlantı ve bir bağlantı olması gerekir. Diğer kuyruk yöneticilerine yönelik bağlantılar gerektiğinde, kuyruk yöneticisi kanalları otomatik olarak yaratır ve başlatır.

Bir kümede çok sayıda kuyruk yöneticisi bulunmayı planlıyorsanız, adanmış havuzlar olarak hareket eden ve uygulama trafiği olmayan kuyruk yöneticilerine sahip olmak için plan yapmayı planlamanız gerekir.

Ek bilgi için "[Dağıtılmış kuyruklarınızın ve kümelerinizin planlanması](#)" sayfa 19 başlıklı konuya bakın.



## Kanal başlatıcısını yapılandırmadan önce yapılması gereken işlemler

1. TCP/IP ya da APPC kullanıp kullanmadığınızı belirleyin.
2. TCP kullanıyorsanız, IBM MQ için en az bir kapı ayırın.
3. Bir DNS sunucusuna gereksiniminiz varsa, gerekirse sunucuyu yüksek kullanılabilirlikli olacak şekilde yapılandırın.
4. APPC kullanıyorsanız, bir LU adı ayırın ve APPC ' i yapılandırın.

## Kanal başlatıcıyı yapılandırdıktan sonra, üretime geçmeden önce işlemler

1. Hangi bağlantıların sahip olacağını planlayın:
  - a. Uzak uygulamalardaki istemci bağlantıları.
  - b. Diğer kuyruk yöneticilerine ve diğer kuyruk yöneticilerine mCA kanalları. Genellikle, her uzak kuyruk yöneticisinden ve her uzak kuyruk yöneticisinden bir kanalınız olur.
2. Kümeleme işlemini ayarlayın ya da var olan bir kümeleme ortamına katılın.
3. Kanal başlatıcının önünde kullanılabilirlik için birden çok TCP yığını, VIPA ya da bir dış yönlendirici kullanmanız gerekip gerekmediğini göz önünde bulundurun.
4. TLS kullanmayı planlıyorsanız:
  - a. Anahtar halkasının ayarlanması
  - b. Sertifikaları ayarlama
5. Kanal kimlik doğrulamasını kullanmayı planlıyorsanız:
  - a. Gelen oturumları MCA kullanıcı kimlikleriyle eşlemeye ilişkin ölçütlere karar verin
  - b. Enable reverse DNS lookup by setting the queue manager parameter **REVDNS**
  - c. Güvenliği gözden geçirin. Örneğin, varsayılan kanalları silin ve bir kanala ilişkin **MCAUSER** özniteisinde yalnızca gerekli yetkiyi içeren kullanıcı kimliklerini belirtin.
6. Kanal başlatıcı tarafından üretilen muhasebe ve istatistik SMF kayıtlarını yakalayın ve bunları gönderin.
7. İş günlüğü iletilerinin izlenmesini otomatikleştirin.
8. Gerekliyse, veri çıkışını iyileştirmek için ağ ortamınızı ayarlayın. TCP, büyük gönderme ve alma arabellekleri, üretilen işi artırır. MQ komutunu aşağıdaki komutları kullanarak belirli TCP arabellek büyüklüklerini kullanmaya zorlayabilirsiniz:

```
RECOVER QMGR(TUNE CHINTCPBDYNSZ nnnnn)  
RECOVER QMGR(TUNE CHINTCPSBDYNSZ nnnnn)
```

which sets the SO\_RCVBUF, and SO\_SNDBUF, for the channels to the size in bytes specified in nnnnn.

### İlgili kavramlar

[“Kuyruk yöneticinizin planlanması” sayfa 157](#)

Bir kuyruk yöneticisi ayarlarken, kuyruk yöneticisinin kuruluşunuzun gereksinimlerini karşıladığı için, planlama yöneticisi kuyruk yöneticisinin büyümesine izin vermelidir.

## Kanal başlatıcı SMF verilerinin planlanması

Kanal başlatıcısı için SMF verilerinin toplanmasının uygulanmasını planlamanız gerekir.

Kanal başlatıcı iki tip kayıt üretir:

- Kanal başlatıcısı ve içindeki görevler hakkında bilgi içeren istatistik verileri.
- [DURUMU GÖRÜNTÜLE](#) komutuna benzer bilgiler içeren kanal hesap verileri.

Şu komutu kullanarak istatistik verilerini toplamaya başlarsınız:

```
START TRACE(STAT) CLASS(4)
```

ve komutu kullanarak komutu durdurun:

```
STOP TRACE(STAT) CLASS(4)
```

Şu komutu kullanarak muhasebe verilerini toplamaya başlarsınız:

```
START TRACE(ACCTG) CLASS(4)
```

ve komutu kullanarak komutu durdurun:

```
STOP TRACE(ACCTG) CLASS(4)
```

Kanal tanımlamasında ya da kuyruk yöneticisinde **STATCHL** özniteliğini kullanmak için hangi kanalların hesap toplandığı kanalların denetlenmesini denetleyebilirsiniz.

- İstemci kanalları için, kuyruk yöneticisi düzeyinde **STATCHL** seçeneğini ayarlamanız gerekir.
- Otomatik olarak tanımlanmış küme gönderen kanalları için, muhasebe verilerinin toplanmasını **STATACLS** kuyruk yöneticisi öznitelikle denetleyebilirsiniz.

Kuyruk yöneticisi için varsayılan **STATCHL** değeri OFF' dir. In order to collect channel accounting data you must change the value of **STATCHL** from the default on either the queue manager or channel definition, in addition to starting class 4 accounting trace.

Aşağıdaki durumlarda SMF kayıtları üretilir:

- The time interval indicated by the CSQ6SYSP **STATIME** parameter has elapsed, or if **STATIME** is zero, on the SMF data collection broadcast. Kanal başlatıcı ve kuyruk yöneticisi için SMF verilerini toplama isteği uyumlulaştırılır.
- Bir STOP TRACE(ACCTG) CLASS(4) ya da STOP TRACE(STAT) CLASS(4) komutu verilir ya da
- Kanal başlatıcı kapatılıyor. Bu noktada, herhangi bir SMF verisi yazılıdır.

Bir kanal SMF aralığı sırasında durursa, bir sonraki SMF işlemi çalıştırıldığında muhasebe verileri SMF ' ye yazılır. Bir istemci bağlanıyorsa, bazı işler yapar ve bağlantı kurar, sonra yeniden bağlanır ve bağlantı kesilir, iki kanal muhasebesi verisi kümesi vardır.

İstatistik verileri genellikle tek bir SMF kaydına sığabiliyor, ancak çok sayıda görev kullanımdaysa birden çok SMF kaydı yaratılabilir.

Muhasebe verileri, etkinleştirildiği her kanal için toplanır ve genellikle tek bir SMF kaydına sığdır. Ancak, çok sayıda kanal etkin olduğunda birden çok SMF kaydı yaratılabilir.

Kanal başlatıcı SMF verilerinin toplanması maliyeti küçüktür. Genellikle CPU kullanımında artış yüzde kaçtır ve çoğu zaman ölçüm hatası içerisindedir.

Before you use this function you need to work with your z/OS systems programmer to ensure that SMF has the capacity for the additional records, and that they change their processes for extracting SMF records to include the new SMF data.

Kanal başlatıcı istatistikleri verileri için, SMF kayıt tipi 115 ve alt tipi 231.

Kanal başlatıcı hesap verileri için, SMF kayıt tipi 116 ve alt tip 10 'tır.

Bu verileri işlemek için kendi programlarınızı yazabilir ya da bir program içeren SupportPac MP1B 'yi, verileri yazdırmak için MQSMF' yi ve bir dağıtma sayfasına içe aktarmak için uygun olan CSV (Comma Separated Values; Virgülle Ayrılmış Değerler) biçiminde veri yaratıyorsanız, bu programdan birini kullanabilirsiniz.

Kanal başlatıcı SMF verilerini yakalama ile ilgili sorunlar yaşıyorsanız, ek bilgi için [Kanal başlatıcı için SMF verilerini yakalarken \(CHINIT\) sorunlarla başa çıkılmasına](#) bakın.

## İlgili görevler

[IBM MQ performans istatistiklerinin yorumlanması](#)

[Kanal hesaplama verileri sorun giderme](#)

## z/OS z/OS TCP/IP ortamınızın planlanması

Ağınızda en iyi iş oranını elde etmek için, 64 KB ya da daha büyük boyutlarda TCP/IP gönderme ve alma arabellekleri kullanmanız gerekir. Bu büyüklüğe sahip sistem, arabellek büyüklüklerini en iyi duruma getirir.

Bkz. [High Latency Networks için Dinamik Sağ Boyutlandırma Nedir?](#) daha fazla bilgi için.

Aşağıdaki Netstat komutunu kullanarak sistem arabelleği büyüklüğünü denetleyebilirsiniz, örneğin:

```
TSO NETSTAT ALL (CLIENT csq1CHIN
```

Sonuçlar, aşağıdaki iki değer de içinde olmak üzere, çok fazla bilgi görüntüler:

```
ReceiveBufferSize: 0000065536  
SendBufferSize: 0000065536
```

65536, 64 KB ' dir. Arabellek büyüklükleriniz 65536 'dan küçükse, TCP/IP yordamında PROFILE DDName içindeki **TCPSENDBFRSIZE** ve **TCPRCVBUFRSIZE** değerlerini artırmak için ağ ekibinizle birlikte çalışmanız gerekir. Örneğin, aşağıdaki komutu kullanabilirsiniz:

```
TCPCONFIG TCPSENDBFRSIZE 65536 TCPRCVBUFRSIZE 65536
```

Sistem genelindeki **TCPSENDBFRSIZE** ya da **TCPRCVBUFRSIZE** ayarlarınızı değiştiremiyorsanız, IBM Yazılım Destek merkezimize başvurun.

## z/OS Kuyruk paylaşım grubunuzun planlanması (QSG)

Paylaşılan bir kuyruğa alma ortamını gerçekleştirmenin en kolay yolu, bir kuyruk yöneticisi yapılandırmasıdır, o kuyruk yöneticisini bir QSG 'ye ekledikten sonra, QSG' ye diğer kuyruk yöneticilerini de ekleyinir.

Kuyruk paylaşım grubu, yapılandırma bilgilerini saklamak için Db2 tablolarını kullanır. Aynı Db2 veri paylaşma grubunu paylaşan tüm QSG ' ler tarafından kullanılan bir tablo kümesi vardır.

Paylaşılan kuyruk iletileri, bir bağlaşımlı tesisinde (CF) bir yapıda saklanır. Her bir QSG ' nin kendi CF yapıları kümesi vardır. Yapılarınızı gereksinimlerinizi karşılayacak şekilde yapılandırmanız gerekir.

İletiler, Paylaşılan İleti Veri Kümelerinde de saklanabilir (SMDS). Message over 63KB in size cannot be stored in the CF. Bu iletileri paylaşılan kuyruklarda saklamak için SMDS ' yi kullanmanız gerekir.

## İleti profilleri ve kapasite planlaması

Paylaşılan kuyruk iletilerinizin ileti tanıtımını anlamamız gerekir. Aşağıda, göz önünde bulundurmanız gereken etkenler şunlardır:

- Ortalama ve ileti boyutu üst sınırı
- Tipik kuyruk derinliği ve kural dışı durum kuyruğu derinliği. Örneğin, iletileri bir gün boyunca tutmak için yeterli kapasitenin olması ve standart kuyruk derinliğinin 100 ileti altında olması gerekebilir.

İleti profili değişirse, daha sonraki bir tarihte yapıların boyutunu artırabilir ya da SMDS uygulayabilirsiniz.

Büyük bir yoğun ileti hacmiyle başa çıkabilmek istiyorsanız, yapı kullanımı kullanıcı tarafından belirlenen eşiklere ulaştığında, IBM MQ 'u SMDS' ye yönelik iletileri boşaltmak için yapılandırabilirsiniz.

CF yapılarını çift yönlü olarak görmek istediğinize karar vermeniz gerekir. Bu, CFRM ilkesinde CF yapısı tanımlaması tarafından denetlenir:

1. Çift yönlü bir yapı iki bağlaşım olanağını kullanır. Bir CF 'yle ilgili bir sorun varsa, hizmette herhangi bir kesinti yoktur ve varsa, yapı üçüncü CF' de yeniden oluşturulabilir. Çift yönlü yapılar, paylaşılan kuyruklar üzerindeki operasyonların performansını önemli ölçüde etkileyebilir.
2. Yapı çift yönlü değilse, CF 'yle ilgili bir sorun, bu CF' deki yapılara ilişkin paylaşılan kuyrukların, yapı başka bir CF ' de yeniden oluşturulabilinceye kadar kullanılmayacağını belirtir.

IBM MQ , bu vakadaki başka bir CF ' deki yapıları otomatik olarak yeniden oluşturmak için yapılandırılabilir. Kalıcı iletiler, kuyruk yöneticilerinin günlüklerinden kurtarılacaktır.

CF tanımlamalarını değiştirmenin kolay olduğunu unutmayın.

Kalıcı ve kalıcı olmayan iletileri tutabilmesi için bir yapı tanımlayabilir ya da kalıcı olmayan iletileri tutabilecek şekilde tanımlayabilirsiniz.

Kalıcı iletileri tutabilen yapıların düzenli aralıklarla yedeklenmeleri gerekir. Bir hata durumunda yapının kurtarılması için gereken süreyi en aza indirmek için CF yapılarınızı en az her saat yedeklemeniz gerekir. Yedek, yedeklemeyi gerçekleştiren kuyruk yöneticisinin günlük verileri kümesinde saklanır.

Paylaşılan kuyruklarınızda iletilerin yüksek verimi olmasını bekliyorsanız, CF yapılarınızı yedeklemek için özel olarak ayrılmış bir kuyruk yöneticisine sahip olmak en iyi uygulamadır. Bu işlem, daha az veri kuyruk yöneticisi günlüklerinden okunabilmek için, yapıların kurtarılması için gereken süreyi kısaltır.

## Kanallar

IBM MQ QSG ' ye bağlanan uygulamalar için tek bir sistem görüntüsü sağlamak için paylaşılan giriş kanallarını tanımlayabilirsiniz. Bunlar ayarlandıysa, kuyruk paylaşım grubu ortamına bir bağlantı geldiyse, QSG ' deki kuyruk yöneticisine gidebilir.

Bu kanallara ilişkin bir ağ yönlendiricisi ya da sanal IP adresi (VIPA) ayarlamamız gerekebilir.

Paylaşılan çıkış kanallarını tanımlayabilirsiniz. Paylaşılan bir çıkış kanalı eşgörünümü, QSG ' deki herhangi bir kuyruk yöneticisinden başlatılabilir.

Ek bilgi için [Paylaşılan kanallar](#) konusuna bakın.

## Güvenlik

You protect IBM MQ resources using an external security manager. RACF kullanıyorsanız, RACF profillerine önek olarak kuyruk yöneticisi adı eklenir. Örneğin, APPLICATION.INPUT , qmgName . APPLICATION . INPUT . adlı MQQUEUE sınıfındaki bir tanımlama kullanılarak korunacaktır

Bir kuyruk paylaşım grubu kullanırken, kuyruk yöneticisi adıyla önekli profillerle kaynakları korumaya devam edebilir ya da kuyruk paylaşım grubu adı ile profiller öneğine devam edebilirsiniz. Örneğin, qsgName . APPLICATION . INPUT .

Kuyruk paylaşım grubu adı ile profil öneğini kullanmayı hedeflemelisiniz; bu, tüm kuyruk yöneticileri için tek bir tanımlama olduğu anlamına gelir, sizi kurtarmanızı sağlar ve kuyruk yöneticileri arasındaki tanımlarda bir uyumsuzluğu önlemelidir.

## İlgili kavramlar

[“Kuyruk yöneticinizin planlanması” sayfa 157](#)

Bir kuyruk yöneticisi ayarlarken, kuyruk yöneticisinin kuruluşunuzun gereksinimlerini karşıladığı için, planlama yöneticisi kuyruk yöneticisinin büyümesine izin vermelidir.

## Bağlantı tesisinizi planlama ve depolama ortamını boşaltma

İlk boyutları ve bağlaşım olanağı (CF) yapılarınızın biçimlerini ve paylaşılan ileti veri kümesi (SMDS) ortamını ya da Db2 ortamını planlarken bu konuyu kullanın.

Bu bölüm, aşağıdaki konularla ilgili bilgi içerir:

- [“Bağlaşım olanağı kaynaklarının tanımlanması” sayfa 189](#)

- Boşaltma depolama düzeneğinin kararlaştırılması
- Yapılarınızın planlanması
- Yapılarınızın boyutunun planlanması
- Paylaşılan kuyrukların yapılara eşlenmesi
- “Paylaşılan ileti veri kümesi (SMDS) ortamınızın planlanması” sayfa 194
- “Db2 ortamınızın planlanması” sayfa 197

## **Bağlaşım olanağı kaynaklarının tanımlanması**

Paylaşılan kuyrukları kullanmayı amaçlıyorsanız, IBM MQ ' un CFRM ilkenizde kullanacağı bağlaşım olanağı yapılarını tanımlamanız gerekir. Bunu yapmak için önce CFRM ilkenizi yapılarla ilgili bilgilerle güncellenmeniz ve daha sonra ilkeyi etkinleştirmeniz gerekir.

Kuruluşunuz, kullanılabilir bağlaşım olanaklarını açıklayan bir CFRM ilkesine sahip olabilir. Denetim verileri yardımcı programı , sağladığınız metinli deyimlere dayalı olarak ilkenin içeriğini değiştirmek için kullanılır. Yeni yapıların adlarını, tanımlandıkları bağlaşım olanaklarını ve yapıların boyutunu tanımlayan ilkeye deyimler eklemeniz gerekir.

CFRM ilkesi, IBM MQ yapılarının çift yönlü olup olmadığını ve bunların hata senaryolarında nasıl yeniden tahsis edileceğini de belirler. Paylaşılan kuyruk kurtarma , bağlaşım olanağını etkileyen hatalara dayanıklılık için CFRM ' nin yapılandırılmasına ilişkin öneriler içerir.

## **Boşaltma depolama ortamınızı belirleme**

Paylaşılan kuyruklara ilişkin ileti verileri, bağlaşım olanağından boşaltılabilir ve Db2 çizelgesinde ya da *paylaşılan ileti veri kümesi* (SMDS) adı verilen IBM MQ yönetilen veri kümesinde saklanabilir. Bağlaşım tesisinde (yani 63 KB ' den büyük) depolanamayacak kadar büyük olan iletiler her zaman boşaltılmalıdır ve daha küçük iletiler isteğe bağlı olarak, bağlaşım olanağı alan kullanımını azaltmak için boşaltılabilir.

Daha fazla bilgi için Paylaşılan iletiler için boşaltma seçeneklerini belirtme başlıklı konuya bakın.

## **Yapılarınızın planlanması**

Bir kuyruk paylaşım grubu (QSG), en az iki yapının tanımlanmasını gerektirir. Yönetim yapısı olarak bilinen ilk yapı, kuyruk paylaşım grubu genelinde IBM MQ iç etkinliğini koordine etmek için kullanılır. Bu yapıda kullanıcı verileri tutulmaz. Değişmez adı *qsg-nameCSQ\_ADMIN* (burada *qsg-name* kuyruk paylaşım grubunuzun adıdır). Sonraki yapılar uygulama yapıları olarak bilinir ve iletileri IBM MQ paylaşılan kuyruklarında tutmak için kullanılır. Her yapı en çok 512 paylaşılan kuyruk tutabilir.

Sistem kuyrukları için *qsg-nameCSQSYSAPPL* adlı bir uygulama yapısı kullanılır. Bu yapının tanımlanması isteğe bağlıdır, ancak belirli özelliklerin kullanılması için gereklidir. Varsayılan olarak, SYSTEM.QSG.CHANNEL.SYNCQ ve SYSTEM.QSG.UR.RESOLUTION.QUEUE kuyrukları, *qsg-nameCSQSYAPPL* yapısında tanımlanır.

### **Birden çok yapı kullanma**

Bir kuyruk paylaşım grubu, en çok 64 bağlaşım olanağı yapısına bağlanabilir. Bu yapılardan biri yönetim yapısı olmalıdır. Tanımlanmışsa, bu yapılardan biri *qsg-nameCSQSYSAPPL* yapısı olabilir. İleti verileri için en çok 63 ( *qsg-nameCSQSYSAPPL* tanımlandıysa 62) yapı kullanabilirsiniz. Aşağıdaki nedenlerden ötürü birden çok uygulama yapısı kullanmayı seçebilirsiniz:

- Büyük olasılıkla çok sayıda ileti bulunduracak kuyruklarınız var ve bu nedenle tüm bağlaşım olanağının tüm kaynaklarını gerektirir.
- Çok sayıda paylaşılan kuyruk için bir gereksiniminiz vardır, bu nedenle her bir yapı yalnızca 512 kuyruk içerebileceği için bu kuyruklar birden çok yapıya bölünmelidir.

- RMF, bir yapının kullanım karakteristiğine ilişkin raporlar, içerdiği kuyrukları bir dizi bağlaşım olanağı arasında dağıtmanızı önerir.
- Veri yalıtma nedeniyle, bazı kuyruk verilerinin diğer kuyruk verilerinden fiziksel olarak farklı bir bağlaşım olanağı içinde tutulmasını istiyorsunuz.
- Kalıcı paylaşılan iletilerin kurtarılması, yapı düzeyinde öznitelikler ve komutlar (örneğin, BACKUP CFSTRUCT) kullanılarak gerçekleştirilir. Yedekleme ve kurtarmayı basitleştirmek için, kalıcı olmayan iletileri tutan kuyrukları, kalıcı iletileri tutan yapılardan farklı yapılara atayabilirsiniz.

Yapıların hangi bağlanti tesislerinde ayrılması gerektiğini seçerken, aşağıdaki noktaları göz önünde bulundurun:

- Veri yalıtma gereksinimleriniz.
- Bağlanti tesisinin uçuculuğu (yani, bir güç kesintisi yoluyla verileri koruma yeteneği).
- Erişim sistemleri ve bağlaşım tesisi arasındaki ya da bağlaşım tesisleri arasındaki arızanın bağımsızlığı.
- Bağlaşım olanağına kurulan bağlaşım olanağı denetim kodu (CFCC) düzeyi ( IBM MQ için Düzey 9 ya da üstü gerekir).

## Yapılarınızın boyutunun planlanması

### Yönetim yapısı

Denetim yapısı (*qsg-name*CSQ\_ADMIN), kuyruk paylaşım grubundaki her kuyruk yöneticisi için 1000 liste girişi içerecek büyüklükte olmalıdır. Bir kuyruk yöneticisi başlatıldığında, kuyruk paylaşım grubuna *tanımlı* kuyruk yöneticisi sayısı için yeterli büyüklükte olup olmadığını görmek için yapı denetlenir. Kuyruk yöneticileri, CSQ5PQSG yardımcı programı tarafından eklendiyse, kuyruk paylaşım grubuna tanımlandıkları varsayılır. MQSC DISPLAY GROUP komutuyla gruba tanımlanan kuyruk yöneticilerini denetleyebilirsiniz.

**Not:** Yapının boyutunu hesaplarken, kuyruk paylaşım grubundaki kuyruk yöneticilerinin sayısına ek olarak büyük iş birimlerinin boyutuna da izin vermelisiniz.

Çizelge 22 sayfa 190 içinde, kuyruk paylaşım grubunda tanımlanan çeşitli sayıda kuyruk yöneticisi için yönetim yapısı için gerekli minimum boyut gösterilir. Bu boyutlar CFCC seviye 14 bağlaşım tesisi yapısı için oluşturulmuştur; CFCC ' nin daha yüksek seviyeleri için muhtemelen daha büyük olması gerekir.

Çizelge 22. Yönetim yapısı boyutu alt sınırı	
Kuyruk paylaşım grubunda tanımlı kuyruk yöneticilerinin sayısı	Gerekli depolama alanı
1	6144 KB
2	6912 KB
3	7976 KB
4	8704 KB
5	9728 KB
6	10496 KB
7	11520 KB
8	12288 KB
9	13056 KB
10	14080 KB
11	14848 KB

Çizelge 22. Yönetim yapısı boyutu alt sınırı (devamı var)

Kuyruk paylaşım grubunda tanımlı kuyruk yöneticilerinin sayısı	Gerekli depolama alanı
12	15616 KB
13	16640 KB
14	17408 KB
15.000	18176 KB
16	19200 KB
17	19968 KB
18	20736 KB
19	21760 KB
20	22528 KB
21	23296 KB
22	24320 KB
23	25088 KB
24	25856 KB
25	27136 KB
26	27904 KB
27	28672 KB
28	29696 KB
29	30464 KB
30	31232 KB
31	32256 KB

Var olan bir kuyruk paylaşım grubuna bir kuyruk yöneticisi eklediğinizde, depolama gereksinimi Çizelge 22 sayfa 190 içinde önerilen boyutu aşmış olabilir. Bu durumda, `qsg-nameCSQ_ADMIN` yapısına ilişkin gerekli saklama alanını tahmin etmek için aşağıdaki yordamı kullanın:

1. Kuyruk paylaşım grubunun var olan bir üyesinde **DISPLAY CFSTATUS(CSQ\_ADMIN)** MQSC komutunu verin.
2. CSQ\_ADMIN yapısına ilişkin ENTSMAX bilgilerini alın.
3. Bu sayı, kuyruk paylaşım grubunda tanımlamak istediğiniz kuyruk yöneticilerinin toplam sayısının 1000 katından azsa, yapı boyutunu artırın.

### Uygulama yapıları

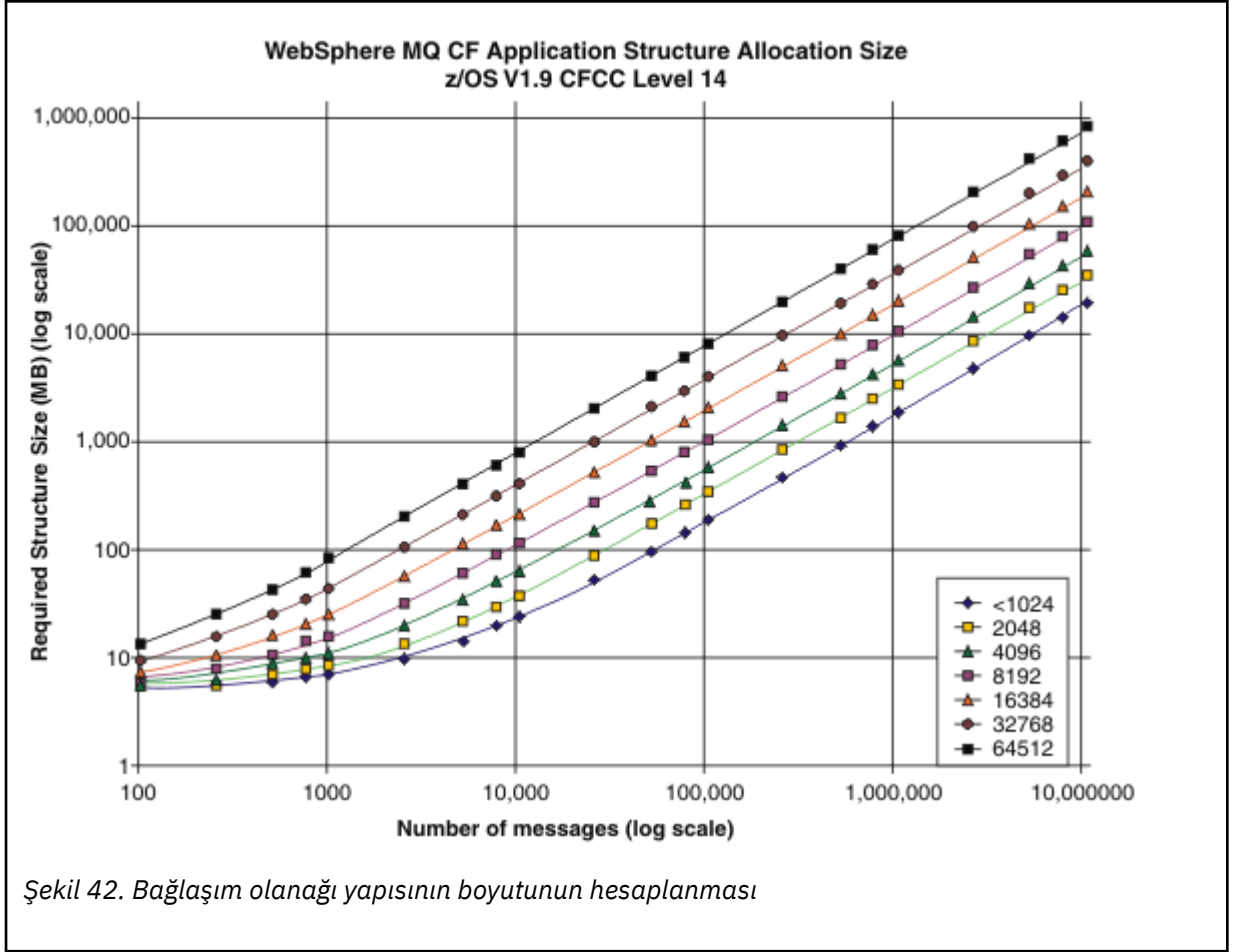
IBM MQ iletilerini tutmak için gereken uygulama yapılarının boyutu, bir yapıda aynı anda tutulacak iletilerin olası sayısına ve boyutuna bağlıdır.

Şekil 42 sayfa 192 içindeki grafik, iletileri paylaşılan kuyruklarınızda tutmak için CF yapılarınızı ne kadar büyük yapmanız gerektiğini gösterir. Ayırma boyutunu hesaplamak için aşağıdaki bilgilere gereksinim duyarsınız:

- Kuyruklarınızdaki iletilerin ortalama boyutu.
- Yapıda saklanacak iletilerin toplam sayısı.

Yatay eksen boyunca iletilerin sayısını bulun. İleti boyutunuza karşılık gelen eğriyi seçin ve dikey eksen den gerekli değeri belirleyin. Örneğin, uzunluğu 1 KB olan 200 000 ileti için 256-512 MB aralığında bir değer verilir.

Çizelge 23 sayfa 192 , aynı bilgileri tablo biçiminde sağlar.



Bağlaşım olanağı yapılarınızı ne kadar büyük yapacağınızı hesaplamaya yardımcı olmak için bu tabloyu kullanın:

Çizelge 23. Bağlaşım olanağı yapısının boyutunun hesaplanması

İleti sayısı	1 KB	2 KB	4 KB	8 KB	16 KB	32 KB	63 KB
100	6 MB	6 MB	7 MB	7 MB	8 MB	10 MB	14 MB
1000	8 MB	9 MB	12 MB	17 MB	27 MB	48 MB	88 MB
10000	25 MB	38 MB	64 MB	115 MB	218 MB	423 MB	821 MB
100000	199 MB	327 MB	584 MB	1097 MB	2124 MB	4177 MB	8156 MB

CFRM ilkeniz aşağıdaki deyimleri içermelidir:

- INITSIZE , ilk kuyruk yöneticisi bağlandığında yapının ayrıldığı KB cinsinden boyuttur.
- SIZE , yapının ulaşabileceği büyüklük üst sınırıdır.
- FULLTHRESHOLD , z/OS ' in IXC585E iletisini yayınladığı eşğin yüzde değerini, yapının dolduğunu belirtmek için belirler.



En iyi uygulama, INITSIZE ve SIZE ' in 2 kat içinde olmasını sağlamaktır. Örneğin, önceden belirlenmiş şekiller ile aşağıdaki deyimleri ekleyebilirsiniz:

```
STRUCTURE NAME(structure-name)
INITSIZE(value from graph in KB, that is, multiplied by 1024)
SIZE(something larger)
FULLTHRESHOLD(85)
```

```
STRUCTURE NAME(QSG1APPLICATION1)
INITSIZE(262144) /* 256 MB */
SIZE(524288) /* 512 MB */
FULLTHRESHOLD(85)
```

Yapı kullanımı uyarı iletilerinin verildiği eşige ulaşırsa, müdahale gerekir. IBM MQ , uygulamaların daha fazla ileti yazmasını önlemek, kuyruklardan ileti almak için daha fazla uygulama başlatmak ya da kuyruğa ileti koyan bazı uygulamaları susturmak için yapıdaki bazı kuyruklarda MQPUT işlemlerini engellemek için kullanılabilir.

Diğer bir seçenek olarak, z/OS olanaklarını kullanarak yapının büyüklüğünü değiştirebilirsiniz. Aşağıdaki z/OS komutu:

```
SETXCF START,ALTER,STRNAME=structure-name,SIZE=newsize
```

Yapının boyutunu *newsize* olarak değiştirir; burada *newsize* , yapıya ilişkin CFRM ilkesinde belirtilen SIZE değerinden küçük, ancak yürürlükteki bağlaşım olanağı boyutundan büyük bir değerdir.

MQSC [DISPLAY CFSTATUS](#) komutuyla bir bağlaşım olanağı yapısının kullanımını izleyebilirsiniz.

Herhangi bir işlem yapılmazsa ve bir kuyruk yapısı doldurulursa, uygulamaya bir MQRC\_STORAGE\_MEDIUM\_FULL dönüş kodu döndürülür. Yönetim yapısı dolursa, tam belirtiler hangi süreçlerin hatayla karşılaştıklarına bağlıdır, ancak bunlar aşağıdaki sorunları içerebilir:

- Komutlara yanıt yok.
- Kesinleştirme işlemi sırasında ortaya çıkan sorunların sonucu olarak kuyruk yöneticisi hatası oluştu.

### CSQSYSAPPL yapısı

*qsg-name*CSQSYSAPPL yapısı, sistem kuyruklarına ilişkin bir uygulama yapısıdır. Çizelge 3 , *qsg-name*CSQSYSAPPL yapısında tanımlanan varsayılan kuyruklara ilişkin ileti veri büyüklüklerinin nasıl tahmin edileceğine ilişkin bir örnek gösterir.

Çizelge 24. Boyutlandırmaya karşı CSQSYSAPPL kullanımını gösteren çizelge.	
<i>qsg-name</i> CSQSYSAPPL kullanımı	Boyutlandırma
SYSTEM.QSG.CHANNEL.SYNCQ	Paylaşılan bir kanalın etkin eşgörünümü başına 500 baytlık 2 ileti
SYSTEM.QSG.UR.RESOLUTION.QUEUE	2 KB ' lik 1000 ileti

Önerilen ilk yapı tanımı değerleri şunlardır:

```
STRUCTURE NAME(qsg-nameCSQSYSAPPL)
INITSIZE(20480) /* 20 MB */
SIZE(30720) /* 30 MB */
FULLTHRESHOLD(85)
```

Bu değerler, paylaşılan kanalları ve grup kurtarma birimlerini kullanımınıza bağlı olarak ayarlanabilir.

### Paylaşılan kuyrukların yapılar eşlenmesi

IBM MQ için bir uygulama yapısı tanımlamak üzere `DEFINE CFSTRUCT` komutunu kullanın. IBM MQ için bir yapı tanımladığınızda, yapı adına QSG adı örneğini eklemeyin. For example, to define an application structure to IBM MQ that has the name `qsg-adtAPPLICATION1` in the CFRM policy, issue the following command:

```
DEFINE CFSTRUCT(APPLICATION1)
```

Kuyruk tanımının CFSTRUCT özneliği, kuyruğu bir yapıyla eşlemek için kullanılır. Bu öznelikte QSG adı öneki olmadan CF yapısının adını belirtin. Örneğin, aşağıdaki komut APPLICATION1 yapısında paylaşılan bir kuyruğu tanımlar:

```
DEFINE QLOCAL(myqueue) QSGDISP(SHARED) CFSTRUCT(APPLICATION1)
```

## Paylaşılan ileti veri kümesi (SMDS) ortamınızın planlanması

SMDS boşaltma ile kuyruk paylaşım grupları kullanıyorsanız, IBM MQ ' un bir paylaşılan ileti veri kümesi grubuna bağlanması gerekir. IBM MQ ileti verilerini saklamak için gereken veri kümesi gereksinimlerini ve veri kümesi gereksinimlerini anlamanıza yardımcı olması için bu konuyu kullanın.

*Paylaşılan ileti veri kümesi* (SMDS anahtar sözcüğünün tanımladığı), bir kuyruk yöneticisi tarafından bir bağlaşım olanağı yapısında saklanan paylaşılan iletilere ilişkin yüklü ileti verilerini saklamak için kullanılan bir veri kümesidir.

**Not:** Bir yapı için SMDS veri kümeleri tanımlanırken, her kuyruk yöneticisi için bir tane olmalıdır.

Bu veri boşaltma biçimi etkinleştirildiğinde, **CFSTRUCT** , kuyruk paylaşım grubundaki her kuyruk yöneticisi için bir veri kümesi olan, paylaşılan ileti veri kümelerinin ilişkili bir grubunu gerektirir. Paylaşılan ileti verisi kümeleri grubu, **CFSTRUCT** tanımındaki **DSGROUP** parametresi kullanılarak IBM MQ olarak tanımlanır. Ek parametreler, kullanılacak arabellek sayısı ve veri kümelerinin genişletme öznelikleri gibi ek isteğe bağlı bilgileri sağlamak için kullanılabilir.

Her kuyruk yöneticisi, o kuyruk yöneticisi aracılığıyla yazılmış iletiler için paylaşılan ileti verilerini depolamak için sahip olduğu veri kümesine yazabilir ve gruptaki tüm veri kümelerinin tamamını okuyabilir.

Yapıyla ilişkilendirilmiş her veri kümesine ilişkin durumu ve öznelikleri tanımlayan bir liste, **CFSTRUCT** tanımlamasının bir parçası olarak içeride tutulur. Böylece, her kuyruk yöneticisi hangi veri kümelerinin kullanılabilir durumda olduğunu öğrenmek için tanımlamayı denetleyebilir.

This data set information can be displayed using the **DISPLAY CFSTATUS TYPE(SMDS)** command to display current status and availability, and the **DISPLAY SMDS** command to display the parameter settings for the data sets associated with a specified **CFSTRUCT**.

Tek tek paylaşılan ileti veri kümeleri, sahip olan kuyruk yöneticisi adının (genellikle **SMDS** anahtar sözcüğü kullanılarak belirtilir) ve **CFSTRUCT** yapı adı birleşimiyle etkili bir şekilde tanımlanır.

Bu bölümde aşağıdaki konular anlatılır:

- [DSGROUP parametresi](#)
- [DSBLOCK parametresi](#)
- [Paylaşılan ileti veri kümesi özellikleri](#)
- [Paylaşılan ileti veri kümesi alanı yönetimi](#)
- [Paylaşılan ileti veri kümelerine erişim](#)
- [Paylaşılan ileti verileri kümesi yaratılması](#)
- [Paylaşılan ileti veri kümesi performansı ve kapasite ile ilgili dikkat edilmesi gereken noktalar](#)
- [Paylaşılan ileti verileri kümesinin etkinleştirilmesi](#)

Bu parametrelere ilişkin ayrıntılar için [DEFINE CFSTRUCT](#) başlıklı konuya bakın.

Paylaşılan ileti veri kümelerinizi yönetmeye ilişkin bilgi için, ek ayrıntılar için [Paylaşılan ileti veri kümelerinin yönetilmesi](#) başlıklı konuya bakın.

## DSGROUP parametresi

**CFSTRUCT** tanımlamasındaki **DSGROUP** parametresi, o yapı için büyük iletilerin saklanacağı veri kümesi grubunu tanımlar. Ek parametreler, boşluk ayırma amacıyla kullanılacak mantıksal blok büyüklüğünü ve arabellek havuzu büyüklüğü ve otomatik veri kümesi genişletme seçeneklerine ilişkin değerleri belirlemek için kullanılabilir.

Veri kümelerinin boşaltılabilmesi için önce **DSGROUP** parametresinin ayarlanması gerekir.

- **CFLEVEL (5)** konumunda yeni bir **CFSTRUCT** tanımlandıysa ve **OFFLOAD (SMDS)** seçeneği belirlenirse ya da varsayılsa, aynı komutta **DSGROUP** parametresi de belirtilmelidir.
- If an existing **CFSTRUCT** is being altered to increase the **CFLEVEL** to **CFLEVEL (5)** and the option **OFFLOAD (SMDS)** is specified or assumed, then the **DSGROUP** parameter must be specified on the same command if it is not already set.

## DSBLOCK parametresi

Her veri kümesindeki alan, **CFSTRUCT** tanımında **DSBLOCK** parametresi kullanılarak belirlenen sabit bir büyük-256 KB 'lik mantıksal blok olarak kuyruklara ayrılır ve her bir mantıksal blok içindeki 4 KB' lik (fiziksel blok boyutu ve denetim aralığı boyutuna karşılık gelen) sayfa aralıkları olarak tek tek iletilere ayrılır. Mantıksal blok büyüklüğü, tek bir G/Ç işleminde okunabilecek ya da yazılabilecek en fazla ileti verisi miktarını da belirler. Bu değer, SMDS arabellek havuzunun arabellek büyüklüğünün aynısıdır.

A larger value of the **DSBLOCK** parameter can improve performance for very large messages by reducing the number of separate I/O operations. Ancak, daha küçük bir değer, her bir etkin istek için gereken arabellek miktarını azaltır. **DSBLOCK** parametresinin varsayılan değeri 256 KB ' dir ve bu parametre, bu gereksinimlerin arasında makul bir denge sağlar. Bu nedenle, bu parametrenin belirtilmesi olağan bir şekilde gerekmez.

## Paylaşılan ileti veri kümesi özellikleri

Paylaşılan ileti veri kümesi VSAM doğrusal veri kümesi (LDS) olarak tanımlanır. Yüklenen her ileti, veri kümesindeki bir ya da daha çok blokta saklanır. Saklanan veriler, bağlaşım olanağı girişlerindeki, genişletilmiş bir sanal saklama alanı gibi doğrudan doğruya adreslenmektedir. Veri kümesinde ayrı bir izin ya da benzer denetim bilgisi saklanmaz.

Doğrudan adresleme şeması, tek bir bloğa sığan iletiler için, bloğu okumak ya da yazmak için tek bir G/Ç işlemine gerek duyulması anlamına gelir. Bir ileti birden çok öbeği kapladığında, yeterli arabelleklerin kullanılabilir olması koşuluyla, her bir bloğa ilişkin G/Ç işlemleri, geçen süreyi en aza indirmek için tam olarak örtüşebilir.

Paylaşılan ileti verileri kümesi, ilk sayfadaki bir üstbilginden oluşan ve kurtarma ve yeniden başlatma durumu bilgilerini ve kuyruk yöneticisi olağan sonlandırmasında serbest blok alanı eşlemeni saklamak için kullanılan bir boşluk eşlemi denetim noktası alanını içeren, küçük bir genel denetim bilgisi içerir.

## Paylaşılan ileti veri kümesi alanı yönetimi

Kapasite, performans ve operasyonel noktalar için arka plan bilgileri olarak, paylaşılan ileti veri kümelerindeki boşluğun kuyruk yöneticileri tarafından nasıl yönetildiği kavramlarını anlamak yararlı olabilir.

Her bir paylaşılan ileti veri kümesinde yer alan boş alan, her bir mantıksal blok içinde kullanımda olan sayfa sayısını belirten bir alan eşlemi kullanarak, sahip olan kuyruk yöneticisi tarafından izlenir. Veri kümesi açık olduğunda ve olağan bir şekilde kapatıldığında veri kümesi veri kümesinde saklandığında, alan eşlemi ana saklama alanında tutulur. (Kurtarma durumlarında, iş alanı eşlemi, kullanılmakta olan veri kümesi sayfalarını bulmak için bağlaşım olanağı yapısındaki iletileri tarayarak otomatik olarak yeniden oluşturulur).

Doldurulan ileti verileri içeren bir paylaşılan ileti yazılırken, kuyruk yöneticisi her ileti öbeği için bir sayfa aralığı ayırır. Belirtilen kuyruk için kısmen kullanılmış bir yürürlükteki mantıksal blok varsa, kuyruk yöneticisi o blokta sonraki boş sayfadan başlayarak boşluk ayırır; tersi durumda, yeni bir mantıksal

blok ayırır. İletinin tamamı yürürlükteki mantıksal bloğa sığmıyorsa, kuyruk yöneticisi ileti verilerini mantıksal bloğun sonunda böler ve sonraki ileti bloğu için yeni bir mantıksal blok ayırır. Bu, tüm ileti için alan ayrılana kadar yinelenir. Son mantıksal blokta kullanılmayan herhangi bir alan, kuyruk için yeni yürürlükteki mantıksal blok olarak saklanır. Veri kümesi olağan şekilde kapatıldığında, yürürlükte olan mantıksal bloklarda kullanılmayan sayfalar, kaydedilmeden önce alan haritasına döndürülür.

Görel olarak yüklenen ileti verileri içeren bir paylaşılan ileti okunduğunda ve silinmeye hazır olduğunda, kuyruk yöneticisi, iletiye ilişkin bağlaşım olanağı girişini, sahip olan kuyruk yöneticisi (aynı kuyruk yöneticisi olabilir) tarafından izlenen bir temizleme listesine aktararak silme işlemi işler. Girişler bu listeye ulaştığında, sahip olan kuyruk yöneticisi girişleri okur ve siler ve serbest bırakılan sayfa aralıklarını alan eşlemine döndürür. Bir mantıksal blokta kullanılan tüm sayfalar serbest bırakıldığında, öbek yeniden kullanım için kullanılabilir duruma gelir.

## Paylaşılan ileti veri kümelerine erişim

Paylaşılan her ileti veri kümesi, kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından erişilebilen, paylaşılan doğrudan erişim depolaması üzerinde olmalıdır.

Olağan çalışma sırasında, her kuyruk yöneticisi okuma/yazma erişimi için kendi paylaşılan ileti veri kümesini açar ve diğer kuyruk yöneticileri için salt okunur erişim için etkin paylaşılan ileti veri kümelerini açar; böylece, bu kuyruk yöneticileri tarafından saklanan iletileri okuyabilir. Bu, her kuyruk yöneticisi kullanıcı kimliğinin, kendi paylaşılan ileti veri kümesi için en az UPDATE erişimi gerektirmesi ve yapı için diğer tüm paylaşılan ileti veri kümelerine okuma erişimi olması anlamına gelir.

If it is necessary to recover shared message data sets using **RECOVER CFSTRUCT**, the recovery process can be executed from any queue manager in the queue sharing group. Kurtarma işlemi gerçekleştirilmek için kullanılacak bir kuyruk yöneticisi, kurtarılması gerekebileceği tüm veri kümelerine UPDATE erişimi gerektirir.

## Paylaşılan ileti verileri kümesi yaratılması

Her paylaşılan ileti veri kümesi olağan durumda, ilgili **CFSTRUCT** tanımlaması yaratılmadan ya da değiştirilmeden önce yaratılmalıdır; **CFSTRUCT** tanımlama değişiklikleri olağan durumda hemen yürürlüğe girecektir ve kuyruk yöneticisi, o yapıya atanmış olan bir paylaşılan kuyruğa erişmeye çalıştığında veri kümesi gerekecektir. SSCSQQPROC (CSQ4SMDS) içinde, paylaşılan bir ileti veri kümesini ayırmak ve önceden biçimlendirmek için kullanılan örnek bir iş sağlanır. İş uyarlanmalıdır ve OFFLOAD (SMDS) içeren bir CFSTRUCT kullanan her kuyruk yöneticisi için paylaşılan bir ileti verisi kümesi ayırmak için çalıştırılmalıdır.

Kuyruk yöneticisi, boşaltma desteğinin etkinleştirildiğini ve paylaşılan ileti veri kümesini açmaya çalıştıysa, ancak henüz oluşturulmamışsa, paylaşılan ileti veri kümesi kullanılamaz olarak işaretlenir. Daha sonra, kuyruk yöneticisi, veri kümesi yaratılincaya kadar büyük iletiler saklayamaz ve kuyruk yöneticisi, örneğin **START SMDSCONN** komutunu kullanarak yeniden deneyecek şekilde bilgilendirilecektir.

Paylaşılan ileti veri kümesi, Access Method Services **DEFINE CLUSTER** komutu kullanılarak VSAM doğrusal veri kümesi olarak yaratılır. The definition must specify **SHAREOPTIONS(2 3)** to allow one queue manager to open it for write access and any number of queue managers to read it at the same time. 4 KB ' lik varsayılan denetim aralığı büyüklüğü kullanılmalıdır. Veri kümesinin 4 GB ' nin ötesinde genişletilmesi gerekiyorsa, bu, VSAM genişletilmiş adreslenebilirliği özneliğine sahip bir SMS veri sınıfı kullanılarak tanımlanmalıdır. Paylaşılan ileti veri kümesi, genişletilmiş adres birimlerinin (EAV) uzatılmış adresleme alanı (EAS) bir parçasında yer almaya hak kazanır.

Each shared message data set can either be empty or pre-formatted to binary zeros (using **CSQJUFMT** or a similar utility such as the sample job SCSQQPROC(CSQ4SMDS)), before its initial use. Boşsa ya da yalnızca açıldığında kısmen biçimlendirilmişse, kuyruk yöneticisi kalan alanı otomatik olarak ikili sıfırlara biçimlendirir.

## Paylaşılan ileti veri kümesi başarıımı ve kapasite ile ilgili önemli noktalar

Her paylaşılan ileti veri kümesi, aynı sistem içindeki bölgelerden, sahip olan kuyruk yöneticisi tarafından ilişkili **CFSTRUCT** ' e yazılan paylaşılan iletiler için verileri yüklenen verileri saklamak için kullanılır. Her ileti

için saklanan veriler, bir tanımlayıcı (şu anda yaklaşık 350 bayt), ileti üstbilgileri ve ileti gövdesiyle birlikte bulunur. Yüklenen her ileti, veri kümesindeki bir ya da daha çok sayfada (4 KB ' lik fiziksel bloklar) saklanır.

Belirli sayıda boşaltılan ileti için gereken veri kümesi alanı, bu nedenle, genel ileti boyutunu (tanımlayıcı dahil) 4 KB ' nin bir sonraki katına yuvarlanarak ve sonra ileti sayısı ile çarpılarak tahmin edilebilir.

Bir sayfa kümesine ilişkin olarak, paylaşılan bir ileti veri kümesi neredeyse tam olarak ayarlandığında, isteğe bağlı olarak otomatik olarak genişletilebilir. Bu otomatik genişletmeye ilişkin varsayılan davranış, **CFSTRUCT** tanımındaki **DSEXPAND** parametresi kullanılarak ayarlanabilir. Bu ayar, **ALTER SMDS** komutundaki **DSEXPAND** parametresini kullanan her bir kuyruk yöneticisi için geçersiz kılınabilir. Otomatik genişletme, veri kümesi %90 dolunca bir alana ulaştığında tetiklenir ve daha fazla alan gerekir. Genişletme işlemine izin verilirse, ancak veri kümesi tanımlandığında ikincil alan ayırması belirtilmediğinden, genişletme girişimi VSAM tarafından reddedilirse, genişletme, veri kümesinin geçerli boyutunun %20 'si ikincil bir ayırma kullanılarak yeniden denir.

Paylaşılan ileti veri kümesi, genişletilmiş adreslenebilirlik öznelikle tanımlansa da, büyüklük üst sınırı yalnızca VSAM ile en çok 16 TB ya da 59 birim ile sınırlanır. Bu, yerel sayfa kümesinin 64 GB ' lik üst sınırından büyük ölçüde büyük.

## Paylaşılan ileti veri kümesinin etkinleştirilmesi

Bir kuyruk yöneticisi bir uygulama bağlaşım olanağı yapısına başarıyla bağlandığında, o yapı tanımlamasının ilişkili bir **DSGROUP** parametresi kullanılarak boşaltma belirtisini belirleyip belirtmediğini denetler. Bu durumda, kuyruk yöneticisi yazma erişimi için kendi paylaşılan ileti verilerini ayırır ve açar; daha sonra, diğer kuyruk yöneticilerinin sahip olduğu var olan paylaşılan ileti veri kümelerinin okuma erişimi için açılır.

Paylaşılan bir ileti veri kümesi ilk kez açıldığında (kuyruk paylaşım grubu içinde etkin olarak kaydedilmeden önce), ilk sayfa henüz geçerli bir üstbilgi içermeyecek. Kuyruk yöneticisi, kuyruk paylaşım grubunu, yapı adını ve sahip olan kuyruk yöneticisini tanımlamak için üstbilgi bilgilerini doldurur.

Üstbilgi tamamlandıktan sonra, kuyruk yöneticisi yeni paylaşılan ileti veri kümesini etkin olarak kaydeder ve yeni veri kümesiyle ilgili diğer etkin kuyruk yöneticilerine bildirim göndermek için bir olay yayınlar.

Bir kuyruk yöneticisi paylaşılan bir ileti veri kümesini her açtığında, doğru veri kümesinin hala kullanılmakta olduğundan ve zarar görmediğinden emin olmak için üstbilgi bilgilerinin geçerliliğini denetler.

## **Db2 ortamınızın planlanması**

Kuyruk paylaşım grupları kullanıyorsanız, IBM MQ , bir veri paylaşım grubunun üyesi olan bir Db2 altsistemine bağlanmanız gerekir. IBM MQ verilerini tutmak için kullanılan Db2 gereksinmelerini anlamana yardımcı olması için bu konuyu kullanın.

IBM MQ ' in bağlanacağı veri paylaşım grubunun adını ve bağlantı kurmak için bir Db2 altsisteminin (ya da Db2 grubunun) adını bilmesi gerekir. Bu grup, bu veri paylaşım grubuna ulaşmak için bu grubun adını ve bu veri paylaşım grubuna bağlanır. Bu adlar, CSQ6SYSP sistem parametre makrosu ( [CSQ6SYSPkomutunu kullanma](#) içinde açıklanmıştır) QSGDATA parametresinde belirtilir.

Veri paylaşımı grubu içinde, paylaşılan Db2 tabloları tutulacak şekilde kullanılır:

- Kuyruk paylaşım grubuna ilişkin yapılandırma bilgileri.
- IBM MQ paylaşılan ve grup nesnelerinin özellikleri.
- İsteğe bağlı olarak, yüklenen IBM MQ iletileriyle ilgili veriler.

IBM MQ , Db2 çizelge alanlarının, çizelgelerinin ve dizinlerinin tanımlanması için örnek işler sağlar. İki örnek iş kümesi sağlanır:

- One for compatibility with earlier versions of IBM MQ
- Bir adet, Db2 V12 ve sonraki yayın düzeyleriyle kullanılmak üzere, Universal Table Spaces (UTS) olanağını kullanan

Since Db2 V11, various types of table space, used by earlier versions of IBM MQ, have been flagged as deprecated. Mümkün olduğunda, yeni bir veri paylaşım grubu ayarlarken, Db2 UTS ' yi kullanan iş kümesini seçmeye teşvik edildiniz.

IBM MQ , daha eski tip çizelge alanı tipleriyle de kullanılabilir ve önceden varolan bir kuyruk paylaşım grubunuz varsa, bu uygun olabilir. Ancak, yeni bir kuyruk paylaşım grubu yaratıyorsanız, bu, yeni bir kuyruk paylaşım grubu olarak UTS ' yi kullanmalıdır.

Db2 V12 İşlev düzeyi 508 artık çok çizelge alanlarının evrensel çizelge alanlarına geçirilmesi için kesintiye neden olmayan bir geçiş süreci sağlar. Bu yaklaşımı kullanarak, var olan kuyruk paylaşım grupları tarafından, tüm kuyruk paylaşım grubunun bir kesilmesini almadan evrensel çizelge alanlarına, çok çizelge alan çizelge alanlarını yeni düzeye geçirebilirsiniz.

Varsayılan olarak Db2 , işleri Db2 kaynaklarının sahibi olarak çalıştıran kişinin kullanıcı kimliğini kullanır. Bu kullanıcı kimliği silinirse, onunla ilişkili kaynaklar silinir ve bu nedenle tablo silinir. Tek bir kullanıcı kimliği yerine, bir grup kimliğini tabloların sahibi olmak için kullanabilirsiniz. Bunu, JOB kartına GROUP=groupname ekleyerek ve SQL deyimlerinden önce SET CURRENT SQLID= ' groupname ' belirtilerek yapabilirsiniz.

IBM MQ , RRS Attach olanağını ( Db2) kullanır. Bu, bağlanmak istediğiniz bir Db2 grubunun adını belirleyebileceğiniz anlamına gelir. Bir Db2 grubu bağlama adına (belirli bir Db2 altsistemi yerine) bağlanma avantajı, IBM MQ ' in bu grubun üyesi olan z/OS görüntüsündeki kullanılabilir Db2 altsistemlerine bağlanabilmesi (ya da yeniden bağlanabileceği). Kuyruk paylaşımını IBM MQ altsistemini çalıştıracacağı ve RRS ' nin etkin olması gereken her bir z/OS görüntüsünde etkin veri paylaşım grubunun üyesi olan bir Db2 altsistemi olmalıdır.

## Db2 depolama

Çoğu kuruluş için gerekli olan Db2 depolama miktarı, 3390 aygıtında yaklaşık 20 ya da 30 silindirdir. Ancak, depolama gereksiniminizi hesaplamak istiyorsanız, aşağıdaki tabloda IBM MQ verileri için Db2 ' in ne kadar depolama gerektirdiğini belirlemenize yardımcı olacak bazı bilgiler verilmektedir. Tabloda, her bir Db2 satırının uzunluğu ve her bir satır ilgili Db2 tablosundan eklendiği ya da bu tablodan silindiği anlatılır. Db2 tablolarına ve dizinlerine ilişkin alan gereksinimlerinin hesaplanması ile ilgili bilgileri *Db2 for z/OS Installation Guide* ' da birlikte kullanın.

<i>Çizelge 25. Db2 depolama gereksinimlerinin planlanması</i>			
<b>Db2 Çizelge Adı</b>	<b>Satır uzunluğu</b>	<b>Aşağıdaki durumlarda bir satır eklenir:</b>	<b>Aşağıdaki durumlarda bir satır silinir:</b>
CSQ.ADMIN_B_QSG	252 bayt	Çizelgeye, CSQ5PQSG yardımcı programının ADD QSG işleviyle birlikte bir kuyruk paylaşım grubu eklenir.	Bir kuyruk paylaşım grubu, CSQ5PQSG yardımcı programının REMOVE QSG işleviyle birlikte çizelgeden kaldırılır. (Bu kuyruk paylaşım grubuyla ilgili tüm satırlar, kuyruk paylaşım grubu kaydı silindiğinde, diğer tüm Db2 çizelgelerinden otomatik olarak silinir.)
CSQ.ADMIN_B_QMGR	En çok 3828 bayt	Çizelgeye, CSQ5PQSG yardımcı programının ADD QMGR işleviyle birlikte bir kuyruk yöneticisi eklenir.	Bir kuyruk yöneticisi, CSQ5PQSG yardımcı programının REMOVE QMGR işleviyle tablodan kaldırılır.

Çizelge 25. Db2 depolama gereksinimlerinizin planlanması (devamı var)

Db2 Çizelge Adı	Satır uzunluğu	Aşağıdaki durumlarda bir satır eklenir:	Aşağıdaki durumlarda bir satır silinir:
CSQ.ADMIN_B_STRUCTURE	1454 bayt	Birinci yerel kuyruk tanımlaması, QSGDISP (SHARED) özniteliğini belirterek, kuyruk paylaşım grubu içinde daha önce bilinmeyen bir yapıya sahip adlar tanımlanır.	QSGDISP (SHARED) özniteliğini belirten son yerel kuyruk tanımlaması, kuyruk paylaşım grubu içindeki bir yapının silinmesine neden olur.
CSQ.ADMIN_B_SCST	342 bayt	Paylaşılan bir kanal başlatılır.	Paylaşılan bir kanal devre dışı olur.
CSQ.ADMIN_B_SSKT	254 byte	NPMSPEED (NORMAL) özniteliğine sahip paylaşılan bir kanal başlatılır.	NPMSPEED (NORMAL) özniteliğine sahip paylaşılan bir kanal devre dışı olur.
CSQ.ADMIN_B_STRBACKUP	514 bayt	CSQ.ADMIN_B_STRUCTURE tablosu. Her giriş, yedek CFSTRUCT komutu çalıştırılıncaya kadar, kukla girişlerin üzerine yazılıncaya kadar bir kukla girişe sahip olur.	A row is deleted from the CSQ.ADMIN_B_STRUCTURE table.
CSQ.OBJ_B_AUTHINFO	3400 bayt	QSGDISP (GROUP) ile bir kimlik doğrulama bilgisi nesnesi tanımlıdır.	QSGDISP (GROUP) ile bir kimlik doğrulama bilgisi nesnesi silinir.
CSQ.OBJ_B_QUEUE	En çok 3707 bayt	<ul style="list-style-type: none"> <li>QSGDISP (GROUP) özniteliğine sahip bir kuyruk tanımlandı.</li> <li>QSGDISP (SHARED) özniteliğine sahip bir kuyruk tanımlandı.</li> <li>DEFTYPE (SHAREDYN) özniteliğine sahip bir model kuyruğu açıldı.</li> </ul>	<ul style="list-style-type: none"> <li>QSGDISP (GROUP) özniteliğine sahip bir kuyruk silindi.</li> <li>QSGDISP (SHARED) özniteliğine sahip bir kuyruk silindi.</li> <li>DEFTYPE (SHAREDYN) özniteliğine sahip bir dinamik kuyruk DELETE seçeneğiyle kapatıldı.</li> </ul>
CSQ.OBJ_B_NAMELIST	En çok 15127 bayt	QSGDISP (GROUP) özniteliğine sahip bir ad listesi tanımlıdır.	QSGDISP (GROUP) özniteliğine sahip bir ad listesi silinir.
CSQ.OBJ_B_CHANNEL	En çok 14127 bayt	QSGDISP (GROUP) özniteliğine sahip bir kanal tanımlıdır.	QSGDISP (GROUP) özniteliğine sahip bir kanal silindi.
CSQ.OBJ_B_STGCLASS	En çok 2865 bayt	QSGDISP (GROUP) özniteliğine sahip bir depolama sınıfı tanımlıdır.	QSGDISP (GROUP) öznitelik sınıfıyla bir depolama sınıfı silindi.
CSQ.OBJ_B_PROCESS	En çok 3347 bayt	QSGDISP (GROUP) özniteliğine sahip bir işlem tanımlandı.	QSGDISP (GROUP) özniteliğine sahip bir işlem silindi.
CSQ.OBJ_B_TOPIC	En çok 14520 bayt	QSGDISP (GROUP) özniteliğine sahip bir konu nesnesi tanımlıdır.	QSGDISP (GROUP) özniteliğine sahip bir konu nesnesi silindi.

Çizelge 25. Db2 depolama gereksinimlerinizin planlanması (devamı var)			
Db2 Çizelge Adı	Satır uzunluğu	Aşağıdaki durumlarda bir satır eklenir:	Aşağıdaki durumlarda bir satır silinir:
CSQ.EXTEND_B_QMGR	430 bayttan az	Çizelgeye, CSQ5PQSG yardımcı programının ADD QMGR işleviyle birlikte bir kuyruk yöneticisi eklenir.	Bir kuyruk yöneticisi, CSQ5PQSG yardımcı programının REMOVE QMGR işleviyle tablodan kaldırılır.
CSQ.ADMIN_B_MESSAGES	87 bayt	Büyük ileti PUT için (BLOB başına 1).	Büyük ileti GET için (BLOB başına 1).
CSQ.ADMIN_MSGS_BAUX1 CSQ.ADMIN_MSGS_BAUX2 CSQ.ADMIN_MSGS_BAUX3 CSQ.ADMIN_MSGS_BAUX4		Bu 4 çizelge, iletinin her bir BLOB 'u için bu 4 çizelgeden birine eklenen büyük iletiler için ileti bilgi yükü içerir. BLOB uzunluğu 511 KB 'ye kadar, bu nedenle ileti boyutu > 711 KB ise, bu ileti için birden çok BLOB bulunur.	

63 KB 'den büyük boyuttaki paylaşılan kuyruk iletilerinin büyük sayılarının kullanılması, IBM MQ sisteminizde önemli başarımlar etkilere sahip olabilir. Daha fazla bilgi için bkz. SupportPac MP16, Capacity Planning and Tuning for IBM MQ for z/OS, at: [SupportPacs for IBM MQ and other project alans.](#)

## **z/OS Yedekleme ve kurtarma planlanması**

Yerinizdeki yedekleme ve kurtarma yordamlarının geliştirilmesi, yüksek maliyetli ve zaman kaybına neden olan veri kayıplarından kaçınmak için hayati önem kazanır. IBM MQ , bir sistem hatasından sonra hem kuyrukların hem de iletilerin yürürlükteki durumlarına geri yüklenmesi anlamına gelir.

Bu konu aşağıdaki bölümleri içerir:

- [“Kurtarma yordamları” sayfa 200](#)
- [“Yedekleme ve Kurtarma İpuçları” sayfa 201](#)
- [“Sayfa kümeleri kurtarılıyor” sayfa 203](#)
- [“CF yapılarının kurtarılıyor” sayfa 204](#)
- [“Belirli kurtarma hedeflerine ulaşılması” sayfa 205](#)
- [“Diğer ürünlere ilişkin yedekleme konuları” sayfa 206](#)
- [“Kurtarma ve CICS” sayfa 207](#)
- [“Kurtarma ve IMS” sayfa 207](#)
- [“Alternatif bir site için kurtarma hazırlığı yapılıyor” sayfa 207](#)
- [“Kuyruk yöneticisi yedekleme etkinliği örneği” sayfa 207](#)

### **Kurtarma yordamları**

IBM MQ için aşağıdaki yordamları geliştirin:

- Kurtarma noktası oluşturma.
- Sayfa kümeleri yedekleniyor.
- CF yapıları yedekleniyor.
- Sayfa kümeleri kurtarılıyor.
- Alan dışı durumların kurtarılıyor ( IBM MQ günlükleri ve sayfa kümeleri).



- CF yapıları kurtarılıyor.

Bunlarla ilgili bilgi için bkz. [Yönetme IBM MQ for z/OS](#) .

Sitenizde aşağıdakiler için kullanılan yordamlara aşına olun:

- Bir donanım ya da güç kesintisinden kurtarma.
- Bir z/OS bileşeni hatasından kurtarma işlemi.
- Yerdışı kurtarma kullanılarak, bir yer kesintisinden kurtarma.

## **Yedekleme ve Kurtarma İpuçları**

Bazı yedekleme ve kurtarma görevlerini anlamak için bu konuyu kullanın.

Kuyruk yöneticisi yeniden başlatma işlemi, günlük bilgilerini sayfa kümelerine uygulayarak verilerinizin tutarlı bir duruma geri döner. Sayfa kümeleriniz zarar görmüşse ya da kullanılmıyorsa, sayfa kümelerinizin yedek kopyalarını kullanarak sorunu çözebilirsiniz (tüm günlükler kullanılabilir durumda ise). Günlük veri kümeleriniz zarar görmüşse ya da kullanılmıyorsa, tamamen kurtarılamayabilir.

Aşağıdaki noktaları göz önünde bulundurun:

- [Belirli aralıklarla yedek kopyaları al](#)
- [İhtiyaç verebileceğiniz arşiv günlüklerini atmayın](#)
- [DDname ögesini sayfa kümesi ilişkilendirmesine değiştirmeyin](#)

### **Yedek kopyaları düzenli olarak al**

*Geri yükleme noktası* , IBM MQ sayfa kümelerinin ve bu sayfa kümelerini kurtarmak için gereken ilgili günlük veri kümelerinin bir dizi yedek kopyasını tanımlamak için kullanılan terimdir. Bu yedek kopyalar, sayfa kümesi kaybı durumunda olası bir yeniden başlatma noktası sağlar (örneğin, sayfa kümesi G/Ç hatası). Bu yedek kopyaları kullanarak kuyruk yöneticisini yeniden başlatıyorsanız, IBM MQ içindeki veriler, bu kopyaların alındığı noktaya kadar tutarlı olur. Bu noktadan sonra tüm günlüklerin kullanılabilir olması koşuluyla, IBM MQ hata noktasına kurtarılabilir.

Yedekleme kopyalarınız ne kadar çok yeni kopyalasa, IBM MQ , sayfa kümelerindeki verileri daha hızlı kurtarabilir. Sayfa kümelerinin kurtarılması, kullanılabilir durumda olan tüm gerekli günlük veri kümelerine bağlıdır.

Kurtarma planlamasında, yedekleme kopyalarının ne sıklıkta ve kaç tam yedekleme döngülerinin tutulacağı saptanmanız gerekir. These values tell you how long you must keep your log data sets and backup copies of page sets for IBM MQ recovery.

Yedekleme kopyalarını almak için ne sıklıkta karar vereceğine karar verirken, bir sayfa kümesini kurtarmak için gereken zamanı göz önünde bulundurun. Gereken süre aşağıdaki gibi belirlenir:

- Geçiş için günlüğe kaydetme miktarı.
- Bir işletmenin arşiv manyetik bant birimlerini bağlama ve çıkarma işlemi sürme süresi.
- Kurtarma için gerekli olan günlüğün bölümünü okumak için gereken süre.
- Değiştirilen sayfaları yeniden işlemek için gereken süre.
- Yedek kopyalar için kullanılan depolama ortamı.
- Yedekleme kopyalarını yapmak ve geri yüklemek için kullanılan yöntem.

Genel olarak, yedek kopyaları ne kadar sık yaparsanız, daha az zaman kurtarma işlemi yapılır, ancak daha fazla zaman, kopyalama işlemi için daha fazla zaman harlanır.

Her kuyruk yöneticisi için, aşağıdakilerin yedek kopyalarını almalısınız:

- Arşiv günlüğü veri kümeleri
- Arşivin zamanında yaratılan BSDS kopyaları

- Sayfa kümeleri
- Nesne tanımlarınız
- CF yapılarınız

Yedek kopyalarınızın kaybolması ya da hasar görmesi riskini azaltmak için aşağıdakileri göz önünde bulundurun:

- Yedek kopyalar, farklı depolama birimlerinde özgün kopyalara depolanıyor.
- Yedek kopyalar farklı bir yerde özgün kopyalara depolanıyor.
- Sayfa kümelerinizin her bir yedeğinin en az iki kopyası ve tek bir günlük kaydı ya da tek bir BSDS kullanıyorsanız, arşiv günlüklerinizin ve BSDS ' nizin iki kopyası bulunur. Çift günlük kaydı ya da BSDS kullanıyorsanız, hem arşiv günlüklerinin, hem de BSDS ' nin tek bir kopyasını yapın.

IBM MQ ' i bir üretim ortamına taşımadan önce, yedekleme yordamlarınızı tam olarak test edin ve belgeleyin.

### **Sayfa kümelerinizin yedeklenmesi**

Sayfa kümelerini düzenli olarak yedeklemeniz gerekir. Bazı kuruluşlar sayfa kümelerini günde iki kez yedeklemektedir.

Yedeklemeyi kullanarak kurtarma işlemi gerçekleştirilebilmesi için, yedekleme işlemi için etkin ve arşiv günlüklerine gereksiniminiz vardır. Kuyruk yöneticisi çalışırken yedeklendiyse, dört denetim noktasını geri almak için yeterli günlük verisine gereksinim duyarsınız.

Sayfa kümelerini yedeklemek için ADRDSSU FastReplication seçeneğini kullanabilir ve kuyruk yöneticisi etkinken bunu yapabilirsiniz. Depolama havuzunda yeterli yer olduğunu doğrulamanız gerektiğini unutmayın.

### **Nesne tanımlarınızın yedeklenmesi**

Nesne tanımlarınızın yedek kopyalarını yaratın. Bunu yapmak için, yardımcı programın COMMAND işlevinin MAKEDEF özelliğini kullanın ( CSQUTIL ' in COMMAND işlevini kullanma' ta açıklanmıştır).

Bu işlemi, kuyruk yöneticisi veri kümelerinizin yedek kopyalarını aldığınızda ve en güncel sürümü alıkoymak için yapmanız gerekir.

### **Bağlaşım tesisi yapılarınızı yedeklemek**

Herhangi bir kuyruk paylaşım grubu ayarladıysanız, bunları kullanmayasınız bile, CF yapılarınızın düzenli olarak yedeklenmesini almalısınız. Bunu yapmak için IBM MQ BACKUP CFSTRUCT komutunu kullanın. Bu komutu yalnızca RECOVER (YES) öznitelikle tanımlanmış CF yapılarında kullanabilirsiniz. Kalıcı paylaşılan iletiler için herhangi bir CF girişi, paylaşılan ileti veri kümesinde (SMDS) ya da Db2' de saklanan ileti verilerine gönderme yapıyor olursa, ofsayt dışı veriler alınır ve CF girişleriyle birlikte yedeklenir. Paylaşılan ileti veri kümeleri ayrı olarak yedeklenmemelidir.

CF yapısını geri yüklemek için gereken süreyi en aza indirmek için, tüm CF yapılarınızın yedeğini almak, her saat için bir yedekleme yapmanız önerilir.

Tek bir kuyruk yöneticisinde tüm CF yapısı yedeklemelerinizi tek bir kuyruk yöneticisinde gerçekleştirebilirsiniz. Bu, günlük kullanımında artışını tek bir kuyruk yöneticisinde sınırlandırmanızı sağlar. Diğer bir seçenek olarak, iş yükünü kuyruk paylaşım grubu boyunca dağıtma avantajına sahip olan kuyruk paylaşım grubundaki tüm kuyruk yöneticilerindeki yedekler de gerçekleştirebilirsiniz. Hangi stratejiyi kullanırsanız kullanın, IBM MQ yedeğin yerini bulabilir ve kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisinden RECOVER CFstruct (kurtarma işlemi) gerçekleştirmenizi sağlar. CF yapısını kurtarmak için kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinin günlüklerine erişilmesi için bu günlüklerin erişilmesi gerekir.

### **İleti güvenliği ilkelerinizin yedeklenmesi**

İleti güvenliği ilkelerinizin yedeğini yaratmak için Advanced Message Security kullanıyorsanız, **dspmqspl** komutunu -export deęiřtirgesiyle çalıştırmak için Message Security Policy Utility (CSQOUTIL) komutunu kullanarak bir yedekleme yaratın ve çıkış olan ilke tanımlarını EXPORT DD ' ye kaydedin.

Kuyruk yöneticisi veri kümelerinizin yedek kopyalarını aldığınızda ve en güncel sürümü alıyorsanız, ileti güvenliği ilkelerinizin yedeğini yaratmanız gerekir.

## Gereksinim duyabilirsiniz arşiv günlüklerini atmayın

IBM MQ might need to use archive logs during restart. Sistemin tam olarak geri yüklenebilmesi için yeterli arşiv günlüklerini tutmanız gerekir. IBM MQ , geri yüklenen bir yedek kopyasından bir sayfa kümesini kurtarmak için bir arşiv günlüğü kullanabilir. Arşiv günlüğünü attığınız takdirde, IBM MQ sayfayı yürürlükteki durumuna geri yükleyemez. Arşiv günlüklerinin nasıl atıldığı ve nasıl atıldığı [Discarding arşiv günlüğü veri kümeleri](#) 'nde açıklanmıştır.

Kuyruk yöneticinizin sayfa kümelerini ve kuyruk paylaşım grubunun yapılarını kurtarmanız için gereken günlük RBA 'yı ve günlük aralığı sıra numarasını (LRSN) görüntülemek için /cpf DIS USAGE TYPE (ALL) komutunu kullanabilirsiniz. Daha sonra, kuyruk yöneticisine ilişkin önyükleme veri kümesi (BSDS) bilgilerini yazdırmak için [Günlük haritası yazdırma yardımcı programı \(CSQJU004\)](#) olanağını kullanmanız ve günlük RBA 'yı içeren günlükleri bulmak için bu bilgileri kullanmalısınız.

CF yapıları için, LRSN 'yi içeren günlükleri bulmak için kuyruk paylaşım grubundaki her kuyruk yöneticisinde CSQJU004 yardımcı programını çalıştırmanız gerekir. Sayfa kümelerini ve yapılarını kurtarabilmek için bu günlüklere ve daha sonraki günlüklere gereksiniminiz vardır.

## DDname 'i sayfa kümesi ilişkilendirmesine değiştirmeyin

IBM MQ , 00 numaralı sayfayı DDname CSQP0000, sayfa numarası 01 olarak DDname CSQP0001ve bu şekilde CSQP0099ile ilişkilendirir. IBM MQ , sayfa kümesinin ilişkilendirildiği DDname 'ye dayalı olarak bir sayfa kümesine ilişkin kurtarma günlüğü kayıtlarını yazar. Bu nedenle, önceden bir PSID DDname ile ilişkilendirilmiş sayfa kümelerini taşıyamazsınız.

## Sayfa kümeleri kurtarıyor

Sayfa kümelerinin kurtarılması ve yeniden başlatma sürelerini en aza indirmeye zamanlarında yer alan etkenleri anlamak için bu konuyu kullanın.

Kurtarma stratejisinde önemli bir faktör, kuyruk yöneticisi kesintisine göz yumma zamanına neden olur. Toplam kesinti süresi, bir yedekten bir sayfanın kurtarılması için geçen süreyi içerebilir ya da olağandışı bir sonlandırma işleminden sonra kuyruk yöneticisini yeniden başlatın. Yeniden başlatma süresini etkileyen etmenler, sayfa kümelerinizi ne sıklıkta yedeklemenize ve denetim noktaları arasında günlüğe ne kadar veri yazılabildiğinizi içerir.

Olağandışı sona erdirme işleminden sonra yeniden başlatma süresini en aza indirmek için, sistem yeniden başlatıldığında, en çok iki etkin günlükten oluşan çalışma birimleri kısa bir süre boyunca devam eder. Örneğin, bir IBM MQ uygulaması tasarlıyorsanız, ilk syncpoint MQI çağrısı ile kesinleştirme noktası arasında uzun bekleme aralığına sahip bir MQGET çağrısı yerleştirmekten kaçınınız; bunun nedeni, uzun süreli bir iş biriminin sonuçlanmasına neden olabilir. Uzun çalışma birimlerinin bir diğer ortak nedeni, kanal başlatıcısı için 5 dakikadan fazla toplu aralıktır.

İş birimlerinin RBA 'sını görüntülemek ve eski olanları çözümlmek için [DISPLAY THREAD](#) (İş Parçasını Görüntüle) komutunu kullanabilirsiniz.

## Bir sayfa setini ne sıklıkla yedeklemeniz gerekir?

Makul bir kısa kurtarma süresi gerekiyorsa, sık kullanılan sayfa kümesi yedeklemesi gereklidir. Bu, bir sayfa kümesi çok küçük olduğunda ya da o sayfa kümesindeki kuyruklarda küçük bir etkinlik olduğunda da geçerlidir.

Bir sayfa kümesinde kalıcı iletiler kullanırsanız, yedekleme sıklığı günlerden çok saatler içinde olmalıdır. Bu, sayfa kümesi sıfır için de geçerli bir durumdur.

Yaklaşık bir yedekleme sıklığını hesaplamak için, hedef toplam kurtarma süresini belirlemeye başlayın. Bu, aşağıdaki bileşenlerden oluşur:

1. Soruna tepki göstermek için geçen süre.

2. Sayfa kümesi yedek kopyasını geri yüklemek için gereken süre.

SnapShot backup/restore işlemini kullanırsanız, bu görevi gerçekleştirmek için geçen süre birkaç saniye olur. SnapShothakkında bilgi için *DFSMSdss Storage Administration Guide* adlı belgeye bakın.

3. Kuyruk yöneticisinin yeniden başlatılması için gereken süre (örneğin, sayfa kümesinin kurtarılması için gereken ek süre de içinde olmak üzere).

Bu, sayfa kümesinin son yedeklenmesinden bu yana etkin ve arşiv günlüklerinden okunması gereken günlük verileri miktarına en çok bağlıdır. Bu tür günlük verilerinin tümü, zarar görmüş sayfa kümesiyle doğrudan ilişkili olarak okunmalıdır.

**Not:** *Fuzzy backup* (bulanık yedekleme) kullanılırken (bir iş birimi etkin durumdayken günlüklerin ve sayfa kümelerinin bir anlık görünümün alındığı durumlarda), en çok üç ek denetim noktasını okumak gerekebilir ve bu, bir ya da daha fazla ek günlük okuma gereksinimi ortaya çıkabilirdi.

Sayfa kümesinin kurtarılmasına ne kadar süre izin vereceğine karar verirken, göz önünde bulundurmanız gereken etkenler şunlardır:

- Normal işleme sırasında verilerin etkin günlüklere yazıldığı ücret, ileti hızının yanı sıra, sisteminize gelen iletilerin nasıl varolduğuna bağlıdır.

Bir kanal üzerinden alınan ya da gönderilen iletiler, yerel olarak üretilen ve alınan iletilerden daha fazla veri günlüğe kaydetme işlemi ile sonuçlanır.

- Verilerin arşivden ve etkin günlüklerden okunabileceği hız.

Günlükler okunurken, ulaşılabilir veri hızı, kullanılan aygıtlara ve belirli DASD altsistemine ilişkin toplam yük ile bağlıdır.

Çoğu manyetik bant birimiyle, büyük bir blok boyutuna sahip arşivlenmiş günlükler için daha yüksek veri hızlarına ulaşmak mümkündür. Ancak, kurtarma için bir arşiv günlüğü gerekiyorsa, etkin günlüklerdeki tüm veriler de okunmalıdır.

## CF yapılarının kurtarılıyor

CF yapılarına ilişkin kurtarma işlemini anlamak için bu konuyu kullanın.

RECOVER CFSTRUCT komutunu işlemek için kuyruk paylaşım grubunda en az bir kuyruk yöneticisinin etkin olması gerekir. CF yapısı kurtarma işlemi, zaten etkin bir kuyruk yöneticisi tarafından kurtarma işlemi gerçekleştirildiğinden, kuyruk yöneticisini yeniden başlatma süresini etkilemez.

Kurtarma işlemi, RECOVERY CFSTRUCT komutu tarafından yönetilen iki mantıksal adımdan oluşur:

1. Yedeğin bulunması ve geri yüklenmesi.
2. CF yapısında, CF yapısını kullanan tüm kuyruk yöneticilerinin günlüklerinden CF yapısında tutulan ve yedeklemede yapılan değişiklikleri uygulayarak, günlüğe kaydedilen tüm güncellemelerin birleştirilmesi.

Çok sayıda günlük verilerinin okunması gerekebileceğinden, ikinci adımın daha uzun sürmesi olasıdır. Sık sık yedeklenmenizi ya da birden çok CF yapısını aynı anda ya da her ikisini de kurtarırsanız, alınan süreyi azaltabilirsiniz.

Kurtarma işlemini gerçekleştiren kuyruk yöneticisi, Db2 içindeki verileri ve önyüklemeye veri kümelerini kullanarak diğer tüm kuyruk yöneticilerinin günlüklerinde ilgili yedekleri bulur. Kuyruk yöneticisi bu yedekleri, son yedeklemeden hemen önce hata noktasına kadar, kuyruk paylaşım grubunda doğru zaman sırasına göre yeniden yürütür.

Bir CF yapısını kurtarmak için gereken süre, yeniden oynatılması gereken kurtarma günlüğü verilerinin miktarına bağlıdır; bu, yedeklerin sıklığına bağlı olarak değişir. En kötü durumda, bir kuyruk yöneticisinin günlüğünü yazmak için gereken günlüğe yazıldığı sürece gerekir. Bu nedenle, örneğin, altı kuyruk yöneticisi içeren bir kuyruk paylaşım grubunuz varsa, bir saatlik günlük etkinliğinin yeniden yürütülmesi altı saat sürebilir. Genel olarak, okumanın toplu olarak yapılabileceği ve farklı kuyruk yöneticisinin günlüklerinin

paralel olarak okunabileceği için, genel olarak bundan daha az zaman alır. Başlangıç noktası olarak, CF yapılarınızı her saat yedeklemenizi tavsiye ederiz.

Başarısız bir CF yapısı varken, tüm kuyruk yöneticileri, paylaşılmayan kuyruklar ve diğer CF yapılarındaki kuyruklarla çalışmaya devam edebilir. Denetim yapısı da başarısız olursa, RECOVER CFSTRUCT komutunu verebilmek için kuyruk paylaşım grubundaki kuyruk yöneticilerinden en az birinin başlatılması gerekir.

CF yapılarının yedeklenmesi, önemli bir günlük yazma kapasitesi gerektirebilir ve bu nedenle yedekleme işlemi sırasında kuyruk yöneticisine büyük bir yük uygulayabilir. Yedeklemeler yapmak için hafif yüklenmiş bir kuyruk yöneticisi seçin; meşgul sistemler için, kuyruk paylaşım grubuna ek bir kuyruk yöneticisi ekleyin ve bunu yalnızca yedekler yapmak için ithaf edin.

## z/OS Belirli kurtarma hedeflerine ulaşılması

Yedekleme sıklığını ayarlayarak belirli kurtarma hedef sürelerini nasıl elde edebileceğiyle ilgili yönergeler için bu konuyu kullanın.

If you have specific recovery targets to achieve, for example, completion of the queue manager recovery and restart processing in addition to the normal startup time within xx seconds, you can use the following calculation to estimate your backup frequency (in hours):

$$\text{Backup frequency (in hours)} = \frac{\text{Required restart time (in secs)} * \text{System recovery log read rate (in MB/sec)}}{\text{Application log write rate (in MB/hour)}}$$

**Not:** Bundan sonra verilen örnekler, sayfa kümelerinizi sık sık yedeklemeniz gerisini vurgulamak üzere tasarlanmıştır. Hesaplamalar, çoğu günlük etkinliğinin çok sayıda kalıcı iletiden türetilmiş olduğunu varsayar. Ancak, günlük etkinliği miktarının kolay hesaplanmadığı durumlar vardır. Örneğin, bir kuyruk paylaşım grubu ortamında, diğer kaynaklara ek olarak paylaşılan kuyrukların güncellendiği bir iş birimi UOW kayıtlarının IBM MQ günlüğüne yazılmasına neden olabilir. Bu nedenle, Formula (A) uygulamasındaki Uygulama günlüğü yazma hızı, yalnızca IBM MQ günlüklerinin dolduracağı gözlemlenen hızdan doğru olarak türetilir.

Örneğin, IBM MQ MQI clients ' un saniyede 100 kalıcı ileti yüklemesi işlemi oluşturacağı bir sistem düşünün. Bu durumda, tüm iletiler yerel olarak oluşturulur.

Her bir ileti 1 KB ' lik kullanıcı uzunluğu ise, her saat günlüğe kaydedilen veri miktarı yaklaşık olarak şöyledir:

$$100 * (1 + 1.3) \text{ KB} * 3600 = \text{approximately } 800 \text{ MB}$$

where

100	= the message rate a second
(1 + 1.3) KB	= the amount of data logged for each 1 KB of persistent messages

75 dakika boyunca genel bir hedef kurtarma süresi göz önünde bulundurun. Soruna yanıt vermek için 15 dakika izin verdiyseniz ve sayfa kümesi yedek kopyasını geri yüklüyorsanız, kuyruk yöneticisi kurtarma ve yeniden başlatma işlemi, formül (A) uygulayarak 60 dakika (3600 saniye) içinde tamamlanmalıdır. Gerekli tüm günlük verilerinin RVA2-T82 DASD üzerinde olduğunu varsayarsak, bu da yaklaşık 2.7 MB ' lik bir kurtarma hızına sahiptir ve bu da bir sayfa kümesi yedekleme sıklığını en az her bir saniyeye döndürmektedir:

3600 seconds \* 2.7 MB a second / 800 MB an hour = 12.15 hours

IBM MQ uygulama gününüz yaklaşık 12 saat sürerse, her gün bir yedekleme uygun olur. Ancak, uygulama günü 24 saat sürerse, her gün iki yedek daha uygun olur.

Başka bir örnek de, tüm iletilerin istek-yanıt uygulamaları için olduğu bir üretim sistemi olabilir (yani, bir alıcı kanalında kalıcı bir ileti alınır ve kalıcı bir yanıt iletilisi oluşturulur ve bir gönderen kanalı aşağı gönderilir).

Bu örnekte, elde edilen toplu iş boyutu bir 'dir ve bu nedenle her ileti için tek bir küme vardır. Bir saniyeye 50 istek yanıtı varsa, toplam yükleme 100 kalıcı ileti saniyedir. Her ileti 1 KB uzunluğdaysa, günlüğe kaydedilen veri miktarı yaklaşık olarak şöyledir:

$50((2 * (1+1.3) \text{ KB}) + 1.4 \text{ KB} + 2.5 \text{ KB}) * 3600 = \text{approximately } 1500 \text{ MB}$

where:

50 = the message pair rate a second  
(2 \* (1 + 1.3) KB) = the amount of data logged for each message pair  
1.4 KB = the overhead for each batch of messages received by each channel  
2.5 KB = the overhead for each batch of messages sent by each channel

Kuyruk yöneticisi kurtarma işlemini gerçekleştirmek ve 30 dakika (1800 saniye) içinde yeniden başlatmak için, gerekli tüm günlük verilerinin RVA2-T82 DASD ' de olduğunu varsayarsak, bu, sayfa kümesi yedeklemenin en az her biri için gerçekleştirilmesini gerektirir:

1800 seconds \* 2.7 MB a second / 1500 MB an hour = 3.24 hours

## Yedekleme sıklığının periyodik olarak incelenmesi

IBM MQ günlük kullanımınızı MB olarak bir saat olarak izleyin. Bu onay kutusunu düzenli aralıklarla gerçekleştirin ve gerekirse, sayfa ayarı yedekleme sıklığını yeniden yapın.

## Diğer ürünlere ilişkin yedekleme konuları

If you are using IBM MQ with CICS or IMS then you must also consider the implications for your backup strategy with those products. Veri olanağı hiyerarşik depolama yöneticisi (DFHSM), veri depolamayı yönetir ve IBM MQ tarafından kullanılan depolama ile etkileşim kurabilirler.

## DFHSM ile yedekleme ve kurtarma

Veri olanağı hiyerarşik depolama yöneticisi (DFHSM), sisteminizdeki depolama aygıtları arasında yer alan otomatik alan kullanılabilirliği ve veri kullanılabilirliği yönetimini gerçekleştirir. Bunu kullanırsanız, verilerin IBM MQ depolama alanından otomatik olarak ve otomatik olarak taşındığını bilmeniz gerekir.

DFHSM, kısa bir süre önce alternatif depolamaya kullanılmayan veri kümelerini taşıyarak DASD alanınızı verimli bir şekilde yönetir. Ayrıca, yeni ya da değiştirilen veri kümelerini manyetik bant ya da DASD yedekleme birimlerine otomatik olarak kopyalayarak, verilerinizin kurtarma için kullanılabilir olmasını da sağlar. Veri kümelerini silebilir ya da başka bir aygıtta taşıyabilir. İşlemlerin işlemleri her gün, belirli bir zamanda gerçekleşir ve silmeden ya da taşımadan önce önceden belirlenmiş bir dönem için veri kümesi tutulmasına izin verir.

Tüm DFHSM işlemlerini el ile de gerçekleştirebilirsiniz. *Data Facility Hierarchical Storage Manager User's Guide* , DFHSM komutlarının nasıl kullanılacağını açıklar. If you use DFHSM with IBM MQ, note that DFHSM does the following:

- Kataloğa alınmış veri kümeleri kullanır.
- Sayfa kümeleri ve günlükler üzerinde çalışır.
- VSAM veri kümelerini destekler.

## Kurtarma ve CICS

CICS kaynaklarının kurtarılması, IBM MQ' in varlığından etkilenmez. CICS , IBM MQ ' yi CICS kaynağı olmayan bir kaynak (ya da dış kaynak yöneticisi) olarak tanıır ve CICS kaynak yöneticisi arabirimini (RMI) kullanarak herhangi bir eşitleme noktası eşgüdümü isteklerinde katılımcı olarak IBM MQ içerir. CICS kurtarma hakkında daha fazla bilgi için *CICS Recovery and Restart Guide* adlı yayına bakın. CICS kaynak yöneticisi arabirimine ilişkin bilgi için *CICS Customization Guide* adlı belgeye bakın.

## Kurtarma ve IMS

IMS recognizes IBM MQ as an external subsystem and as a participant in syncpoint coordination. Dış altsistem kaynaklarına ilişkin IMS kurtarma işlemi, *IMS Customization Guide* adlı belgede ele lanır.

### Alternatif bir site için kurtarma hazırlığı yapıyor

Toplam IBM MQ bilgi işlem merkezinin kaybolması durumunda, kurtarma yerinde başka bir IBM MQ sisteminde kurtarma işlemi yapabilirsiniz.

Kurtarma yerinde bir IBM MQ sistemini kurtarmak için, sayfa kümelerini ve günlükleri düzenli olarak yedeklemeniz gerekir. Tüm veri kurtarma işlemlerinde olduğu gibi, olağanüstü durumdan kurtarma işleminin amaçları, küçük veri, iş yükü işleme (güncelleme) ve mümkün olan süre olarak kaybetmeye devam eder.

Kurtarma yerinde:

- Kurtarma IBM MQ kuyruk yöneticisi **gerekir** , kaybedilen kuyruk yöneticisiyle aynı adı aldı.
- Kurtarma kuyruğu yöneticinde kullanılan sistem değiştirge biriminin, kaybedilen kuyruk yöneticisiyle aynı parametreleri içerdiğinden emin olun.

Olağanüstü durumdan kurtarma işlemine ilişkin işlem [Yönetme IBM MQ for z/OS](#) içinde açıklanmaktadır.

### Kuyruk yöneticisi yedekleme etkinliği örneği

Bu konu, kuyruk yöneticisi yedekleme etkinliğine ilişkin bir örnek olarak gösterilir.

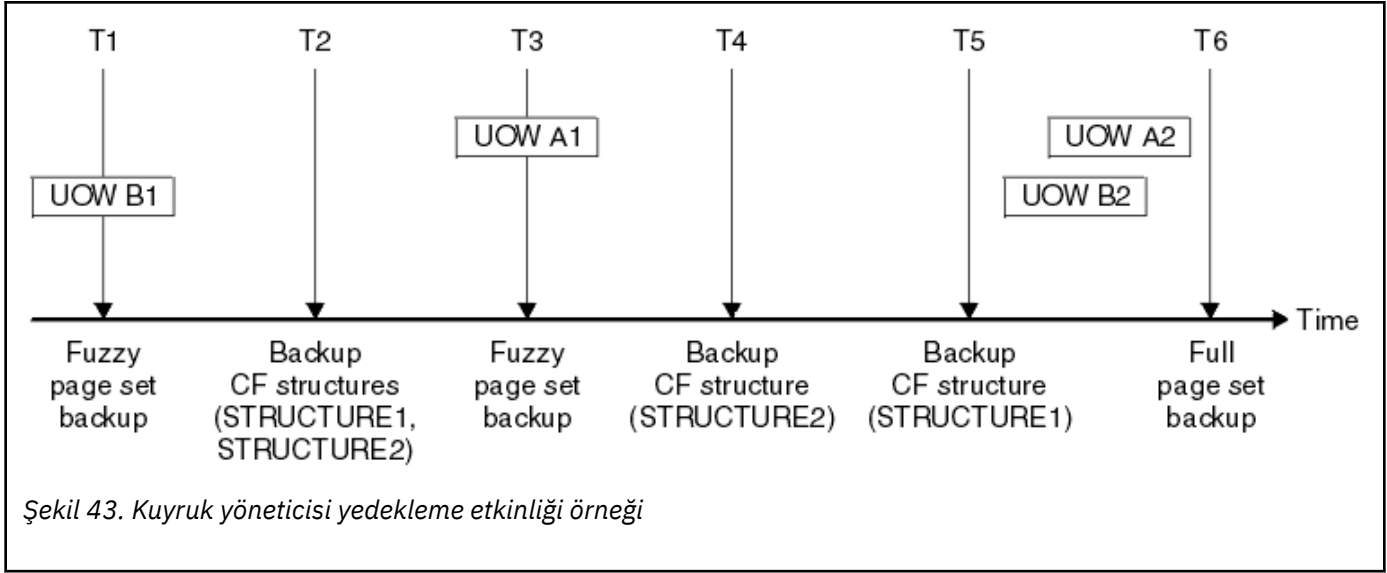
Kuyruk yöneticisi yedekleme stratejinizi planlarken, önemli bir dikkat edilmesi gereken günlük verileri tutarının tutulmasını sağlar. [Günlüklerin yönetilmesi](#) , kuyruk yöneticisinin sistem kurtarma RBA ' sına başvuru yaparak hangi günlük veri kümelerinin gerekli olduğunu nasıl belirleyeceğini açıklar. IBM MQ , aşağıdaki bilgilerle ilgili bilgileri kullanarak sistem kurtarma RBA ' yı belirler:

- Şu anda etkin iş birimleri.
- Sayfa kümesi güncellemeleri, arabellek havuzlarından diske henüz temizlenmemiş olarak ayarlanır.
- CF yapısı yedekleri ve bu kuyruk yöneticisinin günlüğü, bunları kullanan herhangi bir kurtarma işleminde gereken bilgileri içerip içermediğini.

Ortam kurtarma işlemi gerçekleştirebilmek için yeterli günlük verilerini saklamalısınız. Sistem kurtarma RBA zaman içinde artış gösterirken, alıkonması gereken günlük verileri miktarı, yalnızca sonraki yedekleme işlemleri yapılırken azalır. CF yapısı yedeklemeleri IBM MQ tarafından yönetilir ve bu nedenle, sistem kurtarma RBA ' yı raporlarken dikkate alınır. Bu, uygulamada, alıkonması gereken günlük verilerinin miktarı, yalnızca sayfa kümesi yedeklemeleri alındığında azaltılır.



Şekil 43 sayfa 208 , bir kuyruk paylaşım grubunun üyesi olan bir kuyruk yöneticisiyle ilgili yedek etkinliğe bir örnek gösterir; kurtarma RBA ' nın her bir yedeklemeyle nasıl değişir ve bu durum, alıkonması gereken günlük verileri miktarını nasıl etkiler. Örnek olarak, kuyruk yöneticisi yerel ve paylaşılan kaynakları kullanır: sayfa kümeleri ve iki CF yapısı, STRUCTURE1 ve STRUCTURE2.



Zaman içinde her noktada olan şey budur:

#### Zaman noktası T1

Yaklaşık bir yedekleme, Sayfa kümelerini nasıl yedeklemeniz ve kurtarmanız içinde açıklandığı şekilde sayfa kümelerinizin yaratılışdır.

Kuyruk yöneticisinin sistem kurtarma RBA 'si, aşağıdakilerin en düşüğüdür:

- Bu noktada yedeklenmekte olan sayfa kümelerinin kurtarma RBU ' ları.
- CF uygulama yapılarını kurtarmak için gerekli olan en düşük kurtarma RBA ' sı. Bu, daha önce yaratılan STRUCTURE1 ve STRUCTURE2 yedeklemelerinin kurtarılması ile ilgilidir.
- Kuyruk yöneticisi (UOWB1) içinde çalışmakta olan en eski iş birimi için kurtarma RBA ' yı (RBA).

Bu zaman noktasına ilişkin sistem kurtarma RBA, bulanık yedekleme işleminin bir parçası olan DISPLAY USAGE komutu tarafından verilen iletiler tarafından verilir.

#### Zaman noktası T2

CF yapılarının yedeklemeleri yaratılır. CF yapısı STRUCTURE1 önce yedeklenir, ardından STRUCTURE2 işlemleri devam eder.

The amount of log data that must be retained is unchanged, because the same data as determined from the system recovery RBA at T1 is still required to recover using the page set backups taken at T1.

#### Zaman noktası T3

Başka bir yedek yedekleme işlemi yaratılır.

Kuyruk yöneticisinin sistem kurtarma RBA 'si, aşağıdakilerin en düşüğüdür:

- Bu noktada yedeklenmekte olan sayfa kümelerinin kurtarma RBU ' ları.
- The lowest recovery RBA required to recover CF structure STRUCTURE1, because STRUCTURE1 was backed up before STRUCTURE2.
- Kuyruk yöneticisi (UOWA1) içinde çalışmakta olan en eski iş birimi için kurtarma RBA 'si.

Bu zaman noktasına ilişkin sistem kurtarma RBA, bulanık yedekleme işleminin bir parçası olan DISPLAY USAGE komutu tarafından verilen iletiler tarafından verilir.

Bu yeni sistem kurtarma RBA ' nın belirlediği şekilde, artık günlük verilerinin alıkonabileceği şekilde azaltabilirsiniz.



#### Zaman noktası T4

A backup is taken of CF structure STRUCTURE2. The recovery RBA for the recovery of the oldest required CF structure backup relates to the backup of CF structure STRUCTURE1, which was backed up at time T2.

Bu CF yapısı yedeğinin yaratılması, alıkonması gereken günlük verileri miktarı üzerinde bir etki oluşturmaz.

#### Zaman noktası T5

A backup is taken of CF structure STRUCTURE1. The recovery RBA for recovery of the oldest required CF structure backup now relates to recovery of CF structure STRUCTURE2, which was backed up at time T4.

Bu CF yapısı yedeğinin yaratılması, alıkonması gereken günlük verileri miktarı üzerinde bir etki oluşturmaz.

#### Zaman noktası T6

A full backup is taken of your page sets as described in [Sayfa kümelerini nasıl yedeklemeniz ve kurtarmanız](#).

Kuyruk yöneticisinin sistem kurtarma RBA 'si, aşağıdakilerin en düşüğüdür:

- Bu noktada yedeklenmekte olan sayfa kümelerinin kurtarma RBU ' ları.
- CF yapılarını kurtarmak için gerekli olan en düşük kurtarma RBA ' sı. Bu, STRUCTURE2CF yapısının kurtarılması ile ilgilidir.
- Yürürlükteki etkin iş birimi için kuyruk yöneticisi içindeki en eski iş birimi için RBA kurtarma işlemi. Bu durumda, yürürlükteki iş birimi yok.

Bu zaman noktasına ilişkin sistem kurtarma RBA, tam yedekleme işleminin bir parçası olan DISPLAY USAGE komutu tarafından verilen iletiler tarafından verilir.

Yine, günlük verilerinin alıkonması azaltılabilir; bunun nedeni, tam yedekle ilişkili sistem kurtarma RBA ' sının daha yeni bir son olması olabilir.

z/OS

## z/OS UNIX ya da UNIX System Services ortamınızın planlanması

IBM MQ kuyruk yöneticisi (MSTR), kanal başlatıcı (CHIN) ve mqweb sunucusu içindeki bazı işlemler, olağan işlemleri için z/OS UNIX ya da UNIX System Services (USS) olanağını kullanır. Varsayılan USS yapılandırmasını kullanmak istemezseniz yapılandırmanızı planlayın.

Bir sistem çapında varsayılan OMVS kesimi ayarlandığı sürece, IBM MQ ' in UNIX hizmetlerini kullanması için özel bir işlem ya da özelleştirme gerekmez.

IBM MQ 'un USS' yi konuk ya da varsayılan UID ve OMVS kesimini kullanarak çağırmasını istemiyorsanız, yeni bir OMVS kesimi tanımlayın. Tanımladığınız OMVS kesimi varsayılan segmente modellenmiş olabilir; IBM MQ için özel bir izin gerektirmediği ve bir ayrıcalıklı kullanıcı olarak USS içinde çalıştırılmamasını sağlar.

MSTR ve CHIN 'de başlatılan görev kullanıcı tanıtıcıları, RACF OMVS kesiminde bir UID' ye de ihtiyaç duyarlar.

**Not:** Although the MSTR and CHIN jobs make use of USS facilities (for example, to interface with TCP/IP services), they do not need to access any of the content of the USS file system shipped by IBM MQ. Sonuç olarak, MSTR ve CHIN işleri, USS dosya sisteminin yolunu belirlemek için herhangi bir yapılandırma gerektirmez.

V9.1.0

IBM MQ Console ve REST API sistemlerini barındıran mqweb sunucusu, USS dosya sistemindeki IBM MQ kuruluş dizininde bulunan dosyaları kullanır. Ayrıca, yapılandırma ve günlük dosyaları gibi verileri depolamak için kullanılan başka bir dosya sistemine de erişim gerektirir. Bu USS dosya sistemlerine gönderme yapmak için mqweb 'in başlattığı görev JCL ' nin uyarlanmış olması gerekir.

USS dosya sistemindeki IBM MQ dizininin içeriği, IBM MQ' a bağlanan uygulamalar tarafından da kullanılır. Örneğin, IBM MQ classes for Java ya da IBM MQ classes for JMS arabirimlerini kullanan uygulamalar.

İlgili yapılandırma yönergeleri için aşağıdaki konulara bakın:

- [IBM MQ classes for Java ile ilgili ortam değişkenleri](#)
- [IBM MQ classes for Java kitaplıkları](#)
- [Ortam değişkenlerini tanımlama](#)
- [Java Native Interface \(JNI\) kitaplıklarının yapılandırılması](#)

z/OS

## Advanced Message Security Planlaması

TLS (or SSL) can be used to encrypt and protect messages flowing on a network, but this does not protect messages when they are on a queue ("at rest"). Advanced Message Security (AMS) protects the messages from the time that they are first put to a queue, until they are got, so that only the intended recipients of the message can read that message. İletiler, işleme sırasında şifrelenir ve imzalanır ve alma işlemi sırasında korumasız olarak imzalanır.

AMS , iletileri farklı şekillerde korumak için yapılandırılabilir:

1. Bir ileti imzalanabilir. İleti açık bir metindir, ancak bir sağlama toplamı vardır ve bu da imzalanır. Bu, ileti içeriğindeki değişikliklerin algılanmasını sağlar. İmzalı içerikten, verileri kimin imzalamış olduğunu saptayabilirsiniz.
2. Bir ileti şifrelenebilir. İçerik, şifre çözme anahtarı olmayan herkes tarafından görülemez. Şifre çözme anahtarı her alıcı için şifrelenir.
3. Bir ileti şifrelenebilir ve imzalanabilir. Şifre çözme anahtarı her alıcı için şifrelenir ve imzalamadan iletiyi kimin gönderdiğini belirleyebilirsiniz.

Şifreleme ve imzalama sayısal sertifikalar ve anahtar halkaları kullanır.

You can set up a client to use AMS, so the data is protected before the data is put on the client channel. Korunan iletiler uzak bir kuyruk yöneticisine gönderilebilir ve bu iletileri işlemek için uzak kuyruk yöneticisini yapılandırmanız gerekir.

### ayarlar AMS

AMS iş yapma işlemi için bir AMS adres alanı kullanılır. Bu, anahtar halkalarının ve sertifikaların kullanımına erişim vermek ve bunlara erişim vermek için ek güvenlik ayarına sahiptir.

Kuyruklara ilişkin güvenlik ilkelerini tanımlamak için bir yardımcı program (CSQOUTIL) kullanılarak hangi kuyrukların korunacağı yapılandırılır.

### AMS ayarlandıktan sonra

Bir dijital sertifika ve mesaj koyan insanlar için bir anahtarlık ve mesaj alan kişiler için bir anahtarlık ayarlamamız gerekir.

Bir kullanıcı, Alice, z/OS 'un Bob 'a bir ileti göndermesi gerekirse, AMS ' un Bob için genel sertifikasının bir kopyasına ihtiyacı vardır.

Bob, Alice 'ten bir ileti almak isterse, AMS ' in Alice için genel sertifikasına ya da Alice tarafından kullanılan sertifika yetkilisi sertifikasına sahip olması gerekir.



**Uyarı:** Aşağıdakine gereksinim duyarsınız:

- Kuyruklara kimin koyabileceğini ya da kimden bilgi alabileceğini dikkatlice planlayın
- Kişileri ve bunların sertifika adlarını belirleyin.

Hata yapmak çok kolay ve sorunları çözmek zor olabilir.

### İlgili kavramlar

“Kuyruk yöneticinizin planlanması” sayfa 157

Bir kuyruk yöneticisi ayarlarken, kuyruk yöneticisinin kuruluşunuzun gereksinimlerini karşıladığı için, planlama yöneticisi kuyruk yöneticisinin büyümesine izin vermelidir.

## Managed File Transfer planlaması

Use this section as guidance on how you need to set up your system to run Managed File Transfer (MFT) on z/OS.

## Managed File Transfer -donanım ve yazılım gereksinimlerinin planlanması

Use this topic as guidance on how you need to set up hardware and software requirements on your system to run Managed File Transfer (MFT) on z/OS.

### Yazılım gereksinimleri

Managed File Transfer , programı yapılandırabilmek ve çalıştırmak için bazı kabuk komut dosyaları ve JCL ile Java' te yazılır.

**Önemli:** Managed File Transfer' u yapılandırmak için z/OS UNIX sistem hizmetleri (z/OS UNIX) ile ilgili bilgi sahibi olmanız gerekir. Örneğin:

- The file directory structure, with names such as /u/userID/myfile.txt
- z/OS UNIX komutları, örneğin:
  - cd (dizin değiştir)
  - ls (Liste)
  - chmod (dosya izinlerini değiştirin)
  - chown (dosya ya da dizine erişebilen dosya sahipliğini ya da grupları değiştirin)

You require the following products in z/OS UNIX to be able to configure and run MFT:

1. Örneğin, Javadizinde, /java/java80\_bit64\_GA/J8.0\_64/
2. Örneğin, IBM MQ 9.1.0 dizinde, /mqm/V9R2M0
3. If you want to use Db2 for status and history, you need to install Db2 JDBC libraries, for example, in directory /db2/db2v10/jdbc/libs.

### Ürün kaydı

At startup Managed File Transfer checks the registration in sys1.parmlib(IFAPRDxx) concatenation. Aşağıdaki kod, MFT ' yi nasıl kaydettiğinizi gösteren bir örnektir:

```
PRODUCT OWNER('IBM CORP')
NAME('WS MQ FILE TRANS')
ID(5655-MFT)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('WS MQ FILE TRANS')
STATE(ENABLED)
```

### Disk Yeri

**UNIX** PDE 'ler için 100 MB' lik DASD ve z/OS UNIX içinde en az 50 MB gerekir. Sorunları tanılamak için izlemeyi kullandıysanız, z/OS UNIX içinde ek disk alanına gereksinim duyarsınız; örneğin, 50 MB.

## Managed File Transfer -topolojilerin planlanması

Use this topic as guidance on what topology you need on your system to run Managed File Transfer (MFT) on z/OS.

## Managed File Transfer kuyruk yöneticileri

IBM MQ Managed File Transfer topolojileri şunlardan oluşur:

### Aracılar ve ilişkili kuyruk yöneticileri

Aracı, durum bilgilerini korumak ve işe ilişkin istekleri almak için Agent kuyruk yöneticilerinde barındırılan sistem kuyruklarını kullanır.

### Bir komut kuyruğu yöneticisi

Bu, MFT topolojisine bir ağ geçidi işlevi görür. Bu ad, gönderen ve alıcı kanalları ya da kümeleme yoluyla aracı kuyruk yöneticilerine bağlanır. Bazı komutlar çalıştırıldığında, doğrudan komut kuyruğu yöneticisine bağlanır ve belirtilen aracıya bir ileti gönderir. Bu ileti, IBM MQ ağı aracılığıyla aracı kuyruk yöneticisine yönlendirilir; bu ileti aracı tarafından alınıp işlenilir.

### Bir koordinasyon kuyruk yöneticisi

Bu merkezi, topolojinin tamamını bilen merkezi bir merkezdir. Eşgüdüm kuyruk yöneticisi, gönderici ve alıcı kanalları aracılığıyla ya da kümelemeyi kullanarak bir topolojideki tüm aracı kuyruğu yöneticilerine bağlıdır. Araçılar, durum bilgilerini düzenli olarak koordinasyon kuyruğu yöneticisine yayınlamaya ve aktarım şablonlarını orada saklar.

Tek bir kuyruk yöneticisinin bir topoloji içinde birden çok rol gerçekleştirmesi mümkündür. Örneğin, aynı kuyruk yöneticisi hem koordinasyon kuyruğu yöneticisi, hem de bir topoloji için komut kuyruğu yöneticisi olarak yapılandırılabilir.

Birden çok kuyruk yöneticisi kullanıyorsanız, kuyruk yöneticileri arasında kanallar kurmanız gerekir. Bunu yapmak için kümelemeyi kullanarak ya da noktadan noktaya iletişim bağlantılarını kullanarak yapabilirsiniz.

IBM MQ Managed File Transfer for z/OS kullanırken, bir topoloji içinde farklı roller için hangi kuyruk yöneticilerinin kullanılacağını belirlerken dikkate alınacak birçok şey vardır.

### Aracı kuyruk yöneticileri

Bir IBM MQ Managed File Transfer for z/OS aracı için aracı kuyruk yöneticisi z/OS üzerinde çalışıyor olmalıdır.

Eğer:

- The agent is running Managed File Transfer for z/OS on IBM MQ 9.1 or later
  - Ve aracı kuyruk yöneticisi, IBM MQ Advanced for z/OS Value Unit Edition (Advanced VUE) için lisanslanır.
- aracı, CLIENT iletimini kullanarak kuyruk yöneticisine bağlanabilir.



Şekil 44. Kuyruk yöneticisinin Advanced VUE için lisanslanmış olduğu varsayılarak, z/OS üzerindeki MFT 9.1 araçları, CLIENT iletimi kullanılarak bir kuyruk yöneticisine bağlanabilir.

Eğer:

- Aracı IBM MQ 9.0 ya da daha önceki bir sürümle Managed File Transfer for z/OS çalıştırıyor
- Or, the agent queue manager is running Managed File Transfer for z/OS on IBM MQ 9.0 or later, and the agent queue manager is licensed for either MFT, IBM MQ Advanced for z/OS, or Advanced VUE

Aracının, BAĞLAMALAR iletimini kullanarak kuyruk yöneticisine bağlanması gerekir.



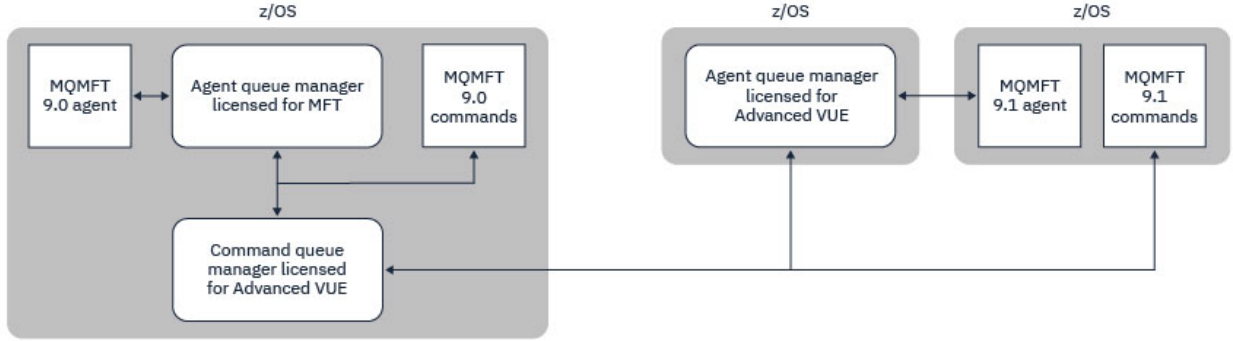
Şekil 45. MFT ya da IBM MQ Advanced için lisanslanan bir aracı kuyruk yöneticisine sahip z/OS ve 9.1 araçlarındaki MFT 9.0 araçları, BATICE (Bağlamalar) iletimi kullanılarak bağlantı kurulmalıdır.

### Komut kuyruğu yöneticileri

Hangi MFT komutları ve işlemleri hangi kuyruk yöneticisine bağlanır konusu, bir Managed File Transfer topolojisi için komut kuyruğu yöneticisine bağlanan tüm komutları gösterir.

**Not:** Bu komutları z/OS üzerinde çalıştırırken, komut kuyruğu yöneticisinin de z/OS üzerinde olması gerekir.

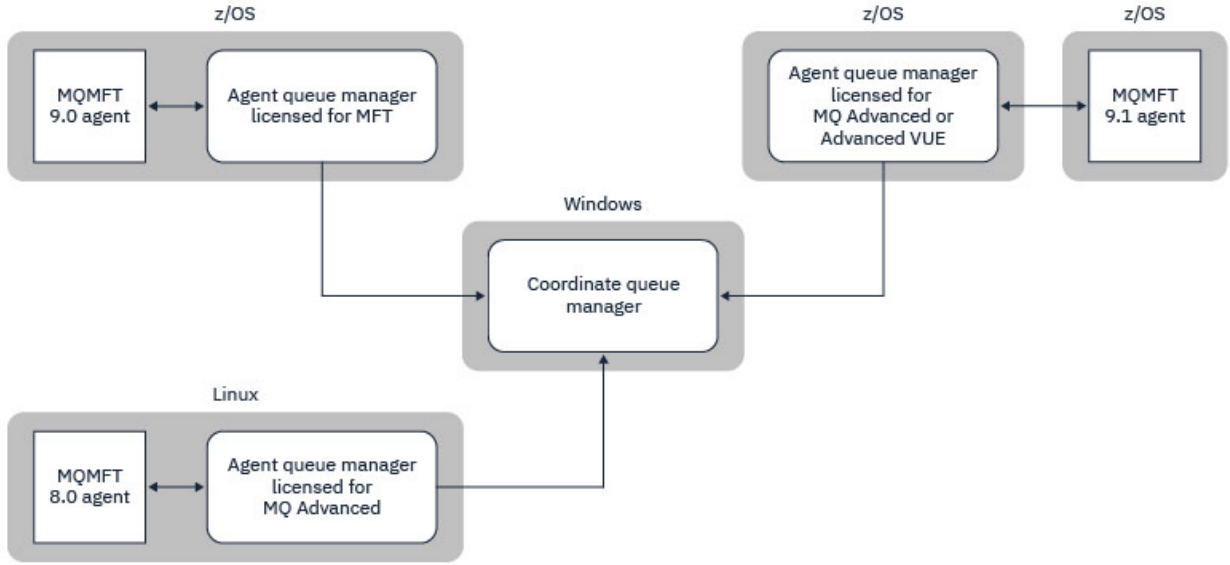
Komut kuyruk yöneticisi Advanced VUE için lisanslandıysa, komutlar CLIENT iletimi kullanılarak kuyruk yöneticisine bağlanabilirler. Tersi durumda, komutların CLUDE iletimi kullanılarak komut kuyruğu yöneticisine bağlanması gerekir.



Şekil 46. Komutlar, MFT topolojisi için komut kuyruk yöneticisine bağlanır. Bu komutları z/OS üzerinde çalıştırırken, komut kuyruğu yöneticisinin de z/OS üzerinde olması gerekir.

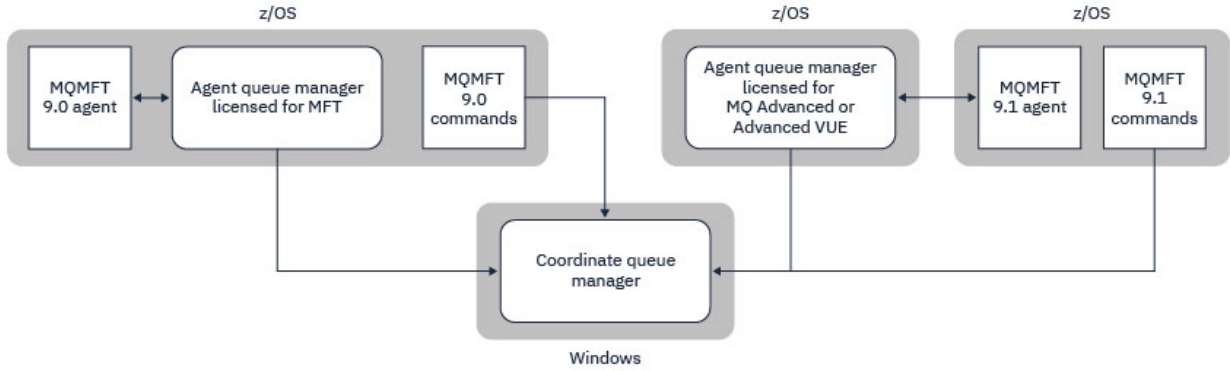
### Koordinasyon kuyruğu yöneticileri

IBM MQ Managed File Transfer for z/OS agents can be part of a topology where the coordination queue manager is either running on z/OS, or is running on a multiplatform.



Şekil 47. z/OS üzerinde çalışan MFT araçları, koordinasyon kuyruğu yöneticisinin bir IBM MQ çoklu platformunda çalıştırıldığı bir MFT topolojisinin parçası olabilir.

Hangi MFT komutları ve işlemleri hangi kuyruk yöneticisine bağlanır konusu, bir Managed File Transfer topolojisi için eşgüdüm kuyruk yöneticisine bağlanan komutları gösterir. Bu komutların z/OS üzerinde çalıştırılması ve daha sonra, farklı bir platformda çalışan koordinasyon kuyruğu yöneticisine bağlanmanızı mümkündür.



Şekil 48. **ftelistsagents** gibi bazı komutlar, bir MFT topolojisi için doğrudan koordinasyon kuyruk yöneticisine bağlanır.

## Kaç ajana ihtiyacım var?

Aracılar veri aktarmada işleri yapar ve veri aktarma isteğinde bulunduğunuzda, bir aracının adını belirtmenizi sağlar.

Varsayılan olarak bir aracı, 25 gönderme ve 25 alma isteğini eşzamanlı olarak işleyebilir. Bu işlemleri yapılandırabilirsiniz. Daha fazla bilgi için [z/OS üzerindeki Managed File Transfer yapılandırma seçenekleri](#) başlıklı konuya bakın.

Aracı meşgulse, iş kuyruklanır. Bir isteği işleme süresi, birden çok etkene bağlıdır; örneğin, gönderilecek veri miktarı, ağ bant genişliği ve ağ üzerindeki gecikme gibi.

Çalışmayı koşturarak işlemek için birden çok aracıya sahip olmak isteyebilirsiniz.

Ayrıca, bir aracının hangi kaynaklara erişebileceğini de denetleyebilirsiniz, bu nedenle bazı araçların sınırlı bir veri alt kümesiyle çalışmasını isteyebilirsiniz.

İstekleri farklı önceliğe sahip olarak işlemek istiyorsanız, birden çok aracı kullanabilir ve iş yükü yöneticisini kullanarak işlerin önceliğini belirleyebilirsiniz.

## Aracıları çalıştırma

Genellikle araçlar uzun süredir çalışmakta olan süreçlerdir. İşlemler, toplu kipte çalışan ya da başlatılan görevler olarak sunulabilir.

## z/OS Managed File Transfer ' in planlanması-güvenlik konuları

Use this topic as guidance on what security considerations you need on your system to run Managed File Transfer (MFT) on z/OS.

## Güvenlik

MFT yapılandırması ve MFT işlemi için hangi kullanıcı kimliklerinin kullanılacağını saptamaya gerek duyarsınız.

Aktardığınız dosyaları ya da kuyrukları belirlemeniz ve hangi kullanıcı kimliklerinin MFT ' ye aktarma isteklerini göndereceğini belirlemeniz gerekir.

Aracıları ve günlüğe kaydediciyi özelleştirirken, MFT hizmetlerini çalıştırmasına izin verilen kullanıcı grubunu ya da MFT yönetimini belirtiyorsunuz.

MFT ' yi uyarlamaya başlamadan önce bu grubu ayarlamalısınız. MFT, IBM MQ kuyruklarını kullandıkça, kuyruk yöneticisinde güvenlik etkinleştirildiyse, MFT aşağıdaki kaynaklara erişim gerektirir:

Çizelge 26. MQADMIN kaynak sınıfı	
Ad	Erişim gereklidir
QUEUE.SYSTEM.FTE.EVENT.agent_name	Güncelle
QUEUE.SYSTEM.FTE.COMMAND.agent_name	Güncelle
CONTEXT.SYSTEM.FTE.COMMAND.agent_name	Güncelle
QUEUE.SYSTEM.FTE.STATE.agent_name	Güncelle
QUEUE.SYSTEM.FTE.DATA.agent_name	Güncelle
QUEUE.SYSTEM.FTE.REPLY.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHAGT1.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHTRN1.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHOPS1.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHSCH1.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHMON1.agent_name	Güncelle
QUEUE.SYSTEM.FTE.AUTHADM1.agent_name	Güncelle

Çizelge 27. MQQUEUE kaynak sınıfı	
Ad	Erişim gereklidir
SYSTEM.FTE.AUTHAGT1.agent_name	Güncelle
SYSTEM.FTE.AUTHTRN1.agent_name	Güncelle
SYSTEM.FTE.AUTHOPS1.agent_name	Güncelle
SYSTEM.FTE.AUTHSCH1.agent_name	Güncelle

Çizelge 27. MQUEUE kaynak sınıfı (devamı var)	
Ad	Erişim gereklidir
SYSTEM.FTE.AUTHMON1.agent_name	Güncelle

Aktarma isteğinde bulunan kullanıcının dosya sisteminin hangi kısımlarıyla erişebileceğini belirlemek için kullanıcı kum bokunu kullanabilirsiniz.

Kullanıcı kum bokunu etkinleştirmek için, kısıtlamak istediğiniz aracıya ilişkin *agent.properties* dosyasına `userSandboxes=true` deyimini ekleyin ve `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` dosyasına uygun değerler ekleyin.

Ek bilgi için [Kullanıcı çalışma yerleriyle çalışma başlıklı](#) konuya bakın.

Bu kullanıcı kimliği `UserSandboxes.xml` kütüklerinde yapılandırılır.

Bu XML dosyasında kullanıcı kimliği ya da kullanıcı kimliği\* ve kullanılabilir (içerilen) bir kaynak listesi var ya da kullanılamaz (dışlanmış). Hangi kaynaklara erişebilecek belirli kullanıcı kimliklerini tanımlamanız gerekir: örneğin:

Çizelge 28. Örnek kullanıcı kimliği, belirli kaynaklara erişimle birlikte			
Kullanıcı kimliği	Erişim	Dahil et ya da Har	Kaynak
Yönetici *	Oku	Ekle	/home/user/**
Yönetici *	Oku	Dışla	/home/user/private/**
Sysprog	Oku	Ekle	/home/user/**
Yönetici *	Oku	Ekle	Application.reply.queue

#### Notlar:

1. `type=queue` belirtilirse, kaynak bir kuyruk adı ya da `queue@qmgr` olur.
2. Kaynak `//` ile başlıyorsa, kaynak bir veri kümesiyle başlar; tersi durumda kaynak, z/OS UNIX içindeki bir dosyadır.
3. Kullanıcı kimliği, MQMD yapısındaki kullanıcı kimliğidir; bu nedenle, iletiyi koyan kullanıcı kimliği bu kullanıcı kimliğini yansıtmayabilir.
4. Yerel kuyruk yöneticisiyle ilgili isteklerde, MQADMIN CONTEXT.\* olanağını kullanabilirsiniz. hangi kullanıcıların bu değeri ayarlayabileceğini sınırlandırmak için.
5. Uzak kuyruk yöneticisinde gelen istekler için, dağıtılmış kuyruk yöneticilerinin, MQMD yapısındaki kullanıcı kimliğinin yetkisiz olarak belirlenmesini önlemek için güvenliği etkinleştirdiğini varsaymanız gerekir.
6. A user ID of SYSPROG1 on a Linux machine, is the same user ID SYSPROG1 for the security checking on z/OS.

## z/OS V9.1.0 Planning to use the IBM MQ Console and REST API on z/OS

IBM MQ Console ve REST API , mqweb olarak bilinen bir WebSphere Liberty (Liberty) sunucusunda çalışan uygulamalardır. mqweb sunucusu başlatılan bir görev olarak çalışır. MQ Console , kuyruk yöneticilerini denetlemek için bir web tarayıcısının kullanılmasına olanak tanır. REST API , uygulamaların kuyruk yöneticisi yönetimini gerçekleştirmesi ve ileti sistemini gerçekleştirmesi için basit bir programlı arabirim sağlar.



## Kuruluş ve yapılandırma dosyaları

You need to install the IBM MQ for z/OS UNIX System Services Web Components feature, which will install the files needed to run the mqweb server in z/OS UNIX System Services (USS). Mqweb sunucusunu yapılandırabilmek ve yönetebilmek için USS ' nin bilgi sahibi olmanız gerekir.

USS ' deki IBM MQ dosyaları, mqweb sunucusunun doğru çalışması için gerekli olan çeşitli öznitelikler kümesiyle kurulur. If you need to copy the IBM MQ USS installation files, for example if you have installed IBM MQ on one system, and run IBM MQ on a different system, you should copy the IBM MQ ZFS created during the installation, and mount it read only at the destination. Dosyaların diğer yollarla kopyalanması, bazı dosya özniteliklerinin kaybolmasına neden olabilir.

Mqweb sunucusunu oluştururken, bir Liberty kullanıcı dizini için konuma karar vermeniz ve bu dizini oluşturmanız gerekir. Bu dizin, yapılandırma ve günlük dosyalarını içerir ve konum /var/mqm/mqwebile benzer bir şey olabilir.

## Using the MQ Console and REST API with queue managers at different levels

MQ Console ve REST API , yalnızca aynı Sürüm, Yayın ve Değişiklik (VRM) içinde çalışan kuyruk yöneticileriyle doğrudan etkileşime girebilirler. For example, the MQ Console and REST API shipped with IBM MQ 9.1.0 can interact only with local queue managers at IBM MQ 9.1.0, and the MQ Console and REST API shipped with IBM MQ 9.0.5 can interact only with local queue managers at IBM MQ 9.0.5.

REST API için, bir ağ geçidi kuyruk yöneticisi yapılandırarak, kuyruk yöneticilerini mqweb sunucusundan farklı bir sürümle yönetebilirsiniz. Ancak, ağ geçidi kuyruk yöneticisi olarak işlev görmek için, mqweb sunucusuyla aynı sürümde en az bir kuyruk yöneticisine gerek vardır. Daha fazla bilgi için bkz. [REST API komutunu kullanarak uzaktan yönetim](#).

## Geçiş

Yalnızca bir kuyruk yöneticiniz varsa, mqweb sunucusunu tek bir başlangıç görevi olarak çalıştırabilir ve kuyruk yöneticinizi yeni düzeye geçirdiğinizde kullandığı kitaplıkları değiştirebilirsiniz.

Birden çok kuyruk yöneticiniz varsa, geçiş sırasında farklı adlara sahip başlangıç görevlerini kullanarak mqweb sunucularını farklı sürümlerde başlatabilirsiniz. Bu adlar istediğiniz herhangi bir ad olabilir. For example, you can start an IBM MQ 9.1.0 mqweb server using a started task named MQWB0910, and an IBM MQ 9.0.5 mqweb server using a started task named MQWB0905.

Daha sonra kuyruk yöneticilerini bir sürümden sonraki bir sürüme geçirdiğinizde, kuyruk yöneticileri daha sonraki sürüm için mqweb sunucusunda kullanılabilir olur ve daha önceki sürüm için artık mqweb sunucusunda kullanılabilir durumda olmaz.

Tüm kuyruk yöneticilerini daha sonraki sürüme geçirdikten sonra, daha önceki sürüme ilişkin mqweb sunucusunu silebilirsiniz.

## HTTP kapıları

Mqweb sunucusu HTTP için en çok iki kapı kullanır:

- Bir tanesi HTTPS için, varsayılan değer 9443.
- Bir tane HTTP. HTTP varsayılan olarak etkinleştirilmez, ancak etkinleştirilirse varsayılan değeri 9080 'dir.

Varsayılan kapı değerleri kullanımdaysa, diğer kapıları ayırmanız gerekir. Birden çok IBM MQ sürümü için eşzamanlı olarak çalışan birden çok mqweb sunucunuz varsa, her sürüm için ayrı kapılar ayırmalısınız. Mqweb sunucusunun kullandığı kapıları ayara ilişkin ek bilgi için [HTTP ve HTTPS kapılarının yapılandırılması](#) başlıklı konuya bakın.

Bir kapıyla ilgili bilgileri görüntülemek için aşağıdaki TSO komutunu kullanabilirsiniz:

```
NETSTAT TCP tcip (PORT portNumber)
```

Burada *tcip* , TCP/IP adres alanının adıdır; *portNumber* , bilgileri görüntülemek için kullanılacak kapının numarasını belirtir.

## **Güvenlik-mqweb sunucusunu başlatma**

mqweb sunucusu kullanıcı kimliği belirli yetkilerine gerek duyar. Daha fazla bilgi için bakınız: [Authority required by the mqweb server started task user ID.](#)

## **Güvenlik- MQ Console ve REST API' in kullanılması**

MQ Console ve REST API' yi kullandığınızda, yapılandırılmış bir kayıt defterine dahil olan bir kullanıcı olarak kimlik doğrulaması gerekir. Bu kullanıcılara, kullanıcıların gerçekleştirebileceği işlemleri belirleyen belirli roller atanır. Örneğin, messaging REST API kullanmak için kullanıcıya MQWebUser rolü atanmış olmalıdır. MQ Console ve REST API için kullanılabilir roller ve bu rollerin verilmesine ilişkin erişim hakkında daha fazla bilgi için bkz. [MQ Console ve REST API üzerindeki roller.](#)

MQ Console ve REST API için güvenliğin yapılandırılmasıyla ilgili daha fazla bilgi için bkz. [MQ Console ve REST API güvenliği.](#)

## 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

**Aşağıdaki paragraf, İngiltere ya da bu tür hükümlerin yerel yasalarla uyumadığı diğer ülkelerde geçerli değildir:** INTERNATIONAL BUSINESS MACHINES CORPORATION BU YAYINI, HAK İHLALİ YAPILMAYACAĞINA DAİR GARANTİLERLE TİCARİLİK VEYA BELİRLİ BİR AMACA UYGUNLUK İÇİN ZİMNİ GARANTİLER DE DAHİL OLMAK VE FAKS BUNLARLA SINIRLI OLMAMAK ÜZERE AÇIK YA DA ZİMNİ HİÇBİR GARANTİ VERMEKSİZİN "OLDUĞU GİBİ" ESASIYLA SAĞLAMAKTADIR. 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 Highway 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 Lisansı Sözleşmesi ya da eşdeğer herhangi bir sözleşmenin kayıt ve koşulları altında 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 sınınamamıştır 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 sınınamamıştır. 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

Programlama arabirimi bilgileri (sağlandıysa), bu programla birlikte kullanılmak üzere uygulama yazılımları yaratmanıza yardımcı olmak üzere hazırlanmıştır.

Bu kitap, müşterinin WebSphere MQ hizmetlerini edinmek üzere program yazmasına olanak tanıyan, amaçlanan programlama arabirimlerine ilişkin bilgiler içerir.

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, değişiklik söz konusu olduğunda kullanılır.

## 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 listesini Web üzerinde "Telif hakkı ve ticari marka bilgileri" [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) adresinde bulabilirsiniz. Diğer ürün ve hizmet adları IBM'in veya diğer şirketlerin ticari markaları olabilir.

Microsoft ve Windows, Microsoft Corporation'ın ABD ve/veya diğer ülkelerdeki ticari markalarıdır.

UNIX, The Open Group şirketinin ABD ve diğer ülkelerdeki tescilli ticari markasıdır.

Linux, Linus Torvalds'ın ABD ve/ya da diđer ÷lkelerdeki tescilli ticari markasıdır.

Bu ÷r÷n, Eclipse Project (<http://www.eclipse.org/>) tarafından geliřtirilen yazılımları ierir.

Java ve Java tabanlı t÷m markalar ve logolar, Oracle firmasının ve/ya da iřtiraklerinin markaları ya da tescilli markalarıdır.







Parça numarası:

(1P) P/N: