

9.3

IBM MQ ' nun denetlenmesi

IBM

Not

Bu bilgileri ve desteklediđi ürünü kullanmadan önce, [“Özel notlar” sayfa 585](#) bölümündeki bilgileri okuyun.

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

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

© Copyright International Business Machines Corporation 2007, 2024.

İçindekiler

Yönetme.....	7
IBM MQ kuyruk yöneticilerini ve ilişkili kaynakları yönetme yolları.....	8
IBM MQ for Multiplatforms 'yi denetim komutlarını kullanarak yönetme.....	10
IBM MQ 'nin MQSC komutlarını kullanarak denetlenmesi.....	12
MQSC komutu sözdizimi.....	12
MQSC giriş dosyası sözdizimi.....	15
runmqsc altında MQSC komutlarının etkileşimli olarak çalıştırılması.....	17
runmqsc altındaki metin dosyalarından MQSC komutlarının çalıştırılması.....	22
Başlatma sırasında bir MQSC komut dosyasından otomatik yapılandırma.....	23
IBM MQ yönetiminin PCF komutlarıyla otomatikleştirilmesi.....	25
IBM MQ Programlanabilir Komut Biçimlerine Giriş.....	25
PCF 'lerin kullanımını basitleştirmek için MQAI' nin kullanılması.....	37
REST API kullanarak yönetim.....	72
administrative REST API ile çalışmaya başlama.....	73
REST API kullanılarak uzaktan yönetim.....	78
REST API zaman damgaları.....	82
REST API Hata işleme.....	82
REST API Keşif.....	84
REST API ulusal dil desteği.....	86
REST API sürümler.....	87
IBM MQ Console kullanarak yönetim.....	89
IBM MQ Console ile çalışmaya başlama.....	89
Yalnızca una unu gelen unun unun gelen una unun una gelen gelen IBM MQ Console -CD.....	90
IBM MQ Console -Yalnızca LTS için hızlı tur.....	116
IBM MQ Console ayarlar.....	138
IBM MQ Explorer kullanarak yönetim.....	138
IBM MQ Explorer ile yapabileceklerin.....	139
IBM MQ Explorer 'in ayarlanması.....	140
IBM MQ Görev Çubuğu uygulamasını kullanma (yalnızcaWindows).....	146
IBM MQ uyarı izleyici uygulaması (yalnızca Windows).....	146
Yerel IBM MQ nesneleriyle çalışma.....	146
Kuyruk yöneticileriyle çalışma.....	147
MQI kanalları durduruluyor.....	157
Yerel kuyruklarla çalışma.....	158
Uzak kuyruklarla çalışılması.....	167
Diğer ad kuyruklarıyla çalışılması.....	170
Model kuyruklarıyla çalışma.....	172
Gönderilmeyen ileti kuyruklarıyla çalışılması.....	173
Yönetimle ilgili konularla çalışma.....	191
Aboneliklerle çalışma.....	195
Hizmetlerle çalışma.....	199
Tetikleme için nesnelere yönetme.....	206
İki sistem arasında dmpmqmsg yardımcı programının kullanılması.....	208
Uzak IBM MQ nesneleriyle çalışma.....	212
Uzak denetim için kuyruk yöneticilerinin yapılandırılması.....	213
Uzak yönetim için komut sunucusunun yönetilmesi.....	217
Uzak bir kuyruk yöneticisinde MQSC komutları verilmesi.....	218
Kodlanmış karakter takımları arasında veri dönüştürme.....	220
YönetmeManaged File Transfer.....	224
MFT aracısını başlatma.....	226
MFT araçlarını listeleme.....	231
MFT aracısının durdurulması.....	232

Yeni dosya aktarımı başlatılması.....	233
Zamanlanmış dosya aktarımı yaratılması.....	237
Bekleyen dosya aktarımlarıyla çalışma.....	238
Dosya aktarımını tetikleme.....	239
Devam eden dosya aktarımlarını izleme.....	240
Aktarma Günlüğündeki Dosya Aktarımlarının Durumunun Görüntülenmesi.....	242
MFT kaynaklarının izlenmesi.....	244
Dosya aktarma şablonlarıyla çalışılması.....	274
Dosyalardan iletilere veri aktarılması.....	277
İletilerden dosyalara veri aktarılması.....	291
Protokol köprüsü.....	302
Connect:Direct köprüsü.....	324
IBM Integration Bus içinden MFT ile çalışma.....	340
MFT kurtarma ve yeniden başlatma.....	340
Durdurulmuş aktarımların kurtarılması için zamanaşımı ayarlanması.....	341
YönetmeMQ Telemetry.....	346
Linux ve AIX üzerinde telemetri için bir kuyruk yöneticisinin yapılandırılması.....	347
Windows üzerinde telemetri için bir kuyruk yöneticisinin yapılandırılması.....	348
Dağıtılmış kuyruğa alma özelliğinin MQTT istemcilerine ileti gönderecek şekilde yapılandırılması.....	350
MQTT müşteri kimliği, yetkilendirme ve kimlik doğrulaması.....	352
TLS kullanarak telemetri kanalı kimlik doğrulaması.....	359
Telemetri kanallarında yayın gizliliği.....	362
MQTT Java istemcilerinin ve telemetri kanallarının TLS yapılandırması.....	363
Telemetri kanalı JAAS yapılandırması.....	368
AMQP istemcisinin yönetilmesi.....	370
AMQP istemcileri tarafından kullanılan IBM MQ nesnelerini görüntüleme.....	370
AMQP istemci tanıtıcısı, yetkilendirme ve kimlik doğrulaması.....	371
Kanallarda yayın gizliliği.....	373
TLS ile AMQP istemcilerini yapılandırma.....	374
AMQP istemcilerinin kuyruk yöneticisiyle bağlantısı kesiliyor.....	375
Çoklu yayın yönetimi.....	375
Çoklu yayın ile çalışmaya başlama.....	375
IBM MQ Çok hedefli konu topolojisi.....	376
Çok hedefli iletilerin boyutunun denetlenmesi.....	377
Çoklu yayın ileti sistemi için veri dönüştürmeyi etkinleştirme.....	379
Çok hedefli uygulama izleme.....	380
Çok hedefli ileti güvenilirliği.....	380
Gelişmiş çoklu yayın görevleri.....	381
YönetmeIBM MQ for IBM i.....	384
CL komutlarını kullanarak IBM MQ for IBM i yönetimi.....	384
IBM MQ for IBM i ' i yönetmenin alternatif yolları.....	398
IBM i için iş yönetimi.....	403
IBM i üzerinde kullanılabilirlik, yedekleme, kurtarma ve yeniden başlatma.....	410
IBM MQ for IBM i ' i susturma.....	453
YönetmeIBM MQ for z/OS.....	456
z/OS üzerinde kuyruk yöneticisi komutları verilmesi.....	457
z/OS üzerindeki işlemleri ve denetim panolarını kullanma.....	471
IBM MQ for z/OS yardımcı programlarının kullanılması.....	479
z/OS üzerinde Command Facility olanağının kullanılması.....	482
z/OS üzerinde IBM MQ nesneleriyle çalışma.....	483
Sistemin birden çok küme iletim kuyruğu kullanılarak uygulanması.....	485
IBM MQ for z/OS yönetimi için programlar yazılıyor.....	488
z/OS üzerinde IBM MQ kaynaklarının yönetilmesi.....	499
z/OS üzerinde kurtarma ve yeniden başlatma.....	539
IBM MQ ve IMS.....	560
z/OS üzerinde Advanced Message Security işletim.....	573
YönetmeIBM MQ Internet Pass-Thru.....	573

MQIPT başlatılıyor ve durduruluyor.....	574
MQIPT komutunu komut satırını kullanarak yönetme.....	576
Yedeklerin yapılması.....	581
Performans ayarı.....	582
Özel notlar.....	585
Programlama arabirimi bilgileri.....	586
Ticari Markalar.....	586

IBM MQ kuyruk yöneticilerinizi ve ilişkili kaynaklarınızı denetlemek için, bu kaynakları etkinleştirmek ve yönetmek için kullanabileceğiniz görevler kümesinden tercih ettiğiniz yöntemi seçin.

Bu görev hakkında

IBM MQ nesnelere yerel olarak ya da uzaktan denetleyebilirsiniz:

Yerel yönetim

Yerel yönetim, yerel sisteminizde tanımladığınız kuyruk yöneticilerine ilişkin denetim görevlerinin gerçekleştirilmesi anlamına gelir. Diğer sistemlere (örneğin, TCP/IP uçbirim öykünme programı **telnet** aracılığıyla) erişebilir ve orada denetim yürütebilirsiniz. IBM MQ içinde, iletişim, işletim sistemi tarafından yönetildiğinden, herhangi bir kanal yer almadığı için bunu yerel yönetim olarak kabul edebilirsiniz.

Daha fazla bilgi için bkz. [“Yerel IBM MQ nesnelere çalışma” sayfa 146.](#)

Uzak yönetim


IBM MQ, uzaktan yönetim aracılığıyla tek bir iletişim noktasından yönetimi destekler. Uzak yönetim, başka bir sistemde işlenen ve IBM MQ Explorer için de geçerli olan yerel sisteminizden komut vermenizi sağlar. Örneğin, uzak bir kuyruk yöneticisindeki bir kuyruk tanımlamasını değiştirmek için uzak bir komut verebilirsiniz. Bu sistemde oturum açmanıza gerek yoktur, ancak uygun kanalların tanımlanmış olması gerekir. Hedef sistemdeki kuyruk yöneticisi ve komut sunucusu çalışıyor olmalıdır.

Bazı komutlar, kuyruk yöneticilerinin yaratılması ya da başlatılması ve komut sunucularının başlatılması gibi, bu yolla verilemez. Bu tip bir görevi gerçekleştirmek için, uzak sistemde oturum açıp komutları oradan vermeniz ya da komutları sizin için yayınlayabilecek bir işlem yaratmanız gerekir. Bu kısıtlama, IBM MQ Explorer için de geçerlidir.




Daha fazla bilgi için bkz. [“Uzak IBM MQ nesnelere çalışma” sayfa 212.](#)


IBM MQ' da kuyruk yöneticilerinizi ve ilgili kaynaklarını yaratmak ve denetlemek için kullanabileceğiniz birçok farklı yöntem vardır. Bu yöntemler, komut satırı arabirimlerini, grafik kullanıcı arabirimlerini ve bir denetim API 'sini içerir.

Platformunuza bağlı olarak IBM MQ ürününü yönetmek için kullanabileceğiniz farklı komut kümeleri vardır:

- [“IBM MQ denetim komutları” sayfa 8](#)
- [“IBM MQ Komut Dosyası \(MQSC\) komutları” sayfa 8](#)
- [“Programlanabilir Komut Biçimleri \(PCF\)” sayfa 8](#)
- [administrative REST API](#)
-  [“IBM i Denetim Dili \(CL\)” sayfa 9](#)

IBM MQ nesnelere yaratmak ve yönetmek için aşağıdaki seçenekler de vardır:

-   [“IBM MQ Explorer” sayfa 9](#)
- [“IBM MQ Console” sayfa 9](#)
-  [“Microsoft Cluster Service \(MSCS\)” sayfa 10](#)

 [IBM MQ for z/OS üzerindeki yönetim arabirimleri ve seçenekleriyle ilgili bilgi için bkz. “YönetmeIBM MQ for z/OS” sayfa 456.](#)

PCF komutlarını kullanarak hem yerel hem de uzak kuyruk yöneticileri için bazı denetim ve izleme görevlerini otomatikleştirebilirsiniz. Bu komutlar, bazı platformlarda IBM MQ Yönetim Arabirimi (MQAI) kullanılarak da basitleştirilebilir. Yönetim görevlerini otomatikleştirme hakkında daha fazla bilgi için bkz. [“IBM MQ yönetiminin PCF komutlarıyla otomatikleştirilmesi” sayfa 25.](#)

İlgili kavramlar

[IBM MQ teknik genel bakış](#)

İlgili görevler

[Planlama](#)

[Yapılandırılıyor](#)

İlgili başvurular

[Komut kümeleri karşılaştırması](#)

IBM MQ kuyruk yöneticilerini ve ilişkili kaynakları yönetme yolları

IBM MQ kuyruk yöneticilerini ve ilişkili kaynakları IBM MQ denetim komutlarını, IBM MQ Script Commands (MQSC), Programlanabilir Command Formats (PCF), administrative REST API, IBM MQ Console ve IBM MQ Explorer komutlarını kullanarak yönetebilirsiniz. IBM i için IBM i Denetim Dili 'ni ve Windows için Microsoft Küme Hizmeti 'ni (MSCS) kullanabilirsiniz.

IBM MQ denetim komutları

Multi

Denetim komutları, bir dizi IBM MQ yönetim görevini gerçekleştirmek için bir yol sağlar. AIX, Linux®, and Windows için, bu komutları sistem komut satırında yayınlayın. IBM i için, bu komutları bir Qshell içinde yayınlayın. Bkz. [“IBM MQ for Multiplatforms 'yi denetim komutlarını kullanarak yönetme” sayfa 10.](#)

IBM MQ Komut Dosyası (MQSC) komutları

Kuyruk yöneticisinin kendisi, kuyruklar, süreç tanımlamaları, ad alanları, kanallar, istemci bağlantısı kanalları, dinleyiciler, hizmetler ve kimlik doğrulama bilgileri nesnelere de içinde olmak üzere kuyruk yöneticisi nesnelere yönetmek için MQSC komutlarını kullanın.

ALW

AIX, Linux, and Windows' da bir **runmqsc** komut istemi açın ve bu komut isteminden yerel ya da uzak bir kuyruk yöneticisine MQSC komutları verin. Bunu etkileşimli olarak yapabilir ya da bir ASCII metin dosyasından bir dizi komut çalıştırabilirsiniz. Daha fazla bilgi için bkz. [“runmqsc altında MQSC komutlarının etkileşimli olarak çalıştırılması” sayfa 17](#) ve [“runmqsc altındaki metin dosyalarından MQSC komutlarının çalıştırılması” sayfa 22.](#)

IBM i

IBM i' ta, bir Komut dosyasında bir komut listesi oluşturduktan sonra **STRMQMQSC** komutunu kullanarak dosyayı çalıştırın. Daha fazla bilgi için bkz [“IBM i üzerinde MQSC komutlarını kullanarak yönetme” sayfa 399.](#)

z/OS

z/OS üzerinde, komuta bağlı olarak MQSC komutları birçok kaynaktan verilebilir. Daha fazla bilgi için bkz [“IBM MQ for z/OS üzerinde MQSC ve PCF komutlarını yayınlatabileceğiniz kaynaklar” sayfa 457.](#)

Programlanabilir Komut Biçimleri (PCF)

Programlanabilir Komut Biçimleri (PCF), bir program ile bir ağdaki herhangi bir kuyruk yöneticisi (PCF ' leri destekleyen) arasında değiş tokuş edilebilecek komut ve yanıt iletilerini tanımlar. IBM MQ nesnelere denetimi için bir sistem yönetimi uygulama programındaki PCF komutlarını kullanabilirsiniz: kimlik doğrulama bilgileri nesnelere, kanallar, kanal dinleyicileri, ad listesi, süreç tanımlamaları, kuyruk yöneticileri, kuyruklar, hizmetler ve depolama sınıfları. Uygulama, yerel kuyruk yöneticisini kullanarak herhangi bir kuyruk yöneticisiyle, yerel ya da uzak kuyruk yöneticisiyle komut ve yanıt bilgilerini iletmek için ağdaki tek bir noktadan çalışabilir.

PCF ' lere ilişkin ek bilgi için bkz. [“IBM MQ Programlanabilir Komut Biçimlerine Giriş” sayfa 25.](#)

Komutlara ve yanıtlara ilişkin PCF ' lerin ve yapıların tanımı için bkz. [Programlanabilir komut biçimleri başvurusu.](#)

administrative REST API

administrative REST API , IBM MQürünü yönetmek için kullanabileceğiniz bir RESTful arabirimi sağlar. administrative REST APIkullandığınızda, IBM MQ nesnesini gösteren bir URL ' de HTTP yöntemini çağırırsınız. Örneğin, aşağıdaki URLüzerinde HTTP yöntemini kullanarak IBM MQ kuruluşlarıyla ilgili bilgi isteyebilirsiniz:

```
https://localhost:9443/ibmmq/rest/v1/admin/installation
```

administrative REST API uygulamasını bir programlama dilinin HTTP/REST uygulamasıyla ya da cURLgibi araçları ya da bir REST istemci tarayıcısı eklentisiyle kullanabilirsiniz.

Daha fazla bilgi için bkz. [administrative REST API](#)

IBM MQ Console

Bir web tarayıcısından IBM MQ ürünü yönetmek için IBM MQ Console komutunu kullanabilirsiniz.

Daha fazla bilgi için bkz “[IBM MQ Console kullanarak yönetim](#)” sayfa 89.


IBM MQ Explorer



IBM MQ Explorerkomutunu kullanarak aşağıdaki işlemleri gerçekleştirebilirsiniz:



- Kuyruk yöneticileri, kuyruklar, süreç tanımlamaları, ad alanları, kanallar, istemci bağlantısı kanalları, dinleyiciler, hizmetler ve kümeler gibi çeşitli kaynakları tanımlayın ve denetleyin.
- Yerel bir kuyruk yöneticisini ve ilişkili işlemlerini başlatır ya da durdurur.
- İş istasyonunuzdaki ya da diğer iş istasyonlarındaki kuyruk yöneticilerini ve ilişkili nesnelere görüntüleyin.
- Kuyruk yöneticilerinin, kümelerin ve kanalların durumunu denetleyin.
- Kuyruk durumundan hangi uygulamaların, kullanıcıların ya da kanalların belirli bir kuyruğun açık olduğunu görmek için denetleyin.

Windows ve Linux for x86-64 sistemlerinde, sistem menüsünü ya da MQExplorer yürütülür dosyakullanarak IBM MQ Explorer ' ı başlatabilirsiniz.

 Linux'ta IBM MQ Explorer ' ı başarıyla başlatmak için ana dizininize bir dosya yazabilmeniz ve ana dizinin var olması gerekir.

Daha fazla bilgi için bkz “[IBM MQ Explorer kullanarak yönetim](#)” sayfa 138.

Uzak kuyruk yöneticilerini z/OSda dahil olmak üzere diğer platformlarda denetlemek için IBM MQ Explorer komutunu kullanabilirsiniz.

  IBM MQ 9.3.0içinden IBM MQ Explorer , IBM MQ kuruluş paketinden kaldırılmıştır. Ayrı bir karşidan yükleme olarak kullanılabilir ve Fix Centraladresinden sağlanan bağımsız IBM MQ Explorer karşidan yüklemesinden kurulabilir. Daha fazla bilgi için, bkz. [IBM MQ Explorer ürününü Linux ve Windows üzerinde bağımsız bir uygulama olarak kurma ve kaldırma](#).

IBM i Denetim Dili (CL)



IBM MQ for IBM i' e yönetim komutları vermenin tercih edilen yolu budur. Komutlar, komut satırında ya da bir CL programı yazılarak verilebilir. Bu komutlar PCF komutlarına benzer işlevleri gerçekleştirir, ancak biçim farklıdır. CL komutları özel olarak sunucular için tasarlanmıştır ve CL yanıtları insan tarafından okunabilir iken, PCF komutları platformdan bağımsızdır ve hem komut hem de yanıt biçimleri program kullanımı için tasarlanmıştır.

IBM i Denetim Dili (CL) ile ilgili tüm ayrıntılar için bkz. [“CL komutlarını kullanarak IBM MQ for IBM i yönetimi” sayfa 384 ve IBM MQ for IBM i CL komutları.](#)

Microsoft Cluster Service (MSCS)

Windows

Microsoft Küme Hizmeti (MSCS), sunucuları bir *küme*ye bağlamanızı sağlayarak verilerin ve uygulamaların kullanılabilirliğini artırmanızı ve sistemi daha kolay yönetmenizi sağlar. MSCS, sunucu ya da uygulama hatalarını otomatik olarak algılayabilir ve kurtarabilir.

MSCS anlamda kümeleri IBM MQ kümeleriyle karıştırmamak önemlidir. Ayrım aşağıdaki gibidir:

IBM MQ Kümeler

Bunlar, bir ya da daha fazla bilgisayardaki iki ya da daha fazla kuyruk yöneticisinden oluşan gruplardır ve otomatik ara bağlantı sağlar ve yük dengeleme ve yedeklilik için kuyruklar arasında paylaşılmasına izin verir.

MSCS kümeleri

Bunlar, birbirine bağlı ve biri arızalanırsa, MSCS ' nin *hata durumunda yedek sisteme geçiş* gerçekleştirecek ve uygulamaların durum verilerini hatalı bilgisayardan kümedeki başka bir bilgisayara aktararak yeniden başlatacağı şekilde yapılandırılmış bilgisayar gruplarıdır.

Microsoft Cluster Service (MSCS) olanağının desteklenmesi , IBM MQ for Windows sisteminizi MSCS kullanacak şekilde nasıl yapılandıracağınıza ilişkin ayrıntılı bilgi sağlar.

İlgili görevler

[“IBM MQ ' nin MQSC komutlarını kullanarak denetlenmesi” sayfa 12](#)

Kuyruk yöneticisinin kendisi, kuyruklar, süreç tanımlamaları, kanallar, istemci bağlantısı kanalları, dinleyiciler, hizmetler, ad alanları, kümeler ve kimlik doğrulama bilgileri nesnelere de içinde olmak üzere kuyruk yöneticisi nesnelere yönetmek için MQSC komutlarını kullanabilirsiniz. MQSC komutları tüm altyapılarda kullanılabilir.

İlgili başvurular

[Yönetim başvurusu](#)

Multi

IBM MQ for Multiplatforms ' yi denetim komutlarını kullanarak yönetme

Denetim komutları, bir dizi IBM MQ yönetim görevini gerçekleştirmek için bir yol sağlar. AIX, Linux ve Windows için, bu komutları sistem komut satırında yayınlayın. IBM için, bu komutları bir Qshell içinde yayınlayın.

Başlamadan önce

Bir kuyruk yöneticisinde çalışan denetim komutlarını kullanırken, çalıştığınız kuyruk yöneticisiyle ilişkili kuruluştaki komutu kullanmanız gerekir.

CHCKLOCL (REQUIRED) ile bağlantı doğrulamasını kullanmak üzere yapılandırılmış bir kuyruk yöneticisinde çalışan denetim komutları kullanılırken, bağlantı hatası gözlenir.

- Denetim komutu buna izin veriyorsa, bir kullanıcı kimliği ve parola belirtin.
- Var olan denetim komutlarının MQSC eşdeğerlerini kullanın.
- Bağlantı kuramayan denetim komutlarının çalıştırılması gerekirken -ns seçeneğini kullanarak kuyruk yöneticisini başlatın.

Not: Farklı platformlar, farklı bir sırayla girilen komut bağımsız değişkenlerini kabul edebilir. Bu, özellikle Linux üzerinde çalışan komutların diğer platformlarda çalışmayabileceği anlamına gelir. Bu nedenle, sözdizimi şemalarında belirtildiği gibi bağımsız değişkenleri her zaman girmeniz gerekir.

Denetim komutlarının tam listesi için bkz. [IBM MQ denetim komutları başvurusu.](#)

Yordam

Linux AIX

AIX and Linux sistemlerinde denetim komutlarını kullanın.

IBM MQ for AIX or Linux sistemlerinde, denetim komutlarını bir kabuk penceresine girmeniz gerekir.

Denetim komutları yayınlamak istiyorsanız, kullanıcı kimliğiniz çoğu denetim komutu için mqm grubunun bir üyesi olmalıdır. Bu konuda daha fazla bilgi için bkz. [AIX, Linux, and Windows üzerinde IBM MQ yönetimi yetkisi](#). Ayrıca, ortama özgü bilgilere de dikkat edin. Kuruluşunuzun kullandığı platform veya platformlar için.

UNIX and Linux ortamlarında, komut adının kendisi, işaretler ve bağımsız değişkenler de içinde olmak üzere denetim komutları büyük ve küçük harfe duyarlıdır. Örneğin, komutta:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE jupiter.queue.manager
```

- Komut adı, CRTMQM değil, crtmqm olmalıdır.
- İşaret, -U değil, -u olmalıdır.
- Gönderilmeyen iletiler kuyruğuna SYSTEM.DEAD.LETTER.QUEUE adı verilir.
- Bağımsız değişken, JUPITER.queue.manager bağımsız değişkeninden farklı olan jupiter.queue.manager olarak belirtilir.

Komutları tam olarak örneklerde gördüğünüz gibi yazmaya özen gösteriniz.

Windows

Windows sistemlerinde denetim komutlarını kullanın.

IBM MQ for Windows içinde, bir komut istemine denetim komutları girilir.

Denetim komutları yayınlamak istiyorsanız, kullanıcı kimliğiniz çoğu denetim komutu için mqm grubunun bir üyesi olmalıdır. Bu konuda daha fazla bilgi için bkz. [AIX, Linux, and Windows üzerinde IBM MQ yönetimi yetkisi](#). Ayrıca, ortama özgü bilgilere de dikkat edin. Kuruluşunuzun kullandığı platform veya platformlar için.

Denetim komutları ve işaretleri büyük ve küçük harfe duyarlı değildir, ancak kuyruk adları ve kuyruk yöneticisi adları gibi bu komutlara ilişkin bağımsız değişkenler büyük ve küçük harfe duyarlıdır.

Örneğin, komutta:


```
crtmqm /u SYSTEM.DEAD.LETTER.QUEUE jupiter.queue.manager
```

- Komut adı büyük harfli ya da küçük harfli olarak ya da ikisinin bir karışımı olarak girilebilir. Bunların tümü geçerlidir: CRTMQM, CRTmqm ve crtmqm.
- İşaret -u, -U, /u ya da /U olarak girilebilir.
- SYSTEM.DEAD.LETTER.QUEUE ve jupiter.queue.manager tam olarak gösterildiği gibi girilmelidir.

IBM i

IBM i sistemlerinde denetim komutlarını kullanın.

IBM MQ for IBM işletim sistemi üzerinde, denetim komutlarını Qshell ortamından çalıştırır. Qshell olanağını kullanmak için IBM i komut satırına STRQSH yazın. F3 tuşuna basarak istediğiniz zaman çıkabilir ve komut satırına geri dönebilirsiniz.

IBM üzerinde az sayıda denetim komutu desteklenmez. Örneğin, IBM i sisteminde birden çok IBM MQ kopyası bulunamayacağından çoklu kuruluş komutları desteklenmez. IBM i üzerinde desteklenmeyen komutlar, [IBM MQ denetim komutları başvurusu](#) içinde  simgesi işaretlenir.

İlgili başvurular

[IBM MQ denetim komutları başvurusu](#)

IBM MQ ' nin MQSC komutlarını kullanarak denetlenmesi

Kuyruk yöneticisinin kendisi, kuyruklar, süreç tanımlamaları, kanallar, istemci bağlantısı kanalları, dinleyiciler, hizmetler, ad alanları, kümeler ve kimlik doğrulama bilgileri nesnelere de içinde olmak üzere kuyruk yöneticisi nesnelere yönetmek için MQSC komutlarını kullanabilirsiniz. MQSC komutları tüm altyapılarda kullanılabilir.

Bu görev hakkında

Kullanılabilecek MQSC komutları, [MQSC komutları başvurusunda](#) ayrıntılı olarak açıklanmıştır.

MQSC komutlarını verme şekliniz altyapınıza bağlıdır:

- **ALW** AIX, Linux, and Windows sistemlerinde, **runmqsc** komut isteminden bir kuyruk yöneticisine MQSC komutları yayınlayın. Bu komut istemini çeşitli şekillerde kullanabilirsiniz:
 - Etkileşimli olarak, bir klavyeden MQSC komutları yayınlanıyor. Bkz. [“runmqsc altında MQSC komutlarının etkileşimli olarak çalıştırılması”](#) sayfa 17.
 - Bir ASCII metin dosyasından MQSC komutları yayınlanıyor. Bkz. [“runmqsc altındaki metin dosyalarından MQSC komutlarının çalıştırılması”](#) sayfa 22.
 - Uzak bir kuyruk yöneticisinde MQSC komutları yayınlanıyor. Bkz. [“Uzak bir kuyruk yöneticisinde MQSC komutları verilmesi”](#) sayfa 218.
- **IBM i** IBM i' ta, bir Komut dosyasında bir komut listesi oluşturduktan sonra **STRMQMMQSC** komutunu kullanarak dosyayı çalıştırın. Daha fazla bilgi için bkz [“IBM i üzerinde MQSC komutlarını kullanarak yönetme”](#) sayfa 399.
- **z/OS** z/OS üzerinde, komuta bağlı olarak MQSC komutları birçok kaynaktan verilebilir. Daha fazla bilgi için bkz [“IBM MQ for z/OS üzerinde MQSC ve PCF komutlarını yayınlatabileceğiniz kaynaklar”](#) sayfa 457.

Yordam

- [“MQSC komutu sözdizimi”](#) sayfa 12
- [“MQSC: Özel karakterler ve soysal değerler”](#) sayfa 14
- [“runmqsc altında MQSC komutlarının etkileşimli olarak çalıştırılması”](#) sayfa 17
- [“runmqsc altındaki metin dosyalarından MQSC komutlarının çalıştırılması”](#) sayfa 22
- [“Başlatma sırasında bir MQSC komut dosyasından otomatik yapılandırma”](#) sayfa 23

İlgili görevler

[MQSC komutlarıyla ilgili sorunların çözülmesi](#)

İlgili başvurular

[runmqsc \(MQSC komutlarını çalıştır\)](#)

MQSC komutu sözdizimi

Kuyruk yöneticisi nesnelere yönetmek için MQSC komutlarını kullanabilirsiniz. MQSC komutları tüm altyapılarda kullanılabilir. Komut sözdiziminin bazı öğeleri altyapıya özgüdür.

Değiştirgelerin sırası

Her komut bir birincil parametre (bir fiil) ile başlar ve bunu ikincil bir parametre (bir isim) izler. Bunu, çoğu komutta bulunan nesnenin adı ya da soysal adı (parantez içinde) izler. Bunu takiben, parametreler genellikle herhangi bir sırada oluşabilir; bir parametrenin karşılık gelen bir değeri varsa, değer, ilgili olduğu parametreden hemen sonra olmalıdır.

Not: **z/OS** z/OS üzerinde, ikincil parametrenin ikinci olması gerekmez.

Boşluklar ve virgüller

Anahtar sözcükler, parantezler ve değerler, boşluk ve virgüllerle ayrılabilir. Sözdizimi çizgelerinde gösterilen bir virgül her zaman bir ya da daha çok boşluklarla değiştirilebilir. Her değiştirmeden hemen önce (birincil değiştirmeden sonra) en az bir boşluk olmalıdır z/OS dışında .

Komutun başında ya da sonunda ve parametreler, noktalama işaretleri ve değerler arasında herhangi bir sayıda boşluk oluşabilir. Örneğin, aşağıdaki komut geçerlidir:

```
ALTER QLOCAL ('Account' ) TRIGDPTH ( 1)
```

Bir çift tırnak işareti içindeki boşluklar önemlidir.

Boşlukların izin verildiği herhangi bir yerde ek virgüller görünebilir ve bunlar boşluklarmış gibi işlenir (elbette tırnak işareti içine alınmış dizgiler dışında).

Yinelenen parametreler

Yinelenen parametrelere izin verilmez. REPLACE NOREPLACE içinde olduğu gibi, bir değiştirgenin "NO" sürümüyle yinelenmesine de izin verilmez.

Dizeler ve tek tırnak işaretleri

Boşluklar, küçük harfler ya da özel karakterler içeren dizgiler, aşağıdakilerden biri doğru değilse, tek tırnak içine alınmalıdır:

- Özel karakterler, aşağıdaki karakterlerden biri ya da daha fazladır:
 - Nokta (.)
 - Eğik Çizgi (/)
 - Alt çizgi (_)
 - Yüzde işareti (%)
- **z/OS** Komut, IBM MQ for z/OS işlemleri ve denetim panolarından yayınlanır.
- Dizgi, yıldız işaretiyle biten soysal bir değerdir. (IBM i üzerinde bunlar tek tırnak içine alınmalıdır)
- Dizgi tek bir yıldız işaretidir; örneğin, TRACE (*) (IBM i üzerinde bunlar tek tırnak içine alınmalıdır)
- Dizgi, iki nokta imi içeren bir aralık belirtimidir; örneğin, CLASS (01:03)

Dizginin kendisi tek tırnak işareti içeriyorsa, tek tırnak işareti iki tek tırnak işareti ile gösterilir.

Multi Çoklu platformlar üzerinde, karakter içermeyen bir dizgi (yani, aralarında boşluk olmayan iki tek tırnak işareti), (") ile aynı şekilde yorumlanan, tek tırnak işareti içine alınmış bir boşluk olarak yorumlanır. Kullanılmakta olan öznitelik aşağıdaki özniteliklerden biriye, boşluk olmayan iki tek tırnak işareti sıfır uzunluklu dizgi olarak yorumlandığında bunun özel durumu olur:

- TOPICSTR
- SUB
- USERDATA
- SELECTOR

z/OS z/OS üzerinde, tek tırnak işareti içine alınmış bir boşluk istiyorsanız, bu alanı (") gibi girmeniz gerekir. Karakter içermeyen bir dizgi ("), giriş () ile aynıdır.

MQCHARV tiplerine (SELECTOR, alt kullanıcı verileri gibi) dayalı dizgi özniteliklerindeki sondaki boşluklar önemli olarak değerlendirilir; bu, 'abc ' ' in 'abc ' değerine eşit olmadığı anlamına gelir.

Boş ayrıçlar

Aralarında önemli bir bilgi olmayan bir kapama ayrıcının ardından gelen bir açma ayrıcı, özellikle belirtildiği durumlar dışında geçerli değildir. Örneğin, aşağıdaki dizgi geçerli değildir:

```
NAME ( )
```

Küçük harf ve büyük harf

Anahtar sözcükler büyük ve küçük harfe duyarlı değildir: ALTER, alter ve ALTER işlemleri kabul edilebilir.

Tırnak işaretleri içinde olmayan her şey büyük harfe katlanır.

Eşanlamlılar

Eşanlamlılar bazı parametreler için tanımlanır. Örneğin, DEF her zaman DEFINE eşanlamlısıdır, bu nedenle DEF QLOCAL geçerlidir. Eşanlamlılar yalnızca minimum dizgiler değildir; DEF1, DEFINE için geçerli bir eşanlamlı değildir.

Not: DELETE parametresi için eşanlamlı yok. Bu, DEFINE eşanlamlısı DEF kullanılırken nesnelere yanlışlıkla silinmesini önler.

Özel karakterler

MQSC komutları, belirli anlamları olan bazı özel karakterleri kullanır. Bu özel karakterler ve bunların nasıl kullanılacağı hakkında daha fazla bilgi için bkz. [“MQSC: Özel karakterler ve soysal değerler” sayfa 14.](#)

İlgili görevler

[MQSC komutlarıyla ilgili sorunların çözülmesi](#)

İlgili başvurular

[runmqsc \(MQSC komutlarını çalıştır\)](#)

MQSC: Özel karakterler ve soysal değerler

Bazı karakterler; örneğin, ters eğik çizgi (\) ve çift tırnak (") karakterlerin MQSC komutlarıyla kullanıldığında özel anlamları vardır. Değiştirgelerle kullanılacak bazı özel karakterlerin soysal değerleri olabilir, ancak doğru belirtilmesi gerekir.

Ters eğik çizgi (\) ve çift tırnak (") içine alın \, yani, metnimize \ ya da " yazarsanız \\ ya da \" girin.


Bir parametrenin soysal değeri varsa, sonuna yıldız işareti (*) girilir; örneğin, ABC*. Soysal değer, bununla başlayan tüm değerler anlamına gelir; bu nedenle ABC* , ABC ile başlayan tüm değerler anlamına gelir. Değerde tırnak işareti gerektiren karakterler kullanılırsa, yıldız imi tırnak işareti içine alınmalıdır; bu nedenle, ' abc* '. Yıldız imi, değerdeki son ya da tek karakter olmalıdır.

Genel değerlerde soru işareti (?) ve iki nokta (:) kullanılamaz.

Bir alanda (örneğin, açıklamanın bir parçası olarak) bu özel karakterlerden herhangi birini kullanmanız gerekirse, dizginin tamamını tek tırnak işareti içine almanız gerekir.

Çizelge 1. Özel anlamları olan karakterlerin açıklamaları	
Karakter	Açıklama
	Ayrırcı olarak boşluk kullanılır. Birden çok boşluk, kesme işareti (') içine alınmış dizgiler dışında, tek bir boşlukla eşdeğerdir. MQCHARV tiplerine dayalı dizgi özniteliklerindeki sondaki boşluklar önemli olarak kabul edilir.
,	Virgüller ayrırcı olarak kullanılır. Birden çok virgöl, tırnak (') içine alınmış dizgiler dışında, tek bir virgülle eşdeğerdir.

Çizelge 1. Özel anlamları olan karakterlerin açıklamaları (devamı var)

Karakter	Açıklama
'	Kesme işareti, bir dizginin başlangıcını ya da sonunu gösterir. IBM MQ , tırnak içine alınmış tüm karakterleri tam olarak girildikleri gibi bırakır. Dizginin uzunluğu hesaplanırken, içeren kesme imleri içerilmez.
"	Bir dizginin içindeki tek tırnak işaretleri, dizginin uzunluğu hesaplanırken IBM MQ tarafından tek bir karakter olarak kabul edilir ve dizgi sonlandırılmaz.
=	 Açık z/OS, eşittir işareti, virgöl ya da boşluk ile biten bir parametre değerinin başlangıcını gösterir.
(Açma ayracı, bir parametre değerinin ya da değer listesinin başlangıcını gösterir.
)	Kapama ayracı, bir parametre değerinin ya da değer listesinin sonunu gösterir.
:	İki nokta üst üste işareti, kapsayıcı bir aralığı belirtir. Örneğin (1: 5), (1,2,3,4, 5) anlamına gelir. Bu gösterim yalnızca TRACE komutlarında kullanılabilir.
*	Yıldız işareti her şey demek. Örneğin, DISPLAY TRACE (*) tüm izlemelerin görüntülenmesi, DISPLAY QUEUE (PAY*) ise PAY ile başlayan tüm kuyrukların görüntülenmesi anlamına gelir.




MQSC giriş dosyası sözdizimi

Uzun komutlara sahipseniz ya da belirli bir komut dizisini arka arkaya kullanıyorsanız, MQSC komutlarını yayınlamak için bir giriş dosyası kullanabilirsiniz. Giriş dosyasının içeriği, bu konuda açıklanan sözdizimini izlemelidir.

Genel Bakış

MQSC komutları, `std` olarak da adlandırılan *standart giriş aygıt* aracılığıyla girilir. Genellikle bu klavye olur, ancak girişin bir giriş dosyasından geleceğini belirtebilirsiniz.

Bu giriş dosyasını, altyapıya özgü aşağıdaki araçlardan herhangi biriyle kullanabilirsiniz:

-  AIX, Linux, and Windows üzerinde **runmqsc** komutu. Bakınız "[runmqsc altındaki metin dosyalarından MQSC komutlarının çalıştırılması](#)" sayfa 22
-  IBM i üzerinde **STRMQM** komutu. Bakınız "[IBM i üzerinde MQSC komutlarını kullanarak yönetme](#)" sayfa 399
-  CSQINP1, CSQINP2 ve CSQINPX kullanıma hazırlama veri kümeleri ya da z/OS üzerinde CSQUTIL toplu iş yardımcı programı. Bakınız "[IBM MQ for z/OS üzerinde MQSC ve PCF komutlarını yayınlatabileceğiniz kaynaklar](#)" sayfa 457

Sözdizimi

MQSC giriş dosyası sözdizimi:

- IBM MQ ortamları arasında taşınabilirlik için, MQSC komut dosyalarındaki satır uzunluğunu 72 karakterle sınırlayın.
- Her komut yeni bir satırda başlamalıdır.
- İlk konumda yıldız işaretiyle (*) başlayan bir satır dikkate alınmaz. Bu, dosyaya yorum eklemek için kullanılabilir.
- Boş satırlar yoksayılır.
- Artı işareti (+), komutun sonraki satırın ilk boş olmayan karakterinden devam ettiğini gösterir. Bir komuta devam etmek için + kullanırsanız, sonraki parametreden önce en az bir boşluk bırakmayı unutmayın

(bunun gerekli olmadığı z/OS dışında). Komut tek bir dizgide yeniden birleştirildiğinde, açıklamalar ya da boş satırlar atılır.

- Eksi işareti (-), komutun sonraki satırın başlangıcından itibaren devam edileceğini gösterir. Komut tek bir dizgide yeniden birleştirildiğinde, açıklamalar ya da boş satırlar atılır.
- Bir Escape PCF (Programlanabilir Komut Biçimi) komutunda bulunan MQSC komutları artı işaretiyle ya da eksik işaretiyle devam edemez. Komutun tamamı tek bir Escape komutunda bulunmalıdır. PCF komutlarına ilişkin bilgi için bkz. "IBM MQ Programlanabilir Komut Biçimlerine Giriş" sayfa 25.
- Çoklu platformlar ve z/OS sistemlerinde, CSQUTIL toplu iş yardımcı programından verilen komutlar için, önceki satırın sonuna bir artı işareti (+) girmiş olsanız da, bir komutu sonlandırmak için noktalı virgül (;) kullanabilirsiniz.
- Bir satır, klavye denetim karakteriyle (örneğin, bir sekme) bitmemelidir.
- **runmqsc** komutunu bir metin dosyasından stdin dosyasını yeniden yönlendirerek istemci kipinde çalıştırırsanız ve kimlik bilgilerini sağlamak için **-u** işaretini sağlarsanız, **runmqsc** komutu parola istemez ve bunun yerine parola stdin' den okunur. stdin aracılığıyla sağlanan ilk veri satırının parola olduğundan emin olmanız gerekir. Bu, "echo" ya da "cat" gibi komut satırı araçları kullanılarak ve parolayı ardından MQSC komut dosyası **runmqsc** komutuna stdin geçirilerek yapılabilir.
- **Windows** Windows üzerinde, bir komut komut dosyasında (örneğin, bir nesne tanımlamasının parçası olarak) pound işareti (£) ve mantıksal NOT ([!]) gibi bazı özel karakterler kullanılırsa, bunlar **DISPLAY QLOCAL** gibi bir komutun çıkışında farklı görüntülenir.

Ayrıca bkz. "MQSC komutu sözdizimi" sayfa 12.

Örnekler

Aşağıdaki örnek, **DEFINE QLOCAL** komutunu gösteren bir MQSC komut dosyasından alınan bir alma örneğidir.

```
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +
DESCR(' ') +
PUT(ENABLED) +
DEFPRTY(0) +
DEFPSIST(NO) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(1024) +
DEFSOPT(SHARED) +
NOHARDENBO +
USAGE(NORMAL) +
NOTRIGGER;
```

Şekil 1. MQSC komut dosyasından al

runmqsc komutu tamamlandığında bir rapor döndürülür. Aşağıdaki örnek, bir rapordan alınan bir özdür:


```

Starting MQSC for queue manager jupiter.queue.manager.
.
.
12:  DEFINE QLOCAL('ORANGE.LOCAL.QUEUE') REPLACE +
:    DESCR(' ') +
:    PUT(ENABLED) +
:    DEFPRTY(0) +
:    DEFPSIST(NO) +
:    GET(ENABLED) +
:    MAXDEPTH(5000) +
:    MAXMSGL(1024) +
:    DEFSOPT(SHARED) +
:    NOHARDENBO +
:    USAGE(NORMAL) +
:    NOTRIGGER;
AMQ8006: IBM MQ queue created.
.
.
.

```

Şekil 2. MQSC komut raporu dosyasından çek

Metin dosyanızı yaratmanıza yardımcı olması için örnek MQSC komut dosyalarını da kullanabilirsiniz:

amqscos0.tst

Örnek programlar tarafından kullanılan nesnelerin tanımları.

amqscic0.tst

CICS hareketlerine ilişkin kuyrukların tanımları.

Linux **AIX** AIX and Linux' da bu dosyalar `MQ_INSTALLATION_PATH/sampdizininde` bulunur. `MQ_INSTALLATION_PATH` , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

Windows Windows' da bu dosyalar `MQ_INSTALLATION_PATH\tools\mqsc\samplesdizininde` bulunur. `MQ_INSTALLATION_PATH` , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

ALW runmqsc altında MQSC komutlarının etkileşimli olarak çalıştırılması

AIX, Linux, and Windowsüzerinde, bir kuyruk yöneticisine etkileşimli olarak MQSC komutları vermek için **runmqsc** komut istemini kullanabilirsiniz. Etkileşimli çalışma özellikle hızlı testler için uygundur.

Başlamadan önce

Çalıştığınız kuyruk yöneticisiyle ilişkili kuruluşa **runmqsc** komutunu kullanmanız gerekir. `dspmq -o installation` komutunu kullanarak, bir kuyruk yöneticisinin hangi kuruluşa ilişkilendirildiği öğrenebilirsiniz.

MQPROMPT ortam değişkenini kullanarak seçtiğiniz bir bilgi istemini ayarlayarak, MQSC ortamında olduğunuzu daha kolay görebilir ve yürürlükteki ortamın bazı ayrıntılarını görebilirsiniz. Daha fazla bilgi için bkz [“MQSC komut istemi ayarlanıyor” sayfa 19.](#)

Linux **AIX** AIX and Linux altyapılarında MQSC komutlarını etkileşimli olarak çalıştırdığınızda, **runmqsc** komut istemi ek komut satırı düzenleyicisi işlevlerini de destekler. Bkz. [“runmqsc için komut geri çağırma ve tamamlama ve Emacs komut tuşlarının etkinleştirilmesi” sayfa 21.](#)

Bu görev hakkında

runmqsc komutu, MQSC komutlarını verebileceğiniz bir komut istemini açmak için kullanılır. Bu komutlar ve sözdizimi, [MQSC komutları başvurusunda](#) açıklanmıştır.

runmqsc komut istemini bu görevde açıklandığı gibi başlattığınızda, komutta ayarlanan işaretlere bağlı olarak, komut istemini üç kipten birinde çalışacak şekilde ayarlıyorsunuz:

- *Doğrulama kipi*; burada MQSC komutları yerel bir kuyruk yöneticisinde doğrulanır, ancak çalıştırılır.
- *Doğrudan kip*; burada MQSC komutları yerel bir kuyruk yöneticisinde çalıştırılır.
- MQSC komutlarının uzak bir kuyruk yöneticisinde çalıştırıldığı *dolaylı kip*.

Aşağıdaki yordam, bilgi istemini doğrudan kipte çalışacak şekilde ayarlar. Diğer seçenekler, ana adımları izleyen örneklerde gösterilir.

Yordam

1. Bir komut penceresi ya da kabuk açın ve aşağıdaki komutu girin:

```
runmqsc QMgrName
```

Burada *QMgrName* , MQSC komutlarını işlemek istediğiniz kuyruk yöneticisinin adını belirtir. Varsayılan kuyruk yöneticisindeki MQSC komutlarını işlemek için *QMgrName* ögesini boş bırakabilirsiniz.

2. Herhangi bir MQSC komutunu gerektiği şekilde yazın. Örneğin, ORANGE . LOCAL . QUEUE adlı bir yerel kuyruk yaratmak için şu komutu girin:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE)
```

Bir satıra sığmayacak kadar çok parametre içeren komutlar için, bir komutun aşağıdaki satırda devam ettiğini belirtmek üzere devam karakterlerini kullanın:

- Eksi işareti (-), komutun aşağıdaki satırın başlangıcından itibaren devam edileceğini belirtir.
- Artı işareti (+), komutun sonraki satırdaki ilk boş olmayan karakterden devam ettirileceğini gösterir.

Komut girişi, devam karakteri olmayan, boş olmayan bir satırın son karakteriyle sona erer. Komut girişini, noktalı virgül (;) girerek de belirtik olarak sonlandırabilirsiniz.

3. Aşağıdaki komutu girerek MQSC komutlarıyla çalışmayı durdurun:

```
end
```

Diğer bir seçenek olarak, işletim sisteminiz için **exit** komutunu, **quit** komutunu ya da EOF karakterini kullanabilirsiniz.

Sonuçlar

MQSC komutlarını verdiğinizde, kuyruk yöneticisi işlemlerinizi onaylayan ya da yaptığınız hataları size anlatan işletmen iletileri döndürür. Örneğin, aşağıdaki ileti bir kuyruğun yaratıldığını doğrular:

```
AMQ8006: IBM MQ queue created.
```

Aşağıdaki ileti bir sözdizimi hatası yaptığınızı gösterir:

```
AMQ8405: Syntax error detected at or near end of command segment below:-  
AMQ8426: Valid MQSC commands are:
```

```
ALTER  
CLEAR  
DEFINE  
DELETE  
DISPLAY  
END  
PING  
REFRESH  
RESET  
RESOLVE  
RESUME  
START
```

```
STOP
SUSPEND
4 : end
```

Bu iletiler, varsayılan olarak görüntü olan standart çıkış aygıtına gönderilir. Komutu doğru girmediyse, doğru sözdizimini bulmak için komuta ilişkin başvuru bilgilerine bakın. Bkz. [MQSC komutları başvurusu](#).

Örnek

Aşağıdakiler, önceki adımlarda kullanılan `runmqsc QMgrName` komutunun değişkenleridir. Bu değişkenler, **runmqsc** komut istemine ilişkin farklı yapılandırmalar oluşturur.

- Aşağıdaki komut, MQSC yorumlayıcısına tek bir MQSC komutu geçirmek için komut süzme özelliğini kullanır.

Windows'ta:

```
echo display chstatus(*) | runmqsc QMgrName
```

Linux'ta:

```
echo "display chstatus(*)" | runmqsc QMgrName
```

- Aşağıdaki komut bir kuyruk yöneticisi adını belirtmediğinden, MQSC komutları varsayılan kuyruk yöneticisinde işlenir.

```
runmqsc
```

- Bu komut, komutları sunmak için QMLOCAL komutunu kullanarak QMREMOTE kuyruk yöneticisine komut gönderir.

```
runmqsc -w 30 -m QMLOCAL QMREMOTE
```

- IBM MQ 9.2.0' den bu komut, komutları çalıştırmadan yerel bir kuyruk yöneticisinde komut sözdiziminin doğru olduğunu doğrular. Doğrulanacak komutların `myprog.in` giriş kütüğünden okunduğunu unutmayın.

```
runmqsc -f myprog.in -v QMgrName
```

Giriş ve çıkış dosyalarıyla çalışma hakkında daha fazla bilgi için bkz. [“runmqsc altındaki metin dosyalarından MQSC komutlarının çalıştırılması” sayfa 22](#).

Sonraki adım

runmqsc komut sözdiziminin, isteğe bağlı parametrelerin ve dönüş kodlarının tüm ayrıntıları için bkz. [runmqsc \(MQSC komutlarının çalıştırılması\)](#).

İlgili görevler

[“runmqsc altındaki metin dosyalarından MQSC komutlarının çalıştırılması” sayfa 22](#)

Uzun komutlara sahipseniz ya da belirli bir komut sırasını arka arkaya kullanıyorsanız, MQSC komutlarını yayınlamak için bir metin dosyası kullanabilirsiniz. `std.in` dosyasını bir metin dosyasından yeniden yönlendirebilirsiniz. Çıkışı bir dosyaya yeniden yönlendirebilirsiniz.

İlgili başvurular

[MQSC komutları başvurusu](#)

MQSC komut istemi ayarlanıyor

AIX, Linux, and Windows işletim sistemlerinde, **runmqsc** komutunu çalıştırdığınızda görüntülenen bilgi istemini ayarlamak için **MQPROMPT** ortam değişkenini kullanın. Bu, MQSC ortamında olduğunuzu görmeyi ve yürürlükteki ortamın bazı ayrıntılarını görmeyi kolaylaştırır.

Bu görev hakkında

runmqsc komutunu çalıştırdığınızda görüntülenen bilgi istemini ayarlayabilirsiniz. Bilgi istemi hem **runmqsc** komutu etkileşimli olarak çalıştırıldığında hem de giriş bir dosyadan ya da standart giriş aygıtından (stdin) **runmqsc** ' e yeniden yönlendirildiğinde eklenir.

Komut istemine düz metin ekleyebilir ve ortam değişkenlerini, +VARIABLE+ gösterimini IBM MQ hizmet nesnesi tanımlamalarıyla aynı şekilde kullanarak da ekleyebilirsiniz. Daha fazla bilgi için bkz "[Hizmet tanımlarında değiştirilebilir eklemeleri kullanma](#)" sayfa 203.

IBM MQ tarafından sağlanan ve aşağıdaki tabloda açıklanan başka bir dizi ek değiştirilebilir ekleme vardır.

Değiştirilebilir ekleme	Açıklama
MQ_HOST_NAME	Sistemin anasistem adı
MQ_FILE_SEP	Platforma özgü dosya ayırıcısı: <ul style="list-style-type: none">► Linux ► AIX AIX and Linux sistemlerinde MQ_FILE_SEP , '/' dir.► Windows Windows sistemlerinde MQ_FILE_SEP konumu şudur: \
MQ_PATH_SEP	Platforma özgü yol ayırıcısı: <ul style="list-style-type: none">► Linux ► AIX AIX and Linux sistemlerinde MQ_PATH_SEP , ':' dir.► Windows Windows sistemlerinde MQ_PATH_SEP konumu şudur: ;
MQ_DATE_TIME	Sabit bir YYYY-MM-DD hh:mm:ss.SSS biçiminde yerel sistem tarihi ve saati; örneğin: 2020-12-25 17:41:37.408

Notlar:

- MQ değiştirilebilir ekleme değerleri, **runmqsc** komutunun ilişkilendirildiği IBM MQ kurulumu ve anasistemiyle ilişkilidir.
- MQPROMPT** , ekler genişletildiğinde en çok 256 karakterle sınırlıdır. Bu değer üzerindeki **MQPROMPT** genişletmeleri, tüm **MQPROMPT** dizgisinin genişlemeler olmadan kesilmesine neden olur.

Örnek

Aşağıdaki örnek, bilgi istemini MQSC olarak ayarlar:

- Linux ► AIX

```
export MQPROMPT="MQSC"
```

- Windows

```
set "MQPROMPT=MQSC"
```

AIX

Aşağıdaki örnek, AIX sisteminde **MQPROMPT** değişkenini ayarlar. Bilgi istemi, bir kullanıcı adını (ilişkili sistem ortam değişkeninden alınan), kuyruk yöneticisi adını ve IBM MQ anasistem adını (IBM MQ değiştirilebilir eklerinden alınan) görüntüleyecek şekilde ayarlanır:

```
sh> export MQPROMPT="+USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
sh> runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
myuser @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

```
C:\ > set "MQPROMPT+=USERNAME+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
C:\ > runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
myuser @ MY.QMGR @ WIN1> DISPLAY QMSTATUS
```

Aşağıdaki örnek, MQ değiştirilebilir eklerinden alınan yukarıdaki **MQPROMPT** örneklerine bir zaman damgası ekler:

```
sh> export MQPROMPT="+MQ_DATE_TIME+ +USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
sh> runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
2020-11-24 18:10:00.404 myuser @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

```
C:\ > set "MQPROMPT+=MQ_DATE_TIME+ +USERNAME+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
C:\ > runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
2020-11-24 18:10:01.007 myuser @ MY.QMGR @ WIN1> DISPLAY QMSTATUS
```

Linux

AIX

runmqsc için komut geri çağırma ve tamamlama ve Emacs komut tuşlarının etkinleştirilmesi

Komut geri çağırma, komut tamamlama ve Emacs komut tuşlarını etkinleştirmek için AIX and Linux üzerindeki **runmqsc** komut istemini kullanın.

Bu görev hakkında

AIX and Linux sistemlerinde, **runmqsc** komut isteminden aşağıdaki ek komut satırı düzenleyicisi işlevlerini kullanılabilir kılabilir:

- Yukarı ok tuşunu ve aşağı ok tuşunu kullanarak önceden girilen komutları geri çağırma
- Sekme tuşunu ve ara çubuğunu kullanarak bir komutun sonraki anahtar sözcüğü için otomatik tamamlama
- **Emacs** komut tuşları ya da benzer komut tuşu işlevleri

Bu işlevleri kullanmak için lanetler kitaplığının kurulması gerekir. Sisteminizde lanetler kitaplığı kurulu değilse, **runmqsc** bilgi isteminde komut satırı düzenleyicisi işlevleri yoktur ve **runmqsc** komut istemi başlatıldığında bir ileti görüntülenir. Kurulacak lanetler kitaplığının adı UNIX platformuna bağlıdır:

- **AIX** AIX'da curses' yi kurun.
- **Linux** Linux'da ncurses' yi kurun.

Yordam

- ncurses ya da cursesürününü kurun.

Not: Aşağıdaki örnekte Linux ile ilgili yönergeler kullanılmıştır.

Var olan ncurses paketlerini bulmak için aşağıdaki komutu çalıştırın:

```
rpm -qa | grep -i ncurses
```

Gerekli ncurses paketleri aşağıdaki gibidir:

```
ncurses-term-6.1-7.20180224.el8.noarch
ncurses-6.1-7.20180224.el8.x86_64
ncurses-base-6.1-7.20180224.el8.noarch
ncurses-c++-libs-6.1-7.20180224.el8.x86_64
ncurses-libs-6.1-7.20180224.el8.x86_64
ncurses-compat-libs-6.1-7.20180224.el8.x86_64
ncurses-devel-6.1-7.20180224.el8.x86_64
```

Aşağıdaki komutu çalıştırarak, önceki metinde listelenen tüm gerekli ncurses paketlerini kurabilirsiniz:

```
yum install ncurses*
```

- Emacs tuş bağ tanımlarını uyarlayın.

Komutlara bağlı tuşları özelleştirebilirsiniz. Örneğin, anahtarları varsayılan Emacs tuş bağ tanımları yerine vi bağ tanımlarına bağlayabilirsiniz.

Anahtarlar, ana dizinde saklanan `.editrc` dosyası düzenlenerek özelleştirilebilir. Daha fazla bilgi için FreeBSD man sayfalarında `editrc` başlıklı konuya bakın.

- Komut geri çağırma, komut tamamlama ve Emacs komut tuşlarını devre dışı bırakın.

Bunu yapmak için `MQ_OVERRIDE_LIBEDIT_LOAD` ortam değişkenini TRUEolarak ayarlayın.

`runmqsc` komut istemi aşağıdaki bilgi iletisini görüntülediğinde bu ortam değişkeni geçici çözüm olarak kullanılabilir:

```
AMQ8521I: Command completion and history unavailable
```

ALW `runmqsc` altındaki metin dosyalarından MQSC komutlarının çalıştırılması

Uzun komutlara sahipseniz ya da belirli bir komut sırasını arka arkaya kullanıyorsanız, MQSC komutlarını yayınlamak için bir metin dosyası kullanabilirsiniz. `std.in` dosyasını bir metin dosyasından yeniden yönlendirebilirsiniz. Çıkışı bir dosyaya yeniden yönlendirebilirsiniz.

Başlamadan önce

Bu görevde, çalıştırmak istediğiniz MQSC komutlarını içeren bir metin dosyası yarattığınız varsayılır. Bu dosyaların ayrıntılı sözdizimi ve örnekleri için bkz. "[MQSC giriş dosyası sözdizimi](#)" sayfa 15.

`MQPROMPT` ortam değişkenini kullanarak, MQSC komut istemini istediğiniz bir bilgi istemine ayarlayabilirsiniz. Daha fazla bilgi için bkz. "[MQSC komut istemi ayarlanıyor](#)" sayfa 19.

Bu görev hakkında

`runmqsc` komutuna ilişkin giriş, `std.in`olarak da adlandırılan *standart giriş aygıtı*ndan alınır. Genellikle bu klavye olur, ancak girişin bir dizisel kapıdan ya da dosyadan geleceğini belirtebilirsiniz.

`runmqsc` komutunun çıkışı, `stdout`olarak da adlandırılan *standart çıkış aygıtı*na çıkışa yazılır. Genellikle bu bir görüntüdür, ancak çıkışı bir dizisel kapıya ya da dosyaya yeniden yönlendirebilirsiniz.

Yordam

1. Yerel bir kuyruk yöneticisinde, komutları çalıştırmadan dosyadaki komut sözdiziminin doğru olduğunu doğrulayın.

Aşağıdaki seçeneklerden biriyle birlikte **runmqsc** komutundaki **-v** işaretini kullanın:

- IBM MQ 9.2.0' den giriş metni dosya adını tanımlamak için **-f** seçeneğini kullanın. Örneğin:

```
runmqsc -f myprog.in -v localQmgrName
```

- IBM MQ 9.2.0'den önceki Long Term Support yayınları ve IBM MQ 9.1.4' den önceki Continuous Delivery yayınları için, MQSC komutlarını giriş metin dosyasından komuta yönlendirmek için **<** işlecini kullanın. Örneğin:

```
runmqsc -v localQmgrName < myprog.in
```

Komutları doğrularken uzak kuyruk yöneticisi belirtemezsiniz. Yani, **-w** işaretini belirtemezsiniz.

Döndürülen rapor, Şekil 2 sayfa 17 içinde gösterilen rapora benzer.

2. Komut sözdizimi doğru olduğunda, **-v** işaretini kaldırın ve **runmqsc** komutunu yeniden çalıştırın.

Artık bir uzak kuyruk yöneticisi belirtebileceğinizi unutmayın.

- IBM MQ 9.2.0 ' den (örneğin) aşağıdaki komutu çalıştırın:

```
runmqsc -f myprog.in QmgrName
```

- IBM MQ 9.2.0 öncesi Long Term Support yayınları ve IBM MQ 9.1.4 öncesi Continuous Delivery yayınları için aşağıdaki komutlardan birini kullanın:
 - **<** işleci, girişi bir metin dosyasından yönlendirir. Örneğin, aşağıdaki komut `myprog.in` metin dosyasında bulunan bir dizi komutu çalıştırır:

```
runmqsc QMgrName < myprog.in
```

- **>** işleci, çıkışı bir metin dosyasına yönlendirir. Örneğin, aşağıdaki komut `myprog.in` metin dosyasında bulunan bir dizi komutu çalıştırır ve bunu `results.out` adlı bir dosyaya çıkarır:

```
runmqsc QMgrName < myprog.in > results.out
```

Şekil 1 sayfa 16 , `myprog.in` ve Şekil 2 sayfa 17 gibi bir komut dosyasından alınan bir ayıklamanın, `results.out` gibi bir rapor dosyasından alınan çıkışa karşılık gelen ayıklamasını gösterir.

Sonraki adım

runmqsc komut sözdiziminin, isteğe bağlı parametrelerin ve dönüş kodlarının tüm ayrıntıları için bkz. [runmqsc \(MQSC komutlarının çalıştırılması\)](#).

İlgili görevler

[“MQSC komut istemi ayarlanıyor” sayfa 19](#)

AIX, Linux, and Windows işletim sistemlerinde, **runmqsc** komutunu çalıştırdığınızda görüntülenen bilgi istemini ayarlamak için **MQPROMPT** ortam değişkenini kullanın. Bu, MQSC ortamında olduğunuzu görmeyi ve yürürlükteki ortamın bazı ayrıntılarını görmeyi kolaylaştırır.

[“runmqsc altında MQSC komutlarının etkileşimli olarak çalıştırılması” sayfa 17](#)

AIX, Linux, and Windows üzerinde, bir kuyruk yöneticisine etkileşimli olarak MQSC komutları vermek için **runmqsc** komut istemini kullanabilirsiniz. Etkileşimli çalışma özellikle hızlı testler için uygundur.

İlgili başvurular

[MQSC komutları başvurusu](#)

Multi Başlatma sırasında bir MQSC komut dosyasından otomatik yapılandırma

IBM MQ 9.2.0' den kuyruk yöneticinizi, her kuyruk yöneticisi başlangıcında bir MQSC komut dosyasının ya da MQSC komut dosyası kümesinin içeriğini otomatik olarak uygulaması için yapılandırabilirsiniz.

Değiştirilebilen bir yapılanışa sahip olmak için bu işlevi kullanabilir ve sonraki kuyruk yöneticisi yeniden başlatıldığında otomatik olarak yeniden yürütülebilir. Örneğin, komut dosyası ya da komut dosyaları sisteme bağlanmış bir sürücüde bulunuyorsa, en son sürümün başladıkları sırada her kuyruk yöneticisine uygulandığı merkezi bir yapılandırmaya sahip olmak mümkündür.

Bunun yararlı olabileceği belirli bir senaryo, tek tip bir kümenin, tümü için geçerli olan tek bir yapılandırma kümesine sahip olarak kümedeki tüm kuyruk yöneticilerine aynı tanımlamaları içermesini sağlamaktır. Bunun bir örneği için bkz. [Yeni bir tek küme oluşturma](#).

Başlamadan önce

Aşağıdakileri kullanabilirsiniz:

1. Tek bir komut dosyası ve MQSC komutlarını kullanarak bir metin dosyası yaratın.
2. Bir MQSC komut dosyaları kümesi:
 - Yapılandırmaların var olduğu bir dizini tanımlamak için, ve
 - Bu dizinde, her biri .mqscuzantılı dosyalar oluşturun; örneğin, queues.mqsc.

Bu komut dosyasının her kuyruk yöneticisi başlangıcında yeniden uygulandığı düşünülürse, komutların yeniden yürütülebilir olması önemlidir. Örneğin, bir **DEFINE** komutu **REPLACE** dizgisini içermelidir; tersi durumda, nesne zaten var olduğu için, komut ikinci kuyruk yöneticisi başlangıcında bir hata olarak görünür.

Bir MQSC komut dosyasında, * önekli satırın bir açıklama olarak kabul edildiğini unutmayın.

MQSC komut dosyalarının otomatik yapılandırmasını etkinleştirme

Önemli: Başlatma sırasında otomatik yapılandırma için desteklenmediğinden, MQTT tipi kanallar için komut yayınlamamalısınız.

crtmqm komutunun **-ic** işaretini kullanarak ve belirli bir dosyayı ya da dizini göstererek yeni bir kuyruk yöneticisi yapılandırabilirsiniz. Sağlanan değer, qm.in.i dosyasında AutoConfig kısmı altında **MQSCConfig** özniteliği olarak saklanır.

Geçerli bir dosyayı ya da dizini gösteren AutoConfig kısmı özniteliğini **MQSCConfig** ekleyerek, var olan bir kuyruk yöneticisini otomatik MQSC yapılandırmasını etkinleştirecek şekilde yapılandırabilirsiniz. Örneğin:

```
AutoConfig:
MQSCConfig=C:\mq_configuration\uniclus.mqsc
```

Otomatik yapılandırma nasıl çalışır?

Kuyruk yöneticisi başlatılırken, **AutoConfig** stanza özniteliği **MQSCConfig** tarafından tanımlanan yapılanış, geçerli sözdizimi sağlamak için **runmqsc** geçerlilik denetiminden geçirilir ve kuyruk yöneticisi veri ağacında autocfg tek bir dosya cached.mqsc olarak saklanır.

Bir dizindeki birden çok dosya işlendiğinde, bunlar alfabetik sırayla işlenir ve bir MQSC end ya da quit komutu içeriyorsa, o dosyanın geri kalan içeriği atlanır.

Kuyruk yöneticisinin ilk başlatılması sırasında, dosya ya da dizin okunamadı ya da MQSC sözdizimi geçerli olmayan bir dosya, kuyruk yöneticisinin hem konsola hem de kuyruk yöneticisi hata günlüğüne uygun bir hata iletişisiyle başlatılmasını önler.

Sonraki yeniden başlatmalarda, gösterilen dosya ya da dizin okunamıyorsa ya da geçersiz MQSC sözdizimi içeriyorsa, önceden önbelleğe alınan dosya kullanılır ve kuyruk yöneticisinin hata günlüğüne yazılan bir ileti bunu vurgular.

V9.3.0 Bu noktada, cached.mqsc içeriği kuyruk yöneticisine uygulanır; tüm MQSC komutları uygulandığında, kuyruk yöneticisi uygulamaların bağlanması için etkinleştirilir. Uygulanmakta olan yapılanışın **runmqsc** günlüğü, kuyruk yöneticisinin hata dizininde autocfgmqsc.LOG adlı bir dosya olarak saklanır.

Ayrıca, başarıyla tamamlanmayan herhangi bir MQSC komutu kuyruk yöneticisi hata günlüğüne kaydedilir ve komutun neden başarısız olduğunu gösterir.

IBM MQ yönetiminin PCF komutlarıyla otomatikleştirilmesi

Bazı yönetim ve izleme görevlerini otomatikleştirmenin kuruluşunuz için yararlı olacağına karar verebilirsiniz. Programlanabilir komut biçimi (PCF) komutlarını kullanarak hem yerel hem de uzak kuyruk yöneticileri için denetim görevlerini otomatikleştirebilirsiniz. Bu bölümde, IBM MQ nesnelere yönetme konusunda deneyiminiz olduğu varsayılır.

PCF komutları

IBM MQ programlanabilir komut biçimi (PCF) komutları, denetim görevlerini bir denetim programına programlamak için kullanılabilir. Bu şekilde, bir programdan kuyruk yöneticisi nesnelere (kuyruklar, süreç tanımlamaları, ad alanları, kanallar, istemci bağlantı kanalları, dinleyiciler, hizmetler ve kimlik doğrulama bilgileri nesnelere) işleyebilir ve kuyruk yöneticilerinin kendisini bile işleyebilirsiniz.

PCF komutları, MQSC komutları tarafından sağlanan aynı işlev aralığını kapsar. Ağıdaki herhangi bir kuyruk yöneticisine tek bir düğümden PCF komutları yayınlamak için bir program yazabilirsiniz. Bu şekilde, yönetim görevlerini hem merkezleştirir hem de otomatikleştirebilirsiniz.

Her PCF komutu, IBM MQ iletilerinin uygulama verileri bölümüne gömülü bir veri yapısıdır. Her komut, MQPUT MQI işlevi kullanılarak hedef kuyruk yöneticisine diğer iletilerle aynı şekilde gönderilir. Komut sunucusu iletiyi alan kuyruk yöneticisinde çalışıyorsa, komut sunucusu iletiyi bir komut iletilisi olarak yorumlar ve komutu çalıştırır. Yanıtları almak için uygulama bir MQGET çağrısı yayınlar ve yanıt verileri başka bir veri yapısında döndürülür. Daha sonra uygulama yanıtı işleyebilir ve buna göre işlem yapabilir.

Not: MQSC komutlarından farklı olarak, PCF komutları ve yanıtları okuyabileceğiniz bir metin biçiminde değildir.

Kısaca, bunlar bir PCF komut iletilisi yaratmak için gereken bazı şeylerdir:

İleti tanımlayıcı

Bu, aşağıdaki özelliklere sahip standart bir IBM MQ ileti tanımlayıcısıdır:

- İleti tipi (*MsgType*) MQMT_REQUEST.
- İleti biçimi (*Format*) MQFMT_ADMIN.

Uygulama Verileri

PCF üstbilgisini içeren PCF iletilisini içerir. Bu iletilide aşağıdakiler yer alır:

- PCF ileti tipi (*Type*) MQCFT_COMMAND belirtiyor.
- Komut tanıtıcısı komutu belirtir; örneğin, *Change Queue* (MQCMD_CHANGE_Q).

PCF veri yapılarının tam açıklaması ve bunların nasıl uygulanacağına ilişkin bilgi için bkz. [“IBM MQ Programlanabilir Komut Biçimlerine Giriş” sayfa 25.](#)

PCF nesnesi öznitelikleri

PCF ' deki nesne öznitelikleri, MQSC komutları için olduğu için sekiz karakterle sınırlı değildir. Bu kılavuzda italik olarak gösterilmiştir. Örneğin, RQMNAME değerinin PCF eşdeğeri *RemoteQMGrName* ' dir.

Kaçış PCF ' leri

Çıkış PCF ' leri, ileti metni içinde MQSC komutlarını içeren PCF komutlarıdır. Uzak kuyruk yöneticisine komut göndermek için PCF ' leri kullanabilirsiniz. Kaçış PCF ' lerine ilişkin ek bilgi için bkz. [Escape](#).

IBM MQ Programlanabilir Komut Biçimlerine Giriş

Programlanabilir Komut Biçimleri (PCF), bir program ile bir ağdaki herhangi bir kuyruk yöneticisi (PCF ' leri destekleyen) arasında değiş tokuş edilebilecek komut ve yanıt iletilerini tanımlar. PCF ' ler kuyruk

yöneticisi yönetimini ve diğer ağ yönetimini basitleştirir. Özellikle ağların boyutu ve karmaşıklığı arttıkça dağıtılmış ağların karmaşık yönetimi sorununu çözmek için kullanılabilir.

Programlanabilir Komut Biçimleri tüm IBM MQ platformlarında desteklenir.

Sorun PCF komutlarının çözülmesi

Dağıtılmış ağların yönetimi karmaşık hale gelebilir. Ağların boyutu ve karmaşıklığı arttıkça yönetim sorunları da büyümeye devam ediyor.

İleti sistemine ve kuyruğa alma denetimine özgü örnekler şunlardır:

- Kaynak yönetimi.
Örneğin, kuyruk oluşturma ve silme.
- Performans izleme.
Örneğin, kuyruk derinliği ya da ileti hızı üst sınırı.
- Kontrol.
Örneğin, kuyruk derinliği üst sınırı, ileti uzunluğu üst sınırı ve kuyrukları etkinleştirme ve devre dışı bırakma gibi kuyruk parametrelerinin ayarlanması.
- İleti yönlendirmesi.

Bir ağ üzerinden alternatif rotaların tanımı.

IBM MQ PCF komutları, kuyruk yöneticisi yönetimini ve diğer ağ yönetimini basitleştirmek için kullanılabilir. PCF komutları, ağ içindeki tek bir kuyruk yöneticisinden ağ yönetimi gerçekleştirmek için tek bir uygulama kullanmanızı sağlar.

PCF ' ler nedir?

PCF 'ler, bir program ile bir ağdaki herhangi bir kuyruk yöneticisi (PCF' leri destekleyen) arasında değiş tokuş edilebilecek komut ve yanıt iletilerini tanımlar. IBM MQ nesnelерinin denetimi için bir sistem yönetimi uygulama programındaki PCF komutlarını kullanabilirsiniz: kimlik doğrulama bilgileri nesneleri, kanallar, kanal dinleyicileri, ad listesi, süreç tanımlamaları, kuyruk yöneticileri, kuyruklar, hizmetler ve depolama sınıfları. Uygulama, yerel kuyruk yöneticisini kullanarak herhangi bir kuyruk yöneticisiyle, yerel ya da uzak kuyruk yöneticisiyle komut ve yanıt bilgilerini iletmek için ağdaki tek bir noktadan çalışabilir.


Her kuyruk yöneticisinin standart bir kuyruk adı olan bir denetim kuyruğu vardır ve uygulamanız o kuyruğa PCF komut iletileri gönderebilir. Her kuyruk yöneticisinin, denetim kuyruğundaki komut iletilerine hizmet vermek için bir komut sunucusu da vardır. Bu nedenle, PCF komut iletileri ağdaki herhangi bir kuyruk yöneticisi tarafından işlenebilir ve yanıt verileri, belirlediğiniz yanıt kuyruğunu kullanarak uygulamanıza geri gönderilebilir. PCF komutları ve yanıt iletileri, olağan İleti Kuyruğu Arabirimi (MQI) kullanılarak gönderilir ve alınır.

Parametreleri de içinde olmak üzere, kullanılabilir PCF komutlarının bir listesi için [Programlanabilir Komut Biçimleri Tanımları](#) başlıklı konuya bakın.

IBM MQ Programlanabilir Komut Biçimlerini Kullanma

IBM MQ uzak yönetimi için bir sistem yönetimi programında PCF ' leri kullanabilirsiniz.

Bu bölümde aşağıdakiler yer alır:

- [“PCF komut iletileri” sayfa 27](#)
- [“IBM MQ içinde PCF yanıtları” sayfa 29](#)
-  [“Genişletilmiş yanıtlar” sayfa 31](#)
- [IBM MQ nesnelерini adlandırma kuralları](#)
- [“IBM MQ içinde PCF komutları için yetki denetimi” sayfa 32](#)

PCF komut iletileri

PCF komut iletileri bir PCF üstbilgisinden, bu üstbilgide tanımlanan deęiřtirgelerden ve kullanıcı tanımlı ileti verilerinden oluşur. İletiler, İleti Kuyruęu arabirim çağrılarını kullanarak yayınlanır.

Her komut ve deęiřtirgeleri, PCF üstbilgisini ve ardından bir dizi deęiřtirge yapısı içeren ayrı bir komut iletileri olarak gönderilir; PCF üstbilgisinin ayrıntıları için bkz. [MQCFH-PCF üstbilgisine](#) bir deęiřtirge yapısı örneęi için bkz. [MQCFST-PCF dizgi deęiřtirgesi](#). PCF üstbilgisi, komutu ve aynı iletide izleyen parametre yapılarının sayısını tanıtır. Her parametre yapısı komuta bir parametre sağlar.

Komut sunucusu tarafından oluşturulan komutlara verilen yanıtlar benzer bir yapıya sahiptir. Bir PCF üstbilgisi ve ardından bir dizi parametre yapısı vardır. Yanıtlar birden çok iletiden oluşabilir, ancak komutlar her zaman tek bir iletiden oluşur.

Multi Çoklu platformlar işletim sistemi üzerinde, PCF komutlarının gönderildięi kuyruęa her zaman SYSTEM.ADMIN.COMMAND.QUEUE.

z/OS z/OS işletim sistemi üzerinde, komutlar SYSTEM.COMMAND.INPUTGIRIřI, ANCAK SYSTEM.ADMIN.COMMAND.QUEUE onun için bir dięer ad olabilir. Bu kuyruęa hizmet veren komut sunucusu, yanıtları, komut iletisinin ileti tanımlayıcısındaki *ReplyToQ* ve *ReplyToQMGr* alanları tarafından tanımlanan kuyruęa gönderir.

PCF komut iletilerinin yayınları

Kuyruklarına PCF komutu ve yanıt iletileri koymak ve kuyruklarından almak için olaęan MQI (Message Queue Interface; İleti Kuyruęu Arabirimi) çağrılarını, MQPUT, MQGET vb. kullanın.

Not:

Komut sunucusunun, o kuyruk yöneticisinde işlenecek PCF komutu için hedef kuyruk yöneticisinde çalıştıęını doğrulayın.

Saęlanan üstbilgi dosyalarının listesi için bkz. [IBM MQ COPY, header, include and module files](#).

PCF komutuna iliřkin ileti tanımlayıcı

IBM MQ ileti tanımlayıcısı, [MQMD-Message descriptor](#) içinde tam olarak belgelenmiştir.

Bir PCF komut iletileri, ileti tanımlayıcısında ařaęıdaki alanları içerir:

Rapor

Herhangi bir geçerli deęer, gerektięi şekilde.

MsgType

Yanıt gerektiren bir iletiyi belirtmek için bu alan MQMT_REQUEST olmalıdır.

Son kullanma tarihi

Herhangi bir geçerli deęer, gerektięi şekilde.

Geri bildirim

MQFB_NONE olarak ayarla

Multi Kodlama

Bir IBM MQ for Multiplatforms sistemine gönderi yapıyorsanız, bu alanı ileti verileri için kullanılan kodlamaya ayarlayın. Gerekirse dönüřtürme gerçekleştirilir.

Multi CodedCharSetId

Bir IBM MQ for Multiplatforms sistemine gönderi yapıyorsanız, bu alanı ileti verileri için kullanılan kodlanmış karakter takımı tanıtıcısına ayarlayın. Gerekirse dönüřtürme gerçekleştirilir.

Biçim

MQFMT_ADMIN olarak ayarlayın.

Öncelik

Herhangi bir geçerli deęer, gerektięi şekilde.

Kalıcılık

Herhangi bir geçerli değer, gerektiği şekilde.

MsgId

Gönderen uygulama herhangi bir değer belirtebilir ya da kuyruk yöneticisinden benzersiz bir ileti tanıtıcısı oluşturmasını istemek için MQMI_NONE belirtilebilir.

CorrelId

Gönderen uygulama herhangi bir değer belirtebilir ya da ilinti tanıtıcısı olmadığını belirtmek için MQCI_NONE belirtilebilir.

ReplyToQ

Yanıtı alacak kuyruğun adı.

ReplyToQMGr

Yanıta ilişkin kuyruk yöneticisinin adı (ya da boşluk).

İleti bağlamı alanları

Bu alanlar, gerektiği şekilde, herhangi bir geçerli değere ayarlanabilir. Olağan durumda, ileti bağlamı alanlarını varsayılan değerlere ayarlamak için Put ileti seçeneği MQPMO_DEFAULT_CONTEXT kullanılır.

version-2 MQMD yapısı kullanıyorsanız, aşağıdaki ek alanları ayarlamanız gerekir:

GroupId

MQGI_NONE olarak ayarla

MsgSeqNumarası

1 olarak ayarla

Görelî Konum

0 olarak ayarla

MsgFlags

MQMF_NONE olarak ayarla

OriginalLength

MQOL_UNDEFINED olarak ayarla

Kullanıcı verilerinin gönderilmesi

PCF yapıları, kullanıcı tanımlı ileti verileri göndermek için de kullanılabilir. Bu durumda, *Format* ileti tanımlayıcısı alanı MQFMT_PCF olarak ayarlanmalıdır.

Belirli bir kuyruқта PCF iletileri gönderilmesi ve alınması**PCF iletilerinin belirli bir kuyruğa gönderilmesi**

Belirli bir kuyruğa ileti göndermek için, mqPutBag çağrısı belirtilen torbanın içeriğini bir PCF iletisine dönüştürür ve iletiyi belirtilen kuyruğa gönderir. Aramadan sonra çantanın içindekiler değişmemiş.

Bu aramaya giriş olarak şunları sağlamanız gerekir:

- MQI bağlantı tanıtıcısı.
- İletin yerleştirileceği kuyruğa ilişkin nesne tanıtıcısı.
- Bir ileti tanımlayıcısı. İleti tanımlayıcıya ilişkin ek bilgi için [MQMD-Message descriptor](#) başlıklı konuya bakın.
- MQPMO yapısını kullanarak İleti Gönderme Seçenekleri. MQPMO yapısıyla ilgili daha fazla bilgi için [MQPMO-Put-message options](#) başlıklı konuya bakın.
- Bir iletiye dönüştürülecek torbanın tutamacı.

Not: Çanta bir yönetim iletisi içeriyorsa ve çantaya değer eklemek için mqAddSorgu çağrısı kullanıldıysa, MQIASY_COMMAND veri ögesinin değeri, MQAI tarafından tanınan bir INQUIRE komutu olmalıdır.

mqPutBag çağrısının tam açıklaması için bkz. [mqPutBag](#).

Belirli bir kuyruktan PCF iletileri alınması

Belirli bir kuyruktan ileti almak için, mqGetBag çağrısı belirli bir kuyruktan bir PCF ileti alır ve ileti verilerini veri torbasına dönüştürür.

Bu aramaya giriş olarak şunları sağlamanız gerekir:

- MQI bağlantı tanıtıcısı.
- İletin okunacağı kuyruğun nesne tanıtıcısı.
- Bir ileti tanımlayıcısı. MQMD yapısı içinde, **Format** değiştirgesi MQFMT_ADMIN, MQFMT_EVENT ya da MQFMT_PCF olmalıdır.

Not: İleti bir iş birimi içinde alınırsa (yani, MQGMO_SYNCPOINT seçeneğiyle) ve ileti desteklenmeyen bir biçime sahipse, iş birimi geriletilebilir. Daha sonra ileti kuyrukta yeniden kullanılır ve mqGetBag çağrısı yerine MQGET çağrısı kullanılarak alınabilir. İleti tanımlayıcıya ilişkin ek bilgi için [MQGMO-Get-message options](#) başlıklı konuya bakın.

- MQGMO yapısını kullanarak ileti seçeneklerini alın. MQGMO yapısıyla ilgili ek bilgi için [MQMD-Message Descriptor](#) başlıklı konuya bakın.
- Dönüştürülen iletiyi içerecek torbanın tutamacı.

mqGetBag çağrısının tam açıklaması için bkz. [mqGetBag](#).

IBM MQ içinde PCF yanıtları

Her komuta yanıt olarak, komut sunucusu bir ya da daha çok yanıt ileti oluşturur. Bir yanıt iletinin biçimi komut iletime benzer.

PCF üstbilgisi, yanıt olduğu komutla aynı komut tanıtıcısı değerine sahiptir (ayrıntılar için bkz. [MQCFH-PCF üstbilgisi](#)). İleti tanıtıcısı ve ilinti tanıtıcısı, isteğin rapor seçeneklerine göre ayarlanır.

Komut iletinin PCF üstbilgi tipi MQCFT_COMMAND ise, yalnızca standart yanıtlar üretilir. Bu komutlar, z/OS'daki tüm platformlarda desteklenir. Eski uygulamalar, z/OS üzerinde PCF ' yi desteklemez; IBM MQ Windows Explorer bu tür bir uygulamadır (ancak, IBM WebSphere MQ 6.0 ya da üstü IBM MQ Explorer z/OS üzerinde PCF ' yi destekler).

Komut iletinin PCF üstbilgisi tipi MQCFT_COMMAND_XR ise, genişletilmiş ya da standart yanıtlar üretilir. Bu komutlar z/OS ve diğer bazı platformlarda desteklenir. z/OS üzerinde yayınlanan komutlar yalnızca genişletilmiş yanıtlar oluşturur. Diğer platformlarda, her iki tip yanıt oluşturulabilir.

Tek bir komut soysal bir nesne adı belirtirse, eşleşen her nesne için kendi iletiminde ayrı bir yanıt döndürülür. Yanıt oluşturma işleminde, soysal adı olan tek bir komut birden çok komut olarak kabul edilir (MQCFC_LAST ya da MQCFC_NOT_LAST denetim alanı dışında). Ters durumda, bir komut ileti tek bir yanıt ileti oluşturur.

Bazı PCF yanıtları, istenmediğinde bile bir yapı döndürebilir. Bu yapı, yanıt tanımında gösterilir (Programlanabilir Komut Biçimleri Tanımları) Her zaman döndürüldüğü gibi. Bu yanıtlar için, verilerin uygulandığı nesneyi tanımlamak için yanıtta nesnelere adlandırmanın nedeni.

Bir yanıtla ilişkin ileti tanımlayıcı

Bir yanıt ileti, ileti tanımlayıcısında aşağıdaki alanları içerir:

MsgType

Bu alan MQMT_REPLY.

MsgId

Bu alan, kuyruk yöneticisi tarafından oluşturulur.

CorrelId

Bu alan, komut iletinin rapor seçeneklerine göre oluşturulur.

Biçim

Bu alan MQFMT_ADMIN 'dir.

Kodlama

MQENC_NATIVE olarak ayarlayın.

CodedCharSetId

MQCCSI_Q_MGR olarak ayarlayın.

Kalıcılık

Komut iletişindeki gibi.

Öncelik

Komut iletişindeki gibi.

Yanıt MQPMO_PASS_IDENTITY_CONTEXT ile üretilir.

Standart yanıtlar

Üstbilgi tipi MQCFT_COMMAND olan komut iletileri, standart yanıtlar üretilir. Bu komutlar, z/OS'daki tüm platformlarda desteklenir.

Üç tip standart yanıt vardır:

- Tamam yanıtı
- Hata yanıtı
- Veri yanıtı

Tamam yanıtı

Bu yanıt, *CompCode* alanı MQCC_OK ya da MQCC_WARNING olan bir komut biçimi üstbilgisiyle başlayan bir iletiden oluşur.

MQCC_OK için *Reason* , MQRC_NONE değeridir.

MQCC_WARNING için *Reason* , uyarının niteliğini gösterir. Bu durumda, komut biçimi üstbilgisini, bu neden koduna uygun bir ya da daha çok uyarı değiştirgeci yapısı izleyebilir.

Her iki durumda da, bir inquire komutu için aşağıdaki bölümlerde açıklandığı gibi daha fazla parametre yapısı izleyebilirsiniz.

Hata yanıtı

Komutta bir hata varsa, bir ya da daha çok hata yanıtı iletisi gönderilir (olağan durumda yalnızca tek bir yanıt iletisi içeren bir komut için bile birden çok hata yanıtı gönderilebilir). Bu hata yanıtı iletilerinde MQCFC_LAST ya da MQCFC_NOT_LAST uygun şekilde ayarlanmış.

Bu tür her ileti, *CompCode* değeri MQCC_FAILED olan bir yanıt biçimi üstbilgisiyle ve belirli bir hatayı tanımlayan bir *Reason* alanıyla başlar. Genel olarak, her ileti farklı bir hatayı açıklar. Ayrıca, her iletinin üstbilgiyi izleyen sıfır ya da bir (hiçbir zaman birden fazla) hata parametresi yapısı vardır. Bu değiştirgeci yapısı (varsa), aşağıdakilerden birini içeren bir *Parameter* alanı olan bir MQCFIN yapısıdır:

- MQIACF_PARAMETER_ID

Yapıdaki *Value* alanı, hatalı parametrenin parametre tanıtıcısıdır (örneğin, MQCA_Q_NAME).

- MQIACF_ERROR_ID

Bu değer, MQRC_UNEXPECTED_ERROR *Reason* değeriyle (komut biçimi üstbilgisinde) kullanılır. MQCFIN yapısındaki *Value* alanı, komut sunucusu tarafından alınan beklenmeyen neden kodudur.

- MQIACF_SELECTOR

Komutla birlikte gönderilen bir liste yapısı (MQCFIL) yinelenen bir seçici içeriyorsa ya da geçerli değilse bu değer oluşur. Komut biçimi üstbilgisindeki *Reason* alanı hatayı tanıtır ve MQCFIN yapısındaki *Value* alanı, hatalı komutun MQCFIL yapısındaki değiştirgeci değeridir.

- MQIACF_ERROR_OFFSET

Bu değer, Ping Channel komutunda bir veri karşılaştırma hatası olduğunda oluşur. Yapıdaki *Value* alanı, Ping Channel karşılaştırma hatasının görece konumudur.

- MQIA_CODED_CHAR_SET_ID

Bu değer, gelen PCF komut iletisinin ileti tanımlayıcısındaki kodlanmış karakter takımı tanıtıcısı hedef kuyruk yöneticisinininkiyle eşleşmediğinde ortaya çıkar. Yapıdaki *Value* alanı, kuyruk yöneticisinin kodlanmış karakter takımı tanıtıcısıdır.

Son (ya da tek) hata yanıtı iletisi, *CompCode* alanı MQCC_FAILED ve *Reason* alanı MQRCCF_COMMAND_FAILED olan bir özet yanıtıdır. Bu iletinin üstbilgiyi izleyen parametre yapısı yok.

Veri yanıtı

Bu yanıt, bir sorma komutuna OK (Tamam) yanıtından (daha önce açıklandığı gibi) oluşur. OK yanıtını, istenen verileri içeren ek yapılar izler; açıklamalar için bkz. Programlanır Komut Biçimleri Tanımları.

Uygulamalar, belirli bir sırada döndürülen bu ek parametre yapılarına bağlı olmamalıdır.

Genişletilmiş yanıtlar

z/OS üzerinde yayınlanan komutlar genişletilmiş yanıtlar oluşturur.

Üç tip genişletilmiş yanıt vardır:

- MQCFT_XR_MSG tipinde ileti yanıtı
- MQCFT_XR_ITEM tipinde öge yanıtı
- MQCFT_XR_SUMMARY tipiyle özet yanıt

Her komut bir ya da daha fazla yanıt kümesi oluşturabilir. Her yanıt kümesi, PCF üstbilgisinin *MsgSeqNumber* alanında sırayla 1 'den numaralandırılmış bir ya da daha fazla iletiden oluşur. Her bir kümedeki son (ya da yalnızca) yanıtın *Control* alanı MQCFC_LAST değerini içeriyor. Kümedeki diğer tüm yanıtlar için bu değer MQCFC_NOT_LAST değeridir.

Herhangi bir yanıt, *Parameter* alanının MQBACF_RESPONSE_SET olarak ayarlandığı bir ya da daha fazla isteğe bağlı MQCFBS yapısı içerebilir; bu değer bir yanıt kümesi tanıtıcısıdır. Tanıtıcılar benzersizdir ve yanıtı içeren yanıt kümesini tanımlar. Her yanıt kümesi için, bunu tanımlayan bir MQCFBS yapısı vardır.

Genişletilmiş yanıtların en az iki parametre yapısı vardır:

- *Parameter* alanı MQBACF_RESPONSE_ID olarak ayarlanmış bir MQCFBS yapısı. Bu alandaki değer, yanıtın ait olduğu yanıt kümesinin tanıtıcısıdır. İlk kümedeki tanıtıcı isteğe bağlı. Sonraki kümelerde, tanıtıcı daha önce bir MQBACF_RESPONSE_SET yapısında bildirilir.
- *Parameter* alanı MQCACF_RESPONSE_Q_MGR_NAME olarak ayarlanmış bir MQCFST yapısı; değer, yanıt kümesinin geldiği kuyruk yöneticisinin adıdır.

Birçok yanıtlar ek parametre yapılarına sahiptir ve bu yapılar aşağıdaki bölümlerde açıklanır.

MQCFC_LAST içeren bir yanıt bulununcaya kadar, bir kümede kaç yanıt olduğunu önceden saptayamazsınız. Herhangi bir küme, ek kümelerin oluşturulduğunu belirtmek için MQBACF_RESPONSE_SET yapılarını içerebileceğinden, önceden kaç yanıt kümesi olduğunu da belirleyemeyebilirsiniz.

Sorma komutlarına genişletilmiş yanıtlar

Sorgu komutları genellikle, belirtilen arama ölçütleriyle eşleşen her öge için bir öge yanıtı (tip MQCFT_XR_ITEM) oluşturur. Öge yanıtının üstbilgisinde MQCC_OK değerini içeren bir *CompCode* alanı ve MQRC_NONE değerini içeren bir *Reason* alanı var. Ayrıca, Programlanır Komut Biçimleri Tanımları başlığı altında açıklandığı gibi, ögeyi ve istenen özneliklerini tanımlayan diğer parametre yapılarını da içerir.

Bir öge hataysa, üstbilgideki *CompCode* alanı MQCC_FAILED değerine sahiptir ve *Reason* alanı belirli bir hatayı tanımlar. Ögeyi tanımlamak için ek parametre yapıları eklenmiştir.

Belirli Inquire komutları, öge yanıtlarına ek olarak genel (ada özgü değil) ileti yanıtları döndürebilir. Bu yanıtlar, MQCFT_XR_MSG tipinde bilgi verici ya da hata yanıtlarıdır.

Inquire komutu başarılı olursa, isteğe bağlı olarak, *CompCode* değeri MÖCC_OK ve *Reason* alan değeri MÖRC_NONE olan bir özet yanıt (MÖCFT_XR_SUMMARY yazın) olabilir.

Sorgu komutu başarısız olursa, öge yanıtları döndürülebilir ve isteğe bağlı olarak, *CompCode* değeri MÖCC_FAILED ve *Reason* alan değeri MÖRCCF_COMMAND_FAILED olan bir özet yanıt (MÖCFT_XSUMMARY tipi) olabilir.

Sorma dışında komutlara genişletilmiş yanıtlar

Başarılı komutlar, üstbilgideki *CompCode* alanının MÖCC_OK değerine sahip olduğu ve *Reason* alanının MÖRC_NONE değerine sahip olduğu ileti yanıtları oluşturur. Her zaman en az bir ileti vardır; bilgi verici (MÖCFT_XR_MSG) ya da özet (MÖCFT_XR_SUMMARY) olabilir. İsteğe bağlı olarak ek bilgi iletileri (MÖCFT_XR_MSG tipi) olabilir. Her bilgi iletilisi, komutla ilgili bilgileri içeren bir dizi ek parametre yapısı içerebilir; oluşabilecek yapılara ilişkin komut tanımlarına bakın.

Başarısız olan komutlar, üstbilgideki *CompCode* alanının MÖCC_FAILED değerine sahip olduğu hata iletilisi yanıtları (MÖCFT_XR_MSG tipi) oluşturur ve *Reason* alanı bu hatayı tanımlar. Her ileti, hatayla ilgili bilgi içeren bir dizi ek parametre yapısı içerebilir: Oluşabilecek yapılara ilişkin tek tek hata açıklamalarına bakın. Bilgi verici ileti yanıtları oluşturulabilir. İsteğe bağlı olarak, *CompCode* değeri MÖCC_FAILED olan bir özet yanıt (MÖCFT_XR_SUMMARY) ve MÖRCCF_COMMAND_FAILED *Reason* alan değeri olabilir.

CommandScope kullanan komutlara genişletilmiş yanıtlar

Bir komut **CommandScope** değiştirgesini kullanıyorsa ya da **CommandScope** değiştirgesini kullanan bir komutun oluşturulmasına neden oluyorsa, komutun alındığı kuyruk yöneticisinden bir ilk yanıt kümesi vardır. Daha sonra, komutun yönlendirildiği her kuyruk yöneticisi için ayrı bir yanıt kümesi (ya da kümeleri) oluşturulur (birden çok ayrı komut verilmiş gibi). Son olarak, giriş kuyruğu yöneticisinden genel bir özet yanıtı (MÖCFT_XR_SUMMARY tipi) içeren bir yanıt kümesi vardır. MÖCACF_RESPONSE_Q_MGR_NAME parametre yapısı, her bir küme oluşturan kuyruk yöneticisini tanımlar.


İlk yanıt kümesi aşağıdaki ek parametre yapılarını içerir:

- MÖIACF_COMMAND_INFO (MÖCFIN). Bu yapıdaki olası değerler şunlardır: MÖCMDI_CMDSCOPE_ACCEPTED ya da MÖCMDI_CMDSCOPE_GENERATED.
- MÖIACF_CMDSCOPE_Q_MGR_COUNT (MÖCFIN). Bu yapı, komutun gönderildiği kuyruk yöneticilerinin sayısını gösterir.

IBM MQ içinde PCF komutları için yetki denetimi

Bir PCF komutu işlendiğinde, gerekli IBM MQ nesne yetkisi denetimleri için komut iletilisindeki ileti tanımlayıcıdaki *UserIdentifier* kullanılır. Yetki denetimi, bu konuda açıklandığı gibi her platformda farklı şekilde gerçekleştirilir.

Denetimler, komutun işlenmekte olduğu sistemde gerçekleştirilir; bu nedenle, bu kullanıcı kimliği hedef sistemde bulunmalı ve komutu işlemek için gerekli yetkilere sahip olmalıdır. İleti uzak bir sistemden geliyorsa, hedef sistemde varolan tanıtıcıya ulaşmanın bir yolu, yerel ve uzak sistemlerde eşleşen bir kullanıcı kimliğine sahip olmaktır.

Not:  z/OS üzerinde yetki denetimine ilişkin bilgi için [Görev 1: z/OS sistem parametrelerinin tanımlanması](#) başlıklı konuya bakın.

IBM MQ for IBM i



Herhangi bir PCF komutunun işlenebilmesi için, kullanıcı kimliğinin hedef sistemdeki IBM MQ nesnesi için *dsp* yetkisi olmalıdır.

Ayrıca, IBM MQ nesne yetkisi denetimleri [Çizelge 2 sayfa 34](#) içinde gösterildiği gibi belirli PCF komutları için gerçekleştirilir.

Çoğu durumda bu denetimler, yerel bir sistemde yayınlanan eşdeğer IBM MQ CL komutları tarafından gerçekleştirilen denetimlerle aynıdır. IBM MQ yetkilerinin IBM i sistem yetkilileriyle eşlenmesine ve IBM MQ CL komutlarına ilişkin yetki gereksinmelerine ilişkin ek bilgi için [IBM i üzerinde güvenliğin ayarlanması](#) başlıklı konuya bakın. Çıktılarla ilgili güvenlik ayrıntıları [Güvenlik çıkışı](#) kullanılarak bağlantı düzeyi güvenliği belgesinde verilmiştir.

Aşağıdaki komutlardan herhangi birini işlemek için kullanıcı kimliği, QMQMADM grup tanıtımının bir üyesi olmalıdır:

- Ping Kanalı
- Kanalı Değiştir
- Kanalı Kopyala
- Kanal Oluştur
- Kanalı Sil
- Kanalı Sıfırla
- Kanalı Çözümle
- Kanalı Başlat
- Kanalı Durdur
- Kanal Başlatıcıyı Başlat
- Kanal Dinleyicisini Başlat

IBM MQ for UNIX, Linux, and Windows

ALW

Herhangi bir PCF komutunun işlenebilmesi için, kullanıcı kimliğinin hedef sistemdeki kuyruk yöneticisi nesnesi için *dsp* yetkisi olmalıdır. Ayrıca, IBM MQ nesne yetkisi denetimleri [Çizelge 2 sayfa 34](#) içinde gösterildiği gibi belirli PCF komutları için gerçekleştirilir.

Aşağıdaki komutlardan herhangi birini işlemek için kullanıcı kimliği *mqm* grubuna ait olmalıdır.

Not: Windows **yalnızca** için, kullanıcı kimliği *Yöneticiler* ya da *mqm* grubuna ait olabilir.

- Kanalı Değiştir
- Kanalı Kopyala
- Kanal Oluştur
- Kanalı Sil
- Ping Kanalı
- Kanalı Sıfırla
- Kanalı Başlat
- Kanalı Durdur
- Kanal Başlatıcıyı Başlat
- Kanal Dinleyicisini Başlat
- Kanalı Çözümle
- Kümeyi Sıfırla
- Kümeyi Yenile
- Kuyruk Yöneticisini Askıya Al
- Kuyruk Yöneticisini Sürdür

IBM MQ Multiplatforms için nesne yetkileri

Multi

<i>Çizelge 2. Nesne yetkileri</i>		
Komut	IBM MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
Kimlik Doğrulama Bilgilerini Değiştir	dsp ve chg	geçerli değil
Kanalı Değiştir	dsp ve chg	geçerli değil
Kanal Dinleyicisini Değiştir	dsp ve chg	geçerli değil
İstemci Bağlantı Kanalını Değiştir	dsp ve chg	geçerli değil
Ad melist değerini değiştir	dsp ve chg	geçerli değil
Süreci Değiştir	dsp ve chg	geçerli değil
Kuyruğu Değiştir	dsp ve chg	geçerli değil
Kuyruk Yöneticisini Değiştir	chg <i>bkz. Not 3 ve Not 5</i>	geçerli değil
Hizmeti Değiştir	dsp ve chg	geçerli değil
Kuyruğu Temizle	kr	geçerli değil
Kimlik Doğrulama Bilgilerini Kopyala	dsp	crt
Doğrulama Bilgilerini Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
Kanalı Kopyala	dsp	crt
Kanalı Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
Kanal Dinleyicisini Kopyala	dsp	crt
Kanal Dinleyicisini Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
İstemci Bağlantı Kanalını Kopyala	dsp	crt
İstemci Bağlantı Kanalını Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
Namelist 'i Kopyala	dsp	crt
Namelist 'i Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: dsp and chg</i>	crt
Süreci Kopyala	dsp	crt
İşlemi Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
Kuyruğu Kopyala	dsp	crt
Kuyruğu Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: dsp and chg</i>	crt
Kimlik Doğrulama Bilgileri Oluştur	<i>(sistem varsayılan kimlik doğrulama bilgileri) dsp</i>	crt
Doğrulama Bilgileri Oluştur (Değiştir) <i>bkz. Not 1</i>	<i>(sistem varsayılan kimlik doğrulama bilgileri) dsp to: chg</i>	crt
Kanal Oluştur	<i>(sistemin varsayılan kanalı) dsp</i>	crt

Çizelge 2. Nesne yetkileri (devamı var)

Komut	IBM MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
Kanal Oluştur (Değiştir) bkz. Not 1	(sistemin varsayılan kanalı) dsp to: chg	crt
Kanal Dinleyicisi Oluştur	(sistem varsayılan dinleyicisi) dsp	crt
Kanal Dinleyicisi Oluştur (Değiştir) bkz. Not 1	(sistem varsayılan dinleyicisi) dsp to: chg	crt
İstemci Bağlantı Kanalı Oluştur	(sistemin varsayılan kanalı) dsp	crt
İstemci Bağlantı Kanalı Oluştur (Değiştir) bkz. Not 1	(sistemin varsayılan kanalı) dsp to: chg	crt
Ad listesi oluştur	(sistem varsayılan ad listesi) dsp	crt
Ad listesi oluştur (Değiştir) bkz. Not 1	(sistem varsayılan ad listesi) dsp to: dsp and chg	crt
Süreç Yarat	(sistemin varsayılan işlemi) dsp	crt
İşlem Yarat (Değiştir) bkz. Not 1	(sistemin varsayılan işlemi) dsp to: chg	crt
Kuyruk Yarat	(sistemin varsayılan kuyruğu) dsp	crt
Kuyruk Yarat (Değiştir) bkz. Not 1	(sistemin varsayılan kuyruğu) dsp to: dsp and chg	crt
Hizmet oluştur	(sistemin varsayılan kuyruğu) dsp	crt
Hizmet Oluştur (Değiştir) bkz. Not 1	(sistemin varsayılan kuyruğu) dsp - chg	crt
Kimlik Doğrulama Bilgilerini Sil	dsp ve dlt	geçerli değil
Yetki Kaydını Sil	(kuyruk yöneticisi nesnesi) chg bkz. Not 4	bkz. Not 4
Kanalı Sil	dsp ve dlt	geçerli değil
Kanal Dinleyicisini Sil	dsp ve dlt	geçerli değil
İstemci Bağlantı Kanalını Sil	dsp ve dlt	geçerli değil
Ad alanını sil	dsp ve dlt	geçerli değil
Süreci Sil	dsp ve dlt	geçerli değil
Kuyruğu sil	dsp ve dlt	geçerli değil
Hizmeti Sil	dsp ve dlt	geçerli değil
Sorma Kimlik Doğrulama Bilgileri	dsp	geçerli değil
Yetki Kayıtlarını Sor	bkz. Not 4	bkz. Not 4
Sorma Kanalı	dsp	geçerli değil
Sorma Kanalı Dinleyicisi	dsp	geçerli değil
Inquire Channel Status (ChannelType MQCHT_CLSSDR için)	inq	geçerli değil
İstemci Bağlantı Kanalını Sor	dsp	geçerli değil

Çizelge 2. Nesne yetkileri (devamı var)

Komut	IBM MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
Namelist 'i sor	dsp	geçerli değil
Sorma Süreci	dsp	geçerli değil
Sorma Kuyruğu	dsp	geçerli değil
Sorma Kuyruğu Yöneticisi	<i>bkz. not 3</i>	geçerli değil
Sorma Kuyruğu Durumu	dsp	geçerli değil
Sorma Hizmeti	dsp	geçerli değil
Ping Kanalı	Ctrl	geçerli değil
Ping Kuyruğu Yöneticisi	<i>bkz. not 3</i>	geçerli değil
Kuyruk Yöneticisini Yenile	(kuyruk yöneticisi nesnesi) chg	geçerli değil
Güvenliği Yenile (SecurityType MQSECTYPE_SSL için)	(kuyruk yöneticisi nesnesi) chg	geçerli değil
Kanalı Sıfırla	ctrlx	geçerli değil
Kuyruk Yöneticisini İlk Durumuna Getir	(kuyruk yöneticisi nesnesi) chg	geçerli değil
Kuyruk İstatistiklerini Sıfırla	dsp ve chg	geçerli değil
Kanalı Çözümle	ctrlx	geçerli değil
Yetki Kaydını Ayarla	(kuyruk yöneticisi nesnesi) chg <i>bkz. Not 4</i>	<i>bkz. Not 4</i>
Kanalı Başlat	Ctrl	geçerli değil
Kanalı Durdur	Ctrl	geçerli değil
Bağlantıyı Durdur	(kuyruk yöneticisi nesnesi) chg	geçerli değil
Dinleyiciyi Başlat	Ctrl	geçerli değil
Dinleyiciyi Durdur	Ctrl	geçerli değil
Hizmeti Başlat	Ctrl	geçerli değil
Hizmeti Durdur	Ctrl	geçerli değil
Esc	<i>bkz. Not 2</i>	<i>bkz. Not 2</i>

Notlar:

1. Bu komut, değiştirilecek nesne varsa geçerlidir; tersi durumda, yetki denetimi Yarat ya da Başkasız Kopyala içindir.
2. Gereken yetki, çıkış metninin tanımladığı MQSC komutuyla belirlenir ve önceki komutlardan biriyle eşdeğerdir.
3. Herhangi bir PCF komutunun işlenebilmesi için, kullanıcı kimliğinin hedef sistemdeki kuyruk yöneticisi nesnesi için dsp yetkisi olmalıdır.
4. Komut sunucusu -a parametresiyle başlatılmadıysa, bu PCF komutuna yetki verilir. Varsayılan olarak, komut sunucusu kuyruk yöneticisi başlatıldığında ve -a parametresi olmadan başlar. Ek bilgi için bkz. [Programlanabilir komut biçimleri başvurusu](#).

5. Bir kuyruk yöneticisi için chg kullanıcı kimliği yetkisi verilmesi, tüm gruplar ve kullanıcılar için yetki kayıtları ayarlama yeteneği sağlar. Bu yetkiyi sıradan kullanıcılara ya da uygulamalara vermeyin.

IBM MQ , güvenlik denetimi için kendi kullanıcı çıkış programlarınızı sağlamanız için bazı kanal güvenliği çıkış noktaları da sağlar. Daha fazla bilgi için bkz. [Kanal görüntülenmesi](#).

Multi

PCF 'lerin kullanımını basitleştirmek için MQAI' nin kullanılması

IBM MQ Yönetim Arabirimi (MQAI), IBM MQ için AIX, IBM i, Linux, ve Windows üzerinde kullanılabilen bir programlama arabirimidir. IBM MQ kuyruk yöneticisinde, nesnelerin özelliklerini (ya da değiştirgelerini) Programlar Komut Biçimlerini (PCF) kullanmaktan daha kolay bir şekilde işlemek için veri torbalarını kullanarak denetim görevleri gerçekleştirir.

MQAI, *veri torbaları* kullanılarak bir kuyruk yöneticisinde denetim görevlerini gerçekleştirir. Veri torbaları, nesnelerin özelliklerini (ya da parametrelerini) PCF ' leri kullanmaktan daha kolay bir şekilde işlemeyi sağlar.

MQAI ' yi kullanmanın avantajları şunlardır:

PCF iletilerinin kullanımını basitleştirin

MQAI, IBM MQ' yi yönetmenin daha kolay bir yoludur. MQAI kullanıyorsanız, kendi PCF iletilerinizi yazmanız gerekmez. Bu, karmaşık veri yapılarıyla ilişkili sorunları önler.

MQI çağrılarını kullanarak yazılan programlarda değiştirge geçirmek için, PCF iletilerinin komutu ve dizgi ya da tamsayı verilerinin ayrıntılarını içermesi gerekir. Bu yapılandırmayı el ile oluşturmak için, her yapı için programınıza birkaç deyim eklemeniz ve bellek alanı ayırmanız gerekir. Bu görev uzun ve zahmetli olabilir.

MQAI kullanılarak yazılan programlar, parametreleri uygun veri çantasına geçirir ve her yapı için yalnızca bir deyim gereksinim duyarsınız. MQAI veri torbalarının kullanılması, dizileri işleme ve depolama alanı ayırma gereksinimlerinizi ortadan kaldırır ve PCF ' nin ayrıntılarından bir dereceye kadar yalıtım sağlar.

Hata koşullarını daha kolay işleme

PCF komutlarından dönüş kodlarını geri almak zordur. MQAI, programın hata koşullarını işlemesini kolaylaştırır.

Uygulamalar arasında veri alışverişi

Uygulama verileri PCF biçiminde gönderilir ve MQAI tarafından paketlenir ve paketinden çıkarılır. İletilerinizin tamsayılar ve karakter dizgilerinden oluşuyorsa, PCF verilerine ilişkin IBM MQ yerleşik veri dönüştürmesinden yararlanmak için MQAI ' yi kullanabilirsiniz. Bu, veri dönüştürme çıkışlarının yazılması gereksinimini ortadan kaldırır.

Veri torbanızı oluşturup doldurduktan sonra, mqExecute çağrısını kullanarak bir kuyruk yöneticisinin komut sunucusuna bir denetim komut iletileri gönderebilirsiniz. Bu çağrı, yanıt iletileri için bekler. mqExecute çağrısı, komut sunucusuyla değiş tokuş işler ve yanıtları bir *yanıt çantası*na döndürür.

MQAI kullanımı örnekleri

Aşağıdaki örnek programlar, çeşitli görevleri gerçekleştirmek için MQAI kullanımını göstermektedir:

- [amqsaicq.c](#): Yerel bir kuyruk oluşturun.
- [amqsailem.c](#): Basit bir olay izleyicisini kullanarak ekrandaki olayları görüntüler.
- [amqsailq.c](#): Tüm yerel kuyrukların ve bunların yürürlükteki derinliklerinin bir listesini yazdırın.
- [amqsaicl.c](#): Tüm kanalların ve türlerinin bir listesini yazdırın.

MQAI uygulamanızı oluşturma

Uygulamanızı MQAI kullanarak oluşturmak için, IBM MQ ile aynı kitaplıklara bağlanın. IBM MQ uygulamalarınızı nasıl oluşturacağınıza ilişkin bilgi için bkz. [Prosedür uygulaması oluşturma](#).

IBM MQ ' in MQAI kullanılarak yapılandırılmasına ilişkin ipuçları ve öneriler

MQAI, komut sunucusunun kendisiyle doğrudan ilgilenmek yerine, komut sunucusuna denetim komutları göndermek için PCF iletilerini kullanır. IBM MQ ' i MQAI kullanarak yapılandırmaya ilişkin ipuçları [“IBM MQ ' i yapılandırmak için MQAI kullanımına ilişkin ipuçları ve öneriler” sayfa 38](#) içinde bulunabilir.

İlgili başvurular

[IBM MQ Yönetim Arabirimi başvurusu](#)

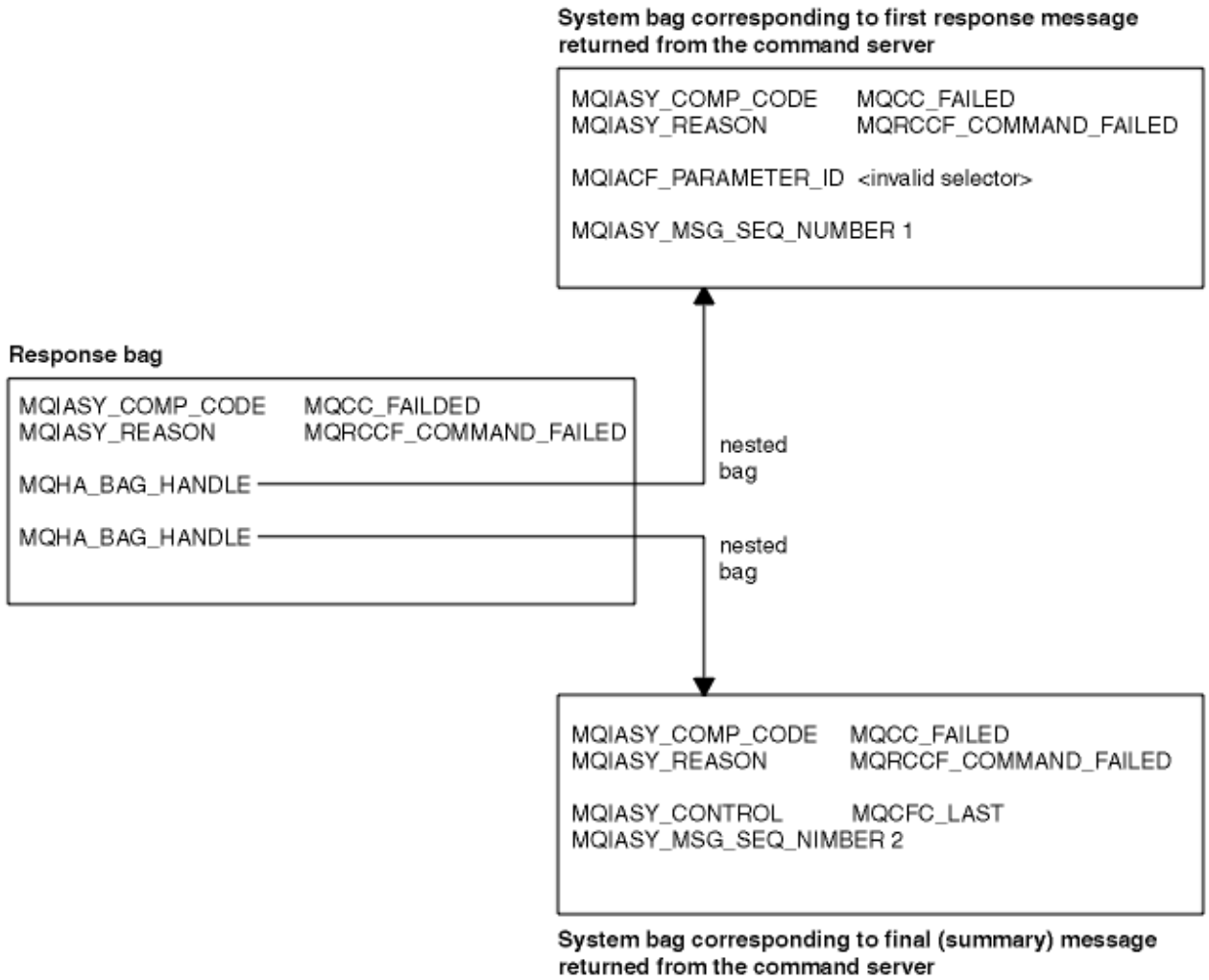
Multi IBM MQ ' i yapılandırmak için MQAI kullanımına ilişkin ipuçları ve öneriler

IBM MQ Denetim Arabirimi (MQAI), komut sunucusunun kendisiyle doğrudan ilgilenmek yerine, komut sunucusuna denetim komutları göndermek için PCF iletilerini kullanır. Aşağıda, MQAI kullanılarak IBM MQ yapılandırması için bazı ipuçları yer alır.

- IBM MQ içindeki karakter dizgileri, sabit uzunluğa kadar doldurulur. C kullanılarak, boş sonlandırılmış dizgiler normalde IBM MQ programlama arabirimlerine giriş parametreleri olarak sağlanabilir.
- Bir dizgi özniteliğinin değerini temizlemek için, boş bir dizgi yerine tek bir boşluk karakterine ayarlayın.
- Değiştirmek istediğiniz öznitelikleri önceden göz önünde bulundurun ve yalnızca bu öznitelikleri sorgulayın.
- Kuyruk adı ya da kanal tipi gibi belirli öznitelikler değiştirilemez. Yalnızca değiştirilebilecek öznitelikleri değiştirmeye çalıştığınızdan emin olun. Belirli bir PCF değişiklik nesnesine ilişkin gerekli ve isteğe bağlı parametrelerin listesine bakın. Bkz. [Programlanabilir Komut Biçimleri Tanımları](#).
- Bir MQAI çağrısı başarısız olursa, hatanın bazı ayrıntıları yanıt çantasına döndürülür. Daha sonra, seçici MQHA_BAG_HANDLE tarafından erişilebilen içiçe yerleştirilmiş bir çantada daha fazla ayrıntı bulunabilir. Örneğin, bir mqExecute çağrısı MQRCCF_COMMAND_FAILED neden koduyla başarısız olursa, bu bilgi yanıt çantasında döndürülür. Bu neden kodunun olası bir nedeni, belirtilen seçicinin komut iletisi tipi için geçerli olmamasıydı ve bu bilgi ayrıntısı, bir torba tutamacı tarafından erişilebilen içiçe yerleştirilmiş bir çantada bulundu.

MQExecute hakkında daha fazla bilgi için bkz. [“mqExecute çağrısını kullanarak qm komut sunucusuna yönetim komutları gönderme” sayfa 71](#)

Aşağıdaki çizge bu senaryoyu göstermektedir:



Multi Gelişmiş MQAI konuları

Dizinleme, veri dönüştürme ve ileti tanımlayıcısının kullanımına ilişkin bilgiler

Dizinleme

Dizinler, ekleme sırasını korumak için bir torbadaki var olan veri öğeleri değiştirilirken ya da kaldırılırken kullanılır.

Veri dönüştürme

Bir MQAI veri torbasında bulunan dizgiler çeşitli kodlanmış karakter kümelerinde olabilir ve bunlar `mqSetInteger` çağrısı kullanılarak dönüştürülebilir.

İleti tanımlayıcısının kullanılması

MQAI, veri torbası yaratıldığında ilk değere ayarlanan bir ileti tanımlayıcısı oluşturur.

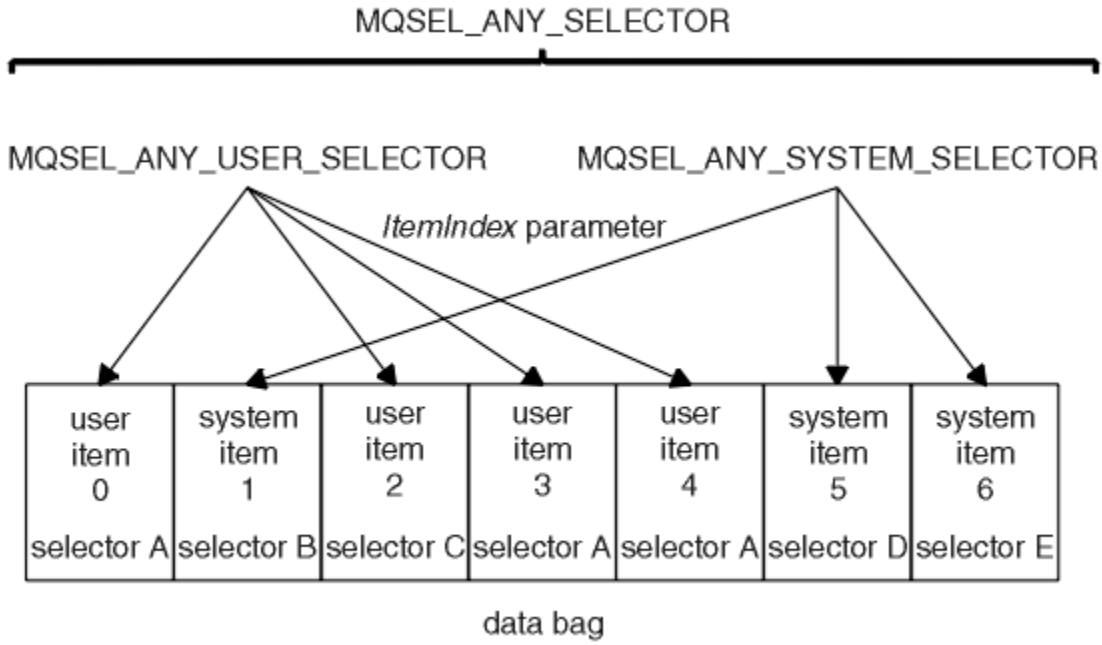
Multi MQAI içinde dinleme

Dizinler, bir torbadaki var olan veri öğeleri değiştirilirken ya da kaldırılırken kullanılır. Veri öğelerinin kolayca alınmasına olanak sağlayan üç tip dizin oluşturma vardır.

Bir torbadaki bir veri öğesi içindeki her seçici ve değer ilişkili üç dizin numarası vardır:

- Aynı seçiciye sahip diğer öğelerle görelî dizin.
- Öğenin ait olduğu seçici (kullanıcı ya da sistem) kategorisiyle görelî dizin.
- Paketteki tüm veri öğeleriyle (kullanıcı ve sistem) görelî dizin.

Bu, Şekil 3 sayfa 40 içinde gösterildiği gibi kullanıcı seçiciler, sistem seçiciler ya da her ikisi tarafından dinlenmeye izin verir.



Şekil 3. Dizinleme

Şekil 3 sayfa 40’inde, kullanıcı ögesi 3 'e (seçici A) aşağıdaki izin çiftleri tarafından başvuruda bulunulabilir:

- seçici A (ItemIndex 1)
- MQSEL_ANY_USER_SELECTOR (ItemIndex 2)
- MQSEL_ANY_SELECTOR (itemIndex 3)

Dizin, C' deki bir dizi gibi sıfıra dayalıdır; 'n' oluşum varsa, dizin aralığı sıfır ile'n-1' arasındadır ve boşluk yoktur.

Dizinler, bir torbadaki var olan veri öğeleri değiştirilirken ya da kaldırılırken kullanılır. Bu şekilde kullanıldığında ekleme sırası korunur, ancak diğer veri öğelerinin izinleri etkilenebilir. Bunun örnekleri için bkz. “Torba içindeki bilgilerin değiştirilmesi” sayfa 68 ve “Veri öğelerini silme” sayfa 70.

Üç tip izin oluşturma, veri öğelerinin kolayca alınmasını sağlar. Örneğin, bir çantada belirli bir seçicinin üç örneği varsa, mqCountÖğe çağrısı o seçicinin eşgörünümünün sayısını sayabilir ve mqInquire* çağrılarını yalnızca bu değerleri sorgulamak için hem seçiciyi hem de dizini belirtebilir. Bu, kanallardaki bazı çıkışlar gibi bir değer listesine sahip olabilecek öznitelikler için yararlıdır.

Multi MQAI içinde veri dönüştürme işlemi

MQAI veri çantasında bulunan dizgiler çeşitli kodlanmış karakter kümelerinde olabilir. Bu dizgiler, mqSetTamsayı çağrısı kullanılarak dönüştürülebilir.

PCF iletilerinde olduğu gibi, MQAI veri torbasında bulunan dizgiler de çeşitli kodlanmış karakter kümelerinde olabilir. Genellikle, bir PCF iletilisindeki tüm dizgiler aynı kodlanmış karakter takımındaki, yani kuyruk yöneticisiyle aynı kümededir.

Bir veri paketindeki her dizgi ögesi iki değer içerir; dizginin kendisi ve CCSID. Çantaya eklenen dizgi, mqAddString ya da mqSetString çağrısının **Buffer** değiştirgesinden alınır. CCSID, MQIASY_CODED_CHAR_SET_ID seçicisini içeren sistem ögesinden alınır. Bu, çanta CCSID olarak bilinir ve mqSetInteger çağrısı kullanılarak değiştirilebilir.

Bir veri torbasında bulunan bir dizginin değerini sorduğunuzda, CCSID, çağrıdaki bir çıkış parametresidir.

Çizelge 3 sayfa 41 , veri torbalarını iletilere dönüştürürken uygulanan kuralları gösterir; tersi de geçerlidir:

Çizelge 3. CCSID işleme

MQAI çağrısı	CCSID	Aranacak giriş	Çağrıya çıkış
mqBagToBuffer	Çanta CCSID (<u>1</u>)	Yoksayıldı	Değiştirilmedi
mqBagToBuffer	Torbadaki dizgi CCSID 'leri	Kullanılan	Değiştirilmedi
mqBagToBuffer	Arabellekteki dizgi CCSID 'leri	Geçerli değildir	Torbadaki dizgi CCSID 'lerinden kopyalandı
mqBufferToBag	Çanta CCSID (<u>1</u>)	Yoksayıldı	Değiştirilmedi
mqBufferToBag	Arabellekteki dizgi CCSID 'leri	Kullanılan	Değiştirilmedi
mqBufferToBagmqBufferToBag	Torbadaki dizgi CCSID 'leri	Geçerli değildir	Arabellekteki dizgi CCSID 'lerinden kopyalandı
mqPutTorba	MQMD CCSID 'si	Kullanılan	Değiştirilmedi (<u>2</u>)
mqPutTorba	Çanta CCSID (<u>1</u>)	Yoksayıldı	Değiştirilmedi
mqPutTorba	Torbadaki dizgi CCSID 'leri	Kullanılan	Değiştirilmedi
mqPutTorba	İletideki dizgi CCSID 'leri gönderildi	Geçerli değildir	Torbadaki dizgi CCSID 'lerinden kopyalandı
mqGetTorbası	MQMD CCSID 'si	İletin veri dönüşümü için kullanılır	Döndürülen verilerin CCSID 'sine ayarla (<u>3</u>)
mqGetTorbası	Çanta CCSID (<u>1</u>)	Yoksayıldı	Değiştirilmedi
mqGetTorbası	İletideki dizgi CCSID 'leri	Kullanılan	Değiştirilmedi
mqGetTorbası	Torbadaki dizgi CCSID 'leri	Geçerli değildir	İletideki dizgi CCSID 'lerinden kopyalandı
mqExecute	İstek çantası CCSID 'si	İstek iletisine ilişkin MQMD için kullanılır (<u>4</u>)	Değiştirilmedi
mqExecute	Yanıt çantası CCSID 'si	Yanıt iletisinin veri dönüşümü için kullanılır (<u>4</u>)	Döndürülen verilerin CCSID 'sine ayarla (<u>3</u>)
mqExecute	İstek çantasındaki dizgi CCSID 'leri	İstek iletisi için kullanılır	Değiştirilmedi
mqExecute	Yanıt çantasındaki dizgi CCSID 'leri	Geçerli değildir	Yanıt iletisindeki dizgi CCSID 'lerinden kopyalandı

Notlar:

1. Bag CCSID, seçici MQIASY_CODED_CHAR_SET_ID ' ye sahip sistem ögesidir.
2. MQCCSI_Q_MGR, gerçek kuyruk yöneticisi CCSID 'sine çevrilir.
3. Veri dönüştürme istenirse, döndürülen verilerin CCSID 'si çıkış değeriyle aynıdır. Veri d "n"ytrme isteşi sunulmazsa, d" ndrlecek verilerin CCSID 'si, ileti deşeri ile ayndr. Veri dönüştürme istenir, ancak başarısız olursa ileti döndürülmez.
4. CCSID MQCCSI_DEFAULT ise, kuyruk yöneticisinin CCSID 'si kullanılır.

İlgili kavramlar

“Kodlanmış karakter takımları arasında veri dönüştürme” sayfa 220

IBM MQ tanımlı biçimlerdeki (yerleşik biçimler olarak da bilinir) ileti verileri, her iki karakter kümesinin tek bir dille ya da benzer dillerden oluşan bir grupta ilgili olması koşuluyla, kuyruk yöneticisi tarafından bir kodlanmış karakter kümesinden diğerine dönüştürülebilir.

“ccsid_part2.tbl dosyası” sayfa 222

ccsid_part2.tbl dosyası, ek CCSID bilgileri sağlamak için kullanılır. ccsid_part2.tbl dosyası, IBM MQ 9.0' den önce kullanılan ccsid.tbl dosyasının yerini alır.

Multi MQAI ' de ileti tanımlayıcısının kullanılması

MQAI ' nın oluşturduğu ileti tanımlayıcısı, veri torbası oluşturulduğunda ilk değere ayarlanır.

PCF komut tipi, seçici MQIASY_TYPE ile sistem ögesinden elde edilir. Veri torbanızı oluşturduğunuzda, bu ögenin ilk değeri oluşturduğunuz torba tipine bağlı olarak ayarlanır:

Çizelge 4. PCF komutu tipi	
Torba tipi	MQIASY_TYPE ögesinin ilk değeri
MQCBO_ADMIN_BAG	MQCFT_COMMAND
MQCBO_COMMAND_BAG	MQCFT_COMMAND
MQCBO_*	MQCFT_USER

MQAI bir ileti tanımlayıcısı oluşturduğunda, **Format** ve **MsgType** değiştirgelerinde kullanılan değerler, Çizelge 4 sayfa 42 içinde gösterildiği gibi seçici MQIASY_TYPE ile sistem ögesinin değerine bağlıdır.

Çizelge 5. MQMD ' nin biçim ve MsgType değıştirgeleri		
PCF komutu tipi	Biçim	MsgType
MQCFT_COMMAND	MQFMT_ADMIN	MQMT_REQUEST
MQCFT_REPORT	MQFMT_ADMIN	MQMT_REPORT
MQCFT_YANITI	MQFMT_ADMIN	MQMT_REPLY
MQCFT_TRACE_ROUTE	MQFMT_ADMIN	MQMT_DATAGRAM
MQCFT_EVENT	MQFMT_EVENT	MQMT_DATAGRAM
MQCFT_*	MQFMT_PCF	MQMT_DATAGRAM

Çizelge 5 sayfa 42 , bir yönetim çantası ya da komut çantası yaratırsanız, ileti tanımlayıcısının *Format* MQFMT_ADMIN ve *MsgType* MQMT_REQUEST olduğunu gösterir. Bu, bir yanıt beklendiğinde komut sunucusuna gönderilen bir PCF istek iletisi için uygundur.

İleti tanımlayıcıdaki diğer parametreler Çizelge 6 sayfa 42 içinde gösterilen değerleri alır.

Çizelge 6. İleti tanımlayıcı değeri	
Değiştirge	Değer
<i>StrucId</i>	MQMD_STRUC_ID
<i>Version</i>	MQMD_VERSION_1
<i>Report</i>	MQRO_NONE
<i>MsgType</i>	Bakınız Çizelge 5 sayfa 42
<i>Expiry</i>	30 saniye (not “1” sayfa 43)
<i>Feedback</i>	MQFB_NONE

Çizelge 6. İleti tanımlayıcı değerleri (devamı var)	
Değiştirge	Değer
Encoding	MQENC_NATIVE
CodedCharSetId	Çanta CCSID 'sine bağlıdır (not “2” sayfa 43)
Format	Bakınız Çizelge 5 sayfa 42
Priority	MQPRI_PRIORITY_AS_Q_DEF
Persistence	MQPER_NOT_KALICI
MsgId	MQMI_NONE
CorrelId	MQCI_NONE
BackoutCount	0
ReplyToQ	Bkz. not “3” sayfa 43
ReplyToQMgr	blank (boşluk)

Notlar:

1. Bu değer, **OptionsBag** parametresi kullanılarak mqExecute çağrısında geçersiz kılınabilir. Bununla ilgili bilgi için bkz. mqExecute.
2. Bkz. “MQAI içinde veri dönüştürme işlemi” sayfa 40.
3. MQMT_REQUEST tipindeki iletiler için, kullanıcı tarafından belirtilen yanıt kuyruğunun ya da MQAI tarafından oluşturulan geçici dinamik kuyruğun adı. Ters durumda boş bırakın.

Multi Yerel kuyruk yaratmak için örnek C programı (amqsaicq.c)

Örnek C programı amqsaicq.c , MQAI kullanarak yerel bir kuyruk yaratır.

```

/*****
/*
/* Program name: AMQSAICQ.C
/*
/* Description: Sample C program to create a local queue using the
/* IBM MQ Administration Interface (MQAI).
/*
/* Statement: Licensed Materials - Property of IBM
/*
/* 84H2000, 5765-B73
/* 84H2001, 5639-B42
/* 84H2002, 5765-B74
/* 84H2003, 5765-B75
/* 84H2004, 5639-B43
/*
/* (C) Copyright IBM Corp. 1999, 2024
/*
/*****
/*
/* Function:
/* AMQSAICQ is a sample C program that creates a local queue and is an
/* example of the use of the mqExecute call.
/*
/* - The name of the queue to be created is a parameter to the program.
/*
/* - A PCF command is built by placing items into an MQAI bag.
/* These are:-
/* - The name of the queue
/* - The type of queue required, which, in this case, is local.
/*
/* - The mqExecute call is executed with the command MQCMD_CREATE_Q.
/* The call generates the correct PCF structure.
/* The call receives the reply from the command server and formats into
/* the response bag.
/*
/*****/

```

```

/*      - The completion code from the mqExecute call is checked and if there */
/*      is a failure from the command server then the code returned by the */
/*      command server is retrieved from the system bag that is */
/*      embedded in the response bag to the mqExecute call. */
/*      */
/* Note: The command server must be running. */
/*      */
/*      */

/*****
/*
/* AMQSAICQ has 2 parameters - the name of the local queue to be created */
/* - the queue manager name (optional) */
/*
*****/
/* Includes */
*****/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h>          /* MQI          */
#include <cmqcfc.h>       /* PCF         */
#include <cmqbc.h>       /* MQAI        */

void CheckCallResult(MQCHAR *, MQLONG , MQLONG );
void CreateLocalQueue(MQHCONN, MQCHAR *);

int main(int argc, char *argv[])
{
    MQHCONN hConn;          /* handle to IBM MQ connection */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG connReason;     /* MQCONN reason code */
    MQLONG compCode;       /* completion code */
    MQLONG reason;         /* reason code */

    /*****
    /* First check the required parameters */
    *****/
    printf("Sample Program to Create a Local Queue\n");
    if (argc < 2)
    {
        printf("Required parameter missing - local queue name\n");
        exit(99);
    }

    /*****
    /* Connect to the queue manager */
    *****/
    if (argc > 2)
        strncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
        MQCONN(QMName, &hConn, &compCode, &connReason);

    /*****
    /* Report reason and stop if connection failed */
    *****/
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("MQCONN", compCode, connReason);
        exit( (int)connReason);
    }

    /*****
    /* Call the routine to create a local queue, passing the handle to the */
    /* queue manager and also passing the name of the queue to be created. */
    *****/
    CreateLocalQueue(hConn, argv[1]);

    /*****
    /* Disconnect from the queue manager if not already connected */
    *****/
    if (connReason != MQRC_ALREADY_CONNECTED)
    {
        MQDISC(&hConn, &compCode, &reason);
        CheckCallResult("MQDISC", compCode, reason);
    }
    return 0;
}

```

```

/*****
/*
/* Function:      CreateLocalQueue
/* Description:  Create a local queue by sending a PCF command to the command
/*               server.
/*
/*
/*****
/*
/* Input Parameters:  Handle to the queue manager
/*                   Name of the queue to be created
/*
/*
/* Output Parameters: None
/*
/*
/* Logic: The mqExecute call is executed with the command MQCMD_CREATE_Q.
/*        The call generates the correct PCF structure.
/*        The default options to the call are used so that the command is sent
/*        to the SYSTEM.ADMIN.COMMAND.QUEUE.
/*        The reply from the command server is placed on a temporary dynamic
/*        queue.
/*        The reply is read from the temporary queue and formatted into the
/*        response bag.
/*
/*        The completion code from the mqExecute call is checked and if there
/*        is a failure from the command server then the code returned by the
/*        command server is retrieved from the system bag that is
/*        embedded in the response bag to the mqExecute call.
/*
/*
/*****
void CreateLocalQueue(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG reason;                /* reason code
    MQLONG compCode;              /* completion code
    MQHBAG commandBag = MQHB_UNUSABLE_HBAG; /* command bag for mqExecute
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute
    MQHBAG resultBag;             /* result bag from mqExecute
    MQLONG mqExecuteCC;           /* mqExecute completion code
    MQLONG mqExecuteRC;           /* mqExecute reason code

    printf("\nCreating Local Queue %s\n\n", qName);

    /*****
    /* Create a command Bag for the mqExecute call. Exit the function if the
    /* create fails.
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &commandBag, &compCode, &reason);
    CheckCallResult("Create the command bag", compCode, reason);
    if (compCode !=MQCC_OK)
        return;

    /*****
    /* Create a response Bag for the mqExecute call, exit the function if the
    /* create fails.
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);
    CheckCallResult("Create the response bag", compCode, reason);
    if (compCode !=MQCC_OK)
        return;

    /*****
    /* Put the name of the queue to be created into the command bag. This will
    /* be used by the mqExecute call.
    /*****
    mqAddString(commandBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, qName, &compCode,
                &reason);
    CheckCallResult("Add q name to command bag", compCode, reason);

    /*****
    /* Put queue type of local into the command bag. This will be used by the
    /* mqExecute call.
    /*****
    mqAddInteger(commandBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
    CheckCallResult("Add q type to command bag", compCode, reason);

    /*****
    /* Send the command to create the required local queue.
    /* The mqExecute call will create the PCF structure required, send it to
    /* the command server and receive the reply from the command server into
    /* the response bag.
    /*****
    mqExecute(hConn,                /* IBM MQ connection handle
              MQCMD_CREATE_Q,        /* Command to be executed

```

```

        MQHB_NONE,          /* No options bag          */
        commandBag,        /* Handle to bag containing commands */
        responseBag,      /* Handle to bag to receive the response*/
        MQHO_NONE,        /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE*/
        MQHO_NONE,        /* Create a dynamic q for the response */
        &compCode,        /* Completion code from the mqExecute */
        &reason);        /* Reason code from mqExecute call */

if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName>\n")
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("MQDISC", compCode, reason);
    exit(98);
}

/*****
/* Check the result from mqExecute call and find the error if it failed. */
*****/
if ( compCode == MQCC_OK )
    printf("Local queue %s successfully created\n", qName);
else
{
    printf("Creation of local queue %s failed: Completion Code = %d\n",
           qName, compCode, reason);
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        /*****
        /* Get the system bag handle out of the mqExecute response bag. */
        /* This bag contains the reason from the command server why the */
        /* command failed. */
        *****/
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &resultBag, &compCode,
                    &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */
        *****/
        mqInquireInteger(resultBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                        &compCode, &reason);
        CheckCallResult("Get the completion code from the result bag",
                        compCode, reason);
        mqInquireInteger(resultBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                        &compCode, &reason);
        CheckCallResult("Get the reason code from the result bag", compCode,
                        reason);
        printf("Error returned by the command server: Completion code = %d :
              Reason = %d\n", mqExecuteCC, mqExecuteRC);
    }
}

/*****
/* Delete the command bag if successfully created. */
*****/
if (commandBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&commandBag, &compCode, &reason);
    CheckCallResult("Delete the command bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created. */
*****/
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}
} /* end of CreateLocalQueue */

/*****
/*
/* Function: CheckCallResult
/*
*****/
/*
/* Input Parameters: Description of call
/* Completion code
/* Reason code
*/

```

```

/*                                                                    */
/* Output Parameters: None                                           */
/*                                                                    */
/* Logic: Display the description of the call, the completion code and the */
/*        reason code if the completion code is not successful        */
/*                                                                    */
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d :
              Reason = %d\n", callText, cc, rc);
}
}

```

Multi Olay izleyicisini kullanarak olayları görüntülemek için örnek C programı (amqsaiem.c)

Örnek C programı amqsaiem.c , MQAI kullanılarak temel bir olay izleyicisini gösterir.

```

*****/
/*                                                                    */
/* Program name: AMQSAIEM.C                                           */
/*                                                                    */
/* Description: Sample C program to demonstrate a basic event monitor */
/*              using the IBM MQ Admin Interface (MQAI).              */
/* Licensed Materials - Property of IBM                               */
/*                                                                    */
/* 63H9336                                                            */
/* (c) Copyright IBM Corp. 1999, 2024. All Rights Reserved.         */
/*                                                                    */
/* US Government Users Restricted Rights - Use, duplication or      */
/* disclosure restricted by GSA ADP Schedule Contract with           */
/* IBM Corp.                                                          */
/*****
/*
/* Function:
/* AMQSAIEM is a sample C program that demonstrates how to write a simple */
/* event monitor using the mqGetBag call and other MQAI calls.          */
/*                                                                    */
/* The name of the event queue to be monitored is passed as a parameter */
/* to the program. This would usually be one of the system event queues:- */
/* SYSTEM.ADMIN.QMGR.EVENT      Queue Manager events                  */
/* SYSTEM.ADMIN.PERFM.EVENT     Performance events                   */
/* SYSTEM.ADMIN.CHANNEL.EVENT   Channel events                       */
/* SYSTEM.ADMIN.LOGGER.EVENT    Logger events                        */
/*                                                                    */
/* To monitor the queue manager event queue or the performance event queue, */
/* the attributes of the queue manager need to be changed to enable */
/* these events. For more information about this, see Part 1 of the */
/* Programmable System Management book. The queue manager attributes can */
/* be changed using either MQSC commands or the MQAI interface.      */
/* Channel events are enabled by default.                             */
/*                                                                    */
/* Program logic
/* Connect to the Queue Manager.                                     */
/* Open the requested event queue with a wait interval of 30 seconds. */
/* Wait for a message, and when it arrives get the message from the queue */
/* and format it into an MQAI bag using the mqGetBag call.           */
/* There are many types of event messages and it is beyond the scope of */
/* this sample to program for all event messages. Instead the program */
/* prints out the contents of the formatted bag.                     */
/* Loop around to wait for another message until either there is an error */
/* or the wait interval of 30 seconds is reached.                    */
/*                                                                    */
/*****
/*
/* AMQSAIEM has 2 parameters - the name of the event queue to be monitored */
/* - the queue manager name (optional)                                     */
/*                                                                    */
/*****
/*****
/* Includes                                                            */
/*****
#include <stdio.h>
#include <string.h>

```

```

#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h> /* MQI */
#include <cmqcfc.h> /* PCF */
#include <cmqbc.h> /* MQAI */

/*****
*/
/* Macros
*/
/*****
*/
#if MQAT_DEFAULT == MQAT_WINDOWS_NT
#define Int64 "I64"
#elif defined(MQ_64_BIT)
#define Int64 "l"
#else
#define Int64 "ll"
#endif

/*****
*/
/* Function prototypes
*/
/*****
*/
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);
void GetQEvents(MQHCONN, MQCHAR *);
int PrintBag(MQHBAG);
int PrintBagContents(MQHBAG, int);

/*****
*/
/* Function: main
*/
/*****
*/
int main(int argc, char *argv[])
{
    MQHCONN hConn; /* handle to connection */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QM name */
    MQLONG reason; /* reason code */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */

    /*****
    */
    /* First check the required parameters
    */
    /*****
    */
    printf("Sample Event Monitor (times out after 30 secs)\n");
    if (argc < 2)
    {
        printf("Required parameter missing - event queue to be monitored\n");
        exit(99);
    }

    /*****
    */
    /* Connect to the queue manager
    */
    /*****
    */
    if (argc > 2)
        stncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(QMName, &hConn, &compCode, &connReason);
    /*****
    */
    /* Report the reason and stop if the connection failed
    */
    /*****
    */
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("MQCONN", compCode, connReason);
        exit( (int)connReason);
    }

    /*****
    */
    /* Call the routine to open the event queue and format any event messages
    */
    /* read from the queue.
    */
    /*****
    */
    GetQEvents(hConn, argv[1]);

    /*****
    */
    /* Disconnect from the queue manager if not already connected
    */
    /*****
    */
    if (connReason != MQRC_ALREADY_CONNECTED)
    {
        MQDISC(&hConn, &compCode, &reason);
        CheckCallResult("MQDISC", compCode, reason);
    }

    return 0;
}

/*****
*/

```



```

/*
/* Function: CheckCallResult
/*
/*
/*****
/*
/* Input Parameters:  Description of call
/*                    Completion code
/*                    Reason code
/*
/* Output Parameters: None
/*
/* Logic: Display the description of the call, the completion code and the
/*        reason code if the completion code is not successful
/*
/*
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d : Reason = %d\n",
            callText, cc, rc);
}

/*****
/*
/* Function: GetQEvents
/*
/*
/*****
/* Input Parameters:  Handle to the queue manager
/*                    Name of the event queue to be monitored
/*
/* Output Parameters: None
/*
/* Logic:  Open the event queue.
/*         Get a message off the event queue and format the message into
/*         a bag.
/*         A real event monitor would need to be programmed to deal with
/*         each type of event that it receives from the queue. This is
/*         outside the scope of this sample, so instead, the contents of
/*         the bag are printed.
/*         The program waits for 30 seconds for an event message and then
/*         terminates if no more messages are available.
/*
/*
/*****
void GetQEvents(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG openReason;          /* MQOPEN reason code
    MQLONG reason;             /* reason code
    MQLONG compCode;           /* completion code
    MQHOBJ eventQueue;         /* handle to event queue

    MQHBAG eventBag = MQHB_UNUSABLE_HBAG; /* event bag to receive event msg
    MQOD  od = {MQOD_DEFAULT};          /* Object Descriptor
    MQMD  md = {MQMD_DEFAULT};          /* Message Descriptor
    MQGMO gmo = {MQGMO_DEFAULT};        /* get message options
    MQLONG bQueueOK = 1;                /* keep reading msgs while true

    /*****
    /* Create an Event Bag in which to receive the event.
    /* Exit the function if the create fails.
    /*****
    mqCreateBag(MQCBO_USER_BAG, &eventBag, &compCode, &reason);
    CheckCallResult("Create event bag", compCode, reason);
    if (compCode !=MQCC_OK)
        return;

    /*****
    /* Open the event queue chosen by the user
    /*****
    strncpy(od.ObjectName, qName, (size_t)MQ_Q_NAME_LENGTH);
    MQOPEN(hConn, &od, MQOO_INPUT_AS_Q_DEF+MQOO_FAIL_IF_QUIESCING, &eventQueue,
        &compCode, &openReason);
    CheckCallResult("Open event queue", compCode, openReason);

    /*****
    /* Set the GMO options to control the action of the get message from the
    /* queue.
    /*****
    gmo.WaitInterval = 30000;          /* 30 second wait for message
    gmo.Options = MQGMO_WAIT + MQGMO_FAIL_IF_QUIESCING + MQGMO_CONVERT;
    gmo.Version = MQGMO_VERSION_2;    /* Avoid need to reset Message ID

```

```

gmo.MatchOptions = MQMO_NONE; /* and Correlation ID after every */
/* mqGetBag
/*****
/* If open fails, we cannot access the queue and must stop the monitor. */
/*****
if (compCode != MQCC_OK)
    bQueueOK = 0;

/*****
/* Main loop to get an event message when it arrives */
/*****
while (bQueueOK)
{
    printf("\nWaiting for an event\n");

    /*****
    /* Get the message from the event queue and convert it into the event */
    /* bag. */
    /*****
    mqGetBag(hConn, eventQueue, &md, &gmo, eventBag, &compCode, &reason);

    /*****
    /* If get fails, we cannot access the queue and must stop the monitor. */
    /*****
    if (compCode != MQCC_OK)
    {
        bQueueOK = 0;

        /*****
        /* If get fails because no message available then we have timed out, */
        /* so report this, otherwise report an error. */
        /*****
        if (reason == MQRC_NO_MSG_AVAILABLE)
        {
            printf("No more messages\n");
        }
        else
        {
            CheckCallResult("Get bag", compCode, reason);
        }
    }

    /*****
    /* Event message read - Print the contents of the event bag */
    /*****
    else
    {
        if ( PrintBag(eventBag) )
            printf("\nError found while printing bag contents\n");

    } /* end of msg found */
} /* end of main loop */
/*****
/* Close the event queue if successfully opened */
/*****
if (openReason == MQRC_NONE)
{
    MQCLOSE(hConn, &eventQueue, MQCO_NONE, &compCode, &reason);
    CheckCallResult("Close event queue", compCode, reason);
}

/*****
/* Delete the event bag if successfully created. */
/*****
if (eventBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&eventBag, &compCode, &reason);
    CheckCallResult("Delete the event bag", compCode, reason);
}

} /* end of GetQEvents */

/*****
/*
/* Function: PrintBag */
/*
/*
/*****
/*
/* Input Parameters: Bag Handle */
/*
/*
/* Output Parameters: None */
/*

```

```

/* Returns:          Number of errors found          */
/*                                                         */
/* Logic: Calls PrintBagContents to display the contents of the bag. */
/*                                                         */
/* *****
int PrintBag(MQHBAG dataBag)
{
    int errors;

    printf("\n");
    errors = PrintBagContents(dataBag, 0);
    printf("\n");

    return errors;
}

/* *****
/* Function: PrintBagContents          */
/* *****
/* Input Parameters:  Bag Handle          */
/*                   Indentation level of bag          */
/* *****
/* Output Parameters: None          */
/* *****
/* Returns:          Number of errors found          */
/* *****
/* Logic: Count the number of items in the bag          */
/* Obtain selector and item type for each item in the bag.          */
/* Obtain the value of the item depending on item type and display the          */
/* index of the item, the selector and the value.          */
/* If the item is an embedded bag handle then call this function again          */
/* to print the contents of the embedded bag increasing the          */
/* indentation level.          */
/* *****
int PrintBagContents(MQHBAG dataBag, int indent)
{
    /* *****
    /* Definitions          */
    /* *****
    #define LENGTH 500          /* Max length of string to be read*/
    #define INDENT 4          /* Number of spaces to indent          */
    /* *****
    /* Variables          */
    /* *****
    MQLONG itemCount;          /* Number of items in the bag          */
    MQLONG itemType;          /* Type of the item          */
    int i;          /* Index of item in the bag          */
    MQCHAR stringVal[LENGTH+1];          /* Value if item is a string          */
    MQBYTE byteStringVal[LENGTH];          /* Value if item is a byte string          */
    MQLONG stringLength;          /* Length of string value          */
    MQLONG ccsid;          /* CCSID of string value          */
    MQINT32 iValue;          /* Value if item is an integer          */
    MQINT64 i64Value;          /* Value if item is a 64-bit          */
    /* *****
    /* integer          */
    MQLONG selector;          /* Selector of item          */
    MQHBAG bagHandle;          /* Value if item is a bag handle          */
    MQLONG reason;          /* reason code          */
    MQLONG compCode;          /* completion code          */
    MQLONG trimLength;          /* Length of string to be trimmed          */
    int errors = 0;          /* Count of errors found          */
    char blanks[] = "          /* Blank string used to          */
    /* *****
    /* Count the number of items in the bag          */
    /* *****
    mqCountItems(dataBag, MQSEL_ALL_SELECTORS, &itemCount, &compCode, &reason);

    if (compCode != MQCC_OK)
        errors++;
    else
    {
        printf("
        printf("

```

```

    printf("
}

/*****
/* If no errors found, display each item in the bag */
/*****
if (!errors)
{
    for (i = 0; i < itemCount; i++)
    {

/*****
/* First inquire the type of the item for each item in the bag */
/*****
mqInquireItemInfo(dataBag, /* Bag handle */
                  MQSEL_ANY_SELECTOR, /* Item can have any selector*/
                  i, /* Index position in the bag */
                  &selector, /* Actual value of selector */
                  /* returned by call */
                  &itemType, /* Actual type of item */
                  /* returned by call */
                  &compCode, /* Completion code */
                  &reason); /* Reason Code */

        if (compCode != MQCC_OK)
            errors++;

        switch(itemType)
        {
        case MQITEM_INTEGER:
/*****
/* Item is an integer. Find its value and display its index, */
/* selector and value. */
/*****
mqInquireInteger(dataBag, /* Bag handle */
                 MQSEL_ANY_SELECTOR, /* Allow any selector */
                 i, /* Index position in the bag */
                 &iValue, /* Returned integer value */
                 &compCode, /* Completion code */
                 &reason); /* Reason Code */

            if (compCode != MQCC_OK)
                errors++;
            else
                printf("%.s %-2d %-4d (%d)\n",
                    indent, blanks, i, selector, iValue);
            break

        case MQITEM_INTEGER64:
/*****
/* Item is a 64-bit integer. Find its value and display its */
/* index, selector and value. */
/*****
mqInquireInteger64(dataBag, /* Bag handle */
                  MQSEL_ANY_SELECTOR, /* Allow any selector */
                  i, /* Index position in the bag */
                  &i64Value, /* Returned integer value */
                  &compCode, /* Completion code */
                  &reason); /* Reason Code */

            if (compCode != MQCC_OK)
                errors++;
            else
                printf("%.s %-2d %-4d (%"Int64"d)\n",
                    indent, blanks, i, selector, i64Value);
            break;

        case MQITEM_STRING:
/*****
/* Item is a string. Obtain the string in a buffer, prepare */
/* the string for displaying and display the index, selector, */
/* string and Character Set ID. */
/*****
mqInquireString(dataBag, /* Bag handle */
                MQSEL_ANY_SELECTOR, /* Allow any selector */
                i, /* Index position in the bag */
                LENGTH, /* Maximum length of buffer */
                stringVal, /* Buffer to receive string */
                &stringLength, /* Actual length of string */
                &ccsid, /* Coded character set ID */
                &compCode, /* Completion code */

```

```

                                &reason);          /* Reason Code          */

/*****
/* The call can return a warning if the string is too long for */
/* the output buffer and has been truncated, so only check   */
/* explicitly for call failure.                               */
*****/
if (compCode == MQCC_FAILED)
    errors++;
else
{
    /*****
    /* Remove trailing blanks from the string and terminate with*/
    /* a null. First check that the string should not have been */
    /* longer than the maximum buffer size allowed.             */
    *****/
    if (stringLength > LENGTH)
        trimLength = LENGTH;
    else
        trimLength = stringLength;
    mqTrim(trimLength, stringVal, stringVal, &compCode, &reason);
    printf("%.s %-2d %-4d '%s' %d\n",
           indent, blanks, i, selector, stringVal, ccsid);
}
break;

case MQITEM_BYTE_STRING:
/*****
/* Item is a byte string. Obtain the byte string in a buffer, */
/* prepare the byte string for displaying and display the     */
/* index, selector and string.                               */
*****/
mqInquireByteString(dataBag, /* Bag handle */
                    MQSEL_ANY_SELECTOR, /* Allow any selector */
                    i, /* Index position in the bag */
                    LENGTH, /* Maximum length of buffer */
                    byteStringVal, /* Buffer to receive string */
                    &stringLength, /* Actual length of string */
                    &compCode, /* Completion code */
                    &reason); /* Reason Code */

/*****
/* The call can return a warning if the string is too long for */
/* the output buffer and has been truncated, so only check   */
/* explicitly for call failure.                               */
*****/
if (compCode == MQCC_FAILED)
    errors++;
else
{
    printf("%.s %-2d %-4d X'",
           indent, blanks, i, selector);

    for (i = 0 ; i < stringLength ; i++)
        printf("

    printf("\n");
}
break;

case MQITEM_BAG:
/*****
/* Item is an embedded bag handle, so call the PrintBagContents*/
/* function again to display the contents.                   */
*****/
mqInquireBag(dataBag, /* Bag handle */
             MQSEL_ANY_SELECTOR, /* Allow any selector */
             i, /* Index position in the bag */
             &bagHandle, /* Returned embedded bag handle */
             &compCode, /* Completion code */
             &reason); /* Reason Code */

if (compCode != MQCC_OK)
    errors++;
else
{
    printf("%.s %-2d %-4d (%d)\n", indent, blanks, i,
           selector, bagHandle);
    if (selector == MQHA_BAG_HANDLE)
        printf("
    else
        printf("

```

```

        PrintBagContents(bagHandle, indent+INDENT);
    }
    break;

    default:
        printf("
    }
}
}
return errors;
}

```

Multi Kanal nesneleri hakkında bilgi almak için örnek C programı (amqsaicl.c)

Örnek C programı amqsaicl.c , MQAI kullanarak kanal nesnelerini sorar.

```

/*****
/*
/* Program name: AMQSAICL.C
/*
/* Description: Sample C program to inquire channel objects
/* using the IBM MQ Administration Interface (MQAI)
/*
/* <N_OCO_COPYRIGHT>
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 2008, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* <NOC_COPYRIGHT>
/*****
/*
/* Function:
/* AMQSAICL is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires all channels and their types.
/*
/* - A PCF command is built from items placed into an MQAI administration
/* bag.
/* These are:-
/* - The generic channel name "*"
/* - The attributes to be inquired. In this sample we just want
/* name and type attributes
/*
/* - The mqExecute MQCMD_INQUIRE_CHANNEL call is executed.
/* The call generates the correct PCF structure.
/* The default options to the call are used so that the command is sent
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.
/* The reply from the command server is placed on a temporary dynamic
/* queue.
/* The reply from the MQCMD_INQUIRE_CHANNEL is read from the
/* temporary queue and formatted into the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/* is a failure from the command server, then the code returned by the
/* command server is retrieved from the system bag that has been
/* embedded in the response bag to the mqExecute call.
/*
/* Note: The command server must be running.
/*
/*****
/*
/* AMQSAICL has 2 parameter - the queue manager name (optional)
/* - output file (optional) default varies
/*****
/*
/* Includes
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

```

```

#if (MQAT_DEFAULT == MQAT_OS400)
#include <recio.h>
#endif

#include <cmqc.h> /* MQI */
#include <cmqcfh.h> /* PCF */
#include <cmqbc.h> /* MQAI */
#include <cmqxc.h> /* MQCD */

/*****
/* Function prototypes */
/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);

/*****
/* DataTypes */
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
typedef _RFILE OUTFILEHDL;
#else
typedef FILE OUTFILEHDL;
#endif

/*****
/* Constants */
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    " *SDR      ", /* MQCHT_SENDER */
    " *SVR      ", /* MQCHT_SERVER */
    " *RCVR     ", /* MQCHT_RECEIVER */
    " *RQSTR    ", /* MQCHT_REQUESTER */
    " *ALL      ", /* MQCHT_ALL */
    " *CLTCN    ", /* MQCHT_CLNTCONN */
    " *SVRCONN  ", /* MQCHT_SVRCONN */
    " *CLUSRCVR", /* MQCHT_CLUSRCVR */
    " *CLUSSDR  ", /* MQCHT_CLUSSDR */
};
#else
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    "sdr      ", /* MQCHT_SENDER */
    "svr      ", /* MQCHT_SERVER */
    "rcvr     ", /* MQCHT_RECEIVER */
    "rqstr    ", /* MQCHT_REQUESTER */
    "all      ", /* MQCHT_ALL */
    "cltconn  ", /* MQCHT_CLNTCONN */
    "svrcn    ", /* MQCHT_SVRCONN */
    "clusrcvr", /* MQCHT_CLUSRCVR */
    "clussdr  ", /* MQCHT_CLUSSDR */
};
#endif

/*****
/* Macros */
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
#define OUTFILE "QTEMP/AMQSAICL(AMQSAICL)"
#define OPENOUTFILE(hdl, fname) \
    (hdl = _Ropen((fname), "wr, rtncode=Y");
#define CLOSEOUTFILE(hdl) \
    _Rclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    _Rwrite((hdl), (buf), (buflen));

#elif (MQAT_DEFAULT == MQAT_UNIX)
#define OUTFILE "/tmp/amqsaicl.txt"
#define OPENOUTFILE(hdl, fname) \
    (hdl = _fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
    _fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    _fwrite((buf), (buflen), 1, (hdl)); _fflush((hdl));

#else

```

```

#define OUTFILE "amqsaicl.txt"
#define OPENOUTFILE(fname) \
    fopen((fname),"w");
#define CLOSEOUTFILE(hdl) \
    fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    fwrite((buf),(buflen),1,(hdl)); fflush((hdl));

#endif

#define ChlType2String(t) ChlTypeMap[(t)-1].name

/*****
/* Function: main
*****/
int main(int argc, char *argv[])
{
    /*****/
    /* MQAI variables
    *****/
    MQHCONN hConn; /* handle to MQ connection */
    MQCHAR qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG reason; /* reason code */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */
    MQHBAG adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute */
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
    MQHBAG cAttrsBag; /* bag containing chl attributes */
    MQHBAG errorBag; /* bag containing cmd server error */
    MQLONG mqExecuteCC; /* mqExecute completion code */
    MQLONG mqExecuteRC; /* mqExecute reason code */
    MQLONG chlNameLength; /* Actual length of chl name */
    MQLONG chlType; /* Channel type */
    MQLONG i; /* loop counter */
    MQLONG numberOfBags; /* number of bags in response bag */
    MQCHAR chlName[MQ_OBJECT_NAME_LENGTH+1]; /* name of chl extracted from bag */
    MQCHAR OutputBuffer[100]; /* output data buffer */
    OUTFILEHDL *outfp = NULL; /* output file handle

    /*****/
    /* Connect to the queue manager
    *****/
    if (argc > 1)
        strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(qmName, &hConn, &compCode, &connReason);

    /*****/
    /* Report the reason and stop if the connection failed.
    *****/
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("Queue Manager connection", compCode, connReason);
        exit( (int)connReason);
    }

    /*****/
    /* Open the output file
    *****/
    if (argc > 2)
    {
        OPENOUTFILE(outfp, argv[2]);
    }
    else
    {
        OPENOUTFILE(outfp, OUTFILE);
    }

    if(outfp == NULL)
    {
        printf("Could not open output file.\n");
        goto MOD_EXIT;
    }

    /*****/
    /* Create an admin bag for the mqExecute call
    *****/
    mqCreateBag(MQCBO_ADMIN_BAG, &adminBag, &compCode, &reason);
    CheckCallResult("Create admin bag", compCode, reason);

    /*****/
    /* Create a response bag for the mqExecute call
    *****/
    mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);

```



```

CheckCallResult("Create response bag", compCode, reason);

/*****
/* Put the generic channel name into the admin bag */
/*****
mqAddString(adminBag, MQCACH_CHANNEL_NAME, MQBL_NULL_TERMINATED, "*",
            &compCode;, &reason;);
CheckCallResult("Add channel name", compCode, reason);

/*****
/* Put the channel type into the admin bag */
/*****
mqAddInteger(adminBag, MQIACH_CHANNEL_TYPE, MQCHT_ALL, &compCode;, &reason;);
CheckCallResult("Add channel type", compCode, reason);

/*****
/* Add an inquiry for various attributes */
/*****
mqAddInquiry(adminBag, MQIACH_CHANNEL_TYPE, &compCode;, &reason;);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the channel names and channel types. */
/* The mqExecute call creates the PCF structure required, sends it to */
/* the command server, and receives the reply from the command server into */
/* the response bag. The attributes are contained in system bags that are */
/* embedded in the response bag, one set of attributes per bag. */
/*****
mqExecute(hConn, /* MQ connection handle */
          MQCMD_INQUIRE_CHANNEL, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          adminBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response */
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE */
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode;, /* Completion code from the mqexecute */
          &reason;); /* Reason code from mqexecute call */

/*****
/* Check the command server is started. If not exit. */
/*****
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName="">\n");
    goto MOD_EXIT;
}

/*****
/* Check the result from mqExecute call. If successful find the channel */
/* types for all the channels. If failed find the error. */
/*****
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
    /*****
    /* Count the number of system bags embedded in the response bag from the */
    /* mqExecute call. The attributes for each channel are in separate bags. */
    /*****
    mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags;,
                &compCode;, &reason;);
    CheckCallResult("Count number of bag handles", compCode, reason);

    for ( i=0; i<numberOfBags; i++)
    {
        /*****
        /* Get the next system bag handle out of the mqExecute response bag. */
        /* This bag contains the channel attributes */
        /*****
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &cAttrsBag,
                    &compCode;, &reason;);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the channel name out of the channel attributes bag */
        /*****
        mqInquireString(cAttrsBag, MQCACH_CHANNEL_NAME, 0, MQ_OBJECT_NAME_LENGTH,
                        chlName, &chlNameLength, NULL, &compCode;, &reason;);
        CheckCallResult("Get channel name", compCode, reason);

        /*****
        /* Get the channel type out of the channel attributes bag */
        /*****

```

```

mqInquireInteger(cAttrsBag, MQIACH_CHANNEL_TYPE, MQIND_NONE, &chlType,
                &compCode, &reason);
    CheckCallResult("Get type", compCode, reason);

    /******
    /* Use mqTrim to prepare the channel name for printing.          */
    /* Print the result.                                             */
    /******
mqTrim(MQ_CHANNEL_NAME_LENGTH, chlName, chlName, &compCode, &reason);
sprintf(OutputBuffer, "%-20s%-9s", chlName, ChlType2String(chlType));
WRITEOUTFILE(outfp, OutputBuffer, 29)
}
}
else /* Failed mqExecute */
{
printf("Call to get channel attributes failed: Cc = %ld : Rc = %ld\n",
      compCode, reason);
/******
/* If the command fails get the system bag handle out of the mqexecute */
/* response bag. This bag contains the reason from the command server */
/* why the command failed.                                           */
/******
if (reason == MQRCCF_COMMAND_FAILED)
{
mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag,
            &compCode, &reason);
    CheckCallResult("Get the result bag handle", compCode, reason);

    /******
    /* Get the completion code and reason code, returned by the command */
    /* server, from the embedded error bag.                               */
    /******
mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                &compCode, &reason);
    CheckCallResult("Get the completion code from the result bag",
                    compCode, reason);
mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                &compCode, &reason);
    CheckCallResult("Get the reason code from the result bag",
                    compCode, reason);
    printf("Error returned by the command server: Cc = %ld : Rc = %ld\n",
          mqExecuteCC, mqExecuteRC);
}
}
}
MOD_EXIT:
/******
/* Delete the admin bag if successfully created.                    */
/******
if (adminBag != MQHB_UNUSABLE_HBAG)
{
mqDeleteBag(&adminBag, &compCode, &reason);
    CheckCallResult("Delete the admin bag", compCode, reason);
}

/******
/* Delete the response bag if successfully created.                */
/******
if (responseBag != MQHB_UNUSABLE_HBAG)
{
mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}

/******
/* Disconnect from the queue manager if not already connected      */
/******
if (connReason != MQRC_ALREADY_CONNECTED)
{
MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
}

/******
/* Close the output file if open                                    */
/******
if(outfp != NULL)
    CLOSEOUTFILE(outfp);

return 0;
}

```

```

/*****
/*
/* Function: CheckCallResult
/*
/*****
/*
/* Input Parameters:  Description of call
/*                    Completion code
/*                    Reason code
/*
/* Output Parameters: None
/*
/* Logic: Display the description of the call, the completion code and the
/*        reason code if the completion code is not successful
/*
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %ld : Reason = %ld\n", callText,
              cc, rc);
}

```

Multi Kuyruklar ve yazdırma bilgileri hakkında bilgi almak için örnek C programı (amqsailq.c)

Örnek C programı amqsailq.c , MQAI kullanarak yerel kuyrukların yürürlükteki derinliğini sorar.

```

/*****
/*
/* Program name: AMQSAILQ.C
/*
/* Description:  Sample C program to inquire the current depth of the local
/*              queues using the IBM MQ Administration Interface (MQAI)
/*
/* Statement:    Licensed Materials - Property of IBM
/*
/*              84H2000, 5765-B73
/*              84H2001, 5639-B42
/*              84H2002, 5765-B74
/*              84H2003, 5765-B75
/*              84H2004, 5639-B43
/*
/*              (C) Copyright IBM Corp. 1999, 2024
/*
/*****
/*
/* Function:
/* AMQSAILQ is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires the current depths of all the local queues.
/*
/* - A PCF command is built by placing items into an MQAI administration
/*   bag.
/*   These are:-
/*     - The generic queue name "*"
/*     - The type of queue required. In this sample we want to
/*       inquire local queues.
/*     - The attribute to be inquired. In this sample we want the
/*       current depths.
/*
/* - The mqExecute call is executed with the command MQCMD_INQUIRE_Q.
/*   The call generates the correct PCF structure.
/*   The default options to the call are used so that the command is sent
/*   to the SYSTEM.ADMIN.COMMAND.QUEUE.
/*   The reply from the command server is placed on a temporary dynamic
/*   queue.
/*   The reply from the MQCMD_INQUIRE_Q command is read from the
/*   temporary queue and formatted into the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/*   is a failure from the command server, then the code returned by
/*   command server is retrieved from the system bag that has been
/*   embedded in the response bag to the mqExecute call.
/*

```

```

/*      - If the call is successful, the depth of each local queue is placed      */
/*      in system bags embedded in the response bag of the mqExecute call.      */
/*      The name and depth of each queue is obtained from each of the bags      */
/*      and the result displayed on the screen.                                  */
/*                                                                              */
/* Note: The command server must be running.                                  */
/*                                                                              */
/*****
/*
/* AMQSAILQ has 1 parameter - the queue manager name (optional)
/*
*****/

/*****
/* Includes
*****/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h>          /* MQI          */
#include <cmqcfc.h>       /* PCF         */
#include <cmqbc.h>        /* MQAI        */

/*****
/* Function prototypes
*****/
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);

/*****
/* Function: main
*****/
int main(int argc, char *argv[])
{
    /*****
    /* MQAI variables
    *****/
    MQHCONN hConn;          /* handle to IBM MQ connection */
    MQCHAR  qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG  reason;         /* reason code */
    MQLONG  connReason;    /* MQCONN reason code */
    MQLONG  compCode;      /* completion code */
    MQHBAG  adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute */
    MQHBAG  responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
    MQHBAG  qAttrsBag;     /* bag containing q attributes */
    MQHBAG  errorBag;      /* bag containing cmd server error */
    MQLONG  mqExecuteCC;   /* mqExecute completion code */
    MQLONG  mqExecuteRC;   /* mqExecute reason code */
    MQLONG  qNameLength;   /* Actual length of q name */
    MQLONG  qDepth;        /* depth of queue */
    MQLONG  i;             /* loop counter */
    MQLONG  numberOfBags;  /* number of bags in response bag */
    MQCHAR  qName[MQ_Q_NAME_LENGTH+1]; /* name of queue extracted from bag*/

    printf("Display current depths of local queues\n\n");

    /*****
    /* Connect to the queue manager
    *****/
    if (argc > 1)
        strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(qmName, &hConn, &compCode, &connReason);

    /*****
    /* Report the reason and stop if the connection failed.
    *****/
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("Queue Manager connection", compCode, connReason);
        exit( (int)connReason);
    }

    /*****
    /* Create an admin bag for the mqExecute call
    *****/
    mqCreateBag(MQCBO_ADMIN_BAG, &adminBag, &compCode, &reason);
    CheckCallResult("Create admin bag", compCode, reason);
    /*****
    /* Create a response bag for the mqExecute call
    *****/

```

```

mqCreateBag(MQCB0_ADMIN_BAG, &responseBag, &compCode, &reason);
CheckCallResult("Create response bag", compCode, reason);

/*****
/* Put the generic queue name into the admin bag */
*****/
mqAddString(adminBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, "*",
            &compCode, &reason);
CheckCallResult("Add q name", compCode, reason);

/*****
/* Put the local queue type into the admin bag */
*****/
mqAddInteger(adminBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
CheckCallResult("Add q type", compCode, reason);

/*****
/* Add an inquiry for current queue depths */
*****/
mqAddInquiry(adminBag, MQIA_CURRENT_Q_DEPTH, &compCode, &reason);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the local queue names and queue depths. */
/* The mqExecute call creates the PCF structure required, sends it to */
/* the command server, and receives the reply from the command server into */
/* the response bag. The attributes are contained in system bags that are */
/* embedded in the response bag, one set of attributes per bag. */
*****/
mqExecute(hConn, /* IBM MQ connection handle */
          MQCMD_INQUIRE_Q, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          adminBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response */
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE */
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode, /* Completion code from the mqExecute */
          &reason); /* Reason code from mqExecute call */

/*****
/* Check the command server is started. If not exit. */
*****/
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName>\n");
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
    exit(98);
}

/*****
/* Check the result from mqExecute call. If successful find the current */
/* depths of all the local queues. If failed find the error. */
*****/
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
    /*****
    /* Count the number of system bags embedded in the response bag from the */
    /* mqExecute call. The attributes for each queue are in a separate bag. */
    *****/
    mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags, &compCode,
                &reason);
    CheckCallResult("Count number of bag handles", compCode, reason);

    for ( i=0; i<numberOfBags; i++)
    {
        /*****
        /* Get the next system bag handle out of the mqExecute response bag. */
        /* This bag contains the queue attributes */
        *****/
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &qAttrsBag, &compCode,
                    &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the queue name out of the queue attributes bag */
        *****/
        mqInquireString(qAttrsBag, MQCA_Q_NAME, 0, MQ_Q_NAME_LENGTH, qName,
                        &qNameLength, NULL, &compCode, &reason);
        CheckCallResult("Get queue name", compCode, reason);
    }
}

```

```

/*****
/* Get the depth out of the queue attributes bag */
/*****
mqInquireInteger(qAttrBag, MQIA_CURRENT_Q_DEPTH, MQIND_NONE, &qDepth,
                &compCode, &reason);
CheckCallResult("Get depth", compCode, reason);

/*****
/* Use mqTrim to prepare the queue name for printing. */
/* Print the result. */
/*****
mqTrim(MQ_Q_NAME_LENGTH, qName, qName, &compCode, &reason);
printf("%4d %-48s\n", qDepth, qName);
}
}
else /* Failed mqExecute */
{
printf("Call to get queue attributes failed: Completion Code = %d :
      Reason = %d\n", compCode, reason);

/*****
/* If the command fails get the system bag handle out of the mqExecute */
/* response bag. This bag contains the reason from the command server */
/* why the command failed. */
/*****
if (reason == MQRCCF_COMMAND_FAILED)
{
mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag, &compCode,
            &reason);
CheckCallResult("Get the result bag handle", compCode, reason);

/*****
/* Get the completion code and reason code, returned by the command */
/* server, from the embedded error bag. */
/*****
mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                &compCode, &reason);
CheckCallResult("Get the completion code from the result bag",
                compCode, reason);
mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                &compCode, &reason);
CheckCallResult("Get the reason code from the result bag",
                compCode, reason);
printf("Error returned by the command server: Completion Code = %d :
      Reason = %d\n", mqExecuteCC, mqExecuteRC);
}
}

/*****
/* Delete the admin bag if successfully created. */
/*****
if (adminBag != MQHB_UNUSABLE_HBAG)
{
mqDeleteBag(&adminBag, &compCode, &reason);
CheckCallResult("Delete the admin bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created. */
/*****
if (responseBag != MQHB_UNUSABLE_HBAG)
{
mqDeleteBag(&responseBag, &compCode, &reason);
CheckCallResult("Delete the response bag", compCode, reason);
}

/*****
/* Disconnect from the queue manager if not already connected */
/*****
if (connReason != MQRC_ALREADY_CONNECTED)
{
MQDISC(&hConn, &compCode, &reason);
CheckCallResult("Disconnect from queue manager", compCode, reason);
}
return 0;
}

*****/
*
* Function: CheckCallResult
*
*/

```

```

*****/
*
* Input Parameters:  Description of call          */
*                   Completion code             */
*                   Reason code                */
*
* Output Parameters: None                       */
*
* Logic: Display the description of the call, the completion code and the */
*        reason code if the completion code is not successful             */
*
*****/
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
  if (cc != MQCC_OK)
    printf("%s failed: Completion Code = %d : Reason = %d\n",
          callText, cc, rc);
}

```

Multi Veri torbaları ve MQAI

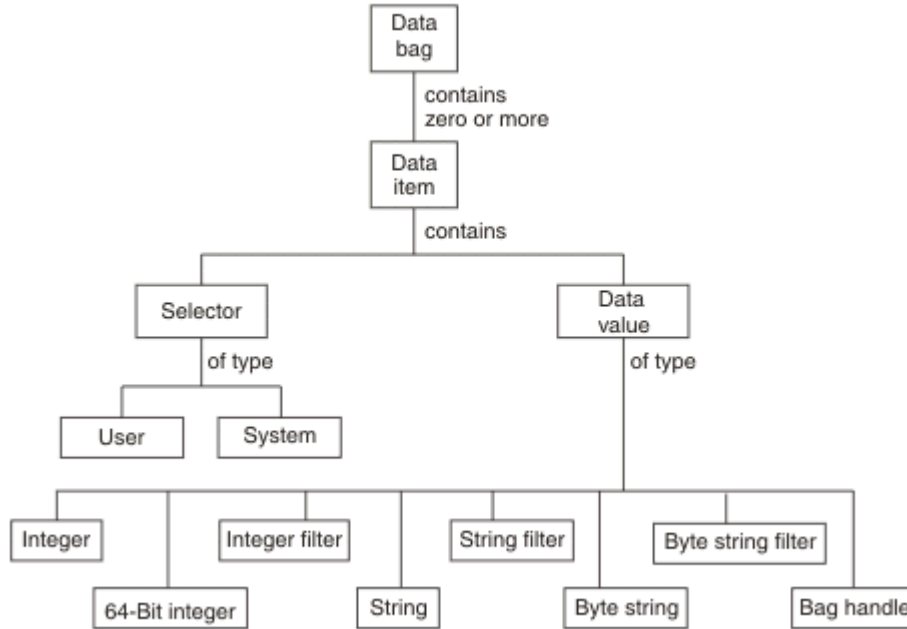
Veri torbası, IBM MQ Yönetim Arabirimi 'ni (MQAI) kullanarak nesnelerin özelliklerini ya da parametrelerini işleme aracıdır.

Veri Torbaları

- Veri torbası sıfır ya da daha fazla *veri öğesi* içerir. Bu veri öğeleri torbaya yerleştirilirken torbanın içinde sipariş edilir. Buna *ekleme sırası* denir. Her veri öğesi, bir tamsayı, 64 bitlik tamsayı, tamsayı süzgeci, dizgi süzgeci, byte dizgisi, byte dizgisi süzgeci ya da başka bir çantanın tanıtıcısı olabilen veri öğesini ve o veri öğesinin *değerini* tanımlayan bir *seçici* içerir. Veri öğeleri "[MQAI içinde kullanılabilir veri öğesi tipleri](#)" sayfa 65 içindeki ayrıntılarda açıklanır

İki tip seçici vardır: *kullanıcı seçiciler* ve *sistem seçiciler*. Bunlar MQAI Seçicileri içinde açıklanmıştır. Seçiciler genellikle benzersizdir, ancak aynı seçici için birden çok değere sahip olabilir. Bu durumda, bir *dizin* gerekli seçicinin belirli bir oluşumunu tanımlar. Dizinler "[MQAI içinde dizinleme](#)" sayfa 39 içinde açıklanmıştır.

Bu kavramların sıradüzeni [Şekil 1'](#) de gösterilir.



Şekil 4. MQAI kavramları sıradüzeni

Sıradüzen önceki bir paragrafta açıklanmıştır.

Veri torbası tipleri

Gerçekleştirmek istediğiniz göreve bağlı olarak oluşturmak istediğiniz veri paketi tipini seçebilirsiniz:

kullanıcı çantası

Kullanıcı verileri için kullanılan basit bir çanta.

yönetim çantası

Bir komut sunucusuna yönetim iletileri göndererek IBM MQ nesnelerini yönetmek için kullanılan veriler için oluşturulan bir torba. Yönetim çantası, [“Veri torbaları oluşturma ve silme” sayfa 64](#) içinde açıklandığı gibi belirli seçenekleri otomatik olarak belirtir.

komut çantası

IBM MQ nesnelerini yönetmeye ilişkin komutlar için de bir çanta oluşturuldu. Ancak, yönetim çantasının aksine, komut çantası otomatik olarak belirli seçenekleri belirtmez, ancak bu seçenekler mevcuttur. Seçenekler hakkında daha fazla bilgi için bkz. [“Veri torbaları oluşturma ve silme” sayfa 64](#).

grup çantası

Gruplanmış veri öğeleri kümesini tutmak için kullanılan bir çanta. IBM MQ nesnelerini yönetmek için grup torbaları kullanılamaz.

Buna ek olarak, **sistem çantası**, komut sunucusundan bir yanıt iletilisi döndürüldüğünde ve bir kullanıcının çıkış çantasına yerleştirildiğinde MQAI tarafından oluşturulur. Bir sistem çantası kullanıcı tarafından değiştirilemez.

Veri torbalarını kullanma Veri torbalarını kullanmanın farklı yolları bu konuda listelenmiştir:

Veri Torbalarının Kullanılması

Veri torbalarını kullanmanın farklı yolları aşağıdaki listede gösterilmiştir:

- Veri torbaları oluşturabilir ve silebilirsiniz [“Veri torbaları oluşturma ve silme” sayfa 64](#).
- [“MQAI kullanılarak veri torbalarının yerleştirilip alınması” sayfa 65](#) veri torbalarını kullanarak uygulamalar arasında veri gönderebilirsiniz.
- Veri torbalarına veri öğeleri ekleyebilirsiniz [“MQAI ile veri öğelerini çantalara ekleme” sayfa 66](#).
- [“Torbaya sorgu komutu ekleme” sayfa 67](#) veri torbasına sorgu komutu ekleyebilirsiniz.
- [“Veri torbaları içinde sorma” sayfa 67](#) veri torbaları içinde sorabilirsiniz.
- Bir veri torbasındaki veri öğelerini sayabilirsiniz [“Veri öğeleri sayılıyor” sayfa 70](#).
- Bir veri torbasındaki bilgileri değiştirebilirsiniz [“Torba içindeki bilgilerin değiştirilmesi” sayfa 68](#).
- [“mqClearBag çağrısını kullanarak bir çantayı temizleme” sayfa 69](#) veri torbasını temizleyebilirsiniz.
- Bir veri torbasını kesebilirsiniz [“mqTruncateBag çağrısını kullanarak bir torbanın kesilmesi” sayfa 69](#).
- Torbaları ve arabellekleri dönüştürebilirsiniz [“Torbaları ve arabellekleri dönüştürme” sayfa 69](#).

Multi Veri torbaları oluşturma ve silme

Veri torbaları oluşturma

MQAI olanağını kullanmak için önce mqCreateBag çağrısını kullanarak bir veri çantası yaratırsınız. Bu aramaya giriş olarak, çuvalın oluşturulmasını denetlemek için bir ya da daha fazla seçenek sunarınız.

mqCreateBag çağrısının **Options** değiştirgesi, bir kullanıcı çantası, komut çantası, grup çantası ya da yönetim çantası yaratmayı seçmenizi sağlar.

Bir kullanıcı çantası, komut çantası ya da grup torbası oluşturmak için aşağıdaki seçeneklerden birini ya da daha fazlasını belirleyebilirsiniz:

- Bir torbada aynı seçicinin iki ya da daha fazla bitişik yinelenmesi olduğunda liste formunu kullanın.

- Parametrelerin doğru sırada olduğundan emin olmak için bir PCF iletisine eklendikçe veri öğelerini yeniden sıralayın. Veri öğeleriyle ilgili daha fazla bilgi için bkz. [“MQAI içinde kullanılabilir veri öğesi tipleri” sayfa 65](#).
- Torbaya eklediğiniz öğeler için kullanıcı seçicilerin değerlerini denetleyin.

Yönetim çantaları otomatik olarak bu seçeneklere işaret ediyor.

Bir veri torbası, tutamacı ile tanımlanır. Çanta tanıtıcısı mqCreatepaketinden döndürülür ve veri torbasını kullanan diğer tüm çağrılarda sağlanmalıdır.

mqCreateBag çağrısının tam açıklaması için bkz. [mqCreateBag](#).

Veri torbaları siliniyor

Kullanıcı tarafından yaratılan veri torbası mqDeleteBag çağrısı kullanılarak da silinmelidir. Örneğin, kullanıcı kodunda bir paket oluşturulursa, bu paket kullanıcı kodunda da silinmelidir.

Sistem torbaları MQAI tarafından otomatik olarak oluşturulur ve silinir. Bu konuda daha fazla bilgi için bkz. [“mqExecute çağrısı kullanarak qm komut sunucusuna yönetim komutları gönderme” sayfa 71](#). Kullanıcı kodu bir sistem çantasını silemez.

mqDeleteBag çağrısının tam açıklaması için bkz. [mqDeleteBag](#).

Multi **MQAI kullanılarak veri torbalarının yerleştirilip alınması**

Veriler, mqPutBag ve mqGetBag çağrılarını kullanarak veri torbaları yerleştirilip alınarak da uygulamalar arasında gönderilebilir. Bu, IBM MQ Denetim Arabirimi 'nin (MQAI) uygulama yerine arabelleği işlemlerini sağlar.

mqPutBag çağrısı, belirtilen torbanın içeriğini bir PCF iletisine dönüştürür ve iletiyi belirtilen kuyruğa gönderir ve mqGetBag çağrısı iletiyi belirtilen kuyruktan kaldırır ve veri torbasına geri dönüştürür. Bu nedenle, mqPutBag çağrısı mqBagToBuffer çağrısının eşdeğeridir ve mqGetBag, MQGET çağrısının eşdeğeridir ve ardından mqBufferToBaggelir.

Belirli bir kuyruktaki PCF iletilerini gönderme ve alma hakkında daha fazla bilgi için bkz. [“Belirli bir kuyruktaki PCF iletileri gönderilmesi ve alınması” sayfa 28](#)

Not: mqGetBag çağrısı kullanmayı seçerseniz, ileti içindeki PCF ayrıntıları doğru olmalıdır; değilse, uygun bir hata ortaya çıkar ve PCF iletisi döndürülmez.

Multi **MQAI içinde kullanılabilir veri öğesi tipleri**

Veri öğeleri, oluşturulduğunda veri torbalarını doldurmak için IBM MQ Yönetim Arabirimi (MQAI) tarafından kullanılır. Bu veri öğeleri kullanıcı ya da sistem öğeleri olabilir.

Bu kullanıcı öğeleri, denetlenmekte olan nesnelere öznitelikleri gibi kullanıcı verilerini içerir. Sistem öğeleri, oluşturulan iletiler üzerinde daha fazla denetim için kullanılmalıdır: örneğin, ileti üstbilgilerinin oluşturulması. Sistem öğeleriyle ilgili daha fazla bilgi için bkz. [“Sistem öğeleri ve MQAI” sayfa 66](#).

Veri Öğesi Tipleri

Bir veri torbası oluşturduğunuzda, bunu tamsayı ya da karakter dizgisi öğeleriyle doldurabilirsiniz. Her üç öğe tipini de sorgulayabilirsiniz.

Veri öğesi tamsayı ya da karakter dizgisi öğeleri olabilir. MQAI içinde kullanılacak veri öğesi tipleri şunlardır:

- Tamsayı
- 64 bitlik tamsayı
- Tamsayı süzgeci
- Karakter dizilimi

- Dizgi süzgeci
- Bayt dizilimi
- Byte dizgisi süzgeci
- Torba tutamacı

Veri Öğelerini Kullanma

Bunlar, veri öğelerini kullanmanın aşağıdaki yollarıdır:

- “Veri öğeleri sayılıyor” sayfa 70.
- “Veri öğelerini silme” sayfa 70.
- “MQAI ile veri öğelerini çantalara ekleme” sayfa 66.
- “Veri öğelerini süzgeçten geçirme ve sorgulama” sayfa 67.

Multi Sistem öğeleri ve MQAI

Sistem öğeleri, aşağıdakiler için IBM MQ Yönetim Arabirimi (MQAI) tarafından kullanılabilir:

- PCF üstbilgilerinin oluşturulması. Sistem öğeleri, PCF komut tanıtıcısını, denetim seçeneklerini, ileti sıra numarasını ve komut tipini denetleyebilir.
- Veri dönüştürme. Sistem öğeleri, çantadaki karakter dizilimi öğelerine ilişkin karakter kümesi tanıtıcısını işler.

Tüm veri öğeleri gibi, sistem öğeleri de bir seçiciden ve bir değerden oluşur. Bu seçiciler ve bunların ne için olduğu hakkında bilgi için bkz. [MQAI Seçicileri](#).

Sistem öğeleri benzersizdir. Bir ya da daha çok sistem öğesi bir sistem seçici tarafından tanımlanabilir. Her sistem seçicinin yalnızca bir oluşumu vardır.

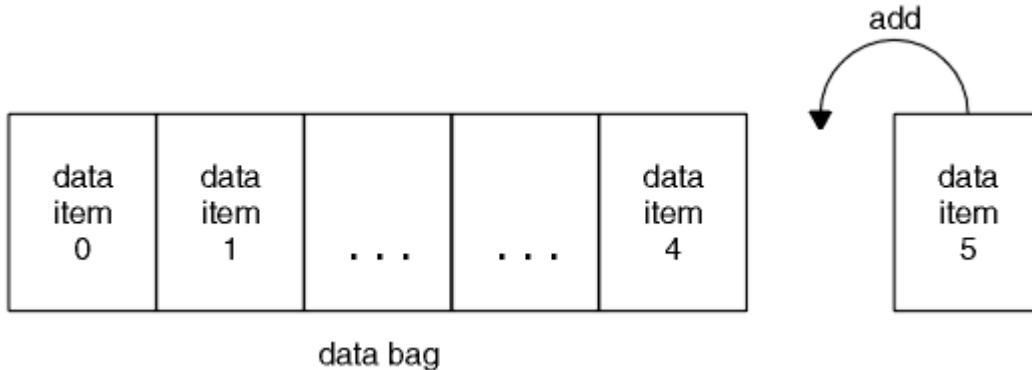
Çoğu sistem öğesi değiştirilebilir (bkz. “Torba içindeki bilgilerin değiştirilmesi” sayfa 68), ancak çanta oluşturma seçenekleri kullanıcı tarafından değiştirilemez. Sistem öğelerini silemezsiniz. (Bkz. “Veri öğelerini silme” sayfa 70.)

Multi MQAI ile veri öğelerini çantalara ekleme

IBM MQ Yönetim Arabirimi (MQAI) kullanılarak bir veri paketi oluşturulduğunda, veri öğeleriyle doldurabilirsiniz. Bu veri öğeleri kullanıcı ya da sistem öğeleri olabilir.

Veri öğeleri hakkında daha fazla bilgi için bkz. “MQAI içinde kullanılabilir veri öğesi tipleri” sayfa 65.

MQAI, tamsayı öğeleri, 64 bitlik tamsayı öğeleri, tamsayı süzgeci öğeleri, karakter dizgisi öğeleri, dizgi süzgeci, byte dizgisi öğeleri ve byte dizgisi süzgeci öğelerini torbalara eklemenizi sağlar ve bu [Şekil 5 sayfa 66](#) içinde gösterilir. Öğeler bir seçici tarafından tanımlanır. Genellikle bir seçici yalnızca bir öğeyi tanımlar, ancak bu her zaman böyle olmaz. Belirtilen seçiciye sahip bir veri öğesi zaten varsa, bu seçicinin ek bir örneği çantanın sonuna eklenir.



Şekil 5. Veri öğeleri ekleme

mqAdd* çağrılarını kullanarak bir çantaya veri öğeleri ekleyin:

- Tamsayı öğeleri eklemek için, [mqAddInteger](#) içinde açıklandığı gibi mqAddTamsayı çağrısı kullanın.
- 64 bitlik tamsayı öğeleri eklemek için, [mqAddInteger64](#) içinde açıklandığı gibi mqAddInteger64 çağrısı kullanın.
- Tamsayı süzgeç öğeleri eklemek için, [mqAddIntegerFilter](#) içinde açıklandığı gibi mqAddIntegerFilter çağrısı kullanın.
- Karakter dizgisi öğeleri eklemek için, [mqAddString](#) içinde açıklandığı gibi mqAdddizgi çağrısı kullanın.
- Dizgi süzgeci öğeleri eklemek için, [mqAddStringFilter](#) içinde açıklandığı gibi mqAddStringFilter çağrısı kullanın.
- Byte dizgisi öğeleri eklemek için, [mqAddByteString](#) içinde açıklandığı gibi mqAddByteString çağrısı kullanın.
- Bayt dizgisi süzgeci öğeleri eklemek için, [mqAddByteStringFilter](#) içinde açıklandığı gibi mqAddByteStringSüzgeç çağrısı kullanın.

Bir torbaya veri öğeleri ekleme hakkında daha fazla bilgi için bkz. [“Sistem öğeleri ve MQAI” sayfa 66.](#)

Multi

Torbaya sorgu komutu ekleme

mqAddSorgu çağrısı, bir çantaya sorgu komutu eklemek için kullanılır. Çağrı özellikle yönetim amaçlı olduğundan, yalnızca yönetim çantalarıyla kullanılabilir. IBM MQ’ından sorgulamak istediğiniz özniteliklerin seçicilerini belirtmenizi sağlar.

mqAddSorgu çağrısının tam açıklaması için bkz. [mqAddInquiry](#).

Multi

Veri öğelerini süzgeçten geçirme ve sorgulama

IBM MQ nesnelerinin özniteliklerini sorgulamak için MQAI kullanırken, programınıza döndürülen verileri iki şekilde denetleyebilirsiniz.

- mqAddInteger ve mqAddString çağrılarını kullanarak döndürülen verileri **süzebilirsiniz** . Bu yaklaşım bir *Selector* ve *ItemValue* çifti belirtmenize olanak sağlar, örneğin:

```
mqAddInteger(inputbag, MQIA_Q_TYPE, MQQT_LOCAL)
```

Bu örnek, kuyruk tipinin (*Selector*) yerel (*ItemValue*) olması gerektiğini ve bu belirtimin, sorguladığınız nesnenin öznitelikleriyle (bu durumda, bir kuyruk) eşleşmesi gerektiğini belirtir.

Süzülebilen diğer öznitelikler, [“IBM MQ Programlanabilir Komut Biçimlerine Giriş” sayfa 25](#)’inde bulunan PCF Inquire * komutlarına karşılık gelir. Örneğin, bir kanalın özniteliklerini sorgulamak için bu ürün belgelerinde Inquire Channel komutuna bakın. Inquire Channel komutunun "Gerekli parametreler" ve "İsteğe bağlı parametreler", süzgeç uygulamak için kullanabileceğiniz seçicileri tanımlar.

- mqAddSorgu çağrısıyla bir nesnenin belirli özniteliklerini **sorgulayabilirsiniz** . İlgilendiğiniz seçiciyi belirtir. Seçiciyi belirtmezseniz, nesnenin tüm öznitelikleri döndürülür.

Aşağıda, bir kuyruğun özniteliklerini süzme ve sorgulama örneği verilmiştir:

```
/* Request information about all queues */
mqAddString(adminbag, MQCA_Q_NAME, "*")

/* Filter attributes so that local queues only are returned */
mqAddInteger(adminbag, MQIA_Q_TYPE, MQQT_LOCAL)

/* Query the names and current depths of the local queues */
mqAddInquiry(adminbag, MQCA_Q_NAME)
mqAddInquiry(adminbag, MQIA_CURRENT_Q_DEPTH)

/* Send inquiry to the command server and wait for reply */
mqExecute(MQCMD_INQUIRE_Q, ...)
```

Multi

Veri torbaları içinde sorma

Aşağıdakiler hakkında bilgi alabilirsiniz:

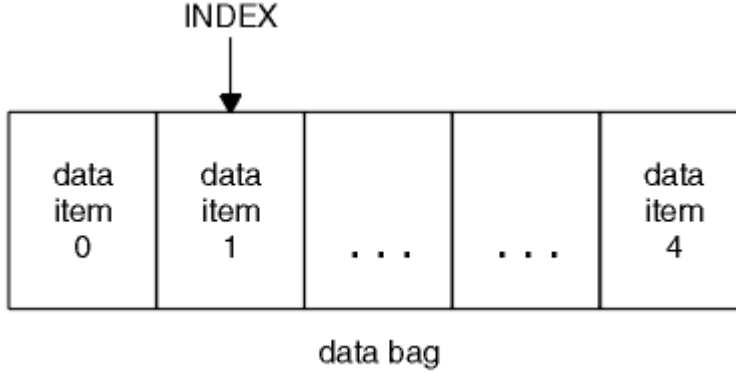
- mqInquireTamsayı çağrısının kullanıldığı bir tamsayı ögesinin değeri. Bkz. [mqInquireTamsayı](#).
- mqInquireInteger64 çağrısının kullanıldığı 64 bitlik bir tamsayı ögesinin değeri. Bkz. [mqInquireInteger64](#).
- mqInquireIntegerFilter çağrısının kullanıldığı bir tamsayı süzgeç ögesinin değeri. Bkz. [mqInquireIntegerFilter](#).
- mqInquireDizgi çağrısının kullanıldığı bir karakter dizgisi ögesinin değeri. Bkz. [mqInquireDizesi](#).
- mqInquireStringFilter çağrısının kullanıldığı dizgi süzgeci ögesinin değeri. Bkz. [mqInquireStringFilter](#).
- mqInquireByteString çağrısının kullanıldığı bir byte dizgisi ögesinin değeri. Bkz. [mqInquireByteString](#).
- mqInquireByteStringSüzgeç çağrısının kullanıldığı bir byte dizgisi süzgeç ögesinin değeri. Bkz. [mqInquireByteStringFilter](#).
- mqInquireBag çağrısını kullanan bir çanta tanıtıcısının değeri. Bkz. [mqInquireBag](#).

mqInquireItemInfo çağrısını kullanarak belirli bir ögenin tipini (tamsayı, 64 bitlik tamsayı, tamsayı süzgeci, karakter dizgisi, dizgi süzgeci, bayt dizgisi süzgeci ya da çanta tanıtıcısı) sorgulayabilirsiniz. Bkz. [mqInquireItemInfo](#).

Multi Torba içindeki bilgilerin değiştirilmesi

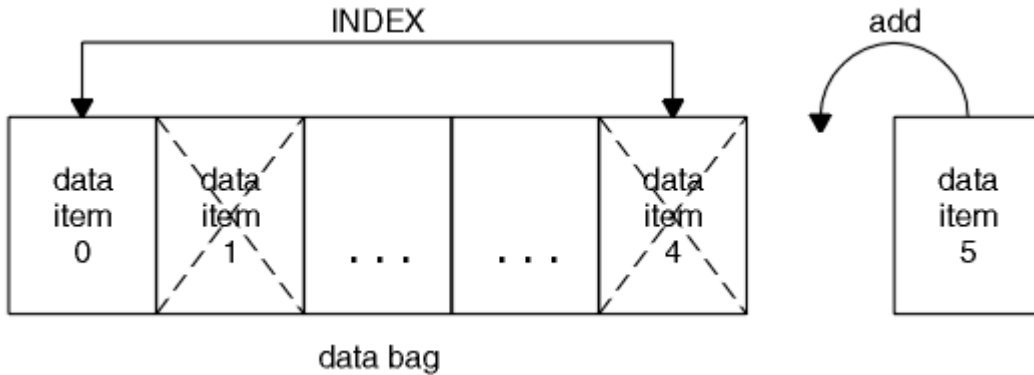
MQAI, mqSet* çağrılarını kullanarak bir çanta içinde bilgi değiştirmenize olanak sağlar. Şunları yapabilirsiniz:

1. Bir çanta içindeki veri öğelerini değiştirin. Dizin, değiştirilecek ögenin oluşumunu tanımlayarak bir parametrenin tek bir eşgörünümünün değiştirilmesini sağlar (bkz. [Şekil 6 sayfa 68](#)).



Şekil 6. Tek bir veri ögesini değiştirme

2. Belirtilen seçicinin var olan tüm oluşumlarını silin ve çantanın sonuna yeni bir oluşum ekleyin. (Bkz. [Şekil 7 sayfa 68](#).) Özel dizin değeri, bir parametrenin **tüm** eşgörünümlerinin değiştirilmesine izin verir.



Şekil 7. Tüm veri öğelerini değiştirme

Not: Dizin, paket içindeki ekleme sırasını korur, ancak diğer veri öğelerinin dizinlerini etkileyebilir.

mqSetTamsayı çağrısı, bir çanta içindeki tamsayı öğelerini değiştirmenizi sağlar. mqSetInteger64 çağrısı, 64 bitlik tamsayı öğelerini değiştirmenizi sağlar. mqSetIntegerFilter çağrısı, tamsayı süzgeci öğelerini değiştirmenizi sağlar. mqSetDizgi çağrısı, karakter dizgisi öğelerini değiştirmenizi sağlar. mqSetStringFilter çağrısı, dizgi süzgeci öğelerini değiştirmenizi sağlar. mqSetByteString çağrısı, bayt dizgisi öğelerini değiştirmenizi sağlar. mqSetByteStringSüzgeç çağrısı, bayt dizgisi süzgeç öğelerini değiştirmenizi sağlar. Alternatif olarak, belirtilen seçicinin var olan tüm yinlemelerini silmek ve çantanın sonuna yeni bir oluşum eklemek için bu çağrılar kullanılabilir. Veri ögesi bir kullanıcı ögesi ya da bir sistem ögesi olabilir.

Bu aramaların tam açıklaması için bkz:

- [mqSetTamsayı](#)
- [mqSetInteger64](#)
- [mqSetIntegerFilter](#)
- [mqSetDizgi](#)
- [mqSetStringFilter](#)
- [mqSetByteString](#)
- [mqSetByteStringSüzgeç](#)

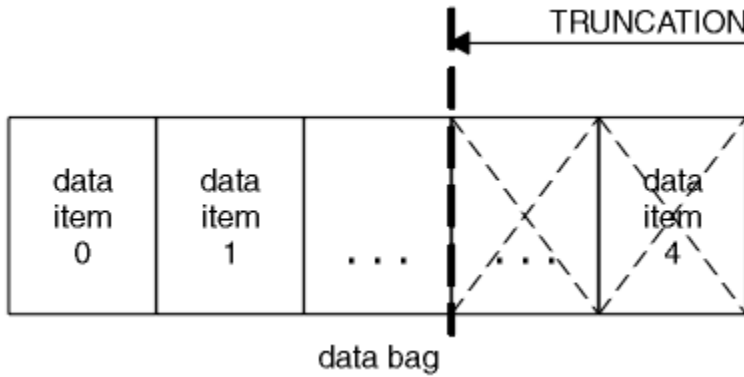
Multi *mqClearBag çağrısını kullanarak bir çantayı temizleme*

mqClearBag çağrısı, bir kullanıcı çantasından tüm kullanıcı öğelerini kaldırır ve sistem öğelerini ilk değerlerine döndürür. Çantanın içindeki sistem torbaları da silinir.

mqClearBag çağrısının tam açıklaması için bkz. [mqClearBag](#).

Multi *mqTruncateBag çağrısını kullanarak bir torbanın kesilmesi*

mqTruncateBag çağrısı, en son eklenen öğeden başlayarak öğeleri torbanın sonundan silerek bir kullanıcı çantasındaki kullanıcı öğelerinin sayısını azaltır. Örneğin, birden fazla ileti oluşturmak için aynı üstbilgi bilgileri kullanılırken kullanılabilir.

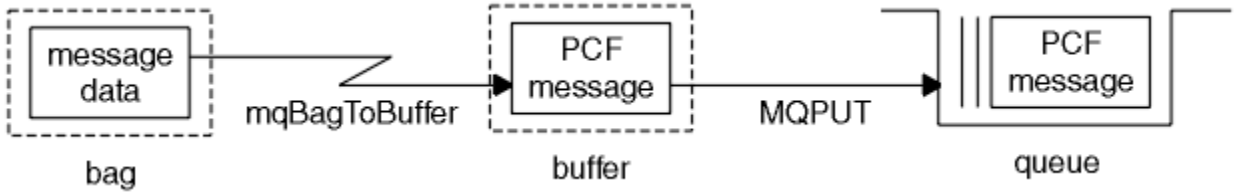


Şekil 8. Torbanın kesilmesi

mqTruncateBag çağrısının tam açıklaması için bkz. [mqTruncateBag](#).

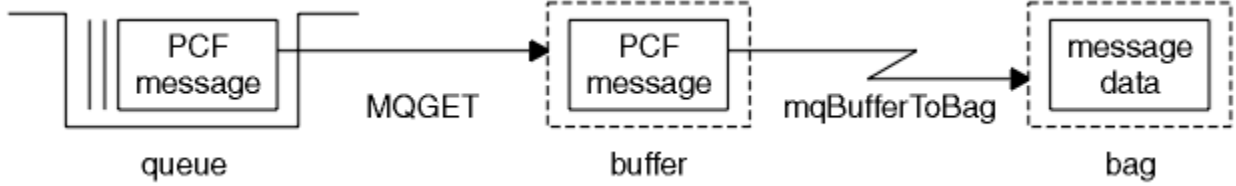
Multi *Torbaları ve arabellekleri dönüştürme*

Uygulamalar arasında veri göndermek için öncelikle mesaj verileri bir torbaya yerleştirilir. Daha sonra, çantadaki veriler mqBagToBuffer çağrısı kullanılarak bir PCF iletisine dönüştürülür. PCF iletisi, MQPUT çağrısı kullanılarak gerekli kuyruğa gönderilir. Bu, Şekil Şekil 9 sayfa 70' de gösterilmektedir. mqBagToBuffer çağrısının tam açıklaması için bkz. [mqBagToBuffer](#).



Şekil 9. Torbaların PCF iletilerine dönüştürülmesi

Veri almak için, ileti MQGET çağrısı kullanılarak bir arabelleğe alınır. Arabellekteki veriler, arabellek geçerli bir PCF iletisi içerdiğinde mqBufferToBag çağrısı kullanılarak bir çantaya dönüştürülür. Bu, Şekil Şekil 10 sayfa 70' de gösterilmektedir. mqBufferToBag çağrısının tam açıklaması için bkz. [mqBufferToBag](#).



Şekil 10. PCF iletilerinin torba biçimine dönüştürülmesi

Multi Veri öğeleri sayılıyor

mqCountÖge çağrısı, bir veri torbasında saklanan kullanıcı öğelerinin, sistem öğelerinin ya da her ikisinin sayısını sayar ve bu sayıyı döndürür. Örneğin, mqCountItems (Bag, 7, ...), seçici 7 olan çantadaki öğelerin sayısını döndürür. Öğeleri tek tek seçicilere, kullanıcı seçicilere, sistem seçicilere ya da tüm seçicilere göre sayabilir.

Not: Bu çağrı, çantadaki benzersiz seçicilerin sayısını değil, veri öğelerinin sayısını sayar. Bir seçici birden çok kez ortaya çıkabilir, bu nedenle çantada veri öğelerinden daha az benzersiz seçici olabilir.

mqCountÖge çağrısının tam açıklaması için bkz. [mqCountÖge](#).

Multi Veri öğelerini silme

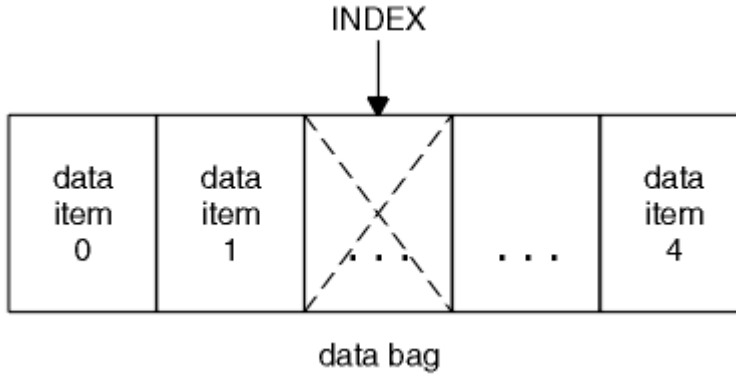
Torbalardan çeşitli şekillerde öğeleri silebilirsiniz. Şunları yapabilirsiniz:

- Bir torbadan bir ya da daha fazla kullanıcı öğesini çıkarın. Ayrıntılı bilgi için bkz. [“mqDeleteÖge çağrısını kullanarak veri öğelerini bir çantadan silme”](#) sayfa 70.
- Bir torbadaki tüm kullanıcı öğelerini silin, yani bir çantayı temizleyin. Ayrıntılı bilgi için bkz. [“mqClearBag çağrısını kullanarak bir çantayı temizleme”](#) sayfa 69.
- Bir torbanın sonundaki kullanıcı öğelerini silin, yani bir torbayı kısaltın. Ayrıntılı bilgi için bkz. [“mqTruncateBag çağrısını kullanarak bir torbanın kesilmesi”](#) sayfa 69.

Multi mqDeleteÖge çağrısını kullanarak veri öğelerini bir çantadan silme

mqDeleteÖge çağrısı, bir çantadan bir ya da daha fazla kullanıcı öğesini kaldırır. Dizin aşağıdakilerden birini silmek için kullanılır:

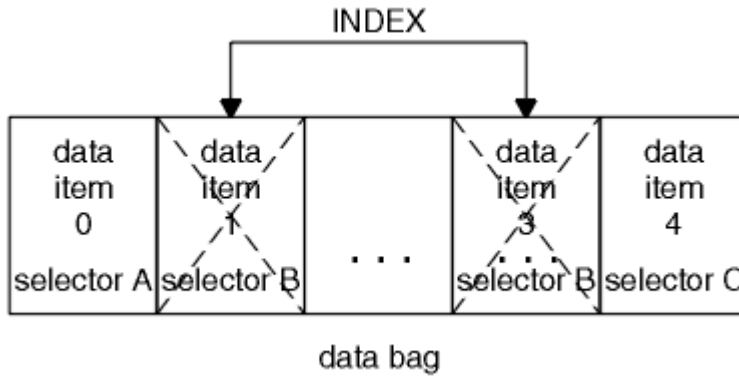
1. Belirtilen seçicinin tek bir oluşumu. (Bkz. [Şekil 11 sayfa 71.](#))



Şekil 11. Tek bir veri ögesini silme

veya

2. Belirtilen seçicinin tüm geçişleri. (Bkz. Şekil 12 sayfa 71.)



Şekil 12. Tüm veri öğeleri siliniyor

Not: Dizin, paket içindeki ekleme sırasını korur, ancak diğer veri öğelerinin dizinlerini etkileyebilir. Örneğin, mqDeleteÖge çağrısı, dizinler silinen öğeden kalan boşluğu dolduracak şekilde yeniden düzenlendiğinden, silinen öğeyi izleyen veri öğelerinin dizin değerlerini korumaz.

mqDeleteÖge çağrısının tam açıklaması için bkz. [mqDeleteÖgesi](#).

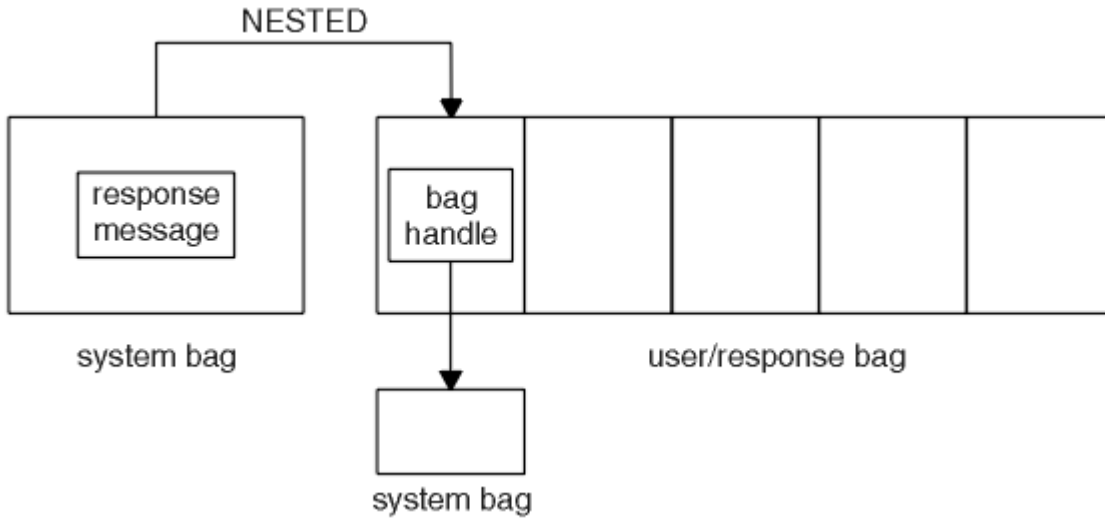
Multi mqExecute çağrısı kullanarak qm komut sunucusuna yönetim komutları gönderme

Bir veri torbası oluşturulup doldurulduğunda, mqExecute çağrısıyla bir kuyruk yöneticisinin komut sunucusuna yönetim komut iletisi gönderilebilir. Bu, komut sunucusuyla değiş tokuş işlemlerini gerçekleştirir ve yanıtları bir torba içinde döndürür.

Veri torbanızı oluşturup doldurduktan sonra, bir kuyruk yöneticisinin komut sunucusuna bir yönetim komut iletisi gönderebilirsiniz. Bunu yapmanın en kolay yolu, mqExecute çağrısıyla yapmaktır. mqExecute çağrısı, kalıcı olmayan bir ileti olarak bir denetim komut iletisi gönderir ve yanıtları bekler. Yanıtlar bir yanıt çantasında döndürülür. Bunlar, örneğin, birkaç IBM MQ nesnesiyle ya da bir dizi PCF hata yanıtı ile ilgili öznelikle ilgili bilgiler içerebilir. Bu nedenle, yanıt çantası yalnızca bir dönüş kodu içerebilir ya da içiçe yerleştirilmiş torbaları içerebilir.

Yanıt iletileri, sistem tarafından oluşturulan sistem torbalarına yerleştirilir. Örneğin, nesnelerin adlarıyla ilgili sorgular için, bu nesne adlarını tutmak için bir sistem çantası oluşturulur ve çanta kullanıcı çantasına eklenir. Daha sonra bu çantaların tanıtıcıları yanıt çantasına yerleştirilir ve iç içe geçmiş torbaya MQHA_BAG_HANDLE seçicisi tarafından erişilebilir. Sistem torbası, silinmezse, yanıt torbası silinceye kadar depoda kalır.

İç içe yerleştirme kavramı Şekil 13 sayfa 72 içinde gösterilmektedir.



Şekil 13. İç içe yerleştirme

mqExecute çağrısına giriş olarak şunları sağlamanız gerekir:

- MQI bağlantı tanıtıcısı.
- Yürütülecek komut. Bu değer MQCMD_ * değerlerinden biri olmalıdır.

Not: Bu değer MQAI tarafından tanınmazsa, değer yine de kabul edilir. Ancak, çantaya değer eklemek için mqAddSorgu çağrısı kullanıldıysa, bu parametre MQAI tarafından tanınan bir INQUIRE komutu olmalıdır. Yani, değiştirge MQCMD_INQUIRE_ * biçiminde olmalıdır.

- İsteğe bağlı olarak, aramanın işlenmesini denetleyen seçenekleri içeren bir çanta tutamacı. Bu, MQAI 'nın her bir yanıt iletilisini beklemesi için milisaniye cinsinden süre üst sınırını belirtebileceğiniz yerdir.
- Verilecek yönetim komutunun ayrıntılarını içeren yönetim çantasının tanıtıcısı.
- Yanıt iletilerini alan yanıt çantasının tanıtıcısı.

Aşağıdaki tutamaçlar isteğe bağlıdır:

- Denetim komutunun yerleştirileceği kuyruğun nesne tanıtıcısı.

Nesne tanıtıcısı belirtilmezse, denetim komutu SYSTEM.ADMIN.COMMAND.QUEUE . Bu varsayılandır.

- Yanıt iletilerinin yerleştirileceği kuyruğun nesne tanıtıcısı.

Yanıt iletilerini, MQAI tarafından otomatik olarak yaratılan dinamik bir kuyruğa yerleştirmeyi seçebilirsiniz. Yaratılan kuyruk yalnızca çağrı süresi boyunca var olur ve mqExecute çağrısından çıkışta MQAI tarafından silinir.

mqExecute çağrısının örnek kullanımları için bkz. [Örnek kod](#)

REST API kullanarak yönetim

Kuyruk yöneticileri ve kuyrukları, Managed File Transfer araçları ve aktarımları gibi IBM MQ nesnelerini yönetmek için administrative REST API olanağını kullanabilirsiniz. Bilgiler, administrative REST API ' e JSON biçiminde gönderilir ve bu bilgilerden alınır. Bu RESTful API 'ler, IBM MQ yönetimini sık kullanılan DevOps ve otomasyon araçlarına yerleştirmenize yardımcı olabilir.

Başlamadan önce

Not: **V9.3.5** administrative REST API bağımsız bir IBM MQ Web Server kuruluşunda kullanılamaz. administrative REST API ürününü çalıştıran IBM MQ bileşenine ilişkin kuruluş seçenekleri hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API](#).

Kullanılabilir REST kaynakları hakkında başvuru bilgileri için bkz. [administrative REST API başvurusu](#).

Yordam

- [“administrative REST API ile çalışmaya başlama” sayfa 73](#)
- [“administrative REST API olanağının kullanılması” sayfa 76](#)
- [“REST API kullanılarak uzaktan yönetim” sayfa 78](#)
- [“REST API zaman damgaları” sayfa 82](#)
- [“REST API Hata işleme” sayfa 82](#)
- [“REST API Keşif” sayfa 84](#)
- [“REST API ulusal dil desteği” sayfa 86](#)


administrative REST API ile çalışmaya başlama


administrative REST API ile hızlı bir şekilde çalışmaya başlayın ve bir kuyruk oluşturmak, güncellemek, görüntülemek ve silmek için cURL komutunu kullanarak birkaç örnek isteği deneyin.


Başlamadan önce

administrative REST API ürününü kullanmaya başlamanız için bu görevdeki örnekler aşağıdaki gereksinimlere sahip olur:

- Bu örnekler, sistemdeki kuyruk yöneticileriyle ilgili bilgileri görüntülemek ve bir kuyruk yaratmak, güncellemek, görüntülemek ve silmek için cURL komutunu kullanır. Bu nedenle, bu görevi tamamlamak için sisteminizde cURL kurulu olmalıdır.
- Bu görevi tamamlamak için, **dspmweb** komutunu kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:


–  z/OS' da **dspmweb** komutunu çalıştırma ve `mwebuser.xml` dosyasına yazma erişiminiz olmalıdır.

–  Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.

–  IBM i üzerinde, komutlar QSHLL içinde çalıştırılmalıdır.

Yordam

1. `mweb` sunucusunu administrative REST API, administrative REST API for MFT, messaging REST API ya da IBM MQ Console tarafından kullanılmak üzere yapılandırdığınızdan emin olun. `mweb` sunucusunu temel bir kayıt dosyasıyla yapılandırma hakkında daha fazla bilgi için bkz. [mweb sunucusu için temel yapılandırma](#).

2.  z/OS' ta, **dspmweb** komutunu kullanabilmek için `WLP_USER_DIR` ortam değişkenini ayarlayın. Şu komutu girerek değişkeni `mweb` sunucusu yapılandırmanızı gösterecek şekilde ayarlayın:

```
export WLP_USER_DIR=WLP_user_directory
```

burada `WLP_user_directory` , `crtmweb` ' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için `mweb` sunucusu yaratılması başlıklı konuya bakın.

3. Aşağıdaki komutu girerek REST API URL 'sini belirleyin:

```
dspmweb status
```

Aşağıdaki adımlardaki örnekler, REST API URL adresinizin varsayılan URL `https://localhost:9443/ibmmq/rest/v1/` olduğunu varsayar. URL 'iniz varsayılan değerden farklıysa, aşağıdaki adımlarda URL 'yi değiştirin.

4. mqadmin kullanıcısıyla temel kimlik doğrulamasını kullanarak qmgr kaynağında GET isteğini deneyin:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/qmgr -X GET -u mqadmin:mqadmin
```

5. mqsc kaynağını kullanarak bir kuyruk yaratın, görüntüleyin, değiştirin ve silin:

Bu örnek, QM1kuyruk yöneticisini kullanır. Aynı adı taşıyan bir kuyruk yöneticisi yaratın ya da sisteminizde var olan bir kuyruk yöneticisini kullanın.

a) Yerel kuyruğu yaratmak için mqsc kaynağında POST isteğinde bulunun:

İsteğin gövdesinde, yeni kuyruğun adı Q1olarak ayarlanır. Temel kimlik doğrulaması kullanılır ve cURL REST isteğinde isteğe bağlı bir ibm-mq-rest-csrf-token HTTP üstbilgisi ayarlanır. POST, PATCH ve DELETE istekleri için bu ek üstbilgi gereklidir:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "define", "qualifier": "qlocal", "name": "Q1"}'
```

b) “5.a” sayfa 74. adımda oluşturulan yerel kuyruğu görüntülemek için mqsc kaynağında POST isteğinde bulunun:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "display", "qualifier": "qlocal", "name": "Q1"}'
```

c) mqsc kaynağında, kuyruğun açıklamasını güncellemek için bir POST isteği gönderin:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "alter", "qualifier": "qlocal", "name": "Q1", "parameters": {"descr": "new description"}'}
```

d) Yeni kuyruk açıklamasını görüntülemek için mqsc kaynağında bir POST isteği gönderin. Yanıtın açıklama alanını içereceği şekilde istek gövdesinde **responseParameters** özniteliğini belirtin:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "display", "qualifier": "qlocal", "name": "Q1", "responseParameters": [{"descr"}]}'
```

e) Kuyruğu silmek için mqsc kaynağında POST isteğinde bulunun:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "delete", "qualifier": "qlocal", "name": "Q1"}'
```

f) Kuyruğun silindiğini kanıtlamak için mqsc kaynağında POST isteğinde bulunun:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "display", "qualifier": "qlocal", "name": "Q1"}'
```

Sonraki adım

- Örnekler, isteği korumak için temel kimlik doğrulamasını kullanır. Bunun yerine, belirteç tabanlı kimlik doğrulamasını ya da istemci tabanlı kimlik doğrulamasını kullanabilirsiniz. Daha fazla bilgi için bkz. [REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma](#) ve [REST API ile belirteç tabanlı kimlik doğrulamasını kullanma](#).
- administrative REST API olanağını kullanma ve sorgu değiştirgeleriyle URL oluşturma hakkında daha fazla bilgi edinin: [“administrative REST API olanağının kullanılması” sayfa 76](#).



- Kullanılabilir administrative REST API kaynakları ve kullanılabilir tüm isteğe bağlı sorgu parametreleri için başvuru bilgilerine göz atın: [administrative REST API reference](#).
- Uzak sistemlerde IBM MQ nesnelerini yönetmek için administrative REST API ' in nasıl kullanılacağını öğrenin: “REST API kullanılarak uzaktan yönetim” sayfa 78.
- administrative REST API ürününü MFT ile nasıl kullanacağınızı öğrenin: “MFT için REST API ile çalışmaya başlama” sayfa 75.
- IBM MQ ileti sistemi için bir RESTful arabirimi olan messaging REST API arabirimini keşfedin: [REST API kullanılarak ileti alışverişi](#).
- Tarayıcı tabanlı bir GUI olan IBM MQ Console dosyasını keşfedin: “IBM MQ Console kullanarak yönetim” sayfa 89.

MFT için REST API ile çalışmaya başlama

administrative REST API for Managed File Transfer ile hızlı bir şekilde çalışmaya başlayın ve MFT aracı durumunu görüntülemek ve aktarımların bir listesini görüntülemek için birkaç örnek istek deneyin.

Başlamadan önce

- Bu örneklerde, aktarımların listesini görüntülemek ve MFT aracı durumunu görüntülemek üzere REST istekleri göndermek için cURL kullanılır. Bu nedenle, bu görevi tamamlamak için sisteminizde cURL kurulu olmalıdır.
- Bu görevi tamamlamak için, **dspmweb** komutunu kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

-  z/OS' da **dspmweb** komutunu çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
-  Diğer tüm işletim sistemlerinde ayrıcalıklı kullanıcı olmanız gerekir.

Yordam

1. MFT için administrative REST API için mqweb sunucusunun yapılandırıldığından emin olun:

- mqweb sunucusunu administrative REST API, administrative REST API for MFT, messaging REST API ya da IBM MQ Console tarafından kullanılmak üzere yapılandırduğunuzdan emin olun. mqweb sunucusunu temel bir kayıt dosyasıyla yapılandırma hakkında daha fazla bilgi için bkz. [mqweb sunucusu için temel yapılandırma](#).
- mqweb sunucusu yapılandırıldıysa, MFT için administrative REST API ' yi etkinleştirmek üzere [mqweb sunucusu için temel yapılandırma](#) adınının 8. adımının tamamlandığından emin olun.

2. 

z/OS' ta, **dspmweb** komutunu kullanabilmek için WLP_USER_DIR ortam değişkenini ayarlayın. Şu komutu girerek değişkeni mqweb sunucusu yapılandırmanızı gösterecek şekilde ayarlayın:

```
export WLP_USER_DIR=WLP_user_directory
```

burada *WLP_user_directory* , crtmqweb' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için [mqweb sunucusu yaratılması](#) başlıklı konuya bakın.

3. Aşağıdaki komutu girerek REST API URL 'sini belirleyin:

```
dspmweb status
```

Aşağıdaki adımlardaki örnekler, REST API URL adresinizin varsayılan URL <https://localhost:9443/ibmmq/rest/v1/> olduğunu varsayar. URL 'iniz varsayılan değerden farklıysa, aşağıdaki adımlarda URL' yi değiştirin.

4. Ad, tip ve durum da içinde olmak üzere tüm araçlara ilişkin temel ayrıntıları döndürmek için agent kaynağında GET isteğinde bulunun:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/mft/agent/ -X GET -u mftadmin:mftadmin
```

5. **fteCreateTransfer** komutunu kullanarak görüntülenecek bazı aktarımlar oluşturun.

mqweb sunucusu, aktarımlarla ilgili bilgileri önbelleğe alır ve bir istek yapıldığında bu bilgileri döndürür. Bu önbellek, mqweb sunucusu yeniden başlatıldığında ilk durumuna getirilir. Başlatılan görevin çıktısına bakarak console.log ve messages.log dosyalarını görüntüleyerek ya da z/OSüzerinde sunucunun yeniden başlatılıp başlatılmadığını görebilirsiniz.

6. mqweb sunucusunun başlatılmasından bu yana yapılan en çok dört aktarımın ayrıntılarını döndürmek için transfer kaynağında GET isteğinde bulunun:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/mft/transfer?limit=4 -X GET -u mftadmin:mftadmin
```

Sonraki adım

- Örnekler, isteğin güvenliğini sağlamak için temel kimlik doğrulamasını kullanır. Bunun yerine belirteç tabanlı kimlik doğrulamasını ya da istemci tabanlı kimlik doğrulamasını kullanabilirsiniz. Daha fazla bilgi için bkz. [REST API ile belirteç tabanlı kimlik doğrulamasını kullanma](#) ve [REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma](#).
- administrative REST API olanağını kullanma ve sorgu değiştiriceleriyle URL oluşturma hakkında daha fazla bilgi edinin: [“administrative REST API olanağının kullanılması” sayfa 76](#).
- Kullanılabilir administrative REST API for MFT kaynaklarına ve kullanılabilir tüm isteğe bağlı sorgu değiştiricelerine ilişkin başvuru bilgilerine göz atın: [administrative REST API reference](#).
- IBM MQ ileti sistemi için bir RESTful arabirimi olan messaging REST API arabirimini keşfedin: [REST API kullanılarak ileti alışverişi](#).
- Tarayıcı tabanlı bir GUI olan IBM MQ Console dosyasını keşfedin: [“IBM MQ Console kullanarak yönetim” sayfa 89](#).

administrative REST API olanağının kullanılması

administrative REST API kullandığınızda, kuyruk yöneticileri ya da kuyruklar gibi çeşitli IBM MQ nesnelerini gösteren URL ' lerde HTTP yöntemlerini çağırırsınız. HTTP yöntemi, örneğin POST, URL ile gösterilen nesne üzerinde gerçekleştirilecek işlemin tipini gösterir. HTTP yönteminin bilgi yükünün bir parçası olarak ya da sorgu parametrelerinde kodlanmış olarak işleme ilgili daha fazla bilgi JSON ' da sağlanabilir. İşlemin gerçekleştirilmesiyle ilgili bilgiler, HTTP yanıtının gövdesi olarak döndürülebilir.

Başlamadan önce

administrative REST API programını kullanmadan önce şunları göz önünde bulundurun:

- administrative REST API kullanabilmek için mqweb sunucusuyla kimlik doğrulaması yapmalısınız. HTTP temel kimlik doğrulamasını, istemci sertifikası kimlik doğrulamasını ya da belirteç tabanlı kimlik doğrulamasını kullanarak kimlik doğrulaması yapabilirsiniz. Bu kimlik doğrulama yöntemlerinin nasıl kullanılacağı hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API güvenlik](#).
- REST API büyük ve küçük harfe duyarlıdır. Örneğin, aşağıdaki URL adresinde HTTP GET işlemi, kuyruk yöneticisi qmgr1 olarak adlandırılırsa bilgi görüntüleyemez.

```
/ibmmq/rest/v1/admin/qmgr/QMGR1
```

- IBM MQ nesne adlarında kullanılacak karakterlerin tümü doğrudan URL içinde kodlanamaz. Bu karakterleri doğru kodlamak için uygun URL kodlamasını kullanmanız gerekir:
 - Eğik çizgi, /, %2F olarak kodlanmalıdır.
 - Yüzde işareti, %, %25 olarak kodlanmalıdır.

- Bazı tarayıcıların davranışı nedeniyle, nesneleri yalnızca nokta ya da eğik çizgi karakterleri kullanarak adlandırmayın.

Bu görev hakkında

Bir nesne üzerinde bir eylem gerçekleştirmek için REST API kullandığınızda, öncelikle o nesneyi göstermek için bir URL oluşturmanız gerekir. Her URL , isteğin gönderileceği anasistem adını ve kapıyı açıklayan bir önekle başlar. URL ' nin geri kalanı, kaynak olarak bilinen belirli bir nesneyi ya da nesne kümesini tanımlar.

Kaynak üzerinde gerçekleştirilecek işlem, URL ' nin sorgu değiştiricilerine gerek duyup duymadığını tanımlar. Ayrıca, kullanılan HTTP yöntemini ve URL' ye ek bilgilerin mi gönderileceğini, yoksa JSON biçiminde mi döndürüleceğini de tanımlar. Ek bilgiler HTTP isteğinin bir kısmını oluşturabilir ya da HTTP yanıtının bir parçası olarak döndürülebilir.

URL' yi oluşturduktan ve HTTP isteğini göndermek için isteğe bağlı bir JSON bilgi yükü oluşturduktan sonra, HTTP isteğini IBM MQadresine gönderebilirsiniz. İsteği, seçtiğiniz programlama dilinde yerleşik olan HTTP uygulamasını kullanarak gönderebilirsiniz. İstekleri, cURLgibi komut satırı araçlarını ya da bir web tarayıcısını ya da web tarayıcısı eklentilerini kullanarak da gönderebilirsiniz.

Önemli: En az “1.a” sayfa 77 ve “1.b” sayfa 77adımlarını gerçekleştirmeniz gerekir.

Yordam

1. URL' yi oluşturun:

a) Aşağıdaki komutu girerek URL önekini belirleyin:

```
dspmweb status
```

Kullanmak istediğiniz URL , /ibmmq/rest/ sözcük grubunu içerir.

b) Kaynağı URL yoluna ekleyin.

Aşağıdaki IBM MQ kaynakları kullanılabilir:

- [/admin/installation](#)
- [/admin/qmgr](#)
- [/denetimci/kuyruk](#)
- [/admin/abonelik](#)
- [/yönetim/kanal](#)
- [/action/qmgr/{qmgrname}/mqsc](#)

Aşağıdaki Managed File Transfer kaynakları kullanılabilir:

- [/admin/agent](#)
- [/admin/transfer](#)
- [/admin/monitor](#)

Örneğin, kuyruk yöneticileriyle etkileşimde bulunmak için aşağıdaki URL' yi oluşturmak üzere URL öneğine /qmgr ekleyin:

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr
```

c) İsteğe bağlı: URL' ye isteğe bağlı ek yol bölümleri ekleyin.

Her bir nesne tipine ilişkin başvuru bilgilerinde, isteğe bağlı bölümler URL ' de { }ögesini çevreleyen kaşlı ayrıçlarla tanımlanabilir.

Örneğin, aşağıdaki URL'yi yaratmak için QM1 kuyruk yöneticisi adını URL ' ye ekleyin:

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr/QM1
```

d) İsteğe bağlı: URL' ye isteğe bağlı bir sorgu parametresi ekleyin.

Soru işareti ekle,?, değişken adı, eşittir imi = ve URL için bir değer ya da değer listesi.

Örneğin, QM1kuyruk yöneticisinin tüm özneliklerini istemek için aşağıdaki URL adresini yaratın:

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr/QM1?attributes=*
```

e) URL' ye isteğe bağlı başka sorgu parametreleri ekleyin.

URL'ye & ve işareti ekleyin ve adım d' yi yineleyin.

2. URL üzerinde ilgili HTTP yöntemini çağırın. İsteğe bağlı JSON bilgi yükünü belirtin ve kimlik doğrulaması için uygun güvenlik kimlik bilgilerini sağlayın. Örneğin:

- Seçtiğiniz programlama dilinin HTTP/REST uygulamasını kullanın.
- REST istemci tarayıcısı eklentisi ya da cURL gibi bir araç kullanın.

REST API kullanılarak uzaktan yönetim

Uzak kuyruk yöneticilerini ve bu kuyruk yöneticileriyle ilişkili IBM MQ nesnelere denetlemek için REST API komutunu kullanabilirsiniz. Bu uzak denetim, mqweb sunucusuyla aynı IBM MQ kuruluşunda değil, aynı sistemde bulunan kuyruk yöneticilerini içerir. Bu nedenle, tüm IBM MQ ağınızı mqweb sunucusunu çalıştıran tek bir kuruluşla yönetmek için REST API komutunu kullanabilirsiniz. Uzak kuyruk yöneticilerini denetlemek için, administrative REST API ağ geçidini, mqweb sunucusu ile aynı kuruluşta en az bir kuyruk yöneticisinin ağ geçidi kuyruk yöneticisi olarak işlev görmesi için yapılandırmanız gerekir. Daha sonra, belirtilen denetim işlemi gerçekleştirmek için REST API kaynak URL ' de uzak kuyruk yöneticisini belirtebilirsiniz.

Başlamadan önce

administrative REST API ağ geçidini devre dışı bırakarak uzaktan yönetimi önleyebilirsiniz. Daha fazla bilgi için [administrative REST API ağ geçidini yapılandırmabaşlıklı konuya](#) bakın.

administrative REST API ağ geçidini kullanmak için aşağıdaki koşulların karşılanması gerekir:

- mqweb sunucusu yapılandırılmalı ve başlatılmalıdır. mqweb sunucusunu yapılandırma ve başlatma hakkında daha fazla bilgi için bkz. "[administrative REST API ile çalışmaya başlama](#)" sayfa 73.
- Ağ geçidi kuyruk yöneticisi olarak yapılandırmak istediğiniz kuyruk yöneticisi, mqweb sunucusu ile aynı kuruluşta olmalıdır.
- Denetlemek istediğiniz uzak kuyruk yöneticisi IBM MQ 8.0 ya da üstü olmalıdır.
- İsteğinizde belirtilen özneliklerin, isteği gönderdiğiniz sistem için geçerli olduğundan emin olmanız gerekir. Örneğin, ağ geçidi kuyruk yöneticisi Windows üzerindeyse ve uzak kuyruk yöneticisi z/OS üzerindeyse, queue kaynağında HTTP GET isteği için dataCollection.statistics özneliğinin döndürülmesini isteyemezsiniz.
- İsteğinizde belirtilen özneliklerin, isteği gönderdiğiniz IBM MQ düzeyi için geçerli olduğundan emin olmanız gerekir. Örneğin, uzak kuyruk yöneticisi IBM MQ 8.0 çalıştırıyorsa, queue kaynağında bir HTTP GET isteği için extended.enableMediaImageOperations özneliğinin döndürülmesini isteyemezsiniz.
- Desteklenen şu REST kaynaklarından birini kullanmalısınız:

- /queue
- /subscription
- /channel
- /mqsc
- /qmgr

/qmgr kaynağı, uzak bir kuyruk yöneticisini sorguladığınızda özneliklerin yalnızca bir altkumesini döndürür: name,

status.started, status.channelInitiatorState, status.ldapConnectionState, status.connectionCount ve status.publishSubscribeState.

Bu görev hakkında

Uzak kuyruk yöneticilerini yönetmek üzere administrative REST API ağ geçidini kullanmak için kuyruk yöneticilerini uzak denetim için hazırlamanız gerekir. Yani, ağ geçidi kuyruk yöneticisi ile uzak kuyruk yöneticisi arasında iletim kuyruklarını, dinleyicileri ve gönderen ve alıcı kanallarını yapılandırmanız gerekir. Daha sonra, URLkaynağında kuyruk yöneticisini belirterek uzak kuyruk yöneticisine bir REST isteği gönderebilirsiniz. Ağ geçidi kuyruk yöneticisi, mqRestGatewayQmgr öznitelikli ağ geçidi kuyruk yöneticisinin adına ayarlamak için **setmqweb** komutu kullanılarak ya da istekle birlikte gönderilen bir üstbilgiye ağ geçidi kuyruk yöneticisinin adı gönderilerek belirtilir. İstek, ağ geçidi kuyruk yöneticisi aracılığıyla uzak kuyruk yöneticisine gönderilir. Yanıt, ağ geçidi kuyruk yöneticisi olarak kullanılan kuyruk yöneticisini gösteren bir üstbilgiyle döndürülür.

Yordam

1. Denetlemek istediğiniz ağ geçidi kuyruk yöneticisi ile uzak kuyruk yöneticileri arasındaki iletişimi yapılandırın. Bu yapılandırma adımları, runmqsc ve PCF tarafından uzak yönetimi yapılandırmak için gereken adımlarla aynıdır.
Bu adımlarla ilgili daha fazla bilgi için bkz. [“Uzak denetim için kuyruk yöneticilerinin yapılandırılması” sayfa 213.](#)
2. Uzak kuyruk yöneticilerinin güvenliğini yapılandır:
 - a) Uzak kuyruk yöneticisinin çalıştığı sistemde ilgili kullanıcı kimliklerinin var olduğunu doğrulayın. Uzak sistemde bulunması gereken kullanıcı kimliği, REST API kullanıcısının rolüne bağlıdır:
 - REST API kullanıcısı MQWebAdmin ya da MQWebAdminRO grubundaya, mqweb sunucusunu başlatan kullanıcı kimliği uzak sistemde var olmalıdır. IBM MQ Appliance üzerinde, mqweb sunucusunu başlatan kullanıcı: mqsystem.
 - REST API kullanıcısı MQWebUser grubundaya, bu REST API kullanıcı kimliğinin uzak sistemde var olması gerekir.
 - b) Uzak kuyruk yöneticisindeki uygun REST API kaynaklarına erişmek için ilgili kullanıcı kimliklerine gerekli yetki düzeylerinin atandığından emin olun:
 - İletileri SYSTEM.ADMIN.COMMAND.QUEUE' e koyma yetkisi.
 - İletileri SYSTEM.REST.REPLY.QUEUE' e koyma yetkisi.
 - Uzak denetim için tanımlanan iletim kuyruklarına erişim yetkisi.
 - Kuyruk yöneticisi özniteliklerini görüntüleme yetkisi.
 - REST isteklerini gerçekleştirme yetkisi. Daha fazla bilgi için, [REST API kaynak başvurusu konularının Güvenlik gereksinimleri bölümüne](#) bakın.
3. Ağ geçidi olarak kullanılacak yerel kuyruk yöneticisini yapılandırın. Varsayılan bir ağ geçidi kuyruk yöneticisi yapılandırabilir, HTTP üstbilgisinde ağ geçidi kuyruk yöneticisini belirtebilir ya da her iki yaklaşımın bir birleşimini kullanabilirsiniz:
 - **setmqweb** komutunu kullanarak varsayılan bir ağ geçidi kuyruk yöneticisini yapılandırın:

```
setmqweb properties -k mqRestGatewayQmgr -v qmgrName
```

Burada *qmgrName* , ağ geçidi kuyruk yöneticisinin adıdır.
Bu ağ geçidi kuyruk yöneticisi, aşağıdaki deyimlerin her ikisi de doğru olduğunda kullanılır:
 - Bir REST isteğinin `ibm-mq-rest-gateway-qmgr` üstbilgisinde kuyruk yöneticisi belirtilmedi.
 - REST API Kaynak URL ' de belirtilen kuyruk yöneticisi yerel bir kuyruk yöneticisi değil.
 - HTTP üstbilgisini `ibm-mq-rest-gateway-qmgr` ağ geçidi kuyruk yöneticisinin adına ayarlayarak her REST isteğinde ağ geçidi kuyruk yöneticisini yapılandırın.
4. Denetlemek istediğiniz uzak kuyruk yöneticisinin adını URLkaynağında ekleyin.

Örneğin, uzak kuyruk yöneticisinden remoteQMkuyruk listesini almak için aşağıdaki URLadresini kullanın:

```
https://localhost:9443/ibmmq/rest/v1/admin/qmgr/remoteQM/queue
```

Sonuçlar

REST yanıtıyla birlikte bir `ibm-mq-rest-gateway-qmgr` üstbilgisi döndürülür. Bu üstbilgi, ağ geçidi kuyruk yöneticisi olarak hangi kuyruk yöneticisinin kullanıldığını belirtir.

Uzak kuyruk yöneticilerini yönetmek için administrative REST API komutunu kullanmakta zorlanıyorsanız:

- Uzak kuyruk yöneticisinin çalışıp çalışmadığını denetleyin.
- Komut sunucusunun uzak sistemde çalışıp çalışmadığını denetleyin.
- Kanal bağlantı kesme aralığının süresinin dolmadığını denetleyin. Örneğin, bir kanal başlatıldıysa, ancak bir süre sonra kapatıldıysa. Kanalları el ile başlattığınızda bu özellikle önemlidir.

Örnek

Aşağıdaki örnekte, iki makinede üç IBM MQ kuruluşu vardır. Machine 1 üzerinde bir Installation 1 ve bir Installation 2 vardır. Machine 2 üzerinde bir Installation 3 vardır. Installation 1 için bir mqweb sunucusu yapılandırıldı. Her kuruluşta tek bir kuyruk yöneticisi vardır ve bu kuyruk yöneticileri uzak denetim için yapılandırılır. Yani, aşağıdaki dinleyiciler, kanallar ve kuyruklar yapılandırılır ve başlatılır:

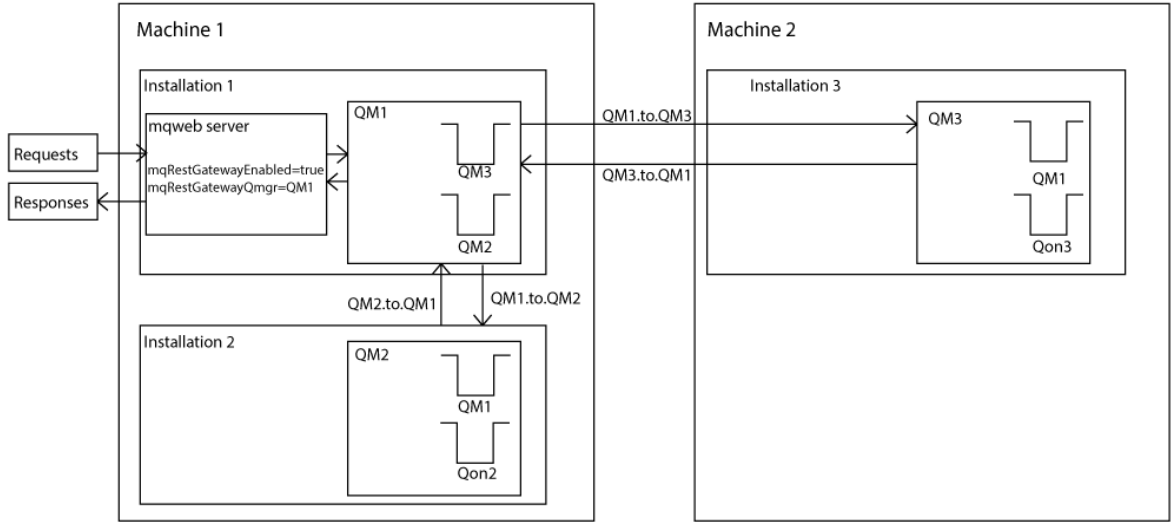
- QM1kuyruk yöneticisinde, Installation 1 içinde, Machine 1 üzerinde:
 - Gönderen kanal QM1.to.QM2
 - Alıcı kanal QM2.to.QM1
 - Gönderen kanal QM1.to.QM3
 - Alıcı kanal QM3.to.QM1
 - İletim kuyruğu QM2
 - İletim kuyruğu QM3
 - 1414 kapısında bir dinleyici yapılandırıldı
- QM2kuyruk yöneticisinde, Installation 2 içinde, Machine 1 üzerinde:
 - Gönderen kanalı QM2.to.QM1
 - Alıcı kanal QM1.to.QM2
 - İletim kuyruğu QM1
 - 1415 kapısında bir dinleyici yapılandırıldı
- QM3kuyruk yöneticisinde, Installation 3 içinde, Machine 2 üzerinde:
 - Gönderen kanalı QM3.to.QM1
 - Alıcı kanal QM1.to.QM3
 - İletim kuyruğu QM1
 - Varsayılan dinleyici

Qon2 kuyruğu QM2üzerinde tanımlanır ve Qon3 kuyruğu QM3üzerinde tanımlanır.

mquser kullanıcısı her iki makinede de tanımlanır, REST API içinde MQWebAdmin rolü verilir ve her kuyruk yöneticisinde uygun kuyruklara erişim yetkisi verilir.

setmqweb komutu, QM1 kuyruk yöneticisini varsayılan ağ geçidi kuyruk yöneticisi olarak yapılandırmak için kullanılır.

Aşağıdaki çizge bu yapılandırmayı göstermektedir:



Şekil 14. REST API kullanılarak uzaktan yönetim için örnek yapılandırma şeması.

Şu REST isteği mqweb sunucusuna gönderildi:

```
GET https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM2/queue?
attributes=general.isTransmissionQueue
```

Şu yanıt alındı:

```
{
  "queue" :
  [ {
    "general": {
      "isTransmissionQueue": true
    },
    "name": "QM1",
    "type": "local"
  },
  {
    "general": {
      "isTransmissionQueue": false
    },
    "name" : "Qon2",
    "type" : "local"
  }
]
```

Şu REST isteği mqweb sunucusuna gönderildi:

```
GET https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM3/queue?
attributes=general.isTransmissionQueue,general.description
```

Şu yanıt alındı:

```
{
  "queue" :
  [ {
    "general": {
      "isTransmissionQueue": true,
      "description": "Transmission queue for remote admin."
    },
    "name": "QM1",
    "type": "local"
  },
  {
    "general": {
      "isTransmissionQueue": false,
      "description": "A queue on QM3."
    }
  }
]
```

```
    },
    "name" : "Qon3",
    "type" : "local"
  }
}
```

REST API zaman damgaları

Tarih ve saat bilgileri administrative REST API tarafından döndürüldüğünde, Eşgüdümlü Evrensel Saat (UTC) biçiminde ve ayarlanmış biçimde döndürülür.

Tarih ve saat şu zaman damgası biçiminde döndürülür:

```
YYYY-MM-DDTHH:mm:ss:sssZ
```

Örneğin, 2012-04-23T18:25:43.000Z; burada Z, saat diliminin Eşgüdümlü Evrensel Saat (UTC) olduğunu gösterir.

Bu zaman damgasının doğruluğu garanti edilmez. Örneğin, mqwweb sunucusu URL kaynağında belirtilen kuyruk yöneticisiyle aynı saat diliminde başlatılmazsa, zaman damgası doğru olmayabilir. Ayrıca, Yaz Saati ayarlamaları gerekiyorsa, zaman damgası doğru olmayabilir.

REST API Hata işleme

REST API , 404 (Bulunamadı) gibi uygun bir HTTP yanıt kodu ve JSON yanıtı döndürerek hataları bildirir. 200-299 aralığında olmayan HTTP yanıt kodları hata olarak kabul edilir.

Hata yanıtı biçimi

Yanıt, UTF-8 kodlamasında JSON biçimindedir. İççe yerleştirilmiş JSON nesnelere içeriyor:

- `error` adlı tek bir JSON dizisini içeren bir dış JSON nesnesi.
- Dizideki her öğe, bir hatayla ilgili bilgileri gösteren bir JSON nesnesidir. Her JSON nesnesi aşağıdaki özellikleri içerir:

tip

Dizgi.
Hatanın tipi.

messageId

Dizgi.
MQWBnnnnXformundaki iletiye ilişkin benzersiz tanıttıcı. Bu tanıttıcı aşağıdaki öğeleri içerir:

MQWB

İletinin IBM MQ Rest API 'sinden kaynaklandığını gösteren bir önek.

nnnn

İletiyi tanımlayan benzersiz bir numara.

X

İletinin önem derecesini belirten tek bir harf:

- Bir ileti yalnızca bilgilendiriciyse I .
- Bir ileti bir sorunla ilgili uyarıysa W .
- E iletisi bir hata oluştuğunu gösteriyorsa.
- Bir ileti önemli bir hata oluştuğunu gösteriyorsa S .

ileti

Dizgi.
Hatanın açıklaması.

Açıklama

Dizgi.

Hatanın açıklaması.

işlem

Dizgi.

Hatayı çözmek için gerçekleştirilebilecek adımların açıklaması.

qmgrName

 Bu alan yalnızca, kuyruk yöneticisinin kuyruk paylaşım grubunun üyesi olduğu z/OS için kullanılabilir. **commandScope** isteğe bağlı sorgu parametresini ya da **queueSharingGroupDisposition** özneliğini belirtmiş olmanız gerekir.

Dizgi.

Hatayla karşılaşan kuyruk yöneticisinin adı.

Bu alan messaging REST API için geçerli değildir.

completionCode

Bu alan yalnızca **type** pc f, javaya da rest olduğunda kullanılabilir.

Numara.

Hatayla ilişkili MQ tamamlanma kodu.

reasonCode

Bu alan yalnızca **type** pc f, javaya da rest olduğunda kullanılabilir.

Numara.

Hatayla ilişkili MQ neden kodu.

kural dışı durumlar

Bu alan yalnızca **type** java olduğunda kullanılabilir.

Dizi.

Zincir Java ya da JMS kural dışı durumları dizisi. Kural dışı durum dizisinin her ögesi bir **stackTrace** dizgi dizisi içerir.

stackTrace dizgi dizisi, satırlara bölünmüş her kural dışı durumun ayrıntılarını içerir.

IBM MQ 9.1.2' den bu alan artık döndürülmez.

Kuyruk paylaşım gruplarıyla ilgili hatalar



Bir kuyruk paylaşım grubunda, belirli komutlar için isteğe bağlı bir **commandScope** sorgu parametresi belirtilebilir. Bu parametre, komutun kuyruk paylaşım grubundaki diğer kuyruk yöneticilerine yayılmasını sağlar. Bu komutlardan herhangi biri bağımsız olarak başarısız olabilir, bu da bazı komutların başarılı olmasına ve bazı komutların kuyruk paylaşım grubu için başarısız olmasına neden olur.

Bir komutun kısmen başarısız olduğu durumlarda, 500 'lük bir HTTP hata kodu döndürülür. Bir hata oluşturan her kuyruk yöneticisi için, bu hataya ilişkin bilgiler **error** JSON dizisinde bir öge olarak döndürülür. Komutu başarıyla çalıştıran her kuyruk yöneticisi için, kuyruk yöneticisinin adı **success** JSON dizisinde bir öge olarak döndürülür.

Örnekler

- Aşağıdaki örnek, var olmayan bir kuyruk yöneticisine ilişkin bilgi alma girişimine ilişkin hata yanıtını göstermektedir:

```
"error": [
  {
    "type": "rest",
    "messageId": "MQWB0009E",
    "message": "MQWB0009E: Could not query the queue manager 'QM1'",
    "explanation": "The MQ REST API was invoked specifying a queue manager name which cannot be located.",
    "action": "Resubmit the request with a valid queue manager name or no queue manager name, to retrieve a list of queue managers. "
```

```
}  
]
```

- ▶ **Z/OS** Aşağıdaki örnek, bazı kuyruk yöneticileri için var olmayan bir kuyruk paylaşım grubundaki bir kuyruğu silme girişimine ilişkin hata yanıtını göstermektedir:

```
"error" : [  
  {  
    "type": "rest",  
    "messageId": "MQWB0037E",  
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason  
code is 3312 : 'MQRCCF_UNKNOWN_OBJECT_NAME'",  
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be  
located.",  
    "action": "Resubmit the request with the name of an existing queue, or with no queue name  
to retrieve a list of queues.",  
    "qmgrName": "QM1"  
  },  
  {  
    "type": "rest",  
    "messageId": "MQWB0037E",  
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason  
code is 3312 : 'MQRCCF_UNKNOWN_OBJECT_NAME'",  
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be  
located.",  
    "action": "Resubmit the request with the name of an existing queue, or with no queue name  
to retrieve a list of queues.",  
    "qmgrName": "QM2"  
  }  
],  
"success" : [{ "qmgrName": "QM3"}, { "qmgrName": "QM4"}]
```

MFT istekleriyle ilgili hatalar

MFT REST API hizmetleri etkinleştirilmediyse ve MFT REST API programını çağırırsanız, aşağıdaki kural dışı durumu alırsınız:

```
{  
  "error": [{  
    "action": "Enable the Managed File Transfer REST API and resubmit the request.",  
    "completionCode": 0,  
    "explanation": "Managed File Transfer REST calls are not permitted as the service is  
disabled.",  
    "message": "MQWB0400E: Managed File Transfer REST API is not enabled.",  
    "msgId": "MQWB0400E",  
    "reasonCode": 0,  
    "type": "rest"  
  }]  
}]
```

MFT REST API hizmetleri etkinleştirildiyse ve mqwebuser.xml dosyasında koordinasyon kuyruk yöneticisi ayarlanmamışsa, aşağıdaki kural dışı durumu alırsınız:

```
{  
  "error": [{  
    "action": "Set the coordination queue manager name and restart the mqweb server.",  
    "completionCode": 0,  
    "explanation": "Coordination queue manager name must be set before using Managed File  
Transfer REST services.",  
    "message": "MQWB0402E: Coordination queue manager name is not set.",  
    "msgId": "MQWB0402E",  
    "reasonCode": 0,  
    "type": "rest"  
  }]  
}]
```

REST API Keşif

REST API belgeleri, IBM Documentation içinde ve Swagger biçiminde bulunur. Swagger, REST API 'lerini belgelemek için yaygın olarak kullanılan bir yaklaşımdır. REST API için Swagger belgeleri, mqweb sunucusunda API Discovery özelliği (apiDiscovery) etkinleştirilerek görüntülenebilir.

Başlamadan önce

Stabilized

Önemli: apiDiscovery aksamı sabitlendi. Bu özelliği kullanmaya devam edebilirsiniz. Şu anda IBM MQ , mpOpenAPI özelliğinin kullanımını desteklemez.

API Discovery kullanarak Swagger belgelerini görüntülemek için mqweb sunucusunun güvenliğini etkinleştirmeniz gerekir. Güvenliği etkinleştirmek için gerekli adımlar hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API güvenlik.](#)

Yordam

1. Aşağıdaki dizinlerden birinde mqwebuser.xml dosyasını bulun:

- ALW MQ_DATA_PATH/web/installations/installationName/servers/mqweb
- z/OS WLP_user_directory/servers/mqweb

Burada WLP_user_directory , crtmqweb komut dosyası mqweb sunucusu tanımlamasını yaratmak için çalıştırıldığında belirtilen dizindir.

2. mqwebuser.xml dosyasına uygun XML ' i ekleyin:

- mqwebuser.xml dosyanızdaki <featureManager> etiketleri varsa, <featureManager> etiketleri içine aşağıdaki XML ' i ekleyin:
<feature>apiDiscovery-1.0</feature>
- <featureManager> etiketleri mqwebuser.xml dosyanda yoksa, <server> etiketleri içine aşağıdaki XML ' i ekleyin:

```
<featureManager>  
  <feature>apiDiscovery-1.0</feature>  
</featureManager>
```

3. Aşağıdaki yöntemlerden birini kullanarak Swagger belgelerini görüntüleyin:

- Bir tarayıcıya aşağıdaki URL adresini girerek REST API ' ye göz atabileceğiniz bir web sayfası görüntüleyin ve deneyin:

`https://host:port/ibm/api/explorer`

Her bir isteğin kimliğini doğrulamanın yanı sıra, her POST, PATCH ya da DELETE isteği için bir ibm-mq-rest-csrf-token üstbilgisi eklemeniz gerekir. Bu üstbilginin içeriği, boşluk da içinde olmak üzere herhangi bir dizgi olabilir.

Bu istek üstbilgisi, isteği doğrulamak için kullanılan kimlik bilgilerinin, kimlik bilgilerinin sahibi tarafından kullanıldığını doğrulamak için kullanılır. Yani, simge siteler arası istek sahteciliği saldırılarını önlemek için kullanılır.

- REST API 'in tamamını açıklayan tek bir Swagger 2 belgesini almak için aşağıdaki URL' ye bir HTTP GET komutu verin:

`https://host:port/ibm/api/docs`

Bu belge, programlı olarak kullanılabilir API ' lerde gezinmek istediğiniz uygulamalar için kullanılabilir.

anasistem

REST API ' in kullanılabilir olduğu anasistem adını ya da IP adresini belirler.

Varsayılan değer localhost'tur.

kapı

administrative REST API ' in kullandığı HTTPS kapı numarasını belirtir.

Varsayılan değer 9443'tur.

Anasistem adı ya da kapı numarası varsayılan değerden değiştirilirse, REST API URL' den doğru değerleri belirleyebilirsiniz. URL' yi görüntülemek için **dspmweb status** komutunu kullanın.

İlgili bilgiler

[mqweb durumunu görüntüle \(mqweb sunucusu durumunu görüntüle\)](#)

REST API ulusal dil desteği

REST API , belirli niteliklere sahip olarak, HTTP isteğinin bir parçası olarak ulusal dilleri belirtme yeteneğini destekler.

Arka Plan

[HTTP üstbilgileri](#) , isteklerde belirli bir davranışın belirtilmesine ve yanıtlarda ek bilgilerin sağlanmasına izin verir.

HTTP üstbilgileri, bilgilerin ulusal dilde döndürülmesini isteme yeteneğidir. REST API mümkün olduğunda bu üstbilgiyi onurlandırır.

Ulusal dil belirtilmesi

ACCEPT-LANGUAGE HTTP üstbilgisinde bir ya da daha çok dil etiketi sağlanabilir. İsteğe bağlı olarak, bir sıralamayı etiketlerle ilişkilendirebilir ve seçim temelinde sıralanmış bir liste belirtimine izin verebilirsiniz. [Bu sayfa](#) , ilkeye ilişkin yararlı bir tartışma sağlar.

REST API , ACCEPT-LANGUAGE üstbilgisinden bir dil seçerek ve o dilde ileti döndürerek bu üstbilgiyi onurlandırır. ACCEPT-LANGUAGE üstbilgisinde REST API ' in destekleyebileceği bir dil yoksa, iletiler varsayılan dilde döndürülür. Bu varsayılan dil, REST API web sunucusunun varsayılan yerel ayarına karşılık gelir.

[“Hangi veriler çevrilir?” sayfa 86](#) bölümünde hangi verilerin çevrildiği açıklanır.

Yanıtlarda ilgili dili belirme

REST API ' den alınan yanıtlardaki CONTENT-LANGUAGE HTTP üstbilgisi, iletilerin döndürüleceği dili gösterir.

Hangi veriler çevrilir?

Hata ve bilgi iletileri çevrilir, diğer metinler çevrilmez.

- Bir kuyruk yöneticisinden döndürülen veriler çevrilmez; örneğin, REST API aracılığıyla bir MQSC komutunun yürütülmesi durumunda, kuyruk yöneticisinin yanıtları kuyruk yöneticisinin ülke değerinde olur.
- `apiDiscovery` özelliğiyle gösterilen REST API için oluşturulan (Swagger) belgeleri İngilizce 'dir.

Hangi diller desteklenir?

İngilizce 'ye ek olarak, REST API hata ve bilgi iletileri aşağıdaki dillere çevrilir.

Çince (Basitleştirilmiş)

zh_CN dil etiketiyle gösterilir

Çince (Geleneksel)

zh_TW dil etiketiyle gösterilir

Cekçe

cs dil etiketiyle gösterilir

Fransızca

fr dil etiketiyle gösterilir

Macarca

hu dil etiketiyle gösterilir

İtalyanca

it dil etiketiyle gösterilir

Japonca

ja dil etiketiyle gösterilir

Korece

ko dil etiketiyle gösterilir

Lehçe

pl dil etiketiyle gösterilir

(Brezilya) Portekizce

pt_BR dil etiketiyle gösterilir

Rusça

ru dil etiketiyle gösterilir

İspanyolca

es dil etiketiyle gösterilir

Örnekler

Örneklerde, web sunucusunun varsayılan İngilizce ülke değeri vardır.

Desteklenen tek bir dili belirtme

İstek üstbilgilerinde ACCEPT - LANGUAGE , frolarak ayarlanır. Bu ayar, Fransızcanın çevrilebilir metin için tercih edilen dil olduğunu belirtir.

Yanıt üstbilgilerinde CONTENT - LANGUAGE , frolarak ayarlanır. Bu ayar, yanıtta hata ve bilgi iletilerinin Fransızca olduğunu gösterir.

Dil listesi belirtilmesi

İstek üstbilgilerinde ACCEPT - LANGUAGE , am, frolarak ayarlanır. Bu ayar, Amharic ve Fransızca 'nın çevrilebilir metin için kabul edilebilir diller olduğunu ve Amharic 'in çevrilebilir metin için tercih edilen dil olduğunu belirtir.

Yanıt üstbilgilerinde CONTENT - LANGUAGE , frolarak ayarlanır. Bu ayar, REST API Amharic 'i desteklemediğinden, yanıtta hata ve bilgi iletilerinin Fransızca olduğunu gösterir.

Tek bir desteklenmeyen dil belirtme

İstek üstbilgilerinde ACCEPT - LANGUAGE , amolarak ayarlanır. Bu ayar, Amharic 'in çevrilebilir metin için tercih edilen dil olduğunu belirtir.

Yanıt üstbilgilerinde CONTENT - LANGUAGE , enolarak ayarlanır. Bu ayar, REST API 'in Amharic 'i desteklememesi nedeniyle yanıtta hata ve bilgi iletilerinin İngilizce olduğunu gösterir.

REST API sürümler

REST API sürüm numarası, REST istekleri için temel URL 'nin bir bölümünü oluşturur. Örneğin, https://localhost:9443/ibmmq/rest/v2/admin/installation. Sürüm numarası, istemcilerin ilerideki yayınlarda sunulabilecek REST API üzerindeki değişikliklerden yalıtılması için kullanılır.

IBM MQ 9.2.0 , REST API sürüm 2 'yi tanıtır. Bu sürüm artışı administrative REST API, messaging REST API ve MFT REST API için geçerlidir. Bu sürüm artışı, REST API için kullanılan URL kaynağını değiştirir. Sürüm 2 'deki kaynak URL adresleri için URL öneki aşağıdaki URL 'dir:

```
https://host:port/ibmmq/rest/v2/
```

Stabilized REST API işlevine tanıtılan bazı değişiklikler, REST API işlevini kullanan istemcilerin güncellenmesi gerekebilecek şekilde var olan REST API işlevini değiştirebilir. Bu tür değişikliklerin istemcileri güncellemeye zorlamasını önlemek için, REST API sürüm numarası artırılır ve var olan işlev önceki sayıyla dengelenir. Var olan işlevi değiştirebilecek yeni işlev, yeni sürüm numarasında REST API

' e eklenir. Bu nedenle, istemciler güncellenmeden önceki sürümde REST API ' i kullanmaya devam edebilirler.

İstemci güncellemesi gerekmesine neden olabilecek REST API değişiklikleri aşağıdaki değişiklikleri içerir:

- REST API'e gönderilen ya da bu öğeden döndürülen JSON' da var olan bir öznitelik için desteğin kaldırılması.
- Bir URL, HTTP filinin ya da üstbilginin kaldırılması. Örneğin, bir URL ya da üstbilgi yeniden adlandırılırsa ya da farklı bir komut kullanılırsa.
- Var olan bir URLadresine gönderilen verilere yeni bir zorunlu JSON özniteliği eklenmesi.
- Var olan bir URLadresine gönderilen verilere yeni bir zorunlu HTTP üstbilgisi eklenmesi.
- Var olan bir URL' ye yeni bir zorunlu sorgu parametresi eklenmesi.

Bu tip bir değişiklik Long Term Support (LTS) yayınında var olan REST API işlevine sunulduğunda, bu değişikliklerin ilki için REST API sürüm numarası artırılır. Bir Continuous Delivery (CD) yayınında yapılan ve REST API kullanan istemcilerde değişiklik gerektirebilecek sonraki değişiklikler yeni sürüm numarasını kullanır.

Bu sürüm numarası, sonraki CD yayın düzeylerinde sonraki LTS yayın düzeyine kadar aynı kalır. Bu nedenle, sürüm numarası LTS yayınları arasında en çok bir kez artar.

Stabilized Sürüm numarası artırıldığında, var olan REST API işlevi eski sürüm numarasında dengelenir. Yani, LTS yayın düzeyinde kullanılabilir olan REST API işlevi eski sürüm numarasında kalır, ancak bu sürümde başka değişiklik yapılmaz. REST API sürümüne eklenen yeni işlevler yeni REST API sürümüne eklenir. Ancak, sürüm artmadan önce CD yayınlarında REST API 'a yapılan eklemelerin REST API' in eski sürümüne dahil edileceği garanti edilmez.

Deprecated Var olan istemciler, değişiklik gerektirmeden eski sürüm numarasında REST API ' i kullanmaya devam edebilirler. REST API ' in eski sürümleri kullanımdan kaldırılabilir ve sonunda kaldırılabilir.

Bazı değişiklikler, REST APIkullanan istemcilerde değişiklik yapılmasını gerektirmez. Bu değişiklikler sürüm numarasının artmasıyla sonuçlanmaz. Bu nedenle, REST API kullanan herhangi bir istemcinin, bu tip değişiklikler kullanıma sunulduğunda güncellenmesine gerek olmadığından emin olun. REST API üzerinde yapılan bu değişiklikler aşağıdaki değişiklikleri içerebilir:

- REST APItarafından döndürülen var olan verilere yeni bir JSON özniteliğinin eklenmesi.
- Yeni bir URLeklenmesi.
- Var olan bir URL' ye yeni bir HTTP komutu eklenmiştir.
- Var olan bir URL' ye yeni bir durum kodu eklenmesi.
- Var olan bir URLadresine gönderilen verilere yeni isteğe bağlı JSON öznitelikleri eklenmiştir.
- Var olan bir URL' ye yeni sorgu değiştirgelerinin eklenmesi.
- Var olan bir URL' ye gönderilen verilere yeni üstbilgilerin eklenmesi.
- REST API' den yeni üstbilgilerin dönüşü.

Yeni Continuous Delivery REST API işlevindeki değişiklikler

CD yayınına eklenen yeni REST API işlevinde, bu yeni işlevde yapılan ve REST API istemcilerinde değişiklik yapılmasını gerektirebilecek değişiklikler sürüm numarasını artırmaz. Yani, yeni işlev sonraki LTS yayın düzeyinden önce sürüm numarasını artırmadan değişebilir. İşlev bir LTS yayınına eklendiğinde, REST API istemcilerinde değişiklik yapılmasını gerektirebilecek sonraki değişiklikler sürüm numarasını artırır.

Örnek

1. LTS yayın düzeyi X 'te REST API , sürüm 1 'de yer almaktadır.
2. CD yayın düzeyi X.0.1' de yeni bir URL desteği eklenir. Bu değişiklik, REST APIkullanan istemcilerde değişiklik yapılmasını gerektirmez. Bu nedenle REST API , sürüm 1 'de kalır.

3. CD X.0.2' de yeni bir URL desteği eklenir. Bu değişiklik, REST API 'sini kullanan istemcilerde değişiklik yapılmasını gerektirmez. Bu nedenle REST API , sürüm 1 'de kalır.
4. LTS yayın düzeyi Y 'de REST API , sürüm 1 'de yer almaktadır.
5. CD yayın düzeyi Y.0.1' de var olan bir URL yeniden adlandırılır. Bu değişiklik, REST APIkullanan istemcilerde değişiklik yapılmasını gerektirebilir. Bu nedenle, REST API ürününün yeni bir sürümü sürüm 2 olarak oluşturulur. Yeniden adlandırılan URL , var olan tüm işlevle birlikte REST APIsürüm 2 'de bulunur. REST API ' e eklenen yeni işlevler sürüm 2 'ye eklenir. Sürüm 1, LTS yayın düzeyi Y ' de sabit kalır.
6. CD yayın düzeyi Y.0.2' de var olan başka bir URL yeniden adlandırılır. Sürüm CD yayın düzeyi Y ' de zaten artırıldıktan sonra, REST API sürüm 2 'de kalır. Sürüm 1, LTS yayın düzeyi Y ' de sabit kalır.
7. LTS yayın düzeyi Z ' de REST API sürüm 2 'de kalır. Sürüm 1, LTS yayın düzeyi Y ' de sabit kalır.

IBM MQ Console kullanarak yönetim

IBM MQ Consolekomutunu kullanarak temel yönetim görevlerini gerçekleştirebilirsiniz.

9.3 CD yayın düzeylerine ilişkin konsol, 9.3 LTS yayın düzeylerine ilişkin konsoldan farklıdır. Geçerli sürümünüz için uygun konulara bakın.

Not: IBM MQ Consolekomutunu kullanırken kuyruk yöneticilerinizin hiçbirinde komut sunucusunu devre dışı bırakmayın. Bir kuyruk yöneticisi için komut sunucusu geçersiz kılındıysa:

- IBM MQ Console , komutların işlenmesinde uzun gecikmelerle yanıt vermez.
- Kuyruk yöneticisine verilen tüm komutlar zamandışı.

İlgili görevler



[IBM MQ Console izleme](#)


IBM MQ Console ile çalışmaya başlama

mqwweb sunucusunu yapılandırın; IBM MQ Consoleiçin URI ' yi belirleyin; konsola bağlanın; konsolda oturum açın.

Başlamadan önce


Bu görevi tamamlamak için, [dspmqweb](#) komutunu kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

-  z/OS' da [dspmqweb](#) komutunu çalıştırma ve mqwebuser.xml dosyasına yazma erişiminiz olmalıdır.
-  Diğer tüm işletim sistemlerinde [ayrıcalıklı kullanıcı](#) olmanız gerekir.

 IBM i üzerinde, komutlar QSHELL içinde çalıştırılmalıdır.

Bu görev hakkında

Aşağıdaki kısıtlamaları dikkate alın:

-  z/OS üzerinde kuyruk yöneticileri yaratılamaz, silinemez, başlatılamaz ya da durdurulamaz.
 - z/OS üzerindeki kanal başlatıcılar başlatılamıyor ya da durdurulamıyor ve kanal başlatıcı durumu görüntülenmiyor.
 - Dinleyiciler görüntülenemiyor ya da denetlenemiyor.
 - Kanal başlatma, ping, çözümleme ve ilk duruma getirme komutları yalnızca CHLDISP (DEFAULT) ile verilebilir.

- QSGDISP (GROUP) ile tanımlanan nesnelere görüntülenemez ya da yönetilemez.
- Kuyruk yöneticisi güvenliği yönetilemiyor.
- Sistem kaynağı kullanımını izlenemiyor.

• Multi

- AMQP kanallarıyla çalışmak için IBM MQ Console kullanamazsınız.
- MQTT kanallarıyla çalışmak için IBM MQ Console kullanamazsınız.

Yordam

1. mqweb sunucusu IBM MQ Console tarafından kullanılmak üzere yapılandırılmadıysa, mqweb sunucusunu yapılandırın.

mqweb sunucusunu temel bir kayıt dosyasıyla yapılandırma hakkında daha fazla bilgi için bkz. [mqweb sunucusu için temel yapılandırma](#).

2. z/OS

z/OS' ta, **dspmqweb** komutunu kullanabilmek için WLP_USER_DIR ortam değişkenini ayarlayın. Şu komutu girerek değişkeni mqweb sunucusu yapılandırmanızı gösterecek şekilde ayarlayın:

```
export WLP_USER_DIR=WLP_user_directory
```

burada *WLP_user_directory* , *crtmqweb*' e geçirilen dizinin adıdır. Örneğin:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Daha fazla bilgi için [mqweb sunucusu yaratılması](#) başlıklı konuya bakın.

3. Aşağıdaki komutu girerek IBM MQ Console için URI ' yi belirleyin:

```
dspmqweb status
```

Komut, aşağıdakine benzer bir çıkış oluşturur:

```
MQWB1124I: Server 'mqweb' is running.
URLS:
https://localhost:9443/ibmmq/rest/v1/
https://localhost:9443/ibmmq/console/
```

IBM MQ Console URI 'si, *console/*sonesiyle biter.

4. Bir tarayıcıda önceki adımdan URL adresini girerek IBM MQ Console adresine bağlanın.

mqweb sunucusuyla sağlanan varsayılan sertifika güvenilir bir sertifika olmadığından tarayıcı tarafından bir güvenlik kural dışı durumu üretilebilir. IBM MQ Console ile devam etmeyi seçin.

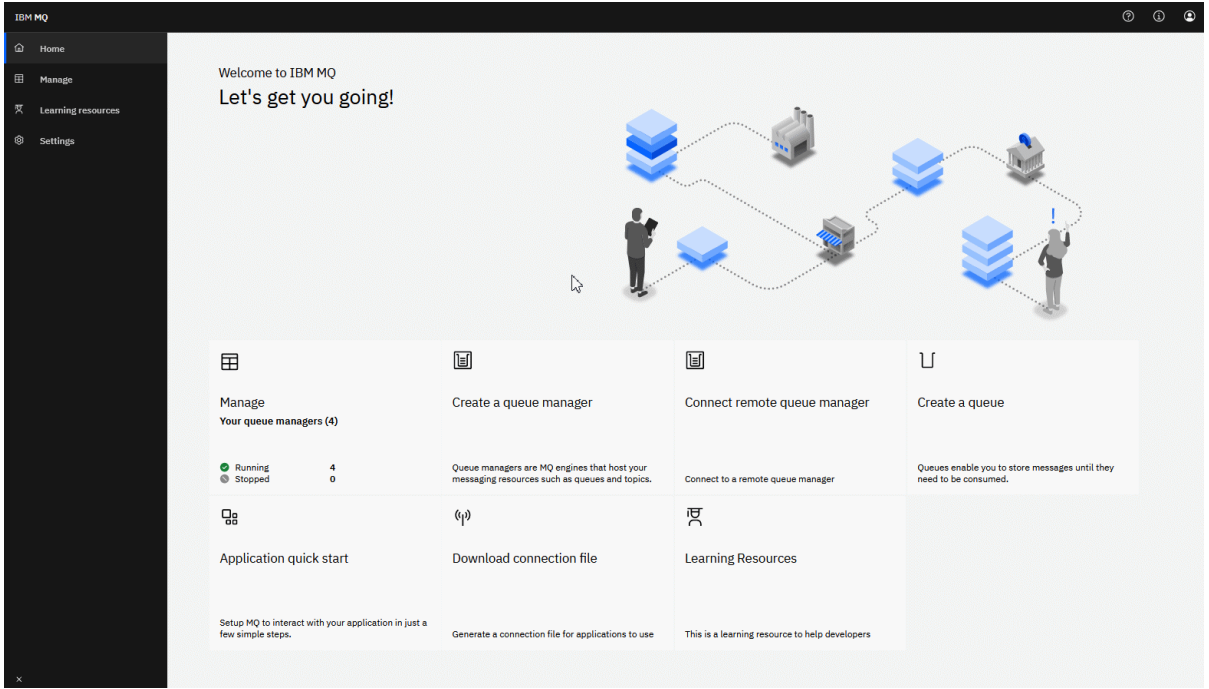
5. IBM MQ Console' da oturum açın. *mqadmin* kullanıcı adını ve *mqadmin* parolasını kullanın.

Sonraki adım

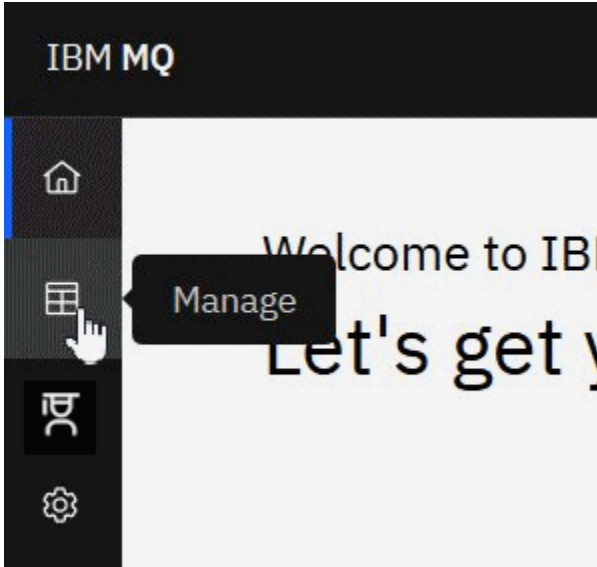
Varsayılan olarak IBM MQ Console , kullanıcıların kimliğini doğrulamak için belirteç tabanlı kimlik doğrulamasını kullanır. İstemci sertifikası kimlik doğrulamasını da kullanabilirsiniz. Daha fazla bilgi için bkz. [REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma](#).

V9.3.1 Yalnızca una unu gelen unun unun gelen una unun una gelen gelen IBM MQ Console -CD

IBM MQ Console ' da ilk oturum açtığınızda, iniş sayfasına yönlendirilirsiniz. Buradan, var olan kuyruk yöneticilerini yönetmeyi seçebilir, bir kuyruk yöneticisi ya da kuyruk yaratabilir, bazı eğitim konularına gidebilir ya da IBM Documentation içinde IBM MQ ürün bilgilerini açabilirsiniz. Yeni ya da var olan kuyruk yöneticileri ve uygulamalar arasında ileti sisteminin ver için hızlı ve kolay bir şekilde ayarlanması sürecinde size yol gösteren uygulama hızlı başlangıcını da başlatabilirsiniz.



Diğer bir seçenek olarak, IBM MQ nesnelerini hemen yönetmeye başlamak için yönet simgesini tıklatabilirsiniz.



Yönet görünümü başlangıçta kuyruk yöneticilerini ve bunların yürürlükteki durumunu gösterir. Ayrıca, yeni kuyruk yöneticileri yaratabilir ve uzak kuyruk yöneticilerini bağlayabilirsiniz.

IBM MQ

Manage

Queue managers

Remote queue managers

Local queue managers

Here you can view and manage all of the queue managers co-located with this instance of the MQ console.

4 of 4 queue managers loaded

Unique name ↑	Status
QM931	Running
QMGR1	Running
QMGR2	Running
QMGR3	Running

Items per page: 10 1-4 of 4 items 1 of 1 pages

Her kuyruk yöneticisinin, çalışmakta olan bir kuyruk yöneticisini durdurmanızı ya da yapılandırmanızı ya da durdurulmuş bir kuyruk yöneticisini başlatmanızı ya da silmenizi sağlayan bir menüsü vardır.

● Running

● Stopped

● Stopped

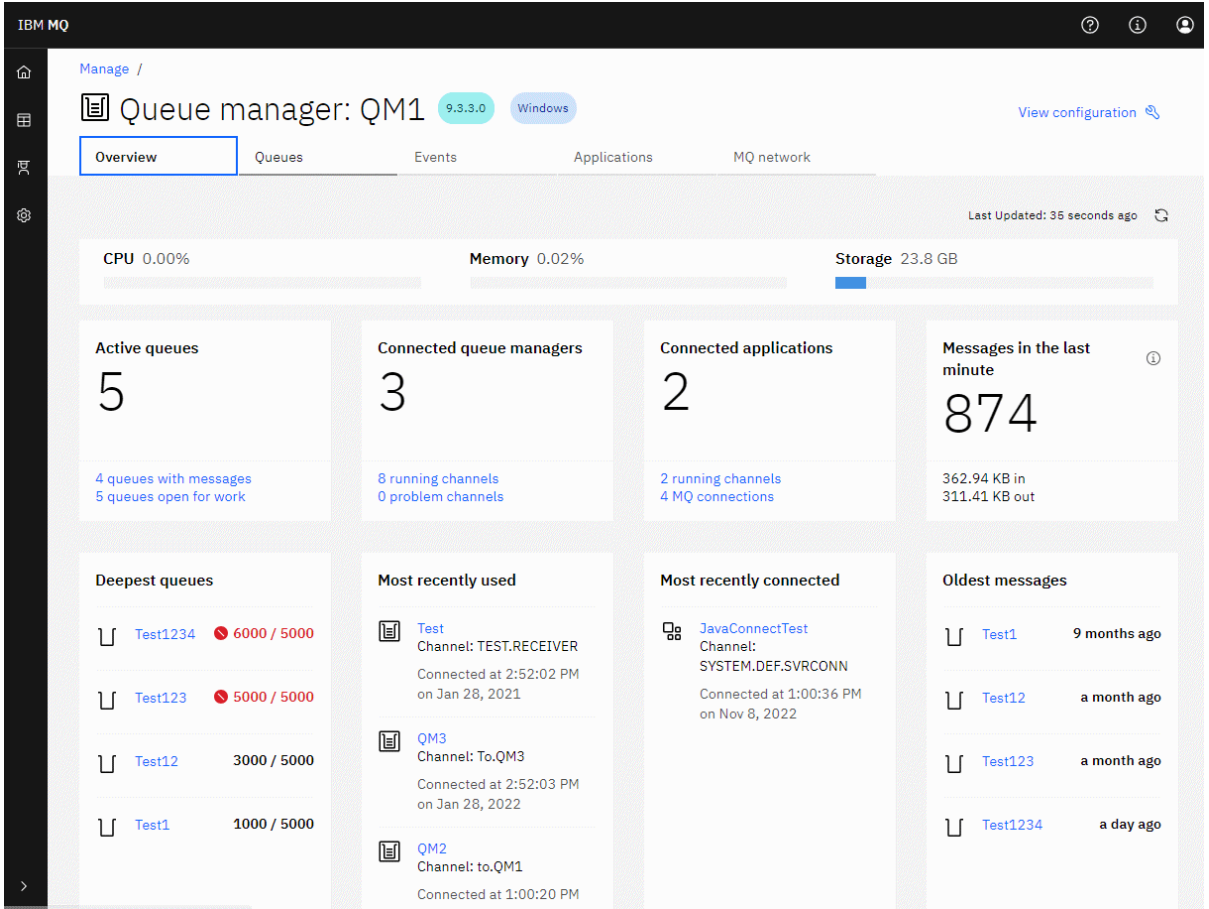
● Stopped

Stop

Configuration

Kuyruk yöneticisine ilişkin Yetki Kayıtları, Kimlik Doğrulama Bilgileri nesnelere ve Kanal Kimlik Doğrulama kayıtları, kuyruk yöneticisinin **Yapılandırma** sayfasının **Güvenlik** sekmesinde bulunabilir. Bu sayfada yeni kayıtlar oluşturabilir ve bunları ekleyebilirsiniz.

Gösterge panosunu açmak için çalışmakta olan bir kuyruk yöneticisinin adını tıklatın.



Kuyruk yöneticisi gösterge panosundan aşağıdaki işlemleri gerçekleştirebilirsiniz:

V 9.3.4 Genel Bakış sekmesinde aşağıdaki bilgileri görüntüleyin:

CPU

Kuyruk yöneticisi tarafından CPU kullanımının yüzde tahmini. (z/OS için geçerli değildir.)

Bellek

Kuyruk yöneticisi tarafından bellek kullanımı tahmini yüzdesi. (z/OS ya da Windows için geçerli değildir.)

Depolama

Kuyruk yöneticisinin bulunduğu diskin boş alanının yüzde tahmini değeri. (z/OS için geçerli değildir.)

Etkin kuyruklar

İleti içeren ya da giriş ya da çıkış için açık olan kuyrukların sayısı.

Bağlı kuyruk yöneticileri

Şu anda bağlı olan kuyruk yöneticilerinin etkin kanallardan türetildiği şekilde sayısı.

Bağlı uygulamalar

Şu anda bağlı olan uygulamaların sayısı.

Son dakikada gönderilen iletiler

Her 10 saniyede bir ileti verimini gösteren PUT/GET sistem konularının bir özetini görüntüler. (z/OS için geçerli değildir.)

Abonelikler

Abonelik sayısını görüntüler. Yalnızca z/OS üzerinde ve sistem konularının izlenmesinin engellendiği diğer platformlarda görünür (bkz. [setmqweb](#) özellikleri).

En derin kuyruklar

Kuyrukları derinlik sırasına göre listeler. Yürürlükteki kuyruk derinliğini ve kuyruk derinliği üst sınırını gösterir.

En son kullanılan

Şu anda bağlı olan kuyruk yöneticilerini, son ileti tarihine göre sıralanmış olarak listeler.

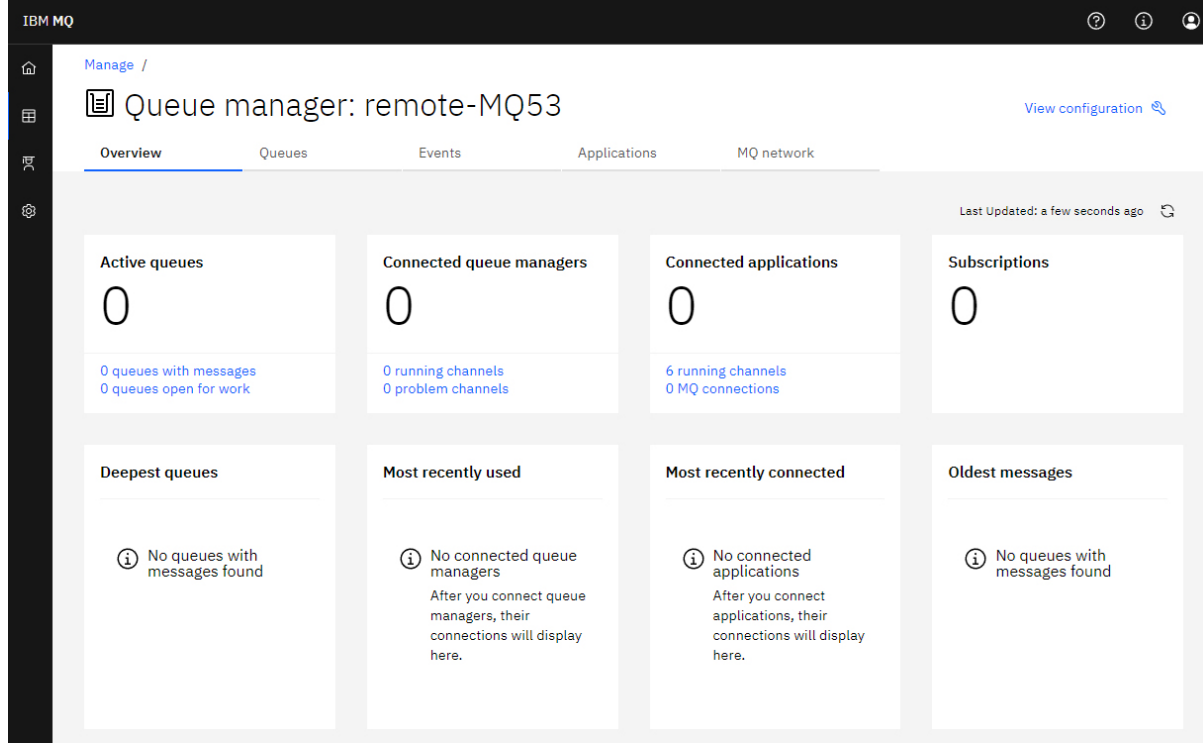
En son bağlanan

Şu anda bağlı olan uygulamaları, kanal başlangıç tarihi ve saati temelinde sıralanmış olarak etkin sunucu bağlantısı kanallarından türetilmiş olarak listeler.

En eski iletiler

En eski ileti tarihi ve saati temelinde sıralanmış kuyrukları listeler.

Genel Bakış sekmesinde görüntülenen bilgiler, sistem konularının izlenmesinden türetilir (Sistem konularında yayınlanan ölçümler konusuna bakın). z/OS , sistem konusu izlemesini desteklemez ve konsol görüntüleme amaçlarının izlenmesi diğer platformlarda devre dışı bırakılabilir (bkz. setmqweb özellikleri). Bu durumlarda, **Genel Bakış** sekmesi daha sınırlı bilgi görüntüler ve görünümü aşağıdaki örneğe benzer:




Kuyruklar sekmesinde:


- Yeni kuyruklar yarat
- Varolan iletileri görüntülemek, yeni iletiler yaratmak ve kuyruğu yapılandırmak için kuyruk adını tıklatın.

Etkinlikler sekmesinde:

Konular

- Yeni konular yarat
- Var olan konuları yapılandır  gösterir
- Eşleşen abonelikleri görüntülemek için konu adını tıklatın

Abonelikler

- Yeni yönetilen ya da yönetilmeyen abonelikler yarat, da,,
- Var olan abonelikleri yapılandır  gösterir

V 9.3.5 **Uygulamalar** sekmesinde:

Genel Bakış

Aşağıdaki istatistiklere genel bakış sağlayan döşemeleri içerir:

Bağlı uygulamalar

Bağlı uygulamaların sayısını görüntüler. Aşağıdaki sekmelere bağlantılar sağlar:

- **Uygulama örnekleri**
- **Bağlantılar**

Kanal örneklerinin çalıştırılması

Uygulama Kanalı etiketinde tanımlanan ya da durdurulan SVRCONN kanal eşgörünümlerinin sayısını ve bu bağlantılardan bunların sayısını görüntüler.

Bağlantılar

Bağlantı sayısını görüntüler. **Bağlantılar** sekmesinde aşağıdaki bilgilere bağlantılar sağlar:

- Yerel bağlantılar (kanal adı olmayanlar)
- Uzak bağlantılar (kanal adı olanlar)

En sık kullanılan uygulamalar

Sık kullanılan uygulamaların listesini, kullandıkları bağlantı sayısına göre sıralanmış olarak görüntüler.

En yaygın kanallar

Etkin olan eşgörünümlerin sayısına göre sıralanmış sık kullanılan kanalların bir listesini görüntüler.

En eski işlemler

Uygulama adına göre en eski hareketlerin listesini görüntüler. Bu hareketlerin açık iş birimlerine sahip bağlantıları vardır ve UOW başlangıç tarihi ve saati temelinde sıralanır.

Uzak bağlantılı sürümler

Ortak bağlantılı IBM MQ sürümlerinin (yani, belirtilen REMOTE_VERSION sürümüne sahip kanal eşgörünümlerinin) bir listesini görüntüler.

Uygulama kanalı güvenliği

Ortak bağlı kanal güvenliği protokollerinin (SECURITY_PROTOCOL) bir listesini görüntüler.

Kanal aktarım hızları

İletilerin ve baytların aktarım hızlarına göre sıralanmış ortak kanalların bir listesini görüntüler. Süreyi hesaplamak için kanal başlangıç tarihini ve saatini kullanır ve hızı hesaplamak için MSGS ve MQIACH_BYTES_SENT/ MQIACH_BYTES_RCVD komutunu kullanır.


Uygulamalar

Kuyruk yöneticisine bağlı uygulamalarla ilgili bilgileri görüntüleyin.


Kanallar

Uygulamalara bağlı kanallardaki etkinliği görüntüleyin.

Uygulama kanalları

- Kanalları başlat, durdur, ping ve yapılandır 
- Yeni kanallar oluştur
- Kanalları sıfırla

Uygulama kanalı örnekleri

- Uygulama kanalı örneklerinin durumunu görüntüle
- Kanallardaki belirsiz iletileri çözümler  gösterir

V 9.3.5

MQ Network sekmesinde:

Genel Bakış

Aşağıdaki istatistiklere genel bakış sağlayan döşemeleri içerir:

Kuyruk yöneticisi kanal örneklerinin çalıştırılması

SVRCONN olmayan kanal eşgörünümlerinin sayısını görüntüler. **Bağlı kuyruk yöneticileri** etiketinde aşağıdaki kanal örneği tiplerine ilişkin bağlantıları görüntüler:

- Tanımlı kanallar
- Durdurulan kanallar

Bağlı kuyruk yöneticileri

MQCA_REMOTE_Q_MGR_NAME ile bağlanan kuyruk yöneticilerinin sayısını görüntüler. MQCMD_INQUIRE_CLUSTER_Q_MGR tarafından döndürülen kuyruk yöneticilerinin sayısını da sağlar.

Küme üyeliği

Yalnızca bir kuyruk yöneticisi kümesi varsa, kümenin adını ve kuyruk yöneticisinin tam ya da kısmi bir havuz olup olmadığını görüntüler. Kümede kaç kuyruk yöneticisinin görüldüğünü görüntüler. Birden çok küme varsa, küme sayısının yanı sıra her bir kümedeki tam ve kısmi havuz kuyruğu yöneticilerinin sayısını görüntüler.

Hatalı kuyruk yöneticisi kanalları

Yeniden deneniyor (durdurulmadı/çalıştırılmıyor) durumundaki kanalların bir listesini görüntüler. Yeniden deneme durumunda kalan yeniden deneme sayısını hesaplar. Listede aşağıdaki durum tiplerine sahip kanallar bulunur:

- MQCHS_AUSED
- MQCHS_REDENIYOR

En uzun ileti gecikmeleri

XMIT zaman göstergesi (uzun dönem) olan kanalların listesini görüntüler.

Gözetimsiz iletim kuyrukları

Kuyruk derinliği sıfır olmayan ve ilişkili tutamaçları olmayan iletim kuyruklarının listesini görüntüler.

Uzak bağlantı sürümleri

Ortak bağlantılı IBM MQ sürümlerinin (yani, belirtilen REMOTE_VERSION sürümüne sahip kanal eşgörünümlerinin) bir listesini görüntüler.

Kuyruk yöneticisi kanal güvenliği

Ortak bağlı kanal güvenliği protokollerinin (SECURITY_PROTOCOL) bir listesini görüntüler.

Küme durumu

Kümenin sağlıklı işletimiyle ilgili bir dizi bağımsız istatistik görüntüler. Durum şunları içerir:

- Küme nesnelerinin sayısı (kuyruklar, konular, kuyruk yöneticileri).
- Askıya alınan kuyruk yöneticilerinin sayısı (MQIACF_SUSPEND YES olarak ayarlandı).
- SYSTEM.CLUSTER.COMMAND.QUEUE (KUYRUK).
- SYSTEM.TEMP.

Bu değerlerin tümü sıfırsa, bu döşeme görüntülenmez ve bunun yerine **Dinleyiciler** döşemesi gösterilir.


Dinleyiciler

Dinleyicilerin listesini ve bunların çalışır durumda olup olmadıklarını görüntüler. Yalnızca **Küme durumu** döşemesi görüntülenmiyorsa gösterilir.

Kuyruk yöneticileri bağlı

Bu kuyruk yöneticisine bağlı olan kuyruk yöneticilerinin ayrıntılarını görüntüleyin.

Kuyruk yöneticisi kanalları

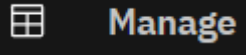
- Kanalları başlat, durdur, ping ve yapılandır 
- Yeni kanallar oluştur
- Kanalları sıfırla

Kuyruk yöneticisi kanal eşgörünümleri


- Kuyruk yöneticisi kanal eşgörünümlerinin durumunu görüntüleme bir) için
- Kanallardaki belirsiz iletileri çözümler gösterir


V9.3.1 IBM MQ Console: Yerel kuyruk yöneticileriyle çalışılması-Yalnızca CD

Yerel kuyruk yöneticilerini Yönet görünümünün en üst düzeyinden oluşturur, yapılandırır ve

denetleyebilirsiniz  gösterilir.

Bu görev hakkında

 Yönet görünümü, şu IBM MQ Console ürününün çalıştığı IBM MQ kuruluşu. Aynı sistemdeki farklı IBM MQ kuruluşlarıyla ilişkili kuyruk yöneticileri listelenmez.

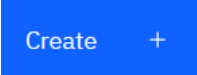
 z/OS işletim sisteminde Manage görünümü, IBM MQ Console ile aynı sürümde olan kuyruk yöneticilerini listeler ve IBM MQ Console ' in çalıştığı sistemde tanımlanır. IBM MQ Console sürümünden farklı bir sürümdeki kuyruk yöneticileri listelenmez.

Çalışmak istediğiniz kuyruk yöneticilerini listeden tek tek seçebilirsiniz.

Not: IBM MQ Console , etkinken (birincil role sahip) yerel bir RDQM kuyruk yöneticisine bağlanabilir, ancak RDQM ' ye özgü özellikler sunmaz.


Yordam

- Yeni bir yerel kuyruk yöneticisi yaratmak için:

- a) Yarat düğmesini  gösterir.
- b) Yeni kuyruk yöneticisi için bir ad girin. Ad en çok 48 karakter içerebilir. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
- c) İsteğe bağlı: Kuyruk yöneticisinin dinleyeceği bir TCP/IP kapısı girin. Kapı numarası 65535 'i aşmamalıdır.
- d) **Oluştur**'u tıklatın. Yeni kuyruk yöneticisi yaratılır ve başlatılır.


- Yerel kuyruk yöneticisini başlatmak için:

- a) Listede başlatmak istediğiniz kuyruk yöneticisini bulun.

- b) Menüden **Başlat** seçeneğini belirleyin .


- Yerel bir kuyruk yöneticisini durdurmak için:



- a) Yerel kuyruk yöneticisi gerecindeki listeden, durdurmak istediğiniz kuyruk yöneticisini seçin.

- b) Menüden **Durdur** seçeneğini belirleyin .

- Yerel bir kuyruk yöneticisini silmek için:

- a) Kuyruk yöneticisi çalışıyorsa durdurun.

- b) Menüden **View configuration** (Yapılandırma görüntüle) seçeneğini belirleyin  ve **Delete queue manager** (Kuyruk yöneticisini sil) seçeneğini belirleyin.

- c) Kuyruk yöneticisini silmek istediğinizi doğrulamak için, kuyruk yöneticisinin adını doğrulama penceresine girin. Kuyruk yöneticisi ve ilişkili tüm nesnelere silinir.
- Yerel bir kuyruk yöneticisinin özelliklerini görüntülemek ve düzenlemek için:
 - a) Kuyruk yöneticisinin çalıştığını doğrulayın ve kuyruk yöneticisi listesinde bu kuyruk yöneticisini bulun.
 - b) Menüden **Görünüm yapılandırması** seçeneğini belirleyin  .
 - c) **Özellikler** sekmesinin seçildiğinden emin olun. Özellikleri görüntüleyin ve gerektiği şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Bir özellik ile ilgili bilgi için, özellik bilgilerini [Kuyruk yöneticisi özellikleri](#) içinde görüntüleyebilirsiniz.
- Yerel kuyruk yöneticisine ilişkin güvenlik ayarlarıyla çalışmak için:
 - a) Kuyruk yöneticisinin çalıştığını doğrulayın ve kuyruk yöneticisi listesinden seçin.
 - b) Menüden **Görünüm yapılandırması** seçeneğini belirleyin  .
 - c) **Güvenlik** sekmesinin seçildiğinden emin olun.
 - d) Kimlik doğrulama nesnelere, yetkilendirme kayıtlarıyla ya da kanal doğrulama nesnelere çalışabilirsiniz. Daha fazla bilgi için aşağıdaki konuları ziyaret edin:
 - [“IBM MQ Console : Kimlik doğrulama bilgileri nesnelere çalışma-Yalnızca LTS” sayfa 121](#)
 - [“IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışılması-Yalnızca LTS” sayfa 123](#)
 - [“IBM MQ Console: Kanal kimlik doğrulama kayıtlarıyla çalışma-Yalnızca LTS” sayfa 123](#)

V9.3.1 IBM MQ Console: Kimlik doğrulama bilgileri nesnelere çalışma-yalnızca CD


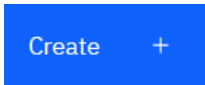
Bir kuyruk yöneticisine kimlik doğrulama bilgisi nesnelere eklemek ve silmek için konsolu kullanabilirsiniz. Ayrıca, özellikleri görüntüleyebilir ve ayarlayabilir ve nesnelere ilişkin yetki kayıtlarını yönetebilirsiniz.



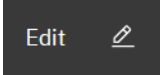


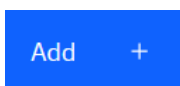
Bu görev hakkında

Kimlik doğrulama bilgileri görünümü, belirli bir kuyruk yöneticisi için var olan kimlik doğrulama bilgilerini listeler. Çalışmak için listeden tek tek kimlik doğrulama bilgilerini seçebilirsiniz.

Kuyruk yöneticisi kimlik doğrulama bilgileri, TLS (Transport Layer Security; İletim Katmanı Güvenliği) için IBM MQ desteğinin bir bölümünü oluşturur. Bu nesnelere, LDAP sunucularında OCSP ya da Sertifika İptal Listeleri (CRL) kullanılarak sertifika iptal denetimi gerçekleştirmek için gerekli tanımlamaları ve kullanıcı kimliği ve parola denetimini etkinleştirmek için gerekli tanımlamaları içerir.

Yordam

- Bir kuyruk yöneticisine ilişkin kimlik doğrulama bilgilerini görüntülemek için:
 - a) Kuyruk yöneticisinin çalıştığını doğrulayın ve kuyruk yöneticisi listesinden seçin.
 - b) Menüden **Görünüm yapılandırması** seçeneğini belirleyin  .
 - c) **Güvenlik** sekmesinin seçildiğinden emin olun.
 - d) Gezinme panosundan **Authentication information** (Kimlik doğrulama bilgileri) seçeneğini belirleyin.
- Bir kimlik doğrulama bilgisi nesnesi eklemek için:
 - a) Kimlik doğrulama bilgileri listesi görünümünde oluştur düğmesini  gösterir.

- b) Kimlik doğrulama bilgileri nesnesinin adını belirtin. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
- c) Kimlik doğrulama bilgileri nesnesinin tipini belirtin.
- d) Nesne tipine uygun ek bilgileri belirtin:
- **CRL LDAP için LDAP sunucusu adını** belirtin. Bu ad, isteğe bağlı bir kapı numarasıyla, LDAP sunucusunun çalıştığı anasistemin anasistem adı, IPv4 noktalı onlu adresi ya da IPv6 onaltılı gösterimidir. İsteğe bağlı olarak, LDAP sunucusuna erişen kullanıcı için bir kullanıcı adı ve parola belirtebilirsiniz.
 - **OCSP için OCSP yanıtlayıcısı URL** belirtin. Bu URL , sertifika iptalini denetlemek için kullanılan yanıt verenin URL 'sidir. Bu değer, OCSP yanıtlayıcısının anasistem adını ve kapı numarasını içeren bir HTTP URL olmalıdır. OCSP yanıtlayıcısı HTTP için varsayılan olan 80 numaralı kapıyı kullanıyorsa, kapı numarası atlanabilir. HTTP URL 'leri RFC 1738 'de tanımlanmıştır.
 - **IDPW OS** için ek gereksinim yoktur, ancak isteğe bağlı olarak bu kimlik doğrulama tipi için ek seçenekler belirtebilirsiniz.
 - **IDPW LDAP için, LDAP sunucusu adını ve Kısa kullanıcı** adını belirtin. LDAP sunucusu adı, LDAP sunucusunun çalıştığı anasistemin anasistem adı, IPv4 noktalı ondalık adresi ya da IPv6 onaltılı gösterimidir ve isteğe bağlı bir kapı numarasıdır. Kısa kullanıcı adı, LDAP kullanıcı kaydındaki bağlantı için kısa ad olarak kullanılan alandır. İsteğe bağlı olarak, bu kimlik doğrulama tipi için ek seçenekler belirtebilirsiniz.
- e) **Ekle**'yi tıklatın.
- Bir kimlik doğrulama bilgileri nesnesini silmek için:
 - a) Listedden silmek istediğiniz kimlik doğrulama bilgileri nesnesine ilişkin anahtar simgesini 
 - b) Nesne özellikleri görünümünde **Kimlik doğrulama bilgileri nesnesini sil** seçeneğini tıklatın.
 - c) **Sil** düğmesini tıklatarak kimlik doğrulama bilgileri nesnesini silmek istediğinizi doğrulayın. Nesne silinir.
 - Bir kimlik doğrulama bilgileri nesnesinin özelliklerini görüntülemek ve düzenlemek için:
 - a) Listedden görüntülemek istediğiniz kimlik doğrulama bilgileri nesnesine ilişkin anahtar simgesini  seçin.
 - b) Görüntülenen özellikleri düzenlemek için **Düzenle** düğmesini 
 - c) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir.
 - d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.
 - Bir kimlik doğrulama bilgi nesnesine ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Listedden yetki kaydını görüntülemek istediğiniz kimlik doğrulama bilgileri nesnesine ilişkin anahtar simgesini  seçin.
 - b) **Güvenlik** sekmesini seçin.
 - c) Var olan bir yetki kaydını düzenlemek ya da silmek için, menüden **Düzenle** ya da **Sil** seçeneğini belirleyin .
 - d) Yeni bir yetki kaydı eklemek için **Ekle** düğmesini  gösterir, yeni yetki kaydının ayrıntılarını belirtin ve **Oluştur**' u tıklatın.


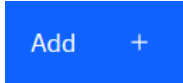
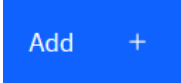
CD

Kullanıcıların ve grupların kuyruk yöneticilerine erişimini, o kullanıcı ya da grup için bir yetki kaydı belirterek denetleyebilirsiniz.

Bu görev hakkında

Bir ileti sistemi kullanıcısının ya da ileti sistemi kullanıcılarının belirli bir kuyruk yöneticisine ilişkin erişimini, yetki kayıtlarını kullanarak ayarlayabilirsiniz. İki tip yetki kaydı vardır: Genel yetkileri denetleyen **kuyruk yöneticisi erişimi** kayıtları ve hangi kullanıcıların ve grupların kuyruk yöneticisi için nesne yaratabileceğini denetleyen **yaratma izni** kayıtları.

Yordam

- Bir kuyruk yöneticisine ilişkin yetki kayıtlarını görüntülemek için:
 - a) Kuyruk yöneticisinin çalıştığını doğrulayın ve kuyruk yöneticisi listesinden seçin.
 - b) Menüden **Görünüm yapılandırması** seçeneğini belirleyin .
 - c) **Güvenlik** sekmesinin seçildiğinden emin olun.
 - d) Gezinme panosundan **Yetki kayıtları** seçeneğini belirleyin. Görünüm, yetki kayıtlarını iki bölmede göstererek genel yetki kayıtlarıyla çalışmanızı ve yetki kayıtları yaratmanızı sağlar.
- Genel yetki kaydı eklemek için:
 - a) **Kuyruk yöneticisi erişimi** liste görünümünde ekle düğmesini  tıklatın.
 - b) Bir kullanıcı ya da grup için bir yetki kaydı ekleyip ekleyeceğinizi seçin.
 - c) Yetki kaydı ekleyeceğiniz kullanıcının ya da grubun adını belirleyin (yetki kaydı bunu adı olarak alır).
 - d) Vermek istediğiniz yetkileri seçin.
 - e) **Oluştur**'u tıklatın.
- Yetki kaydı yaratma eklemek için:
 - a) **Yaratma izni** liste görünümünde ekle düğmesini  tıklatın.
 - b) Bir kullanıcı ya da grup için bir yetki kaydı ekleyip ekleyeceğinizi seçin.
 - c) Yetki kaydı ekleyeceğiniz kullanıcının ya da grubun adını belirleyin (yetki kaydı bunu adı olarak alır).
 - d) Yaratma yetkisi vermekte olduğunuz nesne tiplerini seçin.
 - e) **Oluştur**'u tıklatın.
- Bir yetki kaydını silmek için:
 - a) Silmek istediğiniz yetki kaydını seçin ve **Sil** seçeneğini belirleyin.
 - b) **Sil** düğmesini tıklatarak kimlik doğrulama bilgileri nesnesini silmek istediğinizi doğrulayın. Nesne silinir.
- Bir yetki kaydının özelliklerini görüntülemek ve düzenlemek için:
 - a) Görüntülemek istediğiniz yetki kaydını tıklatın.
 - b) Ayarları gerektiği gibi değiştirin ve değişikliklerinizi kaydetmek için **Kaydet** düğmesini tıklatın.

CD

Bir kuyruk yöneticisine kanal kimlik doğrulama kayıtları eklemek ve silmek için IBM MQ Console kullanabilirsiniz. Kanal kimlik doğrulama kayıtlarına ilişkin özellikleri görüntüleyebilir ve ayarlayabilirsiniz.

Bu görev hakkında

Kanal düzeyinde bağlanan sistemlere verilen erişim üzerinde daha kesin denetim sağlamak için kanal kimlik doğrulama kayıtlarını kullanabilirsiniz.

Güvenliği uygulamak için, kanallarınıza erişimi engellemek üzere engelleyici kanal kimlik doğrulama kayıtlarını kullanabilirsiniz. Belirtilen kullanıcılara erişime izin vermek için adres eşleme kanalı kimlik doğrulama kayıtlarını da kullanabilirsiniz. Kanal kimlik doğrulama kayıtlarıyla ilgili daha fazla bilgi için bkz. [Kanal kimlik doğrulama kayıtları](#).

Yordam

- Bir kuyruk yöneticisine ilişkin kanal kimlik doğrulama bilgilerini görüntülemek için:
 - a) Kuyruk yöneticisinin çalıştığını doğrulayın ve kuyruk yöneticisi listesinden seçin.

b) Menüden **Görünüm yapılandırması** seçeneğini belirleyin .

c) **Güvenlik** sekmesinin seçildiğinden emin olun.

d) Gezinme panosundan **Kanal kimlik doğrulaması** seçeneğini belirleyin.

- Bir kanal kimlik doğrulama kaydı eklemek için:

a) Oluştur düğmesini  gösterir.

b) Kullanmak istediğiniz kural tipini seçin. Bir **İzin Ver, Bloky** da **Uyarır** seçin.


c) Kanal kimlik doğrulama kuralı yapılandırmakta olduğunuz kimlik tipini seçin. Seçtiğiniz kural tipine bağlı olarak farklı kimlik tipleri kullanılabilir.

d) Belirlediğiniz kimlik için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Özel yaratma** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.

e) Kanal kimlik doğrulama kaydını oluşturmak için **Oluştur** 'u tıklatın.

Kanal kimlik doğrulama kayıtlarına ilişkin kullanılabilir ayarlar hakkında daha fazla bilgi için bkz. [Kanal kimlik doğrulama kayıtları](#) ve [SET CHLAUTH](#)


- Bir kanal kimlik doğrulama kaydını silmek için:

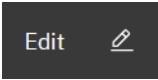
a) Silmek istediğiniz kanal kimlik doğrulama kaydının yanındaki anahtar simgesini  tıklatın.

b) Kanal kimlik doğrulamasını düzenle görünümünde **Kanal kimlik doğrulama nesnesini sil** seçeneğini tıklatın.

c) Kanal kimlik doğrulama kaydını silmek istediğinizi doğrulamak için **Sil** düğmesini tıklatın. Kanal kimlik doğrulama kaydı silindi.

- Bir kanal kimlik doğrulama kaydının özelliklerini görüntülemek ve düzenlemek için:

a) Düzenlemek ya da görüntülemek istediğiniz kanal kimlik doğrulama kaydının yanındaki anahtar simgesini  tıklatın. Özellikler görüntülenir.

b) Düzenle düğmesini tıklatın .

c) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir.






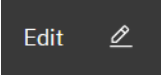

d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.


Dinleyici eklemek ve silmek, dinleyici başlatmak ve durdurmak, dinleyici özelliklerini görüntülemek ve ayarlamak ve bir dinleyiciye ilişkin yetki kayıtlarını yönetmek için IBM MQ Console özelliğini kullanabilirsiniz.

Bu görev hakkında

Dinleyiciler görünümü, belirli bir kuyruk yöneticisi için var olan dinleyicileri görüntüler. Çalışmak için tek tek dinleyicileri seçebilirsiniz.

Yordam

- Bir kuyruk yöneticisinin dinleyicilerini görüntülemek için:
 - a) Kuyruk yöneticisinin çalıştığını doğrulayın ve kuyruk yöneticisi listesinden seçin.
 - b) Menüden **Görünüm yapılandırması** seçeneğini belirleyin .
 - c) **Dinleyiciler** etiketini seçin.
- Dinleyici yaratmak için:
 - a) Oluştur düğmesini  gösterir.
 - b) Yaratmakta olduğunuz dinleyici için gereken bilgileri belirtin.
 - c) **Oluştur**'u tıklatın. Yeni dinleyici yaratılır.
- Dinleyici başlatmak için:
 - a) Listede başlatmak istediğiniz dinleyiciyi bulun.
 - b) Menüden **Başlat** seçeneğini belirleyin .
- Bir dinleyiciyi durdurmak için:
 - a) Listede başlatmak istediğiniz dinleyiciyi bulun.
 - b) Menüden **Durdur** seçeneğini belirleyin .
- Bir dinleyicinin özelliklerini görüntülemek ve düzenlemek için:
 - a) Listedeki dinleyiciyi bulun.
 - b) Menüden **Görünüm yapılandırması** seçeneğini belirleyin .
 - c) **Özellikler** sekmesinin seçildiğinden emin olun. Özellikleri düzenlemek için Düzenle düğmesini .
 - d) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Özelliklerle ilgili daha fazla bilgi için MQ Explorer belgelerinde [Dinleyici özellikleri](#) konusuna bakın.
 - e) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.
- Bir dinleyiciye ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Listedeki dinleyiciyi bulun.
 - b) Menüden **Görünüm yapılandırması** seçeneğini belirleyin .
 - c) **Security** (Güvenlik) sekmesini tıklatın.

- d) Kuyruk yöneticisi yetki kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın, bkz. [“IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışılması-Yalnızca LTS” sayfa 123.](#)
- Bir dinleyiciyi silmek için:
 - a) Listedeki dinleyiciyi bulun.
 - b) Menüden **Görünüm yapılandırması** seçeneğini belirleyin  .
 - c) **Dinleyiciyi sil** ' i tıklatın.

V9.3.1 IBM MQ Console: Uzak kuyruk yöneticisi eklenmesi-Yalnızca CD

Uzak bir sistemde çalışan bir kuyruk yöneticisini denetlemek için IBM MQ Console kullanabilirsiniz.

Başlamadan önce

- Uzak sistemdeki kuyruk yöneticisini, uzaktan yönetilebilmesi için hazırlamanız gerekir; bkz. [“Komut satırını kullanarak IBM MQ Console ' e uzak kuyruk yöneticisi eklenmesi-Yalnızca LTS” sayfa 126](#) içindeki adım [“1” sayfa 127](#), [“2” sayfa 127](#), [“3” sayfa 127](#) ve [“4” sayfa 127](#) .
- IBM MQ Console' den uzak bağlantıları da etkinleştirmeniz gerekir. Daha fazla bilgi için [Uzak kuyruk yöneticisi bağlantı davranışını yapılandırma](#) başlıklı konuya bakın.

Bu görev hakkında

Uzak bağlantı ayrıntılarını belirtmek için JSON biçiminde bir istemci bağlantı tanımlaması çizelgesi (CCDT) kullanılır. Bir metin düzenleyicisini kullanarak JSON CCDT oluşturabilir (bkz. [“Komut satırını kullanarak IBM MQ Console ' e uzak kuyruk yöneticisi eklenmesi-Yalnızca LTS” sayfa 126](#) [“5” sayfa 128](#) . adım) ya da IBM MQ Console kullanarak bir JSON CCDT oluşturabilirsiniz.

Diğer bir seçenek olarak, uzak kuyruk yöneticisini eklerken doğrudan bağlantı ayrıntılarını belirterek IBM MQ Console içinden CCDT yaratabilirsiniz.

Uzak kuyruk yöneticisini IBM MQ Console ' e bağlamak için, gereken tüm görevler için komut satırını (uzak kuyruk yöneticisini hazırlamaya ve CCDT yaratmaya ek olarak) kullanabilirsiniz. Bkz. [“Komut satırını kullanarak IBM MQ Console ' e uzak kuyruk yöneticisi eklenmesi-Yalnızca LTS” sayfa 126.](#)



Uyarı: aşağıdaki iletileri alırsanız:

```
MQWB2026E: The request to connect to the remote queue manager 'rqmgr-qmgr_name' failed
with the error message:
'JM5CC0051: The property 'JMS_IBM_MQMD_AccountingToken' should be set using type '[B',
not 'java.lang.Object'.'
```

byte[] java nesne tipi beklenirken, hesap simgesine bir java.lang.Object geçirmeyi denerdiniz.

Yordam

- Varolan bir CCDT belirterek uzak kuyruk yöneticisi eklemek için:
 - a) Giriş sayfasından **Uzak kuyruk yöneticisine bağlan** seçeneğini tıklatın.
 - b) Uzak kuyruk yöneticisinin adını belirtin.
 - c) İsteğe bağlı olarak, kuyruk yöneticisi için benzersiz bir ad belirtin. Benzersiz bir ad belirtmezseniz, gerçek ad "remote-" önekiyle birlikte kullanılır.
 - d) **JSON CCDT kullanarak bağlan** seçeneğinin belirlendiğinden emin olun.
 - e) **Göz At** düğmesini tıklatın ve kullanmak istediğiniz JSON CCDT ' yi içeren dosyayı seçin.
 - f) Kullanıcı sayfasına gitmek için **İleri** düğmesini tıklatın ve isteğe bağlı olarak uzak kuyruk yöneticisine bağlanmak için bir kullanıcı adı ve parola belirtin. Bu bilgileri belirtmezseniz, kimlik doğrulama bilgileri uzak bağlantı konfigürasyon dosyasından alınır.

- g) Sertifika sayfasına gitmek için **İleri** düğmesini tıklayın. CCDT ' iniz "transmissionSecurity" bilgisini belirtiyorsa, bu bilgi kullanılır. İsteğe bağlı olarak bir sertifikayı yapılandırabilirsiniz (base64 kodlu genel anahtar olarak) ve bu, genel güven deposuna eklenir.

Sertifika, güvenli depoya eklenmeden önce `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt` içinde geçici olarak depolanır. Bağlantı başarıyla eklendiğinde, sertifika bu konumdan silinir.

- h) Özet sayfasını görüntülemek için **İleri** düğmesini tıklayın. Önceki sayfaları yeniden ziyaret etmek ve düzeltme yapmak için **Geri** düğmesini kullanabilirsiniz. Bilgilerden memnunsanız, uzak kuyruk yöneticisine bağlanmak için **Bağlan** düğmesini tıklayın.
- Uzak kuyruk yöneticisi eklemek ve bağlantı bilgilerini el ile belirtmek için:
 - a) Giriş sayfasından **Uzak kuyruk yöneticisine bağlan** seçeneğini tıklayın.
 - b) Uzak kuyruk yöneticisinin adını belirtin.
 - c) İsteğe bağlı olarak, kuyruk yöneticisi için benzersiz bir ad belirtin. Benzersiz bir ad belirtmezseniz, gerçek ad "remote-" önekiyle birlikte kullanılır.
 - d) **El ile Girdi** seçeneğini belirleyin.
 - e) Bağlantının kullanacağı istemci bağlantı kanalının adını girin.
 - f) Uzak kuyruk yöneticisinin çalıştığı anasistemin adını belirtin. Uzak MQ kuruluşları saptanırsa, anasistem adları görüntülenir ve bağlanmak istediğiniz uzak kuyruk yöneticisinin anasistemini seçebilirsiniz. Bazı ağ yapılandırmalarında, uzak MQ yönetim ortamları saptanamadı. Bu durumda, anasistem adını ve kapıyı el ile ekleyin.
 - g) Kullanıcı sayfasına gitmek için **İleri** düğmesini tıklayın ve isteğe bağlı olarak uzak kuyruk yöneticisine bağlanmak için bir kullanıcı adı ve parola belirtin. Bu bilgileri belirtmezseniz, kimlik doğrulama bilgileri uzak bağlantı konfigürasyon dosyasından alınır.
 - h) Sertifika sayfasına gitmek için **İleri** düğmesini tıklayın. Açılan listeden bir SSL CipherSpec seçebilirsiniz. İsteğe bağlı olarak bir sertifikayı yapılandırabilirsiniz (base64 kodlu genel anahtar olarak) ve bu, genel güven deposuna eklenir.

Sertifika, güvenli depoya eklenmeden önce `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt` içinde geçici olarak depolanır. Bağlantı başarıyla eklendiğinde, sertifika bu konumdan silinir.
 - i) Özet sayfasını görüntülemek için **İleri** düğmesini tıklayın. Önceki sayfaları yeniden ziyaret etmek ve düzeltme yapmak için **Geri** düğmesini kullanabilirsiniz. Bilgilerden memnunsanız, uzak kuyruk yöneticisine bağlanmak için **Bağlan** düğmesini tıklayın.

Belirlediğiniz bağlantı bilgileri, Web dizininizdeki CCDT dosyasına yazılır. Yol: `WLP_USER_DIR/generated.ccdt/ccdt-uniqueName`.

Sonuçlar

Uzak kuyruk yöneticisi, IBM MQ Console'deki uzak kuyruk yöneticisi listesinde görünür. Bağlantının başarılı olması koşuluyla, uzak kuyruk yöneticisinin nesnelere, yerel bir kuyruk yöneticisinin nesnelereyle çalıştığınız şekilde denetleyebilirsiniz.

V9.3.1 **Yalnızca komut satırı CD 'sini kullanarak IBM MQ Console uzak kuyruk yöneticisine ekleme**

Komut satırındaki `setmqweb remote` komutunu kullanarak IBM MQ Console ' e bir uzak kuyruk yöneticisi ekleyebilirsiniz. Uzak kuyruk yöneticisi, IBM MQ Console ile aynı sistemde farklı bir kuruluşta çalışan bir kuyruk yöneticisi ya da farklı bir sistemde çalışan bir kuyruk yöneticisi olabilir.

Başlamadan önce

Not: Bu görevdeki adımlar MQSC komutlarını çalıştırmanızı gerektirir:

- **ALW** AIX, Linux, and Windows üzerinde, bir `runmqsc` komut isteminden MQSC komutları yayınlayın. Bkz. `runmqsc` altında MQSC komutlarının etkileşimli olarak çalıştırılması ve `runmqsc`

altındaki metin dosyalarından MQSC komutlarının çalıştırılması. Bu görev için AIX, Linux, and Windows üzerinde çalışıyorsanız, QM1: komutunu kullanan bir runmqsc komut istemi açın.

```
runmqsc QM1
```

- **IBM i** IBM üzerinde, Komut Dosyası dosyasında bir komut listesi oluşturun ve dosyayı **STRMQMMQSC** komutunu kullanarak çalıştırın. Bkz. [Administration using MQSC commands on IBM i](#).
- **z/OS** z/OS üzerinde, komuta bağlı olarak MQSC komutları çeşitli kaynaklardan verilebilir. Bkz. [IBM MQ for z/OS üzerinde MQSC ve PCF komutlarını yayınlatabileceğiniz kaynaklar](#).

mqweb sunucusunun IBM MQ Console ile uzak kuyruk yöneticisi bağlantılarına izin verecek şekilde yapılandırıldığından emin olun. Daha fazla bilgi için [Uzak kuyruk yöneticisi bağlantıları davranışını yapılandırma](#) başlıklı konuya bakın.

Yordam

1. Uzak kuyruk yöneticisinde, **DEFINE CHANNEL** MQSC komutunu kullanarak kuyruk yöneticisinin uzaktan yönetilmesini sağlamak için bir sunucu bağlantısı kanalı yaratın. Örneğin, QM1 kuyruk yöneticisi için bir sunucu bağlantısı kanalı QM1 . SVRCONN yaratmak üzere şu MQSC komutunu girin:

```
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
```

DEFINE CHANNEL ve kullanılabilir seçenekler hakkında daha fazla bilgi için bkz. [DEFINE CHANNEL](#).

2. Uygun bir kullanıcının kuyruk yöneticisini ve kuyruk yöneticisiyle ilişkilendirilmiş MQ nesnelere denetleme yetkisine sahip olduğunu doğrulayın.

- **ALW** AIX, Linux, and Windows üzerinde standart bir komut satırında **setmqaut** denetim komutunu kullanın.
- **z/OS** z/OS üzerinde, yetkili kullanıcıya kuyruk yöneticisine erişim vermek için RACF profillerini tanımlayın.

Örneğin, AIX, Linux, and Windows sistemlerinde `exampleUser` kullanıcıya QM1 kuyruk yöneticisine erişim yetkisi vermek için şu denetim komutunu girin:

```
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp
```

Bu yetkili kullanıcı, aşağıdaki kullanıcılardan biri olabilir:

- Bu kuyruk yöneticisini uzaktan denetlemek istediğiniz sistemde IBM MQ Console ' u çalıştıran mqweb sunucusunu başlatan kullanıcı kimliğiyle aynı kullanıcı kimliği.
 - “7” sayfa 106. adımda **setmqweb remote** komutunda yer alan bir kullanıcı kimliği ve parolayla eşleşen bir kullanıcı kimliği. **setmqweb remote** komutuna kullanıcı kimliği ve parola eklenerek, IBM MQ Console kuyruk yöneticisine bağlandığında kimlik doğrulaması için bu kullanıcı kimliği ve parola kullanılır.
 - Kanal güvenlik kurallarına göre belirlenen bir kullanıcı kimliği. Örneğin, sunucu bağlantısı kanalında, uzaktan yönetim için IBM MQ Console ' u kullandığınız IP adresinden bağlantılara izin verecek bir kanal doğrulama kuralı ayarlayabilir ve tüm bu bağlantıları, kuyruk yöneticisini kullanma yetkisi olan belirli bir kullanıcı kimliğiyle eşleyebilirsiniz. Daha fazla bilgi için bkz. [Kanallar için yeni CHLAUTH kuralları oluşturma](#).
3. **ALW**
- Uzak kuyruk yöneticisinde çalışan bir dinleyici yoksa, **DEFINE LISTENER** MQSC komutunu kullanarak gelen ağ bağlantılarını kabul edecek bir dinleyici yaratın.

Örneğin, uzak kuyruk yöneticisi QM1 için 1414 kapısında bir dinleyici REMOTE . LISTENER yaratmak üzere şu MQSC komutunu girin:

```
runmqsc QM1
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
end
```

4. **START LISTENER** MQSC komutunu kullanarak dinleyicinin çalıştığını doğrulayın.

ALW Örneğin, QM1 kuyruk yöneticisi için REMOTE . LISTENER dinleyicisini başlatmak üzere AIX, Linux, and Windows üzerinde şu MQSC komutunu girin:

```
runmqsc QM1
START LISTENER(REMOTE.LISTENER)
end
```

Z/OS Örneğin, z/OS üzerinde dinleyiciyi başlatmak için şu MQSC komutunu girin:

```
/cpf START LISTENER TRPTYPE(TCP) PORT(1414)
```

z/OS üzerinde bir dinleyici başlatmadan önce kanal başlatıcı adres alanının başlatılması gerektiğini unutmayın.

5. Uzak kuyruk yöneticisi bağlantı bilgilerini içeren bir JSON CCDT dosyası yaratın:

- Uzaktan bağlanmak istediğiniz kuyruk yöneticisiyle aynı kuruluşla ilişkilendirilmiş IBM MQ Console dosyasını kullanarak bir CCDT dosyası oluşturun.

Ana Sayfa panosundan **Bağlantı dosyasını karşıdan yükle** döşemesini tıklatın.

- Bağlantıyı tanımlayan bir JSON biçimi CCDT dosyası yaratın. JSON biçimi CCDT oluşturma hakkında daha fazla bilgi için bkz. [JSON biçimi CCDT ' nin yapılandırılması](#).

CCDT dosyası name, clientConnection ve type bilgilerini içermelidir. İsteğe bağlı olarak transmissionSecurity bilgileri gibi ek bilgiler ekleyebilirsiniz. Tüm CCDT kanal özneliği tanımlamalarına ilişkin ek bilgi için [CCDT kanal özneliği tanımlamalarının tam listesibaşlıklı konuya](#) bakın.

Aşağıdaki örnek, uzak kuyruk yöneticisi bağlantısı için temel bir JSON CCDT dosyasını göstermektedir. Kanalın adını, "1" sayfa 105. adımda oluşturulan örnek sunucu-bağlantı kanalı ile aynı ada ve bağlantı kapısını, dinleyici tarafından kullanılan kapıyla aynı değere ayarlar. Bağlantı anasistemi, örnek uzak kuyruk yöneticisinin (QM1) çalıştığı sistemin anasistem adına ayarlanır:

```
{
  "channel": [ {
    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [ {
        "host": "example.com",
        "port": 1414
      } ],
      "queueManager": "QM1"
    },
    "type": "clientConnection"
  } ]
}
```

6. JSON CCDT dosyasını IBM MQ Console ' un çalıştığı sisteme kopyalayın.

7. IBM MQ Console çalıştıran kuruluşta, uzak kuyruk yöneticisi bilgilerini IBM MQ Console yapılandırmasına eklemek için **setmqweb remote** komutunu kullanın.

En az IBM MQ Console ' e uzak kuyruk yöneticisi eklemek için kuyruk yöneticisi adını, kuyruk yöneticisi için benzersiz bir ad (aynı kuyruk yöneticisi adına sahip olabilecek diğer uzak kuyruk yöneticilerini ayırt etmek için) ve kuyruk yöneticisine ilişkin CCDT URL belirtmeniz gerekir. Benzersiz ad, IBM MQ Console içindeki görünen addir; bu nedenle, bunun bir uzak kuyruk yöneticisi olduğunu açıkça belirten bir ad belirtin; örneğin, "remote-QM2". Uzak kuyruk yöneticisi bağlantısı için kullanılacak kullanıcı adı ve parola ya da güvenli depo ve anahtar deposu ayrıntıları gibi belirtebileceğiniz birkaç ek seçenek

vardır. **setmqweb remote** komutuyla belirtilebilecek parametrelerin tam listesi için bkz. [setmqweb remote](#).

Örneğin, örnek uzak kuyruk yöneticisi QM1' i örnek CCDT dosyasını kullanarak eklemek için aşağıdaki komutu girin:

```
setmqweb remote add -uniqueName "MACHINEAQM1" -qmgrName "QM1" -ccdtURL "c:\myccdts\ccdt.json"
```

Sonuçlar

Uzak kuyruk yöneticisi, uzak bağlantı listesi yenilendiğinde IBM MQ Console içindeki uzak kuyruk yöneticisi listesinde görünür. Bağlantının başarılı olması koşuluyla, uzak kuyruk yöneticisinin nesnelere, yerel bir kuyruk yöneticisinin nesnelere çalıştığınız şekilde denetleyebilirsiniz.

Örnek

Aşağıdaki örnek, QM1kuyruk yöneticisi için uzak kuyruk yöneticisi bağlantısını ayarlar. IBM MQ Console , kullanıcıya verilen yetkiye dayalı olarak kuyruk yöneticisini denetleme yetkisine sahiptir `exampleUser`. Uzak kuyruk yöneticisi bağlantı bilgilerini yapılandırmak için **setmqweb remote** komutu kullanıldığında bu kullanıcının kimlik bilgileri IBM MQ Console ' e sağlanır.

1. QM1 uzak kuyruk yöneticisinin bulunduğu sistemde, bir sunucu bağlantısı kanalı ve bir dinleyici yaratılır. Dinleyici başlatılır ve `exampleUser` kullanıcılarına kuyruk yöneticisini denetleme yetkisi verilir. Örneğin, AIX, Linux, and Windowsüzerinde aşağıdaki komutları çalıştırın:

```
runmqsc QM1
#Define the server connection channel that will accept connections from the Console
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
# Define the listener to use for the connection from the Console
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
# Start the listener
START LISTENER(REMOTE.LISTENER)
end

#Set mq authorization for exampleUser to access the queue manager
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp
```

2. IBM MQ Console ' in çalıştığı sistemde, aşağıdaki bağlantı bilgileriyle bir `QM1_ccdt.json` dosyası yaratılır:

```
{
  "channel": [{
    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [{
        "host": "example.com",
        "port": 1414
      }],
      "queueManager": "QM1"
    },
    "type": "clientConnection"
  }]
}
```

3. IBM MQ Console ' in çalıştığı sistemde, QM1 kuyruk yöneticisine ilişkin uzak kuyruk yöneticisi bağlantı bilgileri mqweb sunucusuna eklenir. `exampleUser` kimlik bilgileri bağlantı bilgilerinde bulunur:

```
setmqweb remote add -uniqueName "remote-QM1" -qmgrName "QM1" -ccdtURL
"c:\myccdts\QM1_ccdt.json" -username "exampleUser" -password "password"
```

4. IBM MQ Console , QM1uzak kuyruk yöneticisini gösterir.

V9.3.1 IBM MQ Console: Nesnelere çalışılması-Yalnızca CD

Her IBM MQ kuyruk yöneticisiyle ilişkilendirilmiş birkaç farklı nesne tipi vardır.

Bu görev hakkında

Aşağıdaki IBM MQ nesnesi tipleriyle çalışmak için konsolu kullanabilirsiniz:

- Kuyruklar
- Olay nesneleri:
 - Konular
 - Abonelikler
- Uygulama nesneleri:
 - Bağlantılar
 - Uygulama kanalları
 - Uygulama kanalı örnekleri
- MQ ağ nesneleri:
 - Bağlı kuyruk yöneticileri
 - Kuyruk yöneticisi kanalları
 - Kuyruk yöneticisi kanal eşgörünümleri

Yordam

Bir IBM MQ nesnesiyle çalışmak için:


1. Kuyruk yöneticisi liste görünümünde, çalışmak istediğiniz nesnelere erişmek için kuyruk yöneticisini tıklatın.
2. Çalışmak istediğiniz nesne tipini seçmek için Kuyruklar, Olaylar, Uygulamalar ya da MQ ağ sekmesini tıklatın.
3. Nesnelere çalışmaya ilişkin ayrıntılı yönergeler için aşağıdaki konulardan birine bakın.

V9.3.1 IBM MQ Console: Kuyruklarla çalışılması-Yalnızca CD

Belirli bir kuyruk yöneticisi için var olan kuyrukları **Kuyruklar** sekmesinde görüntüleyebilirsiniz. Kuyruk ekleyebilir ve silebilir, kuyruktaki iletileri ekleyebilir ve temizleyebilir, iletilere göz atabilir, bir kuyruğun özelliklerini görüntüleyebilir ve ayarlayabilir ve bir kuyruğun yetki kayıtlarını yönetebilirsiniz.

Bu görev hakkında

Kuyruklar görünümü, belirli bir kuyruk yöneticisine ilişkin kuyrukları listeler. Kuyruk listesine bir kuyruk yöneticisini tıklatıp **Kuyruklar** etiketini seçerek erişebilirsiniz. Çalışmak istediğiniz kuyrukları listeden tek tek seçebilirsiniz.

 z/OS üzerindeki kuyruklara ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

Yordam

- Kuyruk eklemek için:

a) **Kuyruklar** sekmesinde, oluştur düğmesini

Create +

b) Yaratmak istediğiniz kuyruk tipini seçin:

- Yerel kuyruk-iletileri ait olduğu kuyruk yöneticisi içinde saklar.
- Diğer ad kuyruğu-aynı kuyruk yöneticisindeki başka bir kuyruğa işaretçi.
- Uzak kuyruk-başka bir kuyruk yöneticisindeki başka bir kuyruğa işaretçi.
- Model kuyruğu-dinamik bir kuyruk yöneticisi yaratıldığında kullanılan bir kuyruk için şablon.

c) Yaratmakta olduğunuz kuyruk tipi için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Özel yaratmaseçeneğini** belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.

d) **Oluştur**'u tıklatın. Yeni kuyruk yaratılır.

• İletileri kuyruğa koymak için:

a) Kuyruk listesi görünümündeki listede ileti eklemek istediğiniz kuyruğu tıklatın. Model kuyruğu seçemezsiniz.

Create +


b) Oluştur düğmesini tıklatın

c) Kuyruğa koymak istediğiniz iletiyi girin.

d) **Oluştur**'u tıklatın.

• Kuyruktaki iletileri temizlemek için:

a) Kuyruk listesinden iletileri temizlemek istediğiniz yerel kuyruğu tıklatın.

b) Kuyruğu temizle simgesini 

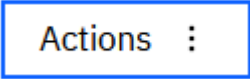
c) Kuyruğu temizlemek istediğinizi doğrulamak için **Kuyruğu temizle**düğmesini tıklatın.

• Kuyruktaki iletilere göz atmak için, kuyruk listesi görünümünde kuyruğu tıklatın. Bu kuyruktaki iletilerin bir listesi görüntülenir.

• Bir kuyruğu silmek için:

a) Kuyruk listesinde, silmek istediğiniz yerel kuyruğu tıklatın.


Actions :


b) İşlemler düğmesini  tıklatın ve **Kuyruğu sil** seçeneğini belirleyin.

c) Kuyruğu silmek istediğinizi doğrulamak için **Sil**düğmesini tıklatın. Kuyruk silindi.

• Bir kuyruğun özelliklerini görüntülemek ve düzenlemek için:

a) Düzenlemek istediğiniz kuyruğun yanındaki **View configuration** (Yapılandırmayı görüntüle)

seçeneğini belirleyin .

Edit 


b) Düzenle düğmesini tıklatın

c) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Özellikler hakkında bilgi için bkz. MQ Explorer belgelerinde [Kuyruk özellikleri](#)

d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.

• Bir kuyruğa ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:

a) Düzenlemek istediğiniz kuyruğun yanındaki **View configuration** (Yapılandırmayı görüntüle)

seçeneğini belirleyin .

b) **Security** (Güvenlik) sekmesini tıklatın.

c) Kuyruk yöneticisi yetki kayıtları için açıklanan yetki kayıtlarıyla çalışın. Bkz. ["IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışılması-Yalnızca LTS" sayfa 123.](#)

9.3.3

Bir kuyrukla ilişkili IBM MQ nesnelerini görüntülemek için:

a) Düzenlemek istediğiniz kuyruğun yanındaki **İlişkili nesnelere görüntüle** seçeneğini belirleyin .

b) Görüntülenen panodaki nesnelere görüntüleyin. Listelenen nesnelere her birine ilişkin daha fazla ayrıntı görüntülemek için bağlantıları tıklayın.


Hangi uygulamaların kuyruklara ileti koymakta olduğunu görüntülemek ve farklı kuyruklar arasındaki ilişkileri görmek için bu panoyu kullanabilirsiniz. Bu, sorunları belirlemenize ve çözümlenize yardımcı olabilir.

V9.3.1 IBM MQ Console: Konuyla çalışma-Yalnızca CD

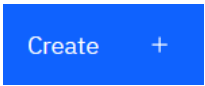


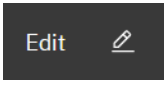
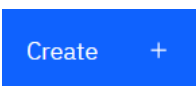
Konu eklemek ve silmek ve bir konunun özelliklerini görüntülemek ve ayarlamak için IBM MQ Console özelliğini kullanabilirsiniz.

Bu görev hakkında

Konular görünümü, belirli bir kuyruk yöneticisi için var olan konuları listeler. Konulara kuyruk yöneticisi **Olaylar** sekmesinden erişebilirsiniz. Üzerinde çalışılacak konuları listeden tek tek seçebilirsiniz.


 z/OS ile ilgili bir konuya ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

Yordam

- Konu eklemek için:
 - a) Kuyruk yöneticisi görünümünden **Olaylar** sekmesini açın ve **Konular**'ı tıklayın.
 - b) Oluştur düğmesini  gösterir.
 - c) Oluşturmakta olduğunuz konu için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Özel yaratma** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
 - d) **Oluştur**'u tıklayın. Yeni konu oluşturulur.
- Bir konuyu silmek için:
 - a) Silmek istediğiniz konunun yanındaki anahtar simgesini  tıklayın.
 - b) Kuyruğu düzenle görünümünde **Konuyu sil** 'i tıklayın.
 - c) Konuyu silmek istediğinizi doğrulamak için **Sil** düğmesini tıklayın. Konu silindi.
- Bir konunun özelliklerini görüntülemek ve düzenlemek için:
 - a) Düzenlemek istediğiniz konunun yanındaki anahtar simgesini  tıklayın.
 - b) Düzenle düğmesini tıklayın .
 - c) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Özelliklerle ilgili bilgi için MQ Explorer belgelerinde Konu özellikleri konusuna bakın.
 - d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklayın.
- Bir konuyla ilgili ileti yayınlamak için en az bir eşleşen aboneliğiniz olmalıdır.
 - a) Konu listesinde yayınlamak istediğiniz konuyu tıklayın.
 - b) Eşleşen abonelik adını tıklayın.
 - c) Oluştur düğmesini tıklayın .
 - d) Yayınlamak istediğiniz iletiyi girin.

Put

- e) Yerleřtir düğmesini gösterir. İleti, eşleşen tüm aboneliklere yazılır.
- Bir konuya abone olmak için bkz. [“IBM MQ Console: Aboneliklerle çalışma-Yalnızca LTS” sayfa 132:](#)
- Bir konuya ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:

- a) Yetki kayıtlarını düzenlemek istediğiniz konunun yanındaki anahtar simgesini  tıklatın.
- b) **Security** (Güvenlik) sekmesini tıklatın.
- c) Kuyruk yöneticisi yetki kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın, bkz. [“IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışılması-Yalnızca LTS” sayfa 123.](#)

V9.3.1 IBM MQ Console: Aboneliklerle çalışılması-Yalnızca CD

Abonelik eklemek ve silmek için IBM MQ Console olanağını kullanabilir ve bir aboneliğin özelliklerini görüntüleyebilir ve ayarlayabilirsiniz.

Bu görev hakkında



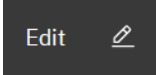
Abonelikler görünümü, belirli bir kuyruk yöneticisi için var olan abonelikleri listeler. Aboneliklere kuyruk yöneticisi **Olaylar** sekmesinden erişebilirsiniz. Üzerinde çalışılacak konuları listeden tek tek seçebilirsiniz. Çalışmak için listeden tek tek abonelikler seçebilirsiniz.

Abonelikler hakkında daha fazla bilgi için bkz. [Aboneler ve abonelikler](#) ve [DEFINE SUB](#).

z/OS

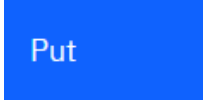
z/OSaboneliğine ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

Yordam

- Abonelik eklemek için:
 - a) Kuyruk yöneticisi görünümünden **Olaylar** sekmesini açın ve **Abonelikler**' i tıklatın.
 - b) Yönetilen ya da yönetilmeyen bir abonelik oluşturmak isteyip istemediğinizi seçin.
 - c) Oluşturmakta olduğunuz abonelik için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Özel yaratma** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
 - d) **Oluştur**'u tıklatın. Yeni abonelik oluşturulur.
- Bir aboneliği silmek için:
 - a) Silmek istediğiniz aboneliğin yanındaki anahtar simgesini  tıklatın.
 - b) Kuyruğu düzenle görünümünde **Aboneliğin silinmesi** öğesini tıklatın.
 - c) Aboneliği silmek istediğinizi doğrulamak için **Sil** düğmesini tıklatın. Abonelik silindi.
- Bir aboneliğin özelliklerini görüntülemek ve düzenlemek için:
 - a) Düzenlemek istediğiniz aboneliğin yanındaki anahtar simgesini  tıklatın.
 - b) Düzenle düğmesini tıklatın 
 - c) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir.
 - d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.
- Aboneliğin abone olduğu konuyla ilgili bir ileti yayınlamak için:
 - a) Konusunu abonelik listesinde yayınlamak istediğiniz aboneliği tıklatın.

b) Oluştur düğmesini  gösterir.

c) Yayınlamak istediğiniz iletiyi girin.


d) Yerleştir düğmesini  gösterir. İleti, yayınladığınız konuyla eşleşen tüm aboneliklere yazılır.

V9.3.1 IBM MQ Console: Kuyruk yöneticisi kanallarıyla çalışılması-Yalnızca CD

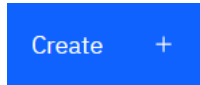




IBM MQ Console kuyruk yöneticisi kanallarıyla çalışabilirsiniz: Kuyruk yöneticisi kanalları ekleyebilir ve bunları silebilir, kanalları başlatabilir ve durdurabilir, kanalları sıfırlayabilir ve çözebilir ve kanalları ve ping kanalları ekleyebilirsiniz. Ayrıca, bir kuyruk yöneticisi kanalının özelliklerini görüntüleyebilir ve ayarlayabilir ve kanala ilişkin yetki kayıtlarını yönetebilirsiniz.

Bu görev hakkında

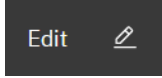
Kuyruk yöneticisi kanalı, bir ağ üzerinden kuyruk yöneticileri arasında ileti iletmek için kullanılan mantıksal bir iletişim bağlantısıdır. Kuyruk yöneticisi kanal görünümü, kaç kanalın çalıştığını, kaç tanesinin yeniden denendiğini ve kaç tanesinin durdurulduğunu hızlı bir şekilde gösteren bir pano içerir.






 z/OS üzerindeki bir kanala ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

Yordam

- Bir kuyruk yöneticisi kanalı eklemek için:
 - a) Kuyruk yöneticisi görünümünden **MQ ağ** sekmesini açın ve **Kuyruk yöneticisi kanalları** 'nı tıklatın ve yarat düğmesini  tıklatın.
 - b) Yaratmak istediğiniz kuyruk yöneticisi kanalı tipini seçin ve sonraki düğmesini tıklatın .
 - c) Oluşturmakta olduğunuz kanal için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Özel yaratma** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
 - d) **Oluştur**'u tıklatın. Yeni kanal, **etkin değil** durumuyla oluşturulur.
- Bir kuyruk yöneticisi kanalı başlatmak için:
 - a) Listede başlatmak istediğiniz kanalı bulun.
 - b) Menüden **Başlat** seçeneğini belirleyin .
- Bir kuyruk yöneticisi kanalını durdurmak için:
 - a) Listede durdurmak istediğiniz kanalı bulun.
 - b) Menüden **Durdur** seçeneğini belirleyin .
- Bir kuyruk yöneticisi kanalının özelliklerini görüntülemek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Yapılandırılmaları görüntüle** seçeneğini belirleyin .

c) **Özellikler** sekmesinin seçildiğinden emin olun. Özellikleri düzenlemek için Düzenle düğmesini




- d) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Özelliklerle ilgili daha fazla bilgi için MQ Explorer belgelerinde [Kanal özellikleri](#) konusuna bakın.
- e) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.
- Bir kuyruk yöneticisi kanalını ilk durumuna getirmek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Gelişmiş** seçeneğini belirleyin .
 - c) **İlk Durumuna Getir** bölümünde bir ileti sıra numarası belirtin.
İki uç, gönderilecek sonraki iletinin sıra numarası konusunda aynı fikirde olmadığı için, bir kanalı başlatmazsa ilk durumuna getirmeniz gerekir. İleti sıra numarası, bu sayıyı belirtir.
 - d) **Reset Channel**(Kanalı Sıfırla) seçeneğini tıklatın.
 - Bir gönderen ya da sunucu kanalını çözmek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Gelişmiş** seçeneğini belirleyin .
 - c) **Çözüm** bölümünde, **İletileri iletim kuyruğuna geri yükle** ya da **İletileri at** seçeneğini tıklatarak yürürlükteki ileti kümesini kesinleştirmeyi ya da geri yüklemeyi seçin.
 - Bir kuyruk yöneticisi kanalına ping komutu göndermek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Ping** (Ping) seçeneğini belirleyin .
 - Bir kuyruk yöneticisi kanalına ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Görünüm yapılandırması** seçeneğini belirleyin .
 - c) **Security** (Güvenlik) sekmesini tıklatın.
 - d) Kuyruk yöneticisi yetki kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın, bkz. [“IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışılması-yalnızca CD” sayfa 100.](#)
 - Bir kuyruk yöneticisi kanalını silmek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Delete channel**(Kanalı sil) seçeneğini tıklatın.

V9.3.1 IBM MQ Console: Uygulama kanallarıyla çalışma-yalnızca CD

Uygulama kanallarıyla çalışmak için IBM MQ Console 'i kullanabilirsiniz: kanallar ekleyebilir ve var olanları silebilir, kanalları başlatabilir ve durdurabilir, kanalları sıfırlayabilir ve çözebilir ve kanallara ping kanalları gönderebilirsiniz. Ayrıca, bir uygulama kanalının özelliklerini görüntüleyebilir ve ayarlayabilir ve kanala ilişkin yetki kayıtlarını yönetebilirsiniz.

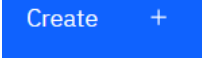
Bu görev hakkında

Uygulama kanalı, uygulamalar tarafından bir ağ üzerinden bir kuyruk yöneticisine bağlanmak için kullanılan mantıksal bir iletişim bağlantısıdır. Uygulama kanalı görünümü, kaç kanal çalıştığından, kaç yeniden denemenin gerçekleştirildiğinden ve kaç tanesinin durdurulduğundan hızlı bir görünüm gösteren bir pano içerir.

 z/OS üzerindeki bir kanala ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

Yordam


- Bir uygulama kanalı eklemek için:
 - a) Kuyruk yöneticisi görünümünden **Uygulamalar** sekmesini açın ve **Uygulama kanalları** 'nı tıklatın ve

oluştur düğmesini  gösterir.

- b) Sonraki düğmesini tıklatın




- c) Oluşturmakta olduğunuz kanal için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Özel yaratma** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
 - d) **Oluştur**'u tıklatın. Yeni kanal, **etkin değil** durumuyla oluşturulur.
- Bir uygulama kanalını başlatmak için:
 - a) Listede başlatmak istediğiniz kanal bulun.

- b) Menüden **Başlat** seçeneğini belirleyin .

- Bir uygulama kanalını durdurmak için:

- a) Listede durdurmak istediğiniz kanal bulun.


- b) Menüden **Durdur** seçeneğini belirleyin .

- Bir uygulama kanalının özelliklerini görüntülemek için:

- a) Listedeki kanal bulun.

- b) Menüden **Görünüm yapılandırması** seçeneğini belirleyin .

- c) **Özellikler** sekmesinin seçildiğinden emin olun. Özellikleri düzenlemek için Düzenle düğmesini




- d) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Özelliklerle ilgili daha fazla bilgi için MQ Explorer belgelerinde [Kanal özellikleri](#) konusuna bakın.

- e) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.





- Bir uygulama kanalını ilk durumuna getirmek için:

- a) Listedeki kanal bulun.

- b) Menüden **Gelişmiş** seçeneğini belirleyin .

- c) **İlk Durumuna Getir** bölümünde bir ileti sıra numarası belirtin.

İki uç, gönderilecek sonraki iletinin sıra numarası konusunda aynı fikirde olmadığı için, bir kanalı başlalmazsa ilk durumuna getirmeniz gerekir. İleti sıra numarası, bu sayıyı belirtir.

- d) **Reset Channel**(Kanalı Sıfırla) seçeneğini tıklatın.
- Bir gönderen ya da sunucu kanalını çözmek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Gelişmiş** seçeneğini belirleyin .
 - c) **Çözüm** bölümünde, **İletileri iletim kuyruğuna geri yükle** ya da **İletileri at** seçeneğini tıklatarak yürürlükteki ileti kümesini kesinleştirmeyi ya da geri yüklemeyi seçin.
- Bir kanala ping komutu göndermek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Ping** (Ping) seçeneğini belirleyin .
- Bir uygulama kanalına ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Security** (Güvenlik) sekmesini tıklatın.
 - d) Kuyruk yöneticisi yetki kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın, bkz. [“IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışılması-yalnızca CD” sayfa 100](#) .
- Bir uygulama kanalını silmek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Delete channel**(Kanalı sil) seçeneğini tıklatın.

V 9.3.2 IBM MQ Console: Uygulamayla çalışma-Yalnızca CD

Bir kuyruk yöneticisine bağlı uygulamalara ilişkin bilgileri görüntülemek için IBM MQ Console olanağını kullanabilirsiniz.

Bu görev hakkında

Bir uygulama, bir sunucu bağlantısı kanalı kullanılarak bir ağ üzerinden bir kuyruk yöneticisine bağlanır. Uygulamalar görünümü, bir kuyruk yöneticisine kaç uygulamanın bağlı olduğuna ilişkin hızlı bir görünüm gösteren bir pano içerir.

Yordam

- Uygulama bilgilerini görüntülemek için:
 - a) Kuyruk yöneticisi görünümünden **Uygulamalar** sekmesini açın.
 - b) Uygulamalar görünümünü açmak için **Bağlı uygulamalar** ' ı tıklatın.
 - c) Bir uygulamanın birden çok örneği varsa, her bir eşgörünümün ayrıntılarını görüntülemek için aşağı oku tıklatın.
 - d) Daha fazla ayrıntı almak için görünümdeki nesnelere tıklatın.

V 9.3.5 z/OS IBM MQ Console: Depolama sınıflarıyla çalışma-Yalnızca CD

z/OS kuyruk yöneticilerine depolama sınıfları eklemek, bunları görüntülemek, silmek ve güncellemek için IBM MQ Console komutunu kullanabilirsiniz.

Bu görev hakkında

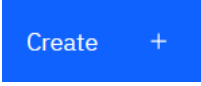
Depolama sınıfları görünümü, belirli bir kuyruk yöneticisi için var olan depolama sınıflarını listeler.

Depolama sınıflarına , kuyruk yöneticisi **Kuyruklar** sekmesindeki kenar çubuğundan erişir.

Depolama sınıflarına ilişkin ek bilgi için [Storage classes for IBM MQ for z/OS and DEFINE STGLASS](#) başlıklı konuya bakın.

Yordam

- Depolama sınıfı eklemek için:
 - a) Kuyruk yöneticisi görünümünden **Kuyruklar** sekmesini açın ve **Depolama sınıfları**' nı tıklatın.

b) **Storage classes** (Depolama sınıfları) ekranında **Create**  tıklatın.


c) Oluşturmakta olduğunuz depolama sınıfı için gerekli bilgileri sağlayın.

Varsayılan olarak, değer sağlamanız gereken önerilen özellikler alt sınırı görüntülenir. **Özel yaratma** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.

d) **Oluştur**  düğmesi gösterilir.

Yeni depolama sınıfı oluşturulur.


- Bir depolama sınıfını silmek için:

a) Silmek istediğiniz depolama sınıfının yanındaki anahtar düğmesini  gösterir.


b) Edit storage class (Depolama sınıfını düzenle) görünümünde **Delete storage class**(Depolama sınıfını sil) seçeneğini tıklatın.

c) Kuyruğu silmek istediğinizi doğrulamak için **Sil** düğmesini tıklatın. Depolama sınıfı silinir.

- Bir depolama sınıfının özelliklerini görüntülemek ve düzenlemek için:

a) Düzenlemek istediğiniz depolama sınıfının yanındaki anahtar düğmesini  gösterir.

b) Düzenle düğmesini tıklatın

Edit 

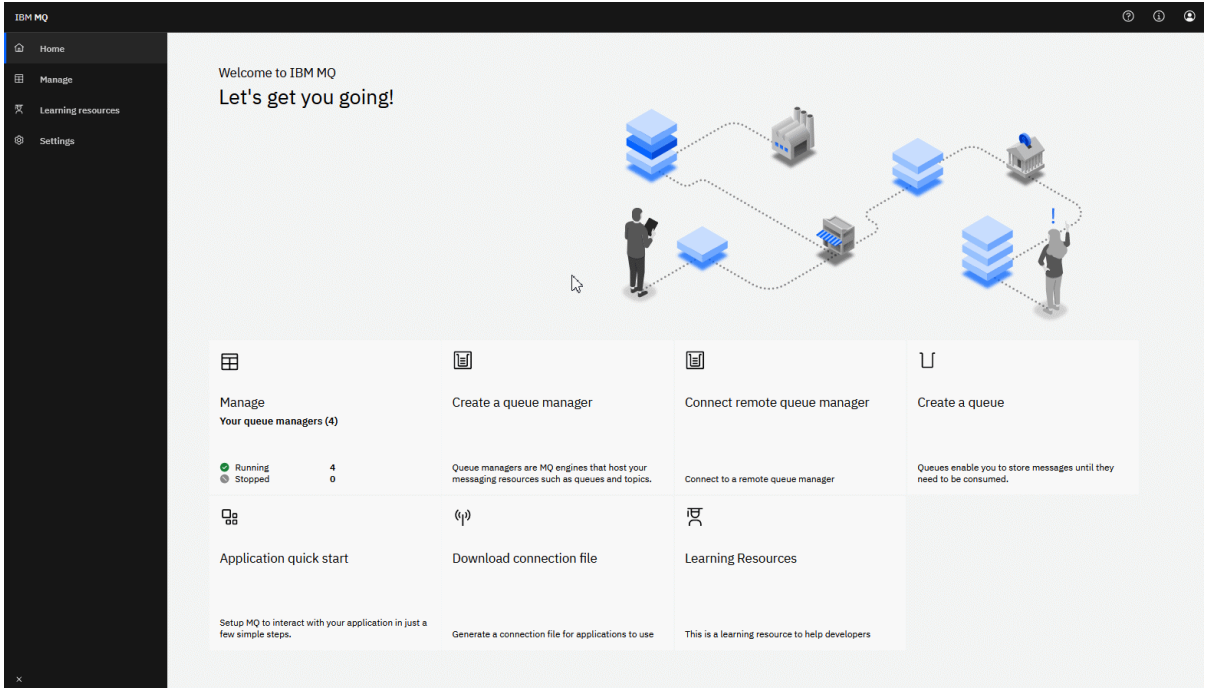
c) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca yaratma sırasında ayarlanabilir.

d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.

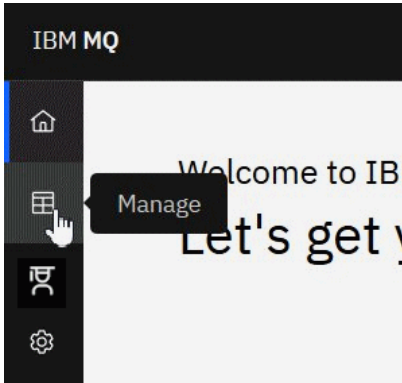
LTS

IBM MQ Console -Yalnızca LTS için hızlı tur

IBM MQ Console ' da ilk oturum açtığınızda, iniş sayfasına yönlendirilirsiniz. Buradan, var olan kuyruk yöneticilerini yönetmeyi seçebilir, bir kuyruk yöneticisi ya da kuyruk yaratabilir, bazı eğitim konularına gidebilir ya da IBM Documentation içinde IBM MQ ürün bilgilerini açabilirsiniz. Yeni ya da var olan kuyruk yöneticileri ve uygulamalar arasında ileti sisteminin hızlı ve kolay bir şekilde ayarlanması sürecinde size yol gösteren uygulama hızlı başlangıcını da başlatabilirsiniz.



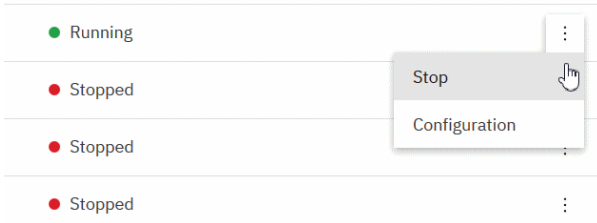
Alternatif olarak, IBM MQ nesnelarını hemen yönetmeye başlamak için yönet simgesini tıklatabilirsiniz.



Yönet görünümü başlangıçta kuyruk yöneticilerini ve yürürlükteki durumlarını gösterir. Ayrıca, yeni kuyruk yöneticileri yaratabilir ve uzak kuyruk yöneticilerini bağlayabilirsiniz.

Queue manager name ↑	Version	Status
MpgQM2	9.1.5.0	Running
MyPortQmgr	9.1.5.0	Stopped
QM1	9.1.5.0	Stopped
qmanotherone	9.1.5.0	Stopped
qmq	9.1.5.0	Stopped
qnd	9.1.5.0	Stopped
qne	9.1.5.0	Stopped
RPilotFinalTest	9.1.5.0	Running
sdwe	9.1.5.0	Stopped
Steves_qm	9.1.5.0	Stopped

Her kuyruk yöneticisinin, çalışmakta olan bir kuyruk yöneticisini durdurmanızı ya da yapılandırmanızı ya da durdurulmuş bir kuyruk yöneticisini başlatmanızı ya da silmenizi sağlayan bir menüsü vardır.



Kuyruk yöneticisine ilişkin Yetki Kayıtları, Kimlik Doğrulama Bilgileri nesnelere ve Kanal Kimlik Doğrulama kayıtları, kuyruk yöneticisinin **Yapılandırma** sayfasının **Güvenlik** sekmesinde bulunabilir; burada yenilerini oluşturabilir ve ekleyebilirsiniz.


Gösterge panosunu açmak için çalışmakta olan bir kuyruk yöneticisinin adını tıklayın.

The screenshot shows the IBM MQ Queue Manager interface for 'MpgQM'. The top navigation bar includes 'Queues', 'Topics', 'Subscriptions', and 'Communication'. The main content area displays three capacity indicators: '1 Full capacity' (1 user-defined, 0 system), '2 Greater than 50% capacity' (1 user-defined, 1 system), and '61 Less than 1% capacity' (15 user-defined, 46 system). Below these is a table of queues with columns for Name, Type, Depth %, and Maximum depth. The table lists several queues, including 'AMQ.MQEXPLORER.5E68C78B20805304' and 'Queueue1' through 'Queueue4'. A 'Create +' button is visible in the top right of the table area.


Name ↑	Type	Depth %	Maximum depth
AMQ.MQEXPLORER.5E68C78B20805304	Local	0%	0/5000
Queueue1	Local	0%	0/1
Queueue2	Local	0%	0/5000
Queueue3	Local	0%	0/5000
Queueue4	Local	0%	0/5000

Kuyruk yöneticisi gösterge panosundan aşağıdaki işlemleri gerçekleştirebilirsiniz:


Kuyruklar sekmesinde:

- Yeni kuyruklar yarat
- Var olan kuyrukları yapılandır 
- Var olan iletileri görüntülemek ve yenilerini yaratmak için bir kuyruk adını tıklatın



Konular sekmesinde:


- Yeni konular yarat
- Var olan konuları yapılandır  gösterir
- Eşleşen abonelikleri görüntülemek için konu adını tıklatın

Abonelikler sekmesinde:

- Yeni yönetilen ya da yönetilmeyen abonelikler oluştur
- Var olan abonelikleri yapılandır  gösterir



İletişim etiketinde:

- Dinleyiciler:
 - Dinleyicileri başlat, durdur ve yapılandır 
 - Yeni dinleyici yarat
- Kuyruk Yöneticisi Kanalları:
 - Kanalları başlat, durdur, ping ve yapılandır 
 - Yeni kanallar oluştur
 - Kanalları sıfırla (**Gelişmiş** menü öğesinden)

- Kanallardaki belirsiz iletileri çözümlerle (**Gelişmiş** menü öğesinden)
- Uygulama Kanalları:
 - Kanalları başlat, durdur, ping ve yapılandır 
 - Yeni kanallar oluştur
 - Kanalları sıfırla (**Gelişmiş** menü öğesinden)
 - Kanallardaki belirsiz iletileri çözümlerle (**Gelişmiş** menü öğesinden)

LTS IBM MQ Console: Yerel kuyruk yöneticileriyle çalışma-Yalnızca LTS

Yerel kuyruk yöneticilerini Yönet görünümünün en üst düzeyinden oluşturur, yapılandırır ve

denetleyebilirsiniz  **Manage**  gösterilir.

Bu görev hakkında

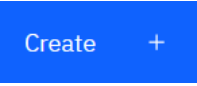



Multi Yönet görünümü, şu IBM MQ Console ürününün çalıştığı IBM MQ kuruluşu. Aynı sistemdeki farklı IBM MQ kuruluşlarıyla ilişkili kuyruk yöneticileri listelenmez.



z/OS z/OS işletim sisteminde Manage görünümü, IBM MQ Console ile aynı sürümde olan kuyruk yöneticilerini listeler ve IBM MQ Console ' in çalıştığı sistemde tanımlanır. IBM MQ Console sürümünden farklı bir sürümdeki kuyruk yöneticileri listelenmez.

Çalışmak istediğiniz kuyruk yöneticilerini listeden tek tek seçebilirsiniz.

Not: IBM MQ Console , eşlenmiş veri kuyruğu yöneticilerini (RDIMM ' ler) desteklemez.

Yordam

- Yeni bir yerel kuyruk yöneticisi yaratmak için:
 - a) Yarat düğmesini  gösterir.
 - b) Yeni kuyruk yöneticisi için bir ad girin. Ad en çok 48 karakter içerebilir. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
 - c) İsteğe bağlı: Kuyruk yöneticisinin dinleyeceği bir TCP/IP kapısı girin. Kapı numarası 65535 'i aşmamalıdır.
 - d) **Oluştur**'u tıklattın. Yeni kuyruk yöneticisi yaratılır ve başlatılır.
- Yerel kuyruk yöneticisini başlatmak için:
 - a) Listede başlatmak istediğiniz kuyruk yöneticisini bulun.
 - b) Menüden **Başlat** seçeneğini belirleyin .
- Yerel bir kuyruk yöneticisini durdurmak için:
 - a) Yerel kuyruk yöneticisi gerecindeki listeden, durdurmak istediğiniz kuyruk yöneticisini seçin.
 - b) Menüden **Durdur** seçeneğini belirleyin .
- Yerel bir kuyruk yöneticisini silmek için:
 - a) Kuyruk yöneticisi çalışıyorsa durdurun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin  ve **Kuyruk yöneticisini sil** seçeneğini belirleyin.

- c) Kuyruk yöneticisini silmek istediğinizi doğrulamak için, kuyruk yöneticisinin adını doğrulama penceresine girin. Kuyruk yöneticisi ve ilişkili tüm nesnelere silinir.
- Yerel bir kuyruk yöneticisinin özelliklerini görüntülemek ve düzenlemek için:
 - a) Kuyruk yöneticisinin çalıştığını doğrulayın ve kuyruk yöneticisi listesinde bu kuyruk yöneticisini bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin  .
 - c) **Özellikler** sekmesinin seçildiğinden emin olun. Özellikleri görüntüleyin ve gerektiği şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Bir özellik ile ilgili bilgi için, özellik bilgilerini [Kuyruk yöneticisi özellikleri](#) içinde görüntüleyebilirsiniz.
- Yerel kuyruk yöneticisine ilişkin güvenlik ayarlarıyla çalışmak için:
 - a) Kuyruk yöneticisinin çalıştığını doğrulayın ve kuyruk yöneticisi listesinden seçin.
 - b) Menüden **Yapılandır** seçeneğini belirleyin  .
 - c) **Güvenlik** sekmesinin seçildiğinden emin olun.
 - d) Kimlik doğrulama nesnelere, yetkilendirme kayıtlarıyla ya da kanal doğrulama nesnelere çalışabilirsiniz. Daha fazla bilgi için aşağıdaki konuları ziyaret edin:
 - [“IBM MQ Console : Kimlik doğrulama bilgileri nesnelere çalışma-Yalnızca LTS” sayfa 121](#)
 - [“IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışması-Yalnızca LTS” sayfa 123](#)
 - [“IBM MQ Console: Kanal kimlik doğrulama kayıtlarıyla çalışma-Yalnızca LTS” sayfa 123](#)

IBM MQ Console : Kimlik doğrulama bilgileri nesnelere çalışma-Yalnızca LTS



Bir kuyruk yöneticisine kimlik doğrulama bilgisi nesnelere eklemek ve silmek için konsolu kullanabilirsiniz. Ayrıca, özellikleri görüntüleyebilir ve ayarlayabilir ve nesnelere ilişkin yetki kayıtlarını yönetebilirsiniz.



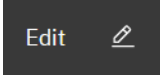



Bu görev hakkında

Kimlik doğrulama bilgileri görünümü, belirli bir kuyruk yöneticisi için var olan kimlik doğrulama bilgilerini listeler. Çalışmak için listeden tek tek kimlik doğrulama bilgilerini seçebilirsiniz.

Kuyruk yöneticisi kimlik doğrulama bilgileri, TLS (Transport Layer Security; İletim Katmanı Güvenliği) için IBM MQ desteğinin bir bölümünü oluşturur. Bu nesnelere, LDAP sunucularında OCSP ya da Sertifika İptal Listeleri (CRL) kullanılarak sertifika iptal denetimi gerçekleştirmek için gerekli tanımlamaları ve kullanıcı kimliği ve parola denetimini etkinleştirmek için gerekli tanımlamaları içerir.

Yordam

- Bir kuyruk yöneticisine ilişkin kimlik doğrulama bilgilerini görüntülemek için:
 - a) Kuyruk yöneticisinin çalıştığını doğrulayın ve kuyruk yöneticisi listesinden seçin.
 - b) Menüden **Yapılandır** seçeneğini belirleyin  .
 - c) **Güvenlik** sekmesinin seçildiğinden emin olun.
 - d) Gezinme panosundan **Authentication information** (Kimlik doğrulama bilgileri) seçeneğini belirleyin.
- Bir kimlik doğrulama bilgileri nesnesi eklemek için:
 - a) Kimlik doğrulama bilgileri listesi görünümünde oluştur düğmesini  gösterir.

- b) Kimlik doğrulama bilgileri nesnesinin adını belirtin. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
- c) Kimlik doğrulama bilgileri nesnesinin tipini belirtin.
- d) Nesne tipine uygun ek bilgileri belirtin:
- **CRL LDAP için LDAP sunucusu adını** belirtin. Bu ad, isteğe bağlı bir kapı numarasıyla, LDAP sunucusunun çalıştığı anasistemin anasistem adı, IPv4 noktalı onlu adresi ya da IPv6 onaltılı gösterimidir. İsteğe bağlı olarak, LDAP sunucusuna erişen kullanıcı için bir kullanıcı adı ve parola belirtebilirsiniz.
 - **OCSP için OCSP yanıtlayıcısı URL** belirtin. Bu URL , sertifika iptalini denetlemek için kullanılan yanıt verenin URL 'sidir. Bu değer, OCSP yanıtlayıcısının anasistem adını ve kapı numarasını içeren bir HTTP URL olmalıdır. OCSP yanıtlayıcısı HTTP için varsayılan olan 80 numaralı kapıyı kullanıyorsa, kapı numarası atlanabilir. HTTP URL 'leri RFC 1738 'de tanımlanmıştır.
 - **IDPW OS** için ek gereksinim yoktur, ancak isteğe bağlı olarak bu kimlik doğrulama tipi için ek seçenekler belirtebilirsiniz.
 - **IDPW LDAP için, LDAP sunucusu adını ve Kısa kullanıcı** adını belirtin. LDAP sunucusu adı, LDAP sunucusunun çalıştığı anasistemin anasistem adı, IPv4 noktalı ondalık adresi ya da IPv6 onaltılı gösterimidir ve isteğe bağlı bir kapı numarasıdır. Kısa kullanıcı adı, LDAP kullanıcı kaydındaki bağlantı için kısa ad olarak kullanılan alandır. İsteğe bağlı olarak, bu kimlik doğrulama tipi için ek seçenekler belirtebilirsiniz.
- e) **Ekle**'yi tıklatın.
- Bir kimlik doğrulama bilgileri nesnesini silmek için:
 - a) Listedeki silmek istediğiniz kimlik doğrulama bilgileri nesnesine ilişkin anahtar simgesini 
 - b) Nesne özellikleri görünümünde **Kimlik doğrulama bilgileri nesnesini sil** seçeneğini tıklatın.
 - c) **Sil** düğmesini tıklatarak kimlik doğrulama bilgileri nesnesini silmek istediğinizi doğrulayın. Nesne silinir.
 - Bir kimlik doğrulama bilgileri nesnesinin özelliklerini görüntülemek ve düzenlemek için:
 - a) Listedeki görüntülemek istediğiniz kimlik doğrulama bilgileri nesnesine ilişkin anahtar simgesini  seçin.
 - b) Görüntülenen özellikleri düzenlemek için **Düzenle** düğmesini 
 - c) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir.
 - d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.
 - Bir kimlik doğrulama bilgi nesnesine ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Listedeki yetki kaydını görüntülemek istediğiniz kimlik doğrulama bilgileri nesnesine ilişkin anahtar simgesini  seçin.
 - b) **Güvenlik** sekmesini seçin.
 - c) Var olan bir yetki kaydını düzenlemek ya da silmek için, menüden **Düzenle** ya da **Sil** seçeneğini belirleyin .
 - d) Yeni bir yetki kaydı eklemek için **Ekle** düğmesini  gösterir, yeni yetki kaydının ayrıntılarını belirtin ve **Oluştur**' u tıklatın.



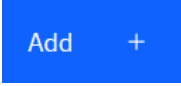


LTS

Kullanıcıların ve grupların kuyruk yöneticilerine erişimini, o kullanıcı ya da grup için bir yetki kaydı belirterek denetleyebilirsiniz.

Bu görev hakkında

Bir ileti sistemi kullanıcısının ya da ileti sistemi kullanıcılarının belirli bir kuyruk yöneticisine ilişkin erişimini, yetki kayıtlarını kullanarak ayarlayabilirsiniz. İki tip yetki kaydı vardır: genel yetkileri denetleyen ve kuyruk yöneticisi için hangi kullanıcıların ve grupların nesne yaratabileceğini denetleyen yetki kayıtları.

Yordam

- Bir kuyruk yöneticisine ilişkin yetki kayıtlarını görüntülemek için:
 - a) Kuyruk yöneticisinin çalıştığını doğrulayın ve kuyruk yöneticisi listesinden seçin.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Güvenlik** sekmesinin seçildiğinden emin olun.
 - d) Gezinme panosundan **Yetki kayıtları** seçeneğini belirleyin. Görünüm, yetki kayıtlarını iki bölmede göstererek genel yetki kayıtlarıyla çalışmanızı ve yetki kayıtları yaratmanızı sağlar.
- Genel yetki kaydı eklemek için:
 - a) Yetki kayıtları listesi görünümünde ekle düğmesini  gösterir.
 - b) Bir kullanıcı ya da grup için bir yetki kaydı ekleyip ekleyeceğinizi seçin.
 - c) Yetki kaydı ekleyeceğiniz kullanıcının ya da grubun adını belirleyin (yetki kaydı bunu adı olarak alır)
 - d) Vermek istediğiniz yetkileri seçin (yetkiler hakkında daha fazla bilgi için bkz. --)
 - e) **Oluştur**'u tıklatın.
- Yetki kaydı yaratma eklemek için:
 - a) Ekle düğmesini  gösterir.
 - b) Bir kullanıcı ya da grup için bir yetki kaydı ekleyip ekleyeceğinizi seçin.
 - c) Yetki kaydı ekleyeceğiniz kullanıcının ya da grubun adını belirleyin (yetki kaydı bunu adı olarak alır)
 - d) Yaratma yetkisi vermekte olduğunuz nesne tiplerini seçin.
 - e) **Oluştur**'u tıklatın.
- Bir yetki kaydını silmek için:
 - a) Silmek istediğiniz yetki kaydına ilişkin menüyü  gösterir ve **Sil** seçeneğini belirleyin.
 - b) **Sil** düğmesini tıklatarak kimlik doğrulama bilgileri nesnesini silmek istediğinizi doğrulayın. Nesne silinir.
- Bir yetki kaydının özelliklerini görüntülemek ve düzenlemek için:
 - a) Silmek istediğiniz yetki kaydına ilişkin menüyü  gösterir ve **Düzenle** seçeneğini belirleyin.
 - b) Ayarları gerektiği gibi değiştirin ve değişikliklerinizi kaydetmek için **Kaydet** düğmesini tıklatın.

Bir kuyruk yöneticisine kanal kimlik doğrulama kayıtları eklemek ve silmek için IBM MQ Console kullanabilirsiniz. Kanal kimlik doğrulama kayıtlarına ilişkin özellikleri görüntüleyebilir ve ayarlayabilirsiniz.


Bu görev hakkında

Kanal düzeyinde bağlanan sistemlere verilen erişim üzerinde daha kesin denetim sağlamak için kanal kimlik doğrulama kayıtlarını kullanabilirsiniz.

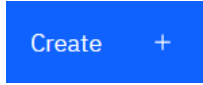
Güvenliği uygulamak için, kanallarınıza erişimi engellemek üzere engelleyici kanal kimlik doğrulama kayıtlarını kullanabilirsiniz. Belirtilen kullanıcılara erişime izin vermek için adres eşleme kanalı kimlik doğrulama kayıtlarını da kullanabilirsiniz. Kanal kimlik doğrulama kayıtlarıyla ilgili daha fazla bilgi için bkz. [Kanal kimlik doğrulama kayıtları](#).

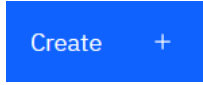
Yordam

- Bir kuyruk yöneticisine ilişkin kanal kimlik doğrulama bilgilerini görüntülemek için:
 - a) Kuyruk yöneticisinin çalıştığını doğrulayın ve kuyruk yöneticisi listesinden seçin.

- b) Menüden **Yapılandır** seçeneğini belirleyin .
- c) **Güvenlik** sekmesinin seçildiğinden emin olun.
- d) Gezinme panosundan **Kanal kimlik doğrulaması** seçeneğini belirleyin.


- Bir kanal kimlik doğrulama kaydı eklemek için:

- a) Oluştur düğmesini  gösterir.
- b) Kullanmak istediğiniz kural tipini seçin. Bir **İzin Ver, Bloky** da **Uyarır** seçin.
- c) Kanal kimlik doğrulama kuralı yapılandırmakta olduğunuz kimlik tipini seçin. Seçtiğiniz kural tipine bağlı olarak farklı kimlik tipleri kullanılabilir.
- d) Belirlediğiniz kimlik için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.


- e) Kanal kimlik doğrulama kaydını oluşturmak için oluştur düğmesini .

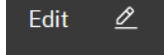
Kanal kimlik doğrulama kayıtlarına ilişkin kullanılabilir ayarlar hakkında daha fazla bilgi için bkz. [Kanal kimlik doğrulama kayıtları](#) ve [SET CHLAUTH](#)

- Bir kanal kimlik doğrulama kaydını silmek için:

- a) Silmek istediğiniz kanal kimlik doğrulama kaydının yanındaki anahtar simgesini  tıklatın.
- b) Kanal kimlik doğrulamasını düzenle görünümünde **Kanal kimlik doğrulama nesnesini sil** seçeneğini tıklatın.
- c) Kanal kimlik doğrulama kaydını silmek istediğinizi doğrulamak için **Sil** düğmesini tıklatın. Kanal kimlik doğrulama kaydı silindi.

- Bir kanal kimlik doğrulama kaydının özelliklerini görüntülemek ve düzenlemek için:

- a) Düzenlemek ya da görüntülemek istediğiniz kanal kimlik doğrulama kaydının yanındaki anahtar simgesini  tıklatın. Özellikler görüntülenir.

- b) Düzenle düğmesini tıklatın .
- c) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir.
- d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.

Yalnızca LTS

Uzak bir sistemde çalışan bir kuyruk yöneticisini denetlemek için IBM MQ Console kullanabilirsiniz.

Başlamadan önce

- Uzak sistemdeki kuyruk yöneticisini, uzaktan yönetilebilmesi için hazırlamanız gerekir; bkz. “Komut satırını kullanarak IBM MQ Console ' e uzak kuyruk yöneticisi eklenmesi-Yalnızca LTS” sayfa 126 içindeki adım “1” sayfa 127, “2” sayfa 127, “3” sayfa 127 ve “4” sayfa 127 .
- IBM MQ Console' den uzak bağlantıları da etkinleştirmeniz gerekir. Daha fazla bilgi için [Uzak kuyruk yöneticisi bağlantı davranışını yapılandırma](#) başlıklı konuya bakın.

Bu görev hakkında

Uzak bağlantı ayrıntılarını belirtmek için JSON biçiminde bir istemci bağlantı tanımlaması çizelgesi (CCDT) kullanılır. Bir metin düzenleyicisini kullanarak JSON CCDT oluşturabilir (bkz. “Komut satırını kullanarak IBM MQ Console ' e uzak kuyruk yöneticisi eklenmesi-Yalnızca LTS” sayfa 126 “5” sayfa 128 . adım) ya da IBM MQ Console kullanarak bir JSON CCDT oluşturabilirsiniz.

Diğer bir seçenek olarak, uzak kuyruk yöneticisini eklerken doğrudan bağlantı ayrıntılarını belirterek IBM MQ Console içinden CCDT yaratabilirsiniz.

Uzak kuyruk yöneticisini IBM MQ Console ' e bağlamak için, gereken tüm görevler için komut satırını (uzak kuyruk yöneticisini hazırlamaya ve CCDT yaratmaya ek olarak) kullanabilirsiniz. >Bkz. “Komut satırını kullanarak IBM MQ Console ' e uzak kuyruk yöneticisi eklenmesi-Yalnızca LTS” sayfa 126



Uyarı: aşağıdaki iletileri alırsanız:

```
MQWB2026E: The request to connect to the remote queue manager 'mqgr-qmgr_name' failed with the error message:
'JM5CC0051: The property 'JMS_IBM_MQMD_AccountingToken' should be set using type '[B', not 'java.lang.Object'.'
```

byte [] java nesne tipi beklenirken, hesap simgesine bir java.lang.Object geçirmeyi denediniz.

Yordam

- Varolan bir CCDT belirterek uzak kuyruk yöneticisi eklemek için:
 - a) Giriş sayfasından **Uzak kuyruk yöneticisine bağlan** seçeneğini tıklatın.
 - b) Uzak kuyruk yöneticisinin adını belirtin.
 - c) İsteğe bağlı olarak, kuyruk yöneticisi için benzersiz bir ad belirtin. Benzersiz bir ad belirtmezseniz, gerçek ad "remote-" önekiyle birlikte kullanılır.
 - d) **JSON CCDT kullanarak bağlan** seçeneğinin belirlendiğinden emin olun.
 - e) **Göz At** düğmesini tıklatın ve kullanmak istediğiniz JSON CCDT ' yi içeren dosyayı seçin.
 - f) Kullanıcı sayfasına gitmek için **İleri** düğmesini tıklatın ve isteğe bağlı olarak uzak kuyruk yöneticisine bağlanmak için bir kullanıcı adı ve parola belirtin. Bu bilgileri belirtmezseniz, kimlik doğrulama bilgileri uzak bağlantı konfigürasyon dosyasından alınır.
 - g) Sertifika sayfasına gitmek için **İleri** düğmesini tıklatın. CCDT ' iniz "transmissionSecurity" bilgisini belirtiyorsa, bu bilgi kullanılır. İsteğe bağlı olarak bir sertifikayı yapıştırabilirsiniz (base64 kodlu genel anahtar olarak) ve bu, genel güven deposuna eklenir.

Sertifika, güvenli depoya eklenmeden önce `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt` içinde geçici olarak depolanır. Bağlantı başarıyla eklendiğinde, sertifika bu konumdan silinir.

- h) Özet sayfasını görüntülemek için **İleri** düğmesini tıklayın. Önceki sayfaları yeniden ziyaret etmek ve düzeltme yapmak için **Geri** düğmesini kullanabilirsiniz. Bilgilerden memnunsanız, uzak kuyruk yöneticisine bağlanmak için **Bağlan** düğmesini tıklayın.
- Uzak kuyruk yöneticisi eklemek ve bağlantı bilgilerini el ile belirtmek için:
 - a) Giriş sayfasından **Uzak kuyruk yöneticisine bağlan** seçeneğini tıklayın.
 - b) Uzak kuyruk yöneticisinin adını belirtin.
 - c) İsteğe bağlı olarak, kuyruk yöneticisi için benzersiz bir ad belirtin. Benzersiz bir ad belirtmezseniz, gerçek ad "remote-" önekiyle birlikte kullanılır.
 - d) **El ile Girdi** seçeneğini belirleyin.
 - e) Bağlantının kullanacağı istemci bağlantı kanalının adını girin.
 - f) Uzak kuyruk yöneticisinin çalıştığı anasistemin adını belirtin. Uzak MQ kuruluşları saptanırsa, anasistem adları görüntülenir ve bağlanmak istediğiniz uzak kuyruk yöneticisinin anasistemi seçebilirsiniz. Bazı ağ yapılandırmalarında, uzak MQ yönetim ortamları saptanamadı. Bu durumda, anasistem adını ve kapyı el ile ekleyin.
 - g) Kullanıcı sayfasına gitmek için **İleri** düğmesini tıklayın ve isteğe bağlı olarak uzak kuyruk yöneticisine bağlanmak için bir kullanıcı adı ve parola belirtin. Bu bilgileri belirtmezseniz, kimlik doğrulama bilgileri uzak bağlantı konfigürasyon dosyasından alınır.
 - h) Sertifika sayfasına gitmek için **İleri** düğmesini tıklayın. Açılan listeden bir SSL CipherSpec seçebilirsiniz. İsteğe bağlı olarak bir sertifikayı yapıştırabilirsiniz (base64 kodlu genel anahtar olarak) ve bu, genel güven deposuna eklenir.

Sertifika, güvenli depoya eklenmeden önce `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt` içinde geçici olarak depolanır. Bağlantı başarıyla eklendiğinde, sertifika bu konumdan silinir.
 - i) Özet sayfasını görüntülemek için **İleri** düğmesini tıklayın. Önceki sayfaları yeniden ziyaret etmek ve düzeltme yapmak için **Geri** düğmesini kullanabilirsiniz. Bilgilerden memnunsanız, uzak kuyruk yöneticisine bağlanmak için **Bağlan** düğmesini tıklayın.

Belirlediğiniz bağlantı bilgileri, Web dizininizdeki CCDT dosyasına yazılır. Yol: `WLP_USER_DIR/generated.ccdt/ccdt-uniqueName`.

Sonuçlar

Uzak kuyruk yöneticisi, IBM MQ Console içindeki uzak kuyruk yöneticisi listesinde görünür. Bağlantının başarılı olması koşuluyla, uzak kuyruk yöneticisinin nesnelere, yerel bir kuyruk yöneticisinin nesnelereyle çalıştığı şekilde denetleyebilirsiniz.

V 9.3.0 LTS Komut satırını kullanarak IBM MQ Console ' e uzak kuyruk yöneticisi eklenmesi-Yalnızca LTS

Komut satırındaki `setmqweb remote` komutunu kullanarak IBM MQ Console ' e bir uzak kuyruk yöneticisi ekleyebilirsiniz. Uzak kuyruk yöneticisi, IBM MQ Console ile aynı sistemde farklı bir kurulumda çalışan bir kuyruk yöneticisi ya da farklı bir sistemde çalışan bir kuyruk yöneticisi olabilir.

Başlamadan önce

Not: Bu görevdeki adımlar MQSC komutlarını çalıştırmanızı gerektirir:

- **ALW** AIX, Linux, and Windows üzerinde, bir `runmqsc` komut isteminden MQSC komutları yayınlayın. Bkz. `runmqsc` altında MQSC komutlarının etkileşimli olarak çalıştırılması ve `runmqsc` altındaki metin dosyalarından MQSC komutlarının çalıştırılması. Bu görev için, QM1: komutunu kullanan bir `runmqsc` komut istemi açın.

```
runmqsc QM1
```

- **IBM i** IBM üzerinde, Komut Dosyası dosyasında bir komut listesi oluşturun ve dosyayı `STRMQMMQSC` komutunu kullanarak çalıştırın. Bkz. [Administration using MQSC commands on IBM i](#).

- **z/OS** z/OS üzerinde, komuta bağlı olarak MQSC komutları çeşitli kaynaklardan verilebilir. Bkz. [IBM MQ for z/OS üzerinde MQSC ve PCF komutlarını yayınlatabileceğiniz kaynaklar](#).

mqwweb sunucusunun IBM MQ Console ile uzak kuyruk yöneticisi bağlantılarına izin verecek şekilde yapılandırıldığından emin olun. Daha fazla bilgi için [Uzak kuyruk yöneticisi bağlantıları davranışını yapılandırma](#) başlıklı konuya bakın.

Yordam

1. Uzak kuyruk yöneticisinde, **DEFINE CHANNEL** MQSC komutunu kullanarak kuyruk yöneticisinin uzaktan yönetilmesini sağlamak için bir sunucu bağlantısı kanalı yaratın. Örneğin, QM1 kuyruk yöneticisi için bir sunucu bağlantısı kanalı QM1 . SVRCONN yaratmak üzere şu MQSC komutunu girin:

```
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
```

DEFINE CHANNEL ve kullanılabilir seçenekler hakkında daha fazla bilgi için bkz. [DEFINE CHANNEL](#).

2. Uygun bir kullanıcının kuyruk yöneticisini ve kuyruk yöneticisiyle ilişkilendirilmiş MQ nesnelere denetim yetkisine sahip olduğunu doğrulayın.

- **ALW** AIX, Linux, and Windows üzerinde standart bir komut satırında **setmqaut** denetim komutunu kullanın.
- **z/OS** z/OS üzerinde, yetkili kullanıcıya kuyruk yöneticisine erişim vermek için RACF profillerini tanımlayın.

Örneğin, AIX, Linux, and Windows sistemlerinde `exampleUser` kullanıcıya QM1 kuyruk yöneticisine erişim yetkisi vermek için şu denetim komutunu girin:

```
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp
```

Bu yetkili kullanıcı, aşağıdaki kullanıcılardan biri olabilir:

- Bu kuyruk yöneticisini uzaktan denetlemek istediğiniz sistemde IBM MQ Console ' u çalıştıran `mqweb` sunucusunu başlatan kullanıcı kimliğiyle aynı kullanıcı kimliği.
- “7” sayfa 128. adımda **setmqweb remote** komutunda yer alan bir kullanıcı kimliği ve parolayla eşleşen bir kullanıcı kimliği. **setmqweb remote** komutuna kullanıcı kimliği ve parola eklenerek, IBM MQ Console kuyruk yöneticisine bağlandığında kimlik doğrulaması için bu kullanıcı kimliği ve parola kullanılır.
- Kanal güvenlik kurallarına göre belirlenen bir kullanıcı kimliği. Örneğin, sunucu bağlantısı kanalında, uzaktan yönetim için IBM MQ Console ' u kullandığınız IP adresinden bağlantılara izin verecek bir kanal doğrulama kuralı ayarlayabilir ve tüm bu bağlantıları, kuyruk yöneticisini kullanma yetkisi olan belirli bir kullanıcı kimliğiyle eşleyebilirsiniz. Daha fazla bilgi için bkz. [Kanallar için yeni CHLAUTH kuralları oluşturma](#).

3. **ALW**

Uzak kuyruk yöneticisinde çalışan bir dinleyici yoksa, **DEFINE LISTENER** MQSC komutunu kullanarak gelen ağ bağlantılarını kabul edecek bir dinleyici yaratın.

Örneğin, uzak kuyruk yöneticisi QM1 için 1414 portunda bir dinleyici REMOTE . LISTENER yaratmak üzere şu MQSC komutunu girin:

```
runmqsc QM1
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
end
```

4. **START LISTENER** MQSC komutunu kullanarak dinleyicinin çalıştığını doğrulayın.

ALW Örneğin, QM1kuyruk yöneticisi için REMOTE . LISTENER dinleyicisini başlatmak üzere AIX, Linux, and Windows üzerinde şu MQSC komutunu girin:

```
runmqsc QM1
START LISTENER(REMOTE . LISTENER)
end
```

z/OS Örneğin, z/OSüzerinde dinleyiciyi başlatmak için şu MQSC komutunu girin:

```
/cpf START LISTENER TRPTYPE(TCP) PORT(1414)
```

z/OSüzerinde bir dinleyici başlatmadan önce kanal başlatıcı adres alanının başlatılması gerektiğini unutmayın.

5. Uzak kuyruk yöneticisi bağlantı bilgilerini içeren bir JSON CCDT dosyası yaratın:

- Uzaktan bağlanmak istediğiniz kuyruk yöneticisiyle aynı kuruluşla ilişkilendirilmiş IBM MQ Console dosyasını kullanarak bir CCDT dosyası oluşturun.

Ana Sayfa panosundan **Bağlantı dosyasını karşıdan yükle** döşemesini tıklatın.

- Bağlantıyı tanımlayan bir JSON biçimi CCDT dosyası yaratın. JSON biçimi CCDT oluşturma hakkında daha fazla bilgi için bkz. [JSON biçimi CCDT ' nin yapılandırılması](#).

CCDT dosyası name, clientConnectionve type bilgilerini içermelidir. İsteğe bağlı olarak transmissionSecurity bilgileri gibi ek bilgiler ekleyebilirsiniz. Tüm CCDT kanal özniteliği tanımlamalarına ilişkin ek bilgi için [CCDT kanal özniteliği tanımlamalarının tam listesibaşlıklı konuya](#) bakın.

Aşağıdaki örnek, uzak kuyruk yöneticisi bağlantısı için temel bir JSON CCDT dosyasını göstermektedir. Kanalın adını, ["1" sayfa 127](#). adımda oluşturulan örnek sunucu-bağlantı kanalı ile aynı ada ve bağlantı kapısını, dinleyici tarafından kullanılan kapıyla aynı değere ayarlar. Bağlantı anasistemi, örnek uzak kuyruk yöneticisinin (QM1) çalıştığı sistemin anasistem adına ayarlanır:

```
{
  "channel": [{
    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [{
        "host": "example.com",
        "port": 1414
      }],
      "queueManager": "QM1"
    },
    "type": "clientConnection"
  }]
}
```

6. JSON CCDT dosyasını IBM MQ Console ' un çalıştığı sisteme kopyalayın.

7. IBM MQ Consoleçalıştıran kuruluşta, uzak kuyruk yöneticisi bilgilerini IBM MQ Console yapılandırmasına eklemek için **setmqweb remote** komutunu kullanın.

En az IBM MQ Console ' e uzak kuyruk yöneticisi eklemek için kuyruk yöneticisi adını, kuyruk yöneticisi için benzersiz bir ad (aynı kuyruk yöneticisi adına sahip olabilecek diğer uzak kuyruk yöneticilerini ayırt etmek için) ve kuyruk yöneticisine ilişkin CCDT URL belirtmeniz gerekir. Uzak kuyruk yöneticisi bağlantısı için kullanılacak kullanıcı adı ve parola ya da güvenli depo ve anahtar deposu ayrıntıları gibi belirtebileceğiniz birkaç ek seçenek vardır. **setmqweb remote** denetim komutuyla belirtilebilecek parametrelerin tam listesi için [setmqweb remotekonusuna](#) bakın.

Örneğin, örnek uzak kuyruk yöneticisi QM1' i örnek CCDT dosyasını kullanarak eklemek için aşağıdaki denetim komutunu girin:

```
setmqweb remote add -uniqueName "MACHINEAQM1" -qmgrName "QM1" -ccdtURL "c:\myccdt\ccdt.json"
```


Sonuçlar

Uzak kuyruk yöneticisi, uzak bağlantı listesi yenilendiğinde IBM MQ Console içindeki uzak kuyruk yöneticisi listesinde görünür. Bağlantının başarılı olması koşuluyla, uzak kuyruk yöneticisinin nesnelere, yerel bir kuyruk yöneticisinin nesnelere çalıştığı şekilde denetleyebilirsiniz.

Örnek

Aşağıdaki örnek, QM1kuyruk yöneticisi için uzak kuyruk yöneticisi bağlantısını ayarlar. IBM MQ Console , kullanıcıya verilen yetkiye dayalı olarak kuyruk yöneticisini denetleme yetkisine sahiptir exampleUser. Uzak kuyruk yöneticisi bağlantı bilgilerini yapılandırmak için **setmqweb remote** komutu kullanıldığında bu kullanıcının kimlik bilgileri IBM MQ Console ' e sağlanır.

1. QM1 uzak kuyruk yöneticisinin bulunduğu sistemde, bir sunucu bağlantısı kanalı ve bir dinleyici yaratılır. Dinleyici başlatılır ve exampleUser kullanıcılarına kuyruk yöneticisini denetleme yetkisi verilir. Örneğin, AIX, Linux, and Windowsüzerinde aşağıdaki komutları çalıştırın:

```
runmqsc QM1
#Define the server connection channel that will accept connections from the Console
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
# Define the listener to use for the connection from the Console
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
# Start the listener
START LISTENER(REMOTE.LISTENER)
end

#Set mq authorization for exampleUser to access the queue manager
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp
```

2. IBM MQ Console ' in çalıştığı sistemde, aşağıdaki bağlantı bilgileriyle bir QM1_ccdt . json dosyası yaratılır:

```
{
  "channel": [{
    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [{
        "host": "example.com",
        "port": 1414
      }],
      "queueManager": "QM1"
    },
    "type": "clientConnection"
  }]
}
```

3. IBM MQ Console ' in çalıştığı sistemde, QM1 kuyruk yöneticisine ilişkin uzak kuyruk yöneticisi bağlantı bilgileri mqweb sunucusuna eklenir. exampleUser kimlik bilgileri bağlantı bilgilerinde bulunur:

```
setmqweb remote add -uniqueName "MACHINEAQM1" -qmgrName "QM1" -ccdtURL
"c:\myccdt\QM1_ccdt.json" -username "exampleUser" -password "password"
```

4. IBM MQ Console , QM1uzak kuyruk yöneticisini gösterir.

LTS IBM MQ Console: Nesnelere çalışma-Yalnızca LTS

Her IBM MQ kuyruk yöneticisiyle ilişkilendirilmiş birkaç farklı nesne tipi vardır.

Bu görev hakkında

Aşağıdaki IBM MQ nesnesi tipleriyle çalışmak için konsolu kullanabilirsiniz:

- Kuyruklar
- Konular
- Abonelikler
- İletişim nesnelere:
 - Dinleyiciler

- Kuyruk yöneticisi kanalları
- Uygulama kanalları

Yordam

Bir IBM MQ nesnesiyle çalışmak için:


1. Kuyruk yöneticisi liste görünümünde, çalışmak istediğiniz nesnelere sahip olan kuyruk yöneticisini tıklatın.
2. Çalışmak istediğiniz nesne tipini seçmek için Kuyruklar, Konular, Abonelikler ya da İletişim etiketini tıklatın.
3. Bu nesne tipiyle çalışmaya ilişkin ayrıntılı yönergeler için aşağıdaki konulardan birine bakın.

IBM MQ Console: Kuyruklarla çalışılması-Yalnızca LTS

Belirli bir kuyruk yöneticisi için var olan kuyrukları **Kuyruklar** sekmesinde görüntüleyebilirsiniz. Kuyruk ekleyebilir ve silebilir, kuyruktaki iletileri ekleyebilir ve temizleyebilir, iletilere göz atabilir, bir kuyruğun özelliklerini görüntüleyebilir ve ayarlayabilir ve bir kuyruğun yetki kayıtlarını yönetebilirsiniz.

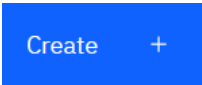
Bu görev hakkında

Kuyruklar görünümü, belirli bir kuyruk yöneticisine ilişkin kuyrukları listeler. Kuyruk listesine bir kuyruk yöneticisini tıklayıp **Kuyruklar** etiketini seçerek erişebilirsiniz. Çalışmak istediğiniz kuyrukları listeden tek tek seçebilirsiniz.

 z/OS üzerindeki kuyruklara ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

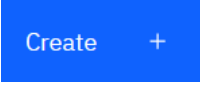
Yordam

- Kuyruk eklemek için:

- a) **Kuyruklar** sekmesinde, oluştur düğmesini .
- b) Yaratmak istediğiniz kuyruk tipini seçin:
 - Yerel kuyruk-iletileri ait olduğu kuyruk yöneticisi içinde saklar.
 - Diğer ad kuyruğu-aynı kuyruk yöneticisindeki başka bir kuyruğa işaretçi.
 - Uzak kuyruk-başka bir kuyruk yöneticisindeki başka bir kuyruğa işaretçi.
 - Model kuyruğu-dinamik bir kuyruk yöneticisi yaratıldığında kullanılan bir kuyruk için şablon.
- c) Yaratmakta olduğunuz kuyruk tipi için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
- d) **Oluştur**'u tıklatın. Yeni kuyruk yaratılır.

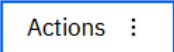


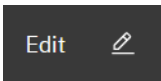

- İletileri kuyruğa koymak için:

- a) Kuyruk listesi görünümündeki listede ileti eklemek istediğiniz kuyruğu tıklatın. Model kuyruğu seçemezsiniz.

- b) Oluştur düğmesini tıklatın .
- c) Kuyruğa koymak istediğiniz iletiyi girin.
- d) **Oluştur**'u tıklatın.

- Kuyruktaki iletileri temizlemek için:

- a) Kuyruk listesinden iletileri temizlemek istediğiniz yerel kuyruğu tıklatın.

- b) Eylemler düğmesini  tıklattın ve **İletileri temizle** seçeneğini belirleyin.
- c) **İletileri temizle** düğmesini tıklatarak kuyruğu temizlemek istediğinizi doğrulayın.
- Kuyruktaki iletilere göz atmak için, kuyruk listesi görünümünde kuyruğu tıklattın. Bu kuyruktaki iletilerin bir listesi görüntülenir.
- Bir kuyruğu silmek için:
 - a) Silmek istediğiniz kuyruğun yanındaki anahtar simgesini  tıklattın.
 - b) Kuyruk düzenle görünümünde **Kuyruğu sil** 'i tıklattın.
 - c) Kuyruğu silmek istediğinizi doğrulamak için **Sil** düğmesini tıklattın. Kuyruk silindi.
- Bir kuyruğun özelliklerini görüntülemek ve düzenlemek için:
 - a) Düzenlemek istediğiniz kuyruğun yanındaki anahtar simgesini  tıklattın.
 - b) Düzenle düğmesini tıklattın .
 - c) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Özellikler hakkında bilgi için bkz. MQ Explorer belgelerinde [Kuyruk özellikleri](#)
 - d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklattın.
- Bir kuyruğa ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Yetki kayıtlarını düzenlemek istediğiniz kuyruğun yanındaki anahtar simgesini  tıklattın.
 - b) **Security** (Güvenlik) sekmesini tıklattın.
 - c) Kuyruk yöneticisi yetki kayıtları için açıklanan yetki kayıtlarıyla çalışın. Bkz. [“IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışılması-Yalnızca LTS” sayfa 123.](#)

IBM MQ Console: Başlıklarla çalışma-Yalnızca LTS

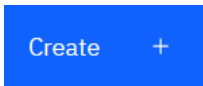
Konu eklemek ve silmek ve bir konunun özelliklerini görüntülemek ve ayarlamak için IBM MQ Console özelliğini kullanabilirsiniz.


Bu görev hakkında


Konular görünümü, belirli bir kuyruk yöneticisi için var olan konuları listeler. Konular listesine bir kuyruk yöneticisini tıklattıp **Konular** sekmesini seçerek erişebilirsiniz. Üzerinde çalışılacak konuları listeden tek tek seçebilirsiniz.

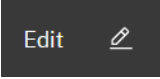
 z/OS ile ilgili bir konuya ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

Yordam

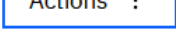
- Konu eklemek için:
 - a) **Konular** sekmesinde, oluştur düğmesini .
 - b) Oluşturmakta olduğunuz konu için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
 - c) **Oluştur**'u tıklattın. Yeni konu oluşturulur.
- Bir konuyu silmek için:

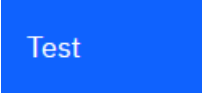
- a) Silmek istediğiniz konunun yanındaki anahtar simgesini  tıklatın.
- b) Kuyruğu düzenle görünümünde **Konuyu sil** i tıklatın.
- c) Konuyu silmek istediğinizi doğrulamak için **Sildüğmesini** tıklatın. Konu silindi.
- Bir konunun özelliklerini görüntülemek ve düzenlemek için:

- a) Düzenlemek istediğiniz konunun yanındaki anahtar simgesini  tıklatın.

- b) Düzenle düğmesini tıklatın 
- c) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Özelliklerle ilgili bilgi için MQ Explorer belgelerinde [Konu özellikleri](#) konusuna bakın.
- d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.
- Bir konuyla ilgili ileti yayınlamak için en az bir eşleşen aboneliğiniz olmalıdır. Gerekirse bir test aboneliği yaratabilirsiniz:

- a) Konu listesinde yayınlamak istediğiniz konuyu tıklatın.
- b) Sonraki işlemeniz, bu konunun eşleşen abonelikleri olup olmadığını bağlıdır:
Eşleşen abonelik yoksa:

- a. İşlemler düğmesini  tıklatın ve **Test konusu**' nu seçin.

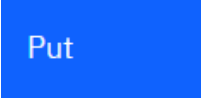
- b. Test düğmesini  gösterir. Test aboneliğine bir test iletisi yazılır.

Konu için eşleşen abonelikler varsa:


- a. Bir abonelik adını tıklatın.

- b. Oluştur düğmesini  gösterir.

- c. Yayınlamak istediğiniz iletiyi girin.

- d. Yerleştir düğmesini  gösterir. İleti, eşleşen tüm aboneliklere yazılır.

- Bir konuya abone olmak için bkz. [“IBM MQ Console: Aboneliklerle çalışma-Yalnızca LTS”](#) sayfa 132:
- Bir konuya ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:

- a) Yetki kayıtlarını düzenlemek istediğiniz konunun yanındaki anahtar simgesini  tıklatın.
- b) **Security** (Güvenlik) sekmesini tıklatın.
- c) Kuyruk yöneticisi yetki kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın, bkz. [“IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışılması-Yalnızca LTS”](#) sayfa 123.


IBM MQ Console: Aboneliklerle çalışma-Yalnızca LTS

Abonelik eklemek ve silmek için IBM MQ Console olanağını kullanabilir ve bir aboneliğin özelliklerini görüntüleyebilir ve ayarlayabilirsiniz.

Bu görev hakkında

Abonelikler görünümü, belirli bir kuyruk yöneticisi için var olan abonelikleri listeler. Bir kuyruk yöneticisini tıklatıp **Abonelikler** sekmesini seçerek abonelikler listesine erişebilirsiniz. Çalışmak için listeden tek tek abonelikler seçebilirsiniz.

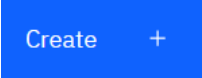
Abonelikler hakkında daha fazla bilgi için bkz. [Aboneler ve abonelikler](#) ve [DEFINE SUB](#).


 z/OSaboneliğine ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.


Yordam


- Abonelik eklemek için:

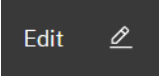
Create +

- Abonelikler** sekmesinde, oluştur düğmesini  .
 - Yönetilen ya da yönetilmeyen bir abonelik oluşturmak isteyip istemediğinizi seçin.
 - Oluşturmakta olduğunuz abonelik için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
 - Oluştur**'u tıklatın. Yeni abonelik oluşturulur.
- Bir aboneliği silmek için:

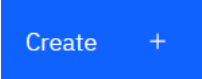
- Silmek istediğiniz aboneliğin yanındaki anahtar simgesini  tıklatın.
 - Kuyruğu düzenle görünümünde **Aboneliğin silinmesi** ögesini tıklatın.
 - Aboneliği silmek istediğinizi doğrulamak için **Sil** düğmesini tıklatın. Abonelik silindi.
- Bir aboneliğin özelliklerini görüntülemek ve düzenlemek için:

- Düzenlemek istediğiniz aboneliğin yanındaki anahtar simgesini  tıklatın.


Edit 

- Düzenle düğmesini tıklatın .
 - Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir.
 - Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.
- Aboneliğin abone olduğu konuyla ilgili bir ileti yayınlamak için:
 - Konusunu abonelik listesinde yayınlamak istediğiniz aboneliği tıklatın.

Create +

- Oluştur düğmesini  gösterir.
- Yayınlamak istediğiniz iletiyi girin.

Put


- Yerleştir düğmesini  gösterir. İleti, yayınladığınız konuyla eşleşen tüm aboneliklere yazılır.

IBM MQ Console: Kuyruk yöneticisi kanallarıyla çalışma-Yalnızca LTS

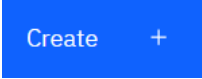




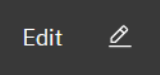

IBM MQ Console kuyruk yöneticisi kanallarıyla çalışabilirsiniz: Kuyruk yöneticisi kanalları ekleyebilir ve bunları silebilir, kanalları başlatabilir ve durdurabilir, kanalları sıfırlayabilir ve çözebilir ve kanalları ve ping kanalları ekleyebilirsiniz. Ayrıca, bir kuyruk yöneticisi kanalının özelliklerini görüntüleyebilir ve ayarlayabilir ve kanala ilişkin yetki kayıtlarını yönetebilirsiniz.





Bu görev hakkında

Kuyruk yöneticisi kanalı, bir ağ üzerinden kuyruk yöneticileri arasında ileti iletmek için kullanılan mantıksal bir iletişim bağlantısıdır. Kuyruk yöneticisi kanal görünümü, kaç kanalın çalıştığını, kaç tanesinin yeniden denendiğini ve kaç tanesinin durdurulduğunu hızlı bir şekilde gösteren bir pano içerir.

 z/OS üzerindeki bir kanala ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

Yordam

- Bir kuyruk yöneticisi kanalı eklemek için:
 - a) **İletişim** sekmesinde, kuyruk yöneticisi kanalları görünümünün görüntülediğinden emin olun ve oluştur düğmesini  tıklatın.
 - b) Yaratmak istediğiniz kuyruk yöneticisi kanalı tipini seçin ve sonraki düğmesini tıklatın .
 - c) Oluşturmakta olduğunuz kanal için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
 - d) **Oluştur**'u tıklatın. Yeni kanal, **etkin değil** durumuyla oluşturulur.
- Bir kuyruk yöneticisi kanalı başlatmak için:
 - a) Listede başlatmak istediğiniz kanalı bulun.
 - b) Menüden **Başlat** seçeneğini belirleyin .
- Bir kuyruk yöneticisi kanalını durdurmak için:
 - a) Listede durdurmak istediğiniz kanalı bulun.
 - b) Menüden **Durdur** seçeneğini belirleyin .
- Bir kuyruk yöneticisi kanalının özelliklerini görüntülemek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Özellikler** sekmesinin seçildiğinden emin olun. Özellikleri düzenlemek için Düzenle düğmesini .
 - d) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Özelliklerle ilgili daha fazla bilgi için MQ Explorer belgelerinde [Kanal özellikleri](#) konusuna bakın.
 - e) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.
- Bir kuyruk yöneticisi kanalını ilk durumuna getirmek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Gelişmiş** seçeneğini belirleyin .
 - c) **İlk Durumuna Getir** bölümünde bir ileti sıra numarası belirtin. İki uç, gönderilecek sonraki iletinin sıra numarası konusunda aynı fikirde olmadığı için, bir kanalı başlalmazsa ilk durumuna getirmeniz gerekir. İleti sıra numarası, bu sayıyı belirtir.


- d) **Reset Channel**(Kanalı Sıfırla) seçeneğini tıklatın.
- Bir gönderen ya da sunucu kanalını çözmek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Gelişmiş** seçeneğini belirleyin .
 - c) **Çözüm** bölümünde, **İletileri iletim kuyruğuna geri yükle** ya da **İletileri at** seçeneğini tıklatarak yürürlükteki ileti kümesini kesinleştirmeyi ya da geri yüklemeyi seçin.
- Bir kuyruk yöneticisi kanalına ping komutu göndermek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Ping** (Ping) seçeneğini belirleyin .
- Bir kuyruk yöneticisi kanalına ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Security** (Güvenlik) sekmesini tıklatın.
 - d) Kuyruk yöneticisi yetki kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın, bkz. [“IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışılması-Yalnızca LTS” sayfa 123.](#)
- Bir kuyruk yöneticisi kanalını silmek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Delete channel**(Kanalı sil) seçeneğini tıklatın.

LTS **IBM MQ Console: Uygulama kanallarıyla çalışma-Yalnızca LTS**

Uygulama kanallarıyla çalışmak için IBM MQ Console 'i kullanabilirsiniz: kanallar ekleyebilir ve var olanları silebilir, kanalları başlatabilir ve durdurabilir, kanalları sıfırlayabilir ve çözebilir ve kanallara ping kanalları gönderebilirsiniz. Ayrıca, bir uygulama kanalının özelliklerini görüntüleyebilir ve ayarlayabilir ve kanala ilişkin yetki kayıtlarını yönetebilirsiniz.


Bu görev hakkında


Uygulama kanalı, uygulamalar tarafından bir ağ üzerinden bir kuyruk yöneticisine bağlanmak için kullanılan mantıksal bir iletişim bağlantısıdır. Uygulama kanalı görünümü, kaç kanal çalıştığından, kaç yeniden denemenin gerçekleştirildiğinden ve kaç tanesinin durdurulduğundan hızlı bir görünüm gösteren bir pano içerir.









 z/OS üzerindeki bir kanala ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.


Yordam

- Bir uygulama kanalı eklemek için:
 - a) **İletişim** sekmesinde uygulama kanalları görünümünün görüntülediğinden emin olun ve oluştur

düğmesini .

- b) Sonraki düğmesini tıklatın .

- c) Oluşturmakta olduğunuz kanal için gerekli bilgileri sağlayın. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
- d) **Oluştur**'u tıklatın. Yeni kanal, **etkin değil** durumuyla oluşturulur.
- Bir uygulama kanalını başlatmak için:
 - a) Listede başlatmak istediğiniz kanalı bulun.
 - b) Menüden **Başlat** seçeneğini belirleyin .
 - Bir uygulama kanalını durdurmak için:
 - a) Listede durdurmak istediğiniz kanalı bulun.
 - b) Menüden **Durdur** seçeneğini belirleyin .
 - Bir uygulama kanalının özelliklerini görüntülemek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Özellikler** sekmesinin seçildiğinden emin olun. Özellikleri düzenlemek için Düzenle düğmesini .
 - d) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Özelliklerle ilgili daha fazla bilgi için MQ Explorer belgelerinde [Kanal özellikleri](#) konusuna bakın.
 - e) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.
 - Bir uygulama kanalını ilk durumuna getirmek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Gelişmiş** seçeneğini belirleyin .
 - c) **İlk Durumuna Getir** bölümünde bir ileti sıra numarası belirtin. İki uç, gönderilecek sonraki iletinin sıra numarası konusunda aynı fikirde olmadığı için, bir kanalı başlarsa ilk durumuna getirmeniz gerekir. İleti sıra numarası, bu sayıyı belirtir.
 - d) **Reset Channel**(Kanalı Sıfırla) seçeneğini tıklatın.
 - Bir gönderen ya da sunucu kanalını çözmek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Gelişmiş** seçeneğini belirleyin .
 - c) **Çözüm** bölümünde, **İletileri iletim kuyruğuna geri yükle** ya da **İletileri at** seçeneğini tıklatarak yürürlükteki ileti kümesini kesinleştirmeyi ya da geri yüklemeyi seçin.
 - Bir kanala ping komutu göndermek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Ping** (Ping) seçeneğini belirleyin .
 - Bir uygulama kanalına ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .

- c) **Security** (Güvenlik) sekmesini tıklatın.
- d) Kuyruk yöneticisi yetki kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın, bkz. [“IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışılması-Yalnızca LTS” sayfa 123.](#)
- Bir uygulama kanalını silmek için:
 - a) Listedeki kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Delete channel**(Kanalı sil) seçeneğini tıklatın.

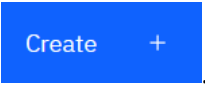




LTS **IBM MQ Console: Dinleyicilerle çalışma-Yalnızca LTS**



Dinleyici eklemek ve silmek, dinleyici başlatmak ve durdurmak, dinleyici özelliklerini görüntülemek ve ayarlamak ve bir dinleyiciye ilişkin yetki kayıtlarını yönetmek için IBM MQ Console özelliğini kullanabilirsiniz.

Bu görev hakkında

Dinleyiciler görünümü, belirli bir kuyruk yöneticisi için var olan dinleyicileri görüntüler. Çalışmak için tek tek dinleyicileri seçebilirsiniz.

Yordam

- Dinleyici yaratmak için:
 - a) **İletişim** sekmesinde, dinleyiciler görünümünün görüntülediğinden emin olun ve oluştur düğmesini .
 - b) Yaratmakta olduğunuz dinleyici için gereken bilgileri belirtin. Varsayılan olarak, değer belirtmeniz için önerilen en düşük özellikler görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
 - c) **Oluştur**'u tıklatın. Yeni dinleyici yaratılır.
- Dinleyici başlatmak için:
 - a) Listede başlatmak istediğiniz dinleyiciyi bulun.
 - b) Menüden **Başlat** seçeneğini belirleyin .
- Bir dinleyiciyi durdurmak için:
 - a) Listede başlatmak istediğiniz dinleyiciyi bulun.
 - b) Menüden **Durdur** seçeneğini belirleyin .
- Bir dinleyicinin özelliklerini görüntülemek ve düzenlemek için:
 - a) Listedeki dinleyiciyi bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Özellikler** sekmesinin seçildiğinden emin olun. Özellikleri düzenlemek için Düzenle düğmesini .
 - d) Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Özelliklerle ilgili daha fazla bilgi için IBM MQ Explorer belgelerinde [Dinleyici özellikleri](#) konusuna bakın.
 - e) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.

- Bir dinleyiciye ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Listedeki dinleyiciyi bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Security** (Güvenlik) sekmesini tıklatın.
 - d) Kuyruk yöneticisi yetki kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın, bkz. [“IBM MQ Console: Kuyruk yöneticisi yetki kayıtlarıyla çalışılması-Yalnızca LTS” sayfa 123.](#)
- Bir dinleyiciyi silmek için:
 - a) Listedeki dinleyiciyi bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Dinleyiciyi sil** i tıklatın.

IBM MQ Console ayarlar

IBM MQ Console için bazı genel ayarları belirtebilirsiniz.

IBM MQ Console ayarları görünümüne geçmek için ayarlar simgesini  tıklatın.

Aşağıdaki özellikleri denetlemek için ayarları kullanın:

- Kuyruk yöneticilerinin her on saniyede bir otomatik yenilenmesi. Bu özellik açılabilir ya da kapatılabilir.
- Sistem nesnelerinin görüntülenip görüntülenmeyeceğini belirler. Bunu tüm nesne tipleri için belirtebilir ya da nesne tiplerini tek tek seçebilirsiniz.
- İzleme bilgilerinin toplanıp toplanmadığını belirler.

Linux

Windows

IBM MQ Explorer kullanarak yönetim

IBM MQ Explorer , ağınızın yerel ya da uzaktan yönetimini Windows ya da Linux x86-64 çalıştıran bir bilgisayardan gerçekleştirmenizi sağlar.

IBM MQ for Windows ve IBM MQ for Linux x86-64 , denetim ya da MQSC komutlarını kullanmaya alternatif olarak yönetim görevlerini gerçekleştirmek için IBM MQ Explorer adlı bir yönetim arabirimi sağlar. [Komut kümelerinin karşılaştırılması](#) , IBM MQ Explorer kullanarak neler yapabileceğinizi gösterir.



IBM MQ Explorer , Windows'ta en ta yine 'lerde üzerinden ya da Linux x86-64' te çalışan bir bilgisayardan, IBM MQ Explorer ' u ilgilendığınız kuyruk yöneticilerine ve kümelere yönlendirerek ağınızın yerel ya da uzaktan yönetimini gerçekleştirmenizi sağlar. Desteklenen herhangi bir platformda çalışan kuyruk yöneticilerine uzaktan bağlanarak z/OS dahil tüm ileti sistemi omurganızın konsoldan görüntülenmesine, keşfedilmesine ve değiştirilmesine olanak sağlar.

Uzak IBM MQ kuyruk yöneticilerini IBM MQ Explorer yönetebilecek şekilde yapılandırmak için bkz. [“IBM MQ Explorer için önkoşul olan yazılım ve tanımlar” sayfa 140.](#)

Bu olanak, genellikle IBM MQ için çalışma ortamını yerel olarak ya da bir Windows ya da Linux x86-64 sistem etki alanında uzaktan ayarlayarak ve ince ayarlayarak ilişkili görevleri gerçekleştirmenizi sağlar.

Linux işletim sistemlerinde, birden çok Eclipse kurulumunuz varsa, IBM MQ Explorer başlatılabilir. Bu durumda, IBM MQ Explorer ' yi diğer Eclipse kurulumu için kullandığınızdan farklı bir kullanıcı kimliği kullanarak başlatın.

Linux'ta IBM MQ Explorer ' ı başarıyla başlatmak için ana dizininize bir dosya yazabilmeniz ve ana dizinin var olması gerekir.

  IBM MQ Explorer , Fix Central adresinden erişilebilen bağımsız IBM MQ Explorer karşıdan yüklemelerinden kurulabilir. Daha fazla bilgi için bkz. [Installing and unInstalling IBM MQ Explorer \(Kurma ve kaldırma\) Linux ve Windows' de bağımsız bir uygulama olarak.](#)

IBM MQ Explorer , bir dizi İçerik Görünümleri ve Özellik iletişim kutusunu kullanarak yönetim görevlerini gerçekleştirmek için kullanılabilir. Bir ya da daha çok Eclipse eklentisi yazarak da IBM MQ Explorer eklentilerini genişletebilirsiniz.

IBM MQ Explorer görevler

IBM MQ Explorer ile aşağıdaki görevleri gerçekleştirebilirsiniz:

- Bir kuyruk yöneticisi yaratın ve silin (yalnızca yerel makinenizde).
- Kuyruk yöneticisini başlatın ve durdurun (yalnızca yerel makinenizde).
- Kuyruklar ve kanallar gibi IBM MQ nesne tanımlarını tanımlayın, görüntüleyin ve değiştirin .
- Kuyruktaki iletilere göz atın.
- Bir kanalı başlatın ve durdurun.
- Bir kanal, dinleyici, kuyruk ya da hizmet nesnelere ilişkin durum bilgilerinin görüntülenmesi .
- Bir kümedeki kuyruk yöneticilerini görüntüleyin.
- Hangi uygulamaların, kullanıcıların ya da kanalların belirli bir kuyruğunun açık olduğunudenetleyin.
- Yeni Küme Yarat sihirbazını kullanarak yeni bir kuyruk yöneticisi kümesi yaratın .
- Kümeye Kuyruk Yöneticisi Ekle sihirbazını kullanarak Kümeye bir kuyruk yöneticisi ekleyin .
- TLS (Transport Layer Security; İletim Katmanı Güvenliği) kanal güvenliği ile kullanılan kimlik doğrulama bilgileri nesnesini yönetin.
- Kanal başlatıcıları, tetikleyici izleyicileri ve dinleyicileri yaratın ve silin.
- komut sunucularını, kanal başlatıcıları, tetikleyici izleyicilerinive dinleyicileribaşlatın ya da durdurun.
- Belirli hizmetleri bir kuyruk yöneticisi başlatıldığında otomatik olarak başlatılarak ayarlayın.
- Kuyruk yöneticilerinin özelliklerini değiştirin.
- Yerel varsayılan kuyruk yöneticisini değiştirin.
- TLS sertifikalarını yönetmek, sertifikaları kuyruk yöneticileriyle ilişkilendirmek ve sertifika depolarını yapılandırmak ve ayarlamak için **strmqikm** (ikeyman) GUI ' yi çağırın (yalnızca yerel makinenizde).
- JMS nesnelerini IBM MQ nesnelerdenve IBM MQ nesnelerini JMS nesnelerdenyaratın.
- Şu anda desteklenen tiplerden herhangi biri için bir JMS Bağlantı Üreticisi yaratın.
- Bir dinleyicinin TCP kapı numarası ya da kanal başlatıcı kuyruk adı gibi herhangi bir hizmete ilişkin parametreleri değiştirin.
- Hizmet izlemesini başlatın ya da durdurun.

İçerik Görünümleri ve Özellik iletişim kutuları

Bir dizi İçerik Görünümleri ve Özellik iletişim kutusu kullanarak yönetim görevlerini gerçekleştirebilirsiniz.

İçerik Görünümü

İçerik Görünümü, aşağıdakileri görüntüleyebilen bir panodur:

- IBM MQ ile ilgili öznitelikler ve yönetim seçenekleri.
- Bir ya da daha çok ilgili nesneyle ilgili öznitelikler ve yönetim seçenekleri.
- Bir kümeyle ilişkin öznitelikler ve yönetim seçenekleri.

Özellik iletişim kutuları

Özellik iletişim kutusu, bazılarını düzenleyebileceğiniz bir alan dizisindeki bir nesneyle ilgili öznitelikleri görüntüleyen bir panodur.

IBM MQ Explorer içinde Navigator görünümünü kullanarak gezinebilirsiniz. Navigator , gerek duyduğunuz İçerik Görünümünü seçmenizi sağlar.

IBM MQ Explorer ürününün genişletilmesi

IBM MQ Explorer , bilgileri Eclipse çerçevesiyle ve Eclipse tarafından desteklenen diğer eklenti uygulamalarıyla tutarlı bir biçimde sunar.

Sistem yöneticileri, IBM MQ Explorer uygulamasını genişleterek, IBM MQ Explorer 'ı IBM MQ' yi yönetme biçimlerini geliştirmek üzere özelleştirme yeteneğine sahiptir.

Daha fazla bilgi için bakınız: [Extending MQ Explorer](#).

Linux Windows IBM MQ Explorer kullanılıp kullanılmayacağına karar verme

Kuruluşunuzda IBM MQ Explorer kullanılıp kullanılmayacağına karar verirken, bu konuda listelenen bilgileri göz önünde bulundurun.

Aşağıdaki noktaları bilmeniz gerekir:

nesne adları

IBM MQ Explorer ile birlikte kuyruk yöneticileri ve diğer nesnelere için küçük harfli adlar kullanırsanız, MQSC komutlarını kullanarak nesnelere çalışırken nesne adlarını tek tırnak içine almanız gerekir ya da IBM MQ bunları tanımaz.

Büyük kuyruk yöneticileri

IBM MQ Explorer , küçük kuyruk yöneticileriyle en iyi şekilde çalışır. Tek bir kuyruk yöneticisinde çok sayıda nesne varsa, IBM MQ Explorer bir görünümde sunmak için gerekli bilgileri çıkarırken gecikmelerle karşılaşabilirsiniz.

Kümeler

IBM MQ kümeleri yüzlerce ya da binlerce kuyruk yöneticisi içerebilir. IBM MQ Explorer , bir ağaç yapısını kullanarak bir kümedeki kuyruk yöneticilerini sunar. IBM MQ Explorer , siz seçinceye kadar kümedeki kuyruk yöneticilerine bağlanmadığından, bir kümenin fiziksel boyutu IBM MQ Explorer ' un hızını önemli ölçüde etkilemez.

IBM MQ Explorer ' in ayarlanması

Bu bölümde, IBM MQ Explorer ürününü kurmak için yapmanız gereken adımlar açıklanmaktadır.

- [“IBM MQ Explorer için önkoşul olan yazılım ve tanımlar” sayfa 140](#)
- [“IBM MQ Explorer için güvenlik” sayfa 141](#)
- [“IBM MQ Explorer içinde kuyruk yöneticilerini ve kümelerini gösterme ve gizleme” sayfa 144](#)
- [“Küme üyeliği ve IBM MQ Explorer” sayfa 145](#)
- [“IBM MQ Explorer için veri dönüştürme” sayfa 145](#)

IBM MQ Explorer için önkoşul olan yazılım ve tanımlar

IBM MQ Explorer ürününü kullanmayı denemeden önce aşağıdaki gereksinimleri karşıladığınızdan emin olun.

IBM MQ Explorer uzak kuyruk yöneticilerine yalnızca TCP/IP iletişim protokolünü kullanarak bağlanabilir.

Aşağıdakileri denetleyin:

1. Uzaktan denetlenen her kuyruk yöneticisinde bir komut sunucusu çalışıyor.
2. Her uzak kuyruk yöneticisinde uygun bir TCP/IP dinleyici nesnesi çalışıyor olmalıdır. Bu nesne IBM MQ dinleyicisi ya da AIX and Linux sistemlerinde inetd cini olabilir.
3. Varsayılan olarak SYSTEM.ADMIN.SVRCONN, tüm uzak kuyruk yöneticilerine ilişkin.

Kanalı aşağıdaki MQSC komutunu kullanarak yaratabilirsiniz:

```
DEFINE CHANNEL(SYSTEM.ADMIN.SVRCONN) CHLTYPE(SVRCONN)
```

Bu komut temel bir kanal tanımı yaratır. Daha kapsamlı bir tanım istiyorsanız (örneğin, güvenliği ayarlamak için), ek parametrelere gereksinim duyarsınız. Daha fazla bilgi için bkz. [DEFINE CHANNEL](#).

4. Sistem kuyruğu, SYSTEM.MQEXPLORER.REPLY.MODEL var olmalıdır.

IBM MQ Explorer için güvenlik

Belirli nesnelere kullanıcı erişimini denetlemeniz önemli olan bir ortamda IBM MQ kullanıyorsanız, IBM MQ Explorer ürününü kullanmanın güvenlik yönlerini göz önünde bulundurmanız gerekebilir.

IBM MQ Explorer ürününü kullanma yetkisi

Herhangi bir kullanıcı IBM MQ Explorer ögesini kullanabilir, ancak kuyruk yöneticilerine bağlanmak, kuyruk yöneticilerine erişmek ve bunları yönetmek için belirli yetkiler gereklidir.

IBM MQ Explorer kullanarak yerel yönetim görevlerini gerçekleştirmek için bir kullanıcının yönetim görevlerini gerçekleştirmek üzere gerekli yetkiye sahip olması gerekir. Kullanıcı mqm grubunun üyesiye, kullanıcının tüm yerel denetim görevlerini gerçekleştirme yetkisi vardır.

Uzak bir kuyruk yöneticisine bağlanmak ve IBM MQ Explorer komutunu kullanarak uzak denetim görevleri gerçekleştirmek için, IBM MQ Explorer komutunu yürüten kullanıcının aşağıdaki yetkileri olması gerekir:

- Hedef kuyruk yöneticisi nesnesi üzerinde CONNECT yetkisi
- Hedef kuyruk yöneticisi nesnesi üzerinde INQUIRE yetkisi
- Hedef kuyruk yöneticisi nesnesi için DISPLAY yetkisi
- Kuyruk için INQUIRE yetkisi, SYSTEM.MQEXPLORER.REPLY.MODEL
- SYSTEM.MQEXPLORER.REPLY.MODEL
- SYSTEM.MQEXPLORER.REPLY.MODEL
- SYSTEM.MQEXPLORER.REPLY.MODEL
- SYSTEM.ADMIN.COMMAND.QUEUE
- Kuyrukta INQUIRE yetkisi, SYSTEM.ADMIN.COMMAND.QUEUE
- Seçilen işlemi gerçekleştirme yetkisi

Not: GİRİŞ yetkisi, bir kuyruktan gelen girişle (alma işlemi) ilişkilidir. OUTPUT yetkisi, kullanıcıdan gelen çıkışla bir kuyrukta (koyma işlemi) ilişkilendirilir.

IBM MQ for z/OS üzerinde bir uzak kuyruk yöneticisine bağlanmak ve IBM MQ Explorer komutunu kullanarak uzak denetim görevlerini gerçekleştirmek için aşağıdakiler sağlanmalıdır:

- Sistem kuyruğu için bir RACF tanıtımı, SYSTEM.MQEXPLORER.REPLY.MODEL
- Kuyruklar için bir RACF profili, AMQ.MQEXPLORER.*

Ayrıca, IBM MQ Explorer 'ı yürüten kullanıcının aşağıdaki yetkilere sahip olması gerekir:

- RACF Sistem kuyruğu için UPDATE (güncelleme) yetkisi, SYSTEM.MQEXPLORER.REPLY.MODEL
- RACF Kuyruklar için UPDATE yetkisi, AMQ.MQEXPLORER.*
- Hedef kuyruk yöneticisi nesnesi üzerinde CONNECT yetkisi
- Seçilen işlemi gerçekleştirme yetkisi
- MQCMDS sınıfındaki tüm hlq.DISPLAY.object tanımları için okuma yetkisi

IBM MQ nesnelere yetki verilmesiyle ilgili bilgi için [AIX, Linux, and Windows sistemlerinde IBM MQ nesnesine erişim verilmesi başlıklı konuya](#) bakın.

Bir kullanıcı, gerçekleştirme yetkisi olmayan bir işlemi gerçekleştirme girişiminde bulunursa, hedef kuyruk yöneticisi yetki hatası yordamlarını çağırır ve işlem başarısız olur.

IBM MQ Explorer içindeki varsayılan süzgeç, tüm IBM MQ nesnelere görüntülenmesidir. Bir kullanıcının DISPLAY yetkisi olmayan IBM MQ nesnelere varsa, yetki başarısızlıkları üretilir. Yetki olayları kaydediliyorsa, kullanıcının DISPLAY yetkisinin olduğu nesnelere görüntülenene nesne aralığını sınırlayın.

IBM MQ Explorer ' den uzak kuyruk yöneticilerine bağlanma güvenliği

IBM MQ Explorer ile her uzak kuyruk yöneticisi arasındaki kanalı güvenli hale getirmeniz gerekir.

IBM MQ Explorer , uzak kuyruk yöneticilerine MQI istemci uygulaması olarak bağlanır. Bu, her uzak kuyruk yöneticisinin bir sunucu bağlantısı kanalı tanımlaması ve uygun bir TCP/IP dinleyicisi olması gerektiği anlamına gelir. Sunucu bağlantı kanalınızı güvenli hale getirmezseniz, kötü amaçlı bir uygulamanın aynı sunucu bağlantı kanalına bağlanması ve kuyruk yöneticisi nesnelere sınırsız yetkiyle erişmesi mümkündür. Sunucu bağlantı kanalınızın güvenliğini sağlamak için, kanalı MCAUSER özniteliği için boş olmayan bir değer belirleyin, kanal kimlik doğrulama kayıtlarını kullanın ya da bir güvenlik çıkışı kullanın.

MCAUSER özniteliğinin varsayılan değeri yerel kullanıcı kimliğidir. Sunucu bağlantı kanalının MCAUSER özniteliği olarak boş olmayan bir kullanıcı adı belirlerseniz, bu kanalı kullanarak kuyruk yöneticisine bağlanan tüm programlar, adı belirtilen kullanıcının kimliğiyle çalışır ve aynı yetki düzeyine sahiptir. Kanal kimlik doğrulama kayıtlarını kullanırsanız bu gerçekleşmez.

IBM MQ Explorer ile güvenlik çıkışının kullanılması

IBM MQ Explorer kullanarak varsayılan bir güvenlik çıkışı ve kuyruk yöneticisine özgü güvenlik çıkışları belirtebilirsiniz.

IBM MQ Explorer' den tüm yeni istemci bağlantıları için kullanılacak varsayılan bir güvenlik çıkışı tanımlayabilirsiniz. Bu varsayılan çıkış, bağlantı kurulduğunda geçersiz kılınabilir. Ayrıca, tek bir kuyruk yöneticisi ya da bir kuyruk yöneticisi kümesi için bir güvenlik çıkışı da tanımlayabilirsiniz; bu, bir bağlantı kurulduğunda yürürlüğe girer. Çıkışları IBM MQ Explorer kullanarak belirtirsiniz. Daha fazla bilgi için bkz. IBM MQ Explorer Yardımı.

TLS etkin MQI kanallarını kullanarak uzak bir kuyruk yöneticisine bağlanmak için IBM MQ Explorer olanağının kullanılması

IBM MQ Explorer , bir MQI kanalını kullanarak uzak kuyruk yöneticilerine bağlanır. TLS güvenliğini kullanarak MQI kanalının güvenliğini sağlamak istiyorsanız, kanalı bir istemci kanal tanımlama çizelgesini kullanarak oluşturmanız gerekir.


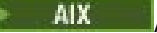

Bir istemci kanal tanımlama çizelgesi kullanılarak MQI kanalı oluşturulması hakkında bilgi için bkz. [IBM MQ MQI clients](#).

Kanalı bir istemci kanal tanımlama çizelgesini kullanarak oluşturduğunuzda, “[Uzak kuyruk yöneticisini barındıran sistemdeki görevler](#)” sayfa 142 ve “[IBM MQ Explorer ürününü barındıran sistemdeki görevler](#)” sayfa 143 içinde açıklandığı gibi TLS etkinleştirilmiş MQI kanalını kullanarak uzak bir kuyruk yöneticisine bağlanmak için IBM MQ Explorer komutunu kullanabilirsiniz.

Uzak kuyruk yöneticisini barındıran sistemdeki görevler

Uzak kuyruk yöneticisini barındıran sistemde aşağıdaki görevleri gerçekleştirin:

1. Bir sunucu bağlantısı ve istemci bağlantısı kanalı çifti tanımlayın ve her iki kanalda sunucu bağlantısında *SSLCIPH* özniteliği için uygun değeri belirtin. *SSLCIPH* özniteliği hakkında daha fazla bilgi için bkz. [TLS ile kanalları koruma](#).
2. Kuyruk yöneticisinin @ipcc dizininde bulunan AMQCLCHL . TAB kanalı tanımlama çizelgesini IBM MQ Explorer' yi barındıran sisteme gönderin.
3. Belirlenen bir kapıda TCP/IP dinleyicisi başlatır.
4. CA ve kişisel TLS sertifikalarını kuyruk yöneticisinin SSL dizinine yerleştirin:

-   AIX and Linux sistemleri için /var/mqm/qmgrs/+QMNAME+/SSL .
-  Windows sistemleri için C:\Program Files\IBM\MQ\qmgrs\+QMNAME+\SSL .

Burada +QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.

5. key . kdbadlı CMS tipinde bir anahtar veritabanı dosyası oluşturun. **strmqikm** (iKeyman) içindeki seçeneği işaretleyerek parolayı bir dosyada saklama GUI ya da -stash seçeneğini **runmqckm** komutuyla ya da **runmqackm** komutuyla kullanarak.

6. CA sertifikalarını önceki adımda oluşturulan anahtar veritabanına ekleyin.

7. Kuyruk yöneticisine ilişkin kişisel sertifikayı anahtar veritabanına aktarın.

Windows sistemlerinde TLS ile çalışma hakkında daha ayrıntılı bilgi için bkz. [AIX, Linux, and Windows üzerinde TLS ile çalışma](#).

IBM MQ Explorer ürününü barındıran sistemdeki görevler

IBM MQ Explorer ürününü barındıran sistemde aşağıdaki görevleri gerçekleştirin:

1. key . jks adlı JKS tipinde bir anahtar veritabanı dosyası oluşturun. Bu anahtar veritabanı dosyası için bir parola belirleyin.
IBM MQ Explorer , TLS güvenliği için Java anahtar deposu dosyalarını (JKS) kullandığından, IBM MQ Explorer için TLS yapılandırmak üzere oluşturulan anahtar deposu dosyası bununla eşleşmelidir.
2. CA sertifikalarını önceki adımda oluşturulan anahtar veritabanına ekleyin.
3. Kuyruk yöneticisine ilişkin kişisel sertifikayı anahtar veritabanına aktarın.
4. Windows ve Linux sistemlerinde, sistem menüsünü, MQExplorer yürütülür dosyasını ya da **strmqcfcg** komutunu kullanarak IBM MQ Explorer ' ı başlatın.
5. IBM MQ Explorer araç çubuğundan **Pencere-> Tercihler** öğelerini tıklayın, **IBM MQ Explorer** öğesini genişletin ve **SSL Client Certificate Stores** öğesini tıklayın. "IBM MQ Explorer ürününü barındıran sistemdeki görevler" sayfa 143' un 1. adımında yaratılan JKS dosyasının adını ve parolasını girin ve **Tamam** düğmesini tıklayın.
6. **Tercihler** penceresini kapatın ve **Kuyruk Yöneticileri** ' ni sağ tıklayın. **Kuyruk Yöneticilerini Göster/Gizle** ' yi tıklayın ve **Kuyruk Yöneticilerini Göster/Gizle** ekranında **Ekle** ' yi tıklayın.
7. Kuyruk yöneticisinin adını yazın ve **Doğrudan bağlan** seçeneğini belirleyin. Next (İleri) düğmesini tıklayın.
8. **Use client channel definition table (CCDT)** (İstemci kanal tanımlama çizelgesini kullan) seçeneğini belirleyin ve uzak kuyruk yöneticisini barındıran sistemdeki "Uzak kuyruk yöneticisini barındıran sistemdeki görevler" sayfa 142 2. adımda uzak kuyruk yöneticisinden aktardığınız kanal çizelgesi dosyasının yerini belirtin.
9. **Finish** (Son) düğmesini tıklayın. Artık IBM MQ Explorer' den uzak kuyruk yöneticisine erişebilirsiniz.

IBM MQ Explorer ile başka bir kuyruk yöneticisiyle bağlantı kurulması

IBM MQ Explorer , IBM MQ Explorer ' in zaten bağlı olduğu bir ara kuyruk yöneticisi aracılığıyla bir kuyruk yöneticisine bağlanmanızı sağlar.

Bu durumda, IBM MQ Explorer , PCF komut iletilerini aşağıdakileri belirterek ara kuyruk yöneticisine koyar:

- Hedef kuyruk yöneticisinin adı olarak, nesne tanımlayıcıdaki (MQOD) *ObjectQMGrAd* parametresi. Kuyruk adı çözümlemesi hakkında daha fazla bilgi için bkz. [Ad çözümlemesi](#).
- İleti tanımlayıcısında (MQMD) yerel userId olarak *UserIdentifier* parametresi.

Bağlantı daha sonra bir ara kuyruk yöneticisi aracılığıyla hedef kuyruk yöneticisine bağlanmak için kullanılırsa, *userId* , ileti tanımlayıcısının (MQMD) *UserIdentifier* değiştirilmesinde yeniden aktılır. Hedef kuyruk yöneticisindeki MCA dinleyicisinin bu iletiyi kabul etmesi için, MCAUSER özniteliği ayarlanmalı ya da *userId* önceden koyma yetkisine sahip olmalıdır.

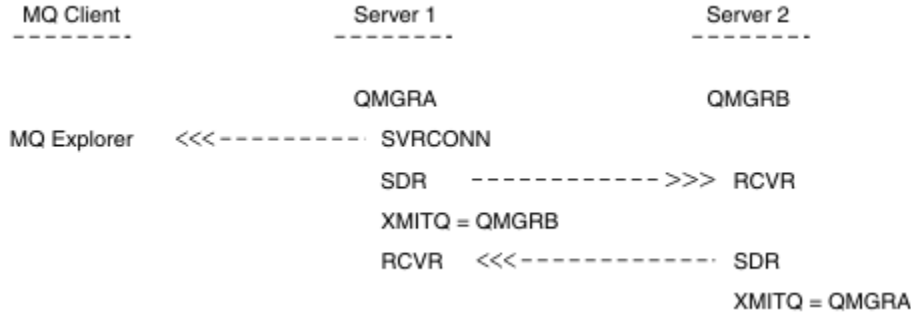
Hedef kuyruk yöneticisindeki komut sunucusu, ileti tanımlayıcısındaki (MQMD) *UserIdentifier* parametresinde *userId* değerini belirterek iletileri iletim kuyruğuna koyar. Bunun başarılı olabilmesi için *userId* , koyma yetkisi olan hedef kuyruk yöneticisinde önceden var olmalıdır.

Aşağıdaki örnek, bir kuyruk yöneticisinin bir ara kuyruk yöneticisi aracılığıyla IBM MQ Explorer' e nasıl bağlanacağını göstermektedir.

Bir kuyruk yöneticisiyle uzak denetim bağlantısı kurun. Aşağıdakileri doğrulayın:

- Sunucudaki kuyruk yöneticisi etkin ve tanımlı bir sunucu bağlantısı kanalı (SVRCONN) var.

- Dinleyici etkin.
- Komut sunucusu etkin.
- SYSTEM.MQ EXPLORER.REPLY.MODEL kuyruğu yaratıldı ve yeterli yetkiniz var.
- Kuyruk yöneticisi dinleyicileri, komut sunucuları ve gönderen kanalları başlatılır.



Bu örnekte:

- IBM MQ Explorer , bir istemci bağlantısı kullanarak QMGRA (Server1üzerinde çalışan) kuyruk yöneticisine bağlanır.
- Server2 üzerindeki QMGRB kuyruk yöneticisi, artık bir ara kuyruk yöneticisi (QMGRA) aracılığıyla IBM MQ Explorer ' e bağlanabilir.
- IBM MQ Explorerile QMGRB bağlantısı kurarken, ara kuyruk yöneticisi olarak QMGRA ögesini seçin

Bu durumda, IBM MQ Exploreriçinden QMGRB ile doğrudan bağlantı yoktur; QMGRB ile bağlantı QMGRAaracılığıyla kurulur.

Server2 üzerindeki QMGRB kuyruk yöneticisi, Server1 üzerindeki QMGRA sunucusuna gönderici-alıcı kanalları kullanılarak bağlanır. QMGRA ile QMGRB arasındaki kanal, uzaktan yönetimin mümkün olabileceği şekilde ayarlanmalıdır; bkz. [“Uzak denetim için kuyruk yöneticilerinin yapılandırılması” sayfa 213.](#)

IBM MQ Explorer içinde kuyruk yöneticilerini ve kümelerini gösterme ve gizleme

IBM MQ Explorer aynı anda birden çok kuyruk yöneticisi görüntüleyebilir. Kuyruk Yöneticilerini Göster/Gizle panosunda (Kuyruk Yöneticileri ağaç düğümüne ilişkin menüden seçilebilir), başka bir (uzak) makineye ilişkin bilgileri görüntüleyip görüntümeyeceğinizi seçebilirsiniz. Yerel kuyruk yöneticileri otomatik olarak saptanır.

Uzak kuyruk yöneticisini göstermek için:

1. **Kuyruk Yöneticileri** ağaç düğümünü sağ tıklayın ve **Kuyruk Yöneticilerini Göster/Gizle** seçeneğini belirleyin.
2. **Ekle**'yi tıklayın. Kuyruk Yöneticilerini Göster/Gizle panosu görüntülenir.
3. Sağlanan alanlara uzak kuyruk yöneticisinin adını ve anasistem adını ya da IP adresini girin.
Anasistem adı ya da IP adresi, uzak kuyruk yöneticisiyle varsayılan sunucu bağlantı kanalı SYSTEM.ADMIN.SVRCONNya da kullanıcı tanımlı bir sunucu bağlantı kanalı.
4. **Bitir**'i tıklayın.

Kuyruk Yöneticilerini Göster/Gizle panosu, tüm görünür kuyruk yöneticilerinin bir listesini de görüntüler. Kuyruk yöneticilerini gezinme görünümünden gizlemek için bu panoyu kullanabilirsiniz.

IBM MQ Explorer , bir kümenin üyesi olan bir kuyruk yöneticisini görüntülense, küme algılanır ve otomatik olarak görüntülenir.

Uzak kuyruk yöneticileri listesini bu panodan dışa aktarmak için:

1. Kuyruk Yöneticilerini Göster/Gizle panosunu kapatın.

2. IBM MQ Explorer Gezinme bölümünde en yüksek **IBM MQ** ağaç düğümünü sağ tıklayın ve **Dışa Aktar IBM MQ Explorer Ayarları** seçeneğini belirleyin.
3. **IBM MQ Explorer > IBM MQ Explorer Ayarlar** ' ı tıklayın.
4. **Bağlantı Bilgileri > Uzak kuyruk yöneticileri** seçeneklerini belirleyin.
5. Dışa aktarılan ayarların saklanacağı dosyayı seçin.
6. Son olarak, uzak kuyruk yöneticisi bağlantı bilgilerini belirtilen dosyaya aktarmak için **Son** düğmesini tıklayın.

Uzak kuyruk yöneticilerinin listesini içe aktarmak için:

1. IBM MQ Explorer Gezinme bölümünde en yüksek **IBM MQ** ağaç düğümünü sağ tıklayın ve **İçe Aktar IBM MQ Explorer Ayarları** seçeneğini belirleyin.
2. **IBM MQ Explorer > IBM MQ Explorer Ayarlar** ' ı tıklayın.
3. **Göz At** düğmesini tıklayın ve uzak kuyruk yöneticisi bağlantı bilgilerini içeren dosyanın yoluna gidin.
4. **Aç**' ı tıklayın. Dosya uzak kuyruk yöneticilerinin bir listesini içeriyorsa, **Bağlantı Bilgileri > Uzak kuyruk yöneticileri** kutusu seçilir.
5. Son olarak, uzak kuyruk yöneticisi bağlantı bilgilerini IBM MQ Explorer içine aktarmak için **Son** düğmesini tıklayın.

Küme üyeliği ve IBM MQ Explorer

IBM MQ Explorer , bir kümenin üyesi olan kuyruk yöneticileriyle ilgili bilgi gerektirir.

Bir kuyruk yöneticisi bir kümenin üyesiye, küme ağacı düğümüne otomatik olarak veri yerleştirilir.

Kuyruk yöneticileri, IBM MQ Explorer çalışırken kümelerin üyeleri haline gelirse, kümeler hakkında güncel yönetim verileriyle IBM MQ Explorer ' i güncel tutmanız gerekir; böylece, bu kümelerle etkili bir şekilde iletişim kurabilir ve istendiğinde doğru küme bilgilerini görüntüleyebilirsiniz. Bunu yapmak için IBM MQ Explorer aşağıdaki bilgilere gereksinim duyar:

- Havuz kuyruğu yöneticisinin adı
- Uzak bir kuyruk yöneticisindeyse, havuz kuyruğu yöneticisinin bağlantı adı

Bu bilgilerle IBM MQ Explorer şunları yapabilir:

- Kümedeki kuyruk yöneticilerinin bir listesini elde etmek için havuz kuyruğu yöneticisini kullanın.
- Kümenin üyesi olan ve desteklenen platformlar ve komut düzeylerinde olan kuyruk yöneticilerini yönetin.

Aşağıdaki durumlarda yönetim olanaklı değildir:

- Seçilen havuz kullanılamaz duruma gelir. IBM MQ Explorer otomatik olarak alternatif bir havuza geçmez.
- Seçilen havuzla TCP/IP üzerinden iletişim kurulamıyor.
- Seçilen havuz, bir altyapıda çalışan bir kuyruk yöneticisinde çalışıyor ve komut düzeyi IBM MQ Explorer tarafından desteklenmiyor.

Denetlenebilen küme üyeleri yerel olabilir ya da TCP/IP ile iletişim kurulabiliyorsa uzak olabilir. IBM MQ Explorer , bir kümenin üyesi olan yerel kuyruk yöneticilerine istemci bağlantısı kullanmadan doğrudan bağlanır.

IBM MQ Explorer için veri dönüştürme

IBM MQ Explorer CCSID 1208 (UTF-8) ile çalışır. Bu, IBM MQ Explorer ' in uzak kuyruk yöneticilerinden alınan verileri doğru şekilde görüntülemesini sağlar. IBM MQ Explorer , bir kuyruk yöneticisine doğrudan bağlanma ya da bir ara kuyruk yöneticisi kullanarak tüm gelen iletilerin CCSID 1208 ' e (UTF-8) dönüştürülmesini gerektirir.

IBM MQ Explorer ile IBM MQ Explorer ' in tanımadığı bir CCSID' ye sahip bir kuyruk yöneticisi arasında bağlantı kurmaya çalışırsanız bir hata iletisi gönderilir.

Desteklenen dönüştürmeler [Kod sayfası dönüşümü](#) başlıklı konuda açıklanmıştır.

Windows IBM MQ Görev Çubuğu uygulamasını kullanma (yalnızca Windows)

IBM MQ Görev Çubuğu uygulaması, sunucudaki Windows sistem tepesinde bir simge görüntüler. Simge, IBM MQ ' un geçerli durumunu ve bazı basit eylemleri gerçekleştirebileceğiniz bir menüyü sağlar.

Windows işletim sisteminde IBM MQ simgesi, sunucudaki sistem tepesinde bulunur ve renk kodlu bir durum simgesiyle örtüşür. Bu simge, aşağıdaki anlamlardan birine sahip olabilir:

Yeşil

Doğru çalışıyor; şu anda uyarı yok

Mavi

Belirsiz; IBM MQ başlatılıyor ya da kapatılıyor

Sarı

Uyarı; bir ya da daha çok hizmet başarısız oldu ya da zaten başarısız oldu

Menüyü görüntülemek için IBM MQ simgesini sağ tıklatın. Menüden aşağıdaki işlemleri gerçekleştirebilirsiniz:

- IBM MQ Uyarı İzleyicisini açmak için **Aç** düğmesini tıklatın.
- IBM MQ Görev Çubuğu uygulamasından çıkmak için **Çık** düğmesini tıklatın.
- IBM MQ Explorer ürününü başlatmak için **IBM MQ Explorer** simgesini tıklatın.
- IBM MQ' ı durdurmak için **Durdur IBM MQ** düğmesini tıklatın.
- IBM MQ Alert Monitor ile ilgili bilgileri görüntülemek için **About IBM MQ** (Hakkında) seçeneğini tıklatın.

Windows IBM MQ uyarı izleyici uygulaması (yalnızca Windows)

IBM MQ uyarı izleyicisi, yerel bir makinedeki IBM MQ ile ilgili sorunları tanımlayan ve kaydeden bir hata saptama aracıdır.

Uyarı izleyicisi, bir IBM MQ sunucusunun yerel kuruluşunun yürürlükteki durumuyla ilgili bilgileri görüntüler. Ayrıca, Windows Advanced Configuration and Power Interface (ACPI) olanağını izler ve ACPI ayarlarının uygulanmasını sağlar.

IBM MQ uyarı izleyicisinden şunları yapabilirsiniz:

- IBM MQ Explorer ' a doğrudan erişim
- Bekleyen tüm uyarılarla ilgili bilgileri görüntüle
- Yerel makinede IBM MQ hizmetini kapatma
- Uyarı iletilerini ağ üzerinden yapılandırılabilir bir kullanıcı hesabına ya da Windows iş istasyonuna ya da sunucusuna yönlendir

Yerel IBM MQ nesneleriyle çalışma

İleti Kuyruğu Arabirimi 'ni (MQI) kullanan uygulama programlarını desteklemek için yerel IBM MQ nesneleri denetleyebilirsiniz.

Bu görev hakkında

Bu bağlamda, yerel yönetim IBM MQ nesneleri oluşturma, görüntüleme, değiştirme, kopyalama ve silme anlamına gelir.

Bu bölümde açıklanan yaklaşımlara ek olarak, IBM MQ Explorer yerel IBM MQ nesneleri yönetmek için kullanılabilir. Daha fazla bilgi için bkz [“IBM MQ Explorer kullanarak yönetim” sayfa 138.](#)

Yordam

- Yerel IBM MQ nesnelarını yönetmenize yardımcı olması için aşağıdaki konulardaki bilgileri kullanın.
 - [MQI kullanan uygulama programları](#)
 - [“IBM MQ ' nin MQSC komutlarını kullanarak denetlenmesi” sayfa 12](#)
 - [“Kuyruk yöneticisi özniteliklerini görüntüleme ve deęiřtirme” sayfa 154](#)
 - [“Yerel kuyruklarla çalışma” sayfa 158](#)
 - [“Diđer ad kuyruklarıyla çalışılması” sayfa 170](#)
 - [“Model kuyruklarıyla çalışma” sayfa 172](#)
 - [“Hizmetlerle çalışma” sayfa 199](#)
 - [“Tetikleme için nesnelari yönetme” sayfa 206](#)

Kuyruk yöneticileriyle çalışma

Bir kuyruk yöneticisini başlatmak ve durdurmak için denetim komutlarını kullanabilirsiniz. Kuyruk yöneticisi özniteliklerini görüntülemek ya da deęiřtirmek için MQSC komutlarını kullanabilirsiniz.

İlgili görevler

[Çoklu platformlarda kuyruk yöneticileri oluşturma](#)

Multi Kuyruk yöneticisinin başlatılması

Bir kuyruk yöneticisi yarattığınızda, komutları ya da MQI çağrılarını işlemlerini sağlamak için kuyruk yöneticisini başlatmanız gerekir.

Bu görev hakkında

strmqm komutunu kullanarak bir kuyruk yöneticisini başlatabilirsiniz. **strmqm** komutunun açıklaması ve seçenekleri için bkz. [strmqm](#).

Linux **Windows** Alternatif olarak, Windows ve Linux (x86 ve x86-64 platformları) sistemlerinde IBM MQ Explorerkomutunu kullanarak bir kuyruk yöneticisi başlatabilirsiniz.

Windows Windows sistemlerinde, sistem IBM MQ Explorerolanađını kullanmaya başladığında otomatik olarak bir kuyruk yöneticisi başlatabilirsiniz. Daha fazla bilgi için bkz [“IBM MQ Explorer kullanarak yönetim” sayfa 138](#).

Yordam

- **strmqm** komutunu kullanarak bir kuyruk yöneticisini başlatmak için, komutu ve ardından başlatmak istediğiniz kuyruk yöneticisinin adını girin.

Örneđin, QMBadlı bir kuyruk yöneticisini başlatmak için řu komutu girin:

```
strmqm QMB
```

Not: Çalıştığınız kuyruk yöneticisiyle ilişkili kuruluřta **strmqm** komutunu kullanmanız gerekir. **dspm** -o **installation** komutunu kullanarak, bir kuyruk yöneticisinin hangi kuruluřla ilişkilendirildiđi öğrenebilirsiniz.

strmqm komutu, kuyruk yöneticisi başlatılıncaya ve bađlantı isteklerini kabul etmeye hazır oluncaya kadar denetimi döndürmez.

- **Linux** **Windows**

IBM MQ Explorerkomutunu kullanarak bir kuyruk yöneticisini başlatmak için aşağıdaki adımları izleyin:

- a) IBM MQ Explorer'ını açın.
- b) Navigator görünümünde kuyruk yöneticisini seçin.

c) **Başlat'**ı tıklatın.

Sonuçlar

Kuyruk yöneticisi başlar.

Kuyruk yöneticisi başlatma işlemi birkaç saniyeden uzun sürerse IBM MQ , başlatma işleminin ilerleyişini aralıklı olarak ayrıntılı olarak açıklayan bilgi iletileri yayınlar.

Multi

Kuyruk yöneticisinin durdurulması

Bir kuyruk yöneticisini durdurmak için **endmqm** komutunu kullanabilirsiniz. Bu komut, bir kuyruk yöneticisini durdurmanın dört yolunu sağlar: denetimli ya da susturulmuş, sona erdirmeye, anında sona erdirmeye, önleyici kapatma ve bekleme sona erdirmeye. Diğer bir seçenek olarak, Windows ve Linux sistemlerinde IBM MQ Explorer komutunu kullanarak bir kuyruk yöneticisini durdurabilirsiniz.

Bu görev hakkında

endmqm komutuyla tek bir yönetim ortamı kuyruk yöneticisini durdurmanın dört yolu vardır:

Denetimli (susturulmuş) sona erdirmeye

Varsayılan olarak, **endmqm** komutu belirtilen kuyruk yöneticisini susturulmuş olarak sona erdirir. Susturulmuş bir kapatma işlemi, bağlı tüm uygulamaların bağlantısı kesilinceye kadar bekler, bu nedenle tamamlanması biraz zaman alabilir.

Anında kapatma

Anında sona erdirmeye için, yürürlükteki MQI çağrılarını tamamlanmasına izin verilir, ancak yeni çağrılar başarısız olur. Bu tip bir sona erdirmeye, uygulamaların kuyruk yöneticisiyle bağlantısını kesmesini beklemez.

Önleyici kapatma

Kuyruk yöneticisi hemen durur. Bu tip bir sona erdirmeye yalnızca kural dışı durumlarda (örneğin, bir kuyruk yöneticisi olağan bir **endmqm** komutunun sonucu olarak durmazsa) kullanılır.

Bekleme sona erdirmeye

Bu sona erdirmeye tipi, denetimin yalnızca kuyruk yöneticisi durduktan sonra size geri döndürülmesi dışında, denetimli sona erdirmeye eşdeğerdir.

endmqm komutu, çok eşgörünümlü bir kuyruk yöneticisinin tüm yönetim ortamlarını, tek bir yönetim ortamı kuyruk yöneticisini durdurduğu gibi durdurur. **endmqm** komutunu etkin yönetim ortamında ya da çok eşgörünümlü bir kuyruk yöneticisinin yedek yönetim ortamlarından birinde yayınlatabilirsiniz. Ancak, kuyruk yöneticisini sona erdirmek için etkin yönetim ortamında **endmqm** komutunu vermeniz gerekir.

IBM MQ 9.1.4' de, temel kuyruk yöneticisi bakım görevlerini kesintiye uğratmadan ya da kesintiye uğratmadan, kuyruk yöneticisini belirlediğiniz bir saniye içinde sona erdirmeye seçeneğiniz vardır (bkz. "Hedef zaman içinde bir kuyruk yöneticisinin sona erdirilmesi" sayfa 150).



Uyarı:

- Kalıcı iletiler, kullanılan kapatma tipinden bağımsız olarak (el ile sona erdirmeye IBM MQ işlemleri de içinde olmak üzere) kalır; kalıcı olmayan iletiler ise, herhangi bir kapatma tipinde hayatta kalacağı garanti edilemez.
NPMCLASS (HIGH) kuyruk özelliğinin belirtilmesi, kalıcı olmayan iletileri en iyi yapılması gerekenlerle kaydeder. **endmqm -t**, **endmqm -tp**, **endmqm -pya** da el ile sona erdirmeye IBM MQ işlemlerinin kullanılması, **endmqm -w** ya da **endmqm -i** ile karşılaştırıldığında NPMCLASS (HIGH) iletilerinin IBM MQ sona erdirmeye ya da yeniden başlatma döngüsünden kalma olasılığını azaltır.
- Kuyruk yöneticisini sona erdirmeye ve yeniden başlatmak için birleşik süre, özellikle **-p** ve **-tp** seçeneklerini kullanırken daha ani bir sona erdirmeye yönteminin kullanılması sonucunda daha uzun olabilir.


Kuyruk yöneticisinin kuyruk yöneticisini sona erdirmek için IBM MQ işlemlerini sonlandırmaya başvurması gerekiyorsa, kuyruk yöneticisi yeniden başlatıldığında kuyruk yöneticisi durumunun daha fazla mutabakatı gerekebilir.

endmqm komutunun ayrıntılı açıklaması ve seçenekleri için bkz. [endmqm](#).

İpucu: Bir kuyruk yöneticisinin kapatılmasına yönelik sorunlar genellikle uygulamalardan kaynaklanır. Örneğin, uygulamalar:

- MQI dönüş kodlarını doğru biçimde denetleme
- Susturma bildirimini isteme
- Kuyruk yöneticisiyle bağlantıyı kesmeden sonlandırma (bir MQDISC çağrısı yayınlayarak)

Kuyruk yöneticisini durdurmaya çalıştığınızda bir sorun ortaya çıkarsa, Ctrl-C tuşlarını kullanarak **endmqm** komutundan çıkabilirsiniz. Daha sonra başka bir **endmqm** komutu yayınlayabilirsiniz, ancak bu sefer gerek duyduğunuz sona erdirmeye tipini belirten bir parametreyle.

 **endmqm** komutunu kullanmaya alternatif olarak, Windows ve Linux sistemlerinde, denetimli ya da anında sona erdirmeye gerçekleştirmek için IBM MQ Explorer komutunu kullanarak bir kuyruk yöneticisini durdurabilirsiniz.

Yordam

- Kuyruk yöneticisini **endmqm** komutunu kullanarak durdurmak için, gerekiyorsa uygun parametreyi ve durdurmak istediğiniz kuyruk yöneticisinin adını girin.

Not: Çalıştığınız kuyruk yöneticisiyle ilişkili kuruluştaki **endmqm** komutunu kullanmanız gerekir. Bir kuyruk yöneticisinin hangi kuruluştaki ilişkilendirildiği öğrenmek için **dspmq** komutunu kullanın:

```
dspmq -o installation
```

- Denetimli (susturulmuş) bir sona erdirmeye gerçekleştirmek için, aşağıdaki örnekte gösterildiği gibi **endmqm** komutunu girin; bu komut QMBadlı bir kuyruk yöneticisini durdurur:

```
endmqm QMB
```

Diğer bir seçenek olarak, **endmqm** komutunun **-c** parametresiyle girilmesi, aşağıdaki örnekte gösterildiği gibi, bir **endmqm QMB** komutuna eşdeğerdir.

```
endmqm -c QMB
```

Her iki durumda da denetim hemen size döndürülür ve kuyruk yöneticisi durduğunda size bildirim gönderilmez. Komutun tüm uygulamalar duruncaya kadar beklemesini ve kuyruk yöneticisinin denetimi size döndürmeden önce sona ermesini istiyorsanız, aşağıdaki örnekte gösterildiği gibi **-w** değiştirgesini kullanın.

```
endmqm -w QMB
```

- Hemen kapatma işlemini gerçekleştirmek için, aşağıdaki örnekte gösterildiği gibi **endmqm** komutunu **-i** parametresiyle birlikte girin:

```
endmqm -i QMB
```

- Önleyici bir sona erdirmeye gerçekleştirmek için, aşağıdaki örnekte gösterildiği gibi **endmqm** komutunu **-p** parametresiyle girin:

```
endmqm -p QMB
```



Uyarı: Önleyici bir kapatma, bağlı uygulamalar için öngörülemeyen sonuçlar doğurabilir. Olağan bir **endmqm** komutunu kullanarak kuyruk yöneticisini durdurmak için yapılan diğer tüm girişimler başarısız olmadıkça bu seçeneği kullanmayın. **ALW** Önleyici kapatma çalışmazsa, bunun yerine “Kuyruk yöneticisinin el ile durdurulması” sayfa 151 komutunu deneyin.

- Otomatik istemci yeniden bağlantısı isteğinde bulunmak için -r parametresiyle **endmqm** komutunu girin. Bu parametre, kuyruk yöneticisi gruplarındaki diğer kuyruk yöneticilerine istemcilerin bağlantırlığının yeniden oluşturulmasını sağlar.

Not: Varsayılan **endmqm** komutu kullanılarak bir kuyruk yöneticisinin sona erdirilmesi, otomatik istemci yeniden bağlantısını tetiklemez.

- Etkin yönetim ortamını sona erdirdikten sonra çok eşgörünümlü bir kuyruk yöneticisinin yedek yönetim ortamına aktarmak için, çok eşgörünümlü kuyruk yöneticisinin etkin yönetim ortamında **endmqm** komutunu -s değiştirilmesiyle girin.
- Çok eşgörünümlü bir kuyruk yöneticisinin yedek yönetim ortamını sona erdirmek ve etkin yönetim ortamını çalışır durumda bırakmak için, çok eşgörünümlü kuyruk yöneticisinin yedek yönetim ortamında -x değiştirilmesiyle **endmqm** komutunu girin.

- **Linux** **Windows**

Windows ve Linux sistemlerinde, IBM MQ Explorer komutunu kullanarak kuyruk yöneticisini durdurmak için aşağıdaki adımları izleyin:

- a) IBM MQ Explorer'ı açın.
- b) Navigator Görünümünden kuyruk yöneticisini seçin.
- c) **Durdur** u tıklatın.
- d) **End Queue Manager** (Kuyruk Yöneticisini Sona Erdir) panosu görüntülenir.
- d) **Denetimli** ya da **Hemenseçeneği**ni belirleyin.
- e) **Tamam**'ı tıklatın.

Kuyruk yöneticisi durur.

İlgili görevler

[AIX üzerinde çok eşgörünümlü kuyruk yöneticilerine bakım düzeyi güncellemeleri uygulama](#)

[Linux üzerinde çok eşgörünümlü kuyruk yöneticilerine bakım düzeyi güncellemeleri uygulama](#)

[Windows üzerinde çok eşgörünümlü kuyruk yöneticilerine bakım düzeyi güncellemeleri uygulama](#)

İlgili başvurular

[endmqm \(kuyruk yöneticisini sona erdir\)](#)

Hedef zaman içinde bir kuyruk yöneticisinin sona erdirilmesi

Kuyruk yöneticisini, temel kuyruk yöneticisi bakım görevlerini kesintiye uğratmadan ya da kesintiye uğratmadan, belirlediğiniz bir saniye sayısı kadar bir süre içinde sona erdirebilirsiniz.

endmqm komutunu kullanırken hedef saati belirtmenin iki yolu vardır. **-t** seçeneği, temel kuyruk yöneticisi bakım görevlerinin tamamlanmasını sağlar; bu, kuyruk yöneticisinin sona erme aşamasını uzatabilir. **-tp** seçeneği, belirtilen hedef zamana uymak için gerekirse temel kuyruk yöneticisi bakım görevlerini kesintiye uğratır.

Bir hedef saat belirttiğinizde, **-w**, **-i** ya da **-p** kapatma tipi başlangıç kapatma tipini gösterir.

Not: An immediate shutdown is still orderly, differing from a controlled shutdown primarily in the way that any running applications are quiesced. **immediate** kapatma işlemi, kuyruk sıkıştırma ve uzun sürebilecek NPMCLASS (HIGH) iletilerini kalıcı kılma gibi bakım işlemleri gerçekleştirmeye devam ederken, sınırlı bir kapatma işlemi, hedef zamana müdahale ettiklerinde bu ek işlemleri durdurur.

Kuyruk yöneticisi, hedef saati karşılamak için sona erdirme tipini gerektiği şekilde yükseltir. Örneğin:

- **-w** ile başlayan 10 saniyelik bir **-t** hedefi, yedi saniyelik bir susturma, iki saniyelik bir kısa süre içinde kuyruk yöneticisinin hemen kapatılması, daha fazla bakım gerektirmeden hemen kapatılması olabilir:

```
endmqm -w -t 10 queue_manager
```

- 10 saniyelik bir **-tp** hedefi, yedi saniyelik susturma, iki saniyelik bir kapatma, oda temizliği de içinde olmak üzere kuyruk yöneticisinin hemen kapatılması, daha fazla bakım gerektirmeden bir saniyelik bir kapatma ve daha sonra, IBM MQ işlemlerini sonlandırmaya başlama olabilir:

```
endmqm -c -tp 10 queue_manager
```

- **-i** konumunda iki saniyelik bir **-tp** hedefi, kuyruk yöneticisinin hemen kapatılması, bakım yapılmadan hemen kapatılması ve daha sonra IBM MQ işlemlerinin sona erdirilmesi olabilir:

```
endmqm -i -tp 2 queue_manager
```

- **-w** konumunda ikinci bir hedef, waitadresinde 0.1 saniye olabilir; örneğin, IBM MQ dönüş kodlarını bağlı uygulamalara gönderecek kadar uzun, 0.9 saniye sonra kuyruk yöneticisinin hemen kapanması (bakım da içinde olmak üzere), ardından IBM MQ işlemlerini hemen sona erdirmeye başlayabilirsiniz.

İlgili başvurular

[endmqm \(kuyruk yöneticisini sona erdir\)](#)

ALW Kuyruk yöneticisinin el ile durdurulması

Bir kuyruk yöneticisini durdurma ve kaldırma için standart yöntemler başarısız olursa, kuyruk yöneticisini el ile durdurmayı deneyebilirsiniz.

Bu görev hakkında

Kuyruk yöneticilerini durdurmanın standart yolu, [“Kuyruk yöneticisinin durdurulması” sayfa 148](#) içinde açıklandığı gibi **endmqm** komutunu kullanmaktır. Bir kuyruk yöneticisini standart şekilde durduramazsanız, kuyruk yöneticisini el ile durdurmayı deneyebilirsiniz. Bunu nasıl yapacağınız, hangi platformu kullandığınıza bağlıdır.

Yordam

- **Windows**
Windows üzerinde bir kuyruk yöneticisini durdurmak için bkz. [“Windows üzerinde bir kuyruk yöneticisinin el ile durdurulması” sayfa 151](#).
- **Linux** **AIX**
AIX ya da Linux üzerinde bir kuyruk yöneticisini durdurmak için bkz. [“AIX and Linux üzerinde bir kuyruk yöneticisinin el ile durdurulması” sayfa 152](#).

İlgili görevler

[Çoklu platformlarda kuyruk yöneticilerinin yaratılması ve yönetilmesi](#)

İlgili başvurular

[sonmqm](#)

Windows üzerinde bir kuyruk yöneticisinin el ile durdurulması

endmqm komutunu kullanarak Windows üzerinde bir kuyruk yöneticisini durduramazsanız, çalışmakta olan işlemleri sona erdirerek ve IBM MQ hizmetini durdurarak kuyruk yöneticisini el ile durdurmayı deneyebilirsiniz.

Bu görev hakkında

İpucu: Windows Task Manager (Görev Yöneticisi) ve **tasklist** komutu görevlerle ilgili sınırlı bilgi verir. Hangi süreçlerin belirli bir kuyruk yöneticisiyle ilgili olduğunu belirlemeye yardımcı olacak daha fazla

bilgi için, <http://www.microsoft.com> adresindeki Microsoft web sitesinden yüklenebilecek *Süreç Gezgini* (procepx.exe) gibi bir araç kullanmayı düşünün.

Windows üzerinde bir kuyruk yöneticisini durdurmak için aşağıdaki adımları tamamlayın.

Yordam

1. Windows Task Manager 'ı (Görev Yöneticisi) kullanarak, çalışmakta olan süreçlerin adlarını (tanıtıcıları) listeleysin.
2. Windows Task Manager (Görev Yöneticisi) ya da **taskkill** komutunu kullanarak işlemleri aşağıdaki sırayla (çalışıyorlarsa) sona erdirin:

Çizelge 7. Çalışıyorsa durdurulacak Windows işlem	
İşlem adı	Açıklama
AMQZMUC0	Kritik süreç yöneticisi
AMQZXMA0	Yürütme denetleyicisi
AMQZFUMA	OAM süreci
AMQZLAA0	LQM araçları
AMQZLSA0	LQM araçları
AMQZMUFO	Yardımcı Program Yöneticisi
AMQZMGRO	Süreç denetleyicisi
AMQZMUR0	Yeniden başlatılabilir süreç yöneticisi
AMQFQPUB	Yayınlama Abone Olma süreci
AMQFCXBA	Aracı işçi işlemi
AMQRMPPA	Süreç havuzlama işlemi
AMQCRSTA	İş parçacığı olmayan yanıt veren iş süreci
AMQCRS6B	LU62 alıcı kanalı ve istemci bağlantısı
AMQRRMFA	Havuz işlemi (kümeler için)
AMQPCSEA	Komut sunucusu
RUNMQTRM	Sunucu için tetikleyici izleyicisi çağırma
RUNMQDLQ	Gönderilmeyen ileti kuyruğu işleyicisini çağır
RUNMQCHI	Kanal başlatıcı işlemi
RUNMQLSR	Kanal dinleyici işlemi
AMQXSSVN	Paylaşılan bellek sunucuları

3. IBM MQ hizmetini Windows Denetim Masası 'nda (Control Panel) **Yönetim Araçları > Hizmetler** ' den durdurun.
4. Tüm yöntemleri denediyseniz ve kuyruk yöneticisi durmamışsa, sisteminizi yeniden başlatın.

Linux → AIX **AIX and Linux üzerinde bir kuyruk yöneticisinin el ile durdurulması**

AIX ya da Linux üzerinde bir kuyruk yöneticisini **endmqm** komutunu kullanarak durduramazsanız, çalışmakta olan işlemleri sona erdirerek ve IBM MQ hizmetini durdurarak kuyruk yöneticisini el ile durdurmayı deneyebilirsiniz.

Bu görev hakkında

AIX ya da Linux üzerinde bir kuyruk yöneticisini durdurmak için aşağıdaki adımları izleyin.

Kuyruk yöneticisini el ile durdurursanız, FFST alınabilir ve FDC dosyaları `/var/mqm/errors` içine yerleştirilebilir. Bu, kuyruk yöneticisinde bir hata olarak görülmemelidir.

Kuyruk yöneticisi, el ile durdurma yöntemini kullanarak durdurduktan sonra da olağan bir şekilde yeniden başlatılır.

Yordam

1. **ps** komutunu kullanarak çalışmaya devam eden kuyruk yöneticisi programlarının işlem tanıtıcılarını (PID) bulun.

Örneğin, kuyruk yöneticisinin adı `QMNAME` ise, aşağıdaki komutu kullanın:

```
ps -ef | grep QMNAME
```

2. **ps** komutuyla keşfedilen PID ' leri belirterek, **kill** komutunu kullanarak çalışmaya devam eden kuyruk yöneticisi işlemlerini sona erdirin.

Bir işlemi sona erdirmek için **kill -KILL <pid>** ya da eşdeğer **kill -9 <pid>** komutunu kullanın.

Öldürmek istediğiniz PID ' ler üzerinde tek tek çalışmanız ve her seferinde bu komutu vermeniz gerekir.

Önemli: 9(SIGKILL) dışında herhangi bir sinyal kullanırsanız, işlem büyük olasılıkla durmaz ve beklenmedik sonuçlar elde edeceksiniz.

Süreçleri aşağıdaki sırayla sonlandırın:

<i>Çizelge 8. Çalışıyorsa durdurulacak AIX and Linux işlem</i>	
İşlem adı	Açıklama
amqzmuc0	Kritik süreç yöneticisi
amqzma0	Yürütme denetleyicisi
amqzfuma	OAM süreci
amqzlaa0	LQM araçları
amqzlsa0	LQM araçları
amqzmuf0	Yardımcı Program Yöneticisi
amqzmur0	Yeniden başlatılabilir süreç yöneticisi
amqzmgr0	Süreç denetleyicisi
amqfqpub	Yayınlama Abone Olma süreci
amqfcxba	Aracı işçi işlemi
amqrmppa	Süreç havuzlama işlemi
amqcrsta	İş parçacığı olmayan yanıt veren iş süreci
amqcrs6b	LU62 alıcı kanalı ve istemci bağlantısı
amqrrmfa	Havuz işlemi (kümeler için)
amqpcsea	Komut sunucusu
mqtrm çalıştırma	Sunucu için tetikleyici izleyicisi çağırma
mqdlq çalıştırma	Gönderilmeyen ileti kuyruğu işleyicisini çağır
runmqchi	Kanal başlatıcı işlemi

Çizelge 8. Çalışıyorsa durdurulacak AIX and Linux işlem (devamı var)	
İşlem adı	Açıklama
mqlsr	Kanal dinleyici işlemi

İlgili görevler

“Kuyruk yöneticisinin durdurulması” sayfa 148

Bir kuyruk yöneticisini durdurmak için **endmqm** komutunu kullanabilirsiniz. Bu komut, bir kuyruk yöneticisini durdurmanın dört yolunu sağlar: denetimli ya da susturulmuş, sona erdirme, anında sona erdirme, önleyici kapatma ve bekleme sona erdirme. Diğer bir seçenek olarak, Windows ve Linux sistemlerinde IBM MQ Explorer komutunu kullanarak bir kuyruk yöneticisini durdurabilirsiniz.

Multi Kuyruk yöneticisinin yeniden başlatılması

Bir kuyruk yöneticisini yeniden başlatmak için **strmqm** komutunu kullanabilir ya da Windows ve Linux x86-64 sistemlerinde IBM MQ Explorer adresinden bir kuyruk yöneticisini yeniden başlatabilirsiniz.

Bu görev hakkında

strmqm komutunu kullanarak bir kuyruk yöneticisini yeniden başlatabilirsiniz. **strmqm** komutunun açıklaması ve seçenekleri için bkz. [strmqm](#).

Linux **Windows** Windows ve Linux x86-64 sistemlerinde, bir kuyruk yöneticisini başlatmak için IBM MQ Explorer komutunu kullanarak yeniden başlatabilirsiniz.

Yordam

- **strmqm** komutunu kullanarak bir kuyruk yöneticisini yeniden başlatmak için, komutu ve ardından yeniden başlatmak istediğiniz kuyruk yöneticisinin adını girin.
Örneğin, **strmqm saturn.queue.manager** adlı bir kuyruk yöneticisini başlatmak için şu komutu girin:

```
strmqm saturn.queue.manager
```

- **Linux** **Windows**
IBM MQ Explorer komutunu kullanarak bir kuyruk yöneticisini başlatmak için aşağıdaki adımları izleyin:
 - a) IBM MQ Explorer'ı açın.
 - b) Navigator görünümünde kuyruk yöneticisini seçin.
 - c) **Başlat**'ı tıklayın.

Sonuçlar

Kuyruk yöneticisi yeniden başlatılır.

Kuyruk yöneticisinin yeniden başlatılması birkaç saniyeden uzun sürerse IBM MQ , başlatma işleminin ilerleyişini kesintili olarak ayrıntılı olarak açıklayan bilgi iletileri yayınlar.

Kuyruk yöneticisi özneliklerini görüntüleme ve değiştirme

Bir kuyruk yöneticisine ilişkin kuyruk yöneticisi değiştirgelerini görüntülemek için **DISPLAY QMGR** MQSC komutunu kullanın. Yerel bir kuyruk yöneticisine ilişkin kuyruk yöneticisi değiştirgelerini değiştirmek için **ALTER QMGR** MQSC komutunu kullanın.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınızı platforma göre değiştir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Yordam

- **runmqsc** komutunda belirtilen kuyruk yöneticisinin özniteliklerini görüntülemek için **DISPLAY QMGR** MQSC komutunu kullanın:

```
DISPLAY QMGR
```

Aşağıdaki örnek, bu komutun tipik çıkışını göstermektedir:

```
V9.3.0 V9.3.0 DISPLAY QMGR
1 : DISPLAY QMGR
AMQ8408I: Display Queue Manager details.
QMNAME(QM1)
ACCTINT(1800)
ACCTQ(OFF)
ACTVCONO(DISABLED)
ADVCAP(DISABLED)
ALTTIME(14.24.34)
AUTHOREV(DISABLED)
CERTLABL(ibmwebspheremqmq1)
CHAD(DISABLED)
CHADEXIT( )
CHLAUTH(ENABLED)
CLWLEXIT( )
CLWLMRUC(999999999)
CMDEV(DISABLED)
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE)
CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
CRDATE(2020-12-22)
CUSTOM( )
DEFCLXQ(SCTQ)
DESCR( )
IMGINTVL(60)
IMGRCOVO(YES)
IMGSCHEM(MANUAL)
INITKEY( )
LOCALEV(DISABLED)
MARKINT(5000)
MAXMSGL(4194304)
MAXPRTY(9)
MONACLS(QMGR)
MONQ(OFF)
PERFMEV(DISABLED)
PSMODE(ENABLED)
PSNPMSG(DISCARD)
PSRTYCNT(5)
QMID(QM1_2020-12-22_15.42.49)
REPOS( )
REVDNS(ENABLED)
SCHINIT(QMGR)
SPLCAP(DISABLED)
SSLCRYP( )
SSLFIPS(NO)
SSLKEYR(C:\ProgramData\IBM\MQ\mqgrs\QM1\ssl\key)
SSLRKEYC(32767)
STATCHL(OFF)
STATMQI(OFF)
STRSTPEV(ENABLED)
SYNCPT
TRIGINT(999999999)
XRCAP(NO)
ACCTCONO(DISABLED)
ACCTMQI(OFF)
ACTIVREC(MSG)
ACTVTRC(OFF)
ALTDATE(2022-05-05)
AMQPCAP(NO)
CCSID(437)
CERTVPOL(ANY)
CHADEV(DISABLED)
CHLEV(DISABLED)
CLWLDATA( )
CLWLLEN(100)
CLWLUSEQ(LOCAL)
CMDLEVEL(930)
CONFIGEV(DISABLED)
CRTIME(15.42.49)
DEADQ( )
DEFXMITQ( )
DISTL(YES)
IMGLOGLN(OFF)
IMGRCOVQ(YES)
INHIBTEV(DISABLED)
IPADDRV(IPV4)
LOGGEREV(DISABLED)
MAXHANDS(256)
MAXPROPL(NOLIMIT)
MAXUMSGS(10000)
MONCHL(OFF)
PARENT( )
PLATFORM(WINDOWS10)
PSCLUS(ENABLED)
PSNPRES(NORMAL)
PSSYNCP(IFPER)
REMOEEEV(DISABLED)
REPOSNL( )
ROUTEREC(MSG)
SCMDSERV(QMGR)
SSLCRLNL( )
SSLEV(DISABLED)
KEYRPWD( )
STATACLS(QMGR)
STATINT(1800)
STATQ(OFF)
SUITEB(NONE)
TREELIFE(1800)
VERSION(09030000)
```

Not: SYNCPT, salt okunur bir kuyruk yöneticisi özniteliğidir.

ALL parametresi, **DISPLAY QMGR** komutunda varsayılan değerdir. Tüm kuyruk yöneticisi özniteliklerini görüntüler. Özellikle, çıkış size varsayılan kuyruk yöneticisi adını, teslim-harf kuyruk adını ve komut kuyruğu adını bildirir.

Şu komutu girerek bu kuyrukların var olduğunu doğrulayabilirsiniz:

```
DISPLAY QUEUE (SYSTEM.*)
```

Kök SYSTEM.* ile eşleşen kuyrukların bir listesini görüntüler. Parantezler gereklidir.

- **runmqsc** komutunda belirtilen kuyruk yöneticisinin özniteliklerini değiştirmek için, değiştirmek istediğiniz öznitelikleri ve değerleri belirterek **ALTER QMGRMQSC** komutunu kullanın.

Örneğin, `jupiter.queue.manager` özniteliklerini değiştirmek için aşağıdaki komutları kullanın:

```
runmqsc jupiter.queue.manager  
ALTER QMGR DEADQ (ANOTHERDLQ) INHIBTEV (ENABLED)
```

ALTER QMGR komutu, kullanılan teslim edilmeyen ileti kuyruğunu değiştirir ve olayları engeller.

ALTER QMGR komutunda belirtilmeyen parametreler, bu parametrelere ilişkin var olan değerlerin değiştirilmeden bırakılmasıyla sonuçlanır.

İlgili görevler

[Çoklu platformlarda kuyruk yöneticileri oluşturma](#)

İlgili başvurular

[Kuyruk yöneticisine ilişkin öznitelikler](#)

[runmqsc \(MQSC komutlarını çalıştır\)](#)

[QMGR ' YI GÖRÜNTÜLE](#)

[ALTER QMGR](#)

Multi

Kuyruk yöneticisinin silinmesi

dltmqm denetim komutunu kullanarak bir kuyruk yöneticisini silebilirsiniz. Diğer bir seçenek olarak, Windows ve Linux sistemlerinde bir kuyruk yöneticisini silmek için IBM MQ Explorer kullanılabilir.

Başlamadan önce



Uyarı:

- Kuyruk yöneticisiyle ilişkili tüm kaynakları (tüm kuyruklar, iletileri ve tüm nesne tanımlamaları da içinde olmak üzere) sildiğiniz için kuyruk yöneticisinin silinmesi önemli bir adımdır. **dltmqm** denetim komutunu kullanırsanız, fikrinizi değiştirmenize izin veren bir bilgi istemi görüntülenmez; Enter tuşuna bastığınızda, ilişkili tüm kaynaklar kaybolur.
- **Windows** Windows işletim sistemi üzerinde, bir kuyruk yöneticisinin silinmesi, kuyruk yöneticisini otomatik başlatma listesinden de kaldırır (açıklamalar için bkz. “Kuyruk yöneticisinin başlatılması” sayfa 147). Komut tamamlandığında bir IBM MQ `queue manager ending` iletisi görüntülenir; kuyruk yöneticisinin silindiği size söylenmez.
- Bir küme kuyruk yöneticisinin silinmesi kümeden kaldırılmasına gerek yoktur. Daha fazla bilgi için [dltmqm](#) içindeki kullanım notlarına bakın.

Bu görev hakkında

dltmqm denetim komutunu kullanarak bir kuyruk yöneticisini silebilirsiniz. **dltmqm** komutuna ve seçeneklerine ilişkin açıklamalar için [dltmqm](#) başlıklı konuya bakın. Yalnızca güvenilir yöneticilerin bu komutu kullanma yetkisine sahip olduğundan emin olun. (Güvenlikle ilgili bilgi için bkz. [AIX, Linux, and Windows](#) üzerinde güvenliğinin ayarlanması.)

Linux

Windows

Alternatif olarak, Windows ve Linux (x86 ve x86-64 platformları) sistemlerinde IBM MQ Explorer komutunu kullanarak bir kuyruk yöneticisini silebilirsiniz.

Yordam

- **dltmqm** komutunu kullanarak bir kuyruk yöneticisini silmek için aşağıdaki adımları izleyin:
 - a) Kuyruk yöneticisini durdurun.
 - b) Şu komutu verin:

```
dltmqm QMB
```

Not: Çalıştığınız kuyruk yöneticisiyle ilişkili kuruluşta **dltmqm** komutunu kullanmanız gerekir. `dspmqr -o installation` komutunu kullanarak, bir kuyruk yöneticisinin hangi kuruluşla ilişkilendirildiği öğrenebilirsiniz.

-  

Bir kuyruk yöneticisini IBM MQ Explorer komutunu kullanarak silmek için aşağıdaki adımları izleyin:

- a) IBM MQ Explorer'ni açın.
- b) Navigator görünümünde kuyruk yöneticisini seçin.
- c) Kuyruk yöneticisi durdurulmazsa durdurun.
Kuyruk yöneticisini durdurmak için, kuyruk yöneticisini farenin sağ düğmesiyle tıklatın ve **Durdur** öğesini seçin.
- d) Kuyruk yöneticisini silin.
Kuyruk yöneticisini silmek için, kuyruk yöneticisini farenin sağ düğmesiyle tıklatın ve **Sil** i seçin.

Sonuçlar

Kuyruk yöneticisi silindi.

MQI kanalları durduruluyor

Bir sunucu bağlantısı kanalı için STOP CHANNEL komutu verdiğinizde, istemci-bağlantı kanalını durdurmak için hangi yöntemin kullanılacağını seçebilirsiniz. Bu, MQGET bekleme çağrısı yayınlayan bir istemci kanalının denetlenebileceği ve kanalın nasıl ve ne zaman durdurulacağına karar verebileceğiniz anlamına gelir.

STOP CHANNEL komutu, kanalın nasıl durdurulacağını gösteren üç kiple verilebilir:

Susturma

Yürürlükteki iletiler işlendikten sonra kanalı durdurur.

Paylaşım sohbetleri etkinleştirilirse, IBM MQ MQI client durdurma isteğinin zamanında farkına varır; bu kez ağın hızına bağlıdır. İstemci uygulaması, sonraki bir IBM MQ çağrısı yayınlanması sonucunda durdurma isteğinin farkında olur.

Zorla

Kanalı hemen durdurur.

Sonlandır

Kanalı hemen durdurur. Kanal bir işlem olarak çalışıyorsa, kanalın işlemini sonlandırabilir ya da kanal bir iş parçacığı olarak çalışıyorsa, onun iş parçacığı olarak çalışabilir.

Bu çok aşamalı bir süreçtir. Kip sonlandırma kullanılırsa, sunucu bağlantısı kanalını önce kip susturma, sonra kip zorlama ve gerekirse kip sonlandırma ile durdurma girişiminde bulunulacaktır. Müşteri, farklı sona erdirme aşamaları sırasında farklı dönüş kodları alabilir. İşlem ya da iş parçacığı sonlandırılırsa, istemci bir iletişim hatası alır.

Uygulamaya döndürülen dönüş kodları, yayınlanan MQI çağrısına ve STOP CHANNEL komutuna göre değişir. İstemci bir MQR_CONNECTION_QUIESCING ya da MQR_CONNECTION_BROKEN dönüş kodunu alacak. Bir istemci MQR_CONNECTION_QUIESCING değerini saptarsa, yürürlükteki hareketi tamamlayıp sonlandırmayı denemelidir. MQR_CONNECTION_BROKEN ile bu yapılamaz. İstemci işlemi tamamlamazsa ve yeterince hızlı sonlandırmazsa, birkaç saniye sonra CONNECTION_BROKEN olur. MODE (FORCE) ya da MODE (TERMINATE) içeren bir STOP CHANNEL komutu, MODE (QUIESCE) komutundan çok, CONNECTION_BROKEN komutuyla sonuçlanabilir.

İlgili kavramlar

[Kanallar](#)

Yerel kuyruklarla çalışma

Bu kısım, yerel, model ve diğer ad kuyruklarını yönetmek için kullanabileceğiniz bazı MQSC komutlarına ilişkin örnekler içerir.

Bu komutlarla ilgili ayrıntılı bilgi için [MQSC komutları](#) konusuna bakın.

İlgili başvurular

[Kuyruklar için adlandırma kısıtlamaları](#)

[Diğer nesnelere ilişkin adlandırma kısıtlamaları](#)

DEFINE QLOCAL ile yerel kuyruk tanımlanması

Bir uygulama için, yerel kuyruk yöneticisi, uygulamanın bağlı olduğu kuyruk yöneticisidir. Yerel kuyruk yöneticisi tarafından yönetilen kuyrukların, o kuyruk yöneticisi için yerel olduğu söyler. Yerel kuyruk yaratmak için **DEFINE QLOCAL** MQSC komutunu kullanın.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu görev hakkında

Yerel kuyruk yaratmak için **DEFINE QLOCAL** MQSC komutunu kullanın. Varsayılan yerel kuyruk tanımında tanımlanan varsayılan değeri kullanabilir ya da kuyruk özelliklerini varsayılan yerel kuyruktaki özelliklerden değiştirebilirsiniz.

Not: Varsayılan yerel kuyruk SYSTEM.DEFAULT.LOCAL.QUEUE ve sistem kuruluşunda oluşturulur.

Yordam

- Yerel kuyruk yaratmak için, aşağıdaki örnekte gösterildiği gibi **DEFINE QLOCAL** komutunu girin. Bu örnekte, **DEFINE QLOCAL** komutu ORANGE.LOCAL.QUEUE (KUYRUK):
 - Bu olanak, önceliğe dayalı bir sipariş temelinde almak, vermek için etkinleştirilir ve çalışır.
 - Bu bir *normal* kuyruğudur; başlatma kuyruğu ya da iletim kuyruğu değildir ve tetikleyici iletileri oluşturmaz.
 - Kuyruk derinliği üst sınırı 5000 iletidir; ileti uzunluğu üst sınırı 4194304 bayttir.

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) +
  DESCR('Queue for messages from other systems') +
  PUT (ENABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (PRIORITY) +
  MAXDEPTH (5000) +
  MAXMSGL (4194304) +
  USAGE (NORMAL)
```

Notlar:

1. Açıklama değeri dışında, örnekte gösterilen tüm öznitelik değerleri varsayılan değerlerdir. Bu örnekler, illüstrasyon amacıyla eklenmiştir. Varsayılan değerlerin istediğiniz gibi olduğundan ya da değiştirilmediğinden eminseniz bunları atlayabilirsiniz. Ayrıca bkz. [“DISPLAY QUEUE ile varsayılan nesne özniteliklerini görüntüleme”](#) sayfa 159.
2. **USAGE (NORMAL)** , bu kuyruğun bir iletim kuyruğu olmadığını gösterir.

3. Aynı kuyruk yöneticisinde ORANGE.LOCAL.QUEUE, bu komut başarısız olur. Bir kuyruğun var olan tanımının üzerine yazılmasını istiyorsanız **REPLACE** özniteliğini kullanın, ancak ayrıca bkz. "[ALTER QLOCAL ya da DEFINE QLOCAL ile yerel kuyruk özniteliklerinin değiştirilmesi](#)" sayfa 160.

İlgili başvurular

[QLOCAL TANIMLAYIN](#)

DISPLAY QUEUE ile varsayılan nesne özniteliklerini görüntüleme

Bir IBM MQ nesnesi tanımlandığında varsayılan nesneden alınan öznitelikleri görüntülemek için **DISPLAY QUEUE** MQSC komutunu kullanabilirsiniz.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu görev hakkında

Bir IBM MQ nesnesi tanımladığınızda, varsayılan nesneden belirtmediğiniz öznitelikleri alır. Örneğin, bir yerel kuyruk tanımladığınızda, kuyruk tanımlamada atladığınız öznitelikleri SYSTEM.DEFAULT.LOCAL.QUEUE. Bu özniteliklerin tam olarak ne olduğunu görmek için **DISPLAY QUEUE** komutunu kullanabilirsiniz.

Yordam

- Yerel bir kuyruğa ilişkin varsayılan nesne özniteliklerini görüntülemek için aşağıdaki komutu kullanın:

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

DISPLAY komutunun sözdizimi, ilgili **DEFINE** komutunun sözdiziminden farklıdır. **DISPLAY** komutunda yalnızca kuyruk adını verebilirsiniz; **DEFINE** komutunda, kuyruğun tipini (QLOCAL, QALIAS, QMODEL ya da QREMOTE) belirtmeniz gerekir.

Öznitelikleri tek tek belirterek seçmeli olarak görüntüleyebilirsiniz. Örneğin:

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +  
MAXDEPTH +  
MAXMSGL +  
CURDEPTH;
```

Bu komut, belirtilen üç özniteliği aşağıdaki gibi görüntüler:

```
AMQ8409: Display Queue details.  
QUEUE (ORANGE.LOCAL.QUEUE)          TYPE (QLOCAL)  
CURDEPTH (0)                          MAXDEPTH (5000)  
MAXMSGL (4194304)
```

CURDEPTH, kuyruktaki iletilerin sayısı olan yürürlükteki kuyruk derinliğidir. Kuyruk derinliğini izleyerek kuyruğun dolmadığından emin olabileceğiniz için, bu, görüntülenecek yararlı bir özniteliktir.

İlgili başvurular

[KUYRUK GÖRÜNTÜLE](#)

[DEFINE kuyrukları](#)

DEFINE QLOCAL ile yerel bir kuyruk tanımlamasının kopyalanması

DEFINE QLOCAL MQSC komutundaki **LIKE** özniteliğini kullanarak bir kuyruk tanımlamasını kopyalayabilirsiniz.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu görev hakkında

Sistem varsayılan yerel kuyruğuyla değil, belirtilen kuyrukla aynı özniteliklere sahip bir kuyruk yaratmak için **LIKE** özniteliğiyle **DEFINE** komutunu kullanabilirsiniz. Bir kuyruk tanımlamasını kopyalamak için **DEFINE** komutunun bu biçimini de kullanabilirsiniz, ancak özgün öğenin özniteliklerinde bir ya da daha çok değişiklik yapabilirsiniz.

Notlar:

1. Bir **DEFINE** komutunda **LIKE** özniteliğini kullandığınızda, yalnızca kuyruk özniteliklerini kopyalayabilirsiniz. Kuyruktaki iletileri kopyalamıyorsunuz.
2. **LIKE** belirtilmeden bir yerel kuyruk tanımlarsanız, bu kuyruk aşağıdaki ile aynıdır:

```
DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Yordam

- Sistemin varsayılan yerel kuyruğu yerine, belirtilen kuyrukla aynı özniteliklere sahip bir kuyruk yaratmak için aşağıdaki örnekte gösterildiği gibi **DEFINE** komutunu girin. Kuyruğu yaratırken tam olarak girildiği gibi kopyalanacak kuyruğun adını girin. Ad küçük harfli karakterler içeriyorsa, adı tek tırnak içine alın.

Bu örnek, ORANGE.LOCAL.QUEUE, sistemin varsayılan yerel kuyruğu yerine:

```
DEFINE QLOCAL (MAGENTA.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE)
```

- Bir kuyruk tanımlamasını kopyalamak, ancak bir ya da daha çok değişikliği özgün özniteliğin yerine koymak için aşağıdaki örnekte gösterildiği gibi **DEFINE** komutunu girin. Bu komut, ORANGE.LOCAL.QUEUE to the queue THIRD.QUEUE, ancak yeni kuyruktaki ileti uzunluğu üst sınırının 4194304 yerine 1024 bayt olacağını belirtir:

```
DEFINE QLOCAL (THIRD.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE) +  
MAXMSGL(1024);
```

İlgili başvurular

[DEFINE kuyrukları](#)

ALTER QLOCAL ya da DEFINE QLOCAL ile yerel kuyruk özniteliklerinin değiştirilmesi

ALTER QLOCAL ya da **DEFINE QLOCAL** MQSC komutunu **REPLACE** özniteliğiyle kullanarak kuyruk özniteliklerini iki şekilde değiştirebilirsiniz.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu görev hakkında

Var olan bir tanımlamayı belirtilen yeni tanımlamayla değiştirmek için **ALTER** ve **DEFINE** komutlarından birinin **REPLACE** özniteliğini kullanabilirsiniz. **ALTER** ile **DEFINE** arasındaki fark, **REPLACE** ile **ALTER**

ögesinin belirlenmemiş parametreleri değiştirmemesi, ancak **REPLACE** ile **DEFINE** ile tüm parametreleri ayarlamasıdır.

Yordam

- Kuyruk özniteliklerini değiştirmek için, aşağıdaki örneklerde gösterildiği gibi **ALTER** komutunu ya da **DEFINE** komutunu kullanın.

Bu örneklerde, ORANGE.LOCAL.QUEUE , 10.000 bayta düşürüldü.

- **ALTER** komutunu kullanarak:

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

Bu komut, ileti uzunluğu üst sınırı olan tek bir özniteliği değiştirir; diğer tüm öznitelikler aynı kalır.

- **DEFINE** komutunu **REPLACE** seçeneğiyle kullanarak, örneğin:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

Bu komut yalnızca ileti uzunluğu üst sınırını değil, varsayılan değerleri verilen diğer tüm öznitelikleri de değiştirir. Örneğin, kuyruk önceden engellenmişse, SYSTEM.DEFAULT.LOCAL.QUEUE.

Varolan bir kuyruktaki ileti uzunluğu üst sınırını azaltırsanız, varolan iletiler bundan etkilenmez. Ancak yeni iletiler, yeni ölçütleri karşılamalıdır.

İlgili başvurular

[ALTER kuyrukları](#)

[ALTER QLOCAL \(YEREL\)](#)

[DEFINE kuyrukları](#)

[QLOCAL TANIMLAYIN](#)

CLEAR QLOCAL ile yerel bir kuyruğun temizlenmesi

Yerel bir kuyruğu temizlemek için **CLEAR QLOCAL MQSC** komutunu kullanabilirsiniz.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Aşağıdaki durumda bir kuyruğu temizleyemezsiniz:

- Kuyruğa eşitleme noktası altında konan kesinleştirilmemiş iletiler var.
- Bir uygulamanın kuyruğu açık.

Bu görev hakkında

CLEAR QLOCAL komutunu kullanarak bir yerel kuyruğu temizlemek istiyorsanız, kuyruğun adı yerel kuyruk yöneticisine tanımlanmalıdır.

Not: Fikrinizi değiştirmenizi sağlayan bir bilgi istemi yoktur; Enter tuşuna bastığınızda iletiler kaybolur.

Yordam

İletileri yerel bir kuyruktan temizlemek için aşağıdaki örnekte gösterildiği gibi **CLEAR QLOCAL** kullanın.

Bu örnekte, tüm iletiler MAGENTA.QUEUE:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

İlgili başvurular

[QLOCAL TEMİZLE](#)

DELETE QLOCAL ile yerel kuyruk silinmesi

Yerel bir kuyruğu silmek için **DELETE QLOCAL** MQSC komutunu kullanabilirsiniz.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Kesinleştirilmemiş iletiler içeren bir kuyruk silinemez.

Bir kuyrupta bir ya da daha çok kesinleştirilmiş ileti varsa ve kesinleştirilmemiş ileti yoksa, yalnızca **PURGE** seçeneğini belirttiğinizde bu ileti silinebilir. Daha sonra, belirtilen kuyrupta kesinleştirilmiş iletiler olsa da silme işlemi devam eder ve bu iletiler de temizlenir.

PURGE yerine **NOPURGE** değerinin belirlenmesi, kuyruğun kesinleştirilmiş ileti içermesi durumunda silinmemesini sağlar.

Yordam

- Bir yerel kuyruğu silmek için aşağıdaki örnekte gösterildiği gibi **DELETE QLOCAL** komutunu kullanın. Bu örnek, PINK.QUEUE :

```
DELETE QLOCAL (PINK.QUEUE) NOPURGE
```

Bu örnek, PINK.QUEUE QUEUE:

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

İlgili başvurular

[QLOCAL ÖĞESİNİ SIL](#)

Örnek programla kuyruklara göz atılması

IBM MQ , bir kuyruktaki iletilerin içeriğine bakmak için kullanabileceğiniz örnek bir kuyruk tarayıcısı sağlar.

Bu görev hakkında

Tarayıcı, aşağıdaki konumlarda hem kaynak hem de yürütülebilir biçimlerde sağlanır; burada *MQ_INSTALLATION_PATH* , IBM MQ ' in kurulu olduğu üst düzey dizini temsil eder.

Windows Windows üzerinde, örnek kuyruk tarayıcısına ilişkin dosya adları ve yollar aşağıdaki gibidir:

Kaynak

MQ_INSTALLATION_PATH\tools\c\samples\

Yürütülebilir

MQ_INSTALLATION_PATH\tools\c\samples\bin\amqsbcg.exe

Linux

AIX

AIX and Linux' de dosya adları ve yolları aşağıdaki gibidir:

Kaynak

MQ_INSTALLATION_PATH/samp/amqsbcg0.c

Yürütülebilir

MQ_INSTALLATION_PATH/samp/bin/amqsbcg

Yordam

- Örnek programı çalıştırmak için, aşağıdaki örnekte gösterildiği gibi bir komut girin.
Örnek program için iki giriş parametresi gerekir; iletilere göz atılacak kuyruğun adı ve bu kuyruğun iyisi olan kuyruk yöneticisi. Örneğin:

```
amqsbcg SYSTEM.ADMIN.QMGREVENT.tpp01 saturn.queue.manager
```

Sonuçlar

Bu komutun tipik sonuçları aşağıdaki örnekte gösterilmiştir:

```
AMQSBCG0 - starts here
*****

MQOPEN - 'SYSTEM.ADMIN.QMGR.EVENT'

MQGET of message number 1
****Message descriptor****

  StrucId   : 'MD   '  Version : 2
  Report    : 0  MsgType : 8
  Expiry    : -1  Feedback : 0
  Encoding  : 546  CodedCharSetId : 850
  Format     : 'MQEVENT '
  Priority   : 0  Persistence : 0
  MsgId     : X'414D512073617475726E2E71756575650005D30033563DB8'
  CorrelId  : X'0000000000000000000000000000000000000000000000000000'
  BackoutCount : 0
  ReplyToQ  : '
  ReplyToQMgr : 'saturn.queue.manager'
  ** Identity Context
  UserIdentifier : '
  AccountingToken :
  X'0000000000000000000000000000000000000000000000000000000000000000'
  ApplIdentityData : '
  ** Origin Context
  PutApplType : '7'
  PutApplName : 'saturn.queue.manager'
  PutDate : '19970417'  PutTime : '15115208'
  ApplOriginData : '

  GroupId : X'0000000000000000000000000000000000000000000000000000000000000000'
  MsgSeqNumber : '1'
  Offset : '0'
  MsgFlags : '0'
  OriginalLength : '104'

**** Message ****

length - 104 bytes

00000000: 0700 0000 2400 0000 0100 0000 2C00 0000 | .....→.....|
00000010: 0100 0000 0100 0000 0100 0000 AE08 0000 | .....|
00000020: 0100 0000 0400 0000 4400 0000 DF07 0000 | .....D.....|
00000030: 0000 0000 3000 0000 7361 7475 726E 2E71 | ...0...saturn.q'|
00000040: 7565 7565 2E6D 616E 6167 6572 2020 2020 | ueue.manager'|
00000050: 2020 2020 2020 2020 2020 2020 2020 2020 | .....|
00000060: 2020 2020 2020 2020

No more messages
MQCLOSE
MQDISC
```

İlgili başvurular

[Tarayıcı örnek programı](#)

Büyük kuyrukların etkinleştirilmesi

IBM MQ , 2 TB ' den büyük kuyrukları destekler.

Windows Windows sistemlerinde, büyük dosyalar için destek herhangi bir ek etkinleştirme olmadan sağlanır.

Linux **AIX** AIX and Linux sistemlerinde, birden çok gigabayt ya da terabaytlık kuyruk dosyaları oluşturmadan önce büyük dosya desteğini belirttik olarak etkinleştirmeniz gerekir. Bunun nasıl yapılacağını öğrenmek için işletim sisteminizin belgelerine bakın.

Tar gibi bazı yardımcı programlar, birden çok gigabayt ya da terabayt dosyalarıyla başa çıkamaz. Büyük dosya desteğini etkinleştirmeden önce, kullandığınız yardımcı programlara ilişkin kısıtlamalar için işletim sistemi belgelerinize bakın.

Kuyruklar için gereksinim duyduğunuz depolama miktarını planlama hakkında bilgi için platforma özgü performans raporları için [MQ Performans belgeleri](#) başlıklı konuya bakın.

IBM MQ 9.1.5' den, yerel ve model kuyruklarında yeni bir öznitelik kullanarak kuyruk dosyalarının boyutunu denetleyebilirsiniz. Ek bilgi için bkz. [“IBM MQ kuyruk dosyalarını değiştirme” sayfa 164](#) .

Multi IBM MQ kuyruk dosyalarını değiştirme

IBM MQ 9.2.0' dan, yerel ve model kuyruklarındaki bir özniteliği kullanarak kuyruk dosyalarının boyutunu denetleyebilirsiniz. İki kuyruk durumu özniteliğini kullanarak, bir kuyruk dosyasının yürürlükteki büyüklüğünü ve şu anda o dosyada kullanılmakta olan blok boyutuna dayalı olarak büyüyebileceği büyüklük üst sınırını görüntüleyebilirsiniz.

Kuyruk dosyalarını değiştirmek için kullanılan öznitelik

Yerel ve model kuyruklarındaki öznitelik:

MAXFSIZE

Kuyruk tarafından kullanılan kuyruk dosyası büyüklüğü üst sınırını megabayt cinsinden belirtir.

MQSC komutlarını, IBM MQ Explorerya da administrative REST API kullanarak bu özniteliğin değerini ayarlayabilir ya da görüntüleyebilirsiniz. IBM MQ Console'inde bu özniteliğin değerini de görüntüleyebilirsiniz.

Ek bilgi için bkz. [MAXFSIZE](#) ve [“IBM MQ kuyruk dosyası boyutunun değiştirilmesi” sayfa 165](#) .

Bu özniteliğin PCF eşdeğeri: **MQIA_MAX_Q_FILE_SIZE**. Bkz. [Kuyruğun Değiştirilmesi, Kopyalanması ve Yaratılması](#).

Kuyruk durumundaki iki öznitelik şunlardır:

YÜRÜRLÜK

Kuyruk dosyasının yürürlükteki büyüklüğünü megabayt cinsinden, en yakın megabayta yuvarlanmış olarak görüntüler.

MQSC komutlarını, IBM MQ Explorerya da administrative REST API kullanarak bu özniteliğin değerini ayarlayabilir ya da görüntüleyebilirsiniz.

Ek bilgi için bkz. [CURFSIZE](#) .

Bu özniteliğin PCF eşdeğeri: **MQIA_CUR_Q_FILE_SIZE**. Bkz. [Soruyla Kuyruk ve Soruyla Kuyruk \(yanıt\)](#).

CURMAXFS

Bir kuyrukta kullanılan yürürlükteki öbek büyüklüğü göz önüne alındığında, kuyruk dosyasının büyüyebileceği büyüklük üst sınırını en yakın megabayta yuvarlanmış olarak gösterir.

MQSC komutlarını, IBM MQ Explorerya da administrative REST API kullanarak bu özniteliğin değerini ayarlayabilir ya da görüntüleyebilirsiniz.

Ek bilgi için bkz. [CURMAXFS](#) .

Bu özniteliğin PCF eşdeğeri: **MQIA_CUR_MAX_FILE_SIZE**. Bkz. [Soruyla Kuyruk ve Soruyla Kuyruk \(yanıt\)](#).

Blok boyutu ve ayrıntı düzeyi

Kuyruk dosyaları, blok adı verilen bölümlere bölünür. Bir kuyruk dosyası büyüklüğü üst sınırını artırmak için, kuyruk yöneticisi tarafından kuyruğun öbek büyüklüğünün ya da ayrıntı düzeyinin değiştirilmesi gerekebilir.

Yeni tanımlanan bir kuyruk büyük bir **MAXFSIZE** değeriyle yaratıldıysa, kuyruk uygun bir blok büyüklüğüyle yaratılır. Ancak, var olan bir kuyrukte **MAXFSIZE** değeri arttıysa (örneğin, **ALTER QLOCAL** MQSC komutu kullanılarak), kuyruk yöneticisinin kuyruğu yeniden yapılandırması için kuyruğun boşaltılmasına izin verilmesi gerekebilir.

Ek bilgi için bkz. [“Bir IBM MQ kuyruk dosyasının ne kadar veri saklayabileceğini hesaplama” sayfa 166](#) .



Uyarı: Bazı dosya sistemleri ve işletim sistemleri, tüm dosya sisteminin boyutuna ya da tek bir dosyanın boyutuna ilişkin sınırlara sahiptir. İşletmenizin kullandığı sistemlerin sınırlarını kontrol etmelisiniz.

İlgili başvurular

[KUYRUKLARIN](#)

[KUYRUK GÖRÜNTÜLE](#)

[QSTATUS DURUMUNU GÖRÜNTÜLE](#)

Multi

IBM MQ kuyruk dosyası boyutunun değiştirilmesi

Bir kuyruk dosyasının büyüklük üst sınırını artırabilir ya da azaltabilirsiniz.

Başlamadan önce

Not: Bu görevdeki adımlar MQSC komutlarını çalıştırmanızı gerektirir:

- **ALW** AIX, Linux, and Windows üzerinde, bir **runmqsc** komut isteminden MQSC komutları yayınlayın. Bkz. [runmqsc altında MQSC komutlarının etkileşimli olarak çalıştırılması ve runmqsc altındaki metin dosyalarından MQSC komutlarının çalıştırılması](#).
- **IBM i** IBM üzerinde, Komut Dosyası dosyasında bir komut listesi oluşturun ve dosyayı **STRMQMMQSC** komutunu kullanarak çalıştırın. Bkz. [Administration using MQSC commands on IBM i](#).

Bir kuyruk dosyası için yeni bir boyut ayarlamadan önce, değiştirmek istediğiniz kuyruk dosyasının boyutunu görmek için [DISPLAY QLOCAL](#) MQSC komutunu kullanın. Örneğin, şu komutu verin:

```
DISPLAY QLOCAL (SYSTEM.DEFAULT.LOCAL.QUEUE) MAXFSIZE
```

Aşağıdaki çıkışı alırsınız:

```
AMQ8409I: Display queue details
          QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)          TYPE (QLOCAL)
          MAXFSIZE (DEFAULT)
```

bu, kuyruk dosyası boyutu üst sınırının 2.088.960 MB varsayılan değeri olduğunu gösterir.

Bu görev hakkında

Aşağıdaki yordamlar aşağıdakileri nasıl gerçekleştirdiğinizi göstermektedir:

- Bir kuyruk dosyasının büyüyebileceği büyüklük üst sınırını azaltın.
- Bir kuyruk dosyasının büyüyebileceği büyüklük üst sınırını artırın.



Uyarı: Uygulamalarınızın yazma şeklini ve başarımlar üzerindeki olası etkisini dikkate almadan kuyruk dosyalarının boyutunu artırmaya dikkat etmelisiniz. Çok büyük bir kuyruk dosyasında rasgele iletilere erişim çok yavaş olabilir.

Bir kuyruk dosyası boyutu üst sınırını varsayılandan fazla artırmayı düşünüyorsanız, ilinti tanıtıcıları ve IBM MQ classes for JMS **JM 3.0** ya da IBM MQ classes for Jakarta Messaging seçici dizgileri gibi ileti seçicileri kullanmakta dikkatli olun. Daha büyük kuyruk dosyaları, kuyruğa ilk giren ilk çıkar erişimi için daha uygundur.

Tek tek kuyruk dosyalarında çok büyük miktarlarda veri olması, yalnızca çevrimsel günlüğe kaydetme için yapılandırılmış kuyruk yöneticisinde ya da tek tek kuyruk için ortam görüntülemenin etkinleştirilmediği yerlerde gerçekleştirilmelidir.

Kuyruk yöneticisinin çalışmasını etkileyebileceğinden, SYSTEM kuyruklarının büyüklüğünü sınırlamamalısınız.

Yordam

1. Kuyruk dosyası büyüklüğü üst sınırını azalt

- 500 gigabayt büyüklüğünde SMALLQUEUEadlı yerel bir dosya yaratmak için aşağıdaki MQSC komutunu verin:

```
DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
2 : DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
AMQ8006I: IBM MQ queue created
```

ve şu iletiyi alırsınız: AMQ8006I:

Not: Dosyada bulunan veri miktarından daha küçük bir değer içeren bir kuyruk yapılandırırsanız, yeni iletiler kuyruğa konamaz.

Bir uygulama yeterli yeri olmayan bir kuyruk dosyasına ileti yerleştirmeyi denerse, uygulama MQRC_Q_SPACE_NOT_ALLOWED dönüş kodunu alır. Kuyruktan yeterince ileti yıkıcı bir şekilde okunduğunda, uygulamalar kuyruğa yeni iletiler yerleştirmeye başlayabilir.

2. Kuyruk dosyası büyüklüğü üst sınırını artırın.

- 5 terabayt büyüklüğünde LARGEQUEUEadlı yerel bir dosya yaratmak için aşağıdaki MQSC komutunu verin:

```
DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
3 : DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
AMQ8006I: IBM MQ queue created
```

Multi

Bir IBM MQ kuyruk dosyasının ne kadar veri saklayabileceğini hesaplama

Bir kuyrukte saklanabilecek veri miktarı, kuyruğun bölündüğü tek tek blokların büyüklüğüyle sınırlıdır. MQSC komutlarını kullanarak öbek büyüklüğünü ve öge boyunu doğrulayın ve bir kuyruk dosyasının büyüklüğünü denetleyin.

Başlamadan önce

Not: Bu görevdeki adımlar MQSC komutlarını çalıştırmanızı gerektirir:

- ALW** AIX, Linux, and Windowsüzerinde, bir **runmqsc** komut isteminden MQSC komutları yayınlayın. Bkz. runmqsc altında MQSC komutlarının etkileşimli olarak çalıştırılması ve runmqsc altındaki metin dosyalarından MQSC komutlarının çalıştırılması.
- IBM i** IBM üzerinde, Komut Dosyası dosyasında bir komut listesi oluşturun ve dosyayı **STRMQMMQSC** komutunu kullanarak çalıştırın. Bkz. Administration using MQSC commands on IBM i.

Yordam

- Blok boyutunu ve ayrıntı düzeyini onaylayın.

Varsayılan blok boyutu 512 bayttır. İki terabayttan büyük kuyruk dosyalarını desteklemek için kuyruk yöneticisinin blok boyutunu artırması gerekir.

MAXFSIZE ürününü bir kuyruk için yapılandırdığınızda blok boyutu otomatik olarak hesaplanır, ancak düzeltilen blok boyutu, kuyrukta önceden ileti varsa kuyruğa uygulanamaz. Bir kuyruk boşaldığında, kuyruk yöneticisi yapılandırılan **MAXFSIZE** ögesini desteklemek için blok boyutunu otomatik olarak değiştirir.

DISPLAY QSTATUS komutunun **CURMAXFS** adlı yeni bir özneliği vardır; bu öznelik, bir kuyruğun yeni bir blok boyutunu kullanacak şekilde değiştirildiğini doğrulamanızı sağlar.

Aşağıdaki örnekte, **CURMAXFS** değeri olan 4177920, kuyruk dosyasının şu anda yaklaşık dört terabayt boyutuna büyülebileceğini doğrular. Kuyrukta yapılandırılan **MAXFSIZE** değeri **CURMAXFS** değerinden büyükse, kuyruk yöneticisi kuyruk dosyasının blok boyutunu yeniden yapılandırmadan önce kuyruğun boşaltılmasını bekler.

```
DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
  2 : DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
AMQ8450I: Display queue status details
  QUEUE(LARGEQUEUE)                TYPE(Queue)
  CURMAXFS(4177920)                CURDEPTH(100000)
```

- Bir kuyruk dosyasının boyutunu denetleyin.

DISPLAY QSTATUS komutundaki **CURFSIZE** özneliğini kullanarak, disk üzerindeki bir kuyruk dosyasının yürürlükteki boyutunu megabayt cinsinden görüntüleyebilirsiniz. Bu, dosya sistemine doğrudan erişimin mümkün olmadığı IBM MQ Appliance gibi platformlarda yararlı olabilir.

```
DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
  1 : DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
AMQ8450I: Display queue status details
  QUEUE(SMALLQUEUE)                TYPE(Queue)
  CURDEPTH(4024)                   CURFSIZE(10)
```

Not: Bir kuyruktan kaldırılan iletiler olduğunda, **CURFSIZE** özneliği hemen azalmaz.

Genellikle, bir kuyruk dosyasındaki alan yalnızca, kuyrukta açık bir uygulama olmadığında ve kuyrukta saklanan belirsiz iletiler olmadığında serbest bırakılır. Kuyruk yöneticisi tarafından yüklenen bir kuyruk dosyasının gerekli olarak kesilmesi ya da sıkıştırılması, denetim noktası, kuyruk yöneticisinin kapatılması ya da kuyruğun ortam görüntüsünün kaydedilmesi sırasında oluşur.

İlgili başvurular

[KUYRUKLARIN](#)

[QSTATUS DURUMUNU GÖRÜNTÜLE](#)

Uzak kuyruklarla çalışılması

Uzak kuyruk, uzak kuyruk yöneticisindeki bir kuyruğa gönderme yapan yerel kuyruk yöneticisindeki bir tanımdır. Uzak kuyruğu yerel bir konumdan tanımlamanıza gerek yoktur; ancak, uzak kuyruğu yerel olarak tanımlarsanız, uzak kuyruğun bulunduğu kuyruk yöneticisinin tanıtıcısıyla nitelenmiş bir ad belirtmek yerine, uzak kuyruğa yerel olarak tanımlanmış adıyla gönderme yapabilir.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu görev hakkında

Bir uygulama yerel bir kuyruk yöneticisine bağlanır ve bir MQOPEN çağrısı verir. Açık çağrıda, belirlenen kuyruk adı, yerel kuyruk yöneticisindeki uzak kuyruk tanımına ilişkin. Uzak kuyruk tanımı, hedef kuyruğun adlarını, hedef kuyruk yöneticisini ve isteğe bağlı olarak bir iletim kuyruğunu sağlar. Uzak kuyruğa bir

ileti koymak için uygulama, MQOPEN çağrısından döndürülen tanıtıcıyı belirterek bir MQPUT çağrısı yayınlar. Kuyruk yöneticisi, iletinin başındaki bir iletim üstbilgisinde uzak kuyruk adını ve uzak kuyruk yöneticisi adını kullanır. Bu bilgiler, iletiyi ağdaki doğru hedefine yönlendirmek için kullanılır.

Denetimci olarak, uzak kuyruk tanımlamasını değiştirerek iletinin hedefini denetleyebilirsiniz.

Yordam

- Uzak bir kuyruk yöneticisinin iyeliğindeki bir kuyruğa ileti koyun.

Uygulama bir kuyruk yöneticisine bağlanır; örneğin, saturn.queue.manager. Hedef kuyruk başka bir kuyruk yöneticisine ait.

MQOPEN çağrısında uygulama şu alanları belirtir:

Alan değeri	Açıklama
<i>ObjectName</i> CYAN.REMOTE.QUEUE	Uzak kuyruk nesnesinin yerel adını belirler. Hedef kuyruğu ve hedef kuyruk yöneticisini tanımlar.
<i>ObjectType</i> (Kuyruk)	Bu nesneyi bir kuyruk olarak tanımlar.
<i>ObjectQmgrName</i> Boş ya da saturn.queue.manager	Bu alan isteğe bağlıdır. Boşsa, yerel kuyruk yöneticisinin adı varsayılır. (Bu, uzak kuyruk tanımlamasının bulunduğu kuyruk yöneticisidir.)

Bundan sonra, uygulama bu kuyruğa bir ileti koymak için bir MQPUT çağrısı yayınlar.

Yerel kuyruk yöneticisinde, aşağıdaki MQSC komutlarını kullanarak uzak bir kuyruğun yerel tanımlamasını yaratabilirsiniz:

```
DEFINE QREMOTE (CYAN.REMOTE.QUEUE) +
DESCR ('Queue for auto insurance requests from the branches') +
RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE) +
QMNAME (jupiter.queue.manager) +
XMITQ (INQUOTE.XMIT.QUEUE)
```

Burada:

QREMOTE (CYAN.REMOTE.QUEUE)

Uzak kuyruk nesnesinin yerel adını belirler. Bu ad, bu kuyruk yöneticisine bağlı uygulamaların MQOPEN uzak kuyruk yöneticisindeki AUTOMOBILE.INSURANCE.QUOTE.QUEUE kuyruğu açma çağrısında jupiter.queue.manager belirtmeleri gereken addr.

DESCR ('Queue for auto insurance requests from the branches')

Kuyruğun kullanımını açıklayan ek metin sağlar.

RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE)

Uzak kuyruk yöneticisindeki hedef kuyruğun adını belirtir. Bu, CYAN.REMOTE.QUEUE. Kuyruk AUTOMOBILE.INSURANCE.QUOTE.QUEUE uzak kuyruk yöneticisinde yerel kuyruk olarak tanımlanmalıdır.

QMNAME (jupiter.queue.manager)

AUTOMOBILE.INSURANCE.QUOTE.QUEUE.

XMITQ (INQUOTE.XMIT.QUEUE)

İletim kuyruğunun adını belirler. Bu isteğe bağlıdır; bir iletim kuyruğunun adı belirlenmezse, uzak kuyruk yöneticisiyle aynı adı taşıyan bir kuyruk kullanılır.

Her iki durumda da, uygun iletim kuyruğu, **Usage** özniteliği MQSC komutlarında bir iletim kuyruğu (USAGE (XMITQ)) olduğunu belirten bir yerel kuyruk olarak tanımlanmalıdır.

- İletileri uzak kuyruğa koy (diğer yöntem).

Uzak kuyruğun yerel tanımının kullanılması, iletileri uzak kuyruğa koymanın tek yolu değildir. Uygulamalar, MQOPEN çağrısının bir parçası olarak uzak kuyruk yöneticisi adını da içeren tam kuyruk adını belirtebilir. Bu durumda, uzak kuyruğun yerel tanımına gerek yoktur. Ancak bu, uygulamaların yürütme sırasında uzak kuyruk yöneticisinin adını bilmesi ya da bu ada erişmesi gerektiği anlamına gelir.

- Uzak kuyruklar için diğer komutları kullanın.

Bir uzak kuyruk nesnesinin özniteliklerini görüntülemek ya da değiştirmek için MQSC komutlarını kullanabilir ya da uzak kuyruk nesnesini silebilirsiniz. Örneğin:

- Uzak kuyruğun özniteliklerini görüntülemek için:

```
DISPLAY QUEUE (CYAN.REMOTE.QUEUE)
```

- Uzak kuyruğu, koymaları etkinleştirecek şekilde değiştirmek için. Bu, hedef kuyruğu etkilemez, yalnızca bu uzak kuyruğu belirten uygulamalar:

```
ALTER QREMOTE (CYAN.REMOTE.QUEUE) PUT(ENABLED)
```

- Bu uzak kuyruğu silmek için. Bu, hedef kuyruğu etkilemez, yalnızca yerel tanımlamasını etkiler:

```
DELETE QREMOTE (CYAN.REMOTE.QUEUE)
```

Not: Bir uzak kuyruğu sildiğinizde, yalnızca uzak kuyruğun yerel gösterimini silersiniz. Uzak kuyruğun kendisini ya da kuyruktaki iletileri silmezsiniz.

Uzak kuyruk tanımlamaları diğer ad olarak kullanılabilir

Başka bir kuyruk yöneticisinde bir kuyruk bulmanın yanı sıra, kuyruk yöneticisi diğer adları ve yanıtlanacak kuyruk diğer adları için uzak kuyruğun yerel tanımını da kullanabilirsiniz. Her iki tip diğer ad da uzak kuyruğun yerel tanımlaması yoluyla çözülür. İletinin hedefine ulaşması için uygun kanalları ayarlamanız gerekir.

Kuyruk yöneticisi diğer adları

Diğer ad, bir iletide belirtildiği gibi, hedef kuyruk yöneticisinin adının, ileti rotasındaki bir kuyruk yöneticisi tarafından değiştirildiği işlemidir. Kuyruk yöneticisi diğer adları, kuyruk yöneticilerinden oluşan bir ağdaki iletilerin hedefini denetlemek için kullanabileceğiniz için önemlidir.

Bunu, denetim noktasında kuyruk yöneticisinde uzak kuyruk tanımlamasını değiştirerek yaparsınız. Gönderen uygulama, belirtilen kuyruk yöneticisi adının bir diğer ad olduğunu farkında değil.

Kuyruk yöneticisi diğer adlarıyla ilgili ek bilgi için [Diğer adlar nedir?](#) başlıklı konuya bakın.

Yanıtın gönderileceği kuyruk diğer adları

İsteğe bağlı olarak, bir uygulama kuyruğa *istek iletisi* yerleştirdiğinde yanıt kuyruğunun adını belirtebilir.

İletiyi işleyen uygulama, yanıt kuyruğunun adını alıyorsa, gerekirse *yanıt iletisinin* nereye gönderileceğini bilir.

Yanıt kuyruğu diğer adı, istek iletisinde belirtildiği gibi, bir yanıt kuyruğunun, ileti rotasındaki bir kuyruk yöneticisi tarafından değiştirilmesine ilişkin işlemidir. Gönderen uygulama, belirtilen yanıt kuyruğu adının bir diğer ad olduğunu farkında değil.

Yanıt kuyruğu diğer adı, yanıt kuyruğunun adını ve isteğe bağlı olarak kuyruk yöneticisini değiştirmenizi sağlar. Bu, yanıt iletileri için hangi rotanın kullanılacağını denetlemenizi sağlar.

İstek iletileri, yanıt iletileri ve yanıt kuyrukları hakkında daha fazla bilgi için bkz. [İleti tipleri ve Yanıt kuyruğu ve kuyruk yöneticisi](#).

Yanıt kuyruğu diğer adlarıyla ilgili daha fazla bilgi için [Yanıt kuyruğu diğer adları ve kümeleribaşlıklı konuya](#) bakın.

Diğer ad kuyruklarıyla çalışılması

Başka bir kuyruğa ya da konuya dolaylı olarak gönderme yapmak için bir diğer ad kuyruğu tanımlayabilirsiniz.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).



Uyarı: Dağıtım listeleri, konu nesnelerini işaret eden diğer ad kuyruklarının kullanılmasını desteklemez. Bir diğer ad kuyruğu dağıtım listesindeki bir konu nesnesini gösteriyorsa, IBM MQ MQRC_ALIAS_BASE_Q_TYPE_ERRORdeğerini döndürür.

Bu görev hakkında

Diğer ad kuyruğunun gönderme yaptığı kuyruk aşağıdakilerden biri olabilir:

- Yerel bir kuyruk (bkz. [“DEFINE QLOCAL ile yerel kuyruk tanımlanması” sayfa 158](#)).
- Uzak kuyruğun yerel tanımlaması (bkz. [“Uzak kuyruklarla çalışılması” sayfa 167](#)).
- Bir konu.

Diğer ad kuyruğu gerçek bir kuyruk değil, yürütme sırasında gerçek (ya da hedef) bir kuyruğa çözülen bir tanımlama. Diğer ad kuyruğu tanımlaması hedef kuyruğu belirtir. Bir uygulama bir diğer ad kuyruğuna MQOPEN çağrısı yaptığında, kuyruk yöneticisi diğer adı hedef kuyruk adına çözer.

Bir diğer ad kuyruğu, yerel olarak tanımlanmış başka bir diğer ad kuyruğuna çözülemiyor. Ancak, bir diğer ad kuyruğu, yerel kuyruk yöneticisinin üyesi olduğu kümelerde başka bir yerde tanımlanan diğer ad kuyruklarına çözülebilir. Ek bilgi için [Ad çözünürlüğü](#) konusuna bakın.

Diğer ad kuyrukları aşağıdakiler için kullanışlıdır:

- Farklı uygulamalara hedef kuyruk için farklı erişim yetkisi düzeyleri verme.
- Farklı uygulamaların aynı kuyrukla farklı şekillerde çalışmasına izin verilmesi. (Farklı varsayılan öncelikler ya da farklı varsayılan kalıcılık değerleri atamak isteyebilirsiniz.)
- Bakım, geçiş ve iş yükü dengelemeyi basitleştirme. (Diğer adı kullanmaya devam eden uygulamanızı değiştirmek zorunda kalmadan hedef kuyruk adını değiştirmek istiyor olabilir.)

Örneğin, iletilerin MY.ALIAS.QUEUE. Bu kuyruk, bir MQOPEN isteği yaptığında ve dolaylı olarak bu kuyruğa bir ileti yerleştirdiğinde bu kuyruğun adını belirtir. Uygulama, kuyruğun bir diğer ad kuyruğu olduğunu farkında değil. Bu diğer adı kullanan her MQI çağrısı için, kuyruk yöneticisi gerçek kuyruk adını çözer; bu, yerel bir kuyruk ya da bu kuyruk yöneticisinde tanımlanmış bir uzak kuyruk olabilir.

TARGET özniteliğinin değerini değiştirerek, MQI çağrılarını büyük olasılıkla başka bir kuyruk yöneticisinde başka bir kuyruğa yeniden yönlendirebilirsiniz. Bu, bakım, geçiş ve yük dengeleme için yararlıdır.

Yordam

- Bir diğer ad kuyruğu tanımlayın.

Aşağıdaki MQSC komutu bir diğer ad kuyruğu yaratır:

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGET (YELLOW.QUEUE)
```

Bu komut, MY.ALIAS.QUEUE , YELLOW.QUEUE. Komut hedef kuyruğu yaratmaz; kuyruk YELLOW.QUEUE yürütme sırasında yok.

Diğer ad tanımlamasını değiştirirseniz, MQI çağrılarını başka bir kuyruğa yönlendirebilirsiniz. Örneğin:

```
ALTER QALIAS (MY.ALIAS.QUEUE) TARGET (MAGENTA.QUEUE)
```

Bu komut, MQI çağrılarını başka bir kuyruğa (MAGENTA.QUEUE).

Tek bir kuyruğun (hedef kuyruk) farklı uygulamalara ilişkin farklı özniteliklere sahip görünmesini sağlamak için diğer ad kuyruklarını da kullanabilirsiniz. Bunu, her uygulama için bir tane olmak üzere iki diğer ad tanımlayarak yaparsınız. İki uygulama olduğunu varsayalım:

- Uygulama ALPHA, YELLOW.QUEUE, ancak bu kuyruktan ileti almasına izin verilmiyor.
- Uygulama BETA, YELLOW.QUEUE, ancak üzerine ileti koymasına izin verilmiyor.

Aşağıdaki MQSC komutu, ALPHA uygulaması için etkinleştirilen ve devre dışı bırakılan bir diğer adı tanımlıyor:

```
DEFINE QALIAS (ALPHAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (ENABLED) +  
GET (DISABLED)
```

Aşağıdaki komut, devre dışı bırakılan ve BETA uygulaması için etkinleştirilen bir diğer adı tanımlar:

```
DEFINE QALIAS (BETAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (DISABLED) +  
GET (ENABLED)
```

ALPHA, ALPHAS.ALIAS.QUEUE ; BETA, BETAS.ALIAS.QUEUE. İkisi de aynı kuyruğa farklı yollarla erişiyor.

Kuyruk diğer adlarını tanımlarken LIKE ve REPLACE özniteliklerini, bu öznitelikleri yerel kuyruklarda kullandığınız gibi kullanabilirsiniz.

- Diğer ad kuyrukları olan diğer komutları kullanın.

Diğer ad kuyruğu özniteliklerini görüntülemek ya da değiştirmek ya da diğer ad kuyruğu nesnesini silmek için uygun MQSC komutlarını kullanabilirsiniz. Örneğin:

Diğer ad kuyruğunun özniteliklerini görüntülemek için **DISPLAY QALIAS** komutunu kullanın:

```
DISPLAY QALIAS (ALPHAS.ALIAS.QUEUE)
```

Diğer adın çözüldüğü temel kuyruk adını değiştirmek için **ALTER QALIAS** komutunu kullanın; burada force seçeneği, kuyruk açık olsa bile değişikliği zorlar:

```
ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGET(ORANGE.LOCAL.QUEUE) FORCE
```

Bu kuyruk diğer adını silmek için **DELETE QALIAS** komutunu kullanın:

```
DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

Bir uygulamada kuyruk açıksa, diğer ad kuyruğunu silemezsiniz.

İlgili kavramlar

[Dağıtım listeleri](#)

İlgili başvurular

[QALIAS DEĞİŞTİR](#)

Model kuyruklarıyla çalışma

Model kuyrukları, uygulamaların gerektiği şekilde kuyruk yaratması için uygun bir yöntem sağlar.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu görev hakkında

Bir kuyruk yöneticisi, bir uygulamadan model kuyruğu olarak tanımlanmış bir kuyruk adını belirten bir MQI çağırısı alırsa *dinamik kuyruk* yaratır. Yeni dinamik kuyruğun adı, kuyruk yaratıldığında kuyruk yöneticisi tarafından oluşturulur. *Model kuyruğu*, bu kuyruktan yaratılan dinamik kuyrukların özniteliklerini belirleyen bir şablondur.

Yordam

- Bir model kuyruğu tanımlayın.

DEFINE QMODEL MQSC komutunu kullanarak, yerel kuyruk tanımladığınız gibi, öznitelikler kümesi olan bir model kuyruğu tanımlayabilirsiniz. Model kuyrukları ve yerel kuyruklar aynı öznitelik kümesine sahiptir; ancak, model kuyruklarında yaratılan dinamik kuyrukların geçici mi, yoksa kalıcı mı olduğunu belirleyebilirsiniz. (Kuyruk yöneticisi yeniden başlatma işlemleri boyunca kalıcı kuyruklar tutulur, geçici kuyruklar tutulmaz.) Örneğin:

```
DEFINE QMODEL (GREEN.MODEL.QUEUE) +
DESCR('Queue for messages from application X') +
PUT (DISABLED) +
GET (ENABLED) +
NOTRIGGER +
MSGDLVSQ (FIFO) +
MAXDEPTH (1000) +
MAXMSGL (2000) +
USAGE (NORMAL) +
DEFTYPE (PERMDYN)
```

Bu komut bir model kuyruğu tanımlaması yaratır. **DEFTYPE** özniteliğinden, bu şablondan yaratılan gerçek kuyrukların kalıcı dinamik kuyruklar olduğunu görebilirsiniz. Belirtilmeyen öznitelikler, SYSYSTEM.DEFAULT.MODEL.QUEUE varsayılan kuyruğu.

Model kuyrukları tanımlarken **LIKE** ve **REPLACE** özniteliklerini, yerel kuyruklarla aynı şekilde kullanabilirsiniz.

- Model kuyruklarıyla diğer komutları kullanın.

Bir model kuyruğunun özniteliklerini görüntülemek ya da değiştirmek ya da model kuyruğu nesnesini silmek için uygun MQSC komutlarını kullanabilirsiniz. Örneğin:

Model kuyruğunun özniteliklerini görüntülemek için **DISPLAY QUEUE** komutunu kullanın:

```
DISPLAY QUEUE (GREEN.MODEL.QUEUE)
```

Bu modelden yaratılan herhangi bir dinamik kuyruğa koymayı etkinleştirmek üzere modeli değiştirmek için **ALTER QMODEL** komutunu kullanın:

```
ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)
```

Bu model kuyruğunu silmek için **DELETE QMODEL** komutunu kullanın:

```
DELETE QMODEL (RED.MODEL.QUEUE)
```

İlgili başvurular

[QMODEL DEĞİŞTİR](#)

[QXX_ENCODE_CASE_ONE modeli TANIMLAYIN](#)

[QMODEL SIL](#)

[KUYRUK GÖRÜNTÜLE](#)

Gönderilmeyen ileti kuyruklarıyla çalışılması

Her kuyruk yöneticisinin genellikle, bir teslim edilmeyen ileti kuyruğu olarak kullanılacak yerel bir kuyruğu vardır; böylece, doğru hedeflerine teslim edilmeyen iletiler daha sonra alınmak üzere saklanabilir. Kuyruk yöneticisine, teslim edilmeyen iletiler kuyruğundan bahsedin ve teslim edilmeyen iletiler kuyruğunda bulunan iletilerin nasıl işleneceğini belirleyin. Gitmeyen ileti kuyruklarının kullanılması, iletilerin teslim edildiği sırayı etkileyebilir, bu nedenle bunları kullanmamayı seçebilirsiniz.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu görev hakkında

SYSTEM.DEAD.LETTER.QUEUE , ürünle birlikte kullanılabilir. Bu kuyruk, kuyruk yöneticisini yarattığınızda otomatik olarak yaratılır. Gerekirse bu tanımı değiştirebilir ve yeniden adlandırabilirsiniz.

Bir teslim mektubu kuyruğunun aşağıdakiler dışında özel bir gereksinimi yoktur:

- Yerel bir kuyruk olmalıdır
- MAXMSGL (ileti uzunluğu üst sınırı) özniteliği, kuyruk yöneticisinin işlemesi gereken en büyük iletileri **artı** fazla olmayan harf üstbilgisinin (MQDLH) boyutunu barındırmak için kuyruğu etkinleştirmelidir.

Gitmeyen ileti kuyruklarının kullanılması, iletilerin teslim edildiği sırayı etkileyebilir, bu nedenle bunları kullanmamayı seçebilirsiniz.

Yordam

- Kuyruk yöneticisine ileti kuyruğundan bahset.

Bunu yapmak için, **crtmqm** komutunda (örneğin, **crtmqm -u DEAD.LETTER.QUEUE**) bir kullanılmaz harf kuyruğu adı belirtin ya da daha sonra bir ad belirtmek için **ALTER QMGR** komutunda **DEADQ** özniteliğini kullanın. Kullanılmadan önce, gönderilmeyen iletiler kuyruğunu tanımlamanız gerekir.

- Bir teslim edilmeyen ileti kuyruğunda bulunan iletilerin nasıl işleneceğini belirleyin.

İletiler teslim edilemediğinde, teslim edilmeyen iletiler kuyruğunun kullanılıp kullanılmayacağını belirlemek için USEDLO kanal özniteliğini ayarladınız. Bu öznitelik, kuyruk yöneticisinin bazı işlevlerinin gitmeyen iletiler kuyruğunu kullanması için yapılandırılabilir, ancak diğer işlevler kullanmaz. Farklı MQSC komutlarında USEDLO kanal özniteliğinin kullanımıyla ilgili ek bilgi için [DEFINE CHANNEL](#), [DISPLAY CHANNEL](#), [ALTER CHANNEL](#) ve [DISPLAY CLUSQMGR](#) başlıklı konuya bakın.

Bir teslim edilmeyen ileti kuyruğunda bulunan iletilerin nasıl işleneceğini ya da kaldırılacağını belirtmek için IBM MQ ileti kuyruğu işleyicisini kullanırsınız. Bkz. "[IBM MQ gönderilmeyen ileti kuyruğundaki iletilerin işlenmesi](#)" sayfa 174.

İlgili kavramlar

[Gönderilmeyen ileti kuyrukları](#)

İlgili görevler

Teslim edilmemiş iletilerde sorun giderme

İlgili başvurular

ALTER QMGR

crtmqm (kuyruk yöneticisi yarat)

IBM MQ gönderilmeyen ileti kuyruğundaki iletilerin işlenmesi

Bir almayan ileti kuyruğundaki (DLQ) iletileri işlemek için, IBM MQ tarafından sağlanan varsayılan DLQ işleyicisini kullanın. İşleyici, DLQ 'daki iletileri, tanımladığınız bir kural çizelgesindeki girişlerle eşleştirir.

Bu görev hakkında

İletiler, kuyruk yöneticileri, ileti kanalı araçları (MCA 'lar) ve uygulamalar tarafından bir DLQ'ya konabilir. DLQ üzerindeki tüm iletilerin başına *alinmeyen harf üstbilgisi* yapısı (MQDLH) eklenmelidir. Bir kuyruk yöneticisi ya da ileti kanalı aracısı tarafından DLQ 'ya konan iletiler her zaman bu üstbilgiye sahiptir; DLQ 'ya ileti koyacak uygulamalar bu üstbilgiyi sağlamalıdır. MQDLH yapısının *Neden* alanı, iletinin neden DLQ üzerinde olduğunu belirten bir neden kodu içerir.

Tüm IBM MQ ortamlarının DLQ üzerindeki iletileri düzenli olarak işlemesi için bir yordam gerekir. IBM MQ , **runmqdlq** MQSC komutunu kullanarak çağırdığınız *kullanılmayan ileti kuyruğu işleyicisi* (DLQ işleyicisi) adlı varsayılan bir yordam sağlar.

DLQ üzerindeki iletilerin işlenmesine ilişkin yönergeler, DLQ işleyicisine kullanıcı tarafından yazılan bir *kural çizelgesi* yoluyla sağlanır. DLQ işleyicisi, DLQ 'daki iletileri kural çizelgesindeki girişlerle eşleştirir; DLQ iletisi kural çizelgesindeki bir girişle eşleştiğinde, DLQ işleyicisi o girişle ilişkilendirilmiş işlemi gerçekleştirir.

İlgili görevler

Teslim edilmemiş iletilerde sorun giderme

İlgili başvurular

Gönderilmeyen ileti kuyrukları

Gönderilmeyen ileti kuyruğu işleyicisi çağrılıyor

runmqdlq denetim komutunu kullanarak, kullanılmayan ileti kuyruğu (DLQ) işleyicisini çağırın. İşlemek istediğiniz DLQ 'ya ve kullanmak istediğiniz kuyruk yöneticisine iki şekilde ad verebilirsiniz.

Başlamadan önce

DLQ işleyicisini çalıştırmak için hem DLQ 'nun kendisine hem de DLQ 'daki iletilerin iletildiği ileti kuyruklarına erişme yetkiniz olmalıdır. DLQ işleyicisinin ileti bağlamında kullanıcı kimliği yetkisi olan kuyruklara ileti koyabilmesi için, diğer kullanıcıların kimliklerini varsayma yetkiniz de olmalıdır.

Bu görev hakkında

Aşağıdaki örnekler, kuyruk yöneticisinin sahibi olduğu ABC1 . DEAD . LETTER . QUEUE adlı DLQ için geçerlidir: ABC1 . QUEUE . MANAGER.

DLQ 'yu ya da kuyruk yöneticisini gösterildiği gibi belirtmezseniz, kuruluşa ilişkin varsayılan kuyruk yöneticisi, o kuyruk yöneticisine ait DLQ ile birlikte kullanılır.

runmqdlq komutu, girişini stdin' den alır. Kurallar tablosundan stdin ' yi yeniden yönlendirerek kurallar tablosunu **runmqdlq** ile ilişkilendirin.

runmqdlq komutuyla ilgili daha fazla bilgi için bkz. [runmqdlq](#).

Yordam

- DLQ 'yu ve kuyruk yöneticisini **runmqdlq** komutunun parametreleri olarak adlandır.

Örneğin, komut isteminden:

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER <qrule.rul
```

- Kurallar çizelgesinde DLQ ve kuyruk yöneticisine ad koyabilirsiniz.

Örneğin:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

İlgili kavramlar

Gönderilmeyen ileti kuyrukları

İlgili görevler

Teslim edilmemiş iletilerde sorun giderme

Örnek DLQ işleyicisi **amqsd1q**

runmqdlq komutu kullanılarak çağrılan kullanılmayan ileti kuyruğu işleyicisine ek olarak IBM MQ , **runmqdlq** tarafından sağlanana benzer bir işlevle **amqsd1q** örnek bir DLQ işleyicisinin kaynağını sağlar.

amqsd1q olanağını, gereksinimlerinizi karşılayacak bir DLQ işleyicisi sağlayacak şekilde uyarlayabilirsiniz. Örneğin, iletileri almayan harf üstbilgileri olmadan işleyebilen bir DLQ işleyicisi istediğinize karar verebilirsiniz. (Varsayılan DLQ işleyicisi ve örnek (**amqsd1q**) yalnızca, DLQ üzerinde, MQDLH ' ye ilişkin bir çift harfli üstbilgiyle başlayan iletileri işler. MQDLH ile başlamayan iletiler hatalı olarak tanımlanır ve DLQ ' da süresiz olarak kalır.)

MQ_INSTALLATION_PATH , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

IBM MQ for Windows içinde amqsd1q kaynağı şu dizinde sağlanır:

```
MQ_INSTALLATION_PATH\tools\c\samples\d1q
```

ve derlenmiş sürüm dizinde sağlanır:

```
MQ_INSTALLATION_PATH\tools\c\samples\bin
```

IBM MQ for UNIX ve Linux sistemlerinde **amqsd1q** kaynağı şu dizinde sağlanır:

```
MQ_INSTALLATION_PATH/samp/d1q
```

ve derlenmiş sürüm dizinde sağlanır:

```
MQ_INSTALLATION_PATH/samp/bin
```

V9.3.0 Örnek programın **amqsd1q**cadlı oluşturulmuş bir sürümü vardır. İstemci kipinde uzak bir kuyruk yöneticisine bağlanmak için bunu kullanabilirsiniz. **amqsd1q** ' den yararlanmak için, **MQSERVER** , **MQCHLLIB** ya da **MQCHLTAB** ortam değişkenlerinden birini, kuyruk yöneticisine nasıl bağlanacağını tanımlamak üzere ayarlamanız gerekir. Örneğin:

```
export MQSERVER="SYSTEM.DEF.SVRCONN/TCP/myappliance.co.uk(1414) "
```

DLQ işleyicisi kural çizelgesi

Gelmeyen ileti kuyruğu işleyicisi kural çizelgesi, DLQ işleyicisinin DLQ ' ya gelen iletileri nasıl işleyeceğini tanımlar.

Bir kural çizelgesinde iki tip giriş vardır:

- Çizelgedeki isteğe bağlı ilk giriş *denetim verilerini* içerir.
- Çizelgedeki diğer tüm girişler, DLQ işleyicisinin izlemesine ilişkin *kurallardır* . Her kural, bir iletinin eşleştirildiği bir *örüntüden* (ileti özellikleri kümesi) ve DLQ ' daki bir ileti belirtilen örüntüyle eşleştiginde yapılacak bir *işlemden* oluşur. Bir kural tablosunda en az bir kural olmalıdır.

Kurallar çizelgesindeki her giriş bir ya da daha çok anahtar sözcükten oluşur.

İlgili kavramlar

Gönderilmeyen ileti kuyrukları

İlgili görevler

Teslim edilmemiş iletilerde sorun giderme

DLQ denetim verileri

Bir teslim edilmeyen ileti kuyruğu işleyicisi kural çizelgesindeki bir denetim verileri girişine anahtar sözcükleri ekleyebilirsiniz.

Not:

- Düşey çizgi (|) alternatifleri ayırır; bunlardan yalnızca biri belirtilebilir.
- Tüm anahtar sözcükler isteğe bağlıdır.

INPUTQ (QueueName| " (varsayılan)

İşlemek istediğiniz DLQ ' nun adı:

1. runmqdlq komutu için parametre olarak sağladığınız herhangi bir INPUTQ değeri, kurallar çizelgesindeki herhangi bir INPUTQ değerini geçersiz kılar.
2. runmqdlq komutunun parametresi olarak bir INPUTQ değeri belirtmezseniz, ancak kurallar çizelgesinde bir değer **belirtirseniz** , kurallar çizelgesindeki INPUTQ değeri kullanılır.
3. DLQ belirtilmezse ya da kural çizelgesinde INPUTQ (") belirtirseniz, kuyruk yöneticisine ait DLQ ' nun adı, runmqdlq komutu için değiştirge olarak verilen addır.
4. runmqdlq komutuna parametre olarak ya da kurallar çizelgesinde bir değer olarak INPUTQ değeri belirtmezseniz, kurallar çizelgesindeki INPUTQM anahtar sözcüğünde belirtilen kuyruk yöneticisine ait DLQ kullanılır.

INPUTQM (QueueManagerAd| " (varsayılan)

INPUTQ anahtar sözcüğünde adı belirtilen DLQ ' nun sahibi olan kuyruk yöneticisinin adı:

1. runmqdlq komutuna parametre olarak sağladığınız herhangi bir INPUTQM değeri, kurallar tablosundaki herhangi bir INPUTQM değerini geçersiz kılar.
2. runmqdlq komutu için parametre olarak bir INPUTQM değeri belirtmezseniz, kurallar tablosundaki INPUTQM değeri kullanılır.
3. Kuyruk yöneticisi belirtilmezse ya da kurallar tablosunda INPUTQM (") belirtilirseniz, kuruluş için varsayılan kuyruk yöneticisi kullanılır.

RETRYINT (Aralık| 60 (varsayılan)

DLQ işleyicisinin, ilk girişimde işlenemeyen ve yinelenen girişimleri istenen DLQ ' daki iletileri yeniden işlemesi için saniye cinsinden aralık. Varsayılan olarak, yeniden deneme aralığı 60 saniyedir.

WAIT (YES (varsayılan) |NO|nnn)

DLQ işleyicisinin, işleyebileceği başka ileti olmadığını saptadığında DLQ ' ya daha fazla ileti gelmesini bekleyip beklememesi gerekip gerekmediğini belirler.

EVET

DLQ işleyicisi süresiz olarak bekliyor.

HAYIR

DLQ işleyicisi DLQ ' nun boş olduğunu ya da işleyebileceği ileti içermediğini saptadığında sona erer.

nnn

DLQ işleyicisi, kuyruğun boş olduğunu ya da işleyebileceği ileti içermediğini algıladıktan sonra sona ermeden önce yeni işin gelmesi için nnn saniye bekler.

Meşgul DLQ ' lar için WAIT (YES) ve WAIT (NO) ya da WAIT (nnn) belirleyin düşük etkinlik düzeyine sahip DLQ ' lar için. DLQ işleyicisinin sonlandırılmasına izin verildiyse, tetikleme kullanarak yeniden çağırın. Tetikleme hakkında daha fazla bilgi için Tetikleyicileri kullanarak IBM MQ uygulamaları başlatma başlıklı konuya bakın.

Denetim verilerini kural çizelgesine dahil etmek yerine, runmqdlq komutunun giriş değıştirmeleri olarak DLQ ve kuyruk yöneticisi adlarını belirtmeniz gerekir. Hem kurallar çizelgesinde hem de runmqdlq komutuna giriş olarak bir değeri belirtirseniz, runmqdlq komutunda belirtilen değeri önceliklidir.

Kurallar çizelgesine bir denetim verileri girişı eklerseniz, bu giriş çizelgedeki **ilk** giriş olmalıdır.

DLQ kuralları (örüntüler ve işlemler)

Kalıpla eşleşen anahtar sözcüklerin (ileti kuyruğundaki iletilerin eşleştiğı anahtar sözcüklerin) ve işlem anahtar sözcüklerinin (DLQ işleyicisinin eşleşen bir iletiyi nasıl işleyeceğini belirleyen anahtar sözcükler) açıklaması. Örnek bir kural da sağlanır.

Örüntü eşleştirme anahtar sözcükleri

DLQ 'daki iletilerin eşleştirileceğı değeri belirtmek için kullandığınız örüntü eşleştirme anahtar sözcükleri aşağıdaki gibidir. (Tüm örüntü eşleştirme anahtar sözcükleri isteğe bağılıdır):

APPLIDAT (*ApplIdentityData* | * (varsayılan)

DLQ 'daki iletinin ileti tanımlayıcısında (MQMD) belirtilen *ApplIdentityVerileri* değeri.

APPLNAME (*PutApplAdı* | * (varsayılan)

DLQ 'daki iletinin tanımlayıcısının (MQMD) *PutApplName* alanında belirtildiğı şekilde, MQPUT ya da MQPUT1 çağrısını yayınlayan uygulamanın adı.

APPLTYPE (*PutApplTip* | * (varsayılan)

DLQ 'daki iletinin ileti tanımlayıcısında (MQMD) belirtilen *PutApplTip* değeri.

DESTQ (*QueueName* | * (varsayılan)

İletin gönderildiğı ileti kuyruğunun adı.

DESTQM (*QueueManagerAdı* | * (varsayılan)

İletin gönderildiğı ileti kuyruğunun kuyruk yöneticisinin adı.

FEEDBACK (*Geribildirim* | * (varsayılan)

MsgType değeri MQFB_REPORT olduğunda, *Geribildirim* raporun niteliğini açıklar.

Sembolik adlar kullanabilirsiniz. Örneğin, hedef kuyruklarına varışlarının doğrulanması gereken DLQ 'daki iletileri tanımlamak için MQFB_COA simgesel adını kullanabilirsiniz.

FORMAT (*Biçim* | * (varsayılan)

İletiyi gönderenin ileti verilerinin biçimini tanımlamak için kullandığı ad.

MSGTYPE (*MsgType* | * (varsayılan)

DLQ 'daki iletinin tipi.

Sembolik adlar kullanabilirsiniz. Örneğin, yanıt gereken DLQ 'da bu iletileri tanımlamak için MQMT_REQUEST simgesel adını kullanabilirsiniz.

PERSIST (*Kalıcılık* | * (varsayılan)

İletin kalıcılık değeri. (Bir iletinin kalıcı olarak saklanması, kuyruk yöneticisinin yeniden başlatılıp başlatılmayacağını belirler.)

Sembolik adlar kullanabilirsiniz. Örneğin, DLQ 'daki kalıcı iletileri tanımlamak için MQPER_PERSISTENT sembolik adını kullanabilirsiniz.

NEDEN (*ReasonCode* | * (varsayılan)

İletin DLQ 'ya neden konduğunu açıklayan neden kodu.

Sembolik adlar kullanabilirsiniz. Örneğin, hedef kuyrukları dolu olduğuna için DLQ 'ya yerleştirilen iletileri tanımlamak için MQRC_Q_FULL sembolik adını kullanabilirsiniz.

REPLYQ (*QueueName* | * (varsayılan)

DLQ 'daki iletinin ileti tanımlayıcısında (MQMD) belirtilen yanıt kuyruğunun adı.

REPLYQM (*QueueManagerAdı* | * (varsayılan)

DLQ 'daki iletinin ileti tanımlayıcısında (MQMD) belirtildiğı şekilde, yanıt kuyruğunun kuyruk yöneticisinin adı.

USERID (*UserIdentifier (varsayılan))**

DLQ 'daki iletinin ileti tanımlayıcısında (MQMD) belirtildiği şekilde, DLQ' da iletiyi oluşturan kullanıcının kullanıcı kimliği.

İşlem anahtar sözcükleri

Eşleşen bir iletinin nasıl işleneceğini tanımlamak için kullanılan işlem anahtar sözcükleri şunlardır:

EYLEM (AT | YOKSAY | YENİDEN DENE | FWD)

DLQ ' da bu kuralda tanımlanan kalıpla eşleşen herhangi bir ileti için yapılacak işlem.

At

İletiyi DLQ ' dan silin.

IGNORE

İletiyi DLQ üzerinde bırakın.

Yeniden deneme

İletiyi hedef kuyruğuna koyma girişimi başarısız olursa, yeniden deneyin. RETRY anahtar sözcüğü, bir işlemi gerçekleştirmek için yapılan deneme sayısını ayarlar. Denetim verilerinin RETRYINT anahtar sözcüğü, girişimler arasındaki aralığı denetler.

FWD

İletiyi FWDQ anahtar sözcüğünde adı belirtilen kuyruğa iletin.

ACTION anahtar sözcüğünü belirtmelisiniz.

FWDQ (*QueueName* | & DESTQ | & REPLYQ)

İŞLEM (FWD) istendiğinde iletinin iletileceği ileti kuyruğunun adı.

QueueName

İleti kuyruğunun adı. FWDQ (") geçerli değil.

& DESTQ

Kuyruk adını MQDLH yapısındaki *DestQName* alanından alın.

& REPLYQ

Kuyruk adını, MQMD ileti tanımlayıcısındaki *ReplyToQ* alanından alın.

FWDQ (& REPLYQ) belirten bir kural, boş *ReplyToQ* alanıyla eşleştğinde hata iletilerini önlemek için, ileti kalıbında REPLYQ (? *) belirtin.

FWDQM (*QueueManagerAdı* | & DESTQM | & REPLYQM | " (varsayılan))

İletinin iletileceği kuyruğun kuyruk yöneticisi.

QueueManagerAdı

ACTION (FWD) isteğinde bulunulduğunda iletinin iletileceği kuyruğun kuyruk yöneticisinin adı.

& DESTQM

Kuyruk yöneticisi adını MQDLH yapısındaki *DestQMGrAd* alanından alın.

& REPLYQM

Kuyruk yöneticisi adını, MQMD ileti tanımlayıcısındaki *ReplyToQMGr* alanından alın.

''

Varsayılan değer olan FWDQM (""), yerel kuyruk yöneticisini tanımlar.

HEADER (YES (varsayılan) |NO)

MQDLH ' nin ACTION (FWD) isteğinde bulunduğu bir iletide kalıp kalmayacağı. Varsayılan olarak, MQDLH iletide kalır. HEADER anahtar sözcüğü FWD dışındaki işlemler için geçerli değil.

PUTAUT (DEF (varsayılan) | CTX)

İletilerin DLQ işleyicisi tarafından konması gereken yetki:

DEF

DLQ işleyicisinin kendisine ilişkin yetkiye sahip iletileri koyun.

CTX

Kullanıcı kimliği yetkisi olan iletileri ileti bağlamına koyun. PUTAUT (CTX) belirtirseniz, diğer kullanıcıların kimliklerini üstlenme yetkiniz olmalıdır.

RETRY (*RetryCount*|1 (varsayılan))

1-999.999.999 aralığında, denetim verilerinin RETRYINT anahtar sözcüğünde belirlenen aralıkta bir işlemi deneme sayısı. DLQ işleyicisi tarafından belirli bir kuralı gerçekleştirmek için yapılan deneme sayısı, DLQ işleyicisinin yürürlükteki yönetim ortamına özgüdür; sayı yeniden başlatma işlemleri boyunca devam etmez. DLQ işleyicisi yeniden başlatılırsa, kural uygulama girişimi sayısı sıfırlanır.

Örnek kural

Aşağıda bir DLQ işleyici kuralları çizelgesinden örnek bir kural verilmiştir:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Bu kural, DLQ işleyicisine, MQPUT ve MQPUT1 engellendiği için DLQ ' ya konan herhangi bir kalıcı iletiyi hedef kuyruğuna teslim etme girişiminde bulunmasını bildirir.

Bir kuralda kullanabileceğiniz tüm anahtar sözcükler bu bölümün geri kalanında açıklanır. Aşağıdakileri unutmayın:

- Varsa, bir anahtar sözcüğün varsayılan değerinin altı çizilir. Çoğu anahtar sözcük için varsayılan değer, herhangi bir değerle eşleşen * (yıldız işareti) değeridir.
- Düşey çizgi (|) alternatifleri ayırır; bunlardan yalnızca biri belirtilebilir.
- ACTION dışındaki tüm anahtar sözcükler isteğe bağlıdır.

DLQ kural çizelgesi kuralları

Gönderi kuyruğu işleyici kuralları çizelgesinin sözdizimi, yapısı ve içeriği bu kurallara uygun olmalıdır.

Kurallar çizelgesi aşağıdaki kurallara uymalıdır:

- Bir kural tablosu en az bir kural içermelidir.
- Anahtar sözcükler herhangi bir sırada olabilir.
- Bir anahtar sözcük herhangi bir kurala yalnızca bir kez eklenebilir.
- Anahtar sözcükler büyük ve küçük harfe duyarlı değildir.
- Bir anahtar sözcük ve parametre değeri, diğer anahtar sözcüklerden en az bir boşluk ya da virgülle ayrılmalıdır.
- Bir kuralın başında ya da sonunda ve anahtar sözcükler, noktalama işaretleri ve değerler arasında herhangi bir sayıda boşluk olabilir.
- Her kural yeni bir satırda başlamalıdır.
- Windows sistemlerinde, çizelgedeki son kural bir satır başı/satır besleme karakteriyle bitmelidir. Bunu, çizelgenin son satırının boş bir satır olması için kuralın sonundaki Enter tuşuna basarak yapabilirsiniz.
- Taşınabilirlik nedeniyle, bir satırın önemli uzunluğu 72 karakterden fazla olmamalıdır.
- Bir satırdaki son boşluk olmayan karakter olarak artı işaretini (+) kullanarak, kuralın sonraki satırdaki ilk boş olmayan karakterden devam ettiğini belirtebilirsiniz. Kuralın sonraki satırın başlangıcından itibaren devam ettiğini belirtmek için, satırdaki son boşluk olmayan karakter olarak eksi işaretini (-) kullanın. Anahtar sözcükler ve deęiřtirgeler içinde devam karakterleri olabilir.

Örneğin:

```
APPLNAME('ABC+  
D')
```

'ABCD' ile sonuçlanır ve

```
APPLNAME('ABC-  
D')
```

' ABC D' ile sonuçlanır.

- Yıldız işaretiyle (*) başlayan açıklama satırları, kurallar çizelgesinin herhangi bir yerinde olabilir.
- Boş satırlar yoksayılr.
- DLQ işleyici kuralları çizelgesindeki her giriş bir ya da daha çok anahtar sözcükle ilişkili değıştirtigelerden oluşur. Parametreler řu sözdizimi kurallarına uymalıdır:
 - Her parametre değeri en az bir anlamlı karakter içermelidir. Tırnak içine alınan değerlerdeki sınırlayıcı tek tırnak işaretleri önemli olarak kabul edilmez. Örneğın, bu parametreler geçerlidir:

FORMAT (' ABC ')	3 önemli karakter
FORMAT (ABC)	3 önemli karakter
FORMAT (' A ')	1 önemli karakter
FORMAT (A)	1 önemli karakter
FORMAT (' ')	1 önemli karakter

Bu parametreler önemli karakterler içermediğinden geçersiz:

```
FORMAT ( ' ' )  
FORMAT ( )  
FORMAT ( )  
FORMAT
```

- Genel arama karakterleri desteklenir. Sondaki boşluk dışında herhangi bir tek karakter yerine soru işaretini (?) kullanabilirsiniz; bitişik ya da sıfır yerine yıldız işaretini (*) kullanabilirsiniz. Yıldız işareti (*) ve soru işareti (?), parametre değerlerinde genel arama karakteri olarak **her zaman** yorumlanır.
- řu anahtar sözcüklerin parametrelerine genel arama karakterleri eklenemez: ACTION, HEADER, RETRY, FWDQ, FWDQM ve PUTAUT.
- Parametre değerlerinde ve DLQ ' daki iletide karşılık gelen alanlarda sondaki boşluklar, genel arama karakteri eşleşmeleri gerçekleştirilirken önemli değildir. Ancak, tek tırnak içine alınan dizgilerin başındaki ve içindeki gömülü boşluklar, genel arama karakteri eşleşmeleri için önemlidir.
- Sayısal parametreler soru işareti (?) genel arama karakteri içerebilir. Sayısal bir parametrenin parçası olarak değil, tüm sayısal parametre yerine yıldız işaretini (*) kullanabilirsiniz. Örneğın, bunlar geçerli sayısal parametrelerdir:

MSGTYPE (2)	Yalnızca yanıt iletileri seçilebilir
MSGTYPE (*)	Herhangi bir ileti tipi seçilebilir
MSGTYPE (' * ')	Herhangi bir ileti tipi seçilebilir

Ancak MSGTYPE (' 2* ') , sayısal bir parametrenin parçası olarak yıldız işareti (*) içerdğinden geçerli değildir.

- Sayısal parametreler 0-999 999 999 aralığında olmalıdır. Parametre değeri bu aralıktaysa, anahtar sözcüğün ilgili olduğu alanda geçerli olmasa da kabul edilir. Sayısal parametreler için simgesel adlar kullanabilirsiniz.
- Bir dizgi değeri, anahtar sözcüğün ilişkili olduğu MQDLH ya da MQMD ' deki alandan kısaysa, değeri alanın uzunluğuyla boşluklarla doldurulur. Yıldız işaretleri dışında değeri alandan uzunsa, bir hata tanısı konur. Örneğın, bunların tümü 8 karakterlik bir alan için geçerli dizgi değerleridir:

' ABCDEFGH '	8 karakter
' A*C*E*G*I '	Yıldız işaretleri hariç 5 karakter
' *A*C*E*G*I*K*M*O * '	Yıldız işaretleri dışında 8 karakter

- Boşluklar, küçük harfli karakterler ya da nokta (.), eğik çizgi (?), alt çizgi (_) ve yüzde imi (%) dışında özel karakterler içeren dizgileri tek tırnak işareti içine alın. Tek tırnak içine alınmamış küçük harfler büyük harfe katlanır. Dizgi bir tırnak işareti içeriyorsa, tırnak işaretinin başlangıcını ve sonunu belirtmek için iki tek tırnak işareti kullanın. Dizginin uzunluğu hesaplandığında, çift tırnak işaretlerinin her geçişi tek bir karakter olarak sayılır.

DLQ kural çizelgesinin işleme şekli

İleti gönderilmeyen kuyruk işleyicisi, kural çizelgesinde, örüntünün DLQ 'daki bir iletiyle eşleştiği bir kural arar.

Arama, tablodaki ilk kuralla başlar ve tablo boyunca sırayla devam eder. DLQ işleyicisi eşleşen bir örüntüye sahip bir kural bulduğunda, o kuralın işlemini yapar. DLQ işleyicisi, bir kural uygulandığında, o kural için yeniden deneme sayısını 1 artırır. İlk deneme başarısız olursa, DLQ işleyicisi RETRY anahtar sözcüğünde belirtilen sayıyla eşleşinceye kadar yeniden dener. Tüm girişimler başarısız olursa, DLQ işleyicisi çizelgede sonraki eşleşen kuralı arar.

Bu işlem, bir işlem başarılı oluncaya kadar sonraki eşleştirme kuralları için yinelenir. Her eşleşen kural RETRY anahtar sözcüğünde belirtilen sayıda denendiğinde ve tüm girişimler başarısız olduğunda, ACTION (IGNORE) varsayılır. Eşleşen bir kural bulunamazsa, ACTION (IGNORE) da varsayılır.

Not:

1. Eşleşen kural örüntüleri yalnızca, DLQ 'da MQDLH ile başlayan iletiler için aranır. MQDLH ile başlamayan iletiler belirli aralıklarla hatalı olarak bildirilir ve DLQ üzerinde süresiz olarak kalır.
2. Bir kuralın yalnızca bir işlemde oluşması için, tüm örüntü anahtar sözcüklerinin varsayılan olarak kullanılmasına izin verilebilir. Ancak, yalnızca işlem kurallarının kuyruktaki MQDLH 'leri olan ve çizelgedeki diğer kurallara uygun olarak işlenmemiş tüm iletilere uygulandığını unutmayın.
3. DLQ işleyicisi başlatıldığında kural çizelgesinin geçerliliği denetlenir ve o sırada hatalar işaretlenir. Kural çizelgesinde istediğiniz zaman değişiklik yapabilirsiniz; ancak, DLQ işleyicisi yeniden başlatılincaya kadar bu değişiklikler yürürlüğe girmez.
4. DLQ işleyicisi, iletilerin içeriğini, MQDLH 'yi ya da ileti tanımlayıcısını değiştirmez. DLQ işleyicisi iletileri her zaman MQPMO_PASS_ALL_CONTEXT ileti seçeneğiyle diğer kuyruklara koyar.
5. Kurallar çizelgesi, geçerlilik denetimi sırasında yinelenen hata oluşmasını ortadan kaldırmak üzere tasarlandığından, kural çizelgesindeki ardışık sözdizimi hataları tanınmayabilir.
6. DLQ işleyicisi DLQ 'yu MQOO_INPUT_AS_Q_DEF seçeneğiyle açar.
7. DLQ işleyicisinin birden çok eşgörünümü, aynı kural çizelgesini kullanarak aynı kuyruk için eşzamanlı olarak çalışabilir. Ancak, DLQ ile DLQ işleyicisi arasında bire bir ilişki olması daha olağandır.

İlgili kavramlar

Gönderilmeyen ileti kuyrukları

İlgili görevler

Teslim edilmemiş iletilerde sorun giderme

Örnek bir DLQ işleyicisi kural çizelgesi

Tek bir denetim verileri girdisi ve birkaç kural içeren, **runmqdlq** komutu için örnek bir ölmeyen ileti kuyruğu kuralları tablosu.

```
*****
*   An example rules table for the runmqdlq command       *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* runmqdlq, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to runmqdlq,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
```

```
* deliver the message to the intended destination.
* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).
```

```
REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)
```

```
* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).
```

```
REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)
```

```
* The AAAA corporation are always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.
```

```
MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)
```

```
* The BBBB corporation never do things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2
```

```
DESTQM(bbbb.1) +
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)
```

```
* The CCCC corporation considers itself very security
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.
```

```
REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)
```

```
* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it should be able
* to cope with the message being lost, so we can afford to
* discard the message. PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)
* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We don't have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.
```

```
REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)
```

```
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)
```

İlgili kavramlar

[Gönderilmeyen ileti kuyrukları](#)

İlgili görevler

[Teslim edilmemiş iletilerde sorun giderme](#)

İlgili başvurular

[runmqdlq \(teslim harfli kuyruk işleyicisini çalıştır\)](#)

IBM MQ for IBM i' da, **STRMQMDLQ** komutunu ayarlayarak DLQ işleyicisini çağırıyorsunuz.

Başlamadan önce

DLQ işleyicisini çalıştırmak için hem DLQ 'nun kendisine, hem de DLQ' daki iletilerin iletileceği ileti kuyruklarına erişme yetkiniz olmalıdır. DLQ ' nun ileti bağlamında kullanıcı kimliği yetkisi olan iletileri kuyruklara koyabilmesi için, diğer kullanıcıların kimliklerini alma yetkiniz de olmalıdır.

Not: Genellikle bir DLQ ' ya ileti yerleştirmekten kaçınmak tercih edilir. DLQ ' ların kullanımı ve önlenmesiyle ilgili bilgi için bkz. [“Gönderilmeyen ileti kuyruklarıyla çalışılması” sayfa 173.](#)

Bu görev hakkında

Teslim edilmeyen ileti kuyruğu (DLQ), bazen *teslim edilmeyen ileti kuyruğu* olarak da adlandırılır, hedef kuyruklarına teslim edilemeyen iletiler için bir tutma kuyruğudur. Bir ağdaki her kuyruk yöneticisinin ilişkili bir DLQ 'su olmalıdır.

Kuyruk yöneticileri, ileti kanalı araçları ve uygulamalar DLQ ' ya ileti yerleştirebilirler. DLQ üzerindeki tüm iletilerin başına *alinmeyen harf üstbilgisi* yapısı (MQDLH) eklenmelidir. Bir kuyruk yöneticisi ya da ileti kanalı aracı tarafından DLQ ' ya konan iletilerin her zaman bir MQDLH 'si vardır. DLQ ' ya ileti koyan uygulamalar için bir MQDLH belirtmelisiniz.

MQDLH yapısının *Neden* alanı, iletinin neden DLQ üzerinde olduğunu belirten bir neden kodu içeriyor.

Tüm IBM MQ ortamlarında, DLQ üzerindeki iletileri işlemek için düzenli olarak çalışan bir yordam olmalıdır. IBM MQ , **STRMQMDLQ** komutunu kullanarak çağırdığınız *gelmeyen ileti kuyruğu işleyicisi* (DLQ işleyicisi) adlı varsayılan bir yordam sağlar. Kullanıcı tarafından yazılan bir *kural çizelgesi* , DLQ işleyicisine, DLQ üzerindeki iletileri işlemek için yönergeler sağlar. DLQ işleyicisi, DLQ ' daki iletileri kural çizelgesindeki girişlerle eşleştirir. Bir DLQ iletisi kural çizelgesindeki bir girişle eşleştğinde, DLQ işleyicisi o girişle ilişkili işlemi gerçekleştirir.

Yordam

- DLQ işleyicisini çağır

DLQ işleyicisini çağırmak için **STRMQMDLQ** komutunu kullanın. İşlemek istediğiniz DLQ ' ya ve kullanmak istediğiniz kuyruk yöneticisine iki şekilde ad verebilirsiniz:

- Komut isteminden **STRMQMDLQ** ' e parametre olarak. Örneğin:

```
STRMQMDLQ UDLMSGQ(ABC1.DEAD.LETTER.QUEUE) SRCMBR(QRULE) SRCFILE(library/QXTSRC)
MQMNAME(MY.QUEUE.MANAGER)
```

- Kurallar masasında. Örneğin:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE)
```

Not: Kurallar çizelgesi, herhangi bir adı alabilen bir kaynak fiziksel dosya içindeki bir üyedir.

Bu örnekler, varsayılan kuyruk yöneticisinin sahibi olduğu ABC1 . DEAD . LETTER . QUEUE adlı DLQ için geçerlidir.

DLQ ' yu ya da kuyruk yöneticisini gösterildiği gibi belirtmezseniz, kuruluşa ilişkin varsayılan kuyruk yöneticisi, o kuyruk yöneticisine ait DLQ ile birlikte kullanılır.

İlgili kavramlar

[Gönderilmeyen ileti kuyrukları](#)

İlgili görevler

[Teslim edilmemiş iletilerde sorun giderme](#)

Gelmeyen ileti kuyruğu işleyicisi kural çizelgesi, DLQ işleyicisinin IBM i DLQ ' ya gelen iletileri nasıl işleyeceğini tanımlar.

DLQ işleyicisi kural çizelgesi, DLQ işleyicisinin DLQ ' ya gelen iletileri nasıl işleyeceğini tanımlar. Bir kural çizelgesinde iki tip giriş vardır:

- Çizelgedeki isteğe bağlı ilk giriş *denetim verileri* içerir.
- Çizelgedeki diğer tüm girişler, DLQ işleyicisinin izlemesine ilişkin *kurallardır* . Her kural, bir iletinin eşleştirildiği bir *örüntüden* (ileti özellikleri kümesi) ve DLQ ' daki bir ileti belirtilen örüntüyle eşleştiginde yapılacak bir *işlemden* oluşur. Bir kural tablosunda en az bir kural olmalıdır.

Kurallar çizelgesindeki her giriş bir ya da daha çok anahtar sözcükten oluşur.

Denetim verileri

Bu bölümde, DLQ işleyici kuralları çizelgesindeki bir denetim verileri girişine ekleyebileceğiniz anahtar sözcükler açıklanmaktadır. Aşağıdakileri unutmayın:

- Varsa, bir anahtar sözcüğün varsayılan değerinin altı çizilir.
- Düşey çizgi (|) alternatifleri ayırır. Bunlardan yalnızca birini belirtebilirsiniz.
- Tüm anahtar sözcükler isteğe bağlıdır.

INPUTQ (*QueueName* | " (varsayılan)

İşlemek istediğiniz DLQ ' nun adı:

1. **STRMQMDLQ** komutu için parametre olarak belirlediğiniz herhangi bir UDLMMSGQ değeri (ya da *DFT), kurallar çizelgesindeki herhangi bir INPUTQ değerini geçersiz kılar.
2. **STRMQMDLQ** komutu için parametre olarak boş bir UDLMMSGQ değeri belirlerseniz, kurallar çizelgesindeki INPUTQ değeri kullanılır.
3. **STRMQMDLQ** komutu için parametre olarak boş bir UDLMMSGQ değeri ve kurallar çizelgesinde boş bir INPUTQ değeri belirlerseniz, sistemin varsayılan ileti kuyruğu kullanılır.

INPUTQM (*QueueManagerAd* | " (varsayılan)

INPUTQ anahtar sözcüğünde adı belirtilen DLQ ' nun sahibi olan kuyruk yöneticisinin adı.

Bir kuyruk yöneticisi belirtmezseniz ya da kurallar tablosunda INPUTQM (") belirtirseniz, sistem kuruluş için varsayılan kuyruk yöneticisini kullanır.

RETRYINT (*Aralık* | 60 (varsayılan)

DLQ işleyicisinin ilk denemede işlenemeyen ve yinelenen girişimleri istenen DLQ ' daki iletileri yeniden işlemeyi deneyeceği saniye cinsinden aralık. Varsayılan olarak, yeniden deneme aralığı 60 saniyedir.

WAIT (YES (varsayılan) | NO | *nnn*)

DLQ işleyicisinin, işleyebileceği başka ileti olmadığını saptadığında DLQ ' ya daha fazla ileti gelmesini bekleyip beklememesi gerekip gerekmediğini belirler.

EVET

DLQ işleyicisinin süresiz olarak beklemesine neden olur.

HAYIR

DLQ işleyicisinin DLQ ' nun boş olduğunu ya da işleyebileceği ileti içermediğini saptadığında sonlanmasına neden olur.

nnn

DLQ işleyicisinin, kuyruğun boş olduğunu ya da işleyebileceği ileti içermediğini algıladıktan sonra, yeni işin sona ermeden önce *nnn* saniye beklemesini sağlar.

Meşgul DLQ ' lar için WAIT (YES) ve WAIT (NO) ya da WAIT (*nnn*) belirleyin düşük etkinlik düzeyine sahip DLQ ' lar için. DLQ işleyicisinin sona ermesine izin veriliyorsa, tetikleme kullanarak yeniden çağırın.

STRMQMDLQ komutunun giriş deęiřtirgesi olarak DLQ ' nun adını, denetim verilerinin kurallar çizelgesine eklenmesine alternatif olarak girebilirsiniz. Hem kurallar çizelgesinde hem de **STRMQMDLQ** komutuna girişte herhangi bir deęer belirtilirse, **STRMQMDLQ** komutunda belirtilen deęer önceliklidir.

Not: Kurallar çizelgesinde bir denetim verileri giriři varsa, bu giriř çizelgedeki ilk giriř olmalıdır.

IBM i

IBM i üzerinde DLQ kuralları (örüntüler ve işlemler)

IBM i ' in gönderilmeyen ileti kuyruęu kurallarının her birine iliřkin kalıpların ve işlemlerin açıklaması.

Ařaęıda bir DLQ işleyici kuralları çizelgesinden örnek bir kural verilmiřtir:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Bu kural, DLQ işleyicisine, MQPUT ve MQPUT1 engellendięi için DLQ ' ya konan kalıcı iletilerin hedef kuyruęuna teslim edilmesi için 3 girişimde bulunmasını bildirir.

Bu bölümde, bir kurala ekleyebileceğiniz anahtar sözcükler açıklanmaktadır. Ařaęıdakileri unutmayın:

- Varsa, bir anahtar sözcüęün varsayılan deęerinin altı çizilir. Çoęu anahtar sözcük için varsayılan deęer, herhangi bir deęerle eřleşen * (yıldız işareti) deęeridir.
- Düşey çizgi (!) alternatifleri ayırır. Bunlardan yalnızca birini belirtebilirsiniz.
- ACTION dıřındaki tüm anahtar sözcükler isteęe baęlıdır.

Bu kısım, örüntü eřleştirme anahtar sözcüklerinin (DLQ ' daki iletilerin eřleştięi anahtar sözcüklerin) tanımlamasıyla bařlar. Daha sonra, işlem anahtar sözcüklerini (DLQ işleyicisinin eřleşen bir iletiyi nasıl işleyeceęini belirleyen anahtar sözcükleri) açıklar.

IBM i

IBM i üzerinde DLQ örüntü eřleştirme anahtar sözcükleri

Örüntü eřleştirme anahtar sözcükleri bir örnekte açıklanmıřtır. IBM i ileti kuyruęundaki iletilerin eřleştirileceęi deęerleri belirlemek için bu anahtar sözcükleri kullanın. Kalıpla eřleşen tüm anahtar sözcükler isteęe baęlıdır.

APPLIDAT (*ApplIdentityData* | * (varsayılan)

İleti tanımlayıcısı MQMD 'de belirtilen DLQ ' daki iletinin *ApplIdentityData* (Veri) deęeri.

APPLNAME (*PutApplAdi* | * (varsayılan)

DLQ üzerindeki iletinin tanımlayıcısının (MQMD) *PutApplName* alanında belirtildięi řekilde, MQPUT ya da MQPUT1 çağrısını yayınlayan uygulamanın adı.

APPLTYPE (*PutApplTip* | * (varsayılan)

DLQ ' daki iletinin ileti tanımlayıcısında (MQMD) belirtilen *PutApplTip* deęeri.

DESTQ (*QueueName* | * (varsayılan)

İletin gönderildięi ileti kuyruęunun adı.

DESTQM (*QueueManagerAdi* | * (varsayılan)

İletin gönderildięi ileti kuyruęunun kuyruk yöneticisi adı.

FEEDBACK (*Geribildirim* | * (varsayılan)

MsgType deęeri MQMT_REPORT olduęunda, *Geribildirim* raporun doęasını açıklar.

Sembolik adlar kullanabilirsiniz. Örneęin, hedef kuyruklarına varıřlarının doęrulanmasını gerektiren DLQ ' da bu iletileri tanımlamak için MQFB_COA simgesel adını kullanabilirsiniz.

FORMAT (*Biçim* | * (varsayılan)

İletiyi gönderenin ileti verilerinin biçimini tanımlamak için kullandıęı ad.

MSGTYPE (*MsgType* | * (varsayılan)

DLQ ' daki iletinin tipi.

Sembolik adlar kullanabilirsiniz. Örneęin, yanıt gerektiren DLQ ' da bu iletileri tanımlamak için MQMT_REQUEST simgesel adını kullanabilirsiniz.

PERSIST (Kalıcılık|* (varsayılan))

İletinin kalıcılık değeri. (Bir iletinin kalıcı olarak saklanması, kuyruk yöneticisinin yeniden başlatılıp başlatılmayacağını belirler.)

Sembolik adlar kullanabilirsiniz. Örneğin, DLQ ' da kalıcı olan iletileri tanımlamak için MQPER_PERSISTENT sembolik adını kullanabilirsiniz.

NEDEN (ReasonCode|* (varsayılan))

İletinin DLQ ' ya neden konduğunu açıklayan neden kodu.

Sembolik adlar kullanabilirsiniz. Örneğin, hedef kuyrukları dolu olduğu için DLQ ' ya yerleştirilen iletileri tanımlamak için MQRC_Q_FULL sembolik adını kullanabilirsiniz.

REPLYQ (QueueName|* (varsayılan))

DLQ ' daki iletinin ileti tanımlayıcısında (MQMD) belirtilen yanıt kuyruğu adı.

REPLYQM (QueueManagerAdı|* (varsayılan))

REPLYQ anahtar sözcüğünde belirtilen yanıt kuyruğunun kuyruk yöneticisi adı.

USERID (UserIDentifier|* (varsayılan))

İleti tanımlayıcısı MQMD 'de belirtildiği gibi, DLQ' da iletiyi oluşturan kullanıcının kullanıcı kimliği.

IBM i IBM i üzerinde DLQ işlemi anahtar sözcükleri

IBM i teslim mektubu kuyruğundaki eşleşen bir iletinin nasıl işleneceğini belirlemek için bu teslim olmayan ileti kuyruğu işlem anahtar sözcüklerini kullanın.

EYLEM (AT | YOKSAY | YENIDEN DENE | FWD)

DLQ üzerinde bu kuralda tanımlanan kalıpla eşleşen herhangi bir ileti için yapılan işlem.

At

İletinin DLQ ' dan silinmesine neden olur.

IGNORE

İletinin DLQ üzerinde tutulmasına neden olur.

Yeniden deneme

DLQ işleyicisinin iletiyi hedef kuyruğuna koymayı yeniden denemesine neden olur.

FWD

DLQ işleyicisinin iletiyi FWDQ anahtar sözcüğünde belirtilen kuyruğa iletmesine neden olur.

ACTION anahtar sözcüğünü belirtmelisiniz. Bir eylemi gerçekleştirmek için yapılan deneme sayısı RETRY anahtar sözcüğüne göre yönetilir. Denetim verilerinin RETRYINT anahtar sözcüğü, girişimler arasındaki aralığı denetler.

FWDQ (QueueName| & DESTQ | & REPLYQ)

ACTION anahtar sözcüğünü seçtiğinizde iletinin iletildiği ileti kuyruğunun adı.

QueueName

İleti kuyruğunun adı. FWDQ (") geçerli değil.

& DESTQ

Kuyruk adını MQDLH yapısındaki *DestQName* alanından alın.

& REPLYQ

Kuyruk adını, MQMD ileti tanımlayıcısındaki *ReplyToQ* alanından alın.

FWDQ (& REPLYQ) belirten bir kural, boş *ReplyToQ* alanıyla bir iletiyle eşleştğinde, hata iletilerini önlemek için ileti kalıbında REPLYQ (? *) belirtebilirsiniz.

FWDQM (QueueManagerAdı| & DESTQM | & REPLYQM | " (varsayılan))

Bir iletinin iletildiği kuyruğun kuyruk yöneticisi.

QueueManagerAdı

ACTION (FWD) anahtar sözcüğünü seçtiğinizde iletinin iletileceği kuyruğa ilişkin kuyruk yöneticisi adı.

& DESTQM

Kuyruk yöneticisi adını MQDLH yapısındaki *DestQMGrAd* alanından alın.

& REPLYQM

Kuyruk yöneticisi adını, MQMD ileti tanımlayıcısındaki *ReplyToQMGr* alanından alın.

..

Varsayılan değer olan FWDQM ('), yerel kuyruk yöneticisini tanımlar.

HEADER (YES (varsayılan) |NO)

MQDLH ' nin ACTION (FWD) isteğinde bulunduğu bir iletide kalıp kalmayacağı. Varsayılan olarak, MQDLH iletide kalır. HEADER anahtar sözcüğü FWD dışındaki işlemler için geçerli değil.

PUTAUT (DEF (varsayılan) | CTX)

İletilerin DLQ işleyicisi tarafından konması gereken yetki:

DEF

DLQ işleyicisinin kendisine ilişkin yetkiye sahip iletileri koyar.

CTX

İletilerin, ileti bağlamında kullanıcı kimliğinin yetkisiyle konmasına neden olur. PUTAUT (CTX) belirtilirse, diğer kullanıcıların kimliklerini üstlenme yetkiniz olmalıdır.

RETRY (RetryCount|1 (varsayılan))

1-999.999.999 aralığında, denetim verilerinin RETRYINT anahtar sözcüğünde belirlenen aralıkta bir işlem girişiminde bulunma sayısı.

Not: DLQ işleyicisi tarafından belirli bir kuralı gerçekleştirmek için yapılan deneme sayısı, DLQ işleyicisinin yürürlükteki yönetim ortamına özgüdür; sayı yeniden başlatma işlemleri boyunca devam etmez. DLQ işleyicisini yeniden başlatırsanız, kural uygulama girişiminde bulunma sayısı sıfırlanır.

IBM i

IBM i ile ilgili DLQ kural çizelgesi kuralları

IBM i ' in teslim edilmeyen ileti kuyruğu kuralları tablosu, sözdizimi, yapısı ve içerikleriyle ilgili belirli kurallara uymalıdır.

- Bir kural tablosu en az bir kural içermelidir.
- Anahtar sözcükler herhangi bir sırada olabilir.
- Bir anahtar sözcük herhangi bir kurala yalnızca bir kez eklenebilir.
- Anahtar sözcükler büyük ve küçük harfe duyarlı değildir.
- Bir anahtar sözcük ve parametre değeri, diğer anahtar sözcüklerden en az bir boşluk ya da virgülle ayrılmalıdır.
- Bir kuralın başında ya da sonunda ve anahtar sözcükler, noktalama işaretleri ve değerler arasında herhangi bir sayıda boşluk oluşabilir.
- Her kural yeni bir satırda başlamalıdır.
- Taşınabilirlik için, bir satırın önemli uzunluğu 72 karakterden fazla olmamalıdır.
- Bir satırdaki son boşluk olmayan karakter olarak artı işaretini (+) kullanarak, kuralın sonraki satırdaki ilk boş olmayan karakterden devam ettiğini belirtebilirsiniz. Kuralın sonraki satırın başlangıcından itibaren devam ettiğini belirtmek için, satırdaki son boşluk olmayan karakter olarak eksi işaretini (-) kullanın. Anahtar sözcükler ve değiştirgeler içinde devam karakterleri olabilir.

Örneğin:

```
APPLNAME (' ABC+  
D ')
```

'ABCD' ile sonuçlanır.

```
APPLNAME (' ABC-  
D ')
```

' ABC D' ile sonuçlanır.

- Yıldız işaretiyle (*) başlayan açıklama satırları, kurallar çizelgesinin herhangi bir yerinde olabilir.
- Boş satırlar yoksayılr.
- DLQ işleyici kuralları çizelgesindeki her giriş bir ya da daha çok anahtar sözcükle ilişkili değıştirtgelerden oluşur. Parametreler řu sözdizimi kurallarına uymalıdır:

- Her parametre değeri en az bir anlamlı karakter içermelidir. Tırnak içine alınan değerlerdeki sınırlayıcı tırnak işaretleri önemli olarak kabul edilmez. Örneğın, bu parametreler geçerlidir:

FORMAT (' ABC ')	3 önemli karakter
FORMAT (ABC)	3 önemli karakter
FORMAT (' A ')	1 önemli karakter
FORMAT (A)	1 önemli karakter
FORMAT (' ')	1 önemli karakter

Bu parametreler önemli karakterler içermediğinden geçersiz:

FORMAT (' ')
FORMAT ()
FORMAT ()
FORMAT

- Genel arama karakterleri desteklenir. Soru işaretini (?), sondaki boşluk dışında herhangi bir tek karakter yerine kullanabilirsiniz. Sıfır ya da daha fazla bitişik karakter yerine yıldız işaretini (*) kullanabilirsiniz. Yıldız işareti (*) ve soru işareti (?), parametre değerlerinde genel arama karakteri olarak **her zaman** yorumlanır.
- řu anahtar sözcüklerin parametrelerine genel arama karakteri ekleyemezsiniz: ACTION, HEADER, RETRY, FWDQ, FWDQM ve PUTAUT.
- Parametre değerlerinde ve DLQ ' daki iletide karşılık gelen alanlarda sondaki boşluklar, genel arama karakteri eşleşmeleri gerçekleştirilirken önemli değildir. Ancak, tırnak işareti içindeki dizgilerin başındaki ve içindeki boşluklar genel arama karakteri eşleşmeleri için önemlidir.
- Sayısal parametreler soru işareti (?) genel arama karakteri içerebilir. Tüm sayısal parametrenin yerine yıldız işaretini (*) ekleyebilirsiniz, ancak yıldız işareti sayısal parametrenin bir parçası olarak eklenemez. Örneğın, bunlar geçerli sayısal parametrelerdir:

MSGTYPE (2)	Yalnızca yanıt iletileri seçilebilir
MSGTYPE (*)	Herhangi bir ileti tipi seçilebilir
MSGTYPE (' * ')	Herhangi bir ileti tipi seçilebilir

Ancak MSGTYPE (' 2* ') , sayısal bir parametrenin parçası olarak yıldız işareti (*) içerdğinden geçerli değildir.

- Sayısal parametreler 0-999 999 999 aralğında olmalıdır. Parametre değeri bu aralıktaysa, anahtar sözcüğün ilgili olduđu alanda geçerli olmasa da kabul edilir. Sayısal parametreler için simgesel adlar kullanabilirsiniz.
- Bir dizgi değeri, anahtar sözcüğün ilişkili olduđu MQDLH ya da MQMD ' deki alandan kısaysa, değeri alanın uzunluğuyla boşluklarla doldurulur. Yıldız işaretleri dışında değeri alandan uzunsa, bir hata tanısı konur. Örneğın, bunların tümü 8 karakterlik bir alan için geçerli dizgi değerleridir:

' ABCDEFGH '	8 karakter
' A*C*E*G*I '	Yıldız işaretleri hariç 5 karakter
' *A*C*E*G*I*K*M*O* '	Yıldız işaretleri dışında 8 karakter

- Boşluk, küçük harf ya da nokta (.), eğik çizgi (?), alt çizgi (_) ve yüzde imi (%) dışında özel karakterler içeren dizgiler tek tırnak işareti içine alınmalıdır. Tırnak içine alınmamış küçük harfler büyük harfe katlanır. Dizgi bir tırnak işareti içeriyorsa, tırnak işaretinin başlangıcını ve sonunu belirtmek için iki tek tırnak işareti kullanılmalıdır. Dizginin uzunluğu hesaplandığında, çift tırnak işaretlerinin her geçişi tek bir karakter olarak sayılır.

IBM i DLQ kural çizelgesinin IBM i üzerinde nasıl işlendiği

Teslim etmeyen ileti kuyruğu işleyicisi, IBM i ileti kuyruğundaki bir iletiyle eşleşen bir örüntüye sahip kural için kural çizelgesinde arama yapar.

Arama, tablodaki ilk kuralla başlar ve tablo boyunca sırayla devam eder. Eşleşen bir örüntüye sahip bir kural bulunduğunda, kural çizelgesi o kuraldaki işlemi dener. DLQ işleyicisi, bir kuralı uygulama girişiminde bulunduğunda, o kural için yeniden deneme sayısını 1 artırır. İlk girişim başarısız olursa, yapılan deneme sayısı RETRY anahtar sözcüğünde belirtilen sayıyla eşleşinceye kadar girişim yinelenir. Tüm girişimler başarısız olursa, DLQ işleyicisi çizelgede sonraki eşleşen kuralı arar.

Bu işlem, bir işlem başarılı oluncaya kadar sonraki eşleştirme kuralları için yinelenir. Her eşleşen kural RETRY anahtar sözcüğünde belirtilen sayıda denendiğinde ve tüm girişimler başarısız olduğunda, ACTION (IGNORE) varsayılır. Eşleşen bir kural bulunamazsa, ACTION (IGNORE) da varsayılır.

Not:

1. Eşleşen kural örüntüleri yalnızca, DLQ ' da MQLDH ile başlayan iletiler için aranır. MQLDH ile başlamayan iletiler belirli aralıklarla hatalı olarak bildirilir ve DLQ üzerinde süresiz olarak kalır.
2. Bir kuralın yalnızca bir işlemde oluşması için, tüm örüntü anahtar sözcükleri varsayılan olarak kullanılabilir. Ancak, yalnızca işlem kurallarının kuyruktaki MQLDH ' leri olan ve çizelgedeki diğer kurallara uygun olarak işlenmemiş tüm iletilere uygulandığını unutmayın.
3. DLQ işleyicisi başlatıldığında kural çizelgesinin geçerliliği denetlenir ve hatalar o sırada işaretlenir. (DLQ işleyicisi tarafından yayınlanan hata iletileri İletiler ve neden kodları içinde açıklanmıştır.) Kural çizelgesinde istediğiniz zaman değişiklik yapabilirsiniz; ancak, DLQ işleyicisi yeniden başlatılınca kadar bu değişiklikler yürürlüğe girmez.
4. DLQ işleyicisi, iletilerin, MQLDH ' nin ya da ileti tanımlayıcısının içeriğini değiştirmez. DLQ işleyicisi iletileri her zaman **MQPMO_PASS_ALL_CONTEXT** ileti seçeneğiyle diğer kuyruklara koyar.
5. Kurallar çizelgesinin geçerliliğinin denetlenmesi yinelenen hataların oluşturulmasını ortadan kaldırdığından, kural çizelgesindeki ardışık sözdizimi hataları tanınmayabilir.
6. DLQ işleyicisi DLQ ' yu **MQOO_INPUT_AS_Q_DEF** seçeneğiyle açar.
7. DLQ işleyicisinin birden çok eşgörünümlü, aynı kural çizelgesini kullanarak aynı kuyruk için eşzamanlı olarak çalışabilir. Ancak, DLQ ile DLQ işleyicisi arasında bire bir ilişki olması daha olağandır.

IBM i IBM i üzerinde örnek bir DLQ işleyici kuralları çizelgesi

IBM üzerindeki bir teslim olmayan ileti kuyruğu işleyici kuralları tablosuna ilişkin örnek kod. Bu örnek kural tablosu, tek bir denetim verileri girdisi ve birkaç kural içerir.

```
*****
*   An example rules table for the STRMQMDLQ command           *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* STRMQMDLQ, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to STRMQMDLQ,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.
```

```

* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation is always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never does things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2

DESTQM(bbbb.1) +
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)

* The CCCC corporation considers itself very security
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.

REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)

* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it must be able
* to cope with the message being lost, so we can afford to
* discard the message.

PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)

* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We do not have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.

REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)

ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)

```

Tüm DLQ iletilerinin işlendiğinden emin olunması

Gitmeyen ileti kuyruğu işleyicisi, DLQ üzerinde görülen, ancak kaldırılmayan tüm iletilerin kaydını tutar. DLQ ' nun mümkün olan en az sayıda ileti içerdiğinden emin olun.

Bu görev hakkında

DLQ işleyicisini, DLQ 'dan gelen iletilerin küçük bir altkümesini almak için süzgeç olarak kullanırsanız, DLQ işleyicisi işlemediği DLQ' da bu iletilerin kaydını tutar. Ayrıca DLQ işleyicisi, DLQ ilk giren ilk çıkar (FIFO)

olarak tanımlanmış olsa da, DLQ ' ya gelen yeni iletilerin görüldüğünü garanti edemez. Kuyruk boş değilse, tüm iletileri denetlemek için DLQ düzenli aralıklarla yeniden taranır.

Bu nedenle, DLQ ' nun mümkün olan en az sayıda ileti içerdiğinden emin olmanız gerekir. Atılmayan ya da başka kuyruklara iletilmeyen iletilerin (herhangi bir nedenle) kuyrukta birikmesine izin verilirse, DLQ işleyicisinin iş yükü artar ve DLQ ' nun kendisi doldurma tehlikesiyle karşı karşıyadır.

DLQ işleyicisinin DLQ ' yu boşaltmasını sağlamak için aşağıdaki önlemleri alın:

Yordam

- Aksi takdirde yoksayacağınız iletiler için, iletileri başka bir kuyruğa taşıyan bir işlem kullanın.

DLQ üzerinde ileti bırakan **ACTION (IGNORE)** komutunu kullanmamaya çalışın ve çizelgedeki diğer kuralların belirttik olarak ele almadığı iletiler için **ACTION (IGNORE)** ' in varsayıldığını unutmayın. Bunun yerine, iletileri başka bir kuyruğa taşıyan bir işlem kullanın. Örneğin:

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

- Çizelgedeki önceki kuralların ele almadığı iletileri işlemek için, çizelgedeki son kuralı "tümünü yakala" yapın.

Örneğin, tablodaki son kural şöyle olabilir:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Bu, çizelgedeki son kurala denk gelen iletileri, el ile işlenebilecekleri REALLY . DEAD . QUEUEkuyruğuna iletir. Böyle bir kuralınız yoksa, iletilerin DLQ ' da süresiz olarak kalması olasıdır.

Yönetimle ilgili konularla çalışma

Denetim konularını yönetmek için MQSC komutlarını kullanın.

Bu komutlarla ilgili ayrıntılı bilgi için [MQSC komutları](#) konusuna bakın.

İlgili kavramlar

[Yönetim konusu nesneleri](#)

İlgili görevler

[“Denetim konusu tanımlanması” sayfa 191](#)

Denetim konusu yaratmak için **DEFINE TOPIC** MQSC komutunu kullanın. Bir yönetim konusu tanımlarken isteğe bağlı olarak her konu özniteliğini ayarlayabilirsiniz.

[“Yönetim konusu nesnesi özniteliklerinin görüntülenmesi” sayfa 192](#)

Bir denetim konusu nesnesini görüntülemek için **DISPLAY TOPIC** MQSC komutunu kullanın.

[“Yönetim konusu özniteliklerini değiştirme” sayfa 193](#)

Konu özniteliklerini, **ALTER TOPIC** komutunu ya da **DEFINE TOPIC** komutunu **REPLACE** özniteliğiyle kullanarak iki şekilde değiştirebilirsiniz.

[“Denetim konusu tanımlamasının kopyalanması” sayfa 194](#)

DEFINE komutundaki **LIKE** özniteliğini kullanarak bir konu tanımlamasını kopyalayabilirsiniz.

[“Denetim konusu tanımlamasının silinmesi” sayfa 194](#)

Bir denetim konusunu silmek için **DELETE TOPIC** MQSC komutunu kullanabilirsiniz.

Denetim konusu tanımlanması

Denetim konusu yaratmak için **DEFINE TOPIC** MQSC komutunu kullanın. Bir yönetim konusu tanımlarken isteğe bağlı olarak her konu özniteliğini ayarlayabilirsiniz.

Başlamadan önce

Not: Bu görevdeki örnek, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu görev hakkında

Konunun belirtik olarak ayarlanmayan herhangi bir özniteliği, SYSTEM.DEFAULT.TOPIC, sistem kuruluşu kurulduğunda oluşturulmuştur.

Örnek

Örneğin, aşağıdaki **DEFINE TOPIC** komutu, aşağıdaki özelliklere sahip ORANGE . TOPIC adlı bir konuyu tanımlar:

- ORANGE konu dizgisine çözülür. Konu dizgilerinin nasıl kullanılabileceğiyle ilgili bilgi için [Konu dizgilerini birleştirme](#) başlıklı konuya bakın.
- ASPARENT olarak ayarlanan herhangi bir öznitelik, bu konunun üst konusu tarafından tanımlandığı şekilde özniteliği kullanır. Bu işlem, SYSTEM.BASE.TOPIC bulundu. Daha fazla bilgi için bkz. [Konu ağaçları](#).

```
DEFINE TOPIC (ORANGE.TOPIC) +  
TOPICSTR (ORANGE) +  
DEFPRTY (ASPARENT) +  
NPMSGDLV (ASPARENT)
```

Not:

- Konu dizgisinin değeri dışında, gösterilen tüm öznitelik değerleri varsayılan değerlerdir. Burada yalnızca bir resim olarak gösterilirler. Varsayılan değerlerin istediğiniz gibi olduğundan ya da değiştirilmediğinden eminseniz bunları atlayabilirsiniz. Ayrıca bkz. [“Yönetim konusu nesnesi özniteliklerinin görüntülenmesi” sayfa 192](#).
- ORANGE.TOPIC, bu komut başarısız olur. Bir konunun varolan tanımlamasının üzerine yazılmasını istiyorsanız REPLACE özniteliğini kullanın, ancak [“Yönetim konusu özniteliklerini değiştirme” sayfa 193](#) başlıklı konuya bakın.

İlgili başvurular

[KONUYU TANIMLAYIN](#)

Yönetim konusu nesnesi özniteliklerinin görüntülenmesi

Bir denetim konusu nesnesini görüntülemek için **DISPLAY TOPIC** MQSC komutunu kullanın.

Başlamadan önce

Not: Bu görevdeki örnekler, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Örnek

Bu komut tüm konuları görüntüler:

```
DISPLAY TOPIC (ORANGE.TOPIC)
```

Öznitelikleri **DISPLAY TOPIC** komutuyla tek tek belirterek seçmeli olarak görüntüleyebilirsiniz. Örneğin:

```
DISPLAY TOPIC (ORANGE.TOPIC) +  
TOPICSTR +  
DEFPRTY +  
NPMSGDLV
```


Bu komut, belirtilen üç özniteliği görüntüler:

```
AMQ8633: Display topic details.
TOPIC(ORANGE.TOPIC)
TOPICSTR(ORANGE)
NPMGDLV(ASPARENT)
TYPE(LOCAL)
DEFPRTY(ASPARENT)
```

Yürütme sırasında kullanılan ASPARENT değerlerini görüntülemek için **DISPLAY TPSTATUS** komutunu kullanın. Örneğin, şunu kullanın:

```
DISPLAY TPSTATUS(ORANGE) DEFPRTY NPMGDLV
```

Komut aşağıdaki ayrıntıları görüntüler:

```
AMQ8754: Display topic status details.
TOPICSTR(ORANGE) DEFPRTY(0)
NPMGDLV(ALLAVAIL)
```

Bir denetim konusu tanımladığınızda, varsayılan denetim konusundan belirtik olarak belirtmediğiniz öznitelikler (SYSTEM.DEFAULT.TOPIC. Bu varsayılan özniteliklerin ne olduğunu görmek için aşağıdaki komutu kullanın:

```
DISPLAY TOPIC (SYSTEM.DEFAULT.TOPIC)
```

İlgili başvurular

[KONU GÖRÜNTÜLE](#)

[TPSTATÜYÜ GÖRÜNTÜLE](#)

Yönetim konusu özniteliklerini değiştirme

Konu özniteliklerini, **ALTER TOPIC** komutunu ya da **DEFINE TOPIC** komutunu **REPLACE** özniteliğiyle kullanarak iki şekilde değiştirebilirsiniz.

Başlamadan önce

Not: Bu görevdeki örnekler, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Örnek

Örneğin, ORANGE.TOPIC, 5 yaşında olmak için aşağıdaki komutlardan birini kullanın:

- **ALTER** komutunu kullanarak:

```
ALTER TOPIC(ORANGE.TOPIC) DEFPRTY(5)
```

Bu komut, bu konuya teslim edilen iletinin varsayılan önceliği olan tek bir özniteliği 5 olarak değiştirir; diğer tüm öznitelikler aynı kalır.

- **DEFINE** komutunu kullanarak:

```
DEFINE TOPIC(ORANGE.TOPIC) DEFPRTY(5) REPLACE
```

Bu komut, bu konuya gönderilen iletilerin varsayılan önceliğini değiştirir. Diğer tüm özniteliklere varsayılan değerleri verilir.

Bu konuya gönderilen iletilerin önceliğini değiştirirseniz, var olan iletiler etkilenmez. Ancak, herhangi bir yeni ileti, yayınlama uygulaması tarafından sağlanmamışsa, belirtilen önceliği kullanır.

İlgili başvurular

[KONUYU DEĞİŞTİR](#)

[KONU GÖRÜNTÜLE](#)

Denetim konusu tanımlamasının kopyalanması

DEFINE komutundaki LIKE özniteliğini kullanarak bir konu tanımlamasını kopyalayabilirsiniz.

Başlamadan önce

Not: Bu görevdeki örnekler, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Örnek

Aşağıdaki komut MAGENTA.TOPIC, özgün konu ile aynı özniteliklere sahip ORANGE.TOPIC(Konu). Konuyu oluşturduğunuzda tam olarak girildiği şekilde kopyalanacak konunun adını girin. Ad küçük harfli karakterler içeriyorsa, adı tek tırnak içine alın.

```
DEFINE TOPIC (MAGENTA.TOPIC) +  
LIKE (ORANGE.TOPIC)
```

DEFINE komutunun bu biçimini kullanarak bir konu tanımlamasını kopyalayabilir, ancak özgün başlık tanımlamasında değişiklik yapabilirsiniz. Örneğin:

```
DEFINE TOPIC (BLUE.TOPIC) +  
TOPICSTR (BLUE) +  
LIKE (ORANGE.TOPIC)
```

BLUE.TOPIC GREEN.TOPIC ve yayınlar doğru abone kuyruğuna teslim edilemediğinde, bunların gitmeyen iletiler kuyruğuna yerleştirilmeyeceğini belirtin. Örneğin:

```
DEFINE TOPIC (GREEN.TOPIC) +  
TOPICSTR (GREEN) +  
LIKE (BLUE.TOPIC) +  
USEDLQ (NO)
```

İlgili başvurular

[KONUYU TANIMLAYIN](#)

Denetim konusu tanımlamasının silinmesi

Bir denetim konusunu silmek için **DELETE TOPIC** MQSC komutunu kullanabilirsiniz.

Başlamadan önce

Not: Bu görevdeki örnek, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Örnek

```
DELETE TOPIC (ORANGE.TOPIC)
```

Uygulamalar artık konuyu yayınlamak üzere açamaz ya da ORANGE.TOPIC. Konu açık olan yayınlama uygulamaları, çözülen konu dizgisini yayınlamaya devam edebilir. Bu konuya önceden yapılan abonelikler, konu silindikten sonra yayınları almaya devam eder.

Bu konu nesnesine başvurmayan, ancak bu konu nesnesinin temsil ettiği çözülmüş konu dizisini kullanan uygulamalar (bu örnekte 'TURUNCU') çalışmaya devam eder. Bu durumda, özellikleri konu ağacında daha yüksek bir konu nesnesinden edinirler. Daha fazla bilgi için bkz. [Konu ağaçları](#).

İlgili başvurular

[KONU YU SIL](#)

Aboneliklerle çalışma

Abonelikleri yönetmek için MQSC komutlarını kullanın.

Bu görev hakkında

Abonelikler, **SUBTYPE** özniteliğinde tanımlanan üç tipten biri olabilir:

ADMIN

Yönetici olarak bir kullanıcı tarafından tanımlanır.

proxy

Kuyruk yöneticileri arasında yayınları yönlendirmek için dahili olarak oluşturulan abonelik.

Uygulama Programlama Arabirimi

Örneğin, MQI MQSUB çağırısı kullanılarak programlı olarak yaratıldı.

Bu komutlarla ilgili ayrıntılı bilgi için [MQSC komutları](#) konusuna bakın.

Yönetim aboneliği tanımlanması

Denetim aboneliği yaratmak için **DEFINE SUB** MQSC komutunu kullanın. Varsayılan yerel abonelik tanımlanmasında tanımlanan varsayılan değeri de kullanabilirsiniz. Ya da varsayılan yerel aboneliğin (SYSTEM.DEFAULT.SUB).

Başlamadan önce

Not: Bu görevdeki örnekler, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Örnek

Aşağıdaki **DEFINE SUB** komutu, aşağıdaki özelliklere sahip ORANGE adlı bir aboneliği tanımlar:

- Sürekli abonelik, sınırsız süre bitimiyle kuyruk yöneticisini yeniden başlatma işlemi boyunca kalıcı olduğu anlamına gelir.
- Yayınlama uygulamaları tarafından belirlenen ileti öncelikleriyle birlikte, ORANGE konu dizisiyle ilgili yapılan yayınları alın.
- Bu abonelik için teslim edilen yayınlar yerel kuyruk SUBQ 'ya gönderilir, bu kuyruk abonelik tanımından önce tanımlanmalıdır.

```
DEFINE SUB (ORANGE) +
TOPICSTR (ORANGE) +
DESTCLAS (PROVIDED) +
DEST (SUBQ) +
EXPIRY (UNLIMITED) +
PUBPRTY (AS PUB)
```

Not:

- Abonelik ve konu dizisi adının eşleşmesi gerekmez.
- Hedef ve konu dizisinin değerleri dışında, gösterilen tüm öznitelik değerleri varsayılan değerlerdir. Burada yalnızca bir resim olarak gösterilirler. Varsayılan değerlerin istediğiniz gibi olduğundan ya da değiştirilmediğinden eminseniz bunları atlayabilirsiniz. Ayrıca bkz. [“Aboneliklerin özniteliklerini görüntüleme” sayfa 196](#).

- Aynı kuyruk yöneticisinde ORANGE adlı bir yerel aboneliğiniz varsa, bu komut başarısız olur. Bir kuyruğun var olan tanımının üzerine yazılmasını istiyorsanız **REPLACE** özniteliğini kullanın, ancak ayrıca bkz. [“Yerel abonelik özniteliklerini değiştirme” sayfa 197](#).
- Kuyruk SUBQ yoksa, bu komut başarısız olur.

İlgili başvurular

[ALT öğeyi TAN](#)

Aboneliklerin özniteliklerini görüntüleme

Kuyruk yöneticisi tarafından bilinen herhangi bir aboneliğin yapılandırılmış özniteliklerini görüntülemek için **DISPLAY SUB** komutunu kullanabilirsiniz.

Başlamadan önce

Not: Bu görevdeki örnekler, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Örnek

```
DISPLAY SUB(ORANGE)
```

Öznitelikleri tek tek belirterek seçmeli olarak görüntüleyebilirsiniz. Örneğin:

```
DISPLAY SUB(ORANGE) +  
SUBID +  
TOPICSTR +  
DURABLE
```

Bu komut, belirtilen üç özniteliği aşağıdaki gibi görüntüler:

```
AMQ8096: IBM MQ subscription inquired.  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
SUB(ORANGE) TOPICSTR(ORANGE)  
DURABLE(YES)
```

TOPICSTR, bu abonenin çalıştığı çözülmüş konu dizgisidir. Bir abonelik bir konu nesnesini kullanacak şekilde tanımlandığında, o nesnedeki konu dizgisi, abonelik oluşturulurken sağlanan konu dizgisinin öneki olarak kullanılır. SUBID, bir abonelik yaratıldığında kuyruk yöneticisi tarafından atanan benzersiz bir tanıttıcıdır. Bu, bazı abonelik adları uzun olabileceği için ya da kullanışsız olabileceği farklı bir karakter kümesinde görüntülenebilecek yararlı bir özniteliktir.

Abonelikleri görüntülemek için kullanılacak diğer bir yöntem SUBID ' yi kullanmaktır:

```
DISPLAY SUB +  
SUBID(414D51204141412020202020202020EE921E4E20002A03) +  
TOPICSTR +  
DURABLE
```

Bu komut öncekiyle aynı çıkışı verir:

```
AMQ8096: IBM MQ subscription inquired.  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
SUB(ORANGE) TOPICSTR(ORANGE)  
DURABLE(YES)
```

Bir kuyruk yöneticisindeki yetkili sunucu abonelikleri varsayılan olarak görüntülenmez. Bunları görüntülemek için PROXY ya da ALL için bir **SUBTYPE** belirtin.

Çalıştırma zamanı özniteliklerini görüntülemek için `DISPLAY SBSTATUS` komutunu kullanabilirsiniz. Örneğin, şu komutu kullanın:

```
DISPLAY SBSTATUS(ORANGE) NUMMSGs
```

Aşağıdaki çıktı görüntülenir:

```
AMQ8099: IBM MQ subscription status inquired.  
SUB(ORANGE)  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
NUMMSGs(0)
```

Bir yönetim aboneliği tanımladığınızda, varsayılan abonelikten belirttik olarak belirtmediğiniz öznitelikler (SYSTEM.DEFAULT.SUB. Bu varsayılan özniteliklerin ne olduğunu görmek için aşağıdaki komutu kullanın:

```
DISPLAY SUB (SYSTEM.DEFAULT.SUB)
```

İlgili başvurular

[ALT öğEYI Gö](#)

Yerel abonelik özniteliklerini değiştirme

Abonelik özniteliklerini, **ALTER SUB** komutunu ya da **DEFINE SUB** komutunu **REPLACE** özniteliğiyle kullanarak iki şekilde değiştirebilirsiniz.

Başlamadan önce

Not: Bu görevdeki örnekler, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Örnek

ORANGE adlı bir aboneliğe gönderilen iletilerin önceliğini 5 olarak değiştirmek istiyorsanız, aşağıdaki komutlardan birini kullanın:

- **ALTER** komutunu kullanarak:

```
ALTER SUB(ORANGE) PUBPRTY(5)
```

Bu komut, bu aboneliğe teslim edilen iletilerin önceliği olan tek bir özniteliği 5 'e değiştirir; diğer tüm öznitelikler aynı kalır.

- **DEFINE** komutunu kullanarak:

```
DEFINE SUB(ORANGE) PUBPRTY(5) REPLACE
```

Bu komut, yalnızca bu aboneliğe teslim edilen iletilerin önceliğini değil, varsayılan değerleri verilen diğer tüm öznitelikleri de değiştirir.

Bu aboneliğe gönderilen iletilerin önceliğini değiştirirseniz, var olan iletiler etkilenmez. Ancak, yeni iletiler belirtilen önceliğe sahiptir.

İlgili başvurular

[ALTER SUB](#)

[ALT öğEYI TAN](#)

Yerel abonelik tanımlamasının kopyalanması

DEFINE komutundaki **LIKE** özniteliğini kullanarak bir abonelik tanımlamasını kopyalayabilirsiniz.

Başlamadan önce

Not: Bu görevdeki örnekler, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Örnek

```
DEFINE SUB(BLUE) +  
LIKE(ORANGE)
```

Alt REAL ' in özniteliklerini THIRD.SUBalt bölümüne kopyalayabilir ve teslim edilen yayınların **correlID** yayıncılarının değil, ÜÇÜNCÜSÜ olduğunu belirtebilirsiniz **correlID**. Örneğin:

```
DEFINE SUB(THIRD.SUB) +  
LIKE(BLUE) +  
DESTCORL(ORANGE)
```

İlgili başvurular

[ALT ÖĞEYİ TAN](#)

Yerel aboneliği silme

Yerel aboneliği silmek için **DELETE SUB** MQSC komutunu kullanabilirsiniz.

Başlamadan önce

Not: Bu görevdeki örnekler, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Örnek

```
DELETE SUB(ORANGE)
```

Bir aboneliği SUBID ' yi kullanarak da silebilirsiniz:

```
DELETE SUB SUBID(414D5120414141202020202020202020EE921E4E20002A03)
```

İlgili başvurular

[ALT ÖĞEYİ SIL](#)

Abonelikteki iletilerin denetlenmesi

Bir abonelik tanımlandığında, bir kuyrukla ilişkilendirilir. Bu abonelikte eşleşen yayınlanan iletiler bu kuyruğa yerleştirilir. Bir abonelik için kuyruğa alınmış iletileri denetlemek üzere MQSC komutlarını kullanın.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu görev hakkında

Aşağıdaki MQSC komutlarının yalnızca ileti alan abonelikleri gösterdiğini unutmayın.

Bir abonelik için kuyruğa alınmış iletileri denetlemek üzere aşağıdaki adımları izleyin:

Yordam

1. DISPLAY SBSTATUS(*sub_name*) NUMMSGSabonelik tipi için kuyruğa alınan iletileri denetlemek üzere bkz. [“Aboneliklerin özniteliklerini görüntüleme” sayfa 196.](#)
2. NUMMSGs değeri sıfırdan büyükse, DISPLAY SUB(*sub_name*) DEST yazarak abonelikle ilişkili kuyruğu tanımlayın.
3. Döndürülen kuyruğun adını kullanarak, iletileri [“Örnek programla kuyruklara göz atılması” sayfa 162](#) içinde açıklanan tekniği izleyerek görüntüleyebilirsiniz.

İlgili başvurular

[SBSTATUS DURUMUNU GÖRÜNTÜLE](#)

Hizmetlerle çalışma

Hizmet nesneleri, ek işlemlerin bir kuyruk yöneticisinin parçası olarak yönetilebileceği bir araçtır. Hizmetlerle, kuyruk yöneticisi başlatıldığında ve sona erdiğinde başlatılıp durdurulan programları tanımlayabilirsiniz. IBM MQ hizmetleri her zaman kuyruk yöneticisini başlatan kullanıcının kullanıcı kimliği altında başlatılır.

Bu görev hakkında

Hizmet nesneleri aşağıdaki tiplerden biri olabilir:

Sunucu

Sunucu, **SERVTYPE** değiştirgesinin SERVER olarak belirtildiği bir hizmet nesnesidir. Sunucu hizmet nesnesi, belirtilen bir kuyruk yöneticisi başlatıldığında yürütülen bir programın tanımınıdır. Sunucu hizmeti nesnelere, genellikle uzun süre çalışan programları tanımlar. Örneğin, bir sunucu hizmeti nesnesi, **runmqtrm** gibi bir tetikleyici izleme işlemini yürütmek için kullanılabilir.

Bir sunucu hizmeti nesnesinin yalnızca bir örneği eşzamanlı olarak çalıştırılabilir. Çalışmakta olan sunucu hizmeti nesnelere durumu, **DISPLAY SVSTATUSMQSC** komutu kullanılarak izlenebilir.

Komut

Komut, **SERVTYPE** değiştirgesinin COMMAND olarak belirtildiği bir hizmet nesnesidir. Komut hizmeti nesnelere sunucu hizmeti nesnelere benzer, ancak bir komut hizmeti nesnesinin birden çok eşgörünümü eşzamanlı olarak çalışabilir ve durumları **DISPLAY SVSTATUSMQSC** komutu kullanılarak izlenemez.

MQSC komutu (**STOP SERVICE**) yürütülürse, programı durdurmadan önce MQSC komutu (**START SERVICE**) tarafından başlatılan programın etkin olup olmadığını saptamak için denetim yapılmaz.

İlgili başvurular

[HİZMETİ TANIMLAYIN](#)

[SVSTATUS DURUMUNU GÖRÜNTÜLE](#)

[Hizmeti başlat](#)

[Hizmeti durdur](#)

Hizmet nesnesi tanımlanması

DEFINE SERVICEMQSC komutuyla bir hizmet nesnesi tanımlarsınız.

Başlamadan önce

Not: Bu görev, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme.](#)

Yordam

- **DEFINE SERVICE**MQSC komutuyla bir hizmet nesnesi tanımlayın.

Tanımlamanız gereken öznitelikler şunlardır:

HİZMET TÜRÜ

Hizmet nesnesinin tipini tanımlar. Olası değerler şunlardır:

SERVER

Bir sunucu hizmet nesnesi.

Bir kerede tek bir sunucu hizmeti nesnesi eşgörünümü yürütülebilir. Sunucu hizmeti nesnelerinin durumu, **DISPLAY SVSTATUSMQSC** komutu kullanılarak izlenebilir.

Komut

Bir komut hizmeti nesnesi.

Bir komut hizmeti nesnesinin birden çok eşgörünümü eşzamanlı olarak yürütülebilir. Bir komut hizmeti nesnelerinin durumu izlenemiyor.

STARTCMD

Hizmeti başlatmak için yürütülen program. Programın tam olarak nitelenmiş bir yolu belirlenmelidir.

TARTARG

Bağımsız değişkenler başlangıç programına iletildi.

STDERR

Hizmet programının standart hatasının (stderr) yeniden yönlendirileceği dosyanın yolunu belirtir.

KESTIK

Hizmet programının standart çıkışının (stdout) yeniden yönlendirileceği dosyanın yolunu belirler.

DURDUR PCMD

Hizmeti durdurmak için yürütülen program. Programın tam olarak nitelenmiş bir yolu belirlenmelidir.

DURDUR

Durdurma programına geçirilen bağımsız değişkenler.

CONTROL

Hizmetin nasıl başlatılacağını ve durdurulacağını belirtir:

El İle

Hizmet otomatik olarak başlatılmayacak ya da otomatik olarak durdurulmayacak. **START SERVICE** ve **STOP SERVICE** komutlarının kullanımıyla denetlenir. Bu varsayılan değerdir.

QMGR

Tanımlanmakta olan hizmet, kuyruk yöneticisi başlatılıp durdurulduğunda başlatılacak ve durdurulacaktır.

STARTONLY

Hizmet, kuyruk yöneticisiyle aynı anda başlatılacak, ancak kuyruk yöneticisi durdurulduğunda durdurulması istenmeyecek.

İlgili görevler

“Hizmetleri yönetme” sayfa 200

Bir hizmet nesnesinin eşgörünümü kuyruk yöneticisi tarafından otomatik olarak başlatılabilir ve durdurulabilir ya da MQSC komutları **START SERVICE** ve **STOP SERVICE** kullanılarak başlatılabilir ve durdurulabilir.

İlgili başvurular

[HİZMETİ TANIMLAYIN](#)

[SVSTATUS DURUMUNU GÖRÜNTÜLE](#)

[Hizmeti başlat](#)

[Hizmeti durdur](#)

Hizmetleri yönetme

Bir hizmet nesnesinin eşgörünümü kuyruk yöneticisi tarafından otomatik olarak başlatılabilir ve durdurulabilir ya da MQSC komutları **START SERVICE** ve **STOP SERVICE** kullanılarak başlatılabilir ve durdurulabilir.

Başlamadan önce

Not: Bu görev, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Yordam

- Kuyruk yöneticisinde **CONTROL** değıştirgesini bir hizmet nesnesinin bir eşgörünümünü otomatik olarak başlatmaya da durduracak şekilde ayarlayın ya da bunu el ile gerçekleştirmek için **START SERVICE** ve **STOP SERVICE** MQSC komutlarını kullanın.

Bir hizmet nesnesi eşgörünümü başlatıldığında, kuyruk yöneticisi hata günlüğüne hizmet nesnesinin adını ve başlatılan işlemin işlem tanıtıcısını içeren bir ileti yazılır. Sunucu hizmeti nesnesinin başlatılmasını gösteren örnek bir günlük girişi:

```
02/15/2005 11:54:24 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5028: The Server 'S1' has started. ProcessId(13031).

EXPLANATION:
The Server process has started.
ACTION:
None.
```

Aşağıda, bir komut hizmeti nesnesi için başlatılacak örnek bir günlük girişi yer almaktadır:

```
02/15/2005 11:53:55 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5030: The Command 'C1' has started. ProcessId(13030).

EXPLANATION:
The Command has started.
ACTION:
None.
```

Bir yönetim ortamı sunucusu hizmeti durduğunda, kuyruk yöneticisi hata günlüklerine hizmetin adını ve bitiş işleminin işlem tanıtıcısını içeren bir ileti yazılır. Aşağıda, bir sunucu hizmeti nesnesinin durdurulması için örnek bir günlük girişi verilmiştir:

```
02/15/2005 11:54:54 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5029: The Server 'S1' has ended. ProcessId(13031).

EXPLANATION:
The Server process has ended.
ACTION:
None.
```

İlgili görevler

[“service.env dosyasında ek ortam değışkenlerinin tanımlanması” sayfa 202](#)

Bir hizmet başlatıldığında, hizmet işleminin başlatıldığı ortam kuyruk yöneticisinin ortamından devralınır. Tanımlamak istediğiniz değışkenleri `service.env` ortamı geçersiz kılma dosyalarından birine ekleyerek hizmet sürecinin ortamında ayarlanacak ek ortam değışkenleri tanımlanabilir.

İlgili başvurular

[Çoklu Platformda STOP SERVICE \(hizmeti durdur\)](#)

[START SERVICE \(hizmeti başlat\)-Çoklu Platformlar üzerinde](#)

service.env dosyasında ek ortam değişkenlerinin tanımlanması

Bir hizmet başlatıldığında, hizmet işleminin başlatıldığı ortam kuyruk yöneticisinin ortamından devralınır. Tanımlamak istediğiniz değişkenleri service.env ortamı geçersiz kılma dosyalarından birine ekleyerek hizmet sürecinin ortamında ayarlanacak ek ortam değişkenleri tanımlanabilir.

Bu görev hakkında

Ortam değişkenlerini ekleyebileceğiniz iki olası dosya vardır:

- Makine kapsamı service.env dosyası
- Kuyruk yöneticisi kapsamı service.env dosyası

Her iki dosya da (varsa) işlenir ve kuyruk yöneticisi kapsam dosyasındaki tanımlamalar, makine kapsamı dosyasındaki tanımlamalardan önceliklidir.

service.env dosyasında herhangi bir ortam değişkeni belirtebilirsiniz. Örneğin, IBM MQ hizmeti bir dizi komut çalıştırır, service.env dosyasında **PATH** kullanıcı değişkenini ayarlamak yararlı olabilir.

Not: Değişkeni ayarladığınız değerler ortam değişkenleri olamaz; örneğin, CLASSPATH= %CLASSPATH% yanlış. Benzer şekilde, Linux PATH= \$PATH :/opt/mqm/bin üzerinde beklenmeyen sonuçlar verir.


CLASSPATH büyük harfle yazılmalı ve sınıf yolu deyimi yalnızca hazır bilgi içerebilir. Bazı hizmetler (örneğin Telemetry) kendi sınıf yollarını ayarlar. service.env içinde tanımlanan **CLASSPATH**, buna eklenir.


service.env dosyasında tanımlanan değişkenlerin biçimi, ad ve değer değişkeni çiftlerinin bir listesidir. Her değişken yeni bir satırda tanımlanmalı ve her değişken, beyaz alan da içinde olmak üzere, belirtik olarak tanımlandığı şekilde alınmalıdır.

Yordam

- service.env makine kapsamı dosyasına ortam değişkenleri ekleyin.


Bu dosyanın yeri:


–  AIX and Linux sistemlerinde /var/mqm.

–  Windows sistemlerinde kuruluş sırasında seçilen veri dizini.

- service.env adlı kuyruk yöneticisi kapsamı dosyasına ortam değişkenleri ekleyin.

Bu dosya, kuyruk yöneticisi veri dizininde bulunur. Örneğin, QMNAME adlı bir kuyruk yöneticisi için ortam geçersiz kılma dosyasının yeri:

–  AIX and Linux sistemlerinde, /var/mqm/qmgrs/QMNAME/
service.env

–  Windows sistemlerinde, C:\ProgramData\IBM\MQ\qmgrs\QMNAME\service.env

service.env dosyası örneği

```
#####  
##*                                     *##  
##* <N_OCO_COPYRIGHT>                 *##  
##* Licensed Materials - Property of IBM *##  
##*                                     *##  
##* 63H9336                             *##  
##* (C) Copyright IBM Corporation 2005, 2024. *##  
##*                                     *##  
##* <NOC_COPYRIGHT>                   *##  
##*                                     *##  
#####  
##* Module Name: service.env           *##  
##* Type       : IBM MQ service environment file *##  
##* Function    : Define additional environment variables to be set *##
```

```
##          for SERVICE programs.          ##
## Usage    : <VARIABLE>=<VALUE>          ##
##          ##                               ##
##*****##
MYLOC=/opt/myloc/bin
MYTMP=/tmp
TRACEDIR=/tmp/trace
MYINITQ=ACCOUNTS.INITIATION.QUEUE
```

İlgili görevler

[“Hizmet tanımlarında değiştirilebilir eklemeleri kullanma” sayfa 203](#)

Bir hizmet nesnesinin tanımlamasındaki belirteçleri yerine koyabilirsiniz. Yerine konan simgeler, hizmet programı yürütüldüğünde otomatik olarak genişletilmiş metinleriyle değiştirilir.

İlgili başvurular

[Ortam değişkenleri açıklamaları](#)

Hizmet tanımlarında değiştirilebilir eklemeleri kullanma

Bir hizmet nesnesinin tanımlamasındaki belirteçleri yerine koyabilirsiniz. Yerine konan simgeler, hizmet programı yürütüldüğünde otomatik olarak genişletilmiş metinleriyle değiştirilir.

Bu görev hakkında

Yerine koyma değeri belirteçleri, aşağıdaki ortak belirteçler listesinden ya da `service.env` dosyasında tanımlanan değişkenlerden alınabilir.

Yordam

- Değiştirilebilir eklemeleri kullanmak için, simgeyi + karakterleri içinde **STARTCMD**, **STARTARG**, **STOPCMD**, **STOPARG**, **STDOUT** ya da **STDERR** dizgilerinin herhangi birine ekleyin.

Bunun örnekleri için bkz. [“Sunucu hizmeti nesnesi kullanılması” sayfa 204](#) ve [“Komut hizmeti nesnesi kullanılması” sayfa 206](#).



Aşağıdakiler, bir hizmet nesnesi tanımlamasında belirteçleri değiştirmek için kullanılacak ortak belirteçlerdir:

MQ_KURULUŞ_YOLU

IBM MQ 'in kurulu olduğu konum.

MQ_DATA_PATH

IBM MQ veri dizininin konumu:

-  AIX and Linux sistemlerinde IBM MQ veri dizini konumu şudur: `/var/mqm/`
-  Windows sistemlerinde, IBM MQ veri dizininin konumu, IBM MQ kuruluşu sırasında seçilen veri dizinidir.

QMNAME

Yürürlükteki kuyruk yöneticisi adı.

MQ_HIZMET_ADI

Hizmetin adı

MQ_SUNUCU_PID

Bu simge yalnızca **STOPARG** ve **STOPCMD** bağımsız değişkenleri tarafından kullanılabilir.

Sunucu hizmeti nesnelere için bu belirteç, **STARTCMD** ve **STARTARG** bağımsız değişkenleriyle başlatılan işlemin işlem tanıtıcısıyla değiştirilir. Ters durumda, bu simge 0 ile değiştirilir.

MQ_Q_MGR_DATA_PATH

Kuyruk yöneticisi veri dizininin konumu.

MQ_Q_MGR_VERI_ADI

Kuyruk yöneticisinin dönüştürülen adı. Ad dönüşümüne ilişkin ek bilgi için [IBM MQ dosya adlarını anlamabaşlıklı konuya](#) bakın.

Sunucu hizmeti nesnesi kullanılması

Bu örnekler, tetikleyici ya da başka bir program başlatmak için bir sunucu hizmet nesnesinin nasıl tanımlanacağını, kullanılacağını ve değiştirileceğini gösterir.

Başlamadan önce

Not: Bu örnekler, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu örnekler, aksi belirtilmedikçe, UNIX biçem yolu ayırıcı karakterleriyle yazılır.

Yordam

1. DEFINE SERVICE MQSC komutunu kullanarak bir sunucu hizmeti nesnesi tanımlayın:

```
DEFINE SERVICE(S1) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+bin/runmqtrm') +
STARTARG('-m +QMNAME+ -q ACCOUNTS.INITIATION.QUEUE') +
STOPCMD('+MQ_INSTALL_PATH+bin/amqsstop') +
STOPARG('-m +QMNAME+ -p +MQ_SERVER_PID+')
```

Burada:

+MQ_INSTALL_PATH+ , kuruluş dizinini gösteren bir simgedir.

+QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.

ACCOUNTS.INITIATION.QUEUE , başlatma kuyruğudur.

amqsstop , IBM MQ ile birlikte sağlanan ve kuyruk yöneticisinden işlem tanıtıcısı için tüm bağlantıları kesmesini isteyen örnek bir programdır. amqsstop PCF komutları oluşturur; bu nedenle, komut sunucusunun çalışıyor olması gerekir.

+MQ_SERVER_PID+ , durdurma programına geçirilen işlem tanıtıcısını gösteren bir simgedir.

Ortak simgelerin listesi için bkz. [“Hizmet tanımlarında değiştirilebilir eklemeleri kullanma”](#) sayfa 203 .

2. Kuyruk yöneticisi bir sonraki başlatıldığında, sunucu hizmeti nesnesinin bir eşgörünümü yürütülür. Ancak, START SERVICE MQSC komutuyla sunucu hizmeti nesnesinin bir eşgörünümünü hemen başlatabilirsiniz:

```
START SERVICE(S1)
```

3. DISPLAY SVSTATUS MQSC komutunu kullanarak sunucu hizmeti işleminin durumunu görüntüleyin:

```
DISPLAY SVSTATUS(S1)
```

4. Sunucu hizmeti nesnesini değiştirin ve ALTER SERVICE MQSC komutunu kullanarak sunucu hizmeti işlemini el ile yeniden başlatarak güncellemeleri almanızı sağlar.

Sunucu hizmet nesnesi, başlatma kuyruğunun JUPITER.INITIATION.QUEUE olarak belirtilmesi için değiştirilir.

```
ALTER SERVICE(S1) +
STARTARG('-m +QMNAME+ -q JUPITER.INITIATION.QUEUE')
```

Not: Çalışmakta olan bir hizmet, yeniden başlatılıncaya kadar hizmet tanımında herhangi bir güncelleme almayacaktır.

5. **STOP SERVICE ve START SERVICE MQSC komutlarını kullanarak, sunucu hizmeti işlemini yeniden başlatın.**

```
STOP SERVICE(S1)
```

İzlenen:

```
START SERVICE(S1)
```

Sunucu hizmeti işlemi yeniden başlatılır ve “4” sayfa 204’ünde yapılan değişiklikleri alır.

Not: MQSC komutu **STOP SERVICE**, hizmet tanımlamasında bir **STOPCMD** bağımsız değişkeni belirtildiyse kullanılabilir.

Geçirilen bağımsız değişkenlere ilişkin diğer örnekler

- **Kuyruk yöneticisi başlatıldığında runserv adlı bir programı başlatmak için bir sunucu hizmeti nesnesi tanımlayın.**

Bunu **DEFINE SERVICE** MQSC komutunu kullanarak yapın.

Bu örnek, Windows biçem yolu ayırıcı karakterleriyle yazılır.

Başlangıç programına geçirilen bağımsız değişkenlerden biri, boşluk içeren bir dizedir. Bu bağımsız değişkenin tek bir dizgi olarak iletilmesi gerekir. Bunu gerçekleştirmek için, komut hizmeti nesnesini tanımlamak üzere aşağıdaki komutta gösterildiği gibi çift tırnak işaretleri kullanılır.

```
DEFINE SERVICE(S1) SERVTYPE(SERVER) CONTROL(QMGR) +
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\'') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')
```

```
DEFINE SERVICE(S4) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\'') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')
```

Burada:

+QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.

"C:\Program Files\Tools\'", tek bir dizgi olarak geçirilecek bir boşluk içeren bir dizedir.

- **Kuyruk yöneticisi başlatıldığında Tetikleme İzleyicisini otomatik olarak başlatmak için kullanılacak bir sunucu hizmeti nesnesi tanımlayın.**

Bunu **DEFINE SERVICE** MQSC komutunu kullanarak yapın.

```
DEFINE SERVICE(TRIG_MON_START) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('runmqtrm') +
STARTARG('-m +QMNAME+ -q +IQNAME+')
```

Burada:

+QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.

+IQNAME+ , başlatma kuyruğunun adını gösteren service.env dosyalarından birinde kullanıcı tarafından tanımlanan bir ortam değişkenidir.

İlgili başvurular

[HİZMETİ DİZ](#)

[HİZMETİ TANIMLAYIN](#)

[SVSTATUS DURUMUNU GÖRÜNTÜLE](#)

[Hizmeti başlat](#)

[Hizmeti durdur](#)

Komut hizmeti nesnesi kullanılması

Bu örnekler, bir kuyruk yöneticisi başlatıldığında ya da durdurulduğunda işletim sisteminin sistem günlüğüne giriş yazan bir programı başlatmak için bir komut hizmeti nesnesinin nasıl tanımlanacağını gösterir.

Başlamadan önce

Not: Bu örnekler, **DEFINE SERVICE** MQSC komutunu çalıştırmanızı gerektirir. Bunu nasıl yapacağınızı platforma göre değiştir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu örnekler, UNIX biçem yolu ayırıcı karakterleriyle yazılır.

Bu görev hakkında

Aşağıdaki örneklerde:

logger , IBM MQ ile birlikte sağlanan ve işletim sisteminin sistem günlüğüne giriş yazabilen örnek bir programdır.

+QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.

Yordam

- Bir kuyruk yöneticisi başlatıldığında ya da durdurulduğunda işletim sisteminin sistem günlüğüne giriş yazan bir programı başlatmak için bir komut hizmeti nesnesi tanımlayın:

```
DEFINE SERVICE(S2) +
CONTROL(QMGR) +
SERVTYPE(COMMAND) +
STARTCMD('/usr/bin/logger') +
STARTARG('Queue manager +QMNAME+ starting') +
STOPCMD('/usr/bin/logger') +
STOPARG('Queue manager +QMNAME+ stopping')
```

- Yalnızca bir kuyruk yöneticisi durdurulduğunda işletim sisteminin sistem günlüğüne giriş yazan bir programı başlatmak için bir komut hizmeti nesnesi tanımlayın:

```
DEFINE SERVICE(S3) +
CONTROL(QMGR) +
SERVTYPE(COMMAND) +
STOPCMD('/usr/bin/logger') +
STOPARG('Queue manager +QMNAME+ stopping')
```

İlgili başvurular

[HİZMETİ TANIMLAYIN](#)

Tetikleme için nesnelere yönetme

Bu örnekler, bir kuyruktaki belirli koşullar karşılandığında uygulamanın otomatik olarak nasıl başlatılacağını gösterir. Örneğin, kuyruktaki ileti sayısı belirtilen sayıya ulaştığında bir uygulama başlatmak isteyebilirsiniz. Bu tesise *triggering*(tetikleme) adı verilir. Tetiklemeyi destekleyen nesnelere tanımlamanız gerekir.

Başlamadan önce

Not: Bu örnekler, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. [MQSC komutlarını kullanarak IBM MQ yönetme](#).

Bu örnekler, UNIX biçem yolu ayırıcı karakterleriyle yazılır.

Bu görev hakkında

Tetiklemeye ilişkin ayrıntılı açıklamalar için [Tetikleyicileri kullanarak IBM MQ uygulamalarının başlatılması](#) başlıklı konuya bakın.

Yordam

- Tetiklemek için bir uygulama kuyruğu tanımlayın.

Uygulama kuyruğu, MQI aracılığıyla ileti alışverişi için uygulamalar tarafından kullanılan yerel bir kuyruktur. Tetikleme, uygulama kuyruğunda bir dizi kuyruk özniteliğinin tanımlanmasını gerektirir.

Kendisini tetikleyen **Trigger** özniteliği tarafından etkinleştirilir (MQSC komutlarında TRIGGER). Bu örnekte, MOTOR.INSURANCE.QUEUE, aşağıdaki gibi:

```
DEFINE QLOCAL (MOTOR.INSURANCE.QUEUE) +  
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +  
MAXMSGL (2000) +  
DEFPSIST (YES) +  
INITQ (MOTOR.INS.INIT.QUEUE) +  
TRIGGER +  
TRIGTYPE (DEPTH) +  
TRIGDPTH (100)+  
TRIGMPRI (5)
```

Burada:

QLOCAL (MOTOR.INSURANCE.QUEUE)

Tanımlanmakta olan uygulama kuyruğunun adı.

PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

Tetikleyici izleme programı tarafından başlatılacak uygulamayı tanımlayan süreç tanımlamasının adıdır.

MAXMSGL (2000)

Kuyruktaki iletilerin uzunluk üst sınırıdır.

DEFPSIST (YES)

Bu kuyruktaki iletilerin varsayılan olarak kalıcı olduğunu belirtir.

INITQ (MOTOR.INS.INIT.QUEUE)

Kuyruk yöneticisinin tetikleme iletisini koyacağı başlatma kuyruğunun adıdır.

TRIGGER

Tetikleyici öznitelik değeridir.

TRIGTYPE (DEPTH)

Gerekli öncelik (TRIGMPRI) iletilerinin sayısı TRIGDPTH ' de belirtilen sayıya ulaştığında bir tetikleme olayının oluşturulduğunu belirtir.

TRIGDPTH (100)

Bir tetikleyici olayı oluşturmak için gereken ileti sayısıdır.

TRIGMPRI (5)

Bir tetikleyici olayı oluşturulup oluşturulmayacağına karar vermede kuyruk yöneticisi tarafından sayılacak iletilerin önceliğidir. Yalnızca 5 ya da daha yüksek önceliğe sahip iletiler sayılır.

- Başlatma kuyruğu tanımla

Bir tetikleme olayı oluşturduğunda, kuyruk yöneticisi uygulama kuyruğu tanımlamasında belirtilen başlatma kuyruğuna bir tetikleyici iletisi koyar. Başlatma kuyruklarının özel ayarları yoktur, ancak MOTOR.INS.INIT.QUEUE :

```
DEFINE QLOCAL (MOTOR.INS.INIT.QUEUE) +
GET (ENABLED) +
NOSHARE +
NOTRIGGER +
MAXMSGL (2000) +
MAXDEPTH (1000)
```

- Süreç tanımla

Bir süreç tanımlaması yaratmak için DEFINE PROCESS komutunu kullanın. Süreç tanımlaması, uygulama kuyruğundaki iletileri işlemek için kullanılacak uygulamayı tanımlar. Uygulama kuyruğu tanımlaması, kullanılacak işlemi adlandırır ve böylece uygulama kuyruğunu, iletilerini işlemek için kullanılacak uygulamayla ilişkilendirir. Bu, MOTOR.INSURANCE.QUEUE. Aşağıdaki MQSC komutu gerekli olan MOTOR.INSURANCE.QUOTE.PROCESS, bu örnekte tanıtlır:

```
DEFINE PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
DESCR ('Insurance request message processing') +
APPLTYPE (UNIX) +
APPLICID ('/u/admin/test/IRMP01') +
USERDATA ('open, close, 235')
```

Burada:

MOTOR.INSURANCE.QUOTE.PROCESS

Süreç tanımlamasının adıdır.

DESCR ('Insurance request message processing')

Bu tanımın ilişkili olduğu uygulama programını açıklar. Bu metin, DISPLAY PROCESS komutunu kullandığınızda görüntülenir. Bu, sürecin ne yaptığını tanımlamanıza yardımcı olabilir. Dizgide boşluk kullanırsanız, dizgiyi tek tırnak içine almanız gerekir.

APPLTYPE (UNIX)

Başlatılacak uygulamanın tipi.

APPLICID ('/u/admin/test/IRMP01')

Tam olarak nitelenmiş bir dosya adı olarak belirtilen, uygulama yürütülebilir dosyasının adıdır. Windows sistemlerinde tipik bir APPLICID değeri c:\app1\test\irmp01.exe olur.

USERDATA ('open, close, 235')

Uygulama tarafından kullanılabilen kullanıcı tanımlı verilerdir.

- Süreç tanımlamasının özneliklerinin görüntülenmesi

Tanımınızın sonuçlarını incelemek için DISPLAY PROCESS komutunu kullanın. Örneğin:

```
DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

24 : DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL
AMQ8407: Display Process details.
DESCR ('Insurance request message processing')
APPLICID ('/u/admin/test/IRMP01')
USERDATA (open, close, 235)
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)
APPLTYPE (UNIX)
```

Var olan bir süreç tanımlamasını değiştirmek için MQSC komutu ALTER PROCESS ve bir süreç tanımlamasını silmek için DELETE PROCESS komutunu da kullanabilirsiniz.

İki sistem arasında dmpmqmsg yardımcı programının kullanılması

dmpmqmsg yardımcı programı (önceki *qload*), bir kuyruğun ya da iletilerinin içeriğini bir dosyaya kopyalamanızı ya da taşımanızı sağlar.

Genel Bakış

dmpmqmsg ile oluşturduğunuz dosya gerektiği şekilde saklanabilir ve daha sonra iletileri kuyruğa yeniden yüklemek için kullanılabilir.

Önemli:

1. Dosya, yardımcı program tarafından anlaşılan belirli bir biçime sahiptir. Ancak, dosyayı yeniden yüklemeyen önce bir düzenleyicide güncelleyebilmek için dosya insan tarafından okunabilir. Dosyayı düzenlerseniz, biçimini değiştirmemelisiniz.
2. IBM MQ 9.1 olanağından, **dmpmqmsg** yardımcı programı AIX, Linux, and Windows için yürütme ortamı kütük kümesiyle birlikte verilir; bu nedenle hem IBM MQ sunucusunda hem de istemcide kullanılabilir. IBM MQ 9.1 ' den önce yardımcı program yalnızca sunucu paketiyle birlikte gönderiliyordu.

Olası kullanımlar şunlardır:

- Kuyruktaki iletilerin bir dosyaya saklanması. Büyük olasılıkla arşivleme amacıyla ve daha sonra bir kuyruğa yeniden yükleyin.
- Daha önce bir dosyaya kaydettiğiniz iletilerle bir kuyruk yeniden yükleniyor.
- Eski iletilerin kuyruktan kaldırılması.
- Gerekirse iletiler arasında doğru zamanı koruyarak, depolanan bir konumdan test iletileri 'yeniden yürütüyor'.



Uyarı: SupportPac MO03 , yerel ya da istemci bağ tanımını belirtmek için **-1** değiştirgesini kullandı. **-1** , **-c** parametresiyle değiştirildi.

-P artık **-c** yerine kod sayfası bilgileri için kullanılır.

Komutla ve kullanılacak değiştirgelerle ilgili ek bilgi için [dmpmqmsg](#) kısmına bakın.

Windows makinesi kullanarak Linux üzerinde dmpmqmsg yardımcı programını kullanma örneği

Bir Linux makinesinde, aynı kuyruk yöneticisinde başka bir kuyruğa (Q2) taşımak istediğiniz iletileri (Q1) içeren bir kuyruk yöneticiniz vardır. **dmpmqmsg** yardımcı programını bir Windows makinesinden başlatmak istiyorsunuz.

Kuyruk (Q1), örnek **amqspu1** (yerel kuyruk yöneticisi) ya da **amqspu1c** (uzak kuyruk yöneticisi) uygulaması kullanılarak eklenen dört iletiye sahiptir.

Linux makinesinde şunları görürsünüz:

```
display q1(Q1) CURDEPTH
2 : display q1(Q1) CURDEPTH
AMQ8409: Display Queue details.
  QUEUE(Q1)
  TYPE(QLOCAL)
  CURDEPTH(4)
```

MQSERVER ortam değişkenini Linux içindeki kuyruk yöneticisini gösterecek şekilde ayarlayın. Örneğin:

```
set MQSERVER=SYSTEM.DEF.SVRCONN/TCP/veracruz.x.com(1414)
```

Burada *veracruz* , makinenin adıdır.

Kuyruktan (Q1) okumak için **dmpmqmsg** yardımcı programını çalıştırın ve çıkışı `c:\temp\mqqload.txt` içinde saklayın.

MQSERVER tarafından oluşturulan Linux anasisteminde ve kapısında çalışan *QM_VER* kuyruk yöneticisine uzak istemci olarak bağlanın. Bağlantıyı uzak istemci olarak şu özniteliği kullanarak elde edebilirsiniz: **-c**.

```
dmpmqmsg -m QM_VER -i Q1 -f c:\temp\mqqload.txt -c
```

```
Read      - Files:    0  Messages:    4  Bytes:    22
Written - Files:    1  Messages:    4  Bytes:    22
```

c:\temp\mqpload.txt çıkış kütüğü, **dmpmqmsg** yardımcı programının anladığı bir biçimi kullanarak metin içerir.

Windows makinesinde, Linux makinesindeki yükleme kuyruğunu (Q2) Windows makinesindeki bir dosyadan yüklemek için **dmpmqmsg** komutunu (-i seçeneği yerine -o seçeneğini kullanarak) verin:

```
dmpmqmsg -m QM_VER -o Q2 -f c:\temp\mqpload.txt -c
Read      - Files:    1  Messages:    4  Bytes:    22
Written - Files:    0  Messages:    4  Bytes:    22
```

Linux makinesinde, kuyrukta artık dosyadan geri yüklenen dört ileti olduğunu unutmayın.

```
display ql(Q2) CURDEPTH
 6 : display ql(Q2) CURDEPTH
AMQ8409: Display Queue details.
  QUEUE(Q2)
  TYPE(QLOCAL)
  CURDEPTH(4)
```

Linux makinesinde,

İletileri özgün kuyruktan silin.

```
clear qllocal(Q1)
 4 : clear qllocal(Q1)
AMQ8022: IBM MQ queue cleared.
```

Özgün kuyrukta başka ileti olmadığını doğrulayın:

```
display ql(Q1) CURDEPTH
 5 : display ql(Q1) CURDEPTH
AMQ8409: Display Queue details.
  QUEUE(Q1)
  TYPE(QLOCAL)
  CURDEPTH(0)
```

Komutun ve deęiřtirgelerinin açıklaması için bkz. [dmpmqmsg](#) .

İlgili kavramlar

[“dmpmqmsg yardımcı programını kullanma örnekleri” sayfa 210](#)

dmpmqmsg yardımcı programını (önceki adıyla **qload**) kullanabileceğiniz basit yöntemler. Bu yardımcı program IBM MQ 8.0' daki ürünle birleştirilmiştir.

dmpmqmsg yardımcı programını kullanma örnekleri

dmpmqmsg yardımcı programını (önceki adıyla **qload**) kullanabileceğiniz basit yöntemler. Bu yardımcı program IBM MQ 8.0' daki ürünle birleştirilmiştir.

Daha önce **qload** yardımcı programı SupportPac MO03olarak mevcuttu.

Kuyruęu bir dosyaya bořa al

Kuyruktaki iletileri bir dosyaya kaydetmek için komut satırında ařaęıdaki seçenekleri kullanın:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile
```

Bu komut, kuyruktaki iletilerin bir kopyasını alır ve bunları belirlenen dosyada saklar.

Bir kuyruğun bir dosya dizisine boşaltılmasının

Dosya adında bir insert karakteri kullanarak bir kuyruğu bir dosya dizisine boşaltabilirsiniz. Bu kipte her ileti yeni bir dosyaya yazılır:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile%n
```

Bu komut, kuyruğu myfile1, myfile2, myfile3vb. dosyalarla boşaltır.

Dosyadan kuyruk yükle

“Kuyruğu bir dosyaya boşalt” sayfa 210'e kaydettiğiniz iletilerle bir kuyruğu yeniden yüklemek için komut satırında aşağıdaki seçenekleri kullanın:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

Bu komut, kuyruğu myfile1, myfile2, myfile3vb. dosyalarla boşaltır.

Bir dosya dizisinden kuyruk yükleme

Dosya adında bir insert karakteri kullanarak bir dizi dosyadan bir kuyruk yükleyebilirsiniz. Bu kipte her ileti yeni bir dosyaya yazılır:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

Bu komut, kuyruğu dosyalara (myfile1, myfile2, myfile3vb.) yükler.

İletileri bir kuyruktan başka bir kuyruğa kopyala

“Kuyruğu bir dosyaya boşalt” sayfa 210'indeki dosya parametresini başka bir kuyruk adıyla değiştirin ve aşağıdaki seçenekleri kullanın:

```
dmpmqmsg -m QM1 -i Q1 -o Q2
```

Bu komut, bir kuyruktaki iletilerin başka bir kuyruğa kopyalanmasını sağlar.

İlk 100 iletiyi bir kuyruktan başka bir kuyruğa kopyala

Önceki örnekteki komutu kullanın ve -r#100 seçeneğini ekleyin:

```
dmpmqmsg -m QM1 -i Q1 -o Q2 -r#100
```

İletilerin bir kuyruktan başka bir kuyruğa taşınması

“Dosyadan kuyruk yükle” sayfa 211'ünde bir varyasyon. Yalnızca bir kuyruğa göz atan -i (küçük harf) ile bir kuyruktan yıkıcı bir şekilde alınan -I (büyük harf) arasındaki ayrımı göz önünde bulundurun:

```
dmpmqmsg -m QM1 -I Q1 -o Q2
```

Bir günden eski iletileri bir kuyruktan diğerine taşıma

Bu örnek, yaş seçiminin kullanımını gösterir. Daha eski, daha genç ya da bir yaş aralığı içinde olan iletiler seçilebilir.

```
dmpmqmsg -m QM1 -I Q1 -o Q2 -T1440
```

Su anda kuyrukta olan iletilerin yaşlarının görüntülenmesi

Komut satırında aşağıdaki seçenekleri kullanın:

```
dmpmqmsg -m QM1 -i Q1 -f stdout -dT
```

İleti dosyasıyla çalış

“Kuyruğu bir dosyaya boşa al” sayfa 210 gibi kuyruğunuzda iletiyi boşa aldıktan sonra dosyayı düzenlemek isteyebilirsiniz.

Ayrıca, dosyanın biçimini, kuyruğu boşa aldığınız sırada belirtmediğiniz görüntüleme seçeneklerinden birini kullanacak şekilde değiştirmek isteyebilirsiniz.

dmpmqmsg yardımcı programını kullanarak, kuyruğun boşa alınması gerçekleştiikten sonra da dosyayı gereken biçimde yeniden işleyebilirsiniz. Komut satırında aşağıdaki seçenekleri kullanın.

```
dmpmqmsg -f c:\oldfile -f c:\newfile -dA
```

Komutun ve değiştiregelerin açıklaması için bkz. [dmpmqmsg](#).

Uzak IBM MQ nesnelereyle çalışma

Uzak kuyruk yöneticilerindeki IBM MQ nesnelere MQSC komutlarını, PCF komutlarını ya da administrative REST API komutunu kullanarak denetleyebilirsiniz. Bu yöntemlerden herhangi birini kullanmadan önce, uzak kuyruk yöneticisine komutların ve yerel kuyruk yöneticisi tarafından alınan yanıtların gönderilebilmesi için, yerel kuyruk yöneticisi ile uzak kuyruk yöneticisi arasında iletim kuyrukları ve kanalları tanımlamanız gerekir. Diğer bir seçenek olarak, bir kuyruk yöneticisi kümesi yapılandırabilir ve daha sonra, aynı uzak denetim yöntemlerini kullanabilirsiniz.

Bu görev hakkında

Kuyruk yöneticilerini uzak denetime hazırlamak için, yerel kuyruk yöneticisinde aşağıdaki nesnelere yapılandırmanız gerekir:

- Bir dinleyici.
- Uzak kuyruk yöneticisinin adını taşıyan bir iletim kuyruğu.
- Uzak kuyruk yöneticisine ilişkin bağlantı ayrıntılarına sahip bir gönderen kanal.
- Uzak kuyruk yöneticisindeki gönderen kanalla aynı ada sahip bir alıcı kanalı.

Uzak kuyruk yöneticisinde aşağıdaki nesnelere de yapılandırmanız gerekir:

- Bir dinleyici.
- Yerel kuyruk yöneticisinin adını taşıyan bir iletim kuyruğu.
- Yerel kuyruk yöneticisine ilişkin bağlantı ayrıntılarına sahip bir gönderen kanal.
- Yerel kuyruk yöneticisindeki gönderen kanalla aynı ada sahip bir alıcı kanalı.

Bu nesnelere yapılandırılmasına ilişkin ek bilgi için bkz. “Uzak denetim için kuyruk yöneticilerinin yapılandırılması” sayfa 213.

Diğer bir seçenek olarak, bir kuyruk yöneticisi kümesi yapılandırabilirsiniz. *Küme*, kuyruk yöneticilerinin karmaşık iletim kuyruğu, kanal ve kuyruk tanımlamaları olmadan tek bir ağ üzerinden doğrudan birbirleriyle iletişim kurabilecekleri şekilde ayarlanmış bir kuyruk yöneticisi grubudur. Kümeler kolaylıkla ayarlanabilir ve genellikle bir şekilde mantıksal olarak ilişkili olan ve veri veya uygulamaları paylaşması gereken kuyruk yöneticilerini içerir. En küçük küme bile sistem yönetimi maliyetlerini düşürür.

Bir kümede kuyruk yöneticilerinden oluşan bir ağ oluşturulması, geleneksel bir dağıtılmış kuyruğa alma ortamının oluşturulmasından daha az tanımlama içerir. Daha az tanımla, ağınıza daha hızlı ve kolay bir şekilde ayarlayabilir ya da değiştirebilir ve tanımlarınızda hata yapma riskini azaltabilirsiniz.

Bir küme ayarlamak için, her kuyruk yöneticisi için bir küme gönderen (CLUSDR) ve bir küme alıcısı (CLUSRCVR) tanımlaması gerekir. İletim kuyruğu tanımlarına ya da uzak kuyruk tanımlarına gerek yoktur. Uzak yönetim ilkeleri, bir küme içinde kullanıldığında aynıdır, ancak tanımların kendileri büyük ölçüde basitleştirilmiştir.

Küme yapılandırmaya ilişkin ek bilgi için [Kuyruk yöneticisi kümesinin yapılandırılması](#) başlıklı konuya bakın.

Yordam

- Uzak IBM MQ nesnelerini nasıl yöneteceğiyle ilgili bilgi için aşağıdaki alt konulara bakın:
 - [“Uzak denetim için kuyruk yöneticilerinin yapılandırılması” sayfa 213](#)
 - [“Uzak yönetim için komut sunucusunun yönetilmesi” sayfa 217](#)
 - [“Uzak bir kuyruk yöneticisinde MQSC komutları verilmesi” sayfa 218](#)
 - [“Kodlanmış karakter takımları arasında veri dönüştürme” sayfa 220](#)

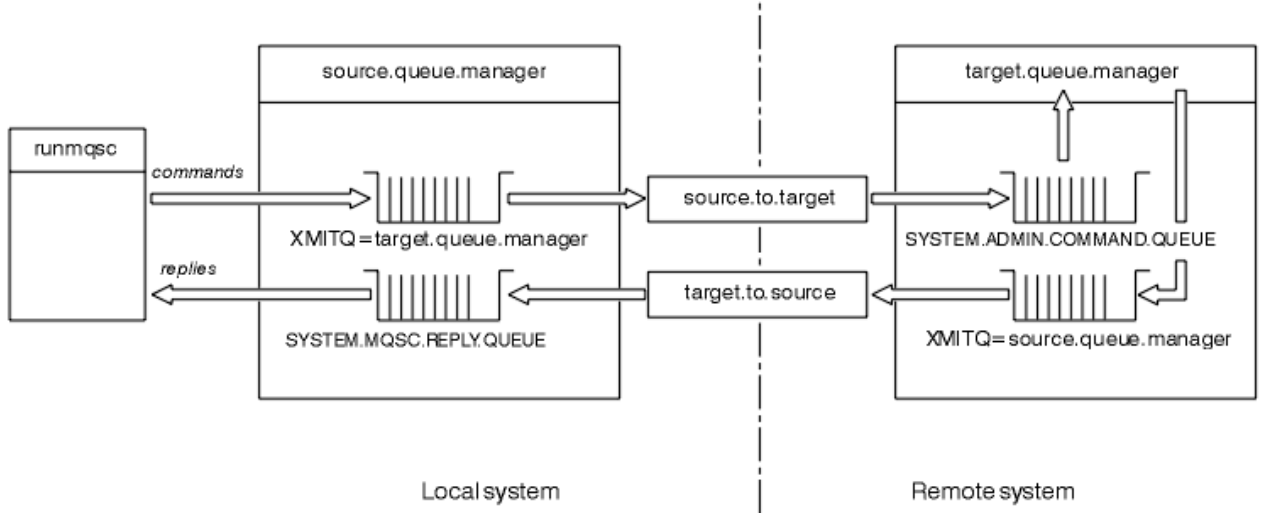
Uzak denetim için kuyruk yöneticilerinin yapılandırılması

administrative REST API, MQSC ya da PCF komutlarını kullanarak uzak kuyruk yöneticisini yerel bir kuyruk yöneticisinden denetleyebilirsiniz. Uzak kuyruk yöneticisi aynı sistemde, farklı bir kuruluştaki ya da aynı ortama sahip farklı bir sistemde ya da farklı bir IBM MQ ortamında olabilir. Bir kuyruk yöneticisini yerel bir kuyruk yöneticisinden uzaktan denetlemeden önce, her kuyruk yöneticisinde bir gönderen ve alıcı kanalı, bir dinleyici ve bir iletim kuyruğu yaratmanız gerekir. Bu kanallar ve kuyruklar, komutların uzak kuyruk yöneticisine ve yerel kuyruk yöneticisine gönderilmesini sağlar. Bu kuyrukları ve kanalları yaratma yordamı, administrative REST API, MQSC ya da PCF komutlarını kullanmak isteyip istemediğiniz ile aynıdır.

Başlamadan önce

- Aşağıdaki yordamda örnek kuyruk yöneticileri `source.queue.manager` ve `target.queue.manager` kullanılmıştır. Bu adımları izlemek için sisteminizde bu kuyruk yöneticilerini oluşturup başlatmanız ya da ilgili adımlarda kendi kuyruk yöneticisi adlarınızı değiştirmeniz gerekir.
- Aşağıdaki yordam iletim tipi olarak TCP/IP 'yi kullanır. Bu görevi tamamlamak için her iki sistemin IP adresini bilmeniz gerekir.
- Aşağıdaki yordam, yerel sistemde 1818 ve uzak sistemde 1819 ağ kapılarını kullanan dinleyicileri yaratır. Diğer kapıları kullanabilirsiniz, ancak uygun adımlarda kapı değerlerinizi değiştirmeniz gerekir.
- Yordamdaki komutları yerel olarak ya da Telnet gibi bir ağ olanağı üzerinden çalıştırmanız gerekir.

Bu görev hakkında



Şekil 15. Uzaktan yönetim için kanalların ve kuyrukların ayarlanması

Şekil 15 sayfa 214 , uzak yönetim için gereksinim duyduğunuz kuyruk yöneticilerinin, kuyrukların ve kanalların yapılandırmasını gösterir:

- `source.queue.manager` nesnesi, administrative REST API, MQSC ya da PCF komutlarını yayınlatabileceğiniz ve bu komutların sonuçlarının döndürüleceği kaynak kuyruk yöneticisidir.
- `target.queue.manager` nesnesi, komutları işleyen ve işleç iletileri oluşturan hedef kuyruk yöneticisinin adıdır.
- Komutlar, uzak kuyruk yöneticisiyle aynı adı taşıyan iletim kuyruğuna yerleştirilir. Bu durumda, `target.queue.manager`. İletim kuyruğu, MCA iletileri alıp uzak kuyruk yöneticisine göndermeden önce iletileri geçici olarak tutan özel bir yerel kuyruktur.
- Komutlar, `source.to.target` kanalı tarafından uzak kuyruk yöneticisindeki `SYSTEM.ADMIN.COMMAND.QUEUE` ' e gönderilir. Kanalın her ucunun ayrı bir tanımı vardır. Bir ucu gönderen, diğer ucu alıcı. İki tanımlama aynı ada sahip olmalı ve birlikte tek bir iletili kanalı oluşturmalıdır.
- Komut çıkışı, komutun gönderildiği yerel kuyruk yöneticisiyle aynı adı taşıyan uzak iletili kuyruğuna yerleştirilir. Bu durumda, `source.queue.manager`.
- Çıkış, `target.to.source` kanalı tarafından, özgün komut tarafından alındığı ve çıktısının alındığı uygun bir yanıt kuyruğuna gönderilir.

Yordam

1. Uzak sistem kuyruk yöneticisinde, `SYSTEM.ADMIN.COMMAND.QUEUE` komut kuyruğunun var olduğundan emin olun. Bu kuyruk, varsayılan olarak bir kuyruk yöneticisi yaratıldığında yaratılır.
2. Uzak sistemde, komut sunucusunun kuyruk yöneticisinde çalışıp çalışmadığını denetleyin. Komut sunucusu çalışmıyorsa, uzak denetim yapılamaz:
 - a) Kuyruk yöneticisi için `runmqsc` ' i başlatın. Örneğin, `target.queue.manager` kuyruk yöneticisi için şu komutu girin:

```
runmqsc target.queue.manager
```

- b) Aşağıdaki komutu girerek komut sunucusunun durumunu görüntüleyin:

```
DISPLAY QMSTATUS CMDSERV
```

- c) Aşağıdaki komutu girerek `runmqsc` komut isteminden çıkın:

```
end
```

- d) Komut sunucusu başlatılmamışsa, başlatın. Örneğin, `target.queue.managerkuyruk` yöneticisi için şu komutu girin:

```
strmqcsv target.queue.manager
```

3. Yerel kuyruk yöneticisinde kanalları, dinleyiciyi ve iletim kuyruğunu tanımlayın:

- a) Kuyruk yöneticisi için **runmqsc** ' i başlatın. Örneğin, `source.queue.managerkuyruk` yöneticisi için şu komutu girin:

```
runmqsc source.queue.manager
```

- b) Gönderen kanalını tanımlayın. Bu gönderen kanal, uzak kuyruk yöneticisindeki alıcı kanalla aynı ada sahip olmalıdır. Örneğin, **CONNNAME** değerini uzak kuyruk yöneticisine ilişkin IP adresiyle ve dinleyicinin kapı numarasıyla değiştirerek aşağıdaki MQSC komutunu girin:

```
DEFINE CHANNEL ('source.to.target') +  
CHLTYPE(SDR) +  
CONNNAME (localhost:1819) +  
XMITQ ('target.queue.manager') +  
TRPTYPE(TCP)
```

- c) Alıcı kanalını tanımlayın. Bu alıcı kanal, uzak kuyruk yöneticisindeki gönderen kanalla aynı ada sahip olmalıdır. Örneğin, şu komutu girin.

```
DEFINE CHANNEL ('target.to.source') +  
CHLTYPE(RCVR) +  
TRPTYPE(TCP)
```

- d) Yerel kuyruk yöneticisinde dinleyiciyi tanımlayın. Örneğin, şu komutu girin.

```
DEFINE LISTENER ('source.queue.manager') +  
TRPTYPE (TCP) +  
PORT (1818)
```

- e) Yerel kuyruk yöneticisinde iletim kuyruğunu tanımlayın. Bu iletim kuyruğu, uzak kuyruk yöneticisiyle aynı ada sahip olmalıdır. Örneğin, şu komutu girin.

```
DEFINE QLOCAL ('target.queue.manager') +  
USAGE (XMITQ)
```

- f) Dinleyiciyi başlatın. Örneğin, şu komutu girin.

```
START LISTENER ('source.queue.manager')
```

- g) Aşağıdaki komutu girerek **runmqsc** komut isteminden çıkın:

```
end
```

4. Uzak kuyruk yöneticisindeki kanalları, dinleyiciyi ve iletim kuyruğunu tanımlayın:

- a) Kuyruk yöneticisi için **runmqsc** ' i başlatın. Örneğin, `target.queue.managerkuyruk` yöneticisi için şu komutu girin:

```
runmqsc target.queue.manager
```

- b) Gönderen kanalını tanımlayın. Bu gönderen kanal, yerel kuyruk yöneticisindeki alıcı kanalla aynı ada sahip olmalıdır. Örneğin, **CONNNAME** değerini yerel kuyruk yöneticisine ilişkin IP adresiyle ve dinleyiciye ilişkin kapı numarasıyla değiştirerek şu MQSC komutunu girin:

```
DEFINE CHANNEL ('target.to.source') +
CHLTYPE(SDR) +
CONNNAME (localhost:1818) +
XMITQ ('source.queue.manager') +
TRPTYPE(TCP)
```

- c) Alıcı kanalını tanımlayın. Bu alıcı kanal, yerel kuyruk yöneticisindeki gönderen kanalla aynı ada sahip olmalıdır. Örneğin, aşağıdaki komutu girin:

```
DEFINE CHANNEL ('source.to.target') +
CHLTYPE(RCVR) +
TRPTYPE(TCP)
```

- d) Dinleyiciyi tanımlayın. Örneğin, şu komutu girin.

```
DEFINE LISTENER ('target.queue.manager') +
TRPTYPE (TCP) +
PORT (1819)
```

- e) İletim kuyruğunu tanımlayın. Bu iletim kuyruğu, yerel kuyruk yöneticisiyle aynı ada sahip olmalıdır. Örneğin, şu komutu girin.

```
DEFINE QLOCAL ('source.queue.manager') +
USAGE (XMITQ)
```

- f) Dinleyiciyi başlatın. Örneğin, şu komutu girin.

```
START LISTENER ('target.queue.manager')
```

- g) Şu komutu girerek **runmqsc** ' den çıkın:

```
end
```

5. Yerel sistemde gönderen kanalını başlatın:

- a) Kuyruk yöneticisi için **runmqsc** ' i başlatın. Örneğin, `source.queue.manager` kuyruk yöneticisi için şu komutu girin:

```
runmqsc source.queue.manager
```

- b) Gönderen kanalını başlatın. Örneğin, şu komutu girin.

```
START CHANNEL ('source.to.target')
```

- c) Şu komutu girerek **runmqsc** ' den çıkın:

```
end
```

6. Uzak sistemde gönderen kanalını başlatın:

- a) Kuyruk yöneticisi için `runmqsc` ' i başlatın. Örneğin, `target.queue.manager` kuyruk yöneticisi için şu komutu girin:

```
runmqsc target.queue.manager
```

- b) Gönderen kanalını başlatın. Örneğin, şu komutu girin.

```
START CHANNEL ('target.to.source')
```

- c) Şu komutu girerek **runmqsc** ' den çıkın:

```
end
```


7. Yerel sistemden uzak kuyruk yöneticisine bir MQSC komutu göndererek yapılanışın başarıyla tamamlandığını doğrulayın:

- a) Uzak kuyruk yöneticisi için yerel sistemden **runmqsc** komut istemini başlatın. Örneğin, şu komutu girin.

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

- b) Aşağıdaki komutu girerek uzak kuyruk yöneticisindeki kuyrukları görüntüleyin:

```
DISPLAY QUEUE (*)
```

Başarılı olursa, uzak kuyruk yöneticisindeki kuyrukların bir listesi görüntülenir.

- c) Bu adımlar işe yaramazsa, her iki sistemdeki kanalların çalışır durumda olup olmadığını denetleyin. Kanallar çalışmıyorsa ve başlatılmıyorsa, kanalların ve iletim kuyruklarının doğru yapılandırıldığını ve komut sunucusunun çalıştığını denetleyin. Örneğin, gönderen kanallar için doğru CONNAME belirtilip belirtilmediğini ve iletim kuyruklarının doğru adlara sahip olup olmadığını denetleyin. Ayrıca, sorunun çözülmesine yardımcı olabilecek güvenlik kural dışı durumları için kuyruk yöneticisi günlüklerine bakın.

Sonuçlar

Kuyruk yöneticileriniz, uzak kuyruk yöneticisini yerel sistemden uzaktan denetleyecek şekilde yapılandırıldı.

Sonraki adım

- MQSC komutlarını kullanarak uzaktan yönetimle ilgili daha fazla bilgi edinin: [“Uzak bir kuyruk yöneticisinde MQSC komutları verilmesi” sayfa 218](#)
- PCF komutlarını kullanarak denetim programları yazmakla ilgili daha fazla bilgi edinin: [“IBM MQ Programlanabilir Komut Biçimlerini Kullanma” sayfa 26](#).
- administrative REST API komutunu uzaktan yönetim için kullanma hakkında daha fazla bilgi edinin: [“REST API kullanılarak uzaktan yönetim” sayfa 78](#).

Uzak yönetim için komut sunucusunun yönetilmesi

Her kuyruk yöneticisiyle ilişkilendirilmiş bir komut sunucusu vardır. Komut sunucusu, uzak kuyruk yöneticilerinden gelen komutları ya da uygulamalardan gelen PCF komutlarını işler. Kuyruk yöneticisine işlenmek üzere komutlar sunar ve bir tamamlanma kodu ya da işletmen iletisi döndürür. Komut sunucusunu başlatabilir, durdurabilir ve durumunu görüntüleyebilirsiniz. Komut sunucusu, PCF komutlarını, MQAI ' yi ve uzak yönetimi içeren tüm yönetim için zorunludur.

Başlamadan önce

Kuyruk yöneticisi özniteliğinin değerine (**SCMDSERV**) bağlı olarak, kuyruk yöneticisi başlatıldığında komut sunucusu otomatik olarak başlatılır ya da el ile başlatılması gerekir. Komut sunucusu otomatik olarak başlatılırsa, komut sunucusunu başlatmak ve durdurmak için strmqcsv ya da endmqcsv komutlarını kullanamazsınız. **SCMDSERV** özniteliğinin değerini, **ALTER QMGRMQSC** komutunu kullanarak değiştirebilirsiniz. Varsayılan olarak, komut sunucusu otomatik olarak başlatılır.

Bir kuyruk yöneticisinin durdurulması, kendisiyle ilişkili komut sunucusunu da sona erdirir.

Yordam

- Komut sunucusunun durumunu görüntüle:

a) Aşağıdaki komutu girerek uygun kuyruk yöneticisi için **runmqsc** komut istemini başlatın:

```
runmqsc target.queue.manager
```

Burada `target.queue.manager` , komut sunucusunun görüntülenmekte olduğu kuyruk yöneticisidir.

b) Aşağıdaki MQSC komutunu girerek komut sunucusu durumunu görüntüleyin:

```
DISPLAY QMSTATUS CMDSERV
```

c) Aşağıdaki komutu girerek **runmqsc** komut isteminden çıkın:

```
end
```

- Komut sunucusu otomatik olarak başlatılacak şekilde ayarlanmazsa, aşağıdaki komutu girerek komut sunucusunu başlatın:

```
strmqcsv target.queue.manager
```

Burada `target.queue.manager` , komut sunucusunun başlatıldığı kuyruk yöneticisidir.

- Komut sunucusu otomatik olarak başlatılacak şekilde ayarlanmamışsa, aşağıdaki komutu girerek komut sunucusunu durdurun:

```
endmqcsv target.queue.manager
```

Burada `target.queue.manager` , komut sunucusunun durdurulmakta olduğu kuyruk yöneticisidir.

Varsayılan olarak, komut sunucusu denetimli bir şekilde durur. Komuta -i işareti ekleyerek komut sunucusunu hemen durdurabilirsiniz.

Uzak bir kuyruk yöneticisinde MQSC komutları verilmesi

Kuyruk yöneticilerini uzak denetim için yapılandırdıktan sonra, uzak bir kuyruk yöneticisinde MQSC komutlarını çalıştırmak için yerel bir sistemde belirli bir **runmqsc** komutu biçimini kullanabilirsiniz. Her komut, SYSTEM.ADMIN.COMMAND.QUEUE. Yanıtlar SYSTEM.MQSC.REPLY.QUEUE (KUYRUK).

Başlamadan önce

MQSC komutlarını kullanarak bir kuyruk yöneticisini uzaktan denetlemeden önce kanalları, iletim kuyruklarını, dinleyicileri ve komut sunucusunu yapılandırmak için [“Uzak denetim için kuyruk yöneticilerinin yapılandırılması” sayfa 213](#) içindeki adımları tamamlamanız gerekir.

Yordam

1. Komut sunucusunun uzak kuyruk yöneticisinde çalıştığından emin olun.

Komut sunucusunun bir kuyruk yöneticisinde nasıl başlatılacağına ilişkin bilgi için bkz. [“Uzak yönetim için komut sunucusunun yönetilmesi” sayfa 217](#).

2. Kaynak kuyruk yöneticisinde MQSC komutlarını şu iki yoldan biriyle çalıştırabilirsiniz:

- **runmqsc** komutunu aşağıdaki komutlarla başlatarak etkileşimli olarak:

– **z/OS** Uzak kuyruk yöneticisi z/OSüzeriyse şu komutu girin:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager
```

– **Multi** Uzak kuyruk yöneticisi z/OSişletim sistemi üzerinde değilse, şu komutu girin:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

- Bir komut dosyasından:

- a. Uzak sistemde çalıştırılacak MQSC komutlarını her satırda bir komut olacak şekilde bir metin dosyasına koyun.
- b. **runmqsc** komutunda -v işaretini kullanarak yerel kuyruk yöneticisinde MQSC komutlarınızı doğrulayın. -v işareti, komutların geçerli olup olmadığını denetler, ancak bunları çalıştırmaz. Bazı komutların uzak kuyruk yöneticisi için geçerliyse, ancak yerel kuyruk yöneticisi için geçerli değilse başarısız olabileceğini unutmayın:

```
runmqsc -v source.queue.manager < myCmdFile.in > results.out
```

myCmdFile.in , denetlenmesi gereken MQSC komutlarını içerir ve results.out dosyası komutlara ilişkin doğrulama sonuçlarını içerir.

- c. Aşağıdaki komutlardan birini girerek uzak kuyruk yöneticisinde komut dosyasını çalıştırın:

- **z/OS** Uzak kuyruk yöneticisi z/OS üzeriyse şu komutu girin:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager < myCmdFile.in >  
results.out
```

- **Multi** Uzak kuyruk yöneticisi z/OS işletim sistemi üzerinde değilse, şu komutu girin:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager < myCmdFile.in >  
results.out
```

Kullanılan değişirgeler şunlardır:

-w saniye

MQSC komutlarının, komutların komut sunucusu giriş kuyruğuna konduğu ve sırayla yürütüldüğü dolaylı kipte çalıştırılacağını belirtir.

seconds değişkeni, uzak kuyruk yöneticisinden gelen bir yanıt için beklenecek süreyi saniye cinsinden belirtir. Bu süreden sonra alınan yanıtlar atılır, ancak MQSC komutları uzak kuyruk yöneticisinde çalıştırılır. Komut zamanaşımına uğradığında yerel kuyruk yöneticisinde şu ileti oluşturulur:

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

MQSC komutlarını çalıştırmayı durdurduğunuzda, yerel kuyruk yöneticisi gelen zamanaşımına uğrayan yanıtları görüntüler ve diğer yanıtları atar.

-x

Uzak kuyruk yöneticisinin bir z/OS kuyruk yöneticisi olduğunu belirtir.

-m localQMGrAd

Uzak kuyruk yöneticisine komut sunmak için kullanmak istediğiniz yerel kuyruk yöneticisinin adını belirtir.

Sonraki adım

MQSC komutlarını uzaktan çalıştırmakta zorlanıyorsanız:

- Uzak kuyruk yöneticisinin çalışıp çalışmadığını denetleyin.
- Komut sunucusunun uzak sistemde çalışıp çalışmadığını denetleyin.
- Kanal bağlantı kesme aralığının süresinin dolmadığını denetleyin. Örneğin, bir kanal başlatıldıysa, ancak bir süre sonra kapatıldıysa. Kanalları el ile başlattığınızda bu özellikle önemlidir.
- Yerel kuyruk yöneticisinden gönderilen isteklerin hedef kuyruk yöneticisine anlamlı olduğunu doğrulayın. Örneğin, uzak kuyruk yöneticisinde desteklenmeyen parametreleri içeren istekler.
- Ayrıca bkz. [MQSC komutlarıyla ilgili sorunların çözülmesi](#).

Kodlanmış karakter takımları arasında veri dönüştürme

IBM MQ tanımlı biçimlerdeki (yerleşik biçimler olarak da bilinir) ileti verileri, her iki karakter kümesinin tek bir dille ya da benzer dillerden oluşan bir grupta ilgili olması koşuluyla, kuyruk yöneticisi tarafından bir kodlanmış karakter kümesinden diğerine dönüştürülebilir.

Örneğin, her ikisi de Batı Avrupa dilleri için geçerli olduğundan, tanıttıcısı (CCSID) 850 ve 500 olan kodlanmış karakter kümeleri arasında dönüştürme desteklenir.

EBCDIC yeni satır (NL) karakter dönüştürmeleri için, [mq5.ini dosyasının tüm kuyruk yöneticileri kısmı ve AMQ_CONVEBCDICNEWLINE](#) ortam değişkenine bakın.

Desteklenen dönüştürmeler, [Veri dönüştürme işlemi](#) tanımlanır.

IBM MQ 9.2.0' den CCSID 37 ile 500 arasında dönüştürme IBM MQ Appliance, Windows, Linux ve macOS üzerinde desteklenir.

Bir kuyruk yöneticisi iletileri yerleşik biçimlerde dönüştüremediği zaman

CCSID 'leri farklı ulusal dil gruplarını gösteriyorsa, kuyruk yöneticisi iletileri yerleşik biçimlerde otomatik olarak dönüştüremez. Örneğin, bir kodlanmış karakter takımındaki karakterlerin çoğu diğerinde gösterilemediği için, CCSID 850 ile CCSID 1025 (Kiril komut dosyası kullanan diller için EBCDIC kodlanmış karakter takımı) arasında dönüştürme desteklenmez. Farklı ulusal dillerde çalışan kuyruk yöneticilerinden oluşan bir ağınız varsa ve bazı kodlanmış karakter takımları arasında veri dönüştürme desteklenmiyorsa, varsayılan dönüştürmeyi etkinleştirebilirsiniz.

`ccsid_part2.tbl` uygulamasının uygulandığı platformlar için daha fazla bilgi için bkz. "[Varsayılan veri dönüştürmenin belirtilmesi](#)" sayfa 223 kullanma `ccsid_part2.tbl` . `ccsid_part2.tbl` dosyasının geçerli olduğu platformlar dışındaki platformlarda varsayılan veri dönüştürmesi "[Varsayılan veri dönüştürme](#)" sayfa 221 başlıklı konuda açıklanmıştır.

Geliştirilmiş Unicode veri dönüştürme desteği

IBM MQ 9.0' den önce, ürünün önceki sürümleri Temel Çok Dilli Planın ötesinde Unicode kod noktaları içeren verilerin dönüştürülmesini desteklemiyordu (U+FFFF üzerindeki kod noktaları). Unicode veri dönüştürme desteği, UTF-16' ın 2 baytlık sabit genişlikli bir alt kümesi olan UTF-8 ya da UCS-2 kodlu Unicode 3.0 standardında tanımlanan kod noktalarıyla sınırlıydı.

Ürün, IBM MQ 9.0' den veri dönüştürmede Unicode 8.0 standardında tanımlanan tüm Unicode karakterlerini destekler. Bu, vekil çiftler de dahil olmak üzere UTF-16 için tam desteği içerir (X' D800 ' - X' DFFF ' aralığında U+FFFF üzerinde bir Unicode kod noktasını gösteren 2 baytlık UTF-16 karakter çifti).

Karakter sıralarının birleştirilmesi, bir CCSID 'de önceden oluşturulmuş bir karakterin başka bir CCSID' de birleştirilmiş bir karakter sırasıyla eşlendiği durumlarda da desteklenir.

Unicode ve CCSID 'lerden 1388, 1390, 1399, 4933, 5488 ve 16884 'e veri dönüştürme, bazı altyapılarda, Unicode tamamlayıcı düzlemlerinde kod noktalarıyla eşlenenler de içinde olmak üzere, bu CCSID' ler için tanımlı olan tüm kod noktalarını destekleyecek şekilde genişletilmiştir.

CCSID 1390, 1399 ve 16884 olduğunda, bu, JIS X 0213 (JIS2004) standardında tanımlanan karakterleri içerir.

Unicode ve altı yeni CCSID ' ye (1374 'ten 1379 'a) dönüştürme için de destek eklenmiştir.

`ccsid_part2.tbl` dosyası

IBM MQ 9.0 ' den ek bir dosya (`ccsid_part2.tbl`) sağlanır.

`ccsid_part2.tbl` dosyası, `ccsid.tbl` dosyasından önceliklidir ve:

- CCSID girişleri eklemenizi ya da değiştirmenizi sağlar
- Varsayılan veri dönüştürmeyi belirtin
- Farklı komut düzeyleri için veri belirtin

ccsid_part2.tbl yalnızca aşağıdaki platformlar için geçerlidir:

- **Linux** Linux -tüm sürümler
- **Windows** Windows

Windows IBM MQ for Windows sistemlerinde ccsid_part2.tbl , varsayılan olarak *MQDataRoot\conv\table* dizininde bulunur. Ayrıca, IBM MQ for Windows üzerinde desteklenen tüm kod kümelerini kaydeder.

Linux IBM MQ for Linux sistemlerinde ccsid_part2.tbl , *MQDataRoot/conv/table* dizininde bulunur ve desteklenen kod kümeleri, IBM MQ tarafından sağlanan dönüştürme çizelgelerinde tutulur.

ccsid_part2.tbl dosyası, ek CCSID bilgileri sağlamak için önceki IBM MQ sürümlerinde kullanılan ccsid.tbl dosyasının yerine geçerse de, ccsid.tbl dosyası IBM MQ tarafından ayrıştırılmaya devam eder ve bu nedenle silinmemelidir.

Daha fazla bilgi için bkz [“ccsid_part2.tbl dosyası” sayfa 222](#).

ccsid.tbl dosya

ccsid_part2.tbl ' in geçerli olduğu platformlar dışındaki platformlarda, ccsid.tbl dosyası aşağıdaki amaçlarla kullanılır:

- **AIX** AIX işletim sisteminde, desteklenen kod kümeleri işletim sistemi tarafından içeride tutulur.
- Herhangi bir ek kod kümesini belirtir. Ek kod kümeleri belirtmek için ccsid.tbl dosyasını düzenlemeniz gerekir (dosyada bunun nasıl yapılacağını gösteren kılavuz bilgiler bulunur).
- Varsayılan veri dönüştürmelerini belirtir.

ccsid.tbl içinde kaydedilen bilgileri güncelleyebilirsiniz; örneğin, işletim sisteminizin gelecekteki bir yayını ek kodlanmış karakter kümelerini destekliyorsa bunu yapmak isteyebilirsiniz.

Varsayılan veri dönüştürme

IBM MQ 9.0 ' den varsayılan veri dönüştürme yöntemi aşağıdaki platformlarda değişmiştir:

- **Linux** Linux -tüm sürümler
- **Windows** Windows

Daha fazla bilgi için bkz. [“Varsayılan veri dönüştürmenin belirtilmesi” sayfa 223](#) kullanma ccsid_part2.tbl .

Normalde veri dönüştürmenin desteklenmediği iki makine arasında kanallar oluşturursanız, kanalların çalışması için varsayılan veri dönüştürmeyi etkinleştirmeniz gerekir.

ccsid_part2.tbl dışındaki platformlarda, varsayılan veri dönüştürmeyi etkinleştirmek için, ccsid.tbl dosyasını düzenleyerek varsayılan bir EBCDIC CCSID 'si ve varsayılan bir ASCII CCSID 'si belirleyin. Bunun nasıl yapılacağını anlatan yönergeler dosyaya eklenir. Bunu kanalları kullanarak bağlanacak tüm makinelerde yapmalısınız. Değişikliğin yürürlüğe girmesi için kuyruk yöneticisini yeniden başlatın.

Varsayılan veri dönüştürme işlemi aşağıdaki gibidir:

- Kaynak ve hedef CCSID 'ler arasında dönüştürme desteklenmiyorsa, ancak kaynak ve hedef ortamların CCSID 'leri EBCDIC ya da her ikisi de ASCII ise, karakter verileri dönüştürmeden hedef uygulamaya geçirilir.
- CCSID 'lerden biri ASCII kodlanmış karakter takımını gösteriyorsa ve diğeri EBCDIC kodlanmış karakter takımını gösteriyorsa, IBM MQ verileri ccsid.tbl içinde tanımlanan varsayılan veri dönüştürme CCSID 'lerini kullanarak dönüştürür.

Not: Dönüştürülmekte olan karakterleri, ileti için belirlenen kodlanmış karakter takımındaki ve varsayılan kodlanmış karakter takımındaki aynı kod değerlerine sahip karakterlerle sınırlandırmayı deneyin. Yalnızca IBM MQ nesne adları için geçerli olan karakter kümesini kullanırsanız (Adlandırma IBM MQ nesnelere tanımlandığı şekilde) Genel olarak bu gereksinimi karşılayacaksınız. Japonya 'da kullanılan EBCDIC CCSID 'ler 290, 930, 1279 ve 5026 için özel durumlar ortaya çıkar; burada küçük harfli karakterler, diğer EBCDIC CCSID' lerinde kullanılanlardan farklı kodlara sahiptir.

İletileri kullanıcı tanımlı biçimlerde dönüştürme

Kuyruk yöneticisi, kullanıcı tanımlı biçimlerdeki iletileri bir kodlanmış karakter kümesinden diğerine dönüştüremiyor. Verileri kullanıcı tanımlı bir biçimde dönüştürmeniz gerekiyorsa, bu biçimlerin her biri için bir veri dönüştürme çıkışı sağlamanız gerekir. Karakter verilerini kullanıcı tanımlı biçimlerde dönüştürmek için varsayılan CCSID 'leri kullanmayın. Verileri kullanıcı tanımlı biçimlerde dönüştürme ve veri dönüştürme çıkışları yazma hakkında daha fazla bilgi için [Veri dönüştürme çıkışları yazmabaşlıklı konuya](#) bakın.

Kuyruk yöneticisi CCSID 'sinin değiştirilmesi

Kuyruk yöneticisinin CCSID 'sini değiştirmek için **ALTER QMGR** komutunun **CCSID** özneliğini kullandığınızda, komut sunucusu ve kanal programları da içinde olmak üzere çalışan tüm uygulamaların durdurulmasını ve yeniden başlatılmasını sağlamak için kuyruk yöneticisini durdurun ve yeniden başlatın.

Kuyruk yöneticisi CCSID 'si değiştirildiğinde çalışan uygulamalar, varolan CCSID 'yi kullanmaya devam edeceği için bu gereklidir.

ccsid_part2.tbl dosyası

ccsid_part2.tbl dosyası, ek CCSID bilgileri sağlamak için kullanılır. ccsid_part2.tbl dosyası, IBM MQ 9.0' den önce kullanılan ccsid.tbl dosyasının yerini alır.

Not: Ek CCSID bilgileri sağlamak için IBM MQ 9.0 ' den önce kullanılan ccsid.tbl dosyası, IBM MQ tarafından ayrıştırılmaya devam eder ve silinmemelidir. Ancak, ccsid_part2.tbl içindeki girdiler ccsid.tbl içindeki diğer girdilerden daha önceliklidir.

ccsid_part2.tbl nedeniyle ccsid.tbl yerine ccsid_part2.tbl kullanmalısınız:

- Unicode kodlama değerleri için destek içerir. Ürün, IBM MQ 9.0' den UTF-16 için tam destek de dahil olmak üzere, veri dönüştürmede Unicode 8.0 standardında tanımlanan tüm Unicode karakterlerini destekler. Daha fazla bilgi için bkz [“Kodlanmış karakter takımları arasında veri dönüştürme” sayfa 220.](#)
- CCSID girişlerinin sürümünü belirlemenizi, böylece girişlerin yalnızca seçilen komut düzeylerine uygulanabilmesini sağlar.



ccsid_part2.tbl dosyasını kullanarak yapabilecekleriniz:

- CCSID girişleri eklenmesi ya da değiştirilmesi
- Varsayılan veri dönüştürmeyi belirtin
- Farklı komut düzeyleri için veri belirtin

ccsid_part2.tbl dosyası yalnızca aşağıdaki altyapılar için geçerlidir:

-  Linux -tüm sürümler
-  Windows

ccsid_part2.tbl dosyasının konumu altyapınıza bağlıdır:

-  Tüm Linux sürümlerinde *MQDataRoot/conv/table* dizini.
-  Windows üzerindeki *MQDataRoot\conv\table* dizini.

CCSID girişleri eklenmesi ya da değiştirilmesi

ccsid_part2.tbl dosyasındaki bir girdi aşağıdaki biçimdedir:

```
<CCSID number> <Base CCSID> <DBCS CodePage> <SBCS CodePage>  
<Type> <Encoding> <ACRI> <Name>
```

CCSID 1200 (UTF-16) için örnek bir giriş:

```
1200 1200 1200 1200 3 8 0 UTF-16
```

Not: ACRI değerine ilişkin daha fazla ayrıntı için ccsid_part2.tbl dosyasındaki açıklamaya bakın.

ccsid_part2.tbl biçiminde:

Tip şuna eşit olabilir:

1=SBCS

2=DBCS

3=MBCS

Kodlama şuna eşit olabilir:

1=EBCDIC

2 = ASCII

3 = ISO

4 = UCS-2

5 = UTF-8

6 = Euc

7 = GB18030

8 = UTF-16

9 = UTF-32

Dosyayı düzenlerken şunları yapın:

- Bir satırın başındaki # simgesini kullanarak bir açıklama belirtebilir. Bu, IBM MQ ' in hattı ayrıştırılmayı denemesini önler.
- Satır içi açıklamalar sağlanamıyor.
- Boş satırlar oluşturmadığınızdan emin olun.
- Dosyanın sonuna yeni girişler eklenmemelidir.

ACRI çizelge bilgilerinden önce yeni CCSID girişleri eklenmelidir.

Varsayılan veri dönüştürmenin belirtilmesi

İki CCSID arasında dönüştürme desteklenmiyorsa, ASCII ya da benzeri CCSID 'ler ile EBCDIC CCSID ' ler arasında dönüştürme için kullanılan varsayılan dönüştürme CCSID ' lerini tanımlayabilirsiniz.

Bu işlevi etkinleştirirseniz, iletim ve ileti üstbilgileri için varsayılan dönüştürme kullanılır ve kullanıcı verileri dönüşümünde de kullanılabilir.

Varsayılan dönüştürmeler, aşağıdakine benzer iki satır yaratılarak etkinleştirilir:

```
default      0      500      1      1      0  
default      0      850      1      2      0
```

İlk satır EBCDIC CCSID 'leri için varsayılan değeri 500 olarak, ikinci satır ise ASCII ve benzeri CCSID ' ler için varsayılan değeri 850 olarak ayarlar.

Farklı komut düzeyleri için veri belirtilmesi

IBM MQ komut düzeylerinin farklı komut düzeylerine ilişkin CCSID girişlerini belirlemek için, sonraki bölümün uygulanmasını istediğiniz IBM MQ komut düzeyinin (ya da komut düzeylerinin) izlediği bir iki nokta işareti kullanın.

Sayı, kuyruk yöneticisinin ya da istemcinin çalışması gereken en düşük komut düzeyini gösterir. Örneğin, yürürlükteki kuyruk yöneticisi komut düzeyi 900 ise ve 800 ya da 900 komut düzeyi işaretiyle karşılaşırsa, CCSID ' ler okunur.

Ancak, 800 düzeyindeki bir kuyruk yöneticisi, 900 bölümündeki CCSID ' leri yoksayar.

Belirlenen komut düzeyi, yeni bir komut düzeyi işareti bulununcaya kadar, bir komut düzeyi işaretinden sonra karşılaşılan tüm CCSID girişleri için geçerlidir.

Komut düzeyini tüm komut düzeylerine ayarlamanız gerekiyorsa, sıfır sayısını belirtin.

İlk ayrıştırma `ccsid_part2.tbl` sırasında IBM MQ , karşılaşılan tüm CCSID ' leri IBM MQ komut düzeylerinde geçerli olarak işler.

Sürüm oluşturma yalnızca IBM MQ ilk komut düzeyi işaretiyle karşılaştığında kullanılır.

Aşağıdaki kod parçacığı, Sürüm Oluşturmanın kullanımına bir örnek gösterir:

```
# Comment Block
# End of Comment Block
# Because no command level flag is specified and we're at the start of the file
# the following CCSIDs will be read on all versions
  819  819  0  819  1  3  0  ISO8859-1
  923  923  0  923  1  3  0  ISO8859-15
 1051 1051  0 1051  1  3  0  IBM-1051
# The colon :900 below shows that the CCSIDs after will only be for MQ cmd level 900 and above
:900
  8629 437  0  437  1  2  0  IBM-437
 12725 437  0  437  1  2  0  IBM-437
 16821 437  0  437  1  2  0  IBM-437
 20917 437  0  437  1  2  0  IBM-437
# The colon :0 below shows that the CCSIDs after will be for all version of MQ
:0
  4946 850  0  850  1  2  0  IBM-850
 33618 850  0  850  1  2  0  IBM-850
 61697 850  0  850  1  2  0  IBM-850
 61698 850  0  850  1  2  0  IBM-850
```

YönetmeManaged File Transfer

Managed File Transfer' yi yönetmek için Managed File Transfer komutlarını kullanın. IBM MQ Explorer ögesini bazı yönetim görevleri için de kullanabilirsiniz.

Bir aracı komut kuyruğuna ileti yerleştirerek aktarma işleminin başlatılması

Bir dosya aktarma iletisini kaynak aracının komut kuyruğuna koyarak da dosya aktarma işlemi başlatabilirsiniz. Örnek bir komut kuyruğu adı: `SYSTEM.FTE.COMMAND.AGENT01`. İletinin doğru kaynak aracının komut kuyruğuna ulaştığından emin olmanız gerekir; ileti, XML ' deki kaynak bilgilerle eşleşmeyen bir aracı tarafından alınır, ileti reddedilir.

Aktarma isteği XML ' i `FileTransfer.xsd` şemasına uymalı ve kök öge olarak `<request>` ögesini kullanmalıdır. Aktarma isteği iletisinin yapısı ve içeriğine ilişkin bilgi için [Dosya aktarma isteği iletisi biçimi](#) konusuna bakın. Aktarma isteği iletisini bir aracı komut kuyruğuna koyma şeklinin göre ve özgüdür. Örneğin, kuyruğa programlı olarak bir ileti koymak için IBM MQ Java API ' sini kullanabilirsiniz.

İlgili kavramlar

[“Dosyalardan iletilere veri aktarılması” sayfa 277](#)

Bir dosyadan IBM MQ kuyruğundaki tek bir iletiye ya da birden çok iletiye veri aktarmak için Managed File Transfer ' in dosyadan iletiye özelliğini kullanabilirsiniz.

[“İletilerden dosyalara veri aktarılması” sayfa 291](#)

Managed File Transfer iletimden dosyaya özelliği, IBM MQ kuyruğundaki bir ya da daha çok iletimdosyaya, veri kümesine (z/OSüzerindeki) ya da kullanıcı dosya alanına veri aktarmanızı sağlar. IBM MQ iletimlerini yaratan ya da işleyen bir uygulamanız varsa, bu iletileri Managed File Transfer ağındaki herhangi bir sistemde bulunan bir dosyaya aktarmak için Managed File Transfer iletimden dosyaya aktarma yeteneğini kullanabilirsiniz.

“Protokol köprüsü” sayfa 302

İletişim kuralı köprüsü, Managed File Transfer (MFT) ağındaki yerel etki alanınızda ya da uzak bir konumda MFT ağındaki bir dosya sunucusunda saklanan dosyalara erişmesini sağlar. Bu dosya sunucusu FTP, FTPS ya da SFTP ağ iletim kurallarını kullanabilir. Her dosya sunucusu için en az bir özel olarak ayrılmış aracı gerekir. Özel olarak ayrılmış ajan, protokol köprüsü aracı olarak bilinir. Bir köprü aracı birden çok dosya sunucusuyla etkileşimde bulunabilir.

“IBM Integration Bus içinden MFT ile çalışma” sayfa 340

FTEOutput ve FTEInput düğümlerini kullanarak IBM Integration Bus içinden Managed File Transfer ile çalışabilirsiniz.

“MFT kurtarma ve yeniden başlatma” sayfa 340

Temsilciniz ya da kuyruk yöneticiniz herhangi bir nedenle (örneğin, bir güç ya da ağ arızası nedeniyle) kullanılmıyorsa, Managed File Transfer bu senaryolarda aşağıdaki gibi kurtarma gerçekleştirir:

İlgili görevler

“MFT aracısını başlatma” sayfa 226

Dosya aktarımı için bir Managed File Transfer aracısını kullanmadan önce aracıyı başlatmanız gerekir.

“Yeni dosya aktarımı başlatılması” sayfa 233

IBM MQ Explorer ' den ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir gruptaki tek bir dosyayı ya da birden çok dosyayı aktarmayı seçebilirsiniz.

“Devam eden dosya aktarımlarını izleme” sayfa 240

Devam eden bir dosya aktarımını, IBM MQ Exploreriçindeki **Yönetilen Dosya Aktarımı-Yürürlükteki Aktarma İlerlemesi** sekmesini kullanarak izleyebilirsiniz. Bu dosya aktarımı IBM MQ Explorer ya da komut satırından başlatılabilir. Sekme, zamanlanan aktarımların başladığı noktada zamanlanan aktarımların ilerleme durumunu da görüntüler.

“Aktarma Günlüğündeki Dosya Aktarımlarının Durumunun Görüntülenmesi” sayfa 242

IBM MQ Exploreriçindeki **Aktarma Günlüğü** ' nü kullanarak dosya aktarımlarının ayrıntılarını görüntüleyebilirsiniz. Bunlar, komut satırından ya da IBM MQ Explorer' den başlatılan aktarımlar olabilir. **Aktarma Günlüğü** ' nde görüntüleneni de uyarlayabilirsiniz.

“MFT kaynaklarının izlenmesi” sayfa 244

Managed File Transfer kaynaklarını izleyebilirsiniz; örneğin, bir kuyruk ya da dizin. Bu kaynaktaki bir koşul karşılandığında, kaynak izleme programı dosya aktarma gibi bir görevi başlatır. IBM MQ Exploreriçin Managed File Transfer eklentisindeki **fteCreateMonitor** komutunu ya da **İzleme Programları** görünümünü kullanarak bir kaynak izleme programı yaratabilirsiniz.

“Dosya aktarma şablonlarıyla çalışılması” sayfa 274

Yinelenen ya da karmaşık aktarımlara ilişkin ortak dosya aktarma ayarlarını saklamak için dosya aktarma şablonlarını kullanabilirsiniz. **fteCreateTemplate** komutunu kullanarak komut satırından bir aktarma şablonu yaratın ya da **Yönetilen Dosya Aktarımı İçin Yeni Şablon Yarat** sihirbazını kullanarak aktarma şablonu yaratmak için IBM MQ Explorer komutunu kullanın ya da **Aktarma ayarlarını şablon olarak sakla** onay kutusunu seçerek dosya aktarımı yaratırken bir şablonu saklayın. **Aktarma Şablonları** penceresi, Managed File Transfer ağındaki yarattığınız tüm aktarma şablonlarını görüntüler.

“MFT araçlarını listeleme” sayfa 231

Komut satırını ya da IBM MQ Explorer' u kullanarak belirli bir kuyruk yöneticisine kayıtlı Managed File Transfer araçlarını listeleyebilirsiniz.

“MFT aracısının durdurulması” sayfa 232

Bir Managed File Transfer aracısını komut satırından durdurabilirsiniz. Bir aracıyı durdurduğunuzda, aracıyı durdurur ve aracının durdurmadan önce geçerli dosya aktarımını tamamlamasına izin vermeniz gerekir. Bir aracıyı hemen durdurmak için komut satırında **-i** parametresini de belirtebilirsiniz. Aracı durduğunda, yeniden başlatıncaya kadar dosyaları aktarmak için bu aracıyı kullanamazsınız.

MFT günlük kaydedicisi yapılandırılması

İlgili başvurular

Dosyaların aktarılmasına ilişkin yönergeler

MFT aracısını başlatma

Dosya aktarımı için bir Managed File Transfer aracısını kullanmadan önce aracıyı başlatmanız gerekir.

Bu görev hakkında

Komut satırından bir Managed File Transfer Agent başlatabilirsiniz. Bu durumda, sistemde oturum kapattığınızda aracı işlemi durur.

ALW AIX, Linux, and Windows işletim sisteminde, bir aracıyı sistemde oturum kapattığınızda çalışmaya devam edecek ve dosya aktarımlarını almaya devam edecek şekilde yapılandırabilirsiniz.

z/OS z/OS'ta, aracıyı etkileşimli bir oturum gerekmeden JCL' den başlatılan bir görev olarak başlatılacak şekilde yapılandırabilirsiniz.

Bir aracı çalışırken kurtarılamaz bir hatayla karşılaşırsa, ilk hata verileri yakalama (FDC) oluşturulur ve aracı durdurulur.

Yordam

- Bir aracıyı komut satırından başlatmak için **fteStartAgent** komutunu kullanın.

Daha fazla bilgi için bkz. [fteStartAgent](#).

- **ALW**

Bir aracıyı, sistemde oturum kapattığınızda çalışmaya devam edecek şekilde yapılandırmak için:

- **Windows** Windows' da aracıyı Windows hizmeti olarak çalışacak şekilde yapılandırın. Daha fazla bilgi için bkz “MFT aracısını Windows hizmeti olarak başlatma” sayfa 227.

- **Linux** **AIX** AIX and Linux üzerinde, aracıyı bir komut dosyası kullanarak yeniden yükleme sırasında otomatik olarak başlatılacak şekilde yapılandırın. Daha fazla bilgi için bkz “AIX and Linux sistemi başlangıcında bir MFT aracısının başlatılması” sayfa 228.

- **z/OS**

z/OS üzerinde, aracıyı etkileşimli bir oturum gerekmeden JCL ' den başlatılan bir görev olarak başlayacak şekilde yapılandırın.

Daha fazla bilgi için bkz “z/OS üzerinde bir MFT aracısının başlatılması” sayfa 230.

İlgili görevler

[“MFT araçlarını listeleme” sayfa 231](#)

Komut satırını ya da IBM MQ Explorer' u kullanarak belirli bir kuyruk yöneticisine kayıtlı Managed File Transfer araçlarını listeleyebilirsiniz.

[“MFT aracısının durdurulması” sayfa 232](#)

Bir Managed File Transfer aracısını komut satırından durdurabilirsiniz. Bir aracıyı durdurduğunuzda, aracıyı durdurur ve aracının durdurmadan önce geçerli dosya aktarımını tamamlamasına izin vermeniz gerekir.

Bir aracıyı hemen durdurmak için komut satırında **-i** parametresini de belirtebilirsiniz. Aracı durduğunda, yeniden başlatıncaya kadar dosyaları aktarmak için bu aracıyı kullanamazsınız.

İlgili başvurular

[MFT aracı durumu değerleri](#)

[fteStartAgent](#)

Bir aracıyı Windows hizmeti olarak başlatabilirsiniz; böylece Windows oturumunu kapattığınızda aracınız çalışmaya devam eder ve dosya aktarımlarını alabilir.

Bu görev hakkında

Windows'da, bir aracıyı komut satırından başlattığınızda, aracı işlemi Windows' da oturum açmak için kullandığınız kullanıcı adını kullanarak çalışır. Sistemde oturum kapattığınızda aracı işlemi durur. Aracının durdurulmasını önlemek için bir aracıyı Windows hizmeti olarak çalışacak şekilde yapılandırabilirsiniz. Windows hizmeti olarak çalıştırılması, Windows ortamı başlatıldığında ya da yeniden başlatıldığında araçları otomatik olarak başlatılacak şekilde yapılandırmanızı da sağlar.

Windows hizmeti olarak çalışan bir aracıyı başlatmak için aşağıdaki adımları tamamlayın. Aracıyı Windows hizmeti olarak çalıştırmak için desteklenen Windows sürümlerinden birinde Managed File Transfer komutunu çalıştırmanız gerekir. Desteklenen ortamların listesi için bkz. [IBM MQ için Sistem Gereksinimleri](#).

Tam adımlar, önceden bir aracı oluşturup oluşturmadığınıza ya da bir aracı oluşturup oluşturmadığınıza bağlıdır. Her iki seçenek de aşağıdaki adımlarda açıklanır.

Yordam

1. Bir Managed File Transfer aracı oluşturuyorsanız, **fteCreateAgent**, **fteCreateCDAgent** ya da **fteCreateBridgeAgent** komutunu kullanın. Aracıyı bir Windows hizmeti olarak çalıştırmak için **-s** parametresini belirtin. Aşağıdaki örnekte, aracı kuyruk yöneticisi QMGR1 olan AGENT1 aracı oluşturulur. Windows hizmeti, ilişkili bir parolası olan fteuser kullanıcı adı kullanılarak çalışır. ftepassword.

```
fteCreateAgent -agentName AGENT1 -agentQMGR QMGR1 -s -su fteuser -sp ftepassword
```

İsteğe bağlı olarak, **-s** değıştirgesinden sonra hizmet için bir ad belirtebilirsiniz. Bir ad belirtmezseniz, hizmet mqmftAgentAGENTQMGR olarak adlandırılır; burada AGENT belirttiğiniz aracı adıdır ve QMGR aracı kuyruk yöneticisi adıdır. Bu örnekte, hizmet için varsayılan ad mqmftAgentAGENT1QMGR1' dir.

Not: -su değıştirgesini kullanarak belirttiğiniz Windows kullanıcı hesabının **Log on as a service** hakları olmalıdır. Bunun nasıl yapılandırılacağına ilişkin bilgi için [Windows hizmeti olarak çalışan bir MFT aracıyla ya da kaydediciyle ilgili sorun giderme başlıklı konuya](#) bakın.

Daha fazla bilgi için bkz. [fteCreateAgent](#), [fteCreateCDAgent: Connect:Direct bridge agent yaratılması](#) ya da [fteCreateBridgeAgent \(MFT iletişim kuralı köprüsü aracı yaratılması ve yapılandırılması\)](#).

2. Aracı yaratmak için önceki adımı izlediyseniz, **fteCreateAgent**, **fteCreateCDAgent** ya da **fteCreateBridgeAgent** komutuyla oluşturulan MQSC komutlarını çalıştırın. Bu komutlar, aracının gerektirdiği IBM MQ kuyruklarını oluşturur.

Örneğin, AGENT1 adlı bir aracı, QMGR1 adlı bir aracı kuyruk yöneticisi ve COORDQMGR1 adlı bir koordinasyon kuyruk yöneticisi için şu komutu çalıştırın:

```
runmqsc QMGR1 MQ_DATA_PATH\mqft\config\COORDQMGR1\agents\AGENT1\AGENT1_create.mqsc
```

3. Bir aracı oluşturmak için önceki adımları izlemediyseniz ve bunun yerine var olan bir aracıyı Windows hizmeti olarak çalışacak şekilde yapılandırmak istiyorsanız, önce aracınızı durdurun ve daha sonra, aracının yapılandırmasını değıştirin.

- a) Aşağıdaki örnekte AGENT1 adlı bir aracı kullanılmaktadır. Aşağıdaki komutu çalıştırın:

```
fteStopAgent AGENT1
```

- b) Aracıyı Windows hizmeti olarak çalışacak şekilde yapılandırmak için **fteModifyAgent** komutunu kullanın:

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

Daha fazla bilgi için bkz. [fteModifyAgent: MFT aracısını Windows hizmetolarak çalıştırma](#).

4. **fteStartAgent** komutunu kullanarak aracıyı başlatın. Diğer bir seçenek olarak, hizmeti başlatmak için Windows masaüstü başlatma menüsünden seçilen Denetim Masası 'nda (Control Panel) bulunan Windows Services aracını kullanabilirsiniz.

```
fteStartAgent AGENT1
```

Windowsoturumunu kapatsanız bile hizmet çalışmaya devam eder. Windows yeniden başlatıldığında da hizmetin yeniden başlatıldığından emin olmak için Windows Services aracındaki **Startup Type** (Başlatma Tipi) alanı varsayılan olarak **Automatic** (Otomatik) olarak ayarlanır. Windows yeniden başlatıldığında hizmetin yeniden başlatılmasını istemiyorsanız, bu seçeneği **El ile** olarak değiştirin.

5. İsteğe bağlı: Aracıyı durdurmak için **fteStopAgent** komutunu kullanın ya da Windows Services aracını kullanın. Örneğin, komut satırından şu komutu çalıştırın:

```
fteStopAgent AGENT1
```

- **fteStopAgent** komutunu bir hizmet olarak çalıştırdığınızda, komut her zaman bu parametreyi belirtip belirtmediğinizden bağımsız olarak **-i** parametresini kullanarak çalışır. **-i** parametresi, devam eden aktarımları tamamlamadan aracıyı hemen durdurur. Bunun nedeni, Windows hizmetinin bir sınırlamasıdır.

Sonraki adım

Windows hizmetinizi başlatmakta sorun yaşıyorsanız, bkz. [Bir MFT aracısıyla ilgili sorunları giderme ya da Windows hizmetolarak çalışan kaydedici](#). Bu konuda, Windows hizmet günlük dosyalarının konumu da açıklanmaktadır.

İlgili görevler

[“MFT araçlarını listeleme” sayfa 231](#)

Komut satırını ya da IBM MQ Explorer' u kullanarak belirli bir kuyruk yöneticisine kayıtlı Managed File Transfer araçlarını listeleyebilirsiniz.

[“MFT aracısının durdurulması” sayfa 232](#)

Bir Managed File Transfer aracısını komut satırından durdurabilirsiniz. Bir aracıyı durdurduğunuzda, aracıyı durdurur ve aracının durdurmadan önce geçerli dosya aktarımını tamamlamasına izin vermeniz gerekir. Bir aracıyı hemen durdurmak için komut satırında **-i** parametresini de belirtebilirsiniz. Aracı durduğunda, yeniden başlatıncaya kadar dosyaları aktarmak için bu aracıyı kullanamazsınız.

İlgili başvurular

[Windows hizmeti olarak çalışan bir MFT aracısıyla ya da kaydediciyle ilgili sorunları giderme](#)

[fteCreateAracı](#)

[fteCreateCDAgent: Connect:Direct köprü aracısı oluşturma](#)

[fteCreateBridgeAgent \(MFT iletişim kuralı köprüsü aracısı oluşturma ve yapılandırma\)](#)

[fteModifyAgent: Bir MFT aracısını Windows hizmeti olarak çalıştırma](#)

Linux

AIX

AIX and Linux sistemi başlangıcında bir MFT aracısının başlatılması

Bir Managed File Transfer Agent , sistem başlatılırken AIX and Linuxüzerinde başlatılacak şekilde yapılandırılabilir. Oturumu kapattığınızda, aracınız çalışmaya devam eder ve dosya aktarımlarını alabilir.

Şu Managed File Transfer komutlarından (**fteCreateAgent**, **fteCreateCDAgent** ya da **fteCreateBridgeAgent**) birini kullanarak bir aracı oluşturup yapılandırdığınızda, AIX and Linux

makinelerinde yeniden yükleme sırasında otomatik olarak başlatılacak şekilde yapılandırmak için aşağıdaki komutu yürüten bir komut dosyası kullanabilirsiniz:

```
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

Burada `mq_install_root` , gerekli Managed File Transfer kuruluşunun kök dizinidir; varsayılan: `/opt/mqm` ve `aracı_adi` , başlatılacak Managed File Transfer Agent ' in adıdır. Bu komut dosyasının kullanımı, işletim sistemine bağlı olarak değişir. Örneğin, Linux'ta kullanılabilir ek seçenekler vardır.

Linux

Linux

Linux sistemleri için, sistem önyükleme işlemi sırasında uygulamaları başlatabileceğiniz birden çok yol vardır. Genel olarak, aşağıdaki adımları izleyin:

1. İçeriği `/etc/rc.mqmft` olan bir dosya oluşturun:

```
#!/bin/sh
su -l mqmft_user"-c mq_install_root/bin/fteStartAgent agent_name"
```

Burada `mqmft_user` , aracı işleminin çalıştırılacağı kullanıcı kimliğidir. Bu kullanıcı kimliği `mqm` grubunun bir üyesi olmalıdır.

2. Dosyayı yürütülebilir yap, örneğin:

```
chmod 755 /etc/rc.mqmft
```

3. `/etc/inittab`' e aşağıdaki satırı ekleyin:

```
mqmft:5:boot:/etc/rc.mqmft
```

Linux üzerinde önyükleme sırasında bir aracıyı başlatmanın diğer yolları, komut dosyası satırlarının `/etc/rc.d/rc.local` dosyasına ya da Linux SuSe' de `/etc/init.d/boot.local` dosyasına eklenmesini içerir. Ortamınız için en uygun yöntemi seçmelisiniz. Aşağıda, desteklenen belirli Linux dağıtımlarında başlatma sırasında bir aracıyı başlatmanın diğer yollarıyla ilgili daha fazla bilgi verilmiştir:

SLES 10 ve 11

SUSE Linux Enterprise Server (SLES) 10 ve 11 sistemleri için aşağıdaki adımları izleyin:

1. Sistem kök kullanıcı kimliği olarak kendi `/etc/init.d/rc.rclocal` dosyanızı oluşturun.
2. `rc.rclocal` dosyasına aşağıdaki satırları ekleyin:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides: rc.rclocal
# Required-Start: $network $syslog
# Required-Stop: $network $syslog
# Default-Stop: 0 1 2 6
# Description: MQMFT agent startup
### END INIT INFO
su -l mqmft_user"-c mq_install_root/bin/fteStartAgent agent_name"
```

3. Aşağıdaki komutları çalıştırın:

```
chmod 755 rc.rclocal
chkconfig --add rc.rclocal
```

systemd ile Linux üzerinde Managed File Transfer araçlarını başlatma

Linux

Aşağıdaki yordamı gerçekleştirin:

1. /etc/systemd/system klasöründe bir dosya oluşturun ve dosyayı adlandırın; örneğin, <agentname>.service. Aşağıdaki içeriği ekleyin; burada <agentname> , MFT_AGT_LNX_0.

```
# vi /etc/systemd/system/MFT_AGT_LNX_0.service
[Unit]
Description=IBM MQ MFT MFT_AGT_LNX_0
[Service]
ExecStart=/opt/mqm/bin/fteStartAgent MFT_AGT_LNX_0
ExecStop=/opt/mqm/bin/fteStopAgent MFT_AGT_LNX_0
Type=forking
User=mqm
Group=mqm
KillMode=none
```

2. Hizmeti etkinleştirmek için aşağıdaki komutları çalıştırın:

```
# systemctl enable MFT_AGT_LNX_0
# systemctl daemon-reload
```

3. Aracıyı başlatmak ve durumunu denetlemek için aşağıdaki komutları çalıştırın:

```
# systemctl start MFT_AGT_LNX_0
# systemctl status MFT_AGT_LNX_0
```

İlgili görevler

[“MFT aracısının durdurulması” sayfa 232](#)

Bir Managed File Transfer aracısını komut satırından durdurabilirsiniz. Bir aracıyı durdurduğunuzda, aracıyı durdurur ve aracının durdurmadan önce geçerli dosya aktarımını tamamlamasına izin vermeniz gerekir. Bir aracıyı hemen durdurmak için komut satırında **-i** parametresini de belirtebilirsiniz. Aracı durduğunda, yeniden başlatıncaya kadar dosyaları aktarmak için bu aracıyı kullanamazsınız.

İlgili başvurular

[fteCreateAracı](#)

[fteCreateCDAgent: Connect:Direct köprü aracısı oluşturma](#)

[fteCreateBridgeAgent \(MFT iletişim kuralı köprüsü aracısı oluşturma ve yapılandırma\)](#)

z/OS üzerinde bir MFT aracısının başlatılması

z/OS işletim sistemlerinde, **fteStartAgent** komutunu bir z/OS UNIX System Services oturumundan çalıştırmanın yanı sıra, etkileşimli bir oturum gerekmeden JCL ' den başlatılan bir görev olarak aracı başlatabilirsiniz.

Başlatılan bir görev, belirli bir kullanıcı kimliği altında çalıştığından ve oturum kapamış olan kullanıcılardan etkilenmediğinden kullanılır.

Not: Başlatılan görevler genellikle oturum açma ayrıcalıklarına sahip olmayabilecek bir yönetici kullanıcı altında çalıştırılır ve bu nedenle aracının altında çalıştığı kullanıcı olarak z/OS sisteminde oturum açamaz. **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** komutları ve **-d** parametresi belirtilen **fteShowAgentDetails** komutu bu aracı için verilemez.

IBM MQ 9.0.2 ve IBM MQ 9.0.0 Fix Pack 1olanağından, z/OS üzerinde Managed File Transfer araçlarıyla **adminGroup** aracı özelliğini kullanabilirsiniz. Bir güvenlik yöneticisi grubu (örneğin, MFTADMIN) tanımlayabilir ve başlatılan görev kullanıcı kimliği ve denetimci TSO kimliklerini bu gruba ekleyebilirsiniz.

Aracı özellikleri dosyasını düzenleyin ve **adminGroup** özelliğini bu güvenlik yöneticisi grubunun adı olacak şekilde ayarlayın.

```
adminGroup=MFTADMIN
```


Daha sonra bu grubun üyeleri, başlatılan bir görev olarak çalışan aracı için **fteStartAgent**, **fteStopAgent** ve **fteSetAgentTraceLevel** komutlarını ve **fteShowAgentDetails** komutunu **-d** parametresiyle birlikte yürütebilirler.

Daha fazla bilgi için [MFT agent.properties](#) dosyasındaki **adminGroup** özelliğine bakın.

Bir Java uygulaması olarak aracı, bir aracı için oluşturulan Managed File Transfer komutu PDSE kitaplığı veri kümesinden BFGAGSTP üyesini kullanarak JCL ' den çalıştırabileceğiniz bir z/OS UNIX System Services uygulamasıdır. Bir MFT komutu PDSE kitaplığı veri kümesinin nasıl oluşturulacağına ve gerekli aracı için nasıl özelleştirileceğine ilişkin daha fazla bilgi için [Creating an MFT Agent or Logger command data set](#) başlıklı konuya bakın.

Aracıların uzak z/OS kuyruk yöneticilerine bağlanmasını etkinleştirme

z/OS üzerindeki bir MFT aracı, aşağıdaki senaryolarda istemci bağlantısı kullanarak z/OS üzerindeki bir kuyruk yöneticisine bağlanabilir:

-  MFT aracı IBM MQ 9.3.4 ya da daha sonraki bir sürümde ya da Long Term Support PH56722 APAR ile uygulanmış ve IBM MQ Advanced for z/OS VUEya da IBM MQ Advanced for z/OS ürün tanıtıcısıyla (PID) ilişkilendirilmiştir.
- MFT aracı IBM MQ 9.3.0 adresinde yer almakta ve IBM MQ Advanced for z/OS VUEPID 'si ile ilişkilendirilmiştir.

Farklı PID ' lerle ilgili bilgi için [IBM MQ ürün tanıtıcıları](#) ve [dışa aktarma bilgileri](#) başlıklı konuya bakın.

Bir MFT kuruluşuyla ilişkili PID ' nin ayarlanmasıyla ilgili bilgi için bkz. **fteSetProductId** .

Aracının altında çalıştığı PID, aracı başlatıldığında günlükte görüntülenir.

z/OS üzerinde çalışan bir MFT aracı, başka bir PID altında çalışır ve bağ tanımlama kipi bağlantısını kullanarak yalnızca yerel bir kuyruk yöneticisine bağlanabilir.

Bir aracı z/OS üzerinde çalışmayan bir kuyruk yöneticisine bağlanmayı denerse, BFGQM1044E iletisi yayınlanır ve aracı başlatma sona erdirilir.

İlgili başvurular

[“z/OS üzerinde bir MFT aracısının durdurulması” sayfa 232](#)

JCL ' den başlatılan bir görev olarak z/OS üzerinde Managed File Transfer Agent çalıştırıyorsanız, aracı **fteStopAgent** komutuna ek olarak z/OS işletmen komutlarını **MODIFY** ve **STOP** kabul eder.

[MFT agent.properties](#) dosyası

MFT araçlarını listeleme

Komut satırını ya da IBM MQ Explorer' u kullanarak belirli bir kuyruk yöneticisine kayıtlı Managed File Transfer araçlarını listeleyebilirsiniz.

Bu görev hakkında

Aracıları komut satırını kullanarak listelemek için bkz. [fteListAgents](#) komutu.

Aracıları IBM MQ Explorer kullanarak listelemek için, Navigator görünümünde koordinasyon kuyruğu yöneticisi adı altındaki **Aracılar** ' ı tıklatın.

Bir aracı **fteListAgents** komutuyla listelenmiyorsa ya da IBM MQ Explorer içinde görüntülenmiyorsa, sorunu bulmak ve düzeltmek için aşağıdaki konudaki tanımlama akış grafiğini kullanın: [MFT aracınız fteListAgents](#) komutuyla listelenmiyorsa ne yapmanız gerekir.

İlgili başvurular

[fteListAracıları](#): Bir koordinasyon kuyruk yöneticisine ilişkin MFT araçlarını listeler

[MFT aracıları durum değerleri](#)

[fteShowAgentDetails](#)

MFT aracısının durdurulması


Bir Managed File Transfer aracısını komut satırından durdurabilirsiniz. Bir aracıyı durdurduğunuzda, aracıyı durdurur ve aracının durdurmadan önce geçerli dosya aktarımını tamamlamasına izin vermeniz gerekir. Bir aracıyı hemen durdurmak için komut satırında **-i** parametresini de belirtebilirsiniz. Aracı durduğunda, yeniden başlatıncaya kadar dosyaları aktarmak için bu aracıyı kullanamazsınız.


Başlamadan önce

Bir kuyruk yöneticisiyle ilişkilendirilmiş araçların adlarını denetlemek istiyorsanız, araçları IBM MQ Explorer komutunu ya da komut satırını kullanarak listeleyebilirsiniz; bkz. [fteListAgents](#) komutu.

Bu görev hakkında

Bir aracıyı komut satırından durdurmak için bkz. [fteStopAgent](#).

 Bir aracı **fteStopAgent** kullanılarak denetimli bir şekilde durdurulursa, yeni yönetilen aktarım isteklerini kabul etmez ve devam eden aktarımların tamamlanmasını bekler. IBM MQ 9.3.0' den aracının hala geçici durumda olduğunu ve bu nedenle henüz kapatılmadığını ve yeniden başlatılmadığını göstermek için, devam eden aktarımlar tamamlanıncaya kadar aracı DURDURULUYOR durumuna girer. Bu durum, **fteListAgents** ve **fteShowAgentDetails** komutlarının çıkışında ve [MFT REST API sorgularında](#) MFT eklentisinin IBM MQ Explorer **Aracılar** görünümünde görünür.

 Aracınızı Windows hizmeti olarak çalışacak şekilde yapılandırdıysanız, **fteStopAgent** komutunun çalıştırılması Windows hizmetini de durdurur. Diğer bir yöntem olarak, Windows Services aracını kullanarak hizmeti durdurarak aracıyı durdurabilirsiniz. Daha fazla bilgi için [“MFT aracısını Windows hizmeti olarak başlatma”](#) sayfa 227 başlıklı konuya bakın.

İlgili görevler

[“MFT aracısını başlatma”](#) sayfa 226

Dosya aktarımı için bir Managed File Transfer aracısını kullanmadan önce aracıyı başlatmanız gerekir.

İlgili başvurular

[MFT aracı durumu değerleri](#)

[fteStopAracı](#)

[“z/OS üzerinde bir MFT aracısının durdurulması”](#) sayfa 232

JCL ' den başlatılan bir görev olarak z/OS üzerinde Managed File Transfer Agent çalıştırıyorsanız, aracı **fteStopAgent** komutuna ek olarak z/OS işletmen komutlarını **MODIFY** ve **STOP** kabul eder.

z/OS üzerinde bir MFT aracısının durdurulması

JCL ' den başlatılan bir görev olarak z/OS üzerinde Managed File Transfer Agent çalıştırıyorsanız, aracı **fteStopAgent** komutuna ek olarak z/OS işletmen komutlarını **MODIFY** ve **STOP** kabul eder.

Başlatılan bir görev, belirli bir kullanıcı kimliği altında çalıştığından ve oturum kapamış olan kullanıcılardan etkilenmediğinden kullanılır.

Not: Başlatılan görevler genellikle oturum açma ayrıcalıklarına sahip olmayabilecek bir yönetici kullanıcı altında çalıştırılır ve bu nedenle aracının altında çalıştığı kullanıcı olarak z/OS sisteminde oturum açamaz. **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** komutları ve **-d** parametresi belirtilen **fteShowAgentDetails** komutu bu aracı için verilemez.

IBM MQ 9.0.2 ve IBM MQ 9.0.0 Fix Pack 1olanağından, z/OS üzerinde Managed File Transfer araçlarıyla **adminGroup** aracı özelliğini kullanabilirsiniz. Bir güvenlik yöneticisi grubu (örneğin, MFTADMIN)

tanımlayabilir ve başlatılan görev kullanıcı kimliği ve denetimci TSO kimliklerini bu gruba ekleyebilirsiniz. Aracı özellikleri dosyasını düzenleyin ve **adminGroup** özelliğini bu güvenlik yöneticisi grubunun adı olacak şekilde ayarlayın.

```
adminGroup=MFTADMIN
```

Daha sonra bu grubun üyeleri, başlatılan bir görev olarak çalışan aracı için **fteStartAgent**, **fteStopAgent** ve **fteSetAgentTraceLevel** komutlarını ve **fteShowAgentDetails** komutunu **-d** parametresiyle birlikte yürütebilirler.

Daha fazla bilgi için [MFT agent.properties](#) dosyası içindeki **adminGroup** özelliğine bakın.

z/OS MODIFY komutu (F) kullanılarak denetlenen aracı kapatma

MODIFY komutu, **fteStopAgent** komutuna alternatif olarak bir aracıyı denetimli bir şekilde durdurmanızı sağlar. Aracı, devam etmekte olan aktarımları tamamlar, ancak aracı yeni aktarımları başlatmaz.

Örneğin:

```
F job_name,APPL=STOP
```

Burada *iş_adi* , aracı işleminin altında çalıştığı işidir.

z/OS STOP (P) komutunu kullanarak hemen aracıyı kapatma

STOP komutu, **-i** parametresiyle **fteStopAgent** komutunu kullanarak hemen durdurmaya eşdeğerdır. Aracı şu anda bir dosya aktarıyor olsa da aracı hemen durdurulur.

Örneğin:

```
P job_name
```

Burada *iş_adi* , aracı işleminin altında çalıştığı işidir.

İlgili başvurular

“z/OS üzerinde bir MFT aracısının başlatılması” sayfa 230

z/OS işletim sistemlerinde, **fteStartAgent** komutunu bir z/OS UNIX System Services oturumundan çalıştırmanın yanı sıra, etkileşimli bir oturum gerekmeden JCL ' den başlatılan bir görev olarak aracı başlatabilirsiniz.

[MFT agent.properties](#) dosyası

Yeni dosya aktarımı başlatılması

IBM MQ Explorer ' den ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir gruptaki tek bir dosyayı ya da birden çok dosyayı aktarmayı seçebilirsiniz.

Bu görev hakkında

Komut satırından yeni bir dosya aktarımı başlatmak için bkz. [fteCreateTransfer command](#).

IBM MQ Explorer' ta **Yeni Yönetilen Dosya Aktarımı Yarat** sihirbazını kullanarak yeni bir dosya aktarımı başlatmak için aşağıdaki adımları izleyin:

Yordam

1. Navigator görünümünde **Yönetilen Dosya Aktarımı** seçeneğini tıklatın. İçerik görünümünde **Yönetilen Dosya Aktarma Merkezi** görüntülenir.
2. Tüm eşgüdüm kuyruğu yöneticileriniz Navigator görünümünde görüntülenir. Aktarma için kullanmak istediğiniz aracının kayıtlı olduğu koordinasyon kuyruğu yöneticisinin adını açın. Aktarma için kullanmak istediğiniz eşgüdüm kuyruğu yöneticisinden başka bir eşgüdüm kuyruğu yöneticisine

bağlıysanız, Navigator görünümünde o eşgüdüm kuyruğu yöneticisi adını farenin sağ düğmesiyle tıklatın ve **Bağlantıyı Kes** ögesini seçin. Daha sonra kullanmak istediğiniz koordinasyon kuyruk yöneticisinin adını sağ tıklatın ve **Bağlan'**ı tıklatın.

3. Aşağıdaki yöntemlerden birini kullanarak **Yeni Yönetilen Dosya Aktarımı Yarat** sihirbazını başlatın:
 - a) Navigator görünümünde şu düğümlerden herhangi birinin adını sağ tıklatın: İlgili eşgüdüm kuyruğu yöneticisi, **Aktarma Şablonları**, **Aktarma Günlüğü** ya da **Bekleyen Aktarmalar**. Daha sonra sihirbazı başlatmak için **Yeni Aktarma** düğmesini tıklatın.
 - b) **Dosya > Yeni > Diğer > Yönetilen Dosya Aktarma Sihirbazları > Yeni Aktarma Sihirbazı** ögesini tıklatın.
4. Sihirbaz panolarındaki yönergeleri izleyin. Ayrıca, her pano için bağlama duyarlı yardım da sağlanır. Windows üzerindeki bağlama duyarlı yardıma erişmek için F1 tuşuna basın. Linux işletim sistemlerinde Ctrl+F1 ya da Shift+F1 tuşuna basın.

İlgili kavramlar

[“Aktarma tanımlama dosyalarının kullanılması” sayfa 234](#)

Dosya aktarımı yaratmak için kullanılacak bir aktarma tanımlama dosyası belirleyebilirsiniz. Aktarma tanımlama dosyası, aktarma yaratmak için gereken bilgilerin bazılarını ya da tümünü tanımlayan bir XML dosyasıdır.

İlgili görevler

[“Zamanlanmış dosya aktarımı yaratılması” sayfa 237](#)

IBM MQ Explorer' den ya da komut satırından yeni bir dosya aktarımı zamanlayabilirsiniz. Zamanlanmış aktarma, bir gruptaki tek dosyaları ya da birden çok dosyayı içerebilir. Zamanlanmış bir dosya aktarma işlemini bir kez gerçekleştirebilir ya da aktarma işlemini birden çok kez yineleyebilirsiniz.

[“Dosya aktarımını tetikleme” sayfa 239](#)

Bir dosya aktarımında, aktarma işleminin gerçekleşebilmesi için önce doğru olması gereken bazı tetikleyici koşulları ayarlayabilirsiniz. Tetikleme koşulları doğru değilse, dosya aktarımı gerçekleşmez ve aktarma işleminin gerçekleşmediği gerçeğini kaydetmek için isteğe bağlı olarak bir günlük iletisi gönderilir. Daha sonra, dosya aktarma isteği atılır. Örneğin, yalnızca kaynak aracısının bulunduğu sistemde adı belirlenmiş bir dosya belirli bir boyutun üzeriyse ya da kaynak aracısının bulunduğu sistemde belirli bir adlandırılmış dosya varsa, gerçekleştirilecek bir dosya aktarımı ayarlayabilirsiniz. Tetiklenen dosya aktarımını IBM MQ Explorer içinden ya da komut satırından ayarlayabilirsiniz.

[“Durdurulmuş aktarımların kurtarılması için zamanaşımı ayarlanması” sayfa 341](#)

Bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan, durdurulmuş dosya aktarımları için bir aktarım kurtarma zamanaşımı ayarlayabilirsiniz. Tek bir aktarma için aktarma kurtarma zamanaşımı da ayarlayabilirsiniz. Belirli bir süreyi saniye cinsinden ayarlarsanız, kaynak aracı durmuş bir dosya aktarımını kurtarmaya çalışır ve aracı zamanaşımına ulaştığında aktarma başarısız olur.

İlgili başvurular

fteCreateTransfer: yeni bir dosya aktarımı başlatır

[Dosya aktarma isteği iletme biçimi](#)

[Dosyaların aktarılmasına ilişkin yönergeler](#)

Aktarma tanımlama dosyalarının kullanılması

Dosya aktarımı yaratmak için kullanılacak bir aktarma tanımlama dosyası belirleyebilirsiniz. Aktarma tanımlama dosyası, aktarma yaratmak için gereken bilgilerin bazılarını ya da tümünü tanımlayan bir XML dosyasıdır.

Aktarma tanımlama dosyaları, tek bir aktarma işleminde birden çok kaynak dosya ve birden çok hedef dosya belirlemek istediğinizde yararlı olur. Karmaşık bir dosya aktarımını sunmak için bir aktarma tanımlama dosyası kullanabilirsiniz. Aktarma tanımlaması dosyasını yeniden kullanabilir ve paylaşabilirsiniz.

Bir aktarma tanımlaması dosyası için iki biçim kullanabilirsiniz; bu biçimler biraz değişse de, her ikisi de FileTransfer.xsd şemasına uygun olur. Bu şemayı, Managed File Transfer kuruluşunun samples\schema dizininde bulabilirsiniz.

Aşağıdaki iki aktarma tanımlaması dosyası biçimi desteklenir:

- Bir aktarıma ilişkin kaynak ve hedef dosyaların tanımlaması. Bu tanımlama, kök olarak bir **transferSpecifications** ögesini kullanır.
- Kaynak ve hedef dosyalar ile kaynak ve hedef araçlar da içinde olmak üzere, tüm aktarımın tanımlaması. Bu tanımlama, kök olarak bir **request** ögesini kullanır.
 - Bu biçime sahip dosyalar, **-gt** parametresi kullanılarak **fteCreateTransfer** komutundan oluşturulabilir.

Aşağıdaki örnek, bir aktarma için yalnızca kaynak ve hedef dosyaları belirleyen bir aktarma tanımlama dosyası biçimini göstermektedir:

```
<?xml version="1.0" encoding="UTF-8"?>
<transferSpecifications xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <item checksumMethod="MD5" mode="text">
    <source recursive="false" disposition="leave">
      <file>textTransferTest.txt</file>
    </source>
    <destination type="directory" exist="overwrite">
      <file>c:\targetfiles</file>
    </destination>
  </item>
</transferSpecifications>
```

Aktarma tanımlaması dosyasının bu biçimini sunmak için, komut satırında kaynak ve hedef araçları belirtmeniz gerekir:

```
fteCreateTransfer -sa AGENT1 -sm agent1qm -da AGENT2 -dm agent2qm -td
c:\definitions\example1.xml
```

Aşağıdaki örnek, bir aktarma için gerekli tüm bilgileri belirten bir aktarma tanımlama dosyası biçimidir:

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="3.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName>
      <userID>fteuser</userID>
    </originator>
    <sourceAgent agent="AGENT1" QMgr="agent1qm"/>
    <destinationAgent agent="AGENT2" QMgr="agent2qm"/>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:\sourcefiles\*.jpg</file>
        </source>
        <destination type="directory" exist="error">
          <file>/targetfiles/images</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

fteCreateTransfer komutunda **-gt** parametresini kullanarak bu biçime sahip bir dosya oluşturabilirsiniz. Bu biçimle bir aktarma tanımlaması dosyası sunduğunuzda, komut satırında başka bir şey belirtmenize gerek yoktur:

```
fteCreateTransfer -td c:\definitions\example2.xml
```

Aktarım tanımlaması dosyasına ek olarak olağan parametreleri aktararak, komut satırına ilişkin kaynak ve hedef aracı bilgilerini geçersiz kılabilirsiniz. Örneğin:

```
fteCreateTransfer -da AGENT9 -dm agent9qm -td c:\definitions\example2.xml
```

Bu örnek, aktarma tanımlaması dosyasında tanımlanan hedef aracıyı **AGENT9** ve aktarma tanımlaması dosyasında **agent9qm** olarak tanımlanan hedef kuyruk yöneticisiyle geçersiz kılmak için komut satırı seçeneklerini kullanır.

Açıklanan biçimlerin her ikisi de bir ya da daha fazla < item> ögesi içerebilir. < item> ögesiyle ilgili daha fazla bilgi için bkz. [Dosya aktarma isteği iletisi biçimi](#). Bu aktarma öğelerinin her biri, aktarmanın davranışını denetlemek için ek özniteliklerle birlikte bir kaynak ve hedef dosya çifti tanımlar. Örneğin, aşağıdaki davranışı belirtebilirsiniz:

- Aktarımın sağlama toplamı kullanıp kullanmadığını belirler
- Aktarmanın metin mi, ikili mi olduğu
- Aktarım tamamlandıktan sonra kaynak dosyanın silinip silinmeyeceğini belirler
- Dosya varsa, hedef dosyanın üzerine yazılıp yazılmayacağını belirler

Aktarma tanımı dosyalarını kullanmanın bir yararı, komut satırından kullanılmayan ek seçenekleri belirleyebilmektir. Örneğin, iletiden dosyaya aktarımlar gerçekleştirirken, bir aktarma tanımlaması dosyası kullanarak groupId özniteliğini belirtebilirsiniz. Bu öznitelik, kuyruktan okunan iletilerin IBM MQ grup tanıtıcısını belirtir. Aktarma tanımlama dosyalarının başka bir yararı da, her dosya çifti için farklı seçenekler belirleyebilmektir. Örneğin, bir sağlama toplamının kullanılıp kullanılmayacağını ya da dosyanın dosya temelinde metin kipinde mi, yoksa ikili kipte mi aktarılacağını belirleyebilirsiniz. Komut satırını kullanırsanız, aktarımda yer alan her dosya için aynı seçenekler geçerli olur.

Örneğin:

```
<item checksumMethod="none" mode="binary">
  <source disposition="leave">
    <file>c:\sourcefiles\source1.doc</file>
  </source>
  <destination type="file" exist="error">
    <file>c:\destinationfiles\destination1.doc</file>
  </destination>
</item>

<item checksumMethod="MD5" mode="text">
  <source disposition="delete">
    <file>c:\sourcefiles\source2.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file encoding="UTF8" EOL="CRLF">c:\destinationfiles\destination2.txt</file>
  </destination>
</item>

<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>c:\originfiles\source3.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file>c:\targetfiles\destination3.txt</file>
  </destination>
</item>
```

z/OS Dağıtılmış bir sistemden z/OS sistemine dosya aktarmak için öğeleri kullanabilirsiniz:

```
z/OS
<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>textTransferTest.txt</file>
  </source>
  <destination type="dataset" exist="overwrite">
    <file encoding="IBM-1047">//TEXT.TRANS.TEST</file>
  </destination>
</item>
```

z/OS Bu örnek, textTransferTest.txt dosyasını kaynak araçından metin kipinde hedef araçtaki //TEXT.TRANS.TEST veri kümesine aktarır. Bu aktarma, kaynak verileri kaynak aracısının varsayılan kodlamasından (kaynak kodlama özniteliği belirtilmedi) kod sayfasına dönüştürür: IBM-1047.

Zamanlanmış dosya aktarımı yaratılması

IBM MQ Explorer' den ya da komut satırından yeni bir dosya aktarımı zamanlayabilirsiniz. Zamanlanmış aktarma, bir gruptaki tek dosyaları ya da birden çok dosyayı içerebilir. Zamanlanmış bir dosya aktarma işlemini bir kez gerçekleştirebilir ya da aktarma işlemini birden çok kez yineleyebilirsiniz.

Bu görev hakkında

Bir dosya aktarma zamanlamasını bir kez gerçekleştirecek ya da aşağıdaki aralıklarla gerçekleştirecek şekilde ayarlayabilirsiniz:

- Her dakika
- Saatlik
- Günlük
- Haftada bir
- Aylık
- Yıllık

Daha sonra, aşağıdaki noktalarda duracak oluşumları belirtebilirsiniz:

- Tanımlı bir saat ve tarihte
- Tanımlı bir geçiş sayısından sonra

Diğer bir seçenek olarak, yinelemeler sonsuza kadar devam eder.

Zamanlanmış bir aktarma her gün aynı saatte çalışıyorsa, zamanlamanın saat değiştiğinde gerçekleştireceği zamanı ayarlamak için aracı özellikleri dosyasındaki **adjustScheduleTimeForDaylightSaving** özneliğini kullanın. Daha fazla bilgi için bkz. [MFT agent.properties](#) dosyası .

Komut satırını kullanarak yeni bir zamanlanmış dosya aktarımı oluşturmak için, [fteCreateTransfer](#) komutu için zamanlama parametrelerini (**-tb**, **-ss**, **-oi**, **-of**, **-ocve** **-es**) kullanın.

IBM MQ Explorer'inde **Yeni Yönetilen Dosya Aktarımı Yarat** sihirbazını kullanarak yeni bir zamanlanmış dosya aktarımı yaratmak için aşağıdaki adımları izleyin:

Yordam

1. Navigator görünümünde **Yönetilen Dosya Aktarımı** seçeneğini tıklatın. İçerik görünümünde **Yönetilen Dosya Aktarma Merkezi** görüntülenir.
2. Tüm eşgüdüm kuyruğu yöneticileriniz Navigator görünümünde görüntülenir. Aktarma için kullanmak istediğiniz aracının kayıtlı olduğu koordinasyon kuyruğu yöneticisinin adını açın. Aktarma için kullanmak istediğiniz eşgüdüm kuyruğu yöneticisinden başka bir eşgüdüm kuyruğu yöneticisine bağlarsanız, Navigator görünümünde o eşgüdüm kuyruğu yöneticisi adını farenin sağ düğmesiyle tıklatın ve **Bağlantıyı Kes** ögesini seçin. Daha sonra kullanmak istediğiniz koordinasyon kuyruk yöneticisinin adını sağ tıklatın ve **Bağlan** 'ı tıklatın.
3. Aşağıdaki yöntemlerden birini kullanarak **Yeni Yönetilen Dosya Aktarımı Yarat** sihirbazını başlatın:
 - a) Navigator görünümünde şu düğümlerden herhangi birinin adını sağ tıklatın: İlgili eşgüdüm kuyruğu yöneticisi, **Aktarma Şablonları**, **Aktarma Günlüğü** ya da **Bekleyen Aktarmalar**. Daha sonra sihirbazı başlatmak için **Yeni Aktarma** düğmesini tıklatın.
 - b) **Dosya > Yeni > Diğer > Yönetilen Dosya Aktarma Sihirbazları > Yeni Aktarma Sihirbazı** ögesini tıklatın.
4. Sihirbaz panolarındaki yönergeleri izleyin. **Zaman çizelgesi aktarımını etkinleştir** onay kutusunu seçtiğinizden emin olun ve **Zamanlama** sekmesine zamanlama ayrıntılarınızı girin. Zamanlanmış dosya aktarımları, aktarımı etkileyebilecek bir sorun yoksa, zamanlama başlangıç saatinden itibaren bir dakika içinde başlar. Örneğin, ağınızla ya da aracınızla ilgili, zamanlanmış aktarmanın başlatılmasını engelleyen sorunlar olabilir. Her pano için bağlama duyarlı yardım sağlanır. Windows üzerindeki bağlama duyarlı yardıma erişmek için F1 tuşuna basın. Linux işletim sistemlerinde Ctrl+F1 ya da Shift+F1 tuşuna basın.

Sonuçlar

Zamanlanmış dosya aktarımlarıyla ilgili iletiler hakkında bilgi için bkz. [Zamanlanmış dosya aktarma günlüğü ileti biçimleri](#).

Bekleyen dosya aktarımlarıyla çalışma

IBM MQ Explorer' den beklemede olan zamanlanmış dosya aktarımlarını görüntüleyebilirsiniz. **Bekleyen Aktarımlar** penceresi, bağlı olduğunuz koordinasyon kuyruk yöneticisine kayıtlı bekleyen tüm aktarımları görüntüler.


Bu görev hakkında

Henüz başlatılmamış bir zamanlanmış dosya aktarımının durumunu görüntülemek için aşağıdaki adımları izleyin:

Yordam

1. Navigator görünümünde **Yönetilen Dosya Aktarımı** nesnesini açın. İçerik görünümünde **Yönetilen Dosya Aktarma Merkezi** görüntülenir.
2. Tüm eşgüdüm kuyruğu yöneticileriniz Navigator görünümünde görüntülenir. Zamanlanmış aktarma için kullandığınız eşgüdüm kuyruğu yöneticisinin adını açın. Bağlı olduğunuz koordinasyon kuyruk yöneticisini değiştirmek istiyorsanız, Navigator görünümünde kullanmak istediğiniz koordinasyon kuyruk yöneticisinin adını sağ tıklayın ve **Bağlan'** ı tıklayın.
3. **Bekleyen Aktarımlar** ögesini tıklayın. **Bekleyen Aktarımlar** penceresi İçerik görünümünde görüntülenir.
4. **Bekleyen Aktarımlar** penceresi, zamanlanmış dosya aktarımlarınıza ilişkin aşağıdaki ayrıntıları görüntüler:
 - a) **Ad** Zamanlanmış dosya aktarımının numarası. Bu numara otomatik olarak atanır.
 - b) **Kaynak** Kaynak aracının adı.
 - c) **Kaynak kütük** Anasistemde aktarılacak kütüğün adı.
 - d) **Hedef** Hedef aracının adı.
 - e) **Hedef Dosya** Hedef sisteme aktarıldıktan sonra dosyanın adı.
 - f) **Zamanlanmış Başlangıç (seçilen saat dilimi)** Dosya aktarımının, yöneticinin seçilen saat diliminde başlatılmak üzere zamanlandığı saat ve tarih. Görüntülenen saat dilimini değiştirmek için **Pencere > Tercihler > IBM MQ Explorer > Yönetilen Dosya Aktarımı** seçeneğini tıklayın ve **Saat dilimi:** listesinden alternatif bir saat dilimi seçin. **Tamam'** ı tıklayın.
 - g) **Yineleme Sıklığı** Zamanlanmış aktarımı yinelemeyi seçtiyseniz, aktarmayı yinelemek istediğiniz belirtilen aralık bir sayı olarak gösterilir.
 - h) **Yineleme Tipi** Zamanlanmış aktarımı yinelemeyi seçtiyseniz, dosya aktarımı için belirlediğiniz yineleme aralığı tipi. Tip şu değerlerden biri olabilir: dakika, saat, gün, hafta, aya da yıl.
 - i) **Yinele:** Zamanlanmış aktarımı yinelemeyi seçtiyseniz, yinelenen dosya aktarımının ne zaman durmasını istediğinize ilişkin ayrıntılar. Örneğin, belirli bir tarih ve saat ya da belirli bir geçiş sayısından sonra.

Sonuçlar

Bekleyen Aktarımlar penceresinde görüntülenenleri yenilemek için, İçerik görünümü araç çubuğundaki Yenile düğmesini  tıklayın.

Bekleyen bir dosya aktarımını iptal etmek için, ilgili aktarımı farenin sağ düğmesiyle tıklayın ve **İptal'** i seçin. Aktarma işleminin iptal edilmesi, dosya aktarma isteğini tamamen atar.

Dosya aktarımını tetikleme

Bir dosya aktarımında, aktarma işleminin gerçekleşebilmesi için önce doğru olması gereken bazı tetikleyici koşulları ayarlayabilirsiniz. Tetikleme koşulları doğru değilse, dosya aktarımı gerçekleşmez ve aktarma işleminin gerçekleşmediği gerçeğini kaydetmek için isteğe bağlı olarak bir günlük iletisi gönderilir. Daha sonra, dosya aktarma isteği atılır. Örneğin, yalnızca kaynak aracısının bulunduğu sistemde adı belirlenmiş bir dosya belirli bir boyutun üzeriyse ya da kaynak aracısının bulunduğu sistemde belirli bir adlandırılmış dosya varsa, gerçekleştirilecek bir dosya aktarımı ayarlayabilirsiniz. Tetiklenen dosya aktarımını IBM MQ Explorer içinden ya da komut satırından ayarlayabilirsiniz.

Bu görev hakkında

Tetikleme koşulunun karşılanması için bir kaynağı sürekli olarak izleyebilirsiniz. Kaynak izlemeyle ilgili daha fazla bilgi için bkz. "[MFT kaynaklarının izlenmesi](#)" sayfa 244.

Ayarlayabileceğiniz üç farklı tetikleme koşulu vardır. Koşullar aşağıdaki gibidir:

- Kaynak aracıyla aynı sistemde belirli bir dosya varsa
- Belirli bir dosya kaynak aracıyla aynı sistemde yoksa
- Belirli bir dosya, kaynak aracının bulunduğu sistemde belirli bir boyutun üzeriyse (büyüklük bayt, KB, MB ya da GB olarak ifade edilebilir). Bu ölçü birimleri 2¹⁰ kuralını kullanır; örneğin, 1 KB 1024 bayta ve 1 MB 1024 KB ' ye eşittir.

Önceki listedeki tetikleyici tipleri iki şekilde birleştirilebilir:

- Tek bir koşul için, kaynak aracının bulunduğu sistemde birden çok dosya belirtebilirsiniz. Belirtilen dosyalardan herhangi biri koşula (Boole işleci OR) uyuyorsa, bu işlem aktarımı tetikler.
- Birden çok koşul belirtebilirsiniz. Bu, yalnızca tüm koşullar karşılandıysa (Boole işleci AND) aktarımı tetikler.

Tetiklenen bir aktarımı zamanlanmış bir aktarımla da birleştirebilirsiniz. Ek bilgi için [Zamanlanmış Dosya Aktarımı Yaratılması](#) başlıklı konuya bakın. Bu durumda tetikleme koşulları, zamanlamanın başlayacağı sırada ya da zamanlamanın her başlatılacağı zaman yinelenen bir zamanlama için değerlendirilir.

Tetiklenen aktarımlar, protokol köprüsü araçlarında desteklenmez.

Komut satırını kullanarak tetiklenmiş bir kütük aktarımı yaratmak için, [fteCreateTransfer](#) komutundaki **-tr** değiştirgesini kullanın.

IBM MQ Explorer içinde **Yeni Yönetilen Dosya Aktarımı Yarat** sihirbazını kullanarak zamanlanmış bir dosya aktarımı yaratmak için aşağıdaki adımları izleyin:

Yordam

1. Navigator görünümünde **Yönetilen Dosya Aktarımı** seçeneğini tıklatın. İçerik görünümünde **Yönetilen Dosya Aktarma Merkezi** görüntülenir.
2. Tüm eşgüdüm kuyruğu yöneticileriniz Navigator görünümünde görüntülenir. Zamanlanmış aktarma için kullandığınız eşgüdüm kuyruğu yöneticisinin adını açın. Bağlı olduğunuz koordinasyon kuyruk yöneticisini değiştirmek istiyorsanız, Navigator görünümünde kullanmak istediğiniz koordinasyon kuyruk yöneticisinin adını sağ tıklatın ve **Bağlan** ' ı tıklatın.
3. Aşağıdaki yöntemlerden birini kullanarak **Yeni Yönetilen Dosya Aktarımı Yarat** sihirbazını başlatın:
 - a) Navigator görünümünde şu düğümlerden herhangi birinin adını sağ tıklatın: İlgili eşgüdüm kuyruğu yöneticisi, **Aktarma Şablonları**, **Aktarma Günlüğü** ya da **Bekleyen Aktarmalar**. Daha sonra sihirbazı açmak için **Yeni Aktarma** ' yı tıklatın.
 - b) **Dosya > Yeni > Diğer > Yönetilen Dosya Aktarma Sihirbazları > Yeni Aktarma Sihirbazı** ögesini tıklatın.
4. Sihirbaz panolarındaki yönergeleri izleyin. **Tetikleyiciler** etiketinde **Tetiklenmiş aktarımı etkinleştir** onay kutusunu seçtiğinizden emin olun ve tetikleme ayarlamak için o etiketteki alanları doldurun. Her


pano için bağlama duyarlı yardım sağlanır. Windows üzerindeki bağlama duyarlı yardıma erişmek için F1 tuşuna basın. Linux işletim sistemlerinde **Ctrl+F1** ya da **Shift+F1** tuşlarına basın.

Devam eden dosya aktarımlarını izleme

Devam eden bir dosya aktarımını, IBM MQ Explorer'deki **Yönetilen Dosya Aktarımı-Yürürlükteki Aktarma İlerlemesi** sekmesini kullanarak izleyebilirsiniz. Bu dosya aktarımı IBM MQ Explorer ya da komut satırından başlatılabilir. Sekme, zamanlanan aktarımların başladığı noktada zamanlanan aktarımların ilerleme durumunu da görüntüler.

Bu görev hakkında

Uzak bir sistemdeki bir eşgüdüm kuyruk yöneticisiyle ilişkili aktarımları izlemek için IBM MQ Explorer komutunu kullanmak istiyorsanız, "[IBM MQ Explorer ' in bir uzak eşgüdüm kuyruk yöneticisini izleyecek şekilde yapılandırılması](#)" sayfa 241 başlıklı konudaki yönergeleri izleyin.

IBM MQ Explorer durdurulup yeniden başlatıldıktan sonra önceki dosya aktarma bilgileri saklanmaz. Yeniden başlatma sırasında, geçmiş aktarımlarla ilgili bilgiler **Yürürlükteki Aktarma İlerleme Durumu** sekmesinden temizlenir. IBM MQ Explorer açıkken, **Tamamlanan aktarımları kaldır**  görüntüsünü kullanarak tamamlanan aktarımları temizleyebilirsiniz.


Yordam


IBM MQ Explorer komutunu ya da komut satırını kullanarak yeni bir dosya aktarımı başlattıktan sonra, **Yürürlükteki Aktarma İlerleme Durumu** etiketinde aktarma işleminin ilerleyişini izleyebilirsiniz. Devam eden her aktarma için aşağıdaki bilgiler görüntülenir:

- Kaynak.** Dosyayı kaynak sistemden aktarmak için kullanılan aracının adı.
- Hedef.** Hedef sistemde dosyayı almak için kullanılan aracının adı.
- Yürürlükteki kütük.** Aktarılmakta olan dosyanın adı. Önceden aktarılan tek tek dosyanın bölümü B, KiB, MiB içinde görüntülenir. GiB ya da TiB ve parantez içinde dosyanın toplam boyutu. Görüntülenen ölçü birimi, dosyanın boyutuna bağlıdır.
B, saniye başına bayttır. KiB/s, saniye başına kibibayttır; burada 1 kibibayt 1024 bayta eşittir. MiB/s, saniye başına mebibayttır; burada 1 mebibayt, 1 048 576 bayta eşittir. GiB/s, saniye başına gibibayttır; burada 1 gibibayt, 1 073 741 824 bayta eşittir. TiB/s, saniye başına tebibayttır; burada 1 tebibayt, 1 099 511 627 776 bayta eşittir.
- Dosya numarası.** Birden çok dosya aktarıyorsanız, bu sayı, aktarma işleminin toplam dosya grubunun ne kadar uzağında olduğunu gösterir.
- İlerleme.** İlerleme çubuğu, geçerli dosya aktarımının ne kadar tamamlandığını yüzde olarak gösterir.
- Oran.** Dosyanın aktarılma hızı KiB/s (kibibayt/saniye; burada 1 kibibayt 1024 bayta eşittir.)
- Başlatıldı (seçilen saat dilimi).** Dosya aktarımının başlatıldığı saat, yöneticinin seçilen saat diliminde gösterilir. Görüntülenen saat dilimini değiştirmek için **Pencere > Tercihler > IBM MQ Explorer > Yönetilen Dosya Aktarımı** seçeneğini tıklatın ve **Saat dilimi:** listesinden alternatif bir saat dilimi seçin. **Tamam'**ı tıklatın.

Aktarma, dosya aktarılırken bir kurtarma durumuna girerse, başlatma zamanı, dosya aktarımının sürdürülme zamanını yansıtacak şekilde güncellenir.

Sonuçlar

Bu sekme düzenli olarak bilgilerini otomatik olarak yeniler, ancak **Yürürlükteki Aktarma İlerlemesi** sekmesinde görüntülenenin yenilenmiş bir görünümünü zorlamak için İçerik görünümü araç çubuğundaki **Yenile**  simgesini tıklatın.

Dosya aktarımlarını **Yürürlükteki Aktarma İlerleme Durumu** sekmesinden silmek için, İçerik görünümü araç çubuğundaki **Tamamlanan aktarımları kaldır**  simgesini tıklatın. Bu düğmenin tıklanması, dosya

aktarma ayrıntılarını yalnızca etiketten kaldırır; yürürlükteki ya da zamanlanmış bir aktarımı durdurmaz ya da iptal etmez.

Sekmeyi kapattıktan sonra **Yürürlükteki Aktarma İlerleme Durumu** sekmesine dönmek istiyorsanız, **Pencere > Görünümü Göster > Diğer > Diğer > Yönetilen Dosya Aktarımı-Yürürlükteki Aktarma İlerleme Durumu** seçeneğini tıklayarak sekmeyi görüntüleyebilirsiniz. **Tamam'**ı tıklatın.

Sonraki adım

Buna ek olarak, özel dosya aktarımı izlemesi için uygulamalar geliştirilmesi de mümkündür. Bu, uygun Managed File Transfer yönetim konusuna (programsal ya da yönetimsel olarak) abonelik oluşturularak gerçekleştirilebilir ve izleme uygulaması, konuyla ilgili Managed File Transfer dosya aktarma etkinliği yayınlarını alabilir. Abonelik konusu ve yayın iletisi biçimiyle ilgili daha fazla bilgi için [Dosya aktarma ilerleme durumu iletisi örnekleri](#) başlıklı konuya bakın.

İlgili görevler

“IBM MQ Explorer ' in bir uzak eşgüdüm kuyruk yöneticisini izleyecek şekilde yapılandırılması” sayfa 241
Uzak bir sistemde çalışan bir eşgüdüm kuyruğu yöneticisiyle ilişkili dosya aktarımlarını izlemek için IBM MQ Explorer seçeneğini kullanın. IBM MQ Explorer programını çalıştırabilecek bir sisteme gereksinim duyarsınız. Uzak koordinasyon kuyruk yöneticisine bağlanabilmek için IBM MQ Explorer bileşeninin kurulması gerekir.

“Aktarma Günlüğündeki Dosya Aktarımlarının Durumunun Görüntülenmesi” sayfa 242
IBM MQ Explorer içindeki **Aktarma Günlüğü** ' nü kullanarak dosya aktarımlarının ayrıntılarını görüntüleyebilirsiniz. Bunlar, komut satırından ya da IBM MQ Explorer' den başlatılan aktarımlar olabilir. **Aktarma Günlüğü** ' nde görüntülenenleri de uyarlayabilirsiniz.

IBM MQ Explorer ' in bir uzak eşgüdüm kuyruk yöneticisini izleyecek şekilde yapılandırılması

Uzak bir sistemde çalışan bir eşgüdüm kuyruğu yöneticisiyle ilişkili dosya aktarımlarını izlemek için IBM MQ Explorer seçeneğini kullanın. IBM MQ Explorer programını çalıştırabilecek bir sisteme gereksinim duyarsınız. Uzak koordinasyon kuyruk yöneticisine bağlanabilmek için IBM MQ Explorer bileşeninin kurulması gerekir.

Bu görev hakkında

Varsayımlar: Kuyruk yöneticisini uzak bağlantılara izin verecek şekilde yapılandırarak uzak koordinasyon kuyruk yöneticisine bağlanma yetkisi.

Bunun nasıl yapılandırılacağına ilişkin ek bilgi için [Kanal kimlik doğrulamasıyla istemci kipinde kuyruk yöneticisine bağlanma ve MFT' e özgü kaynaklar için yetkilerin yönetilmesi](#) başlıklı konuya bakın.

Windows ya da Linux işletim sisteminde çalışmayan bir sistemdeki araçlar arasındaki kuyruk yöneticilerini ve dosya aktarımlarını izlemek için, IBM MQ Explorer ' yi aşağıdaki adımları kullanarak uzak sisteme bağlanacak şekilde yapılandırın:

Yordam

1. Yerel IBM MQ Explorer' yi başlatın.
2. IBM MQ Explorer yüklendiğinde, **Managed File Transfer** (Yönetilen Dosya Aktarımı) klasörünü sağ tıklayın ve **New configuration** (Yeni konfigürasyon) seçeneğini belirleyin.
3. Sihirbazda ilerleyin, Koordinasyon ve Komutlar kuyruk yöneticisini seçin ve yapılandırma için bir ad tanımlayın.
4. Tanımlamayı tamamlamak için **Son** düğmesini tıklayın.
5. Tanımlama tamamlandığında, tanımlamayı sağ tıklayın ve **Bağlan** seçeneğini belirleyin.

Sonuçlar

Şimdi IBM MQ Explorer ' u başlatın ve koordinasyon kuyruk yöneticisiyle ilişkili Managed File Transfer ağına ilişkin aktarım etkinliğini izlemek için bunu kullanın.

İlgili görevler

[“Devam eden dosya aktarımlarını izleme” sayfa 240](#)

Devam eden bir dosya aktarımını, IBM MQ Explorer’indeki **Yönetilen Dosya Aktarımı-Yürürlükteki Aktarma İlerlemesi** sekmesini kullanarak izleyebilirsiniz. Bu dosya aktarımı IBM MQ Explorer ya da komut satırından başlatılabilir. Sekme, zamanlanan aktarımların başladığı noktada zamanlanan aktarımların ilerleme durumunu da görüntüler.


[“Aktarma Günlüğündeki Dosya Aktarımlarının Durumunun Görüntülenmesi” sayfa 242](#)

IBM MQ Explorer’indeki **Aktarma Günlüğü** ' nü kullanarak dosya aktarımlarının ayrıntılarını görüntüleyebilirsiniz. Bunlar, komut satırından ya da IBM MQ Explorer' den başlatılan aktarımlar olabilir. **Aktarma Günlüğü** ' nde görüntülenenleri de uyarlayabilirsiniz.

Aktarma Günlüğündeki Dosya Aktarımlarının Durumunun Görüntülenmesi

IBM MQ Explorer’indeki **Aktarma Günlüğü** ' nü kullanarak dosya aktarımlarının ayrıntılarını görüntüleyebilirsiniz. Bunlar, komut satırından ya da IBM MQ Explorer' den başlatılan aktarımlar olabilir. **Aktarma Günlüğü** ' nde görüntülenenleri de uyarlayabilirsiniz.



Yordam

1. Navigator görünümünde **Yönetilen Dosya Aktarımı** nesnesini açın ve aktarma günlüğünü görüntülemek istediğiniz eşgüdüm kuyruğu yöneticisinin adını açın.
2. Navigator görünümünde **Aktarma Günlüğü** ' nü tıklatın. İçerik görünümünde **Aktarma Günlüğü** görüntülenir.
3. **Aktarma Günlüğü** penceresi dosya aktarımlarınıza ilişkin aşağıdaki ayrıntıları görüntüler:
 - a) **Kaynak** Kaynak dosyanın bulunduğu sistemdeki aracının adı.
 - b) **Hedef** Dosyayı aktarmak istediğiniz sistemdeki aracının adı.
 - c) **Tamamlanma Durumu** Dosya aktarımının durumu. Durum şu değerlerden biri olabilir: "Başlatıldı", "Devam Ediyor", "Başarılı", "Kısmen Başarılı", "İptal Edildi" ya da "Başarısız Oldu".
 - d) **İye** Aktarma isteğini gönderen anasistemdeki kullanıcı kimliği.
 - e) **Başlatma (seçilen saat dilimi)** Dosya aktarma isteğinin Managed File Transfer aracısı tarafından kabul edildiği saat ve tarih, yöneticinin seçilen saat diliminde gösterilir. Görüntülenen saat dilimini değiştirmek için **Pencere > Tercihler > IBM MQ Explorer > Managed File Transfer** düğmesini tıklatın ve **Saat dilimi:** listesinden alternatif bir saat dilimi seçin. **Tamam**'ı tıklatın.
 - f) **Kaydedilen Durum (seçilen saat dilimi)** (Bu sütun varsayılan olarak görüntülenmez. Sütunu **Aktarma Günlüğü Sütunlarını Yapılandır**  penceresini kullanarak görüntülemeyi seçebilirsiniz.) Sistem yöneticisi tarafından seçilen saat diliminde, tamamlanma durumunun kaydedildiği saat ve tarih.
 - g) **İş Adı** Kullanıcı tarafından **fteCreateTransfer -jn** parametresini kullanarak ya da bir Ant komut dosyasında belirtilen tanıtıcı
 - h) **Aktarma Tanıtıcısı** Dosya aktarımına ilişkin benzersiz tanıtıcı.
 - i) **Connect: Direct Süreç Numarası, Süreç Adı, Birincil Düğüm, İkincil Düğüm, Kaynak Tipi ve Hedef Tipi** ile ilgili ayrıntılar listelenir.

Sonuçlar

Not: Aktarma günlüğünün iç biçimi APAR IC99545 için IBM MQ 8.0.0 Fix Pack 1 içinde değiştirildi. Sonuç olarak, bir IBM MQ Explorer V8.0.0.1 ya da sonraki bir sürüme yükselttilip V8.0.0.0 düzeyine geri yüklenirse, IBM MQ Explorer V8.0.0.1 sürümündeyken gerçekleşen aktarımlar için denetim XML ' i görüntülenmez. Bu aktarımlara ilişkin **Özellikler** penceresindeki XML panosu boş bir metin kutusu içerir.

Tamamlanan bir aktarıma ilişkin daha fazla ayrıntı görüntülemek için, artı işaretini (+) tıklatarak ilgilendiğiniz aktarımı genişletin. Daha sonra, bu aktarımda yer alan tüm kaynak ve hedef dosya adlarını görebilirsiniz. Ancak, aktarma işlemi devam ediyorsa ve çok sayıda dosyadan oluşuyorsa, yalnızca şimdiye kadar aktarılmış olan dosyaları görüntüleyebilirsiniz.

Aktarma Günlüğü'nde görüntüleneni yenilemek için, İçerik görünümü araç çubuğundaki **Yenile** düğmesini  tıklatın. IBM MQ Explorer'yi durdurup yeniden başlattıktan sonra, Aktarma Günlüğü 'nde yer alan dosya aktarma bilgileri günlükte kalır. Tamamlanan tüm dosya aktarımlarını günlükten silmek istiyorsanız, İçerik görünümü araç çubuğunda **Tamamlanan Aktarımları Kaldır**  seçeneğini tıklatın.

Tamamlanan bir dosya aktarımını günlükten silmek için, aktarımı sağ tıklatın ve **Sil**'i tıklatın. Bir aktarımı silerseniz, devam eden ya da zamanlanmış bir aktarımı durdurmaz ya da iptal etmez; yalnızca saklanan geçmiş verileri silersiniz.

Bir aktarımın benzersiz tanıtıcısını panoya kopyalamak için, o aktarımı sağ tıklatın ve **Kimliği Kopyala**'yı tıklatın.

Aktarma için meta veriler ve tam denetim XML 'i **Özellikler** işlemi altında beliren menüden kullanılabilir.

İlgili görevler

[“Devam eden dosya aktarımlarını izleme” sayfa 240](#)

Devam eden bir dosya aktarımını, IBM MQ Explorer'indeki **Yönetilen Dosya Aktarımı-Yürürlükteki Aktarma İlerlemesi** sekmesini kullanarak izleyebilirsiniz. Bu dosya aktarımı IBM MQ Explorer ya da komut satırından başlatılabilir. Sekme, zamanlanan aktarımların başladığı noktada zamanlanan aktarımların ilerleme durumunu da görüntüler.

[“Aktarma Günlüğünün Yapılandırılması” sayfa 243](#)

Hangi bilgilerin görüntülendiğini ve bilgilerin IBM MQ Explorer'indeki **Aktarma Günlüğü** 'nde nasıl görüntülendiğini yapılandırabilirsiniz.

[“Durdurulmuş aktarımların kurtarılması için zamanlamayı ayarlanması” sayfa 341](#)

Bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan, durdurulmuş dosya aktarımları için bir aktarım kurtarma zamanlamayı ayarlayabilirsiniz. Tek bir aktarma için aktarma kurtarma zamanlamayı da ayarlayabilirsiniz. Belirli bir süreyi saniye cinsinden ayarlarsanız, kaynak aracı durmuş bir dosya aktarımını kurtarmaya çalışır ve aracı zamanlamasına ulaştığında aktarma başarısız olur.

Aktarma Günlüğünün Yapılandırılması

Hangi bilgilerin görüntülendiğini ve bilgilerin IBM MQ Explorer'indeki **Aktarma Günlüğü** 'nde nasıl görüntülendiğini yapılandırabilirsiniz.

Bu görev hakkında


Aktarma Günlüğü'ndeki sütunların sırasını yeniden düzenlemek için, taşımak istediğiniz sütunun başlığını tıklatın ve sütunu yeni konumuna sürükleyin. Yeni sütun sırası yalnızca siz bir sonraki duruncaya ve IBM MQ Explorer'inden başlatılıncaya kadar korunur.

Aktarma Günlüğü'ndeki girdileri süzmek için **Görüntülenen günlük girdilerine süzgeç uygula** alanına bir dizgi girin. Tüm girişleri günlüğe geri yüklemek için, girdiğiniz dizgiyi alandan silin. Bu alanda geçerli bir Java düzenli ifadesi kullanabilirsiniz. Daha fazla bilgi için bkz. [MFT tarafından kullanılan düzenli ifadeler](#).

Aktarma günlüğünde hangi kolonların görüntüleneceğini uyarlamak için, **Aktarma Günlüğü Kolonları**

Yapılandır  simgesini kullanın. **Aktarma Günlüğü Kolonları Yapılandır** penceresini başlatmak ve kullanmak için aşağıdaki adımları kullanın.

Yordam

1. İçerik görünümünde **Aktarma Günlüğü** 'nü açtığınızdan emin olun. İçerik görünümü araç çubuğunda **Aktarma Günlüğü Sütunlarını Yapılandır**  simgesini tıklatın. **Aktarma Günlüğü Sütunlarını Yapılandır** penceresi açılır.

2. **Aktarma Günlüğü** görünümünüzü uyarlamak için, göstermek ya da gizlemek istediğiniz kolonlara ilişkin onay kutularını tek tek seçin ya da temizleyin. **Tümünü Seç**, ardından tüm onay kutularını seçmek için **Tamam** düğmesini tıklatabilir ya da tüm onay kutularını temizlemek için **Tümünün Seçimlerini Kaldır**, ardından **Tamam** düğmesini tıklatabilirsiniz.

İlgili görevler

“Devam eden dosya aktarımlarını izleme” sayfa 240

Devam eden bir dosya aktarımını, IBM MQ Explorer’indeki **Yönetilen Dosya Aktarımı-Yürürlükteki Aktarma İlerlemesi** sekmesini kullanarak izleyebilirsiniz. Bu dosya aktarımı IBM MQ Explorer ya da komut satırından başlatılabilir. Sekme, zamanlanan aktarımların başladığı noktada zamanlanan aktarımların ilerleme durumunu da görüntüler.

“Aktarma Günlüğündeki Dosya Aktarımlarının Durumunun Görüntülenmesi” sayfa 242

IBM MQ Explorer’indeki **Aktarma Günlüğü** 'nü kullanarak dosya aktarımlarının ayrıntılarını görüntüleyebilirsiniz. Bunlar, komut satırından ya da IBM MQ Explorer' den başlatılan aktarımlar olabilir. **Aktarma Günlüğü** 'nde görüntüleneni de uyarlayabilirsiniz.

MFT kaynaklarının izlenmesi

Managed File Transfer kaynaklarını izleyebilirsiniz; örneğin, bir kuyruk ya da izin. Bu kaynaktaki bir koşul karşılandığında, kaynak izleme programı dosya aktarma gibi bir görevi başlatır. IBM MQ Explorer’indeki Managed File Transfer eklentisindeki **fteCreateMonitor** komutunu ya da **İzleme Programları** görünümünü kullanarak bir kaynak izleme programı yaratabilirsiniz.

Bu görev hakkında

Managed File Transfer kaynak izlemesi aşağıdaki terminolojiyi kullanır:

Kaynak izleme programı

Kaynak izleme programı, kaynak içeriğinin değişip değişmediğini görmek için bir kaynağı (dizin ya da kuyruk gibi) önceden tanımlanmış bir aralıkta yoklayan işlemdir. Varsa, içindekiler bu monitöre ilişkin koşullar kümesiyle karşılaştırılır. Eşleşme varsa, bu izleme programına ilişkin görev başlatılır.

Kaynak

Kaynak izleme programının, tetikleme koşullarıyla karşılaştırılacak her yoklama aralığında inceleyeceği sistem kaynağı. İzlenen kaynak, kuyruklar, izinler ya da iç içe izin yapıları olabilir.

Koşul ve Tetikleme koşulu

Koşul, değerlendirilen bir ifadedir (genellikle izlenen kaynağın içeriğine göre). İfade doğru olarak değerlendirilirse, koşul genel tetikleme koşuluna katkıda bulunur.

Tetikleme koşulu, tüm koşullar karşılandığında karşılanan genel koşuldur. Tetikleyici koşulu yerine getirildiğinde görev devam edebilir.

Görev

Görev, tetikleme koşulu ya da koşul kümesi karşılandığında başlatılan işlemdir. Desteklenen görevler, dosya aktarımı ve komut çağrısıdır.

Tetikleme kütüğü

Tetikleyici dosyası, bir görevin (genellikle aktarma) başlayabileceğini belirtmek için izlenen bir dizine yerleştirilen bir dosyadır. Örneğin, işlenecek tüm dosyaların bilinen bir konuma geldiğini ve aktarılabilirliğini ya da başka bir şekilde işlem yapılabilirliğini gösterebilir. Tetikleyici dosyasının adı, değişken yerine koyma değeri kullanılarak aktarılacak kütükleri belirlemek için kullanılabilir. Daha fazla bilgi için bkz “MFT kaynak izleyicisi görevlerinin değişken yerine koyma değeri ile uyarlanması” sayfa 255.

Tetikleyici dosyası, hazır dosya ya da go dosyası olarak da bilinir. Ancak, bu belgede genellikle tetikleyici dosyası olarak adlandırılır.

Kaynak izleme, protokol köprüsü araçlarında ya da Connect:Direct köprü araçlarında desteklenmez.

İlgili kavramlar

Bir aracının aşırı yüklenmesini önlemek için MFT kaynak izleme programını yapılandırmaya ilişkin kılavuz

İlgili başvurular

fteCreateMonitor: MFT kaynak izleme programı yaratılması

fteListİzleme Programları: liste MFT kaynak izleme programları

fteDeleteMonitor: MFT kaynak izleyicisini silme

MFT izleyici isteği ileti biçimleri

MFT kaynak izleme kavramları

Managed File Transfer kaynak izleme özelliğine ilişkin temel kavramlara genel bakış.

Kaynak izleyicileri

Komut satırından yeni bir kaynak izleme programı yaratan ve başlatan **fteCreateMonitor** komutunu kullanarak bir kaynak izleme programı yaratırsınız. Kaynak izleyicisi bir Managed File Transfer aracısıyla ilişkilendirilir ve yalnızca aracı çalışırken etkindir. İzleme aracısı durduğunda, kaynak izleme programı da durur. Kaynak izleme programı yaratıldığında aracı zaten çalışıyorsa, kaynak izleme programı hemen başlar. İzleme aracısı, kaynak izleyici tarafından başlatılan görevin kaynak aracısı da olmalıdır.

Kaynak izleyicisi adları, araçları içinde benzersiz olmalıdır. Kaynak izleyicisi adı en az bir karakter uzunluğunda olmalı ve yıldız işareti (*), yüzde (%) ya da soru işareti (?) karakterleri içermemelidir. Bir kaynak izleyicisi adının sağlandığı durum yoksayılr ve kaynak izleyicisi adı büyük harfe dönüştürülür. Önceden var olan bir adla kaynak izleme programı yaratmayı denerseniz, istek yoksayılr ve bu girişim kaynak izleme günlüğü konusuna kaydedilir.

Not: Zamanlanmış aktarımları içeren bir görev tanımlamasıyla kaynak izleyici yaratamazsınız.

V 9.3.0 IBM MQ 9.3.0' den önce, bir kaynak izleme programını durdurmanın tek yolu, izleme işlemini çalıştıran aracıyı durdurmaktır. Bir kaynak izleme programını yeniden başlatmak için aracıyı tümüyle yeniden başlatmanız gerekir. IBM MQ 9.3.0' den bir aracıyı durdurmaya ya da yeniden başlatmaya gerek kalmadan kaynak izleyicilerini başlatabilir ve durdurabilirsiniz. Daha fazla bilgi için bkz [“Kaynak izleme programlarının başlatılması ve durdurulması”](#) sayfa 247.

Bir aracıda yaratılabilecek kaynak izleme programı sayısı üzerinde herhangi bir kısıtlama yoktur ve tümü aynı öncelikte çalıştırılır. Çakışan izlenen kaynakların, çakışan tetikleme koşullarının ve kaynakların ne sıklıkta yokladığını göz önünde bulundurun.

Çakışan kaynak izleyicileri aşağıdakilere neden olabilir:

- Kaynak konum/öğeler üzerinde olası çekişme.
- Aynı kaynak öğeler için yinelenen aktarma istekleri olabilir.
- Kaynak öğelerin çakışması nedeniyle aktarımlar için beklenmeyen hatalar ya da hatalar.

Birden çok monitör aynı konumu tararsa ve aynı öğeler üzerinde tetiklenirse, aynı öğe için yönetilen aktarım istekleri gönderen iki farklı izleme programı sorunu ortaya çıkabilir.

Kaynak izleme programları, her yoklama aralığı döneminden sonra kaynakların içeriğine bakar. Kaynağın içeriği tetikleyici koşullarıyla karşılaştırılır ve bu koşullar karşılandıysa, kaynak izleme programıyla ilişkili görev çağrılır.

Görev zamanuyumsuz olarak başlatılır. Bir koşul eşleşmesi varsa ve görev başlatılırsa, kaynak izleme programı kaynak içeriğinde daha fazla değişiklik olup olmadığını yoklamaya devam eder. Örneğin, `reports.go` adlı bir dosya izlenen bir dizine geldiği için bir eşleşme oluşursa, görev bir kez başlatılır. Bir sonraki yoklama aralığında, dosya hala var olsa bile, görev yeniden başlatılmaz. Ancak, dosya silinir ve yeniden dizine yerleştirilirse ya da dosya güncellenirse (son değiştirme tarihi özneteliği değiştirilecek şekilde), sonraki tetikleme koşulu denetimi görevin yeniden çağrılmasına neden olur.

IBM MQ 9.1.5öncesinde, bir kaynak izleyicisi yoklama aralığından daha uzun sürecek bir yoklama gerçekleştirirse, bir sonraki yoklama, yürürlükteki yoklama arasında boşluk olmadan biter bitmez başlatılacağı anlamına gelir; bu da, kaynak izleyicilerinin bir aracıya iş gönderme süresini etkileyebilir. Bu, birinci yoklama sırasında bulunan öğeler ikinci yoklama gerçekleştiğinde hala oradaysa performans sorunlarına neden olabilir.

IBM MQ 9.1.5' den kaynak izleme programı ScheduledExecutorhizmetini kullanır ve bir sonraki yoklamayı, yalnızca önceki yoklama tamamlandıktan ve yapılandırılan yoklama aralığı süresinden sonra başlatır. Bu, yoklama süresi yoklama aralığından daha uzunsa, bir önceki yoklamadan hemen sonra başlayan başka bir yoklama yerine, yoklama aralıkları arasında her zaman bir boşluk olacağı anlamına gelir.

IBM MQ 9.1.3' den bir dosya aktarılamadıysa, dosyayı silme ve dizine yeniden yerleştirmeye gerek kalmadan başka bir aktarma isteğinin sunulmasına izin veren kaynak izleyicisi geçmişini temizleyebilir ya da dosyayı son değiştirme tarihi özniteliğini değiştirecek şekilde güncelleyebilirsiniz. Örneğin, dosyanın aktarılması gerektiği, ancak dosyanın değiştirilmesinin mümkün olmadığı durumlarda, geçmişin temizlenmesi yararlı olur. Daha fazla bilgi için bkz [“Kaynak izleyicisi geçmişi temizleniyor” sayfa 273.](#)

Kaynaklar

Managed File Transfer içindeki kaynak izleyicileri aşağıdaki iki tip kaynağın içeriğini yoklayabilir:

Dizinler ya da içiçe dizin yapıları

Sık rastlanan bir senaryo, tetikleyici dosyasının varlığına ilişkin bir dizini izlemektir. Bir dış uygulama birden çok dosyayı işliyor ve bunları bilinen bir kaynak dizine yerleştiriyor olabilir. Uygulama işlemeyi tamamladığında, bir tetikleyici dosyasını izlenen bir konuma yerleştirerek dosyaların aktarılmaya hazır olduğunu ya da başka bir şekilde işlem görmeye hazır olduğunu gösterir. Tetikleyici dosyası bir Managed File Transfer kaynak izleme programı tarafından saptanabilir ve bu dosyaların kaynak dizinden başka bir Managed File Transfer Agent dizinine aktarılması başlatılır.

Varsayılan olarak, belirtilen dizin izlenir. Alt dizinleri incelemek için **fteCreateTransfer** komutunda özyineleme düzeyini ayarlayın.

Bir dizinin izlenmesine ilişkin iki örnek şunlardır:

- Bir tetikleyici dosyasını izleyin (örneğin, `trigger.file`) ve bir genel arama karakteri aktarın (örneğin, `*.zip`).
- `*.zip` dosyasını izleyin ve daha sonra `${FilePath}` dosyasını aktarın (örneğin, aktarımı tetikleyen dosya). Değişken yerine koyma değeri hakkında daha fazla bilgi için bkz. [“MFT kaynak izleyicisi görevlerinin değişken yerine koyma değeri ile uyarlanması” sayfa 255.](#)

Not: `*.zip` programını izleyen ve daha sonra `*.zip` aktaran bir izleme programı yaratmayın. İzleme programı, sisteminizdeki her `.zip` dosyası için `*.zip` aktarma işlemi başlatmayı dener. Yani, izleme programı `*.zip` için `*` aktarım sayısı oluşturur.

Bir dizini izlemek için kaynak izleme programı yaratma örneği için bkz. [“Dizinin izlenmesi ve değişken yerine koyma değerinin kullanılması” sayfa 253.](#)

IBM MQ Kuyruklar

Bir kuyruğu izleme örneği, bir dış uygulamanın ileti oluşturup bunları aynı grup tanıtıcısına sahip bilinen bir kuyruğa koymasındır. Uygulama iletileri kuyruğa koymayı tamamladığında, grubun tamamlandığını gösterir. İletilerin tam grubu bir Managed File Transfer kaynak izleme programı tarafından saptanabilir ve ileti grubunun kaynak kuyruktan bir dosyaya aktarılması başlatılır. Bir kuyruğu izlemek için kaynak izleme programı yaratmaya ilişkin bir örnek için bkz. [“Örnek: MFT kaynağını yapılandırma” sayfa 255.](#)

Not: Her kuyruk için yalnızca bir izleme programı belirleyebilirsiniz. Bir IBM MQ kuyruğunu yoklamak için birden çok izleme programı belirlerseniz, öngörülemez davranış ortaya çıkar.

İzleme veri kümeleri desteklenmiyor.

Koşullar ve tetikleme koşulları

Kaynak başka bir dizgi ya da kalıpla eşleşen bir değer içerdiğinde koşul karşılanır. Koşullar aşağıdakilerden biri olabilir:

- Dosya adı (kalıp) ile eşleştir
- Dosya adı (kalıp) üzerinde eşleşme yok
- Dosya boyutu

- Dosya boyutu bir dizi yoklama için aynı kalırsa eşleştir

Dosya adı eşleştirmesi şu şekilde ifade edilebilir:

- Tam dizgi eşleşmesi
- MFT ile genel arama karakterlerini kullanma konusunda açıklandığı gibi basit genel arama karakteri eşleşmesi
- Düzenli ifade eşleşmesi

Dosya adları, hiçbir zaman eşleşmeyen dosya adlarını tanımlayan bir genel arama karakteri ya da Java düzenli ifadesi kullanılarak dosya adı eşleştirmesinden de çıkarılabilir.

Eşleşen bir dosya algılandığında, dosyanın son değiştirildiği zaman damgası korunur. Sonraki yoklamalar dosyanın değiştirildiğini saptarsa, tetikleme koşulu yeniden yerine getirilir ve görev başlatılır. Koşul, bir dosyanın ne zaman var olmadığını saptayacaksa, izlenen dizindeki hiçbir dosya dosya adı kalıbıyla eşleşmiyorsa, görev başlatılır. Daha sonra, dosya adı kalıbıyla eşleşen dizine bir dosya eklenirse, görev yalnızca dosya silindiğinde başlatılır.

Görevler

Managed File Transfer , kaynak izleme programları tarafından başlatılacak şekilde yapılandırabileceğiniz aşağıdaki iki görev tipini destekler:

Dosya aktarma görevleri

Dosya aktarma görevleri, diğer dosya aktarma görevleriyle aynı şekilde tanımlanır. Bir izleme programının gerektirdiği görev XML ' ini oluşturmanın yararlı bir yolu, `fteCreateTransfer` komutunu **-gt** parametresiyle çalıştırmaktır. Bu komut, aktarma belirtimi de içinde olmak üzere, XML belgesi olarak bir görev tanımlaması oluşturur. Daha sonra görev XML belgesinin adını, `fteCreateMonitor` komutunda **-mt** değiştirgesinin değeri olarak iletiyorsunuz. **fteCreateMonitor** çalıştırıldığında, görev XML belgesini okur. **fteCreateMonitor** çalıştırdıktan sonra, görev XML dosyasında yapılan değişiklikler izleme programı tarafından kullanılmaz.

Bir dosya aktarma görevini kullanırken, bir göreve kaç tetikleyici koşulunun toplu olarak dahil edileceğini seçebilirsiniz. Varsayılan değer, bir tetikleyici koşulunun bir görevi başlatması içindir. Toplu olarak tek bir görevde toplanacak tetikleyici koşullarının sayısını seçmek için `fteCreateMonitor` komutunu **-bs** seçeneğiyle çalıştırabilirsiniz.

Komut görevleri

Komut görevleri Ant komut dosyalarını çalıştırabilir, yürütülebilir programları çağırabilir ya da JCL işlerini çalıştırabilir. Daha fazla bilgi için bkz. “Komutları ve komut dosyalarını başlatmak için MFT izleme görevlerinin yapılandırılması” sayfa 249.

Tetikleyici dosyaları

Kaynak izleme programındaki bir tetikleyici dosyasının içeriğini kullanarak, tek bir aktarma isteğinde aktarılacak bir dosya kümesi tanımlayabilirsiniz. Eşleşen bir tetikleyici dosyası her saptandığında, içeriği kaynak dosya yolları ve isteğe bağlı olarak hedef dosya yolları için ayrıştırılır. Bu dosya yolları, belirlediğiniz görev aktarma XML dosyasındaki dosya öğelerini tanımlamak için kullanılır ve aracıya tek aktarma isteği olarak sunulur. Kaynak izleme programının tanımlaması, tetikleyici içeriğinin etkinleştirilip etkinleştirilmediğini belirler.

Her tetikleyici dosyasının biçimi, her metin satırına aktarılacak tek bir dosya yoludur. Satırın varsayılan biçimi tek bir kaynak dosya yolu ya da virgülle ayrılmış bir kaynak ve hedef dosya yoludur.

Daha fazla bilgi ve örnekler için bkz. “Tetikleyici dosyası kullanılması” sayfa 264.

Kaynak izleme programlarının başlatılması ve durdurulması

IBM MQ 9.3.0' den önce, bir kaynak izleme programını durdurmanın tek yolu, izleme işlemini çalıştıran aracıyı durdurmaktır. Bir kaynak izleme programını yeniden başlatmak için aracıyı tümüyle yeniden başlatmanız gerekir. Daha fazla bilgi için, bkz. “MFT aracısını başlatma” sayfa 226 ve “MFT aracısının durdurulması” sayfa 232.

V 9.3.0 IBM MQ 9.3.0olanağından, **fteStartMonitor** ve **fteStopMonitor** komutlarını kullanarak bir aracıyı durdurmaya ya da yeniden başlatmaya gerek kalmadan kaynak izleyicilerini başlatabilir ve durdurabilirsiniz. Bu, örneğin, aşağıdaki durumlarda kullanışlıdır:

- Bir araçta birden çok kaynak izleyicisi varsa ve bunların yalnızca bir kısmı hatalarla karşılaştıysa, ancak geri kalan kaynak izleme programları çalışmaya devam ediyorsa, başarısız olan kaynak izleme programlarını yeniden başlatmak istersiniz.
- Bir kaynak izleme programını bazı bakım işlerini gerçekleştirmek üzere durdurmak istiyorsanız ya da kaynak izleme programı belirli bir süre için gerekli değilse ve gereksiz yere çalışmasını istemiyorsanız, bu nedenle değerli sistem kaynaklarını tüketin.

V 9.3.0 Daha fazla bilgi için, bkz. “MFT Kaynak İzleme Programının Başlatılması” sayfa 269 ve “MFT Kaynak İzleme Programının Durdurulması” sayfa 270.

V 9.3.0

<i>Çizelge 9. Hangi komutun çalıştırıldığına bağlı olarak bir kaynak izleme programının davranışı</i>	
Komut	Kaynak izleme programının davranışı
V 9.3.0 fteStartMonitor	Aracı çalışıyorsa, kaynak izleme programı durdurulmuşsa başlatılır.
V 9.3.0 fteStopMonitor	Aracı çalışıyorsa, kaynak izleme programı başlatılmış durumdaysa durdurulur.
fteStartAgent	Kaynak izleme programı, aracı başlatma, önceki fteStopMonitor aramalarına bakılmaksızın' in bir parçası olarak başlatılır.
fteStopAgent	Çalışmakta olan kaynak izleme programları durdurulur.

Kaynak izleyicilerinin yedeklenmesi ve geri yüklenmesi

Önceden tanımladığınız kaynak izleyicilerini yedekleyebilir ve ileride bunları yeniden kullanabilirsiniz. Aşağıdaki gibi kullanabileceğiniz çeşitli seçenekler vardır:

- Kaynak izleyicisi yapısını bir XML dosyasına aktarmak için **fteCreateMonitor** komutunu **-ox** değiştirgesiyle ve kaynak izleyicisi yapısını bir XML dosyasından içe aktararak kaynak izleyicisini geri yüklemek için **-ix** değiştirgesiyle birlikte kullanın.
- Tek bir kaynak izleyicisinin tanımlamasını bir XML dosyasına aktarmak için **-ox** ile birlikte **fteListMonitors** komutunu kullanın.
- Birden çok kaynak izleyicisi tanımlamasını belirtilen bir dizine aktarmak için, **fteListMonitors** komutunu **-od** ile birlikte kullanın. Her kaynak izleyicisi tanımlaması ayrı XML dosyasına saklanır. Tek bir kaynak izleyicisi tanımlamasını belirtilen bir dizine aktarmak için **-od** seçeneğini de kullanabilirsiniz.

Daha fazla bilgi için bkz “MFT kaynak izleyicilerini yedekleme ve geri yükleme” sayfa 271.

Kaynak izleyicisi günlük kaydı

IBM MQ 9.1.0' den Managed File Transfer , kaynak izleyici günlük kaydını içerir. Daha fazla bilgi için bkz “MFT kaynak izleyicilerinin günlüğe kaydedilmesi” sayfa 267.

İlgili kavramlar

“MFT kaynak izleyicisi görevlerinin değişken yerine koyma değeri ile uyarlanması” sayfa 255

Etkin bir kaynak izleme programının tetikleme koşulları karşılandığında, tanımlanan görev çağırılır.

Aktarma ya da komut görevini her seferinde aynı hedef aracıyla ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımlamasını da değiştirebilirsiniz. Bunu, görev tanımlaması XML ' ine değişken adları ekleyerek yaparsınız. İzleme programı tetikleyici koşullarının karşılandığını ve görev

tanımlamasının deęişken adlarını içerdiğini saptadığında, deęişken adlarını deęişken deęerleriyle deęiştirir ve görevi çağırır.

İlgili görevler

“Komutları ve komut dosyalarını başlatmak için MFT izleme görevlerinin yapılandırılması” sayfa 249
Kaynak izleme programları, ilişkili görev olarak dosya aktarımlarını gerçekleştirmekle sınırlı deęildir. İzleme programını, yürütülebilir programlar, Ant komut dosyaları ya da JCL işleri de içinde olmak üzere, izleme aracısından dięer komutları çağırarak üzere de yapılandırabilirsiniz. Komutları çağırarak için, izleme görevi tanımlaması XML dosyasını, bağımsız deęişkenler ve özellikler gibi ilgili komut çağırısı deęiştirgelerine sahip bir ya da daha çok komut öęesi içerecek şekilde düzenleyin.

“Örnek: MFT kaynağını yapılandırma” sayfa 255

Bir kaynak izleme programı tarafından izlenecek kaynak olarak IBM MQ kuyruğunu, **fteCreateMonitor** komutuyla **-mq** deęiştirgesini kullanarak belirtebilirsiniz.

“Kuyruğun izlenmesi ve deęişken yerine koyma deęerinin kullanılması” sayfa 261

fteCreateMonitor komutunu kullanarak, bir kuyruğu izleyebilir ve iletileri izlenen kuyruktan bir dosyaya aktarabilirsiniz. İzlenen kuyruktan okunacak ilk iletideki herhangi bir IBM MQ ileti özelliğinin deęeri, görev XML tanımlamasında yerine konabilir ve aktarma davranışını tanımlamak için kullanılabilir.

İlgili başvurular

fteCreateMonitor: MFT kaynak izleme programı yaratılması

fteListİzleme Programları: liste MFT kaynak izleme programları

fteDeleteMonitor: MFT kaynak izleyicisini silme

Komutları ve komut dosyalarını başlatmak için MFT izleme görevlerinin yapılandırılması

Kaynak izleme programları, ilişkili görev olarak dosya aktarımlarını gerçekleştirmekle sınırlı deęildir. İzleme programını, yürütülebilir programlar, Ant komut dosyaları ya da JCL işleri de içinde olmak üzere, izleme aracısından dięer komutları çağırarak üzere de yapılandırabilirsiniz. Komutları çağırarak için, izleme görevi tanımlaması XML dosyasını, bağımsız deęişkenler ve özellikler gibi ilgili komut çağırısı deęiştirgelerine sahip bir ya da daha çok komut öęesi içerecek şekilde düzenleyin.

Bu görev hakkında

İzleme aracısının aramasını istediğiniz yürütülür programın, Ant komut dosyasının ya da JCL işinin dosya yolu, izleme aracısının `commandPath` içinde yer almalıdır. Komut yolu özelliği hakkında bilgi için bkz. [commandPath MFT property](#).

Görev tanımlaması XML belgesini aşağıdaki yollardan biriyle yaratabilirsiniz:

- Görev tanımlaması XML belgesini `FileTransfer.xsd` şemasına göre el ile yaratın.
- Oluşturulan bir XML belgesini görev tanımlamanız için temel olarak kullanın.

Bir aktarma görevi ya da komut görevi istesiniz de, görev tanımlaması bir `<request>` kök öęesiyle başlamalıdır. `<request>` öęesinin alt öęesi `<managedTransfer>` ya da `<managedCall>` olmalıdır. Genellikle çalıştırılacak tek bir komut ya da komut dosyası olduğunda `<managedCall>` ve görevin bir dosya aktarımı ve isteęe baęlı olarak en fazla dört komut çağırısı içermesini istiyorsanız `<managedTransfer>` seçeneğini belirleyebilirsiniz.

Yordam

- Görev tanımlaması XML belgesini `FileTransfer.xsd` şemasına göre el ile yaratmak için bkz. [“Şemaya göre el ile görev tanımlaması XML yaratılması” sayfa 250](#).
- Oluşturulan bir belgeyi deęiştirerek bir görev tanımlaması yaratmak için, **fteCreateTransfer -gt** parametresi tarafından oluşturulan XML belgesini düzenleyin. Daha fazla bilgi için bkz [“Oluşturulan bir belgeyi deęiştirerek görev tanımlaması belgesi yaratılması” sayfa 252](#).

Şemaya göre el ile görev tanımlaması XML yaratılması

FileTransfer.xsd şemasına göre el ile bir görev tanımlaması XML dosyası yaratabilirsiniz.

Bu görev hakkında

FileTransfer.xsd şeması `MQ_INSTALLATION_PATH/mqft/samples/schema` içinde bulunabilir. Bu şemayla ilgili daha fazla bilgi için bkz. [Dosya aktarma isteği iletisi biçimi](#).

Örnek

Aşağıdaki örnekte, `cleanup.xml`, olarak kaydedilen ve `RunCleanup.xml` adlı bir Ant komut dosyasını çağırmak için `<managedCall>` ögesini kullanan örnek bir görev tanımlaması XML belgesi gösterilmektedir. `RunCleanup.xml` Ant komut dosyası, izleme aracısının `commandPath` üzerinde bulunmalıdır.

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedCall>
    <originator>
      <hostName>hostName</hostName>
      <userID>userID</userID>
      <mqmdUserID>mqmdUserID</mqmdUserID>
    </originator>
    <agent QMgr="QM1" agent="AGENT1"/>
    <reply QMGR="QM1">reply</reply>
    <transferSet priority="1">
      <metaDataSet>
        <metaData key="name1">value1</metaData>
      </metaDataSet>
      <call>
        <command name="RunCleanup.xml" type="antscript" retryCount="2"
          retryWait="30" successRC="0">
          <target>check_exists</target>
          <target>copy_to_archive</target>
          <target>rename_temps</target>
          <target>delete_files</target>
          <property name="trigger.filename" value="{FileName}"/>
          <property name="trigger.path" value="{FilePath}"/>
        </command>
      </call>
    </transferSet>
  </managedCall>
</request>
```

`<agent>` ögesi, `commandPath` içinde Ant komut dosyasıyla yapılandırılan Managed File Transfer Agent ögesini belirtir.

`<call><command>` . . . yapısı, çalıştırmak istediğiniz yürütülür dosyayı ya da komut dosyasını tanımlar. Komut, aşağıdaki değerlerden birine sahip olabilen isteğe bağlı bir `type` özniteliğini alır:

antscript

Ayrı bir JVM ' de Ant komut dosyasını çalıştırır.

Yürütülebilir

Yürütülebilir bir programı çağırır.

jcl

Bir JCL işini çağırır.

`type` özniteliğini atlarsanız, yürütülür varsayılan değeri kullanılır.

`name` özniteliği, herhangi bir yol bilgisi olmadan, çalıştırmak istediğiniz Ant komut dosyasının, yürütülür dosyanın ya da JCL işinin adını belirtir. Aracı, aracının `agent.properties` dosyasındaki `commandPath` özelliği tarafından belirtilen konumlarda komut dosyasını ya da programı arar.

`retrycount` özniteliği, program başarılı bir dönüş kodu döndürmezse, programın kaç kez yeniden çağrılacağını belirtir. Bu öznitelige atanan değer negatif olmamalıdır. `retrycount` özniteliğini belirtmezseniz, varsayılan değer olan sıfır kullanılır.

retrywait özniteliği, program çağrısını yeniden denemeden önce beklenecek süreyi saniye cinsinden belirtir. Bu özniteliğe atanan değer negatif olmamalıdır. retrywait özniteliğini belirtmezseniz, varsayılan değer olan sıfır kullanılır.

successrc özniteliği, program çağırmanın ne zaman başarılı bir şekilde çalışacağını saptamak için kullanılan bir ifadedir. Komuta ilişkin işlem dönüş kodu bu ifade kullanılarak değerlendirilir. Değer, Boole OR işlecini belirtmek için düşey çubuk (|) karakteriyle birlikte bir ya da daha çok ifadeden ya da ve (&) işaretinden oluşabilir. Boole işlecini (AND) belirtmek için kullanılır. Her ifade aşağıdaki ifade tiplerinden biri olabilir:

- Süreç dönüş kodu ile sayı arasında bir eşitlik testini gösteren sayı.
- Sayı ile işlem dönüş kodu arasında büyük bir testi belirtmek için başına büyüktür karakteri (>) eklenmiş bir sayı.
- Başına küçüktür karakteri eklenmiş bir sayı (<) Sayı ile süreç dönüş kodu arasında daha küçük bir testi belirtmek için.
- Sayı ile işlem dönüş kodu arasında eşit olmayan bir sınamayı belirtmek için başına ünlem işareti karakteri (!) eklenmiş bir sayı. Örneğin: > 2 & < 7 &! 5 | 0 | 14, şu dönüş kodlarının başarılı olduğu şeklinde yorumlanır: 0, 3, 4, 6, 14. Diğer tüm dönüş kodları başarısız olarak yorumlanır.

successrc özniteliğini belirtmezseniz, varsayılan değer olan sıfır kullanılır. Bu, komutun ancak ve ancak sıfır kodu döndürürse başarılı bir şekilde çalıştırılmış olduğu anlamına gelir.

Bir Ant komut dosyası için genellikle <target> ve <property> öğelerini belirtirsiniz. <target> öğe değerleri, Ant komut dosyasındaki hedef adlarıyla eşleşmelidir.

Yürütülebilir programlar için <argument> öğeleri belirtebilirsiniz. İç içe geçmiş bağımsız değişken öğeleri, program çağrısının bir parçası olarak çağrılan programa geçirilecek bağımsız değişkenleri belirtir. Program bağımsız değişkenleri, bağımsız değişken öğeleri tarafından belirtilen değerlerden bağımsız değişken öğelerine rastlanan sırayla oluşturulur. Bir program çağrısının iç içe geçmiş öğeleri olarak sıfır ya da daha fazla bağımsız değişken öğesi belirtebilirsiniz.

Denetimci, <managedCall> öğesini içeren görev tanımlaması XML belgesini kullanarak izleyiciyi olağan olarak tanımlar ve başlatır. Örneğin:

```
fteCreateMonitor -ma AGENT1 -mm QM1 -md /monitored -mn MONITOR01 -mt /tasks/cleanuptask.xml -pi 30 -pu seconds -tr match,*go
```

Aktarma tanımlaması XML belgesinin yolu, **fteCreateMonitor** komutunu çalıştırdığınız yerel dosya sisteminde olmalıdır (bu örnekte /tasks/cleanuptask.xml). cleanuptask.xml belgesi yalnızca kaynak izleyicisini yaratmak için kullanılır. cleanuptask.xml belgesinin başvurduğu görevler (Ant komut dosyaları ya da JCL işleri), izleme aracısının komut yolunda olmalıdır. İzleme tetikleyicisi koşulu yerine getirildiğinde, görev tanımlaması XML 'indeki tüm değişkenler, izleme programındaki gerçek değerlerle değiştirilir. Bu nedenle, örneğin \${FilePath}, aracıya /monitored/cleanup.goile gönderilen istek iletilisinde değiştirilir. İstek iletilisi aracı komut kuyruğuna yerleştirilir. Komut işlemcisi, isteğin bir program çağrısı için olduğunu saptar ve belirlenen programı başlatır. antscript tipinde bir komut çağrılırsa, yeni bir JVM başlatılır ve Ant görevi yeni JVM altında çalışır. Değişken yerine koyma değerinin kullanılmasıyla ilgili daha fazla bilgi için [Görevleri değişken yerine koyma ile özelleştirme](#) başlıklı konuya bakın.

İlgili kavramlar

“MFT kaynak izleyicisi görevlerinin değişken yerine koyma değeri ile uyarlanması” sayfa 255

Etkin bir kaynak izleme programının tetikleme koşulları karşılandığında, tanımlanan görev çağrılır.

Aktarma ya da komut görevini her seferinde aynı hedef aracıyla ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımlamasını da değiştirebilirsiniz. Bunu, görev tanımlaması XML 'ine değişken adları ekleyerek yaparsınız. İzleme programı tetikleyici koşullarının karşılandığını ve görev tanımlamasının değişken adlarını içerdiğini saptadığında, değişken adlarını değişken değerleriyle değiştirir ve görevi çağırır.

İlgili başvurular

[Dosya aktarma isteği iletili biçimi](#)

[commandPath MFT özelliği](#)

Oluşturulan bir belgeyi değiştirerek görev tanımlaması belgesi yaratılması
fteCreateTransfer' un **-gt** seçeneğiyle oluşturulan XML belgesini değiştirerek, izleme görevi tanımlaması belgesini yaratabilirsiniz.

Bu görev hakkında

Oluşturulan belgenin bir `<request>` ve onu izleyen `<managedTransfer>` ögesi var. Bu görev tanımlamasını geçerli bir `<managedCall>` yapısına dönüştürmek için aşağıdaki adımları izleyin:

Yordam

1. `<managedTransfer>` başlangıç ve bitiş etiketlerini `<managedCall>` etiketleriyle değiştirin.
2. `<schedule>` öğelerini ve alt düğümlerini kaldırın.
3. İzleme aracı yapılandırma ayrıntılarıyla eşleşmesi için `<sourceAgent>` başlangıç ve bitiş etiketlerini `<agent>` ile değiştirin.
4. `<destinationAgent>` ve `<trigger>` öğelerini kaldırın.
5. `<item>` öğelerini kaldırın.
6. `preSourceCall`, `postSourceCall`, `preDestinationCall` ve `postDestinationCall` öğelerini kaldırın.
7. `<transferSet>` ögesi içine yeni bir `<call>...</call>` yapısı ekleyin. Bu yapı, aşağıdaki örnekte gösterildiği gibi komut tanımını içerir:

```
<call>
  <command name="RunCleanup.xml" type="antscript" retryCount="2"
  retryWait="30" successRC="0">
    <target>check_exists</target>
    <target>copy_to_archive</target>
    <target>rename_temps</target>
    <target>delete_files</target>
    <property name="trigger.filename" value="{FileName}"/>
    <property name="trigger.path" value="{FilePath}"/>
  </command>
</call>
```

Örnek

Tüm dosya aktarma ayrıntıları da içinde olmak üzere `<managedTransfer>` ögesini koruyabilir ve en çok dört komut çağrısı ekleyebilirsiniz. Bu durumda, `<metaDataSet>` ve `<item>` öğeleri arasında aşağıdaki çağrı öğelerinden herhangi birini ekleyebilirsiniz:

preSourceÇağrısı

Aktarmaya başlamadan önce kaynak aracıdaki bir programı çağırın.

postSource(PostSource) Çağrısı

Aktarımı tamamladıktan sonra kaynak aracıdaki bir programı çağırın.

preDestination(ön Hedef) Çağrısı

Aktarmaya başlamadan önce hedef aracıdaki bir programı çağırın.

postDestination(PostDestination) Çağrısı

Aktarımı tamamladıktan sonra hedef aracıdaki bir programı çağırın.

Bu öğelerin her biri, önceki örnekte açıklandığı gibi `<command>` öge yapısını alır. `FileTransfer.xsd` şeması, çeşitli çağrı öğeleri tarafından kullanılan tipleri tanımlar.

Aşağıdaki örnek, bir görev tanımlaması belgesinde `preSourceCall`, `postSourceCall`, `preDestinationCall` ve `postDestinationCall` öğelerini göstermektedir:

```
⋮
<transferSet priority="1">
  <metaDataSet>
    <metaData key="key1">value1</metaData>
  </metaDataSet>
  <preSourceCall>
    <command name="send.exe" retryCount="0" retryWait="0" successRC="0"
```

```

        type="executable">
        <argument>report1.pdf</argument>
        <argument>>true</argument>
    </command>
</preSourceCall>
<postSourceCall>
    <command name="//DO_IT.JCL" retryCount="0" retryWait="0" successRC="0"
        type="jcl">
        <argument>argument</argument>
    </command>
</postSourceCall>
<preDestinationCall>
    <command name="ant_script.xml" retryCount="0" retryWait="0" successRC="0"
        type="antscript">
        <target>step1</target>
        <property name="name" value="value"/>
    </command>
</preDestinationCall>
<postDestinationCall>
    <command name="runit.cmd" retryCount="0" retryWait="0" successRC="0" />
</postDestinationCall>
<item checksumMethod="none" mode="binary">
:

```

Farklı komut tiplerini aktarıma karıştırabilirsiniz. Bağımsız değişken, hedef ve özellik öğeleri isteğe bağlıdır.

Dizinin izlenmesi ve değişken yerine koyma değerinin kullanılması

ftCreateMonitor komutunu kullanarak bir dizini izleyebilirsiniz. Bir yerine koyma değişkeninin değeri, görev XML tanımlamasında yerine konabilir ve aktarma davranışını tanımlamak için kullanılabilir.

Bu görev hakkında

Bu örnekte, kaynak aracıya AGENT_HOP adı verilir. AGENT_HOP ' un izlediği dizinin adı /test/monitored. Aracı, dizini her 5 dakikada bir yoklamaktadır.

Bir .zip dosyası dizine yazıldıktan sonra, dosyayı dizine yazan uygulama aynı dizine bir tetikleyici dosyası yazar. Tetikleyici dosyasının adı .zip dosyasının adıyla aynı, ancak dosya uzantısı farklı. Örneğin, file1.zip dosyası dizine yazıldıktan sonra, file1.go dosyası dizine yazılır. Kaynak izleyicisi, *.go örneğiyle eşleşen dosyalar için dizini izler ve ilişkili .zip dosyasının aktarılmasını istemek için değişken yerine koyma değerini kullanır.

Yordam

1. İzleme programının tetiklendiğinde gerçekleştirdiği görevi tanımlayan görev XML dosyasını yaratın.

```

<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
    xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
    <managedTransfer>
        <originator>
            <hostName>blue.example.com</hostName>
            <userID>USER1</userID>
        </originator>
        <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
        <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
        <transferSet>
            <item mode="binary" checksumMethod="none">
                <source>
                    <file>/test/monitored/${fileName}{token=1}{separator=.}.zip</file>
                </source>
                <destination type="file" exist="overwrite">
                    <file>/out/${fileName}{token=1}{separator=.}.zip</file>
                </destination>
            </item>
        </transferSet>
    </managedTransfer>
</request>

```

Tetikleyici dosya ile ilişkili değerlerle değiştirilen değişkenler **koyu** olarak vurgulanır. Bu görev XML ' i /home/USER1/task.xml dosyasına kaydedildi

2. /test/monitoreddizinini izlemek için bir kaynak izleme programı yaratın.

Şu komutu verin:

```
fteCreateMonitor -ma AGENT_HOP -mm QM_HOP -md /test/monitored  
-mn myMonitor -mt /home/USER1/task.xml  
-tr match,*.go -pi 5 -pu minutes
```

3. Bir kullanıcı ya da program jump.zip dosyasını /test/monitoreddizinine yazar ve jump.go dosyasını dizine yazar.

4. İzleme programı jump.gokütüğünün varlığıyla tetiklenir. Aracı, tetikleyici dosyasıyla ilgili bilgileri görev XML ' ine yerleştirir.

Bu, görev XML ' inin aşağıdakilere dönüştürülmesine neden oluyor:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<request version="4.00"  
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">  
  <managedTransfer>  
    <originator>  
      <hostName>blue.example.com</hostName>  
      <userID>USER1</userID>  
    </originator>  
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />  
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />  
    <transferSet>  
      <item mode="binary" checksumMethod="none">  
        <source>  
          <file>/test/monitored/jump.zip</file>  
        </source>  
        <destination type="file" exist="overwrite">  
          <file>/out/jump.zip</file>  
        </destination>  
      </item>  
    </transferSet>  
  </managedTransfer>  
</request>
```

Sonuçlar

XML görevi tarafından tanımlanan aktarma gerçekleştirilir. jump.zip dosyası AGENT_HOP tarafından /test/monitored dizininden okunur ve AGENT_SKIP ' in çalıştığı sistemde bulunan /out/jump.zip adlı bir dosyaya aktarılır.

İlgili kavramlar

[“MFT kaynak izleyicisi görevlerinin değişken yerine koyma değeri ile uyarlanması” sayfa 255](#)

Etkin bir kaynak izleme programının tetikleme koşulları karşılandığında, tanımlanan görev çağırılır.

Aktarma ya da komut görevini her seferinde aynı hedef aracıyla ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımlamasını da değiştirebilirsiniz. Bunu, görev tanımlaması XML ' ine değişken adları ekleyerek yaparsınız. İzleme programı tetikleyici koşullarının karşılandığını ve görev tanımlamasının değişken adlarını içerdiğini saptadığında, değişken adlarını değişken değerleriyle değiştirir ve görevi çağırır.

İlgili görevler

[“Komutları ve komut dosyalarını başlatmak için MFT izleme görevlerinin yapılandırılması” sayfa 249](#)

Kaynak izleme programları, ilişkili görev olarak dosya aktarımlarını gerçekleştirmekle sınırlı değildir.

İzleme programını, yürütülebilir programlar, Ant komut dosyaları ya da JCL işleri de içinde olmak üzere, izleme aracısından diğer komutları çağırarak de yapılandırabilirsiniz. Komutları çağırarak için, izleme görevi tanımlaması XML dosyasını, bağımsız değişkenler ve özellikler gibi ilgili komut çağırısı değiştirgelerine sahip bir ya da daha çok komut ögesi içerecek şekilde düzenleyin.

İlgili başvurular

fteCreateMonitor: MFT kaynak izleme programı yaratılması

Örnek: MFT kaynağını yapılandırma

Bir kaynak izleme programı tarafından izlenecek kaynak olarak IBM MQ kuyruğunu, **fteCreateMonitor** komutuyla **-mq** deęiřtirgesini kullanarak belirtebilirsiniz.

Bu görev hakkında

Bu örnekte, izlenecek kaynak **MONITORED_QUEUE** kuyruğudur. Bu kuyruk, izleme aracısının kuyruk yöneticisinde (**QM_NEPTUNE**) olmalıdır. Kuyruğun izlendięi koşul, eksiksiz bir ileti grubunun varlıęıdır. Koşul karşılandığında gerçekleştirilecek görev **task.xml** dosyasında tanımlanır.

Not: Tek bir kuyruęu izlemek için birden çok kaynak izleme programı yaratmayın. Bunu yaparsanız, öngörülemeyen davranışlar ortaya çıkar.

Yordam

Ařaęıdaki komutu yazın:

```
fteCreateMonitor -ma AGENT_NEPTUNE -mn myMonitor -mm QM_NEPTUNE -mq MONITORED_QUEUE  
-mt task.xml -tr completeGroups -pi 5 -pu minutes
```

İzleme programı, **completeGroups** koşulunun doęru olup olmadıęını görmek için her beř dakikada bir kuyruęu denetler. Kuyrukta bir ya da daha çok tamamlanmış grup varsa, izleme programı, **task.xml** dosyasında tanımlanan görevi her tam grup için bir kez çalıştırır.

İlgili kavramlar

“MFT kaynak izleyicisi görevlerinin deęişken yerine koyma deęeri ile uyarlanması” sayfa 255

Etkin bir kaynak izleme programının tetikleme koşulları karşılandığında, tanımlanan görev çağırılır.

Aktarma ya da komut görevini her seferinde aynı hedef aracıyla ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımlamasını da deęiřtirebilirsiniz. Bunu, görev tanımlaması XML ' ine deęişken adları ekleyerek yaparsınız. İzleme programı tetikleyici koşullarının karşılandığını ve görev tanımlamasının deęişken adlarını içerdini saptadığında, deęişken adlarını deęişken deęerleriyle deęiřtirir ve görevi çağırır.

İlgili görevler

“Komutları ve komut dosyalarını başlatmak için MFT izleme görevlerinin yapılandırılması” sayfa 249

Kaynak izleme programları, ilişkili görev olarak dosya aktarımlarını gerçekleřtirmekle sınırlı deęildir.

İzleme programını, yürütülebilir programlar, Ant komut dosyaları ya da JCL işleri de içinde olmak üzere, izleme aracısından dięer komutları çağırarak deęer de yapılandırabilirsiniz. Komutları çağırarak için, izleme görevi tanımlaması XML dosyasını, bağımsız deęişkenler ve özellikler gibi ilgili komut çağırısı deęiřtirgelerine sahip bir ya da daha çok komut öęesi içerecek şekilde düzenleyin.

“Kuyruğun izlenmesi ve deęişken yerine koyma deęerinin kullanılması” sayfa 261

fteCreateMonitor komutunu kullanarak, bir kuyruęu izleyebilir ve iletileri izlenen kuyruktan bir dosyaya aktarabilirsiniz. İzlenen kuyruktan okunacak ilk iletideki herhangi bir IBM MQ ileti özellięinin deęeri, görev XML tanımlamasında yerine konabilir ve aktarma davranışını tanımlamak için kullanılabilir.

İlgili başvurular

fteCreateMonitor: MFT kaynak izleme programı yaratılması

MFT kaynak izleyicisi görevlerinin deęişken yerine koyma deęeri ile uyarlanması

Etkin bir kaynak izleme programının tetikleme koşulları karşılandığında, tanımlanan görev çağırılır.

Aktarma ya da komut görevini her seferinde aynı hedef aracıyla ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımlamasını da deęiřtirebilirsiniz. Bunu, görev tanımlaması XML ' ine deęişken adları ekleyerek yaparsınız. İzleme programı tetikleyici koşullarının karşılandığını ve görev tanımlamasının deęişken adlarını içerdini saptadığında, deęişken adlarını deęişken deęerleriyle deęiřtirir ve görevi çağırır.



Uyarı: Değişken adları büyük ve küçük harfe duyarlı değildir.

Yerine koyma değeri için kullanılan değişkenler yalnızca pozitif tetikleyici koşulları için kullanılabilir. Yalnızca eşleşme ve fileSize tetikleme koşulları değişkenlerin değiştirilmesine neden olur. Bir noMatch koşulu kullanılırsa ve görev tanımlamasında yerine koyma değişkeni adları varsa, görev çağrılmaz ve izleme programı 110 dönüş kodunu ve BFGDM0060Ehata iletisini görüntüler.

İzlenen kaynak bir kuyruksa

İzlenen kuyruktan okunacak ilk iletideki herhangi bir IBM MQ ileti özelliğinin değeri, görev XML tanımlamasında yerine konabilir.

Kullanıcı tanımlı ileti özelliklerine usr . öneki eklenir, ancak değişken adına bu önek eklenmez. Değişken adlarının başında bir dolar işareti (\$) karakteri olmalı ve bu karakter kaşlı ayraç içine alınmalıdır { }.

Örneğin, \${destFileName} , kaynak kuyruktan okunacak ilk iletinin usr . destFileName ileti özelliğinin değeriyle değiştirilir. Daha fazla bilgi için bkz. [Kaynak kuyruklardaki iletilerden MFT tarafından okunan MQ ileti özellikleri](#) ve [“Kuyruğun izlenmesi ve değişken yerine koyma değerinin kullanılması” sayfa 261](#).

Bir değişken ileti özelliği olarak tanımlanmazsa, izleyici bir BFGDM0060E hatası bildirir ve dönüş kodu 110 döndürür (İzleme görevi değişkeni yerine koyma başarısız oldu). Buna ek olarak, aracı olay günlüğüne şu hata iletisini yazar (outputN.log):

```
BFGDM0113W: Trigger failure for <monitor name> for reason BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

İzleme programı için ılımlı ya da ayrıntılı kaynak izleyicisi günlük kaydı etkinleştirilmişse, izleme programı aracının kaynak izleyicisi olay günlüğüne aşağıdaki iletiyi yazar (resmoneventN.log):

```
BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

Kaynak izleyicisi günlük kaydına ilişkin ek bilgi için bkz. [“MFT kaynak izleyicilerinin günlüğe kaydedilmesi” sayfa 267](#) .

Aşağıdaki tablo, varsayılan olarak hangi yerine koyma değişkenlerinin sağlandığı gösterir. Örneğin, \${AGENTNAME} kaynak izleyici aracısının adıyla değiştirilir.

Çizelge 10. Varsayılan olarak sağlanan yerine koyma değişkenleri	
Değişken	Açıklama
ARACI ADI	Kaynak izleyici aracısının adı.
queueName	İzlenmekte olan kuyruğun adı.
Kodlama	Kuyruktaki ilk iletinin ya da gruptaki ilk iletinin karakter kodlaması.
MESSAGEID	Kuyruktaki ilk iletinin ya da gruptaki ilk iletinin IBM MQ ileti tanıtıcısı.
GRUPID	Yalnızca tek bir ileti bulunursa, grubun ya da ileti tanıtıcısının IBM MQ grup tanıtıcısı. Bu değişken, tam grupları izliyorsanız ayarlanır.
CurrentTimeDamgasi	İzleme programının tetiklediği yerel saate dayalı bir zaman damgası. Zaman damgası değeri, aracı için benzersizdir.
CurrentTimeStamp UTC	İzleme programının tetiklediği UTC saat dilimine dayalı zaman damgası. Zaman damgası değeri, aracı için benzersizdir.

İzlenen kaynak bir dizinse

Aşağıdaki çizelge, görev XML tanımlamasında yerine konabilecek değişken adları kümesini göstermektedir.

<i>Çizelge 11. Yerine konabilecek değişkenler</i>	
Değişken	Açıklama
FilePath	Tetikleyici dosyasının tam yol adı.
FileName	Tetikleyicinin dosya adı kısmı.
LastModified(Son Değişirme) Saati	Tetikleyici dosyasının en son değiştirildiği saat. Bu saat, aracının çalıştığı saat diliminin yerel saati olarak ifade edilir ve ISO 8601 saati olarak biçimlendirilir.
LastModified(Son Değişirme Tarihi)	Tetikleyici dosyasının en son değiştirildiği tarih. Bu tarih, aracının çalıştığı saat diliminin yerel tarihi olarak ifade edilir ve ISO 8601 tarihi olarak biçimlendirilir.
LastModifiedTimeUTC	Tetikleyici dosyasının en son değiştirildiği saat. Bu saat, UTC saat dilimine dönüştürülen yerel saat olarak ifade edilir ve ISO 8601 saati olarak biçimlendirilir.
LastModifiedDateUTC	Tetikleyici dosyasının en son değiştirildiği tarih. Bu tarih, UTC saat dilimine dönüştürülen yerel tarih olarak ifade edilir ve ISO 8601 tarihi olarak biçimlendirilir.
AgentName	Kaynak izleyici aracısının adı.
CurrentTimeDamgası	İzleme programının tetiklediği yerel saate dayalı bir zaman damgası. Zaman damgası değeri, aracı için benzersizdir.
CurrentTimeStampUTC	İzleme programının tetiklediği UTC saat dilimine dayalı zaman damgası. Zaman damgası değeri, aracı için benzersizdir.

İzlenen kaynak bir tetikleyici dosyaysa

Aşağıdaki çizelge, bir kaynak izleme programı aktarılması gereken dosyaları saptamak için bir tetikleyici dosyasının içeriğini kullandığında yerine konabilecek değişken adları kümesini gösterir.

<i>Çizelge 12. Tetikleyici dosyası kullanılırken yerine konabilecek değişkenler</i>	
Değişken	Açıklama
contentSource	Kaynak dosyanın tam yol adı.
contentDestination	Hedef dosyanın tam yol adı.

Değişken adlarından önce bir dolar işareti (\$) karakteri olmalı ve kaşılı ayraç içine alınmalıdır, { }. Örneğin, `FilePath`, eşleşen tetikleyici dosyasının tam olarak nitelenmiş dosya yoluyla değiştirilir.

Daha fazla iyileştirme sağlamak için değişken adlarına uygulanabilen iki özel anahtar sözcük vardır. Bunlar:

belirteç

Yerine konacak simge dizini (soldan 1 'den başlayarak ve sağdan -1 'den başlayarak)

ayırıcı

Değişken değerini bölümlenmek için tek bir karakter. Varsayılan değer, AIX and Linux altyapılarında eğik çizgi karakteri (/) ya da Windows altyapılarında ters eğik çizgi karakteridir (\), ancak ayırıcı, değişken değerinde görünebilecek herhangi bir geçerli karakter olabilir.

Bir değişken adında ayırıcı anahtar sözcüğü belirtilirse, değişken değeri ayırıcı karaktere göre simgelere bölünür.

Simge anahtar sözcüğüne atanan değer, değişken adını değiştirmek için hangi simgenin kullanılacağını seçmek üzere dizin olarak kullanılır. Belirteç dizini değişkendir ilk karakterle görelilik olup 1 'den başlar. Simge anahtar sözcüğü belirtilmezse, tüm değişken eklenir.

İleti XML ' inde bir aracı adıyla değiştirilen değerler büyük ve küçük harfe duyarlı olmayan bir şekilde işlenir. Tüm Managed File Transfer Agent adları büyük harflidir. Paris değeri, ileti XML ' inde bir aracı özneliğiyle değiştirilirse, bu değer PARISaracısına başvuru olarak yorumlanır.

İlgili kavramlar

“Örnekler: Kaynak izleyicisi tanımlamaları için değişken yerine koyma değeri” sayfa 258

XML ve IBM MQ Explorerkullanan kaynak izleyicisi tanımlamaları için değişken yerine koyma değeri örnekleri.

İlgili görevler

Değişken yerine koyma değeri birden çok dosyanın tek bir dosya adına geçmesine neden olursa ne yapmanız gerekir?

Örnekler: Kaynak izleyicisi tanımlamaları için değişken yerine koyma değeri

XML ve IBM MQ Explorerkullanan kaynak izleyicisi tanımlamaları için değişken yerine koyma değeri örnekleri.

Değişken yerine koyma değerinin nasıl çalıştığını gösteren örnekler

Eşleşen tetikleyici dosyasının dosya yolunun Windows üzerinde

c:\MONITOR\REPORTS\Paris\Report2009.doc ve AIX and Linux platformlarında /MONITOR/REPORTS/Paris/Report2009.doc olduğu varsayılarak, değişkenler aşağıdaki çizelgede gösterildiği gibi değiştirilir.

Değişken belirtimi	Değişken yerine koyma işleminden sonra
\${FilePath}	Windows :c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux : /MONITOR/REPORTS/Paris/Report2009.doc
\${FilePath{token=1}{separator=.}}	Windows :c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux : /MONITOR/REPORTS/Paris/Report2009
\${FilePath{token=2}{separator=.}}	Windows: doc AIX and Linux : doc
\${FilePath{token=3}}	Windows : RAPORLAR AIX and Linux : Paris

Aşağıdaki tabloda gösterildiği gibi, değişkenin son karakterine göre belirteç seçmek için negatif bir simge dizini de belirtebilirsiniz. Çizelgedeki örnekler aynı değişken değerini (c:\MONITOR\REPORTS\Paris\Report2009.doc on Windows ve /MONITOR/REPORTS/Paris/Report2009.doc on AIX and Linux) kullanır.

Değişken belirtimi	Değişken yerine koyma işleminden sonra
\${FilePath}	Windows :c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux : /MONITOR/REPORTS/Paris/Report2009.doc

Çizelge 14. Negatif simge dizini kullanma örnekleri (devamı var)

Değişken belirtimi	Değişken yerine koyma işleminden sonra
\${FilePath{token=-2}{separator=.}}	Windows :c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux : /MONITOR/REPORTS/Paris/Report2009
\${FilePath{token=-2}{separator=\}}	Windows : Paris AIX and Linux : Paris
\${FilePath{token=-4}}	Windows : İZLEME PROGRAMI AIX and Linux : İZLEME PROGRAMI

Yerine koyma değeri için kullanılan değişkenler yalnızca aşağıdaki pozitif tetikleyici koşulları ve pozitif tetikleme koşulu kuralı için bir kural dışı durum olan noSizeDeğiştir seçeneği için kullanılabilir:

- eşleşen
- fileSize
- noSizeDeğiştir

Bir noMatch koşulu kullanılırsa ve görev tanımlamasında yerine koyma değişkeni adları varsa, görev çağrılmaz ve izleme programı 110 dönüş kodunu ve BFGDM0060Ehata iletisini görüntüler.

XML Kullanımı-Örnek

Aşağıdaki örnek görev tanımlaması XML, aktarım için kaynak aracı olarak izleme aracı adını kullanır (Paris), dosya yolundaki sondan bir önceki dizin adını aktarma için hedef aracı adı olarak kullanır (Report2009) ve aktarılan dosyayı, .rptuzantısıyla tetikleyici dosyası adının kökü olarak yeniden adlandırır.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="${AgentName}" QMgr="QM1" />
    <destinationAgent agent="${FilePath{token=-2}}" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/${FileName{token=1}{separator=.}}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Bu, görev XML 'inin aşağıdakilere dönüştürülmesine neden oluyor:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT1" QMgr="QM1" />
  </managedTransfer>
</request>
```

```

<destinationAgent agent="Paris" QMgr="QMD" />
<transferSet>
  <item mode="binary" checksumMethod="MD5">
    <source recursive="false" disposition="leave">
      <file>c:/incoming/reports/summary/report.doc</file>
    </source>
    <destination type="file" exist="overwrite">
      <file>/reports/Report2009.rpt</file>
    </destination>
  </item>
</transferSet>
</managedTransfer>
</request>

```

<destinationAgent> ögesinin agent özniteliğindeki `${FilePath}{token=-2}` değişkeni Parisdeğeriyle değiştirilir. Bu değer büyük ve küçük harfe duyarlı olmayan bir şekilde işlenir ve PARISaracısına başvuru olarak yorumlanır.

IBM MQ Explorer ile ilgili örnekler

IBM MQ Exploreraracılığıyla bir kaynak izleme programı yaratılırken ve izleme programı özellikleri ve tetikleyici koşulları belirtildiğinde, bu seçenek izleme programına aktarma öğeleri ekleme seçeneği verilir. Aşağıdaki örneklerde, bir kaynak izleyicisi eşleşmesinden kaynaklanan aktarımları özelleştirmek için `${FilePath}` ve `${FileName}` değişkenlerinin "**Aktarma öğesi ekle panosunda**" nasıl kullanılabileceği gösterilmektedir.

Örnek 1

Bir tetikleyici koşulu karşılandığında kaynak dosyayı başka bir konuma aktarmak için `${FilePath}` değişkeni kullanılabilir:

- Kaynak **Dosya adı** ' nı `${FilePath}` olarak ayarlayın.
- Hedef için **Tip** açılan menüsünden **Dizin** seçeneğini belirleyin.
- Hedef **Dosya adı** değerini, kaynak dosyanın aktarılmasını istediğiniz yer olarak ayarlayın; örneğin, C:\MFT\out\olabilir.

Örnek 2

Kaynak dosyayı başka bir konuma aktarmak ve dosyanın uzantısını değiştirmek için, `${FileName}` değişkeni `${FilePath}` değişkeniyle birlikte kullanılabilir:

Aşağıdaki örnekte, kaynak dosyanın dosya yolunun

C:\MONITOR\REPORTS\Paris\Report2009.docdeğerine eşit olduğu varsayılır:

- Kaynak **Dosya adı** ' nı `${FilePath}` olarak ayarlayın.
- Hedef **Dosya adı** ' nı, kaynak dosyanın aktarılmasını istediğiniz yer ve ardından `${FileName}{token=1}{separator=.}` ve ardından dosyanın yeni uzantısı olarak ayarlayın. Örneğin, bu C:\MFT\out\`${FileName}{token=1}{separator=.}`.rptolabilir; bu, kaynak dosya adıyla C:\MFT\out\Report2009.rpt ile aynı olabilir.

Örnek 3

Aktarım hedefini belirlemek üzere kaynak dosyanın dosya yolunun bir bölümünü kullanmak için `${FilePath}` değişkeni, simge ve ayırıcı belirtilerleriyle birlikte kullanılabilir.

Aşağıdaki örnekte, kaynak dosyanın dosya yolunun

C:\MONITOR\REPORTS\Paris\Report2009.docdeğerine eşit olduğu varsayılır.

Dosyanın hedefini saptamak için kaynak dosya yolunun bir kısmı kullanılabilir.

C:\MONITOR\REPORTS\Paris\Report2009.docdosya yolu örneği kullanılarak, dosya kaynak dosyanın konumuna (bu örnekte Paris) bağlı olarak bir klasöre aktarılacaksa, aşağıdaki işlem yapılabilir:

- Kaynak **Dosya adı** ' nı `${FilePath}` olarak ayarlayın.

- Hedef **Dosya adı** ' nı, her bir konuma ilişkin klasörlerin yerleştirileceği hedef olarak ayarlayın ve dosya yolunun hedef kısmını ve dosya adını ekleyin. Örneğin, bu C:\MFT\out\\${FilePath}\token=-2\separator=\}\}\\${FileName} olabilir; bu, kaynak dosya adıyla C:\MFT\out\Paris\Report2009.doc ile aynı olabilir.

İlgili kavramlar

“MFT kaynak izleyicisi görevlerinin değişken yerine koyma değeri ile uyarlanması” sayfa 255 Etkin bir kaynak izleme programının tetikleme koşulları karşılandığında, tanımlanan görev çağrılır. Aktarma ya da komut görevini her seferinde aynı hedef aracıyla ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımlamasını da değiştirebilirsiniz. Bunu, görev tanımlaması XML ' ine değişken adları ekleyerek yaparsınız. İzleme programı tetikleyici koşullarının karşılandığını ve görev tanımlamasının değişken adlarını içerdiğini saptadığında, değişken adlarını değişken değerleriyle değiştirir ve görevi çağırır.

İlgili görevler

Değişken yerine koyma değeri birden çok dosyanın tek bir dosya adına geçmesine neden olursa ne yapmanız gerekir?

Kuyruğun izlenmesi ve değişken yerine koyma değerinin kullanılması

ftCreateMonitor komutunu kullanarak, bir kuyruğu izleyebilir ve iletileri izlenen kuyruktan bir dosyaya aktarabilirsiniz. İzlenen kuyruktan okunacak ilk iletideki herhangi bir IBM MQ ileti özelliğinin değeri, görev XML tanımlamasında yerine konabilir ve aktarma davranışını tanımlamak için kullanılabilir.

Bu görev hakkında

Bu örnekte, kaynak aracıya QM_VENUS ' a bağlanan AGENT_VENUS adı verilir. AGENT_VENUS tarafından izlenen kuyruğa START_QUEUE adı verilir ve QM_VENUS üzerinde bulunur. Aracı, her 30 dakikada bir kuyruğa yoklama yapar.

Kuyruğa tam bir ileti grubu yazıldığında, izleme görevi ileti grubunu hedef araçlardan birindeki bir dosyaya gönderir; bunların tümü QM_MARS kuyruk yöneticisine bağlanır. İleti grubunun aktarıldığı dosyanın adı, gruptaki ilk iletide IBM MQ ileti özelliği `usr.fileName` tarafından tanımlanır. İleti grubunun gönderildiği aracının adı, gruptaki ilk iletide IBM MQ ileti özelliği `usr.toAgent` tarafından tanımlanır. `usr.toAgent` üstbilgisi ayarlanmazsa, hedef aracı için kullanılacak varsayılan değer AGENT_MAGENTA olur.

`useGroups="true"` değerini belirlediğinizde, `groupId="${GROUPID}"` değerini de belirlemezseniz, aktarma kuyruktaki ilk iletiyi alır. Örneğin, `fileNamedosyasını` oluşturmak için değişken yerine koyma değeri kullanıyorsanız, `a.txt` içeriğinin doğru olmaması olasıdır. Bunun nedeni, `fileName` adlı dosyanın izleme programı tarafından oluşturulduğu, ancak aktarma işleminin gerçekte `fileName` adlı dosyayı oluşturmaması gereken bir iletiyi aldığı içindir.

Yordam

1. İzleme programının tetiklendiğinde gerçekleştirdiği görevi tanımlayan görev XML dosyasını yaratın.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="${toAgent}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/${fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

```
</item>
</transferSet>
</managedTransfer>
</request>
```

IBM MQ ileti üstbilgilerinin değerleriyle değiştirilen değişkenler **koyu** olarak vurgulanır. Bu görev XML ' i /home/USER1/task.xml dosyasına kaydedildi

2. START_QUEUE kuyruğunu izlemek için bir kaynak izleme programı yaratın.

Şu komutu verin:

```
fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA
```

3. Bir kullanıcı ya da program, bir grup iletiyi START_QUEUE kuyruğuna yazar.

Bu gruptaki ilk iletide aşağıdaki IBM MQ ileti özellikleri ayarlanmıştır:

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. İzleme programı, tam grup yazıldığında tetiklenir. Aracı, IBM MQ ileti özelliklerini görev XML ' ine değiştirir.

Bu, görev XML ' inin aşağıdakilere dönüştürülmesine neden oluyor:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/larmer.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Sonuçlar

Görev XML ' i tarafından tanımlanan aktarma gerçekleştirilir. AGENT_VENUS tarafından START_QUEUE 'dan okunan iletilerin tam grubu, AGENT_VIOLET' in çalıştığı sistemde /reports/larmer.rpt adlı bir dosyaya yazılır.

Sonraki adım

Her iletinin ayrı bir dosyaya aktarılması

Bir kuyruğu izlemek ve her iletinin ayrı bir dosyaya aktarılmasını istiyorsanız, bu konuda daha önce anlatılana benzer bir teknik kullanabilirsiniz.

1. **fteCreateMonitor** komutunda **-tr completeGroups** değiştirgesini belirterek, izleme programını daha önce açıklandığı gibi yaratın.
2. Görev XML ' inde aşağıdakileri belirtin:

```
<queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
```

Ancak, iletileri kaynak kuyruğa koyduğunuzda, bunları bir IBM MQ grubuna koymayın. Her iletiye IBM MQ ileti özellikleri ekleyin. Örneğin, `usr.filename` özelliğini her ileti için benzersiz bir dosya adı değeriyle belirtin. Bu, Managed File Transfer Agent 'in kaynak kuyruktaki her iletiyi ayrı bir grup olarak kabul etmesine neden olur.

İlgili kavramlar

[“İletilerden dosyalara veri aktarılması” sayfa 291](#)

Managed File Transfer iletimden dosyaya özelliği, IBM MQ kuyruğundaki bir ya da daha çok iletimdosyaya, veri kümesine (z/OSüzerindeki) ya da kullanıcı dosya alanına veri aktarmanızı sağlar. IBM MQ iletilerini yaratan ya da işleyen bir uygulamanız varsa, bu iletileri Managed File Transfer ağınzdaki herhangi bir sistemde bulunan bir dosyaya aktarmak için Managed File Transfer iletimden dosyaya aktarma yeteneğini kullanabilirsiniz.

[“MFT kaynak izleyicisi görevlerinin değişken yerine koyma değeri ile uyarlanması” sayfa 255](#)

Etkin bir kaynak izleme programının tetikleme koşulları karşılandığında, tanımlanan görev çağırılır. Aktarma ya da komut görevini her seferinde aynı hedef aracıyla ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımlamasını da değiştirebilirsiniz. Bunu, görev tanımlaması XML 'ine değişken adları ekleyerek yaparsınız. İzleme programı tetikleyici koşullarının karşılandığını ve görev tanımlamasının değişken adlarını içerdiğini saptadığında, değişken adlarını değişken değerleriyle değiştirir ve görevi çağırır.

[Bir kuyruk kaynak izleme programı tarafından başlatılan bir aktarma tarafından yaratılan hedef dosyalar yanlış veri içeriyorsa ne yapmanız gerekir?](#)

İlgili görevler

[“Komutları ve komut dosyalarını başlatmak için MFT izleme görevlerinin yapılandırılması” sayfa 249](#)

Kaynak izleme programları, ilişkili görev olarak dosya aktarımlarını gerçekleştirmekle sınırlı değildir. İzleme programını, yürütülebilir programlar, Ant komut dosyaları ya da JCL işleri de içinde olmak üzere, izleme aracısından diğer komutları çağırarak üzere de yapılandırabilirsiniz. Komutları çağırarak için, izleme görevi tanımlaması XML dosyasını, bağımsız değişkenler ve özellikler gibi ilgili komut çağırısı değiştirgelerine sahip bir ya da daha çok komut ögesi içerecek şekilde düzenleyin.

[“Örnek: MFT kaynağını yapılandırma” sayfa 255](#)

Bir kaynak izleme programı tarafından izlenecek kaynak olarak IBM MQ kuyruğunu, **fteCreateMonitor** komutuyla **-mq** değiştirgesini kullanarak belirtebilirsiniz.

İlgili başvurular

[**fteCreateMonitor**: MFT kaynak izleme programı yaratılması](#)

[Kaynak kuyruklardaki iletilerden MFT tarafından okunan MQ ileti özellikleri](#)

İletiler arası aktarımlar için izleyici yeniden deneme davranışını yapılandırma

Bir kaynak izleme programı tarafından tetiklenen bir iletimden dosyaya aktarma işlemi başarısız olursa ve izleme programını kuyrukta tetikleyen ileti grubunu bırakırsa, bu aktarma işlemi sonraki yoklama aralıklarında yeniden sunulur. Aktarmanın yeniden gönderilme sayısı, izleme aracısının **monitorGroupRetryLimit** özelliğiyle sınırlıdır.

Bu görev hakkında

Yeni bir iletimden dosyaya aktarma işlemi tetiklendiğinde, aktarma görevi için yeni bir aktarma tanıtıcısı oluşturulur.

Aracı yeniden başlatılırsa, aktarma işleminin tetiklenme sayısı `agent.properties` dosyasındaki **monitorGroupRetryLimit** değerini aşmış olsa da, izleme programı yeniden aktarma işlemi tetikler. **monitorGroupRetryLimit** özelliğinin değeri, ileti grubu kuyrukta bulunmaya devam ederse, izleme programının iletimden dosyaya aktarma işlemi yeniden tetikleme sayısı üst sınırdır. Bu özelliğin varsayılan değeri 10 'dur. Bu özelliğin değeri herhangi bir pozitif tamsayı değerine ya da -1 değerine ayarlanabilir. Bu

"zellik için -1 değeri belirlenirse, izleme programı, tetikleme durumu yerine getirilinceye kadar, aktarma işlemini sresiz sayıyla yeniden tetikler.

Bir aktarma girişimi, aktarma işleminin tetiklenme sayısının **monitorGroupRetryLimit** değerini aşmasına neden olursa, aracı olay günlüğüne bir hata yazar.

Tek bir ileti tek bir grupmuş gibi işlenir ve ileti kuyrukta kalırken her yoklama aralığında aktarma yeniden tetiklenir ve aktarma işleminin tetiklenme sayısı **monitorGroupRetryLimit** değerinden az olur.

İzleme aracısında **monitorGroupRetryLimit** özelliğini ayarlamak için aşağıdaki adımları izleyin:

Yordam

1. **fteStopAgent** komutunu kullanarak izleme aracısını durdurun.
2. İzleme aracısına ilişkin agent.properties dosyasını aşağıdaki satırı içerecek şekilde düzenleyin:

```
monitorGroupRetryLimit=number_of_retries
```

agent.properties dosyası *MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/monitoring_agent_namedizininde* bulunur.

3. **fteStartAgent** komutunu kullanarak izleme aracısını başlatın.

İlgili görevler

“Örnek: MFT kaynağını yapılandırma” sayfa 255

Bir kaynak izleme programı tarafından izlenecek kaynak olarak IBM MQ kuyruğunu, **fteCreateMonitor** komutuyla **-mq** deęiştirgesini kullanarak belirtebilirsiniz.

Tetikleyici dosyası kullanılması

Kaynak izleme programındaki bir tetikleyici dosyasının içeriğini kullanarak, tek bir aktarma isteğinde aktarılacak bir dosya kümesi tanımlayabilirsiniz. Eşleşen bir tetikleyici dosyası her saptandığında, içeriği kaynak dosya yolları ve isteğe bağlı olarak hedef dosya yolları için ayrıştırılır. Bu dosya yolları, belirlediğiniz görev aktarma XML dosyasındaki dosya öğelerini tanımlamak için kullanılır ve aracıya tek aktarma isteği olarak sunulur. Kaynak izleme programının tanımlaması, tetikleyici içeriğinin etkinleştirilip etkinleştirilmediğini belirler.

-tc (tetikleyici içeriği) deęiştirgesini belirterek bir izleme programı yarattığınızda dosya içeriği tetikleme etkinleştirilebilir. Bu **-tc** parametresi yalnızca match ve noSizeChangedosya tetikleme seçenekleri için geçerlidir. İzleme programı yaratılmasıyla ilgili ek bilgi için bkz. **fteCreateMonitor**: MFT kaynak izleme programı yaratılması.

Tetikleyici içerik dosyası kullanılırken, her satırın varsayılan biçimi şöyledir:

- Tek bir kaynak dosya yolu ya da
- Virgülle ayrılmış bir kaynak dosya yolu ve hedef dosya yolu

burada beyaz alan karakterleri, dosya yollarının bir parçası olarak işlenir. **fteCreateMonitor** komutunda **-tcx** ve **-tcc** parametrelerini belirterek varsayılan satır biçimini deęiştirebilirsiniz. Daha fazla bilgi için bkz “Gelişmiş seçenekler” sayfa 265.

Bir tetikleyici dosyası ayrıştırıldıktan sonra, dosya yolları listesi oluşturulur ve belirttiğiniz aktarma görevi XML ' ine uygulanır. Tüm izleme programlarında olduğu gibi, aktarma görevi XML dosyası, **fteCreateTransfer** komutu tarafından tek bir öğe ya da dosya tanımlı olarak oluşturulan tam bir aktarma görevi XML dosyasıdır. Tek öğe, kaynak ve hedef dosya yolları için yerine koyma deęişkenleri olarak $\{contentSource\}$ ve isteğe bağlı olarak $\{contentDestination\}$ yerine koyma deęişkenlerini kullanmalıdır. İzleme programı, aktarma görevi XML dosyasını, tetikleyici dosyasındaki her bir satır (dosya yolu) için bir dosya öğesi içerecek şekilde genişletir.

-tc parametresi her tetikleyici dosyası için bir aktarma isteği belirttiğinden, **-bs** parametresiyle dosya içeriği tetikleme kullanamazsınız.

Örnek

Aşağıdaki örnek, `trig` ile biten ve o dosyadaki dosya yollarını okuyan bir dosyada tetiklenecek bir izleme programını tanımlar.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
```

fteCreateTransfer komutu, kaynak dosya yolu `${contentSource}` olan tek bir dosya için `task.xml` adlı bir dosya oluşturur. Örneğin:

```
<item checksumMethod="MD5" mode="binary">
  <source disposition="leave" recursive="false">
    <file>${contentSource}</file>
  </source>
</item>
```

fteCreateMonitor komutu, `/home/trigdir` dizininde `trig` ile biten dosyaları tarar ve o tetikleyici dosyasındaki tüm yollar için `task.xml` 'e dayalı tek bir aktarma isteği yaratmak üzere içeriği kullanır. Tetikleyici dosyasının biçimi, virgül ayırıcı olmadan her satırda bir dosya yolu (yalnızca kaynak) olmalıdır. Örneğin:

```
/home/file/first.txt
/home/file/second.txt
/home/different/third.txt
:
```

Tüm dosyalar, dosya adı ile birlikte `/file/destdir` dizinine teslim edilir, dosya yolu ile teslim edilmez; yani, `/home/file/first.txt`, `/file/destdir/first.txt`'e teslim edilir.

Diğer bir seçenek olarak, **fteCreateTransfer** komutundaki **-dd /file/destdir** deyişirgesini **-df \${contentDestination}** olarak ve bir tetikleyici dosyasının içeriğinin biçimini *kaynak dosya yolu,hedef dosya yolu* olarak deyiştirirseniz, aynı hedef aracı için farklı hedef yollar tanımlayabilirsiniz. Örneğin:

```
/home/file/first.txt,/home/other/sixth.txt
```

Daha sonra hedef konum `/home/other/sixth.txt` olur.

Yerine koyma deyişkenleri bölümlenebilir. Örneğin, dosya adı kısmını sağlanan yoldan `${contentDestination{token=-1}}` kullanarak ayırabilirsiniz. Bu nedenle, **fteCreateTransfer** hedefi **-df /file/destdir/\${contentDestination{token=-1}}** olarak tanımlanırsa, `/home/file/first.txt` için yeni hedef `/file/destdir/sixth.txt` olur.

Gelişmiş seçenekler

-tcx *regex* parametresini kullanarak tetikleyici dosyasının içeriğine ilişkin varsayılan satır biçimini deyiştirebilirsiniz. Gerekli satır biçimiyle eşleşen ve bir ya da iki yakalama grubu sağlayan bir düzenli ifade sağlayın. İlk yakalama grubu kaynak ve ikinci, isteğe bağlı, yakalama grubu hedef. Örneğin:

- Kaynak ve hedef yol kısa çizgi ile ayrılır:

```
((?:[^-])+) - ((?:[^-])+)
```

Bu örnekte, ayırıcı üç konumda tanımlanır ve tire işaretinin üç örneği de (-) herhangi bir karaktere dönüştürülebilir. Özel karakterlerden kaçtığınızdan emin olun.

- Kaynak ve hedef yollar, sondaki boşluklarla virgülle ayrılır. Bir sayı işaretiyle (#) gösterilen açıklamalar yoksayılır.

```
((?:[^\,])+), ((?:[^\,])+) *(?:#\.[*])+
```

Dosya yolları sayı işareti (#) içeremez. Genellikle bir girdi aşağıdaki gibidir: /home/source/from.txt,/home/destination/to.txt # some comment.

-**tc**r değiştirgesini kullanırsanız, ifadenin hataları saptayabilmesi ve tetikleyici dosyalarını doğru bir şekilde ayrıştırabilmesi için düzenli ifadenin iyi tasarlanmış ve sınanmış olduğundan emin olun.

-**tcc destSrc** parametresini kullanarak yakalama sırasını tersine çevirebilirsiniz. Bu parametreyi belirtirseniz, ilk yakalama grubu hedef dosya yolu, ikinci grup kaynak dosya yoludur.

Hataların nasıl işlendiğini

Boş tetikleyici dosyası

Tetikleyici dosyası boşsa, sonuç dosya aktarımı değildir. Yani, izleme programı bir aktarma isteği yaratır, ancak dosya ögesi belirlenmez.

Hata içeren tetikleyici dosyası

Bir tetikleyici dosyasındaki herhangi bir giriş beklenen biçime göre ayrıştırılamazsa, aktarma isteği oluşturulmaz. Bir izleyici hata günlüğü yayınlanır ve hata olay günlüğüne de kaydedilir. Tetikleyici dosya işlendi olarak işaretlenir ve dosya güncelleninceye kadar izleme programı dosyayı yeniden işlemeyi denemez.

Uyumsuz aktarma görevi XML 'i

Aktarma görevi XML 'i tetikleyici dosyasıyla eşleşmelidir; yani, aktarma görevi XML' inde hem \$ {contentSource} , hem de \${contentDestination} varsa, bu izleme programına ilişkin tüm tetikleyici dosyalarının kaynak ve hedef dosya yolları ve tersi için benzer şekilde olması gerekir. İlk durumda, tetikleyici dosyası yalnızca kaynak dosya yolunu sağladıysa, izleme programı \${contentDestination} hedefine ilişkin bir yerine koyma hatası bildirir.

Örnekler

Aşağıdaki örnek, bir tetikleyici dosyasının içeriğinin yalnızca kaynak dosya yoluna sahip olduğu temel bir içerik tetikleyicidir:

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*trig"
-tc
```

-**tc**r parametresi, bir boşluk karakteriyle ayrılmış herhangi bir karakter dizisinin iki yakalama grubunu tanımlar. -**tcc destSrc** parametresi ve seçeneği, yakalama gruplarının hedef olarak daha sonra kaynak olarak işleneceğini belirtir.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -df ${contentDestination} $
{contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*trig"
-tc
-tcr "((?:[^\ ])+) ((?:[^\ ])+)" -tcc destSrc
```

İlgili kavramlar

“MFT kaynak izleyicisi görevlerinin değişken yerine koyma değeri ile uyarlanması” sayfa 255

Etkin bir kaynak izleme programının tetikleme koşulları karşılandığında, tanımlanan görev çağrılır.

Aktarma ya da komut görevini her seferinde aynı hedef aracıyla ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımlamasını da değiştirebilirsiniz. Bunu, görev tanımlaması XML 'ine değişken adları ekleyerek yaparsınız. İzleme programı tetikleyici koşullarının karşılandığını ve görev tanımlamasının değişken adlarını içerdiğini saptadığında, değişken adlarını değişken değerleriyle değiştirir ve görevi çağırır.

İlgili görevler

“Kuyruğun izlenmesi ve değişken yerine koyma değerinin kullanılması” sayfa 261

fteCreateMonitor komutunu kullanarak, bir kuyruğu izleyebilir ve iletileri izlenen kuyruktan bir dosyaya aktarabilirsiniz. İzlenen kuyruktan okunacak ilk iletideki herhangi bir IBM MQ ileti özelliğinin değeri, görev XML tanımlamasında yerine konabilir ve aktarma davranışını tanımlamak için kullanılabilir.

İlgili başvurular

fteCreateMonitor: MFT kaynak izleme programı yaratılması

fteCreateTransfer: yeni bir dosya aktarımı başlatır

MFT kaynak izleyicilerinin günlüğe kaydedilmesi

Günlük kaydını kullanarak kaynak izleyicileri hakkında tanılama bilgileri edinebilirsiniz.

Bu görev hakkında

Kaynak izleyicisi günlük kaydını denetlemek için **fteSetAgentLogLevel** komutunu ya da `agent.properties` dosyasını kullanarak kaynak izleyicisinin günlük kaydını kullanabilirsiniz.

Bilgileri yakalamak için var olan izleme noktalarının kullanılmaya devam ettiğine dikkat edin.

Kaynak izleyicisi günlükleri `resmoneventN.log` adlı bir dosyaya yazılır; burada *N* bir sayıyı gösterir; örneğin, `resmonevent0.log`. Olay günlüğü dosyaları, bir izleme programı bir kaynağı (örneğin, bir dizin ya da kuyruk) yokladığında gerçekleşen bazı işlemleri kaydeder.



Uyarı: Bir aracının tüm kaynak izleyicileri aynı günlük dosyasına yazar.

Örneğin, bir `resmoneventN.log` dosyasının çıktısı için bkz. [MFT dizin kaynağı izleme programınız dosyaları tetikle miyorsa ne yapmanız gerekir](#).

Aşağıdaki çizelge, kaynak izleyicinin günlük dosyasına yazdığı olayların tipini listeler. Üçüncü kolon, en düşük düzeyin INFO ve en yüksek düzeyin VERBOSE olduğu her olayı yakalamak için gereken günlük düzeyini açıklar.

Daha yüksek bir günlük düzeyi ayarlandığında, alt düzey olayları da yazacağını unutmayın. Örneğin, günlük düzeyi MODERATE olarak ayarlandığında, VERBOSE düzeyi olaylar değil, INFO düzeyi olaylar da yazılır.

Numara	Olay	Günlük düzeyi	Açıklama
1	İzleme Programı Yaratıldı	BİLGİ	Bir kaynak izleme programı yaratıldı.
2	İzleme Programı Silindi	BİLGİ	Bir kaynak izleyicisi silindi.
3	İzleme Programı Durduruldu	BİLGİ	Bir kaynak izleme programı durduruldu.
4	İzleme Programı Başlatıldı	BİLGİ	Bir kaynak izleme programı başlatıldı.
5	Yoklamayı Başlat	BİLGİ	Bir kaynak izleme programı yeni bir yoklama döngüsü başlattı.
6	Yoklamayı Sona Erdir	BİLGİ	Kaynak izleyici yoklama döngüsü sona erdi.
7	Örüntü Eşleşmesi	Ayrıntılı	Tetikleyici izleme dizininde ya da kuyrukta belirtilen kalıpla eşleşen bir ileti bulundu.
8	Örüntü Uyuşmazlığı	Ayrıntılı	Tetikleyici izleme dizininde eşleşmeyen bir dosya ya da kuyrukta belirtilen kalıpla eşleşmeyen bir ileti bulundu.
9	Aktarma İsteği	BİLGİ	Kaynak izleme programı bir aktarma başlattı.
10	Dizin çok derin	Ayrıntılı	Kaynak izleme programı tarafından izlenen dizin, yoklamak için kaynak izleme programı konfigürasyonunda belirtilen sayıdan daha fazla alt dizin içeriyor.
11	Dosya kilitli	orta düzeyde	Kaynak izleme programı tarafından izlenen tetikleyici dosyası başka bir işlem tarafından kilitlendi.

Numara	Olay	Günlük düzeyi	Açıklama
12	Dosya boyutu küçük	orta düzeyde	Tetikleyici dosyası, kaynak izleyicisi yapılandırılmasında belirtilen büyüklükten küçük.
13	Dosya boyutu dengesiz	orta düzeyde	Tetikleyici dosyası, kaynak izleyicisi konfigürasyonunun beklediğinden daha sık değiştiriliyor.
14	Çok fazla yoklama	orta düzeyde	Bir kaynak izleyicisi, dengesiz bir tetikleyici dosyasını çok fazla yokladı.
15.000	Eşleşen Öğeler	BİLGİ	Kaynak izleme programı tarafından yoklanan dizinde bulunan tetikleyici dosyalarının toplam sayısı.
16	Öğeleri Aktar	BİLGİ	Aktarma isteğindeki toplam öğe sayısı.
17	FDC Oluşturuldu	orta düzeyde	Bir kaynak izleyicisi kural dışı durum üretti.
18	Aktarma İsteği	BİLGİ	Aktarım isteği kaynak izleme programı tarafından sunuldu.
19	İzleme Programının Başlatılması Başarısız Oldu	orta düzeyde	Kaynak izleyicisi başlatılmadı.
20	Geçmiş Temizlendi	BİLGİ	İzleme geçmişi bilgileri temizlendi.
21	İzleme Geçmişini Temizleme Başarısız Oldu	BİLGİ	İzleme geçmişi bilgilerini temizleme girişimi başarısız oldu.
22	Aktarım Tanıtıcısı	BİLGİ	Aktarma isteğinin tanıtıcısı izleme programı tarafından gönderildi.
23	Toplu İş	BİLGİ	Eşleşen öğeler için toplam aktarma isteği sayısı: N ; burada N bir sayıdır.
V 9.3.2 24	Bağlantılı	Ayrıntılı	Bir kaynak izleyicisi aracı kuyruk yöneticisine bağlandı.
V 9.3.2 25	Bağlantısız	Ayrıntılı	Bir kaynak izleme programının aracı kuyruk yöneticisiyle bağlantısı kesildi.
V 9.3.2 26	Bağlantı kesilirken hata oluştu	Ayrıntılı	Bir kaynak izleyicisi, aracı kuyruk yöneticisiyle bağlantısını keserken bir sorunla karşılaştı.

Yordam

- Kaynak izleyicisi günlük kaydını açmak ve kapatmak üzere **fteSetAgentLogLevel** komutunu kullanmak için, **logMonitor** değiştirgesinin açıklaması ve farklı seçenekleri nasıl kullandığına ilişkin örnekler için [fteSetAgentLogLevel](#) başlıklı konuya bakın.
- Kaynak izleyici günlük kaydını denetlemek üzere `agent.properties` dosyasını kullanmak için aşağıdaki günlüğe kaydetme etkinliklerini gerçekleştirmenize olanak tanıyan ek özelliklerin açıklaması için `MFT agent.properties` dosyası konusuna bakın:
 - Günlük kaydını aç ya da kapat
 - Her günlük dosyasının boyutunu sınırla
 - Kaynak izleyicilerinin oluşturabileceği günlük sayısını sınırla

Örnek

Aşağıdaki örnek ileti kümeleri verbose level logging for agent HA2, on queue manager MFTDEMO:

```
<?xml version="1.0"?>
<log:log version="6.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:log="https://www.ibm.com/log">
  <log:originator>
    <log:request>
      <log:hostName>192.168.7.1</log:hostName>
      <log:userID>johndoe</log:userID>
    </log:request>
  </log:originator>
  <log:endpoint agent="HA2" QMgr="MFTDEMO"/>
  <log:logMonitor>MON1="verbose"</log:logMonitor>
</log:log>
```

İlgili başvurular

[fteSetAgentLogLevel](#) komutu

[MFT agent.properties](#) dosyası

V 9.3.0 MFT Kaynak İzleme Programının Başlatılması

IBM MQ 9.3.0' den **fteStartMonitor** komutunu kullanarak bir aracıyı durdurmaya ya da yeniden başlatmaya gerek kalmadan kaynak izleyicilerini başlatabilirsiniz.

Başlamadan önce

`agent.properties` dosyasında **authorityChecking** özneliği true olarak ayarlanarak kullanıcı yetki yönetimi etkinleştirildiyse, kaynak izleme programını başlatmak için İzleme ya da İzleme işlemleri yetkinizin olması gerekir. Kullanıcı yetkisi yönetimiyle ilgili daha fazla bilgi için [MFT aracı işlemlerinde kullanıcı yetkilerinin kısıtlanmasında](#) başlıklı konuya bakın.

Bu görev hakkında

fteStartMonitor komutunu, Managed File Transfer komutları bileşeninin kurulu olduğu herhangi bir sistemden çalıştırabilirsiniz; bu, kaynak izleme programını herhangi bir yerden başlatabileceğiniz ve kaynak izleme programına sahip olan aracının çalıştığı sistemle sınırlı olmadığı anlamına gelir. Bu komuta ilişkin gerekli ve isteğe bağlı parametrelere ilişkin bilgi için [fteStartMonitor \(MFT kaynak izleme programının başlatılması\)](#) başlıklı konuya bakın.

Yordam

- **fteStartMonitor** komutu çalıştırılmadan önce ya da çalıştırdıktan sonra bir aracının durumunu öğrenmek için, aşağıdaki örnekte gösterildiği gibi **-v** parametresiyle birlikte **fteListMonitors** komutunu kullanın:

```
fteListMonitors -ma monitoring_agent_name -v
```

- Aynı makinede çalışan bir aracıda kaynak izleyici başlatmak için **fteStartMonitor** komutunu aşağıdaki gibi girin:

```
fteStartMonitor -mn monitor_name -ma agent_name
```

- Farklı bir makinede çalışan bir aracıda kaynak izleyici başlatmak için **fteStartMonitor** komutunu aşağıdaki gibi girin:

```
fteStartMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```

Komut kuyruğu yöneticisi izleme aracı için de aracı kuyruk yöneticisiyse, **-mm** deęiřtirgesi isteęe baęlıdır; tersi durumda, aracı kuyruk yöneticisini **-mm** deęiřtirgesiyle belirtmeniz gerekir.

Sonuçlar

Aracı çalışıyorsa, kaynak izleme programı durdurulmuşsa başlatılır. Komut, aşağıdaki iletileri çıkarır ve aracının output0.log içindeki bir olayı günlüğe kaydeder.

```
BFGCL0816I: 'agent_name' aracısının 'monitor_name' kaynak izleyicisini başlatma isteği yayınlandı.  
BFGCL0251I: İstek başarıyla tamamlandı.
```

Kaynak izleme programını başlatamazsa komutun çıkışına ilişkin bilgi için [fteStartMonitor](#) (MFT kaynak izleme programının başlatılması) başlıklı konuya bakın.

İlgili kavramlar

“MFT kaynak izleme kavramları” sayfa 245

Managed File Transfer kaynak izleme özelliğine ilişkin temel kavramlara genel bakış.

İlgili görevler

“MFT Kaynak İzleme Programının Durdurulması” sayfa 270

IBM MQ 9.3.0' den **fteStopMonitor** komutunu kullanarak bir aracıyı durdurmaya ya da yeniden başlatmaya gerek kalmadan kaynak izleyicilerini durdurabilirsiniz.

İlgili başvurular

[fteStartİzleme Programı](#) (MFT kaynak izleme programının başlatılması)

V 9.3.0 MFT Kaynak İzleme Programının Durdurulması

IBM MQ 9.3.0' den **fteStopMonitor** komutunu kullanarak bir aracıyı durdurmaya ya da yeniden başlatmaya gerek kalmadan kaynak izleyicilerini durdurabilirsiniz.

Başlamadan önce

`agent.properties` dosyasında **authorityChecking** özneliği `true` değerine ayarlanarak kullanıcı yetki yönetimi etkinleştirildiyse, bir kaynak izleme programını durdurmak için İzleme ya da İzleme işlemleri yetkinizin olması gerekir. Kullanıcı yetkisi yönetimiyle ilgili daha fazla bilgi için [MFT aracı işlemlerinde kullanıcı yetkilerinin kısıtlanması](#) başlıklı konuya bakın.

Bu görev hakkında

fteStopMonitor komutunu, Managed File Transfer komutları bileşeninin kurulu olduğu herhangi bir sistemden çalıştırabilirsiniz; bu, bir kaynak izleme programını herhangi bir yerden durdurabileceğiniz anlamına gelir ve kaynak izleme programına sahip olan aracının çalıştığı sistemle sınırlı değildir. Bu komuta ilişkin gerekli ve isteğe bağlı değiştirgelerle ilgili bilgi için [fteStopMonitor](#) (MFT kaynak izleme programının durdurulması) başlıklı konuya bakın.

Bir kaynak izleyicisi durdurulduğunda, aracının kaynak izleyicisi olay günlüğüne (`resmoneventnumber.log`) bir ileti yazar. Kaynak izleme programı **fteStopMonitor** komutuyla durdurulursa, ileti durdurma isteğini yayınlayan kullanıcının adını içerir:

```
Kaynak İzleyicisi '< mquser_id>' adlı kullanıcı tarafından durduruldu
```

Kaynak izleme programı **fteStopMonitor** komutu kullanılarak önceden durdurulmuş olsa da, aracı yeniden başlatılırsa, kaynak izleme programı otomatik olarak başlatılır.

Aracılar, izleme isteklerini koşut yerine dizeysel olarak durdurur; örneğin, bir aracı M1 izleme programını durdurma isteği alırsa ve ardından hızlı bir şekilde M2 izleme programını durdurma isteği alırsa, önce M2' yi durdurmaya çalışmadan önce M1 durur.

Yordam

- **fteStopMonitor** komutu çalıştırılmadan önce ya da çalıştırıldıktan sonra bir aracının durumunu öğrenmek için, aşağıdaki örnekte gösterildiği gibi **-v** parametresiyle birlikte **fteListMonitors** komutunu kullanın:

```
fteListMonitors -ma monitoring_agent_name -v
```

- Bir kaynak izleyicisini aynı makinede çalışan bir araçıda durdurmak için **fteStopMonitor** komutunu aşağıdaki gibi girin:

```
fteStopMonitor -mn monitor_name -ma agent_name
```

- Farklı bir makinede çalışan bir araçıda kaynak izleyicisini durdurmak için **fteStopMonitor** komutunu aşağıdaki gibi girin:

```
fteStopMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```

Komut kuyruğu yöneticisi izleme aracı için de aracı kuyruk yöneticisiyse, **-mm** değiştirgesi isteğe bağlıdır; tersi durumda, aracı kuyruk yöneticisini **-mm** değiştirgesiyle belirtmeniz gerekir.

Sonuçlar

Aracı çalışıyorsa, kaynak izleme programı başlatılmış durumdaysa durdurulur. Komut, aşağıdaki iletileri çıkarır ve aracının output0.log içindeki bir olayı günlüğe kaydeder.

```
BFGCL0813I: 'SOURCE' aracısına ilişkin 'MNTR' kaynak izleyicisini durdurma isteği yayınlandı.  
BFGCL0251I: İstek başarıyla tamamlandı.
```

Kaynak izleme programını durduramazsa, komutun çıkışına ilişkin bilgi için [fteStopMonitor \(MFT kaynak izleme programının durdurulması\)](#) başlıklı konuya bakın.

İlgili kavramlar

“MFT kaynak izleme kavramları” sayfa 245

Managed File Transfer kaynak izleme özelliğine ilişkin temel kavramlara genel bakış.

İlgili görevler

“MFT Kaynak İzleme Programının Başlatılması” sayfa 269

IBM MQ 9.3.0' den **fteStartMonitor** komutunu kullanarak bir aracıyı durdurmaya ya da yeniden başlatmaya gerek kalmadan kaynak izleyicilerini başlatabilirsiniz.

İlgili başvurular

[fteStopİzleyici \(MFT kaynak izleyicisinin durdurulması\)](#)

MFT kaynak izleyicilerini yedekleme ve geri yükleme

Tanımlamalarını, yedekten yeni bir kaynak izleyicisi yaratmak üzere içe aktarabileceğiniz bir XML dosyasına aktararak, ileride kullanmak üzere kullanılabilir olmasını istediğiniz kaynak izleyicilerini yedekleyebilirsiniz.

Bu görev hakkında

Daha önce tanımladığınız kaynak izleyicilerini yedeklemeniz gerekebilir; böylece, ileride bunların tanımlamalarını yeniden kullanabilirsiniz (örneğin, kaynak izleyicilerini farklı bir altyapıda yeniden yaratmak için ya da kuyruk yöneticisi sorunları nedeniyle bir kaynak izleyicisinin yeniden yaratılması gerekiyorsa).

fteCreateMonitor komutunu ya da **fteListMonitors** komutunu **-ox** parametresiyle kullanarak tek bir kaynak yöneticisi tanımlamasını yedekleyebilirsiniz. Her iki durumda da, kaynak yöneticisi tanımlaması bir XML dosyasına (dışa) aktarılarak yedeklenir. Daha sonra, tanımlamayı XML dosyasından içe aktararak yeni bir kaynak yöneticisi yaratmak için **fteCreateMonitor** komutunun **-ix** değiştirgesini kullanabilirsiniz.

-ox değiştirgesiyle, bir kerede tek bir kaynak izleyicisi tanımlamasını yedekleyebilirsiniz.

IBM MQ 9.1' dan, **-od** parametresi **fteListMonitors** komutuna eklenir. Bu değiştirgeyi belirterek, tanımlamalarını toplu olarak belirtilen bir dizine aktararak, aynı anda birden çok kaynak izleme programını yedekleyebilirsiniz. Her kaynak izleyicisi tanımlaması, adı *agent_name.monitor_name.xml* biçiminde olan ayrı bir XML dosyasına saklanır.

-od parametresi özellikle, her kaynak tanımlaması için **ftelistmonitors -ox** komutunu ayrı olarak çalıştırmak yerine **ftelistmonitors -od** komutunu yalnızca bir kez çalıştırmanız gerektiğinden yedeklemek istediğiniz çok sayıda kaynak izleyicisi varsa yararlıdır ya da her kaynak izleyicisi için **ftelistmonitors -ox** komutunu çalıştırmak üzere ayrı bir komut dosyası kullanın.

Yordam

- Bir kaynak izleyicisinin tanımlamasını bir XML dosyasına aktararak yedeklemek için aşağıdaki komutlardan birini kullanın:

- **-ox** parametresiyle **ftelistmonitors** komutu.
- **-od** parametresiyle **ftelistmonitors** komutu.

-ox deęiřtirgesini kullanırken, aşağıdaki örnekte gösterildięi gibi **-ma** ve **-mn** deęiřtirgelerini de belirtmeniz gerekir:

```
ftelistmonitors -ma AGENT1 -mn MONITOR1 -ox filename1.xml
```

- Birden çok kaynak izleyicisi tanımlamasını belirli bir dizindeki XML dosyalarına aktararak yedeklemek için, aşağıdaki örnekte gösterildięi gibi **ftelistmonitors** komutunu **-od** deęiřtirgesiyle birlikte kullanın:

```
ftelistmonitors -od /usr/mft/resmonbackup
```

Kaynak izleyicilerini toplu olarak yedeklerken geçerli bir hedef dizin belirtmelisiniz. Hedef yol belirtilmemesi, aşağıdaki örnekte gösterildięi gibi bir hata iletilmesiyle sonuçlanır:

BFGCL0762E: Çıkış dizini belirtilmedi. Geçerli bir yol belirterek komutu yeniden çalıştırın.

-od parametresi **-ox** parametresiyle birleştirilmemelidir; tersi durumda, aşağıdaki hata iletilmesi görüntülenir:

BFGCL0761E: Hem '-od', hem de '-ox' deęiřtirgelerini birlikte belirtmek geçerli deęildir.

Yedeğin içereceęi belirli bir kaynak izleme programı kümesi tanımlayabilirsiniz. Örneğin, bir aracının adını belirtmek için **-ma** deęiřtirgesini kullanarak, aşağıdaki örnekte gösterildięi gibi, o aracıya ilişkin tüm kaynak izleyicilerini yedekleyebilirsiniz:

```
ftelistmonitors -ma AGENT1 -od /usr/mft/resmonbackup
```

Aracı adlarını ve/veya izleme programı adlarını eşleřtirmek için kullanılacak bir kalıp tanımlarken yıldız iřareti (*) karakteri ekleyerek genel arama karakteri eşleřtirmesini de kullanabilirsiniz. Aşağıdaki örnek, belirli bir kalıpla eşleřen ve adı belirli bir kalıpla eşleřen tüm kaynak izleyicilerini yedekler:

```
ftelistmonitors -ma AGENT* -mn MON* -od /usr/mft/resmonbackup
```

Komut çalışırken, aşağıdaki ilerleme raporu iletilerini görüntüler:

Toplam *sayı* eşleřen kaynak izleyicisi tanımlaması bulundu.

Dosya sistemine saklanan *sayı* kaynak izleme programı tanımlamasının *dizini* .

Ayrıntılı seçeneęi kullanıyorsanız, çalışan toplam görüntülenmeye devam etmektedir, ancak görüntülenmek yerine

index / number kaynak izleme programı tanımlaması dosya sistemine saklandı

Komut, saklanmakta olan izleme programı tanımlamasının adını görüntüler; örneğin:

BFGCL0762I: 'XFERAGENT' aracısının 'FILEMON' izleme programının tanımı FILEMON.XFERAGENT.XML .

- Belirli bir aracıyı belirtilen bir dizindeki bir XML dosyasına aktararak bir kaynak izleyicisini yedeklemek için, **-od** deęiřtirgesiyle **ftelistmonitors** komutunu kullanın:

```
ftelistmonitors -ma AGENT1 -mn MONITOR1 -od /usr/mft/resmonbackup
```

-od deęiřtirgesinin tek bir kaynak izleme programını yedeklemek için kullanılması, çıkış kütüğü adının *agent name.monitor name.xml* biçiminde olması dışında, **-ox** deęiřtirgesinin kullanılmasına benzer.

- Kaynak izleyicisi tanımlamalarını bir yedekten geri yüklemek için, aşağıdaki örnekte gösterildiği gibi **fteCreateMonitor** komutunu **-ix** değiştirgesiyle birlikte kullanın:

```
fteCreateMonitor -ix file name
```

-od değiştirgesinin kullanılmasına ilişkin ek örnekler için [fteListMonitors: list MFT resource itors](#) (İzleyiciler: liste kaynak izleyicileri) konusuna bakın.

İlgili başvurular

fteCreateMonitor: MFT kaynak izleme programı yaratılması

[fteListİzleme Programları](#): liste MFT kaynak izleme programları

Kaynak izleyicisi geçmişi temizleniyor

Bir hata nedeniyle daha önce aktarılmamış bir dosya için başka bir dosya aktarma isteğinin sunulabilmesi için kaynak izleme programının geçmişi temizleyebilirsiniz. Kaynak izleyicisi geçmişi temizlemek için **fteClearMonitorHistory** komutunu ya da IBM MQ Explorer komutunu kullanabilirsiniz.

Başlamadan önce

`agent.properties` dosyasında **authorityChecking** özneliği `true` olarak ayarlanarak kullanıcı yetki yönetimi etkinleştirildiyse, izleme geçmişi temizleyen kullanıcının aşağıdaki çizelgede gösterildiği gibi uygun yetkiye sahip olması gerekir.

Çizelge 15. fteClearMonitorHistory komutunu çalıştırmak için gereken kullanıcı yetkisi		
İzleyici geçmişi temizleyen kullanıcı	MFT erişim yetkisi	Gerekli yetki
Kaynak izleyicisini yaratanla aynı kullanıcı.	Monitör	BROWSE on SYSTEM.FTE.AUTHMON1. <monitor_agent_name> Bu yetki, kaynak izleme programını yaratmak ya da silmek için gereken yetkiyle aynıdır.
Kaynak izleme programını yaratan kullanıcı dışında herhangi bir kullanıcı.	İzleme işlemleri	SYSTEM.FTE.AUTHPS1 üzerinde SET. <aracı_adi> Bu yetki, kaynak izleme programını silmek için gereken yetkiyle aynıdır.

Kullanıcı yetkisi yönetimiyle ilgili daha fazla bilgi için [MFT aracı işlemlerinde kullanıcı yetkilerinin kısıtlanması](#) başlıklı konuya bakın.

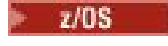
Gerekli yetkisi olmayan bir kullanıcı kaynak izleyicisi geçmişi temizleme girişiminde bulunursa, **fteClearMonitorHistory** komutu bir hata iletisi görüntüler ve hatayı aracının `output0.log` dosyasına kaydeder. Daha fazla bilgi için bkz. [fteClearMonitorHistory: clear resource monitor history](#).

Bu görev hakkında

Bir dosya aktarımı başlatıldıysa ve herhangi bir nedenle dosya aktarılamıyorsa, izleme geçmişi dosyanın daha önceki bir yoklamada görüldüğünü ve o zamandan bu yana değiştirilmediğini belirttiği için, kaynak izleme programı bu dosyayı bir sonraki yoklamasında aktarmak üzere seçmez (bkz. "[MFT kaynak izleme kavramları](#)" sayfa 245).

IBM MQ 9.1.3' den önce bir dosya aktarılamazsa, dosya silinip yeniden dizine yerleştirilirse ya da dosya, son değiştirme tarihi özneliği değiştirilecek şekilde güncellenecek şekilde güncellenirse ya da kaynak izleyicisinin kendisi yeniden yaratılırsa, dosya aktarımı yeniden başlatılabilir.

Ancak, IBM MQ 9.1.3' den **fteClearMonitorHistory** komutunu kullanarak ya da IBM MQ Explorer komutunu kullanarak kaynak izleyici geçmişini temizleyebilirsiniz. Geçmişin temizlenmesi, aktarma işlemi başarısız olan bir dosyaya ilişkin başka bir aktarma isteğinin, dosyayı silme ve dizine yeniden ekleme gereksinimi olmadan sunulmasına ya da son değiştirme tarihi özneliğini değiştirmek için dosyayı güncellemesine olanak sağlar; bu, örneğin, dosyanın aktarılması gereken, ancak dosyanın değiştirilmesinin mümkün olmadığı durumlarda yararlı olur. Bir kaynak izleme programının geçmişinin temizlenmesi, aktarılamayan bir dosyaya ilişkin başka bir aktarma isteği sunmak için kaynak izleme programının yeniden yaratılabilmesi gerekmediği anlamına da gelir.

 z/OS üzerinde Managed File Transfer ile verilen örnek SCSQFCMD üyesi, bir izleme programının geçmişini temizlemek için bir JCL komut dosyası içerir.

Yordam

- Kaynak izleyicisi geçmişini temizlemek üzere **fteClearMonitorHistory** komutunu kullanmak için komutu aşağıdaki biçimde girin:

```
fteClearMonitorHistory -p <configuration> -ma <agent name> -mn <monitor name> -w 1000
```

Yalnızca **-ma** ve **-mn** parametreleri gereklidir. Diğer tüm parametreler isteğe bağlıdır. Örnekler de içinde olmak üzere **fteClearMonitorHistory** komutunun nasıl kullanılacağına ilişkin ek bilgi için [fteClearMonitorHistory: clear resource monitor history](#) başlıklı konuya bakın.

Geçmiş başarıyla temizlenirse, komut şu iletiyi verir:

```
BFGCL0780I: 'aracı adı' aracının 'izleyici adı' kaynak izleyicisinin geçmişini temizleme isteği yayınlandı.  
BFGCL0251I: İstek başarıyla tamamlandı.
```

ve başarıyı aracının output0.log dosyasına kaydeder.

Kaynak izleyicisi geçmişini temizleme girişimi başarısız olursa, **fteClearMonitorHistory** bir hata ileti çıkarır ve hatayı aracının output0.log dosyasına kaydeder.

- Kaynak izleyicisi geçmişini temizlemek üzere IBM MQ Explorer MFT eklentisinde kaynak izleyicisi görünümünü kullanmak için, kaynak izleyicisini farenin sağ düğmesiyle tıklatın ve açılan menüden **Geçmiş Temizle** öğesini seçin.

Geçmiş başarıyla temizlenirse aşağıdaki ileti görüntülenir:

```
BFGUI00171: Kaynak izleyicisi geçmişi başarıyla temizlendi.
```

Geçmiş temizleme girişimi başarısız olursa bir hata ileti görüntülenir. Örneğin:

```
BFGUI0016E Belirtilen kaynak izleme programının geçmişi temizlenemedi-neden: 2059
```

Dosya aktarma şablonlarıyla çalışılması

Yinelenen ya da karmaşık aktarımlara ilişkin ortak dosya aktarma ayarlarını saklamak için dosya aktarma şablonlarını kullanabilirsiniz. **fteCreateTemplate** komutunu kullanarak komut satırından bir aktarma şablonu yaratın ya da **Yönetilen Dosya Aktarımı İçin Yeni Şablon Yarat** sihirbazını kullanarak aktarma şablonu yaratmak için IBM MQ Explorer komutunu kullanın ya da **Aktarma ayarlarını şablon olarak sakla** onay kutusunu seçerek dosya aktarımı yaratırken bir şablonu saklayın. **Aktarma Şablonları** penceresi, Managed File Transfer ağıңызda yarattığınız tüm aktarma şablonlarını görüntüler.

Bu görev hakkında


Komut satırından bir aktarma şablonu yaratmak için **fteCreateTemplate** komutunu kullanın. Daha sonra, komut satırında yarattığınız bir aktarma şablonunu sunmak istediğinizde, IBM MQ Exploreriçinde **Sun** düğmesini tıklatın.

IBM MQ Exploreriçindeki aktarma şablonlarını görüntülemek için aşağıdaki adımları izleyin:

Yordam

1. Navigator görünümünde **Yönetilen Dosya Aktarımı** nesnesini açın. İçerik görünümünde **Yönetilen Dosya Aktarma Merkezi** görüntülenir.
2. Tüm koordinasyon kuyruğu yöneticileriniz Navigator görünümünde listelenir. Zamanlanmış aktarma için kullandığınız eşgüdüm kuyruğu yöneticisinin adını açın. Bağlı olduğunuz koordinasyon kuyruk yöneticisini değiştirmek istiyorsanız, Navigator görünümünde kullanmak istediğiniz koordinasyon kuyruk yöneticisinin adını sağ tıklayın ve **Bağlan'** ı tıklayın.
3. **Aktarma Şablonları'** nı tıklayın. İçerik görünümünde **Aktarma Şablonları** penceresi görüntülenir.
4. **Aktarma Şablonları** penceresi dosya aktarımlarınıza ilişkin aşağıdaki ayrıntıları listeler:
 - a) **Ad** Dosya aktarma şablonunuzun adı.
 - b) **Kaynak** Dosyayı kaynak sistemden aktarmak için kullanılan aracının adı.
 - c) **Kaynak kütük** Anasistemde aktarılacak kütüğün adı.
Bu alanı görüntülemek için aktarma şablonu bilgilerini açın.
 - d) **Hedef** Hedef sistemde dosyayı almak için kullanılan aracının adı.
 - e) **Hedef Dosya** Hedef sisteme aktarıldıktan sonra dosyanın adı.
Bu alanı görüntülemek için aktarma şablonu bilgilerini genişletin.
 - f) **Zamanlanmış Başlangıç (seçilen saat dilimi)** Dosya aktarımının, yönetici tarafından kullanılan saat diliminde başlatılmak üzere zamanlandığı saat ve tarih. Görüntülenen saat dilimini değiştirmek için **Pencere > Tercihler > IBM MQ Explorer > Managed File Transfer** düğmesini tıklayın ve **Saat dilimi:** listesinden alternatif bir saat dilimi seçin. **Tamam'** ı tıklayın.
 - g) **Tetikleme Olayları** Kütük aktarımının başlatılmasını tetikleyen olayın tipi. Tip şu değerlerden biri olabilir: exists, not existsya da aşılıyor.

Sonuçlar

Aktarma Şablonları penceresinde görüntülenenleri yenilemek için, İçerik görünümü araç çubuğundaki Yenile düğmesini  tıklayın.

Bir aktarma şablonunu sunmak ve şablonda tanımlanan aktarımı başlatmak için, şablon adını farenin sağ düğmesiyle tıklayın ve **Sun**ögesini seçin.

Bir aktarma şablonunu değiştirmek için, şablon adını farenin sağ düğmesiyle tıklayın ve **Düzenle**ögesini seçin. Özgün şablona dahil edilen tüm dosyalar, özgün şablondaki bir grubun parçası olarak eklenmemiş olsa da, bir aktarım grubunun parçası olarak listelenir. Şablondan bir dosyayı kaldırmak istiyorsanız, gruptan dosya belirtimini seçmeniz ve **Seçeni kaldır'** ı tıklatmanız gerekir. Şablona yeni dosya belirtileri eklemek istiyorsanız, şablon panosundaki alanları kullanın ve **Gruba ekle** düğmesini tıklayın. Düzenlemelerinizi yaptığınızda, düzenlenen şablona yeni bir ad vermeniz istenir.

Aktarma şablonundan dosya aktarımı yaratmak için, şablon adını farenin sağ düğmesiyle tıklayın ve **Yeni Aktarma Olarak Düzenle**ögesini seçin.

Bir aktarma şablonunun yinelenen bir kopyasını yaratmak için, şablon adını farenin sağ düğmesiyle tıklayın ve **Çoğalt**ögesini seçin. Yinelenen aktarma şablonu, özgün şablonla aynı adla otomatik olarak kaydedilir ve sonuna "(kopya)"eklenir.

Bir aktarma şablonunu silmek için, şablon adını farenin sağ düğmesiyle tıklayın ve **Sil**ögesini seçin.

İlgili görevler

[“IBM MQ Explorer kullanarak dosya aktarma şablonu yaratılması” sayfa 276](#)

IBM MQ Explorer içinden ya da komut satırından bir dosya aktarma şablonu yaratabilirsiniz. Daha sonra şablon ayrıntılarını kullanarak yeni dosya aktarımları oluşturmak için bu şablonu kullanabilir ya da dosya aktarımını başlatmak için şablonu gönderebilirsiniz.

İlgili başvurular

ftCreateTemplate: yeni dosya aktarma şablonu yarat

[ftListŞablonları](#)

IBM MQ Explorer kullanarak dosya aktarma şablonu yaratılması

IBM MQ Explorer içinden ya da komut satırından bir dosya aktarma şablonu yaratabilirsiniz. Daha sonra şablon ayrıntılarını kullanarak yeni dosya aktarımları oluşturmak için bu şablonu kullanabilir ya da dosya aktarımını başlatmak için şablonu gönderebilirsiniz.

Bu görev hakkında

Komut satırından dosya aktarma şablonu yaratmak için [fteCreateTemplate](#) komutunu kullanın.

IBM MQ Explorer' ta **Create New Template for Managed File Transfer** (Yönetilen Dosya Aktarımı İçin Yeni Şablon Yarat) sihirbazını kullanarak bir dosya aktarma şablonu yaratmak için aşağıdaki adımları izleyin:

Yordam

1. Navigator görünümünde **Yönetilen Dosya Aktarımı** seçeneğini tıklatın. İçerik görünümünde **Yönetilen Dosya Aktarma Merkezi** görüntülenir.
2. Tüm eşgüdüm kuyruğu yöneticileriniz Navigator görünümünde görüntülenir. Zamanlanmış aktarma için kullandığınız eşgüdüm kuyruğu yöneticisinin adını açın. Bağlı olduğunuz koordinasyon kuyruk yöneticisini değiştirmek istiyorsanız, Navigator görünümünde kullanmak istediğiniz koordinasyon kuyruk yöneticisinin adını sağ tıklatın ve **Bağlan** ı tıklatın.
3. **Create New Template for Managed File Transfer** (Yönetilen Dosya Aktarımı İçin Yeni Şablon Yarat) sihirbazını başlatmak için **Transfer Templates** (Aktarma Şablonları) öğesini farenin sağ düğmesiyle tıklatın ve **New Template**(Yeni Şablon) öğesini seçin.
4. Sihirbaz panolarındaki yönergeleri izleyin. Her pano için bağlama duyarlı yardım sağlanır. Windows üzerindeki bağlama duyarlı yardıma erişmek için F1 tuşuna basın. Linux işletim sistemlerinde Ctrl+F1 ya da Shift+F1 tuşuna basın.

Gerekli tüm aktarma ayrıntılarını içeren bir şablon yarattıysanız, bu onay kutusu önceden seçilmediyse, **Aktarma Özeti** sayfasında **Aktarma ayarlarını şablon olarak sakla** onay kutusunu işaretlediğinizden emin olun. Ayrıca Ad alanına şablon için bir ad girin. Henüz gerekli tüm aktarma ayrıntılarını içermeyen bir şablon yaratırsanız, **Aktarma ayarlarını şablon olarak sakla** onay kutusu sizin için otomatik olarak işaretlenir.

İlgili görevler

[“Dosya aktarma şablonlarıyla çalışılması” sayfa 274](#)

Yinelenen ya da karmaşık aktarımlara ilişkin ortak dosya aktarma ayarlarını saklamak için dosya aktarma şablonlarını kullanabilirsiniz. **fteCreateTemplate** komutunu kullanarak komut satırından bir aktarma şablonu yaratın ya da **Yönetilen Dosya Aktarımı İçin Yeni Şablon Yarat** sihirbazını kullanarak aktarma şablonu yaratmak için IBM MQ Explorer komutunu kullanın ya da **Aktarma ayarlarını şablon olarak sakla** onay kutusunu seçerek dosya aktarımı yaratırken bir şablonu saklayın. **Aktarma Şablonları** penceresi, Managed File Transfer ağıңызda yarattığınız tüm aktarma şablonlarını görüntüler.

İlgili başvurular

fteCreateTemplate: yeni dosya aktarma şablonu yarat

[fteListŞablonları](#)

[fteDeleteŞablonları](#)

Dosya aktarma şablonu tanımlamasının yedeklenmesi

Dosya aktarma şablonları, aktarmaya ilişkin kaynak ve hedef dosya belirtilmelerini tanımlayan bir XML belgesi içerir. Bir dosya aktarma şablonunu yeniden yaratmak için bu XML dosyasını **fteCreateTemplate** komutuna giriş olarak kullanabilirsiniz.

Bu görev hakkında

Bir aktarma şablonuna ilişkin kaynak ve hedef dosya belirtilmelerini içeren XML belgesini yedeklemek için `fteCreateTransfer` command komutunu ya da IBM MQ Explorer kullanın. Aktarma şablonu XML biçimli bir yedek dosyası yaratmak için aşağıdaki adımları izleyin:

Yordam

- Birinci yöntem: Yeni bir dosyaya aktarma şablonu XML iletisi oluşturmak için `fteCreateTransfer` komutundaki **-gt** parametresini kullanın.
- İkinci yöntem: Şablonu IBM MQ Explorer kullanarak oluşturun.
Aktarma şablonu özeti sayfasına geldiğinizde:
 - a) *İstek iletisi XML önizlemesi*ni kopyalayın.
 - b) Bu aktarma şablonu XML iletisini yeni bir dosyaya saklar.
- Üçüncü yöntem: Var olan şablonları yedeklemek için IBM MQ Explorer kullanın.
 - a) **Yönetilen Dosya Aktarımı > Kuyruk Yöneticisi Adı > Aktarma Şablonları** konusuna gidin.
 - b) Aktarma panosunda, yedeklenmesi gereken şablonu vurgulayın, sağ tıklayın ve beliren menüden **Düzenle** seçeneğini belirleyin.
 - c) *Aktarma şablonu özeti* sayfasına ulaşmaya kadar **İleri** düğmesini tıklayın.
 - d) *İstek iletisi XML önizlemesi*ni kopyalayın.
 - e) Bu aktarma şablonu XML iletisini yeni bir dosyaya saklar.

Sonuçlar

Yukarıdaki yöntemlerden biri tarafından yaratılan aktarma şablonu XML ileti dosyasını, `fteCreateTemplate` komutuna giriş olarak kullanabilirsiniz. Bu komutu nasıl kullanacağınıza ilişkin ayrıntılar için **`fteCreateTemplate`** komutuna bakın.

İlgili başvurular

`fteCreateŞablon` komutu

`fteListtKomutu` çalıştırır

Dosyalardan iletilere veri aktarılması

Bir dosyadan IBM MQ kuyruğundaki tek bir iletiye ya da birden çok iletiye veri aktarmak için Managed File Transfer ' in dosyadan iletiye özelliğini kullanabilirsiniz.

İletiden dosyaya aktarımlarla ilgili bilgi için bkz. "İletilerden dosyalara veri aktarılması" sayfa 291.

Bir dosyadan iletiye aktarıma ilişkin hedef aracı, bir protokol köprüsü aracı ya da bir Connect:Direct köprü aracı olamaz.

Dosya verilerini IBM MQ ileti verilerine aktarabilirsiniz. IBM MQ iletileri uygulamalar tarafından okunabilir ve kullanılabilir. Aşağıdaki dosya iletiden iletiye aktarma tipleri desteklenir:

- Tek bir dosyadan tek bir iletiye. İletinin ayarlanmış bir IBM MQ grup tanıtıcısı yok.
- Dosyayı belirli bir uzunluktaki iletilere bölerek, tek bir dosyadan birden çok iletiye. İletilerin tümü aynı IBM MQ grup tanıtıcısına sahiptir.
- Java düzenli ifade sınırlayıcısında bir metin dosyasını bölerek, tek bir dosyadan birden çok iletiye. İletilerin tümü aynı IBM MQ grup tanıtıcısına sahiptir.
- Onaltılı sınırlayıcıda ikili bir dosyayı bölerek, tek bir dosyadan birden çok iletiye. İletilerin tümü aynı IBM MQ grup tanıtıcısına sahiptir.

Bir ikili dosyayı sınırlayıcı olarak bir bayt sırası kullanarak bölmek istiyorsanız, **`fteCreateTransfer`** komutunun **-sqdb** parametresini kullanın. Daha fazla bilgi için bkz. -sqdb değiştirgesi.

Varsayılan olarak, bir dosyadan iletiye aktarma işlemi tarafından oluşturulan iletiler kalıcıdır. İletiler kalıcı olmayan ya da hedef kuyruk tarafından tanımlanan kalıcılık değerine sahip olacak şekilde ayarlanabilir.

Bir dosyanın birden çok iletiye bölünmesini belirtirseniz, dosyadan oluşturulan tüm iletiler aynı IBM MQ grup tanıtıcısına sahiptir. Bir dosyanın birden çok iletiye bölünmesini belirtmezseniz, dosyadan yalnızca bir ileti oluşturulur ve bu iletide IBM MQ grup tanıtıcısı ayarlanmaz.

Dosyaları büyük iletilere ya da birçok küçük iletiye aktarıyorsanız, bazı IBM MQ ya da Managed File Transfer özelliklerini değiştirmeniz gerekebilir. Hakkında bilgi için bkz. [MQ özneliklerini ve ileti boyutuyla ilişkili MFT özelliklerini ayarlama kılavuzu](#).

Not: Hedef kuyruk kümelenmiş bir kuyruksa ya da kümelenmiş bir kuyruğa diğer adsa, enableClusterQueueInputOutput özelliği true olarak ayarlanmamışsa, bir dosyayı kuyruğa aktarırken bir hata iletisi alırsınız. Daha fazla bilgi için bkz. [Hedef kuyruk kümelenmiş bir kuyruksa ya da kümelenmiş bir kuyruğun diğer adıdır](#)

İlgili görevler

[“Aracıyı dosya-ileti aktarımları gerçekleştirecek şekilde yapılandırma” sayfa 279](#)

Varsayılan olarak, araçlar dosyadan iletiye ya da iletiye dosya aktarımları gerçekleştiremez. Bu işlevi etkinleştirmek için enableQueueInputOutput aracı özelliğini true olarak ayarlamamız gerekir. IBM MQ kümelenmiş kuyruklara yazmayı etkinleştirmek için enableClusterQueueInputOutput özelliğini de true değerine ayarlamamız gerekir.

[“Örnek: Tek bir dosyanın tek bir iletiye aktarılması” sayfa 280](#)

fteCreateTransfer komutuyla **-dq** parametresini kullanarak dosya aktarımının hedefi olarak bir kuyruk belirtebilirsiniz. Kaynak dosya, hedef kuyrukta ayarlanan ileti uzunluğu üst sınırından küçük olmalıdır. Hedef kuyruğun, hedef aracının bağlandığı kuyruk yöneticisiyle aynı kuyruk yöneticisinde olması gerekmez, ancak bu iki kuyruk yöneticisinin iletişim kurabilmesi gerekir.

[“Örnek: Tek bir dosyayı uzunluğa göre birden çok iletiye bölme” sayfa 281](#)

fteCreateTransfer komutunun **-qs** parametresini kullanarak bir dosyayı birden çok IBM MQ iletiye bölebilirsiniz. Dosya, her biri tek bir iletiye yazılan sabit uzunluklu bölümlere bölünür.

[“Örnek: Bir metin dosyasını düzenli ifade sınırlayıcısıyla bölme ve sınırlayıcıyı iletilere dahil etme” sayfa 284](#)

Belirli bir Java düzenli ifadesinin her eşleşmesinde dosyayı bölerek ve sonuçtaki iletilere düzenli ifade eşleşmesini dahil ederek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** ve **-qi** parametrelerini kullanın.

[“Örnek: Düzenli ifade sınırlayıcı kullanarak bir metin dosyasını birden çok iletiye bölme” sayfa 283](#)

Belirli bir Java düzenli ifadesinin her eşleşmesi için dosyayı bölerek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** parametresini kullanırsınız.

[“Örnek: Dosyadan iletiye aktarımda IBM MQ ileti özelliklerinin ayarlanması” sayfa 286](#)

Aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliklerinin ayarlanıp ayarlanmayacağını belirlemek için **fteCreateTransfer** komutundaki **-qmp** parametresini kullanabilirsiniz. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

[“Örnek: Dosyadan iletiye aktarımda kullanıcı tanımlı özellikleri ayarlama” sayfa 288](#)

Kullanıcı tanımlı meta veriler, aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliği olarak ayarlanır. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

[“Yeni dosya aktarımı başlatılması” sayfa 233](#)

IBM MQ Explorer ' den ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir gruptaki tek bir dosyayı ya da birden çok dosyayı aktarmayı seçebilirsiniz.

İlgili başvurular

[“Dosyadan iletiye aktarma başarısız oldu” sayfa 291](#)

Aracı hedef kuyruğa dosya verileri yazmaya başladıktan sonra bir dosyadan iletiye aktarma işlemi başarısız olursa, aracı, bir hatanın olduğu iletileri kullanan bir uygulamaya göstermek için kuyruğa bir ileti yazar.

[Hedef kuyruklara yazılan iletilerde MFT tarafından ayarlanan MQ ileti özellikleri](#)
[İleti boyutuyla ilişkili MQ özneliklerini ve MFT özelliklerini ayarlamaya ilişkin kılavuz](#)

Aracıyı dosya-ileti aktarımları gerçekleştirecek şekilde yapılandırma

Varsayılan olarak, araçlar dosyadan iletiye ya da iletiye dosya aktarımları gerçekleştiremez. Bu işlevi etkinleştirmek için `enableQueueInputOutput` aracı özelliğini `true` olarak ayarlamanız gerekir. IBM MQ kümelenmiş kuyruklara yazmayı etkinleştirmek için `enableClusterQueueInputOutput` özelliğini de `true` değerine ayarlamanız gerekir.

Bu görev hakkında

`enableQueueInputOutput` özelliği `true` olarak ayarlanmamış bir hedef aracıya dosyadan iletiye aktarma girişiminde bulunursanız, aktarma başarısız olur. Eşgüdüm kuyruk yöneticisine yayınlanan aktarma günlüğü iletişi aşağıdaki iletiyi içerir:

```
BFGI00197E: An attempt to write to a queue was rejected by the destination agent. The agent must have enableQueueInputOutput=true set in the agent.properties file to support transferring to a queue.
```

Aracının kuyruklara yazmasını ve kuyruklardan okumasını sağlamak için aşağıdaki adımları gerçekleştirin:

Yordam

1. **fteStopAgent** komutunu kullanarak hedef aracıyı durdurun.
2. `agent.properties` dosyasını `enableQueueInputOutput=true` satırını içerecek şekilde düzenleyin.
`agent.properties` dosyası `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_namedizininde` bulunur.
3. İsteğe bağlı: `agent.properties` dosyasını `enableClusterQueueInputOutput=true` satırını içerecek şekilde düzenleyin. `agent.properties` dosyası `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_namedizininde` bulunur.
4. **fteStartAgent** komutunu kullanarak hedef aracıyı başlatın.

İlgili kavramlar

[“Dosyalardan iletilere veri aktarılması” sayfa 277](#)

Bir dosyadan IBM MQ kuyruğundaki tek bir iletiye ya da birden çok iletiye veri aktarmak için Managed File Transfer ' in dosyadan iletiye özelliğini kullanabilirsiniz.

İlgili görevler

[“Örnek: Tek bir dosyanın tek bir iletiye aktarılması” sayfa 280](#)

fteCreateTransfer komutuyla **-dq** parametresini kullanarak dosya aktarımının hedefi olarak bir kuyruk belirtebilirsiniz. Kaynak dosya, hedef kuyrukta ayarlanan ileti uzunluğu üst sınırından küçük olmalıdır. Hedef kuyruğun, hedef aracının bağlandığı kuyruk yöneticisiyle aynı kuyruk yöneticisinde olması gerekmez, ancak bu iki kuyruk yöneticisinin iletişim kurabilmesi gerekir.

[“Örnek: Tek bir dosyayı uzunluğa göre birden çok iletiye bölme” sayfa 281](#)

fteCreateTransfer komutunun **-qs** parametresini kullanarak bir dosyayı birden çok IBM MQ iletiye bölebilirsiniz. Dosya, her biri tek bir iletiye yazılan sabit uzunluklu bölümlere bölünür.

[“Örnek: Bir metin dosyasını düzenli ifade sınırlayıcısıyla bölme ve sınırlayıcıyı iletilere dahil etme” sayfa 284](#)

Belirli bir Java düzenli ifadesinin her eşleşmesinde dosyayı bölerek ve sonuçtaki iletilere düzenli ifade eşleşmesini dahil ederek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** ve **-qi** parametrelerini kullanın.

[“Örnek: Düzenli ifade sınırlayıcı kullanarak bir metin dosyasını birden çok iletiye bölme” sayfa 283](#)

Belirli bir Java düzenli ifadesinin her eşleşmesi için dosyayı bölerek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** parametresini kullanırsınız.

“Örnek: Dosyadan iletiye aktarımda IBM MQ ileti özelliklerinin ayarlanması” sayfa 286

Aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliklerinin ayarlanıp ayarlanmayacağını belirlemek için **fteCreateTransfer** komutundaki **-qmp** parametresini kullanabilirsiniz. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

“Örnek: Dosyadan iletiye aktarımda kullanıcı tanımlı özellikleri ayarlama” sayfa 288

Kullanıcı tanımlı meta veriler, aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliği olarak ayarlanır. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

İlgili başvurular

fteStopAgent

fteStartAgent

MFT agent.properties dosyası

“Dosyadan iletiye aktarma başarısız oldu” sayfa 291

Aracı hedef kuyruğa dosya verileri yazmaya başladıktan sonra bir dosyadan iletiye aktarma işlemi başarısız olursa, aracı, bir hatanın oluştuğu iletileri kullanan bir uygulamaya göstermek için kuyruğa bir ileti yazar.

Örnek: Tek bir dosyanın tek bir iletiye aktarılması

fteCreateTransfer komutuyla **-dq** parametresini kullanarak dosya aktarımının hedefi olarak bir kuyruk belirtebilirsiniz. Kaynak dosya, hedef kuyrukta ayarlanan ileti uzunluğu üst sınırından küçük olmalıdır. Hedef kuyruğun, hedef aracının bağlandığı kuyruk yöneticisiyle aynı kuyruk yöneticisinde olması gerekmez, ancak bu iki kuyruk yöneticisinin iletişim kurabilmesi gerekir.

Bu görev hakkında

Kaynak dosya /tmp/single_record.txt olarak adlandırılır ve kaynak aracı AGENT_NEPTUNE ile aynı sistemde bulunur. AGENT_NEPTUNE kaynak aracısı QM_NEPTUNE kuyruk yöneticisini kullanıyor. Hedef aracı AGENT_VENUS ve bu aracı QM_VENUS kuyruk yöneticisine bağlanır. RECEIVING_QUEUE hedef kuyruğu, QM_MERCURY kuyruk yöneticisinde bulunur. QM_MERCURY, QM_VENUS kuyruk yöneticisiyle aynı IBM MQ ağında yer almaktadır ve bu ağa QM_VENUS kuyruk yöneticisi tarafından erişilebilir.

Yordam

Aşağıdaki komutu yazın:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -dm QM_VENUS  
-dq RECEIVING_QUEUE@QM_MERCURY /tmp/single_record.txt
```

Hedef kuyruk, hedef aracı tarafından kullanılan kuyruk yöneticisine göre farklı bir kuyruk yöneticisindeyse, **-dq** değıştirgesinin değerini şu biçimde belirtmeniz gerekir: *queue_name@queue_manager_name*. Değerde *@queue_manager_name* belirtmezseniz, hedef aracı hedef kuyruğun hedef aracı kuyruk yöneticisinde bulunduğunu varsayar. Bu kural dışı durum, enableClusterQueueInputÇıkış aracısı özelliği true olarak ayarlandığında ortaya çıktı. Bu durumda hedef aracı, kuyruğun yerini belirlemek için standart IBM MQ çözüme yordamlarını kullanır.

AGENT_NEPTUNE kaynak aracısı, /tmp/single_record.txt dosyasından verileri okur ve bu verileri AGENT_VENUS hedef aracısına aktarır. Hedef aracı AGENT_VENUS, verileri RECEIVING_QUEUE@QM_MERCURY kuyruğundaki kalıcı bir iletiye gönderir. İletin ayarlanmış bir IBM MQ grup tanıtıcısı yok.

İlgili kavramlar

“Dosyalardan iletilere veri aktarılması” sayfa 277

Bir dosyadan IBM MQ kuyruğundaki tek bir iletiye ya da birden çok iletiye veri aktarmak için Managed File Transfer ' in dosyadan iletiye özelliğini kullanabilirsiniz.

İlgili görevler

[“Aracıyı dosya-ileti aktarımları gerçekleştirecek şekilde yapılandırma” sayfa 279](#)

Varsayılan olarak, araçlar dosyadan iletiye ya da iletiye dosya aktarımları gerçekleştiremez. Bu işlevi etkinleştirmek için enableQueueInputOutput aracı özelliğini true olarak ayarlamanız gerekir. IBM MQ kümelenmiş kuyruklara yazmayı etkinleştirmek için enableClusterQueueInputOutput özelliğini de true değerine ayarlamanız gerekir.

[“Örnek: Tek bir dosyayı uzunluğa göre birden çok iletiye bölme” sayfa 281](#)

fteCreateTransfer komutunun **-qs** parametresini kullanarak bir dosyayı birden çok IBM MQ iletiye bölebilirsiniz. Dosya, her biri tek bir iletiye yazılan sabit uzunluklu bölümlere bölünür.

[“Örnek: Bir metin dosyasını düzenli ifade sınırlayıcısıyla bölme ve sınırlayıcıyı iletilere dahil etme” sayfa 284](#)

Belirli bir Java düzenli ifadesinin her eşleşmesinde dosyayı bölerek ve sonuçtaki iletilere düzenli ifade eşleşmesini dahil ederek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** ve **-qi** parametrelerini kullanın.

[“Örnek: Düzenli ifade sınırlayıcı kullanarak bir metin dosyasını birden çok iletiye bölme” sayfa 283](#)

Belirli bir Java düzenli ifadesinin her eşleşmesi için dosyayı bölerek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** parametresini kullanırsınız.

[“Örnek: Dosyadan iletiye aktarımda IBM MQ ileti özelliklerinin ayarlanması” sayfa 286](#)

Aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliklerinin ayarlanıp ayarlanmayacağını belirlemek için **fteCreateTransfer** komutundaki **-qmp** parametresini kullanabilirsiniz. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

[“Örnek: Dosyadan iletiye aktarımda kullanıcı tanımlı özellikleri ayarlama” sayfa 288](#)

Kullanıcı tanımlı meta veriler, aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliği olarak ayarlanır. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

[“Yeni dosya aktarımı başlatılması” sayfa 233](#)

IBM MQ Explorer ' den ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir gruptaki tek bir dosyayı ya da birden çok dosyayı aktarmayı seçebilirsiniz.

İlgili başvurular

[“Dosyadan iletiye aktarma başarısız oldu” sayfa 291](#)

Aracı hedef kuyruğa dosya verileri yazmaya başladıktan sonra bir dosyadan iletiye aktarma işlemi başarısız olursa, aracı, bir hatanın olduğu iletileri kullanan bir uygulamaya göstermek için kuyruğa bir ileti yazar.

Örnek: Tek bir dosyayı uzunluğa göre birden çok iletiye bölme

fteCreateTransfer komutunun **-qs** parametresini kullanarak bir dosyayı birden çok IBM MQ iletiye bölebilirsiniz. Dosya, her biri tek bir iletiye yazılan sabit uzunluklu bölümlere bölünür.

Bu görev hakkında

Kaynak dosya /tmp/source .file olarak adlandırılır ve 36 KB ' dir. Kaynak dosya, kaynak aracı AGENT_NEPTUNE ile aynı sistemde bulunuyor. AGENT_NEPTUNE kaynak aracı, QM_NEPTUNE kuyruk yöneticisine bağlanır. Hedef aracı QM_MERCURY kuyruk yöneticisine bağlanan AGENT_MERCURY ' dir. RECEIVING_QUEUE hedef kuyruğu, QM_MERCURY kuyruk yöneticisinde de bulunur. Aktarım, kaynak dosyayı 1 KB boyutundaki bölümlere böler ve bu bölümlerin her birini RECEIVING_QUEUE içindeki bir iletiye yazar.

Yordam

Aşağıdaki komutu yazın:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -qs 1K /tmp/source.file
```

AGENT_NEPTUNE kaynak aracı, /tmp/source.file dosyasından verileri okur ve bu verileri AGENT_MERCURY hedef aracısına aktarır. Hedef aracı AGENT_MERCURY, verileri RECEIVING_QUEUE@QM_MERCURY kuyruğundaki otuz altı 1 KB kalıcı iletiye yazar. Bu iletilerin tümü aynı IBM MQ grup tanıtıcısına sahip ve gruptaki son iletide IBM MQ LAST_MSG_IN_GROUP işareti ayarlanmış.

İlgili kavramlar

[“Dosyalardan iletilere veri aktarılması” sayfa 277](#)

Bir dosyadan IBM MQ kuyruğundaki tek bir iletiye ya da birden çok iletiye veri aktarmak için Managed File Transfer ' in dosyadan iletiye özelliğini kullanabilirsiniz.

İlgili görevler

[“Aracıyı dosya-ileti aktarımları gerçekleştirecek şekilde yapılandırma” sayfa 279](#)

Varsayılan olarak, araçlar dosyadan iletiye ya da iletiye dosya aktarımları gerçekleştiremez. Bu işlevi etkinleştirmek için enableQueueInputOutput aracı özelliğini true olarak ayarlamamız gerekir. IBM MQ kümelenmiş kuyruklara yazmayı etkinleştirmek için enableClusterQueueInputOutput özelliğini de true değerine ayarlamamız gerekir.

[“Örnek: Tek bir dosyanın tek bir iletiye aktarılması” sayfa 280](#)

fteCreateTransfer komutuyla **-dq** parametresini kullanarak dosya aktarımının hedefi olarak bir kuyruk belirtebilirsiniz. Kaynak dosya, hedef kuyrukta ayarlanan ileti uzunluğu üst sınırından küçük olmalıdır. Hedef kuyruğun, hedef aracının bağlandığı kuyruk yöneticisiyle aynı kuyruk yöneticisinde olması gerekmez, ancak bu iki kuyruk yöneticisinin iletişim kurabilmesi gerekir.

[“Örnek: Bir metin dosyasını düzenli ifade sınırlayıcısıyla bölme ve sınırlayıcıyı iletilere dahil etme” sayfa 284](#)

Belirli bir Java düzenli ifadesinin her eşleşmesinde dosyayı bölerek ve sonuçtaki iletilere düzenli ifade eşleşmesini dahil ederek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** ve **-qi** parametrelerini kullanın.

[“Örnek: Düzenli ifade sınırlayıcı kullanarak bir metin dosyasını birden çok iletiye bölme” sayfa 283](#)

Belirli bir Java düzenli ifadesinin her eşleşmesi için dosyayı bölerek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** parametresini kullanırsınız.

[“Örnek: Dosyadan iletiye aktarımda IBM MQ ileti özelliklerinin ayarlanması” sayfa 286](#)

Aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliklerinin ayarlanıp ayarlanmayacağını belirlemek için **fteCreateTransfer** komutundaki **-qmp** parametresini kullanabilirsiniz. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

[“Örnek: Dosyadan iletiye aktarımda kullanıcı tanımlı özellikleri ayarlama” sayfa 288](#)

Kullanıcı tanımlı meta veriler, aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliği olarak ayarlanır. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

[“Yeni dosya aktarımı başlatılması” sayfa 233](#)

IBM MQ Explorer ' den ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir gruptaki tek bir dosyayı ya da birden çok dosyayı aktarmayı seçebilirsiniz.

İlgili başvurular

[“Dosyadan iletiye aktarma başarısız oldu” sayfa 291](#)

Aracı hedef kuyruğa dosya verileri yazmaya başladıktan sonra bir dosyadan iletiye aktarma işlemi başarısız olursa, aracı, bir hatanın olduğu iletileri kullanan bir uygulamaya göstermek için kuyruğa bir ileti yazar.

Örnek: Düzenli ifade sınırlayıcı kullanarak bir metin dosyasını birden çok iletiye bölme

Belirli bir Java düzenli ifadesinin her eşleşmesi için dosyayı bölerek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** parametresini kullanırsınız.

Bu görev hakkında

Dosya, her biri tek bir iletiye yazılan değişken uzunluklu bölümlere bölünür. Metin dosyası, dosyadaki metnin belirli bir düzenli ifadeyle eşleştiği her noktada bölünür. Kaynak dosya /tmp/names.text olarak adlandırılır ve aşağıdaki içeriğe sahiptir:

```
Jenny Jones,John Smith,Jane Brown
```

Dosyanın nerede bölüneceğini belirten düzenli ifade, virgül karakteridir (,).

Kaynak dosya, QM_NEPTUNE kuyruk yöneticisine bağlanan *AGENT_NEPTUNE* kaynak aracısıyla aynı sistemde bulunuyor. *RECEIVING_QUEUE* hedef kuyruğu, *QM_MERCURY* kuyruk yöneticisinde bulunur. *QM_MERCURY*, *AGENT_MERCURY* hedef aracısı tarafından kullanılan kuyruk yöneticisidir. Aktarım, kaynak dosyayı bölümlere böler ve bu bölümlerin her birini *RECEIVING_QUEUE* adlı kuyruktaki bir iletiye yazar.

Yordam

Aşağıdaki komutu yazın:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -t text -dqdp postfix -dqdt "," /tmp/names.text
```

AGENT_NEPTUNE kaynak aracısı, /tmp/names.text dosyasından verileri okur ve bu verileri *AGENT_MERCURY* hedef aracısına aktarır. Hedef aracı *AGENT_MERCURY*, verileri *RECEIVING_QUEUE* kuyruğundaki üç kalıcı iletiye yazar. Bu iletilerin tümü aynı IBM MQ grup tanıtıcısına sahip ve gruptaki son iletide IBM MQ *LAST_MSG_IN_GROUP* işareti ayarlanmış.

İletilerdeki veriler aşağıdaki gibidir.

- İlk ileti:

```
Jenny Jones
```

- İkinci mesaj:

```
John Smith
```

- Üçüncü mesaj:

```
Jane Brown
```

İlgili kavramlar

[“Dosyalardan iletilere veri aktarılması” sayfa 277](#)

Bir dosyadan IBM MQ kuyruğundaki tek bir iletiye ya da birden çok iletiye veri aktarmak için Managed File Transfer ' in dosyadan iletiye özelliğini kullanabilirsiniz.

İlgili görevler

[“Aracıyı dosya-ileti aktarımları gerçekleştirecek şekilde yapılandırma” sayfa 279](#)

Varsayılan olarak, araçlar dosyadan iletiye ya da iletiye dosya aktarımları gerçekleştiremez. Bu işlevi etkinleştirmek için enableQueueInputOutput aracı özelliğini true olarak ayarlamamız gerekir. IBM MQ kümelenmiş kuyruklara yazmayı etkinleştirmek için enableClusterQueueInputOutput özelliğini de true değerine ayarlamamız gerekir.

“Örnek: Tek bir dosyanın tek bir iletiye aktarılması” sayfa 280

fteCreateTransfer komutuyla **-dq** parametresini kullanarak dosya aktarımının hedefi olarak bir kuyruk belirtebilirsiniz. Kaynak dosya, hedef kuyruksa ayarlanan ileti uzunluğu üst sınırından küçük olmalıdır. Hedef kuyruğun, hedef aracının bağlandığı kuyruk yöneticisiyle aynı kuyruk yöneticisinde olması gerekmez, ancak bu iki kuyruk yöneticisinin iletişim kurabilmesi gerekir.

“Örnek: Tek bir dosyayı uzunluğa göre birden çok iletiye bölme” sayfa 281

fteCreateTransfer komutunun **-qs** parametresini kullanarak bir dosyayı birden çok IBM MQ iletiye bölebilirsiniz. Dosya, her biri tek bir iletiye yazılan sabit uzunluklu bölümlere bölünür.

“Örnek: Bir metin dosyasını düzenli ifade sınırlayıcısıyla bölme ve sınırlayıcıyı iletilere dahil etme” sayfa 284

Belirli bir Java düzenli ifadesinin her eşleşmesinde dosyayı bölerek ve sonuçtaki iletilere düzenli ifade eşleşmesini dahil ederek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** ve **-qi** parametrelerini kullanın.

“Örnek: Dosyadan iletiye aktarımda IBM MQ ileti özelliklerinin ayarlanması” sayfa 286

Aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliklerinin ayarlanıp ayarlanmayacağını belirlemek için **fteCreateTransfer** komutundaki **-qmp** parametresini kullanabilirsiniz. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

“Örnek: Dosyadan iletiye aktarımda kullanıcı tanımlı özellikleri ayarlama” sayfa 288

Kullanıcı tanımlı meta veriler, aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliği olarak ayarlanır. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

“Yeni dosya aktarımı başlatılması” sayfa 233

IBM MQ Explorer ' den ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir gruptaki tek bir dosyayı ya da birden çok dosyayı aktarmayı seçebilirsiniz.

İlgili başvurular

“Dosyadan iletiye aktarma başarısız oldu” sayfa 291

Aracı hedef kuyruğa dosya verileri yazmaya başladıktan sonra bir dosyadan iletiye aktarma işlemi başarısız olursa, aracı, bir hatanın olduğu iletileri kullanan bir uygulamaya göstermek için kuyruğa bir ileti yazar.

MFT tarafından kullanılan düzenli ifadeler

Örnek: Bir metin dosyasını düzenli ifade sınırlayıcısıyla bölme ve sınırlayıcıyı iletilere dahil etme

Belirli bir Java düzenli ifadesinin her eşleşmesinde dosyayı bölerek ve sonuçtaki iletilere düzenli ifade eşleşmesini dahil ederek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** ve **-qi** parametrelerini kullanın.

Bu görev hakkında

Tek bir metin dosyasını bir kuyruksa birden çok iletiye aktarır. Dosya, her biri tek bir iletiye yazılan değişken uzunluklu bölümlere bölünür. Metin dosyası, dosyadaki metnin belirli bir düzenli ifadeyle eşleştiği her noktada bölünür. Kaynak dosya /tmp/customers.text olarak adlandırılır ve aşağıdaki içeriğe sahiptir:

```
Customer name: John Smith
Customer contact details: john@example.net
Customer number: 314
```

```
Customer name: Jane Brown
Customer contact details: jane@example.com
Customer number: 42
```

```
Customer name: James Jones
```

```
Customer contact details: jjones@example.net
Customer number: 26
```

Dosyanın nerede bölüneceğini belirten düzenli ifade `Customer\snumber:\s\d+biçimidir`; bu, "Müşteri numarası: " metniyle eşleşir ve onu izleyen herhangi bir sayıdan oluşur. Komut kabuğunun düzenli ifadeyi değerlendirmesini önlemek için, komut satırında belirtilen düzenli ifadeler çift tırnak işareti içine alınmalıdır. Düzenli ifade, Java düzenli ifadesi olarak değerlendirilir. Daha fazla bilgi için bkz. [MFT](#) tarafından kullanılan düzenli ifadeler.

Varsayılan olarak, düzenli bir ifadenin eşleştirebileceği karakter sayısı beş olarak ayarlanır. Bu örnekte kullanılan düzenli ifade, beş karakterden uzun dizgilerle eşleşir. Beş karakterden uzun eşleşmeleri etkinleştirmek için aracı özellikleri dosyasını `maxDelimiterMatchLength` özelliğini içerecek şekilde düzenleyin.

Varsayılan olarak, düzenli ifadeyle eşleşen metin iletilere dahil edilmez. Bu örnekte olduğu gibi, düzenli ifadeyle eşleşen metni iletilere dahil etmek için `-qi` parametresini kullanın. Kaynak dosya, `QM_NEPTUNE` kuyruk yöneticisine bağlanan kaynak aracı `AGENT_NEPTUNE` ile aynı sistemde bulunuyor. `RECEIVING_QUEUE` hedef kuyruğu, `QM_MERCURY` kuyruk yöneticisinde bulunur. `QM_MERCURY`, hedef aracı `AGENT_MERCURY` tarafından kullanılan kuyruk yöneticisidir. Aktarım, kaynak dosyayı bölümlere ayırır ve bu bölümlerin her birini `RECEIVING_QUEUE` üzerindeki bir iletiye yazar.

Yordam

1. Aşağıdaki komutu kullanarak hedef aracıyı durdurun:

```
fteStopAgent AGENT_MERCURY
```

2. `AGENT_MERCURY` için aracı özellikleri dosyasına aşağıdaki satırı ekleyin:

```
maxDelimiterMatchLength=25
```

Not: `maxDelimiterMatchLength` değerinin artırılması performansı düşürebilir.

3. Aşağıdaki komutu kullanarak hedef aracıyı başlatın:

```
fteStartAgent AGENT_MERCURY
```

4. Aşağıdaki komutu yazın:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY
-dq RECEIVING_QUEUE
text -dqdt "Customer\snumber:\s\d+" -qi -dqdp postfix /tmp/customers.text
```

`AGENT_NEPTUNE` kaynak aracı, `/tmp/customers.text` dosyasından verileri okur ve bu verileri `AGENT_MERCURY` hedef aracısına aktarır. Hedef aracı `AGENT_MERCURY`, verileri `RECEIVING_QUEUE` kuyruğundaki üç kalıcı iletiye yazar. Bu iletilerin tümü aynı IBM MQ grup tanıtıcısına sahip ve gruptaki son iletide `IBM MQ LAST_MSG_IN_GROUP` işareti ayarlanmış.

İletilerdeki veriler aşağıdaki gibidir.

- İlk ileti:

```
Customer name: John Smith
Customer contact details: john@example.net
Customer number: 314
```

- İkinci mesaj:

```
Customer name: Jane Brown
Customer contact details: jane@example.com
Customer number: 42
```

- Üçüncü mesaj:

```
Customer name: James Jones
Customer contact details: jjones@example.net
Customer number: 26
```

İlgili kavramlar

“Dosyalardan iletilere veri aktarılması” sayfa 277

Bir dosyadan IBM MQ kuyruğundaki tek bir iletiye ya da birden çok iletiye veri aktarmak için Managed File Transfer ' in dosyadan iletiye özelliğini kullanabilirsiniz.

İlgili görevler

“Aracıyı dosya-ileti aktarımları gerçekleştirecek şekilde yapılandırma” sayfa 279

Varsayılan olarak, araçlar dosyadan iletiye ya da iletiye dosya aktarımları gerçekleştiremez. Bu işlevi etkinleştirmek için enableQueueInputOutput aracı özelliğini true olarak ayarlamamız gerekir. IBM MQ kümelenmiş kuyruklara yazmayı etkinleştirmek için enableClusterQueueInputOutput özelliğini de true değerine ayarlamamız gerekir.

“Örnek: Tek bir dosyanın tek bir iletiye aktarılması” sayfa 280

fteCreateTransfer komutuyla **-dq** parametresini kullanarak dosya aktarımının hedefi olarak bir kuyruk belirtebilirsiniz. Kaynak dosya, hedef kuyruksa ayarlanan ileti uzunluğu üst sınırından küçük olmalıdır. Hedef kuyruğun, hedef aracının bağlandığı kuyruk yöneticisiyle aynı kuyruk yöneticisinde olması gerekmez, ancak bu iki kuyruk yöneticisinin iletişim kurabilmesi gerekir.

“Örnek: Tek bir dosyayı uzunluğa göre birden çok iletiye bölme” sayfa 281

fteCreateTransfer komutunun **-qs** parametresini kullanarak bir dosyayı birden çok IBM MQ iletiye bölebilirsiniz. Dosya, her biri tek bir iletiye yazılan sabit uzunluklu bölümlere bölünür.

“Örnek: Düzenli ifade sınırlayıcı kullanarak bir metin dosyasını birden çok iletiye bölme” sayfa 283

Belirli bir Java düzenli ifadesinin her eşleşmesi için dosyayı bölerek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** parametresini kullanırsınız.

“Örnek: Dosyadan iletiye aktarımda IBM MQ ileti özelliklerinin ayarlanması” sayfa 286

Aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliklerinin ayarlanıp ayarlanmayacağını belirlemek için **fteCreateTransfer** komutundaki **-qmp** parametresini kullanabilirsiniz. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

“Örnek: Dosyadan iletiye aktarımda kullanıcı tanımlı özellikleri ayarlama” sayfa 288

Kullanıcı tanımlı meta veriler, aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliği olarak ayarlanır. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

“Yeni dosya aktarımı başlatılması” sayfa 233

IBM MQ Explorer ' den ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir gruptaki tek bir dosyayı ya da birden çok dosyayı aktarmayı seçebilirsiniz.

İlgili başvurular

MFT agent.properties dosyası

MFT tarafından kullanılan düzenli ifadeler

Örnek: Dosyadan iletiye aktarımda IBM MQ ileti özelliklerinin ayarlanması

Aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliklerinin ayarlanıp ayarlanmayacağını belirlemek için **fteCreateTransfer** komutundaki **-qmp** parametresini kullanabilirsiniz. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

Bu görev hakkında

-qmp true deęiřtirgesini **fteCreateTransfer** komutuna ekleyin. Bu örnekte, komutu gönderen kullanıcının MQMD kullanıcı kimlięi řudur: larmer.

Yordam

Ařaęıdaki komutu yazın:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM -qmp true  
-t text /tmp/source_file.txt
```

The IBM MQ message properties of the first message written by the destination agent, AGENT_SATURN, to the queue, MY_QUEUE, on queue manager, MyQM, are set to these values:

```
usr.WMQFTETTransferId=414cbaedefa234889d999a8ed09782395ea213ebbc9377cd  
usr.WMQFTETTransferMode=text  
usr.WMQFTESourceAgent=AGENT_JUPITER  
usr.WMQFTEDestinationAgent=AGENT_SATURN  
usr.WMQFTEFileName=source_file.txt  
usr.WMQFTEFileSize=1024  
usr.WMQFTEFileLastModified=1273740879040  
usr.WMQFTEFileIndex=0  
usr.WMQFTEmqmdUser=larmer
```

İlgili kavramlar

[“Dosyalardan iletilere veri aktarılması” sayfa 277](#)

Bir dosyadan IBM MQ kuyruęundaki tek bir iletiye ya da birden çok iletiye veri aktarmak için Managed File Transfer ' in dosyadan iletiye özellięini kullanabilirsiniz.

İlgili görevler

[“Aracıyı dosya-ileti aktarımları gerçekleřtiren şekilde yapılandırma” sayfa 279](#)

Varsayılan olarak, araçlar dosyadan iletiye ya da iletiye dosya aktarımları gerçekleřtiremez. Bu işlevi etkinleřtirmek için enableQueueInputOutput aracı özellięini true olarak ayarlamamız gerekir. IBM MQ kümelenmiř kuyruklara yazmayı etkinleřtirmek için enableClusterQueueInputOutput özellięini de true deęerine ayarlamamız gerekir.

[“Örnek: Tek bir dosyanın tek bir iletiye aktarılması” sayfa 280](#)

fteCreateTransfer komutuyla **-dq** parametresini kullanarak dosya aktarımının hedefi olarak bir kuyruk belirtebilirsiniz. Kaynak dosya, hedef kuyrukta ayarlanan ileti uzunluęu üst sınırından küçük olmalıdır. Hedef kuyruęun, hedef aracının baęlandıęı kuyruk yöneticisiyle aynı kuyruk yöneticisinde olması gerekmez, ancak bu iki kuyruk yöneticisinin iletiřim kurabilmesi gerekir.

[“Örnek: Tek bir dosyayı uzunluęa göre birden çok iletiye bölme” sayfa 281](#)

fteCreateTransfer komutunun **-qs** parametresini kullanarak bir dosyayı birden çok IBM MQ iletiye bölebilirsiniz. Dosya, her biri tek bir iletiye yazılan sabit uzunluklu bölümlere bölünür.

[“Örnek: Bir metin dosyasını düzenli ifade sınırlayıcısıyla bölme ve sınırlayıcıyı iletilere dahil etme” sayfa 284](#)

Belirli bir Java düzenli ifadesinin her eřleřmesinde dosyayı bölerek ve sonuçtaki iletilere düzenli ifade eřleřmesini dahil ederek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** ve **-qi** parametrelerini kullanın.

[“Örnek: Düzenli ifade sınırlayıcı kullanarak bir metin dosyasını birden çok iletiye bölme” sayfa 283](#)

Belirli bir Java düzenli ifadesinin her eřleřmesi için dosyayı bölerek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** parametresini kullanırsınız.

[“Örnek: Dosyadan iletiye aktarımda kullanıcı tanımlı özellikleri ayarlama” sayfa 288](#)

Kullanıcı tanımlı meta veriler, aktarım tarafından hedef kuyruęa yazılan ilk iletide IBM MQ ileti özellięi olarak ayarlanır. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine eriřmeden bir iletiye iliřkin bilgileri almasını sağlar.

[“Yeni dosya aktarımı başlatılması” sayfa 233](#)

IBM MQ Explorer ' den ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir gruptaki tek bir dosyayı ya da birden çok dosyayı aktarmayı seçebilirsiniz.

İlgili başvurular

[“Dosyadan iletiye aktarma başarısız oldu” sayfa 291](#)

Aracı hedef kuyruğa dosya verileri yazmaya başladıktan sonra bir dosyadan iletiye aktarma işlemi başarısız olursa, aracı, bir hatanın olduğu iletileri kullanan bir uygulamaya göstermek için kuyruğa bir ileti yazar.

[Hedef kuyruklara yazılan iletilerde MFT tarafından ayarlanan MQ ileti özellikleri](#)

Örnek: Dosyadan iletiye aktarımda kullanıcı tanımlı özellikleri ayarlama

Kullanıcı tanımlı meta veriler, aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliği olarak ayarlanır. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

Bu görev hakkında

usr.account özelliğini RFH2 üstbilgisinde 123456 olarak ayarlamak için **fteCreateTransfer** komutuna -qmp true ve -md account=123456 parametrelerini ekleyin.

Yordam

Aşağıdaki komutu yazın:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM  
-qmp true -md account=123456 /tmp/source_file.txt
```

Standart IBM MQ ileti özellikleri kümesine ek olarak, kullanıcı tanımlı özellik, hedef aracı tarafından yazılan ilk iletinin ileti üstbilgisinde (AGENT_SATURN), kuyruk yöneticisinde (MyQM) MY_QUEUEkuyruğuna ayarlanır. Üstbilgi şu değere ayarlandı:

```
usr.account=123456
```

usr öneki, kullanıcı tanımlı meta verilerin adının başına eklenir.

İlgili kavramlar

[“Dosyalardan iletilere veri aktarılması” sayfa 277](#)

Bir dosyadan IBM MQ kuyruğundaki tek bir iletiye ya da birden çok iletiye veri aktarmak için Managed File Transfer ' in dosyadan iletiye özelliğini kullanabilirsiniz.

İlgili görevler

[“Aracıyı dosya-ileti aktarımları gerçekleştirecek şekilde yapılandırma” sayfa 279](#)

Varsayılan olarak, araçlar dosyadan iletiye ya da iletiye dosya aktarımları gerçekleştiremez. Bu işlevi etkinleştirmek için enableQueueInputOutput aracı özelliğini true olarak ayarlamamız gerekir. IBM MQ kümelenmiş kuyruklara yazmayı etkinleştirmek için enableClusterQueueInputOutput özelliğini de true değerine ayarlamamız gerekir.

[“Örnek: Tek bir dosyanın tek bir iletiye aktarılması” sayfa 280](#)

fteCreateTransfer komutuyla **-dq** parametresini kullanarak dosya aktarımının hedefi olarak bir kuyruk belirtebilirsiniz. Kaynak dosya, hedef kuyrukta ayarlanan ileti uzunluğu üst sınırından küçük olmalıdır. Hedef kuyruğun, hedef aracının bağlandığı kuyruk yöneticisiyle aynı kuyruk yöneticisinde olması gerekmez, ancak bu iki kuyruk yöneticisinin iletişim kurabilmesi gerekir.

[“Örnek: Tek bir dosyayı uzunluğa göre birden çok iletiye bölme” sayfa 281](#)

fteCreateTransfer komutunun **-qs** parametresini kullanarak bir dosyayı birden çok IBM MQ iletiye bölebilirsiniz. Dosya, her biri tek bir iletiye yazılan sabit uzunluklu bölümlere bölünür.

“Örnek: Bir metin dosyasını düzenli ifade sınırlayıcısıyla bölme ve sınırlayıcıyı iletilere dahil etme” sayfa 284

Belirli bir Java düzenli ifadesinin her eşleşmesinde dosyayı bölerek ve sonuçtaki iletilere düzenli ifade eşleşmesini dahil ederek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** ve **-qi** parametrelerini kullanın.

“Örnek: Düzenli ifade sınırlayıcı kullanarak bir metin dosyasını birden çok iletiye bölme” sayfa 283

Belirli bir Java düzenli ifadesinin her eşleşmesi için dosyayı bölerek tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** parametresini kullanırsınız.

“Örnek: Dosyadan iletiye aktarımda IBM MQ ileti özelliklerinin ayarlanması” sayfa 286

Aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliklerinin ayarlanıp ayarlanmayacağını belirlemek için **fteCreateTransfer** komutundaki **-qmp** parametresini kullanabilirsiniz. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

“Yeni dosya aktarımı başlatılması” sayfa 233

IBM MQ Explorer ' den ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir gruptaki tek bir dosyayı ya da birden çok dosyayı aktarmayı seçebilirsiniz.

İlgili başvurular

Hedef kuyruklara yazılan iletilerde MFT tarafından ayarlanan MQ ileti özellikleri

Örnek: Dosyadan iletiye aktarma için kullanıcı tanımlı bir ileti özelliği ekleme

İletiler arası yönetilen aktarımlar için Managed File Transfer kullanıyorsanız, sonuçtaki ileti için kullanıcı tanımlı bir ileti özelliği ekleyebilirsiniz.

Bu görev hakkında

Özel bir ileti özelliği tanımlamak için aşağıdaki yöntemlerden herhangi birini kullanabilirsiniz:

- Aktarma isteğinde **-md** parametresini belirleyin. Daha fazla bilgi için, bkz. “Örnek: Dosyadan iletiye aktarımda kullanıcı tanımlı özellikleri ayarlama” sayfa 288.
- Bir Ant görevi kullanın; fte: filecopy ya da fte:filetaşıma işlevini kullanabilirsiniz. Aşağıdaki örnek bir fte: filecopy görevidir:

```
<project xmlns:fte="antlib.com.ibm.wmqfte.ant.taskdefs" default="complete">
<!-- Initialise the properties used in this script.-->

<target name="init" description="initialise task properties">
  <property name="src.file" value="/home/user/file1.bin"/>
  <property name="dst.queue" value="TEST.QUEUE@qm2"/>
  <fte:uuid property="job.name" length="8"
prefix="copyjob#"/>
</target>
<target name="step1" depends="init" description="transfer file">

<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="copy.result">

<fte:metadata>
<fte:entry name="fileName" value="{FileName}"/>
</fte:metadata>

<fte:filespec srcfilespec="{src.file}" dstqueue="{dst.queue}"
dstmsgprops="true"/>

</fte:filecopy>

</target>
</project>
```

- Bir kaynak izleyicisi ve değişken yerine koyma değeri kullanın. Aşağıdaki örnek, bazı aktarma görevi XML ' ini göstermektedir:

```

<?xml version="1.0" encoding="UTF-8"?>
<monitor:monitor
xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="5.00"
xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./Monitor.xsd">
  <name>METADATA</name>
  <pollInterval units="minutes">5</pollInterval>
  <batch maxSize="5"/>
  <agent>AGENT1</agent>
  <resources>
    <directory recursionLevel="0">e:\temp</directory>
  </resources>
  <triggerMatch>
    <conditions>
      <allOf>
        <condition>
          <fileMatch>
            <pattern>*.txt</pattern>
          </fileMatch>
        </condition>
      </allOf>
    </conditions>
  </triggerMatch>
  <tasks>
    <task>
      <name/>
      <transfer>
        <request version="5.00"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
          <managedTransfer>
            <originator>
              <hostName>mqjason.raleigh.ibm.com.</hostName>
              <userID>administrator</userID>
            </originator>
            <sourceAgent QMgr="AGENTQM" agent="AGENT1"/>
            <destinationAgent QMgr="AGENTQM" agent="AGENT2"/>
            <transferSet priority="0">
              <metaDataSet>
                <metaData key="FileName">${FileName}</metaData>
              </metaDataSet>
              <item checksumMethod="MD5" mode="text">
                <source disposition="delete" recursive="false">
                  <file>${FilePath}</file>
                </source>
                <destination type="queue">
                  <queue persistent="true"
setMqProps="true">TEST.QUEUE@AGENTQM</queue>
                </destination>
              </item>
            </transferSet>
          </job>
          <name>Metadata_example</name>
        </job>
      </managedTransfer>
    </request>
  </transfer>
</task>
</tasks>
<originator>
  <hostName>mqjason.raleigh.ibm.com.</hostName>
  <userID>administrator</userID>
</originator>
</monitor:monitor>

```

İlgili görevler

“Örnek: Dosyadan iletiye aktarımda IBM MQ ileti özelliklerinin ayarlanması” sayfa 286

Aktarım tarafından hedef kuyruğa yazılan ilk iletide IBM MQ ileti özelliklerinin ayarlanıp ayarlanmayacağını belirlemek için **fteCreateTransfer** komutundaki **-qmp** parametresini kullanabilirsiniz. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesini ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiye ilişkin bilgileri almasını sağlar.

İlgili başvurular

[fte: filecopy Ant görevi](#)

[fte: filetaşıma Ant görevi](#)

Dosyadan iletiye aktarma başarısız oldu

Aracı hedef kuyruğa dosya verileri yazmaya başladıktan sonra bir dosyadan iletiye aktarma işlemi başarısız olursa, aracı, bir hatanın olduğu iletileri kullanan bir uygulamaya göstermek için kuyruğa bir ileti yazar.

Bir hata oluştuğunda hedef kuyruğa yazılan ileti:

- Boşluk
- Aracı tarafından hedef kuyruğa yazılan önceki iletiyle aynı IBM MQ grup tanıtıcısına sahip
- IBM MQ LAST_MSG_IN_GROUP işareti ayarlı
- İleti özellikleri etkinleştirildiyse, ek IBM MQ iletileri özelliklerini içerir. Daha fazla bilgi için, hedef kuyruklara yazılan iletilerde MFT tarafından ayarlanan [MQ ileti özellikleri](#) konusuna bakın.

Örnek

Aşağıdaki komut çalıştırılarak aktarma istenir:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq RECEIVING_QUEUE  
-qmp true -qs 1K /tmp/source1.txt
```

source1.txt dosyası 48 KB 'dir. Aktarma işlemi bu dosyayı 1 KB iletiye böler ve bu iletileri RECEIVING_QUEUE hedef kuyruğuna yazar.

Aktarma işlemi devam ederken, aracı RECEIVING_QUEUE kuyruğuna 16 ileti yazdıktan sonra, kaynak araçıda bir hata oluşur.

Aracı, RECEIVING_QUEUE değerine boş bir ileti yazar. Standart ileti özellikleri kümesine ek olarak, boş iletide aşağıdaki ileti özellikleri ayarlanmıştır:

```
usr.WMQFTEResultCode = 40  
usr.WMQFTESupplement = BFGTR0036I: The transfer failed to complete successfully.
```

IBM MQ 9.3.0' den bir dosyadan aktarma başarısız olduğunda, sınırlayıcı boyutu denetimi hatası nedeniyle yalnızca bir boş ileti gönderilir. Buna ek olarak, aktarma başarısızlığının nedeni, sınırlayıcının hedef araçıdaki küme büyüklüğünü aşması ise, kullanıcı özellikleri de bu iletiye eklenir.

İlgili kavramlar

[“Dosyalardan iletilere veri aktarılması” sayfa 277](#)

Bir dosyadan IBM MQ kuyruğundaki tek bir iletiye ya da birden çok iletiye veri aktarmak için Managed File Transfer ' in dosyadan iletiye özelliğini kullanabilirsiniz.

İlgili görevler

[“Aracıyı dosya-ileti aktarımları gerçekleştirecek şekilde yapılandırma” sayfa 279](#)

Varsayılan olarak, araçlar dosyadan iletiye ya da iletiye dosya aktarımları gerçekleştiremez. Bu işlevi etkinleştirmek için enableQueueInputOutput aracı özelliğini true olarak ayarlamanız gerekir. IBM MQ kümelenmiş kuyruklara yazmayı etkinleştirmek için enableClusterQueueInputOutput özelliğini de true değerine ayarlamanız gerekir.

[“Yeni dosya aktarımı başlatılması” sayfa 233](#)

IBM MQ Explorer ' den ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir gruptaki tek bir dosyayı ya da birden çok dosyayı aktarmayı seçebilirsiniz.

İlgili başvurular

[MFT agent.properties dosyası](#)

[Hedef kuyruklara yazılan iletilerde MFT tarafından ayarlanan MQ ileti özellikleri](#)

İletilerden dosyalara veri aktarılması

Managed File Transfer iletiden dosyaya özelliği, IBM MQ kuyruğundaki bir ya da daha çok iletiden dosyaya, veri kümesine (z/OSüzerindeki) ya da kullanıcı dosya alanına veri aktarmanızı sağlar. IBM MQ iletilerini

yaratan ya da işleyen bir uygulamanız varsa, bu iletileri Managed File Transfer ağındaki herhangi bir sistemde bulunan bir dosyaya aktarmak için Managed File Transfer iletimden dosyaya aktarma yeteneğini kullanabilirsiniz.

Dosyadan iletiye aktarımlarla ilgili bilgi için bkz. [“Dosyalardan iletilere veri aktarılması” sayfa 277.](#)



Uyarı: İletiden dosyaya aktarıma ilişkin kaynak aracı bir protokol köprüsü aracı ya da Connect:Direct köprü aracı olamaz.

IBM MQ ileti verilerini bir dosyaya aktarabilirsiniz. Aşağıdaki iletimden dosyaya aktarma tipleri desteklenir:

- Tek bir iletimden tek bir dosyaya
- Birden çok iletimden tek bir dosyaya
- Aynı IBM MQ grup tanıtıcısına sahip birden çok iletimden tek bir dosyaya.
- Dosyaya yazılan her iletideki veriler arasında bir metin ya da ikili sınırlayıcı da içinde olmak üzere, birden çok iletimden tek bir dosyaya.

Büyük iletimlerden ya da birçok küçük iletimden dosya aktarıyorsanız, bazı IBM MQ ya da Managed File Transfer özelliklerini değiştirmeniz gerekebilir. İleti boyutuyla ilişkili [MQ özniteliklerini ve MFT özelliklerini ayarlama kılavuzubaşlıklı konuya bakın.](#)

IBM MQ 9.1.0'den bir iletimden dosya aktarımına kaynak aracı, IBM MQ' un önceki sürümlerinde yıkıcı GET ' den farklı olarak kaynak kuyruktaki iletilere göz atmaktadır. Tüm iletilere (ileti gruplaması kullanılıyorsa) göz atıldıktan ve hedef dosyaya veri yazıldıktan sonra, iletiler kaynak kuyruktan kaldırılır. Bu, bir aktarma başarısız olursa ya da iptal edilirse iletilerin kaynak kuyruktan kalmasını sağlar. Bu değişiklik nedeniyle, dosya aktarımlarında ileti çalıştırmak için GET yetkisiyle birlikte BROWSE yetkisi de sağlanmalıdır.

IBM MQ 9.0.0 Fix Pack 2 ve IBM MQ 9.0.4sürümünden Managed File Transfer , daha önce APAR [IT18213](#)tarafından kaldırılan aktarma tanıtıcısı ve aktarma isteği XML bilgi yükü içindeki groupId özniteliklerinin değerini geri yüklemek için güncellenir. Bu iki tanıtıcı eşdeğerse, kaynak aracı, iletimden dosyaya aktarma için giriş kuyruğunda yapılan ilk MQGET girişimi için bir ileti tanıtıcısı eşleştirme seçeneği (grup tanıtıcısı eşleştirme seçeneğinin tersine) olarak tanıtıcıyı kullanır.

İlgili görevler

“Örnek: MFT kaynağını yapılandırma” sayfa 255

Bir kaynak izleme programı tarafından izlenecek kaynak olarak IBM MQ kuyruğunu, **fteCreateMonitor** komutuyla **-mq** deęiştirgesini kullanarak belirtebilirsiniz.

İlgili başvurular

[Kaynak kuyruklardaki iletimlerden MFT tarafından okunan MQ ileti özellikleri](#)

[İleti boyutuyla ilişkili MQ özniteliklerini ve MFT özelliklerini ayarlama ile ilgili kılavuz](#)

Aracıyı ileti-dosya aktarımları gerçekleştirecek şekilde yapılandırma

Varsayılan olarak, araçlar ileti aktarımlarını dosyalanacak ya da iletiye dosyalanacak şekilde gerçekleştiremez. Bu işlevi etkinleştirmek için enableQueueInputOutput aracı özelliğini true olarak ayarlamamız gerekir.

Bu görev hakkında

enableQueueInputOutput özelliği true olarak ayarlanmamış bir kaynak araçından dosya aktarımı için bir ileti gerçekleştirme girişiminde bulunursanız, aktarma başarısız olur. Eşgüdüm kuyruk yöneticisine yayınlanan aktarma günlüğü iletisi aşağıdaki iletiyi içerir:

```
BFGI00197E: An attempt to read from a queue was rejected by the source agent.
The agent must have enableQueueInputOutput=true set in the agent.properties file
to support transferring from a queue.
```

Aracının kuyruklara yazmasını ve kuyruklardan okumasını sağlamak için aşağıdaki adımları gerçekleştirin:

Yordam

1. **fteStopAgent** komutunu kullanarak kaynak aracıyı durdurun.
2. `agent.properties` dosyasını `enableQueueInputOutput=truesatırını` içerecek şekilde düzenleyin.
`agent.properties` dosyası `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/source_agent_namedizininde` bulunur.
3. **fteStartAgent** komutunu kullanarak kaynak aracıyı başlatın.

İlgili kavramlar

[“İletilerden dosyalara veri aktarılması” sayfa 291](#)

Managed File Transfer iletimden dosyaya özelliği, IBM MQ kuyruğundaki bir ya da daha çok iletimdosyaya, veri kümesine (z/OSüzerindeki) ya da kullanıcı dosya alanına veri aktarmanızı sağlar. IBM MQ iletilerini yaratan ya da işleyen bir uygulamanız varsa, bu iletileri Managed File Transfer ağındaki herhangi bir sistemde bulunan bir dosyaya aktarmak için Managed File Transfer iletimden dosyaya aktarma yeteneğini kullanabilirsiniz.

İlgili görevler

[“Örnek: Kuyruktan tek bir dosyaya aktarma” sayfa 293](#)

fteCreateTransfer komutuyla **-sq** parametresini kullanarak dosya aktarımının kaynağı olarak bir IBM MQ kuyruğu belirtebilirsiniz.

[“Örnek: Bir ileti grubunun kuyruktan tek bir dosyaya aktarılması” sayfa 294](#)

fteCreateTransfer komutuyla **-sq** ve **-sqgi** parametrelerini kullanarak, dosya aktarımının kaynağı olarak IBM MQ kuyruğundaki tek bir tam grubu belirtebilirsiniz.

[“Örnek: Her iletideki verilerden önce bir metin sınırlayıcı ekleme” sayfa 296](#)

Bir kaynak kuyruktan bir dosyaya metin kipinde aktarma yaparken, **fteCreateTransfer** komutuyla **-sq**, **-sqdt** ve **-sqdp** parametrelerini kullanarak, tek tek iletilerdeki verilerden önce bir metin sınırlayıcı eklenmesini belirtebilirsiniz.

[“Örnek: Her iletideki verilerden sonra ikili sınırlayıcı ekleme” sayfa 297](#)

İkili kipte bir kaynak kuyruktan bir dosyaya aktarılırken, **fteCreateTransfer** komutuyla **-sq**, **-sqdbve** **-sqdp** parametrelerini kullanarak tek tek iletilerden gelen verilerden sonra bir ikili sınırlayıcı eklenmesini belirtebilirsiniz.

[“Kuyruğun izlenmesi ve değişken yerine koyma değerinin kullanılması” sayfa 261](#)

fteCreateMonitor komutunu kullanarak, bir kuyruğu izleyebilir ve iletileri izlenen kuyruktan bir dosyaya aktarabilirsiniz. İzlenen kuyruktan okunacak ilk iletideki herhangi bir IBM MQ ileti özelliğinin değeri, görev XML tanımlamasında yerine konabilir ve aktarma davranışını tanımlamak için kullanılabilir.

[“Örnek: IBM MQ ileti özellikleri kullanılarak iletimden dosyaya aktarma başarısız” sayfa 300](#)

`usr.UserReturnCode` IBM MQ ileti özelliğini sıfır dışında bir değere ayarlayarak bir iletinin dosya aktarımının başarısız olmasına neden olabilir. `usr.UserSupplement` IBM MQ ileti özelliğini ayarlayarak, başarısızlığın nedenine ilişkin ek bilgileri de belirtebilirsiniz.

İlgili başvurular

[MFT agent.properties dosyası](#)

Örnek: Kuyruktan tek bir dosyaya aktarma

fteCreateTransfer komutuyla **-sq** parametresini kullanarak dosya aktarımının kaynağı olarak bir IBM MQ kuyruğu belirtebilirsiniz.

Bu görev hakkında

Kaynak veriler, `START_QUEUE` kuyruğundaki üç iletide bulunur. Bu kuyruk, kaynak aracının kuyruk yöneticisinde (`QM_NEPTUNE`) bulunmalıdır.

Yordam

Aşağıdaki komutu yazın:

Örnek: Her iletideki verilerden önce bir metin sınırlayıcı ekleme

Bir kaynak kuyruktan bir dosyaya metin kipinde aktarma yaparken, **fteCreateTransfer** komutuyla **-sq**, **-sqdt** ve **-sqdp** parametrelerini kullanarak, tek tek iletilerdeki verilerden önce bir metin sınırlayıcı eklenmesini belirtebilirsiniz.

Bu görev hakkında

Bu örnekte, START_QUEUE kuyruğunda dört ileti vardır. Bu kuyruk kaynak aracının kuyruk yöneticisinde, QM_NEPTUNE. Her iletideki verilerden önce eklenecek metin sınırlayıcı, Java hazır bilgi dizgisi olarak ifade edilebilir; örneğin: \n\u002D\u002D\u002D\n.

Yordam

Aşağıdaki komutu yazın:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/output.txt  
-t text -sqdt "\n\u002D\u002D\u002D\n" -sqdp prefix -sq START_QUEUE
```

Metin sınırlayıcı, kaynak aracı AGENT_NEPTUNE tarafından START_QUEUE 'daki dört iletinin her birinden alınan verilerin başına eklenir. Bu veriler hedef dosyaya (/out/output .txt) yazılır.

İlgili kavramlar

[“İletilerden dosyalara veri aktarılması” sayfa 291](#)

Managed File Transfer iletiden dosyaya özelliği, IBM MQ kuyruğundaki bir ya da daha çok iletidendosyaya, veri kümesine (z/OSüzerindeki) ya da kullanıcı dosya alanına veri aktarmanızı sağlar. IBM MQ iletilerini yaratan ya da işleyen bir uygulamanız varsa, bu iletileri Managed File Transfer ağınzdaki herhangi bir sistemde bulunan bir dosyaya aktarmak için Managed File Transfer iletiden dosyaya aktarma yeteneğini kullanabilirsiniz.

İlgili görevler

[“Aracıyı ileti-dosya aktarımları gerçekleştirecek şekilde yapılandırma” sayfa 292](#)

Varsayılan olarak, araçlar ileti aktarımlarını dosyalanacak ya da iletiye dosyalanacak şekilde gerçekleştiremez. Bu işlevi etkinleştirmek için enableQueueInputOutput aracı özelliğini true olarak ayarlamamız gerekir.

[“Örnek: Kuyruktan tek bir dosyaya aktarma” sayfa 293](#)

fteCreateTransfer komutuyla **-sq** parametresini kullanarak dosya aktarımının kaynağı olarak bir IBM MQ kuyruğu belirtebilirsiniz.

[“Örnek: Bir ileti grubunun kuyruktan tek bir dosyaya aktarılması” sayfa 294](#)

fteCreateTransfer komutuyla **-sq** ve **-sqgi** parametrelerini kullanarak, dosya aktarımının kaynağı olarak IBM MQ kuyruğundaki tek bir tam grubu belirtebilirsiniz.

[“Örnek: Her iletideki verilerden sonra ikili sınırlayıcı ekleme” sayfa 297](#)

İkili kipte bir kaynak kuyruktan bir dosyaya aktarılırken, **fteCreateTransfer** komutuyla **-sq**, **-sqdbve** **-sqdp** parametrelerini kullanarak tek tek iletilerden gelen verilerden sonra bir ikili sınırlayıcı eklenmesini belirtebilirsiniz.

[“Kuyruğun izlenmesi ve değişken yerine koyma değerinin kullanılması” sayfa 261](#)

fteCreateMonitor komutunu kullanarak, bir kuyruğu izleyebilir ve iletileri izlenen kuyruktan bir dosyaya aktarabilirsiniz. İzlenen kuyruktan okunacak ilk iletideki herhangi bir IBM MQ ileti özelliğinin değeri, görev XML tanımlamasında yerine konabilir ve aktarma davranışını tanımlamak için kullanılabilir.

[“Örnek: IBM MQ ileti özellikleri kullanılarak iletiden dosyaya aktarma başarısız” sayfa 300](#)

usr. UserReturnCode IBM MQ ileti özelliğini sıfır dışında bir değere ayarlayarak bir iletinin dosya aktarımının başarısız olmasına neden olabilir. usr. UserSupplement IBM MQ ileti özelliğini ayarlayarak, başarısızlığın nedenine ilişkin ek bilgileri de belirtebilirsiniz.

İlgili başvurular

[fCreateTransfer](#): yeni bir dosya aktarımı başlatır

Örnek: Her iletideki verilerden sonra ikili sınırlayıcı ekleme

İkili kipte bir kaynak kuyruktan bir dosyaya aktarılırken, **fteCreateTransfer** komutuyla **-sq**, **-sqdbve** **-sqdp** parametrelerini kullanarak tek tek iletilerden gelen verilerden sonra bir ikili sınırlayıcı eklenmesini belirtebilirsiniz.

Bu görev hakkında

Bu örnekte, START_QUEUE kuyruğunda üç ileti vardır. Bu kuyruk kaynak aracının kuyruk yöneticisinde, QM_NEPTUNE. Her iletideki verilerden sonra eklenecek ikili sınırlayıcı, onaltılı baytların virgülle ayrılmış bir listesi olarak ifade edilmelidir; örneğin: x34, xE7, xAE.

Yordam

Aşağıdaki komutu yazın:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/binary.file  
-sqdp postfix -sqdb x34,xE7,xAE -sq START_QUEUE
```

İkili sınırlayıcı, AGENT_NEPTUNE kaynak aracı tarafından START_QUEUE 'daki üç iletinin her birinden verilerin sonuna eklenir. Bu veriler hedef dosyaya (/out/binary.file) yazılır.

İlgili kavramlar

[“İletilerden dosyalara veri aktarılması” sayfa 291](#)

Managed File Transfer iletiden dosyaya özelliği, IBM MQ kuyruğundaki bir ya da daha çok iletidendosyaya, veri kümesine (z/OSüzzerindeki) ya da kullanıcı dosya alanına veri aktarmanızı sağlar. IBM MQ iletilerini yaratan ya da işleyen bir uygulamanız varsa, bu iletileri Managed File Transfer ağınzdaki herhangi bir sistemde bulunan bir dosyaya aktarmak için Managed File Transfer iletiden dosyaya aktarma yeteneğini kullanabilirsiniz.

İlgili görevler

[“Aracıyı ileti-dosya aktarımları gerçekleştirecek şekilde yapılandırma” sayfa 292](#)

Varsayılan olarak, araçlar ileti aktarımlarını dosyalanacak ya da iletiye dosyalanacak şekilde gerçekleştiremez. Bu işlevi etkinleştirmek için enableQueueInputOutput aracı özelliğini true olarak ayarlamanız gerekir.

[“Örnek: Kuyruktan tek bir dosyaya aktarma” sayfa 293](#)

fteCreateTransfer komutuyla **-sq** parametresini kullanarak dosya aktarımının kaynağı olarak bir IBM MQ kuyruğu belirtebilirsiniz.

[“Örnek: Bir ileti grubunun kuyruktan tek bir dosyaya aktarılması” sayfa 294](#)

fteCreateTransfer komutuyla **-sq** ve **-sqgi** parametrelerini kullanarak, dosya aktarımının kaynağı olarak IBM MQ kuyruğundaki tek bir tam grubu belirtebilirsiniz.

[“Örnek: Her iletideki verilerden önce bir metin sınırlayıcı ekleme” sayfa 296](#)

Bir kaynak kuyruktan bir dosyaya metin kipinde aktarma yaparken, **fteCreateTransfer** komutuyla **-sq**, **-sqdt** ve **-sqdp** parametrelerini kullanarak, tek tek iletilerdeki verilerden önce bir metin sınırlayıcı eklenmesini belirtebilirsiniz.

[“Kuyruğun izlenmesi ve değişken yerine koyma değerinin kullanılması” sayfa 261](#)

fteCreateMonitor komutunu kullanarak, bir kuyruğu izleyebilir ve iletileri izlenen kuyruktan bir dosyaya aktarabilirsiniz. İzlenen kuyruktan okunacak ilk iletideki herhangi bir IBM MQ ileti özelliğinin değeri, görev XML tanımlamasında yerine konabilir ve aktarma davranışını tanımlamak için kullanılabilir.

[“Örnek: IBM MQ ileti özellikleri kullanılarak iletiden dosyaya aktarma başarısız” sayfa 300](#)

usr. UserReturnCode IBM MQ ileti özelliğini sıfır dışında bir değere ayarlayarak bir iletinin dosya aktarımının başarısız olmasına neden olabilir. usr. UserSupplement IBM MQ ileti özelliğini ayarlayarak, başarısızlığın nedenine ilişkin ek bilgileri de belirtebilirsiniz.

İlgili başvurular

[fCreateTransfer](#): yeni bir dosya aktarımı başlatır

Kuyruğun izlenmesi ve deęişken yerine koyma deęerinin kullanılması

fteCreateMonitor komutunu kullanarak, bir kuyruęu izleyebilir ve iletileri izlenen kuyruktan bir dosyaya aktarabilirsiniz. İzlenen kuyruktan okunacak ilk iletideki herhangi bir IBM MQ ileti özelliğinin deęeri, görev XML tanımlamasında yerine konabilir ve aktarma davranışını tanımlamak için kullanılabilir.

Bu görev hakkında

Bu örnekte, kaynak aracıya QM_VENUS ' a baęlanan AGENT_VENUS adı verilir. AGENT_VENUS tarafından izlenen kuyruęa START_QUEUE adı verilir ve QM_VENUS üzerinde bulunur. Aracı, her 30 dakikada bir kuyruęa yoklama yapar.

Kuyruęa tam bir ileti grubu yazıldığında, izleme görevi ileti grubunu hedef araçlardan birindeki bir dosyaya gönderir; bunların tümü QM_MARS kuyruk yöneticisine baęlanır. İleti grubunun aktarıldığı dosyanın adı, gruptaki ilk iletide IBM MQ ileti özelliği `usr.fileName` tarafından tanımlanır. İleti grubunun gönderildiği aracının adı, gruptaki ilk iletide IBM MQ ileti özelliği `usr.toAgent` tarafından tanımlanır. `usr.toAgent` üstbilgi ayarlanmazsa, hedef aracı için kullanılacak varsayılan deęer AGENT_MAGENTA olur.

`useGroups="true"` deęerini belirlediğinizde, `groupId="{GROUPID}"` deęerini de belirlemezseniz, aktarma kuyruktaki ilk iletiyi alır. Örneğin, `fileName` dosyasını oluşturmak için deęişken yerine koyma deęeri kullanıyorsanız, `a.txt` içeriğinin doğru olmaması olasıdır. Bunun nedeni, `fileName` adlı dosyanın izleme programı tarafından oluşturulduğu, ancak aktarma işleminin gerçekte `fileName` adlı dosyayı oluşturmaması gereken bir iletiyi aldığı içindir.

Yordam

1. İzleme programının tetiklendiğinde gerçekleştirdiği görevi tanımlayan görev XML dosyasını yaratın.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="{toAgent}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/{fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

IBM MQ ileti üstbilgilerinin deęerleriyle deęiştirilen deęişkenler **koyu** olarak vurgulanır. Bu görev XML ' i /home/USER1/task.xml dosyasına kaydedildi

2. START_QUEUE kuyruęunu izlemek için bir kaynak izleme programı yaratın.

Şu komutu verin:

```
fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA
```

3. Bir kullanıcı ya da program, bir grup iletiyi START_QUEUE kuyruęuna yazar.

Bu gruptaki ilk iletide aşağıdaki IBM MQ ileti özellikleri ayarlanmıştır:

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. İzleme programı, tam grup yazıldığında tetiklenir. Aracı, IBM MQ ileti özelliklerini görev XML ' ine değiştirir.

Bu, görev XML ' inin aşağıdakilere dönüştürülmesine neden oluyor:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/larmer.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Sonuçlar

Görev XML ' i tarafından tanımlanan aktarma gerçekleştirilir. AGENT_VENUS tarafından START_QUEUE 'dan okunan iletilerin tam grubu, AGENT_VIOLET' in çalıştığı sistemde /reports/larmer.rpt adlı bir dosyaya yazılır.

Sonraki adım

Her iletinin ayrı bir dosyaya aktarılması

Bir kuyruğu izlemek ve her iletinin ayrı bir dosyaya aktarılmasını istiyorsanız, bu konuda daha önce anlatılana benzer bir teknik kullanabilirsiniz.

1. **fteCreateMonitor** komutunda **-tr completeGroups** değiştirgesini belirterek, izleme programını daha önce açıklandığı gibi yaratın.
2. Görev XML ' inde aşağıdakileri belirtin:

```
<queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
```

Ancak, iletileri kaynak kuyruğa koyduğunuzda, bunları bir IBM MQ grubuna koymayın. Her iletiye IBM MQ ileti özellikleri ekleyin. Örneğin, `usr.fileName` özelliğini her ileti için benzersiz bir dosya adı değeriyle belirtin. Bu, Managed File Transfer Agent ' in kaynak kuyruktaki her iletiyi ayrı bir grup olarak kabul etmesine neden olur.

İlgili kavramlar

“İletilerden dosyalara veri aktarılması” sayfa 291

Managed File Transfer iletiden dosyaya özelliği, IBM MQ kuyruğundaki bir ya da daha çok iletidensosyaya, veri kümesine (z/OSüzerindeki) ya da kullanıcı dosya alanına veri aktarmanızı sağlar. IBM MQ iletilerini yaratan ya da işleyen bir uygulamanız varsa, bu iletileri Managed File Transfer ağınzdaki herhangi bir sistemde bulunan bir dosyaya aktarmak için Managed File Transfer iletiden dosyaya aktarma yeteneğini kullanabilirsiniz.

“MFT kaynak izleyicisi görevlerinin değişken yerine koyma değeri ile uyarlanması” sayfa 255
Etkin bir kaynak izleme programının tetikleme koşulları karşılandığında, tanımlanan görev çağrılır. Aktarma ya da komut görevini her seferinde aynı hedef aracıyla ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımlamasını da değiştirebilirsiniz. Bunu, görev tanımlaması XML ' ine değişken adları ekleyerek yaparsınız. İzleme programı tetikleyici koşullarının karşılandığını ve görev tanımlamasının değişken adlarını içerdiğini saptadığında, değişken adlarını değişken değerleriyle değiştirir ve görevi çağırır.

Bir kuyruk kaynak izleme programı tarafından başlatılan bir aktarma tarafından yaratılan hedef dosyalar yanlış veri içeriyorsa ne yapmanız gerekir?

İlgili görevler

“Komutları ve komut dosyalarını başlatmak için MFT izleme görevlerinin yapılandırılması” sayfa 249
Kaynak izleme programları, ilişkili görev olarak dosya aktarımlarını gerçekleştirmekle sınırlı değildir. İzleme programını, yürütülebilir programlar, Ant komut dosyaları ya da JCL işleri de içinde olmak üzere, izleme aracısından diğer komutları çağırarak de yapılandırabilirsiniz. Komutları çağırarak için, izleme görevi tanımlaması XML dosyasını, bağımsız değişkenler ve özellikler gibi ilgili komut çağırısı değiştirgelerine sahip bir ya da daha çok komut ögesi içerecek şekilde düzenleyin.

“Örnek: MFT kaynağını yapılandırma” sayfa 255

Bir kaynak izleme programı tarafından izlenecek kaynak olarak IBM MQ kuyruğunu, **fteCreateMonitor** komutuyla **-mq** değiştirgesini kullanarak belirtebilirsiniz.

İlgili başvurular

fteCreateMonitor: MFT kaynak izleme programı yaratılması

Kaynak kuyruklardaki iletilerden MFT tarafından okunan MQ ileti özellikleri

Örnek: IBM MQ ileti özellikleri kullanılarak iletiden dosyaya aktarma başarısız

`usr.UserReturnCode` IBM MQ ileti özelliğini sıfır dışında bir değere ayarlayarak bir iletinin dosya aktarımının başarısız olmasına neden olabilir. `usr.UserSupplement` IBM MQ ileti özelliğini ayarlayarak, başarısızlığın nedenine ilişkin ek bilgileri de belirtebilirsiniz.

Bu görev hakkında

Bu örnekte, `INPUT_QUEUE` kuyruğu ile `/home/user/output.file` dosyası arasında bir aktarma işlemi devam etmektedir.

Bir kullanıcı ileti yaratıyor ve bunları `INPUT_QUEUE` kuyruğuna yerleştiriyor. Kaynak aracı `INPUT_QUEUE` kuyruğundaki iletileri tüketiyor ve aktarma verilerini hedef aracıya gönderiyor. Hedef aracı bu verileri `/home/user/output.file` dosyasına yazıyor.

`INPUT_QUEUE` kuyruğuna ileti yazan kullanıcı, devam eden aktarımı durdurmak ve hedef dosyaya önceden yazılmış verileri silmek istiyor.

Yordam

1. Kullanıcı, aşağıdaki IBM MQ ileti özellikleri ayarlanmış olan `INPUT_QUEUE` kuyruğuna bir ileti yazar:

```
usr.UserReturnCode=1
usr.UserSupplement="Cancelling transfer - sent wrong data."
```

2. Kaynak aracı, IBM MQ ileti özelliklerini okur ve kuyruktaki iletileri işlemeyi durdurur. Hedef aracı, hedef dizine yazılan tüm dosya verilerini siler.
3. Kaynak aracı, koordinasyon kuyruk yöneticisine aktarma hatasını bildiren bir aktarım günlüğü iletisi gönderir.

İleti aşağıdaki bilgileri içerir:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
```


İlgili başvurular

[Kaynak kuyruklardaki iletilerden MFT tarafından okunan MQ ileti özellikleri](#)

Protokol köprüsü

İletişim kuralı köprüsü, Managed File Transfer (MFT) ağınızın, yerel etki alanınızda ya da uzak bir konumda MFT ağınızın dışındaki bir dosya sunucusunda saklanan dosyalara erişmesini sağlar. Bu dosya sunucusu FTP, FTPS ya da SFTP ağ iletişim kurallarını kullanabilir. Her dosya sunucusu için en az bir özel olarak ayrılmış aracı gerekir. Özel olarak ayrılmış ajan, protokol köprüsü aracısı olarak bilinir. Bir köprü aracısı birden çok dosya sunucusuyla etkileşimde bulunabilir.

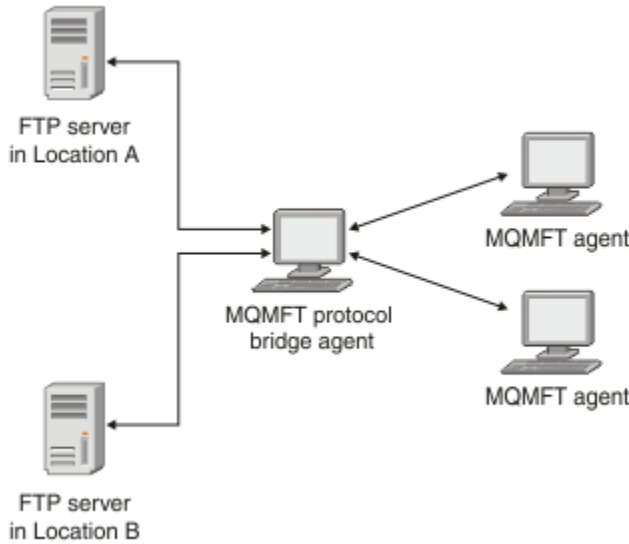
İletişim kuralı köprüsü, Managed File Transfer' in Hizmet bileşeninin bir parçası olarak kullanılabilir. Farklı dosya sunucularına bağlanan MFT işletim sistemini çalıştıran tek bir sistemde birden çok özel olarak ayrılmış aracınız olabilir.

Dosyaları aynı anda birden çok uç noktaya aktarmak için bir protokol köprüsü aracısını kullanabilirsiniz. MFT , dosyaları aktarmak istediğiniz farklı iletişim kuralı dosya sunucularını tanımlamak için düzenleyebileceğiniz `ProtocolBridgeProperties.xml` adlı bir dosya sağlar. **fteCreateBridgeAgent** komutu, varsayılan protokol kütüğü sunucusunun ayrıntılarını `ProtocolBridgeProperties.xml` ' e sizin için ekler. Bu dosya, [Protokol köprüsü özellikleri dosya biçimi](#) ' de açıklanmıştır.

Protokol köprüsü aracısını kullanarak aşağıdaki işlemleri gerçekleştirebilirsiniz:

- FTP, FTPS ya da SFTP kullanarak MFT ağındaki dosyaları uzak bir sunucuya yükleyin.
- FTP, FTPS ya da SFTP kullanarak uzak sunucudan MFT ağına dosya yükleme

Not: Protokol köprüsü aracısı yalnızca dosyaların mutlak dosya yoluyla erişmesine izin veren FTP, FTPS ya da SFTP sunucularını destekleyebilir. Bir aktarma isteğinde görel bir dosya yolu belirtilirse, protokol köprüsü aracısı, protokol sunucusunda oturum açmak için kullanılan ana dizine dayalı olarak, görel yolu mutlak bir dosya yoluna dönüştürmeyi dener. Yalnızca geçerli dizine dayalı dosyalara erişime izin veren protokol sunucuları, protokol köprüsü aracısı tarafından desteklenmez.



Çizge, farklı konumlarda iki FTP sunucusunu gösterir. FTP sunucuları, Managed File Transfer aracılıyla dosya değiş tokuşu için kullanılır. İletişim kuralı köprü aracısı, FTP sunucuları ile MFT ağının geri kalanı arasındadır ve her iki FTP sunucusuyla da iletişim kuracak şekilde yapılandırılır.

İletişim kuralı köprüsü aracısına ek olarak MFT ağınızda başka bir aracının olduğundan emin olun. Protokol köprüsü aracısı yalnızca FTP, FTPS ya da SFTP sunucusu için bir köprüdür ve aktarılan dosyaları yerel diske yazmaz. Dosyaları FTP, FTPS ya da SFTP sunucusundan ya da FTP ' den aktarmak istiyorsanız, dosya aktarımının hedefi ya da kaynağı olarak (FTP, FTPS ya da SFTP sunucusunu temsil eden) protokol köprüsü aracısını ve ilgili kaynak ya da hedef olarak başka bir standart aracını kullanmanız gerekir.

Protokol köprüsünü kullanarak dosya aktardığınızda, köprünün aktarmak istediğiniz dosyaları içeren kaynak ya da hedef dizini okuma izni olmalıdır. Örneğin, yalnızca yürütme izinleri (d -- x -- x -- x) olan /home/fte/bridge dizininden dosya aktarmak istiyorsanız, bu dizinden gerçekleştirmeyi denediğiniz aktarımlar aşağıdaki hata iletilisiyle başarısız olur:

```
BFGBR0032E: Attempt to read filename from the protocol file server
has failed with server error 550. Failed to open file.
```

Protokol köprüsü aracısının yapılandırılması

Protokol köprüsü aracısı, standart bir MFT aracısı gibidir. **fteCreateBridgeAgent** komutunu kullanarak bir protokol köprüsü aracısı oluşturun. Bir protokol köprüsü aracısını, [Protokol köprüsü özellikleri dosyası](#) biçiminde açıklanan `ProtocolBridgeProperties.xml` dosyasını kullanarak yapılandırabilirsiniz. Daha önceki bir sürümü kullanıyorsanız, aracıyı [Gelişmiş aracı özellikleri: Protokol köprüsü ve Gelişmiş aracı özellikleri: Protokol köprüsü aracısı günlük kaydı](#) içinde açıklanan belirli iletişim kuralı köprüsü özelliklerini kullanarak yapılandırın. Tüm sürümler için, [“Dosya sunucusuna ilişkin kimlik bilgilerini eşleme” sayfa 310](#) içinde açıklandığı gibi bir kimlik bilgisi eşlemesi de yapılandırabilirsiniz. Belirli bir iletişim kuralı dosya sunucusu için bir iletişim kuralı köprüsü aracısı yapılandırdıktan sonra, bu aracıyı yalnızca bu amaçla kullanabilirsiniz.

Protokol köprüsünün kurtarılması

Dosya sunucusu kullanılmadığı için protokol köprüsü aracısı dosya sunucusuna bağlanamıyorsa, dosya sunucusu kullanılabilir oluncaya kadar tüm dosya aktarma istekleri kuyruğa alınır. Protokol köprüsü aracısı, aracı yanlış kimlik bilgilerini kullandığı için dosya sunucusuna bağlanamıyorsa, aktarma başarısız olur ve aktarma günlüğü iletilisi bu hatayı yansıtır. Protokol köprüsü aracısı herhangi bir nedenle sona erdirilirse, istenen tüm dosya aktarımları korunur ve protokol köprüsü yeniden başlatıldığında devam eder.

Dosya aktarımı sırasında, dosyalar genellikle hedefte geçici dosyalar olarak yazılır ve aktarma tamamlandığında yeniden adlandırılır. Ancak, aktarım hedefi sınırlı yazma olarak yapılandırılan bir iletişim kuralı dosya sunucusuysa (kullanıcılar dosyaları iletişim kuralı dosya sunucusuna yükleyebilir, ancak karşıya yüklenen dosyaları herhangi bir şekilde değiştiremez; etkin bir şekilde kullanıcılar yalnızca bir kez yazabilir), aktarılan dosyalar doğrudan hedefe yazılır. Başka bir deyişle, aktarma sırasında bir sorun oluşursa, kısmen yazılan dosyalar hedef iletişim kuralı dosya sunucusunda kalır ve Managed File Transfer bu dosyaları silemez ya da düzenleyemez. Bu durumda transfer başarısız olur.

İlgili görevler

“Örnek: Bir iletişim kuralı köprüsü aracısının UNIX SFTP sunucusuyla özel anahtar kimlik bilgilerini kullanacak şekilde yapılandırılması” [sayfa 315](#)

Bu örnek, `ProtocolBridgeCredentials.xml` dosyasını nasıl oluşturabileceğinizi ve yapılandırabileceğinizi gösterir. Bu örnek tipik bir örnektir ve ayrıntılar platformunuza göre değişebilir, ancak ilkeler aynı kalır.

“`ProtocolBridgeProperties.xml` dosyasını kullanarak iletişim kuralı dosya sunucularına ilişkin özellikleri tanımlama” [sayfa 304](#)

Dosyaları aracı yapılandırma dizininde Managed File Transfer tarafından sağlanan `ProtocolBridgeProperties.xml` dosyasına aktarmak ve bu dosyayı kullanarak aktarmak istediğiniz bir ya da daha fazla iletişim kuralı dosya sunucusunun özelliklerini tanımlayın.

İlgili başvurular

[fteCreateBridgeAgent \(MFT iletişim kuralı köprüsü aracısı oluşturma ve yapılandırma\)](#)

“[Dosya sunucusuna ilişkin kimlik bilgilerini eşleme](#)” [sayfa 310](#)

Managed File Transfer içindeki kullanıcı kimlik bilgilerini, protokol köprüsü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanarak ya da kendi kullanıcı çıkışınızı yazarak dosya sunucusundaki kullanıcı kimlik bilgileriyle eşleyin. Managed File Transfer , kullanıcı kimlik bilgileri eşlemesi gerçekleştiren örnek bir kullanıcı çıkışı sağlar.

[ProtocolBridgeCredentialExit.java](#) arabirimi

[Örnek iletişim kuralı köprüsü kimlik bilgileri kullanıcı çıkışı](#)

[Protokol köprüsüne göre FTPS sunucusu desteği](#)

ProtocolBridgeProperties.xml dosyasını kullanarak iletişim kuralı dosya sunucularına ilişkin özellikleri tanımlama

Dosyaları aracı yapılandırma dizininde Managed File Transfer tarafından sağlanan ProtocolBridgeProperties.xml dosyasına aktarmak ve bu dosyayı kullanarak aktarmak istediğiniz bir ya da daha fazla iletişim kuralı dosya sunucusunun özelliklerini tanımlayın.

Bu görev hakkında

ftCreateBridgeAgent komutu, aracı yapılandırma dizininde `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` ProtocolBridgeProperties.xml dosyasını oluşturur. Komut çalıştırıldığında bir varsayılan değer belirlenmişse, komut, varsayılan protokol dosya sunucusu için dosyada bir giriş de yaratır.

BFGCL0392I iletisi, ProtocolBridgeProperties.xml dosyasının yerini belirtir.

```
<?xml version="1.0" encoding="IBM-1047"?>
<!--
This ProtocolBridgeProperties.xml file determines the protocol servers that will be accessed by
the
MQMFT protocol bridge agent.

Each protocol server is defined using either a <tns:ftpServer>, <tns:ftpsServer>, or
<tns:sftpServer>
element - depending on the protocol used to communicate with the server. When the protocol
bridge agent participates in a managed file transfer it will determine which server to used
based on
the prefix (if any) present on the file path. For example a file path of 'server1:/home/user/
file.txt' would
be interpreted as a request to transfer /home/user/file.txt using 'server1'. The server name
is compared
to the 'name' attribute of each <tns:ftpServer>, <tns:ftpsServer> or <tns:sftpServer> element
in this
XML document and the first match is used to determine which protocol server the protocol bridge
agent will connect to. If no match is found then the managed file transfer operation will fail.

If a file path is not prefixed with a server name, for example '/home/user/file.txt' then this
XML
document can specify a default server to use for the managed file transfer. To specify a
default server use the <tns:defaultServer> element as the first element inside the
<tns:serverProperties>
element. The default server will be used whenever the protocol bridge agent participates in
a managed file transfer for file names which do not specify a prefix.

An optional <tns:limits> element can be specified within each server definition. This element
contains
attributes that govern the amount of resources used by each defined server.

An optional <tns:credentialsFile> element can be specified within each serverProperties
definition. This
element contains a path to a file containing credentials to be used when connecting to defined
servers.

An example ProtocolBridgeProperties.xml file is as follows:

<?xml version="1.0" encoding="UTF-8"?>
<tns:serverProperties xmlns:tns="http://wmmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">

  <tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />

  <tns:defaultServer name="myFTPserver" />

  <tns:ftpServer name="myFTPserver" host="windows.hursley.ibm.com" port="1234"
platform="windows"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
  listFormat="unix" limitedWrite="false">

    <tns:limits maxListFileNames="100" maxListDirectoryLevels="999999999"
      maxReconnectRetry="2" reconnectWaitPeriod="10"
      maxSessions="60" socketTimeout="30" />

  </tns:ftpServer>
```



```

    <tns:ftpsServer name="myFTPSserver" host="unix.hursley.ibm.com" platform="unix"
      timeZone="Europe/London" locale="en_GB" fileEncoding="UTF8"
      listFormat="unix" limitedWrite="false" ftpsType="explicit"
      trustStore="C:\FTE\keystores\myFTPSserver\FTPSKeyStore.jks"
trustStorePassword="password">
      <tns:limits maxReconnectRetry="10" connectionTimeout="10"/>
    </tns:ftpsServer>
    <tns:sftpServer name="mySFTPserver" host="windows.hursley.ibm.com" platform="windows"
      timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
      limitedWrite="false">
      <tns:limits connectionTimeout="60"/>
    </tns:sftpServer>
  </tns:serverProperties>

```

This example shows the outermost <tns:serverProperties> element which must exist for the document to be valid, an optional <tns:defaultServer> element, as well as definitions for an FTP, FTPS and SFTP server.

The attributes of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements determine the characteristics of the connection established to the server. These attributes correspond to the command line parameters for the 'fteCreateBridgeAgent' command.

The following attributes are valid for all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: name, host, port, platform, fileEncoding, limitedWrite and controlEncoding.

The following attributes are valid for the <tns:ftpServer> and <tns:ftpsServer> elements: timeZone, locale, listFormat, listFileRecentDateFormat, listFileOldDateFormat, and monthShortNames.

The following attributes are valid for the <tns:ftpServer> element only: passiveMode

The following attributes are valid for the <tns:ftpsServer> element only: ftpsType, trustStore, trustStorePassword, trustStoreType, keyStore, keyStorePassword, keyStoreType, ccc, protFirst, auth, and connectTimeout.

The following attributes are valid for the <tns:limits> element within all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: maxListFileNames, maxListDirectoryLevels, maxReconnectRetry, reconnectWaitPeriod, maxSessions and socketTimeout

```

-->
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
ProtocolBridgeProperties.xsd">
  <!-- By default the location of the credentials file is in the home directory of the user
that started the -->
  <!-- protocol bridge agent. If you wish to specify a different location use the
credentialsFile element to -->
  <!-- do this. For
example:
  <!-- <tns:credentialsFile path="/test/
ProtocolBridgeCredentials.xml"/>
  <tns:defaultServer name="WINMVSCA.HURSLEY.IBM.COM" />
  <tns:ftpServer name="WINMVSCA.HURSLEY.IBM.COM" host="WINMVSCA.HURSLEY.IBM.COM"
platform="unix"
      timeZone="Europe/London" locale="en-GB" fileEncoding="US-ASCII"
      listFormat="unix" limitedWrite="false" />
  <!-- Define servers here -->
</tns:serverProperties>

```

Komut şu iletiyi üretebilir: BFGCL0532I:

Bu aracının çalışması için el ile bir ek kimlik bilgileri dosyası oluşturulmalıdır. Varsayılan olarak bu dosya ProtocolBridgeCredentials.xml olarak adlandırılır ve ana sayfada bulunur

Aracıyı başlatan kullanıcının dizini. Örneğin, bu kullanıcı aracıyı başlatmışsa konum: \$HOME/ProtocolBridgeCredentials.xml

Kimlik bilgileri dosyası kullanıyorsanız:

1. Nasıl yaratılacağı hakkında daha fazla bilgi için aşağıdaki metne bakın.
2. Kimlik bilgileri dosyası, sınırlı izinlere sahip bir dizinde olmalıdır. Örneğin, diğer kullanıcılar için okuma erişimi olmamalıdır.
3. Başlatılan aracının kullanıcı kimliği için \$HOME ortam değişkeninde kimlik bilgileri dosyasının konumunu belirtin ya da ProtocolBridgeProperties.xml dosyasını düzenleyin ve konumu belirtin:

```
<tns:credentialsFile path="/test/ProtocolBridgeCredentials.xml"/>
```

Varsayılan olmayan başka protokol sunucuları eklemek istiyorsanız, özelliklerini tanımlamak için bu dosyayı düzenleyin. Bu örnek, ek bir FTP sunucusu ekler.

Not: Protokol köprüsü aracı dosya kilitlemeyi desteklemiyor. Bunun nedeni, Managed File Transfer 'in bir dosya sunucusunda dosya kilitleme mekanizmasını desteklememesidir.

Yordam

1. Aşağıdaki satırları dosyaya <tns:serverProperties>' un alt ögesi olarak ekleyerek bir iletişim kuralı dosya sunucusu tanımlayın:

```
<tns:ftpServer name="myserver" host="myhost.hursley.ibm.com" port="1234"
platform="windows"
timeZone="Europe/London" locale="en-GB" fileEncoding="UTF-8"
listFormat="unix" limitedWrite="false" >
<tns:limits maxListFileNames="10" maxListDirectoryLevels="500"/>
```

2. Daha sonra özniteliklerin değerini değiştirin:

- name , iletişim kuralı dosya sunucunuzun adıdır
- host , protokol kütüğü sunucusunun anasistem adı ya da IP adresidir.
- port , iletişim kuralı dosya sunucusunun kapı numarasıdır
- platform , iletişim kuralı dosya sunucusunun çalıştığı altyapıdır
- timeZone , iletişim kuralı dosya sunucusunun çalıştığı saat dilimidir
- locale , iletişim kuralı dosya sunucusunda kullanılan dildir
- fileEncoding , iletişim kuralı dosya sunucusunun karakter kodlamasıdır
- listFormat , protokol kütüğü sunucusundan döndürülen kütük listeleme biçimidir.
- limitedWrite , bir dosya sunucusuna yazılırken varsayılan kipin izlenip izlenmeyeceğini, geçici bir dosya yaratılıp yaratılmayacağını ve aktarma tamamlandığında bu dosyanın yeniden adlandırılıp adlandırılmayacağını belirler. Yalnızca yazma olarak yapılandırılan bir dosya sunucusu için, dosya doğrudan son adıyla yaratılır. Bu özelliğin değeri true ya da false olabilir. limitedWrite özniteliği ve doNotUseTempOutputFile aracı özelliği, protokol köprüsü araçları durumunda birlikte kullanılır. Geçici dosyaları kullanmak istiyorsanız, doNotUseTempOutputFiledeğerini ayarlamamanız ve limitedWrite değerini false olarak ayarlamamanız gerekir. Ayarların başka bir birleşimi, geçici dosyaların kullanılmayacağı anlamına gelir.
- maxListFileNames , protokol dosya sunucusundaki bir dizinde dosya adları taranırken toplanan ad sayısı üst sınırdır.
- maxListDirectoryLevels , protokol dosya sunucusundaki bir dizinde dosya adları taranırken yinelenen dizin düzeyi sayısı üst sınırdır.

Gerekli ya da isteğe bağlı olup olmadıkları ve varsayılan değerleri de içinde olmak üzere bu özniteliklere ilişkin daha fazla ayrıntı için bkz. [Protokol köprüsü özellikleri dosya biçimi](#).

İlgili başvurular

[Protokol köprüsü özellikleri dosya biçimi](#)

İletişim kuralı dosya sunucusu özellikleri arıyor: ProtocolBridgePropertiesExit2

Çok sayıda iletişim kuralı dosya sunucunuz varsa, aktarımlarda başvuru ile iletişim kuralı dosya sunucusu özelliklerini aramak için `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` arabirimini uygulayabilirsiniz. Bu arabirimi bir `ProtocolBridgeProperties.xml` dosyasını korumak yerine uygulayabilirsiniz.

Bu görev hakkında

Managed File Transfer , iletişim kuralı dosya sunucusu özelliklerini inceleyen örnek bir kullanıcı çıkışı sağlar. Daha fazla bilgi için bkz. "[Protokol dosya sunucusu özelliklerini aramak için örnek kullanıcı çıkışının kullanılması](#)" sayfa 307.

Protokol köprüsü özelliklerini arayan her kullanıcı çıkışı `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` arabirimini gerçekleştirmelidir. Daha fazla bilgi için bkz. `ProtocolBridgePropertiesExit.java` arabirimi.

Birden çok iletişim kuralı sunucusu özelliğini diğer kullanıcı çıkışlarına benzer şekilde zincirleyebilirsiniz. Çıkışlar, aracı özellikleri dosyasındaki `protocolBridgePropertiesExitClasses` özelliği kullanılarak belirtilen sırayla çağrılır. Başlatma yöntemlerinin tümü ayrı olarak döndürülür ve bir ya da daha fazla false değeri döndürürse aracı başlatılamaz. Hata, aracı olay günlüğünde bildirilir.

Tüm çıkışların `getProtocolServerProperties` yöntemleri için yalnızca bir genel sonuç döndürülür. Yöntem sonuç kodu olarak bir özellik nesnesi döndürürse, bu değer döndürülen sonuçtur ve sonraki çıkışların `getProtocolServerProperties` yöntemleri çağrılmaz. Yöntem sonuç kodu olarak boş değer döndürürse, sonraki çıkışın `getProtocolServerProperties` yöntemi çağrılır. Sonraki bir çıkış yoksa, boş sonuç döndürülür. Genel sonuç kodu boş değer, protokol köprüsü aracısı tarafından bir arama hatası olarak kabul edilir.

`ProtocolBridgePropertiesExit2.java` arabirimini kullanmanız önerilir, ancak `ProtocolBridgePropertiesExit.java` arabirimi hakkında bilgi için bkz. "[İletişim kuralı dosya sunucusu özellikleri arıyor: ProtocolBridgePropertiesExit](#)" sayfa 309.

Çıkışınızı çalıştırmak için aşağıdaki adımları tamamlayın:

Yordam

1. Protokol sunucusu özellikleri kullanıcı çıkışını derleyin.
2. Derlenmiş çıkışı ve paket yapısını içeren bir Java arşiv (JAR) dosyası yaratın.
3. Çıkış sınıfını içeren JAR dosyasını, protokol köprüsü aracısının `exits` dizinine koyun. Bu dizin, `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` dizininde bulunur.
4. İletişim kuralı köprüsü aracısının özellik dosyasını `protocolBridgePropertiesExitClasses` özelliğini içerecek şekilde düzenleyin. Bu özelliğin değeri için, protokol köprüsü sunucusu özellikleri kullanıcı çıkışını gerçekleştiren sınıfların virgülle ayrılmış bir listesini belirtin. Çıkış sınıfları, bu listede belirttikleri sırayla çağrılır. Daha fazla bilgi için bkz. `MFT agent.properties` dosyası.
5. İsteğe bağlı olarak `protocolBridgePropertiesConfiguration` özelliğini belirtebilirsiniz. Bu özellik için belirttiğiniz değer, `protocolBridgePropertiesExitClasses` ile belirtilen çıkış sınıflarının `initialize()` yöntemine Dizgi olarak geçirilir. Daha fazla bilgi için bkz. `MFT agent.properties` dosyası.

Protokol dosya sunucusu özelliklerini aramak için örnek kullanıcı çıkışının kullanılması

Managed File Transfer , iletişim kuralı dosya sunucusu özelliklerini inceleyen örnek bir kullanıcı çıkışı sağlar.

Bu görev hakkında

`MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` dizininde ve [Örnek protokol köprüsü özellikleri kullanıcı çıkışı](#) konusunda, protokol köprüsü özelliklerini inceleyen örnek bir kullanıcı çıkışı sağlanır.

`SamplePropertiesExit2.java` çıkışı, protokol sunucularına ilişkin özellikleri içeren bir özellikler dosyasını okur. Özellikler dosyasındaki her bir girdinin biçimi şöyledir:

```
serverName=type://host:port
```

Özellikler dosyasının konumu, `protocolBridgePropertiesConfiguration` protokol köprüsü aracı özelliğinden alınır.

Örnek kullanıcı çıkışını çalıştırmak için aşağıdaki adımları tamamlayın:

Yordam

1. `SamplePropertiesExit2.java` dosyasını derleyin.
2. Derlenen çıkışı ve paket yapısını içeren bir JAR dosyası yaratın.
3. JAR dosyasını `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/exits` dizinine koyun.
4. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` dosyasını, satırı içerecek şekilde düzenleyin:

```
protocolBridgePropertiesExitClasses=SamplePropertiesExit2
```

5. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` dizininde bir protokol köprüsü özellikler dosyası (örneğin, `protocol_bridge_properties.properties`) oluşturun. Bu dosyayı, girdileri şu biçimde içerecek şekilde düzenleyin:

```
serverName=type://host:port
```

6. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties` dosyasını, satırı içerecek şekilde düzenleyin:

```
protocolBridgePropertiesConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/protocol_bridge_properties.properties
```

`protocol_bridge_properties.properties` dosyasının mutlak yolunu kullanmalısınız.

7. **fteStartAgent** komutunu kullanarak protokol köprüsü aracısını başlatın.

İlgili kavramlar

[“Protokol köprüsü” sayfa 302](#)

İletişim kuralı köprüsü, Managed File Transfer (MFT) ağınızın, yerel etki alanınızda ya da uzak bir konumda MFT ağınızın dışındaki bir dosya sunucusunda saklanan dosyalara erişmesini sağlar. Bu dosya sunucusu FTP, FTPS ya da SFTP ağ iletişim kurallarını kullanabilir. Her dosya sunucusu için en az bir özel olarak ayrılmış aracı gerekir. Özel olarak ayrılmış ajan, protokol köprüsü aracı olarak bilinir. Bir köprü aracı birden çok dosya sunucusuyla etkileşimde bulunabilir.

İlgili başvurular

[ProtocolBridgePropertiesExit.java arabirimi](#)

[Örnek protokol köprüsü özellikleri kullanıcı çıkışı](#)

[MFT agent.properties dosyası](#)

[fteCreateBridgeAgent \(MFT iletişim kuralı köprüsü aracı oluşturma ve yapılandırma\)](#)

İletişim kuralı dosya sunucusu özellikleri aranıyor: ProtocolBridgePropertiesExit

Çok sayıda iletişim kuralı dosya sunucunuz varsa, aktarımlarda başvuru ile iletişim kuralı dosya sunucusu özelliklerini aramak için `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` arabirimini uygulayabilirsiniz.

Bu görev hakkında

Bir `ProtocolBridgeProperties.xml` dosyasını korumak yerine `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` arabirimini uygulayabilirsiniz. `ProtocolBridgePropertiesExit2.java` arabirimini kullanın. `ProtocolBridgePropertiesExit2.java` içindeki **getCredentialLocation** yöntemi, ana dizininiz olan `ProtocolBridgeCredentials.xml` dosyasının varsayılan konumunu kullanır.

Protokol köprüsü özelliklerini arayan her kullanıcı çıkışı `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` arabirimini gerçekleştirmelidir:

```
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;
import java.util.Properties;

/**
 * An interface that is implemented by classes that are to be invoked as part of
 * user exit routine processing. This interface defines methods that will be
 * invoked by a protocol bridge agent to look up properties for protocol servers
 * that are referenced in transfers.
 * <p>
 * There will be one instance of each implementation class for each protocol
 * bridge agent. The methods can be called from different threads so the methods
 * must be synchronised.
 */
public interface ProtocolBridgePropertiesExit {

    /**
     * Invoked once when a protocol bridge agent is started. It is intended to
     * initialize any resources that are required by the exit.
     *
     * @param bridgeProperties
     *     The values of properties defined for the protocol bridge.
     *     These values can only be read, they cannot be updated by the
     *     implementation.
     * @return {@code true} if the initialization is successful and {@code
     *     false} if unsuccessful. If {@code false} is returned from an exit
     *     the protocol bridge agent will not start.
     */
    public boolean initialize(final Map<String, String> bridgeProperties);

    /**
     * Obtains a set of properties for the specified protocol server name.
     * <p>
     * The returned {@link Properties} must contain entries with key names
     * corresponding to the constants defined in
     * {@link ProtocolServerPropertyConstants} and in particular must include an
     * entry for all appropriate constants described as required.
     *
     * @param protocolServerName
     *     The name of the protocol server whose properties are to be
     *     returned. If a null or a blank value is specified, properties
     *     for the default protocol server are to be returned.
     * @return The {@link Properties} for the specified protocol server, or null
     *     if the server cannot be found.
     */
    public Properties getProtocolServerProperties(
        final String protocolServerName);

    /**
     * Invoked once when a protocol bridge agent is shut down. It is intended to
     * release any resources that were allocated by the exit.
     *
     * @param bridgeProperties
     *     The values of properties defined for the protocol bridge.
     *     These values can only be read, they cannot be updated by the
     *     implementation.
     */
}
```

```
*/  
public void shutdown(final Map<String, String> bridgeProperties);  
}
```

Birden çok iletişim kuralı sunucusu özelliğini diğer kullanıcı çıkışlarına benzer bir şekilde zincirleyebilirsiniz. Çıkışlar, aracı özellikleri dosyasındaki `protocolBridgePropertiesExitClasses` özelliği kullanılarak belirtilen sırayla çağrılır. Başlatma yöntemlerinin tümü ayrı olarak döndürülür ve bir ya da daha fazla false değeri döndürürse aracı başlatılamaz. Hata, aracı olay günlüğünde bildirilir.

Tüm çıkışların `getProtocolServerProperties` yöntemleri için yalnızca bir genel sonuç döndürülür. Yöntem sonuç kodu olarak bir özellikler nesnesi döndürürse, bu değer döndürülen sonuçtur ve sonraki çıkışların `getProtocolServerProperties` yöntemleri çağrılmaz. Yöntem sonuç kodu olarak boş değer döndürürse, sonraki çıkışın `getProtocolServerProperties` yöntemi çağrılır. Sonraki bir çıkış yoksa, boş sonuç döndürülür. Genel sonuç kodu boş değer, protokol köprüsü aracısı tarafından bir arama hatası olarak kabul edilir.

Yordam

Çıkışınızı çalıştırmak için aşağıdaki adımları tamamlayın:

1. Protokol sunucusu özellikleri kullanıcı çıkışını derleyin.

2. Derlenmiş çıkışı ve paket yapısını içeren bir Java arşiv (JAR) dosyası yaratın.

3. Çıkış sınıfını içeren JAR dosyasını, protokol köprüsü aracısının `exits` dizinine koyun.

Bu dizin, `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` dizininde bulunur.

4. İletişim kuralı köprüsü aracısının özellik dosyasını

`protocolBridgePropertiesExitClasses` özelliğini içerecek şekilde düzenleyin.

Bu özelliğin değeri için, protokol köprüsü sunucusu özellikleri kullanıcı çıkışını gerçekleştiren sınıfların virgülle ayrılmış bir listesini belirtin. Çıkış sınıfları, bu listede belirtildikleri sırayla çağrılır. Daha fazla bilgi için bkz. [MFT agent.properties dosyası](#).

5. İsteğe bağlı olarak `protocolBridgePropertiesConfiguration` özelliğini belirtebilirsiniz.

Bu özellik için belirttiğiniz değer, `protocolBridgePropertiesExitClasses` ile belirtilen çıkış sınıflarının `initialize ()` yöntemine Dizgi olarak geçirilir. Daha fazla bilgi için bkz. [MFT agent.properties dosyası](#).

Dosya sunucusuna ilişkin kimlik bilgilerini eşleme

Managed File Transfer içindeki kullanıcı kimlik bilgilerini, protokol köprüsü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanarak ya da kendi kullanıcı çıkışınızı yazarak dosya sunucusundaki kullanıcı kimlik bilgileriyle eşleyin. Managed File Transfer , kullanıcı kimlik bilgileri eşlemesi gerçekleştiren örnek bir kullanıcı çıkışı sağlar.

İlgili kavramlar

“Protokol köprüsü” sayfa 302

İletişim kuralı köprüsü, Managed File Transfer (MFT) ağınızın, yerel etki alanınızda ya da uzak bir konumda MFT ağınızın dışındaki bir dosya sunucusunda saklanan dosyalara erişmesini sağlar. Bu dosya sunucusu FTP, FTPS ya da SFTP ağ iletişim kurallarını kullanabilir. Her dosya sunucusu için en az bir özel olarak ayrılmış aracı gerekir. Özel olarak ayrılmış ajan, protokol köprüsü aracısı olarak bilinir. Bir köprü aracısı birden çok dosya sunucusuyla etkileşimde bulunabilir.

İlgili görevler

“[ProtocolBridgeCredentials.xml dosyasını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerini eşleme](#)” sayfa 311

Managed File Transfer içindeki kullanıcı kimlik bilgilerini, protokol köprüsü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanarak dosya sunucusundaki kullanıcı kimlik bilgileriyle eşleyin. Managed File Transfer , kimlik bilgilerinizi dahil etmek için düzenleyebileceğiniz bir XML dosyası sağlar.

“[Çıkış sınıflarını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerini eşleme](#)” sayfa 313

Protokol köprüsü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanmak istemiyorsanız, kendi kullanıcı çıkışınızı yazarak Managed File Transfer içindeki kullanıcı kimlik bilgilerini dosya sunucusundaki

kullanıcı kimlik bilgileriyle eşleyebilirsiniz. Kimlik bilgisi eşleme kullanıcı çıkışlarını yapılandırırsanız, bunlar varsayılan kimlik bilgisi eşleme işlevinin yerini alır.

“Örnek: Bir iletişim kuralı köprüsü aracısının UNIX SFTP sunucusuyla özel anahtar kimlik bilgilerini kullanacak şekilde yapılandırılması” sayfa 315

Bu örnek, `ProtocolBridgeCredentials.xml` dosyasını nasıl oluşturabileceğinizi ve yapılandırabileceğinizi gösterir. Bu örnek tipik bir örnektir ve ayrıntılar platformunuza göre değişebilir, ancak ilkeler aynı kalır.

İlgili başvurular

[ProtocolBridgeCredentialExit.java arabirimi](#)

[Örnek iletişim kuralı köprüsü kimlik bilgileri kullanıcı çıkışı](#)

[MFT agent.properties dosyası](#)

ProtocolBridgeCredentials.xml dosyasını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerini eşleme

Managed File Transfer içindeki kullanıcı kimlik bilgilerini, protokol köprüsü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanarak dosya sunucusundaki kullanıcı kimlik bilgileriyle eşleyin. Managed File Transfer , kimlik bilgilerinizi dahil etmek için düzenleyebileceğiniz bir XML dosyası sağlar.

Bu görev hakkında

`ProtocolBridgeCredentials.xml` dosyası kullanıcı tarafından el ile oluşturulmalıdır. Varsayılan olarak, bu dosyanın konumu, protokol köprüsü aracısını başlatan kullanıcının ana dizinidir, ancak bu, aracı tarafından erişilebilen dosya sisteminde herhangi bir yerde saklanabilir. Farklı bir konum belirtmek için, `<credentialsFile>` ögesini `ProtocolBridgeProperties.xml` dosyasına ekleyin. Örneğin,

```
<tns:credentialsFile path="/example/path/to/ProtocolBridgeCredentials.xml"/>
```

Bir protokol köprüsü aracısını kullanmadan önce, bu dosyayı anasistem, kullanıcı ve kimlik bilgilerini içerecek şekilde düzenleyerek kimlik bilgileri eşlemesini ayarlayın. Daha fazla bilgi ve örnekler için bkz. [Protokol köprüsü kimlik bilgileri dosyası biçimi](#).

Yordam

1. Ad özniteliğinin değerini `ProtocolBridgeProperties.xml` dosyasındaki sunucu adına değiştirmek için `<tns:server name="server name">` satırını düzenleyin.

Genel arama karakterleri ya da düzenli ifadeler içeren bir sunucu adı kullandığınızı belirtmek için örüntü özniteliğini kullanabilirsiniz. Örneğin,

```
<tns:server name="serverA*" pattern="wildcard">
```

2. `<tns:server>`' un alt öğeleri olarak dosyaya kullanıcı kimliği ve kimlik bilgisi bilgilerini ekleyin. Dosyaya aşağıdaki öğelerden birini ya da birkaçını ekleyebilirsiniz:

- İletişim kuralı dosya sunucusu bir FTP, FTPS ya da SFTP sunucusuysa, aktarım isteyen kullanıcının kimliğini doğrulamak için parolaları kullanabilirsiniz. Dosyaya aşağıdaki satırları ekleyin:

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  serverPassword="Server Password">
</tns:user>
```

Daha sonra özniteliklerin değerini değiştirin.

- name , MFT aktarma isteğiyle ilişkilendirilmiş MQMD kullanıcı kimliğiyle eşleşen bir Java düzenli ifadesidir.

- serverUserId , iletişim kuralı dosya sunucusuna oturum açma kullanıcı kimliği olarak geçirilen değerdir. serverUserId özniteliği belirtilmezse, bunun yerine MFT aktarma isteğiyle ilişkili MQMD kullanıcı kimliği kullanılır.
- serverPassword , serverUserId ile ilişkili paroladır.

name özniteliği bir Java düzenli ifadesi içerebilir. Kimlik bilgileri eşleyici, MFT aktarma isteğinin MQMD kullanıcı kimliğini bu düzenli ifadeyle eşleştirmeyi dener. Protokol köprüsü aracısı, < tns: user> öğelerinin ad özniteliğindeki MQMD kullanıcı kimliğini, öğelerin dosyada bulunma sırasına göre düzenli ifadeyle eşleştirmeyi dener. Bir eşleşme bulunduğunda, protokol köprüsü aracısı daha fazla eşleşme aramaz. Bir eşleşme bulunursa, ilgili serverUserId ve serverPassword değerleri iletişim kuralı dosya sunucusuna oturum açma kullanıcı kimliği ve parolası olarak geçirilir. MQMD kullanıcı kimliği eşleşmeleri büyük ve küçük harfe duyarlıdır.

- İletişim kuralı dosya sunucusu bir SFTP sunucusuysa, aktarma isteğinde bulunan kullanıcının kimliğini doğrulamak için genel ve özel anahtarları kullanabilirsiniz. Dosyaya aşağıdaki satırları ekleyin ve özniteliklerin değerini değiştirin. < tns: user> ögesi bir ya da daha fazla < tns:privateKey> ögesi içerebilir.

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  hostKey="Host Key">
  <tns:privateKey associationName="association"
    keyPassword="Private key password">
    Private key file text
  </tns:privateKey>
</tns:user>
```

- name , MFT aktarma isteğiyle ilişkilendirilmiş MQMD kullanıcı kimliğiyle eşleşen bir Java düzenli ifadesidir.
- serverUserId , iletişim kuralı dosya sunucusuna oturum açma kullanıcı kimliği olarak geçirilen değerdir. serverUserId özniteliği belirtilmezse, bunun yerine MFT aktarma isteğiyle ilişkili MQMD kullanıcı kimliği kullanılır.
- hostKey , oturum açılırken sunucudan döndürülen beklenen anahtardır
- key , serverUserId özel anahtardır
- keyPassword , ortak anahtarlar oluşturmak için kullanılan anahtarın parolasıdır
- associationName , izleme ve günlüğe kaydetme amacıyla kullanılan bir değerdir

name özniteliği bir Java düzenli ifadesi içerebilir. Kimlik bilgileri eşleyici, MFT aktarma isteğinin MQMD kullanıcı kimliğini bu düzenli ifadeyle eşleştirmeyi dener. Protokol köprüsü aracısı, < tns: user> öğelerinin ad özniteliğindeki MQMD kullanıcı kimliğini, öğelerin dosyada bulunma sırasına göre düzenli ifadeyle eşleştirmeyi dener. Bir eşleşme bulunduğunda, protokol köprüsü aracısı daha fazla eşleşme aramaz. Bir eşleşme bulunursa, iletişim kuralı dosya sunucusuyla MFT kullanıcısının kimliğini doğrulamak için ilgili serverUserId ve key değerleri kullanılır. MQMD kullanıcı kimliği eşleşmeleri büyük ve küçük harfe duyarlıdır.

Bir protokol köprüsü aracısıyla özel anahtarların kullanılmasıyla ilgili daha fazla bilgi için bkz. “Örnek: Bir iletişim kuralı köprüsü aracısının UNIX SFTP sunucusuyla özel anahtar kimlik bilgilerini kullanacak şekilde yapılandırılması” sayfa 315.

Not:

Aktarma isteği komut kuyruğuna yazıldığında, kaynak aracı komut kuyruğu bir z/OS ya da IBM i sisteminde olduğunda MQMD kullanıcı kimliği büyük harfe dönüştürülebilir. Sonuç olarak, aynı kaynak kullanıcıya ilişkin MQMD kullanıcı kimliği özgün vakada kimlik bilgileri çıkışına gelebilir ya da aktarma isteğinde belirtilen kaynak aracıya bağlı olarak büyük harfe dönüştürülebilir. Varsayılan kimlik bilgisi eşleme çıkışı, sağlanan MQMD kullanıcı kimliğiyle büyük ve küçük harfe duyarlı eşleşmeler gerçekleştirir; eşleme dosyasında buna izin vermeniz gerekebilir.

İlgili başvurular

[İletişim kuralı köprüsü kimlik bilgileri dosyası biçimi](#)

Çıkış sınıflarını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerini eşleme

Protokol köprüsü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanmak istemiyorsanız, kendi kullanıcı çıkışınızı yazarak Managed File Transfer içindeki kullanıcı kimlik bilgilerini dosya sunucusundaki kullanıcı kimlik bilgileriyle eşleyebilirsiniz. Kimlik bilgisi eşleme kullanıcı çıkışlarını yapılandırırsanız, bunlar varsayılan kimlik bilgisi eşleme işlevinin yerini alır.

Bu görev hakkında

Managed File Transfer , kullanıcı kimlik bilgileri eşlemesi gerçekleştiren örnek bir kullanıcı çıkışı sağlar. Daha fazla bilgi için bkz [“Örnek protokol köprüsü kimlik bilgisi kullanıcı çıkışının kullanılması” sayfa 314.](#)

Eşleme iletişim kuralı köprüsü kimlik bilgileri için bir kullanıcı çıkışı aşağıdaki arabirimlerden birini gerçekleştirmelidir:

- Bir protokol köprüsü aracısının bir varsayılan protokol dosyası sunucusuna/sunucusundan dosya aktarmasına izin veren `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`
- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2`, birden çok uç noktaya/noktadan dosya aktarmanızı sağlar.

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2` arabirimi, `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` ile aynı işlevi içerir ve genişletilmiş işlevi de içerir. Daha fazla bilgi için bkz. [ProtocolBridgeCredentialExit.java arabirimi](#) ve [ProtocolBridgeCredentialExit2.java arabirimi](#).

Kimlik bilgisi çıkışları, diğer kullanıcı çıkışlarına benzer şekilde zincirleme olarak bağlanabilir. Çıkışlar, aracı özellikleri dosyasındaki `protocolBridgeCredentialConfiguration` özelliği kullanılarak belirtilen sırayla çağrılır. Başlatma yöntemlerinin tümü ayrı olarak döndürülür ve bir ya da daha fazla false değeri döndürürse aracı başlatılamaz. Hata, aracı olay günlüğünde bildirilir.

Tüm çıkışların `mapMQUserantıcı` yöntemleri için aşağıdaki gibi tek bir genel sonuç döndürülür:

- Yöntem sonuç kodu olarak `USER_SUCCESSFULLY_EŞLENMİŞ` ya da `USER_DENIED_ACCESS` değerini döndürürse, bu değer döndürülen sonuçtur ve sonraki çıkışların `mapMQUserantıcı` yöntemleri çağrılmaz.
- Yöntem sonuç kodu olarak `NO_MAPPING_FOUND` değerini döndürürse, sonraki çıkışın `mqMQUserId` yöntemi çağrılır.
- Sonraki bir çıkış yoksa, `NO_MAPPING_FOUND` sonucu döndürülür.
- `UER_DENIED_ACCESS` ya da `NO_MAPPING_FOUND` genel sonuç kodu, köprü aracısı tarafından bir aktarma hatası olarak kabul edilir.

Çıkışınızı çalıştırmak için aşağıdaki adımları tamamlayın:

Yordam

1. Protokol köprüsü kimlik bilgisi kullanıcı çıkışını derleyin.
2. Derlenmiş çıkışı ve paket yapısını içeren bir Java arşiv (JAR) dosyası yaratın.
3. Çıkış sınıfını içeren JAR dosyasını köprü aracısının `exits` dizinine yerleştirin. Dizin, `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` dizinindedir.
4. İletişim kuralı köprüsü aracısının özellik dosyasını `protocolBridgeCredentialExitClasses` özelliğini içerecek şekilde düzenleyin. Bu özelliğin değeri için, protokol köprüsü kimlik bilgisi çıkış yordamını gerçekleştiren sınıfların virgülle ayrılmış bir listesini belirtin. Çıkış sınıfları, bu listede belirtildikleri sırayla çağrılır. Daha fazla bilgi için bkz. [MFT agent.properties dosyası](#).
5. İletişim kuralı köprüsü aracısının özellik dosyasını aşağıdakileri içerecek şekilde düzenleyin:

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

Bir aracıya ilişkin `agent.properties` dosyası `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/bridge_agent_name` dizininizde bulunur.

`agent.properties` dosyasını değiştirirseniz, değişiklikleri almak için aracıyı yeniden başlatmanız gerekir.

6. İsteğe bağlı olarak `protocolBridgeCredentialConfiguration` özelliğini belirtebilirsiniz. Bu özellik için belirttiğiniz değer, `protocolBridgeCredentialExitClasses` ile belirtilen çıkış sınıflarının `initialize ()` yöntemine `String` nesnesi olarak geçirilir. Daha fazla bilgi için bkz. [MFT agent.properties](#) dosyası.
7. İletişim kuralı köprüsü aracısını **`fteStartAgent`** komutuyla başlatın.

Örnek protokol köprüsü kimlik bilgisi kullanıcı çıkışının kullanılması

Managed File Transfer , kullanıcı kimlik bilgileri eşlemesi gerçekleştiren örnek bir kullanıcı çıkışı sağlar.

Bu görev hakkında

`MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` dizininde ve [Örnek iletişim kuralı köprüsü kimlik bilgisi kullanıcı çıkışı](#) konusunda örnek bir iletişim kuralı köprüsü kimlik bilgisi çıkışı sağlanır. Bu örnek, `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` arabirimine dayalıdır.

`SampleCredentialExit.java` çıkışı, sunucu kullanıcı kimliklerine ve sunucu parolalarına aktarma istekleriyle ilişkili MQMD kullanıcı kimliklerini eşleyen bir özellikler dosyasını okur. Özellikler dosyasının konumu, `protocolBridgeCredentialConfiguration` protokol köprüsü aracısı özelliğinden alınır.

Örnek kullanıcı çıkışını çalıştırmak için aşağıdaki adımları tamamlayın:

Yordam

1. `SampleCredentialExit.java` dosyasını derleyin.
2. Derlenmiş çıkışı ve paket yapısını içeren bir JAR dosyası yaratın.
3. JAR dosyasını `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/exits` dizinine yerleştirin.
4. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` dosyasını, satırı içerecek şekilde düzenleyin:

```
protocolBridgeCredentialExitClasses=SampleCredentialExit
```

5. İletişim kuralı köprüsü aracısının özellik dosyasını aşağıdakileri içerecek şekilde düzenleyin:

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

Bir aracıya ilişkin `agent.properties` dosyası `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` dizininizde bulunur.

`agent.properties` dosyasını değiştirirseniz, değişiklikleri almak için aracıyı yeniden başlatmanız gerekir.

6. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` dizininde bir kimlik bilgileri özellikleri dosyası (`credentials.properties`) oluşturun ve bu dosyayı aşağıdaki biçimde girdileri içerecek şekilde düzenleyin:

```
mqUserId=serverUserId,serverPassword
```

7. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` dosyasını, satırı içerecek şekilde düzenleyin:

```
protocolBridgeCredentialConfiguration=MQ_DATA_PATH/mqft/  
config/coordination_queue_manager/agents/bridge_agent_name/credentials.properties
```

`credentials.properties` dosyasının mutlak yolunu kullanmalısınız.

8. **fteStartAgent** komutunu kullanarak protokol köprüsü aracısını başlatın.

İlgili kavramlar

[“Protokol köprüsü” sayfa 302](#)

İletişim kuralı köprüsü, Managed File Transfer (MFT) ağınızın, yerel etki alanınızda ya da uzak bir konumda MFT ağınızın dışındaki bir dosya sunucusunda saklanan dosyalara erişmesini sağlar. Bu dosya sunucusu FTP, FTPS ya da SFTP ağ iletişim kurallarını kullanabilir. Her dosya sunucusu için en az bir özel olarak ayrılmış aracı gerekir. Özel olarak ayrılmış ajan, protokol köprüsü aracısı olarak bilinir. Bir köprü aracısı birden çok dosya sunucusuyla etkileşimde bulunabilir.

İlgili başvurular

[ProtocolBridgeCredentialExit.java arabirimi](#)

[ProtocolBridgeCredentialExit2.java arabirimi](#)

[Örnek iletişim kuralı köprüsü kimlik bilgileri kullanıcı çıkışı](#)

[MFT agent.properties dosyası](#)

[fteCreateBridgeAgent \(MFT iletişim kuralı köprüsü aracısı oluşturma ve yapılandırma\)](#)

Örnek: Bir iletişim kuralı köprüsü aracısının UNIX SFTP sunucusuyla özel anahtar kimlik bilgilerini kullanacak şekilde yapılandırılması

Bu örnek, `ProtocolBridgeCredentials.xml` dosyasını nasıl oluşturabileceğinizi ve yapılandırabileceğinizi gösterir. Bu örnek tipik bir örnektir ve ayrıntılar platformunuza göre değişebilir, ancak ilkeler aynı kalır.

Bu görev hakkında

Yordam

1. SFTP sunucusuyla kimlik doğrulaması yapmak için kullanılacak genel ve özel bir anahtar oluşturun.

Örneğin, bir Linux anasisteminde, genel/özel anahtar çiftini oluşturmak için 'openssh' paketinin bir parçası olarak sağlanan **ssh-keygen** aracısını kullanabilirsiniz.

Varsayılan olarak, bağımsız değişken olmadan, **ssh-keygen** komutu iki anahtar dosyası için varsayılan değer olan bir konum ve geçiş tümcesi ister:

```
id_rsa      <-- Private key  
id_rsa.pub  <-- Public key
```



Uyarı: Yeni bir OpenSSH sürümünden **ssh-keygen** komutunu kullanıyorsanız (RHEL 8 ile sağlanan gibi), kullanılan anahtar biçimi protokol köprüsü aracısıyla uyumlu değildir ve SFTP sunucusuna aktarma girişimleri şu iletiyle başarısız olur:

```
BFGBR0216E: Authentication to protocol server 'sftp.host.address' failed  
because of invalid private key.
```

OpenSSH' nin bu yeni sürümleriyle uyumlu bir özel anahtar oluşturmak için, **ssh-keygen** komutuna ilişkin aşağıdaki bağımsız değişkenle anahtar biçimini belirtin:

```
ssh-keygen -m PEM
```

id_rsa özel anahtarının içeriği aşağıdaki satırların ilk ve son satırlarına sahiptir:

```
-----BEGIN RSA PRIVATE KEY-----  
.....  
-----END RSA PRIVATE KEY-----
```

Bu, protokol köprüsü aracısıyla uyumludur.

2. id_rsa.pub dosyasının tüm içeriğini SFTP sunucusundaki SFTP kullanıcısının ~/.ssh/authorized_keys dosyasına kopyalayın.

Bu dosya ve ~/.ssh dizini üzerindeki dosya izinlerinin, anahtar kimlik doğrulamasına izin vermek üzere SFTP sunucusu için uygun şekilde ayarlandığından emin olun. Bu izinler genellikle şunlardır:

```
~/.ssh Mode 700  
~/.ssh/authorized_keys Mode 600
```

3. Managed File Transfer , MD5 algoritması kullanılarak bir anasistem ssh parmak izi oluşturulmasını gerektirir. SFTP sunucusunun anasistem ssh parmak izini almak için aşağıdaki komutlardan birini çalıştırın.

- Red Hat® Enterprise Linux sürüm 6.x ve altı ve Linux Ubuntu 14.04 için aşağıdaki komutu çalıştırın:

```
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
```

- Red Hat Enterprise Linux 7.x, Linux Ubuntu 16.04 ve SuSE Linux 12.4 ile başlayarak, ssh-keygen komutu varsayılan olarak SHA256 algoritmasını kullanarak ssh parmak izi oluşturur. ssh parmak izini MD5 algoritmasını kullanarak oluşturmak için aşağıdaki komutu çalıştırın:

```
ssh-keygen -l -E MD5 -f /etc/ssh/ssh_host_rsa_key.pub
```

Komutun çıkışı aşağıdaki örneğe benzer:

```
2048 MD5:64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27 no comment (RSA)
```

ProtocolBridgeCredentials.xml dosyasında hostKey olarak kullanılacak çıkışın yalnızca onaltılı kısmını çıkarın (bkz. adım “4” sayfa 316). Bu nedenle, bu örnekte 64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27 dosyasını açmanız gerekir.

4. Protokol köprüsü aracı sisteminde ProtocolBridgeCredentials.xml dosyasını düzenleyin. Aşağıdaki örnekte italik olarak gösterilen değerlerin yerine kendi değerlerinizi kullanın:

```
<tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"  
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials  
ProtocolBridgeCredentials.xsd ">  
  
<tns:agent name="Agent_name">  
  
<tns:server name="SFTP_name">  
  
<tns:user name="mq_User_ID" serverUserId="SFTP_user_ID"  
hostKey="ssh_host_finger">  
<tns:privateKey associationName="name" keyPassword="pass_phrase">  
Complete contents of the id_rsa file including the entries  
-----BEGIN RSA PRIVATE KEY-----  
  
-----END RSA PRIVATE KEY-----  
</tns:privateKey>  
</tns:user>  
</tns:server>
```

```
</tns:agent>  
</tns:credentials>
```

Burada:

- *Aracı_adi* , protokol köprüsü aracısının adıdır.
- *SFTP_anasistem_adi* , ProtocolBridgeProperties.xml dosyasında gösterildiği gibi SFTP sunucusunun adıdır.
- *mq_User_ID* , aktarma isteğiyle ilişkilendirilmiş MQMD kullanıcı kimliğidir.
- *SFTP_user_ID* , adım 2 'de kullanılan SFTP kullanıcı kimliğidir. SFTP ' ye iletilen değer, oturum açma kullanıcı kimliği olarak işlev görür.
- *ssh_host_finger* , adım 3 'te toplanan parmak izidir.
- *ad* , izleme ve günlüğe kaydetme amacıyla kullanılmasını belirleyebileceğiniz bir addır.
- *parola_ifadesi* , 1. adımda ssh-keygen içinde sağladığınız geçiş tümceciğidir.
- *id_rsa dosyasının tam içeriği* , 1. adımda oluşturulan id_rsa dosyasının tam içeriğidir. Bir bağlantı hatasını önlemek için aşağıdaki girişlerin her ikisini de eklediğinizden emin olun:

```
-----BEGIN RSA PRIVATE KEY-----  
-----END RSA PRIVATE KEY-----
```

< tns: privatekey> ögesini çoğaltarak ek anahtarlar ekleyebilirsiniz.

5. Aracı başlatılmadıysa, protokol köprüsü aracısını başlatın. Diğer bir seçenek olarak, protokol köprüsü aracısı belirli aralıklarla ProtocolBridgeCredentials.xml dosyasını yoklayıp değişiklikleri alır.

İlgili başvurular

[İletişim kuralı köprüsü kimlik bilgileri dosyası biçimi](#)

[fteCreateBridgeAgent \(MFT iletişim kuralı köprüsü aracısı oluşturma ve yapılandırma\)](#)

[MFT agent.properties dosyası](#)

[“Dosya sunucusuna ilişkin kimlik bilgilerini eşleme” sayfa 310](#)

Managed File Transfer içindeki kullanıcı kimlik bilgilerini, protokol köprüsü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanarak ya da kendi kullanıcı çıkışınızı yazarak dosya sunucusundaki kullanıcı kimlik bilgileriyle eşleyin. Managed File Transfer , kullanıcı kimlik bilgileri eşlemesi gerçekleştiren örnek bir kullanıcı çıkışı sağlar.

FTPS sunucusu için protokol köprüsünün yapılandırılması

FTP sunucusunu yapılandırırken benzer bir şekilde bir FTPS sunucusu yapılandırın: sunucu için bir köprü aracısı oluşturun, sunucu özelliklerini tanımlayın ve kullanıcı kimlik bilgilerini eşleyin.

Bu görev hakkında

Bir FTPS sunucusunu yapılandırmak için aşağıdaki adımları tamamlayın:

Yordam

1. **fteCreateBridgeAgent** komutunu kullanarak FTPS sunucusu için bir protokol köprüsü aracısı oluşturun. FTP için geçerli olan parametreler FTPS için de geçerlidir, ancak FTPS ' ye özgü üç zorunlu parametre de vardır:
 - a) **-bt** parametresi. Bu parametrenin değeri olarak FTPS belirtin.
 - b) Güvenli depo dosyası için **-bts** parametresi. Komut, yalnızca sunucu kimlik doğrulamasının gerekli olduğunu ve güvenilir depo dosyasının konumunu belirtmeniz gerektiğini varsayar.

FTPS protokolünün belirtik biçimi varsayılan olarak **fteCreateBridgeAgent** komutu tarafından yapılandırılır, ancak protokol köprüsü özellikleri dosyasını değiştirerek örtük formu yapılandırabilirsiniz. Protokol köprüsü her zaman pasif kipte FTPS sunucularına bağlanır.

fteCreateBridgeAgent komutuyla ilgili daha fazla bilgi için bkz. [fteCreateBridgeAgent \(bir MFT iletişim kuralı köprüsü aracısı oluşturma ve yapılandırma\)](#).

Güvenilir depo dosyalarının nasıl oluşturulacağına ilişkin yönergelere gereksinim duyarsanız, [Oracle keytool](#) belgelerinde anahtar aracısıyla ilgili bilgilere bakın.

2. Protokol köprüsü özellikleri dosyasındaki bir <ftpsServer> ögesi içinde FTPS sunucusu özelliklerini tanımlayın: `ProtocolBridgeProperties.xml`. Daha fazla bilgi için, bkz. [“ProtocolBridgeProperties.xml dosyasını kullanarak iletişim kuralı dosya sunucularına ilişkin özellikleri tanımlama” sayfa 304](#). Protokol köprüsü özellikleri dosyasını düzenleyerek istemci kimlik doğrulamasını da etkinleştirebilirsiniz. Tüm yapılandırma seçeneklerinin ayrıntıları için bkz. [Protokol köprüsü özellikleri dosya biçimi](#).
3. Managed File Transfer içindeki kullanıcı kimlik bilgilerini, protokol köprüsü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanarak ya da kendi kullanıcı çıkışınızı yazarak FTPS sunucusundaki kullanıcı kimlik bilgileriyle eşleyin. Daha fazla bilgi için bkz. [“Dosya sunucusuna ilişkin kimlik bilgilerini eşleme” sayfa 310](#).
4. Varsayılan olarak, güvenilir depo dosyası JKS biçiminde yapılandırılır; biçimi değiştirmek istiyorsanız, protokol köprüsü özellikler dosyasını düzenleyin.

Örnek

Protokol köprüsü özellikleri dosyasındaki bir FTPS sunucusu için örnek bir girdi aşağıdaki gibi gösterilir:

```
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">
  <tns:defaultServer name="ftpserver.mycompany.com" />

  <tns:ftpsServer name="ftpserver.mycompany.com" host="ftpserver.mycompany.com" port="990"
  platform="windows"
  timeZone="Europe/London" locale="en_US" fileEncoding="UTF8"
  listFormat="unix" limitedWrite="false"
  trustStore="c:\mydirec\truststore.jks" />

  <!-- Define servers here -->
</tns:serverProperties>
```

Sonraki adım

Desteklenen ve desteklenmeyen FTPS protokolünün parçaları hakkında bilgi için [İletişim kuralı köprüsü](#) tarafından FTPS sunucu desteği başlıklı konuya bakın.

İlgili kavramlar

[“Protokol köprüsü” sayfa 302](#)

İletişim kuralı köprüsü, Managed File Transfer (MFT) ağınızın, yerel etki alanınızda ya da uzak bir konumda MFT ağınızın dışındaki bir dosya sunucusunda saklanan dosyalara erişmesini sağlar. Bu dosya sunucusu FTP, FTPS ya da SFTP ağ iletişim kurallarını kullanabilir. Her dosya sunucusu için en az bir özel olarak ayrılmış aracı gerekir. Özel olarak ayrılmış ajan, protokol köprüsü aracısı olarak bilinir. Bir köprü aracısı birden çok dosya sunucusuyla etkileşimde bulunabilir.

İlgili görevler

[“ProtocolBridgeCredentials.xml dosyasını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerini eşleme” sayfa 311](#)

Managed File Transfer içindeki kullanıcı kimlik bilgilerini, protokol köprüsü aracısının varsayılan kimlik bilgisi eşleme işlevini kullanarak dosya sunucusundaki kullanıcı kimlik bilgileriyle eşleyin. Managed File Transfer, kimlik bilgilerinizi dahil etmek için düzenleyebileceğiniz bir XML dosyası sağlar.

[“ProtocolBridgeProperties.xml dosyasını kullanarak iletişim kuralı dosya sunucularına ilişkin özellikleri tanımlama” sayfa 304](#)

Dosyaları aracı yapılandırma dizininde Managed File Transfer tarafından sağlanan `ProtocolBridgeProperties.xml` dosyasına aktarmak ve bu dosyayı kullanarak aktarmak istediğiniz bir ya da daha fazla iletişim kuralı dosya sunucusunun özelliklerini tanımlayın.

İlgili başvurular

[fteCreateBridgeAgent \(MFT iletişim kuralı köprüsü aracısı oluşturma ve yapılandırma\)](#)

[İletişim kuralı köprüsü kimlik bilgileri dosyası biçimi](#)

[Protokol köprüsü özellikleri dosya biçimi](#)

[Protokol köprüsüne göre FTPS sunucusu desteği](#)

V9.3.0 Tek tek dosya sunucularına dosya aktarımı sayısını sınırlamaya ilişkin senaryolar ve örnekler

Düzeltilen protokol köprüsü aracısının bazı örneklerle birlikte **maxActiveDestinationTransfers** ve **failTransferWhenCapacityReached** öznitelikleriyle nasıl çalıştığı.

maxActiveDestinationTransfers değerine dayalı olarak protokol köprüsü aracısının çalışmasını gösteren senaryolar

1. senaryo

Bir protokol köprüsü aracısının ProtocolBridgeProperties.xml dosyası iki dosya sunucusu tanımlar içerir:

- Genel **maxActiveDestinationTransfers** özniteliğini ayarlamadınız.
- **maxActiveDestinationTransfers** özniteliğini hem fileServerA hem de FileServerB üzerinde ayarlamadınız.
- Protokol köprüsü aracısı **maxDestinationTransfers** özniteliğini varsayılan değere ayarladınız.

Protokol köprüsü aracısı **maxDestinationTransfers** özniteliğini varsayılan değer olan 25 değerine ayarladıysanız:

- Hedef aracı, fileServerA' e iki yönetilen aktarım işlemeye başlar.
- Her iki transfer de tamamlandı.

Bu noktada istemci, fileServerA ' in başarısız olduğunu anlar ve ProtocolBridgeProperties.xml dosyasında fileServerA için aşağıdaki değerleri ayarlar:

```
maxActiveDestinationTransfers = 0
failTransferWhenCapacityReached = true
```

- fileServerA için başka bir aktarım ve fileServerB için birkaç aktarım daha gelir:
Önceki adımda ayarlanan özelliklere dayalı olarak, fileServerA ' e yönetilen aktarım reddedilir ve başarısız olarak işaretlenir, oysa fileServerB aktarımları standart var olan akışta işlenir.
- Bir süre sonra istemci fileServerA ' in yeniden çalıştığını öğrenir, bu nedenle istemci ProtocolBridgeProperties.xml içinde önceden eklenen değeri kaldırır ya da yorumlar. fileServerA için yeni bir yönetilen aktarım gelir ve standart var olan akışta işlenir.

2. senaryo

- Bir dosya sunucusu için **maxActiveDestinationTransfers** özniteliğini ayarladınız ve **failTransferWhenCapacityReached** özniteliğini ayarladınız.
- İletişim kuralı köprüsü aracısı, dosya sunucusuna bu sayıda yönetilen aktarım için hedef aracı olarak hareket ediyor.
- **maxActiveDestinationTransfers** özniteliğinin değeri 1 azaltılır.

Protokol köprüsü aracısı, yapılandırmasını dinamik olarak günceller ve **maxActiveDestinationTransfers** ' i etkin durumdayken yeni değere ayarlar. Devam eden yönetilen aktarımlar bu güncellemeden etkilenmez ve tamamlanmasına izin verilir.

3. senaryo

Bir protokol köprüsü aracısı için ProtocolBridgeProperties.xml dosyası iki dosya sunucusu tanımlaması içerir:

- Genel **maxActiveDestinationTransfers** özneliğini ayarlamadınız.
- **failTransferWhenCapacityReached** özneliğini ayarlı değilsiniz.
- **maxActiveDestinationTransfers** değerini fileServerA üzerinde 1 olarak ayarladınız.
- fileServerB üzerinde **maxActiveDestinationTransfers** özneliğini ayarlamadınız.

Protokol köprüsü aracısının **maxDestinationTransfers** özneliği 5 olarak ayarlanmışsa:

- Protokol köprüsü aracısından fileServerA 'e etkin hedef aktarımı sayısı üst sınırı 1 'dir (hedef aracıda 5 hedef aktarım yuvası olmasına rağmen, fileServerA' e yapılan yönetilen aktarımlar için yalnızca 1 kullanılabilir).

Bu, fileServerA başarısız olduğunda kullanışlıdır. fileServerA yeniden çalıştıktan sonra, izin verilen hedef aktarımların tam kapasitesine izin vermek için **maxActiveDestinationTransfers** değeri 5 'e yükseltilebilir.

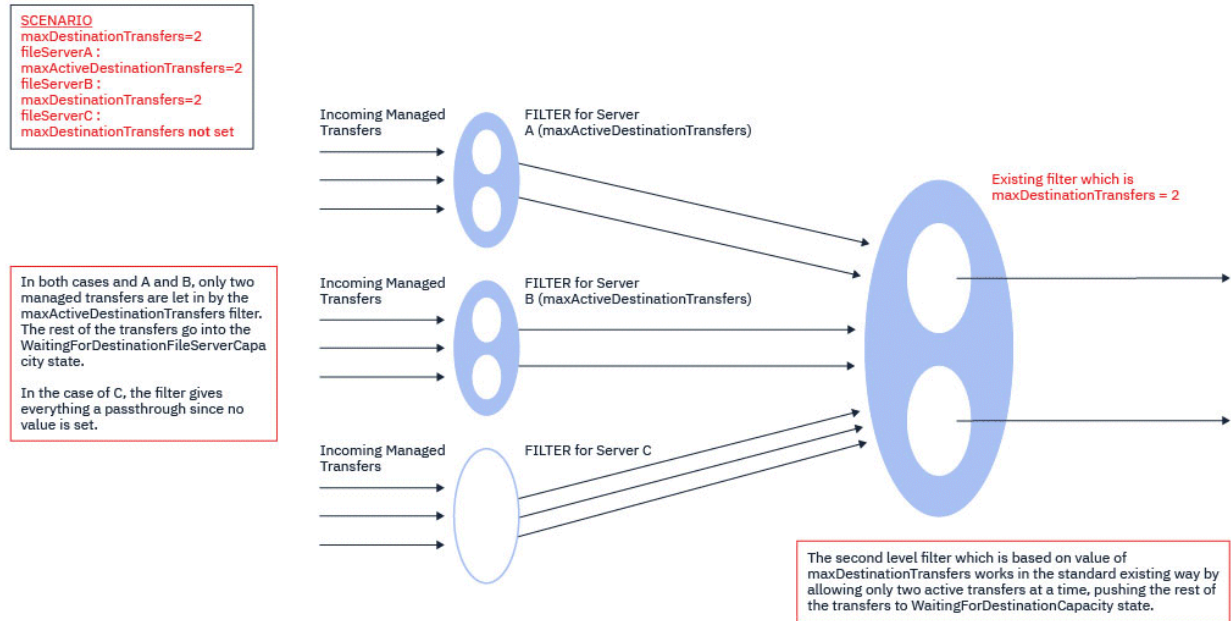
- Protokol köprüsü aracısından fileServerB ' e etkin hedef aktarımı sayısı üst sınırı 5 'tir.

Bu dosya sunucusu için **maxActiveDestinationTransfers** ayarlanmadığı için, protokol köprüsü aracısı kendisine yönetilen aktarımlar için hedef aktarma yuvalarının 5 'ini de kullanabilir.

4. senaryo

Aşağıdaki çizgede:

- **maxDestinationTransfers** özneliğini agent . properties dosyasında 2 olarak ayarladınız.
- fileServerA üzerinde **maxActiveDestinationTransfers** değerini 2 olarak ayarladınız.
- **maxActiveDestinationTransfers** özneliğini fileServerB üzerinde 2 olarak ayarladınız.
- fileServerC üzerinde **maxActiveDestinationTransfers** özneliğini ayarlamadınız.



Çizgenin gösterdiği gibi, **maxActiveDestinationTransfers** ve **maxDestinationTransfers** öznelikleri birbirinden bağımsızdır.

Her bir sunucu için **maxActiveDestinationTransfers** değerleri denetlenir. Bu değere dayalı olarak, aktarımların devam etmesine ya da **WaitingForDestinationFileServerCapacity** durumuna itilmesine izin verilir.

Daha sonra izin verilen aktarımlar, **maxDestinationTransfers** ile karşılaştırılarak var olan standart denetim akışından geçer.

5. senaryo



Uyarı: **maxDestinationTransfers** özniteliğinin değerini göz altında tutmanız gerektiğinden, **maxActiveDestinationTransfers** özniteliklerinin değerlerini ayarlarken dikkatli olmanız gerekir.

Bunu yapmazsanız, aşağıdaki metinde açıklandığı gibi bir durum oluşabilir:

- Genel **maxActiveDestinationTransfers** özniteliği için bir değer ayarlamadınız.
- `agent.properties` dosyasında **maxDestinationTransfers**= 2 değerini ayarladınız.
- `fileServerA` üzerinde **maxActiveDestinationTransfers**= 2 değerini ayarladınız.
- `fileServerB` üzerinde **maxActiveDestinationTransfers** için bir değer ayarlamadınız.

Aşağıdaki olay sırasının oluştuğunu varsayın:

- Protokol köprüsü aracısı, bir dosyayı `fileServerA` ' e aktarma isteği alır. Protokol köprüsü aracısı şu anda hiçbir şey yapmıyor, bu nedenle bu yönetilen aktarım isteğini kabul ediyor.

Aktarım yuvaları şimdi şöyle görünüyor:

- Hedef Aktarımlar: 1
- `fileServerA` için Hedef Aktarımlar: 1
- `fileServerB` için hedef aktarımlar: 0
- Şimdi, protokol köprüsü aracısı `fileServerA` içeren yönetilen bir aktarım için hedef aracı olarak hareket etmek için başka bir istek alır. Bir kez daha bu isteği kabul eder ve aktarma yuvaları şöyle görünür:
 - Hedef Aktarımlar: 2
 - `fileServerA` için Hedef Aktarımlar: 2
 - `fileServerB` için hedef aktarımlar: 0

Aracıdaki iki `Destination Transfer` yuvası artık dolu olduğundan, aracı `fileServerA` ' e yapılan aktarımlardan biri tamamlanıncaya kadar yönetilen aktarımlara katılamaz.

- Kısa bir süre sonra `fileServerA` başarısız olur ve bu, yönetilen iki aktarımının kurtarma işlemine geçmesine neden olur. Bu yönetilen aktarımların kullandığı `Destination transfer` yuvaları bu süre boyunca kullanılmaya devam eder.
- Daha sonra, protokol köprüsü aracısı bir dosyayı `fileServerB` ' e aktarma isteği alır. `Destination Transfers for fileServerB` yuvalarında bu aktarma için bir alan vardır; ancak, aracıya ilişkin tüm `Destination Transfer` yuvaları kullanılır ve aktarım daha sonra yeniden denenmesi için birikim üzerine konabilir.

Sonuç olarak, `fileServerA` ' e yapılan aktarımlardan en az biri tamamlanıncaya ve `Destination Transfer` yuvasını serbest bırakıncaya kadar `fileServerB` ' e aktarma engellenir.

Bu durumun olmasını önlemek için:

- Boş yuvaların kalması için dosya sunucularındaki **maxActiveDestinationTransfers** değerini **maxDestinationTransfers** değerinden küçük olacak şekilde ayarlayın.
- Ya da **maxActiveDestinationTransfers** özniteliğinin değerini tüm uç nokta sunucularına eşit olarak dağıtın.

maxActiveDestinationTransfers özniteliğinin değerlerine dayalı olarak protokol köprüsü aracısının davranışı

Not: Aşağıdaki tabloda listelenen tüm hata durumlarında, **maxActiveDestinationTransfers** özniteliği geçerli olmayan bir değere ayarlanırsa, protokol köprüsü aracısı bu özniteliğin ayarlanmadığını varsayar.

maxActiveDestinationTransfers	Örnek değer	Açıklama
Belirlenmedi	Belirlenmedi	Transferler her zamanki gibi gidiyor. * ftp * uç noktası için aktarma sayısına sınır konmaz.
Belirtilmiş	0	Bu özel * ftp * uç noktasına aktarıma izin verilmez.
Negatif değer	-1	output0.log Value -1 günlüğüne kaydedilen hata, negatif olmayan bir tamsayı için geçerli değil. Protokol köprüsü aracısı özniteliğin ayarlanmadığını varsayar.
Tamsayı olmayan değer	abc	output0.log Value abc günlüğüne kaydedilen hata, bir tamsayı için geçerli değil. Protokol köprüsü aracısı özniteliğin ayarlanmadığını varsayar.
Boş	""	maxActiveDestinationTransfers özniteliğinin ' ' ' değeri, negatif olmayan bir tamsayı için geçerli değil.
Belirtilmiş	5	Bu * ftp * uç noktası için herhangi bir zaman noktasında yalnızca beş etkin aktarımın çalıştırılmasına izin verir. Aşırı aktarımlar, failTransferWhenCapacityErişim özniteliği değerine göre yeniden denir ya da reddedilir.

maxActiveDestinationTransfers ve failTransferWhenCapacityReached özniteliklerinin birleşimine ilişkin protokol köprüsü aracısının davranışı

failTransferWhenCapacityUlaştı değeri	maxActiveDestinationTransfers değeri	Sonuç
Yanlış	3	Bu uç nokta sunucusuna üç etkin aktarıma izin verilir. Daha fazla transfer tekrar denir.
Doğru	3	Bu uç nokta sunucusuna üç etkin aktarıma izin verilir. Diğer aktarımlar reddedilir ve başarısız olarak işaretlenir.
Belirlenmedi	3	failTransferWhenCapacity için varsayılan değer false olarak değerlendirilir.

failTransferWhenCapacityUlařtı deęeri	maxActiveDestinationTransfers deęeri	Sonu
		Sonu olarak, bu u nokta sunucusuna u etkin aktarıma izin verilir. Daha fazla transfer tekrar denendir.
Boole deęeri dıřındaki deęerler	Belirtildi	output.logdosyasına hata kaydedildi. failTransferWhenCapacityiin belirtilen deęer bir Boole deęeri deęil. failTransferWhenCapacity(Hata Durumunda Kapasiteye Ulařıldı) iin varsayılan deęer dikkate alınır.

maxDestinationTransfers ve failTransferWhenCapacityReached zneliklerinin birleřimine iliřkin protokol kprs aracasının davranıřı

failTransferWhenCapacityUlařtı deęeri	maxDestinationAktarmalar deęeri	Sonu
Doęru	10	Eřzamanlı etkin aktarma sayısı 10 'a ulařtıęında, protokol kprs aracası tarafından ynetilen 11 th aktarımı bařarısız olur.
Yanlıř	10	Var olan davranıř. Eřzamanlı etkin aktarma sayısı 10 'a ulařtıęında, 11 th ynetilen aktarım, bir yuvanın serbest bırakılmasını beklerken kuyruęa alınır.
Belirlenmedi	10	Var olan davranıř

hata iletileri

Var olan ileti:

BFGS0082I

Protokol kprs aracası aktarımı reddettięinde, protokol kprs aracası **maxDestinationTransfers** znelięinde tanımlanan aktarım sayısı st sınırını zaten alıřtırdıęında, kaynak aracının output0.log dosyasına kaydedilir.

Yeni iletiler:

BFGSS0085I

Protokol kprs aracası ynetilen bir aktarımı reddettięinde ve yeniden denedięinde, kaynak aracının output0.log dosyasına kaydedilir.

BFGSS0086I

Protokol kprs aracası ynetilen bir aktarımı reddedip yeniden denedięinde ve hedef ęe dosya sunucusu adını iermedięinde kaynak aracının output0.log dosyasına kaydedilir

BFGSS0084E

Protokol köprüsü aracı reddettiğinde, **maxActiveDestinationTransfers** özneliğinde belirtilen koşut zamanlı aktarma sayısı üst sınırını aşmak için Explorer ve audit.xml dosyasına kaydedilir ve yönetilen bir aktarımı başarısız olarak işaretler.

BFGSS0087E

Protokol köprüsü aracı reddettiğinde, **maxActiveDestinationTransfers** özneliğinde belirtilen hedef aktarım sayısı üst sınırını aşmak için Explorer ve audit.xml dosyasına kaydedilir ve yönetilen bir aktarımı başarısız olarak işaretler.

BFGSS0088W

maxActiveDestinationTransfers özneliğinin değeri **maxDestinationTransfers** özneliğinin değerini aştığında output0.loggünlüğüne kaydedilir.

BFGSS0089I

Hedef protokol köprüsü aracısının output0.log dosyasına, IBM MQ 9.3.0ya da daha sonraki bir sürümde olmayan bir kaynak araçıda çalışırken kaydedilir.

İlgili kavramlar

“Protokol köprüsü” sayfa 302

İletişim kuralı köprüsü, Managed File Transfer (MFT) ağınızın, yerel etki alanınızda ya da uzak bir konumda MFT ağınızın dışındaki bir dosya sunucusunda saklanan dosyalara erişmesini sağlar. Bu dosya sunucusu FTP, FTPS ya da SFTP ağ iletişim kurallarını kullanabilir. Her dosya sunucusu için en az bir özel olarak ayrılmış aracı gerekir. Özel olarak ayrılmış ajan, protokol köprüsü aracı olarak bilinir. Bir köprü aracı birden çok dosya sunucusuyla etkileşimde bulunabilir.

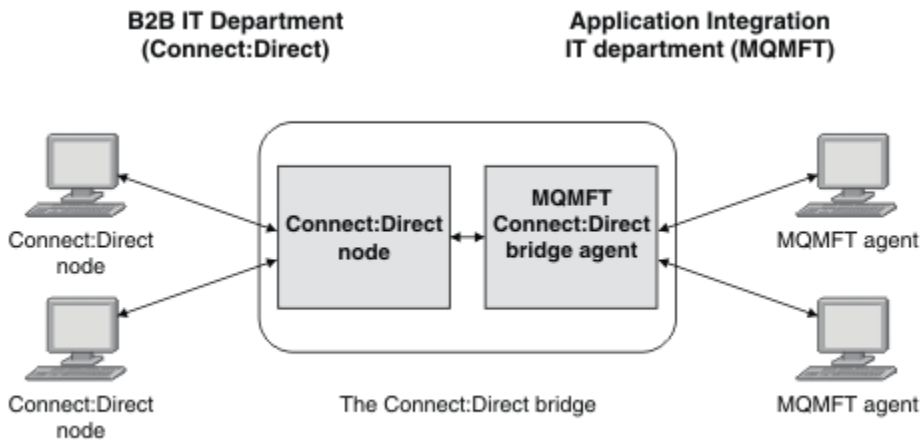
İlgili görevler

“ProtocolBridgeProperties.xml dosyasını kullanarak iletişim kuralı dosya sunucularına ilişkin özellikleri tanımlama” sayfa 304

Dosyaları aracı yapılandırma dizininde Managed File Transfer tarafından sağlanan ProtocolBridgeProperties.xml dosyasına aktarmak ve bu dosyayı kullanarak aktarmak istediğiniz bir ya da daha fazla iletişim kuralı dosya sunucusunun özelliklerini tanımlayın.

Connect:Direct köprüsü

Var olan bir IBM Sterling Connect:Direct ağına/ağından dosya aktarabilirsiniz. Dosyaları MFT ile IBM Sterling Connect:Direct arasında aktarmak için Managed File Transfer bileşeni olan Connect:Direct köprüsünü kullanın.



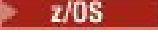

Bu şekilde, iki bölüm arasındaki bir MFT Connect:Direct köprüsü gösterilir: B2B BT bölümü ve Uygulama Bütünleştirme BT bölümü. B2B BT departmanı, Connect:Direct 'i şirketin çözüm ortaklarına/çözüm ortaklarından dosya aktarmak için kullanır. Uygulama Bütünleştirme BT departmanı, ileti sistemi altyapısı olarak IBM MQ 'yi kullanır ve bu nedenle kısa bir süre önce dosya aktarım çözümü olarak Managed File Transfer 'i seçmiştir.

İki bölüm, MFT Connect:Direct köprüsünü kullanarak B2B BT bölümündeki Connect:Direct ağı ile Application Integration IT bölümündeki MFT ağı arasında dosya aktarabilir. Connect:Direct köprüsü, Connect:Direct düğümüyle iletişim kuran bir MFT aracısını içeren bir Managed File Transfer bileşenidir. MFT aracı, Connect:Direct düğümüyle yapılan aktarımlara adanmıştır ve Connect:Direct köprü aracı olarak bilinir.

Connect:Direct köprüsü, Managed File Transfer' un Service and Agent bileşenlerinin bir parçası olarak kullanılabilir ve aşağıdaki görevler için kullanılabilir:


1. Bir dosyanın ya da birden çok dosyanın MFT aracısından Connect:Direct düğümüne aktarılmasını başlatmak için Managed File Transfer komutlarını kullanın.
2. Bir dosyanın ya da birden çok dosyanın Connect:Direct düğümünden MFT aracısına aktarılmasını başlatmak için Managed File Transfer komutlarını kullanın.
3. Kullanıcı tanımlı bir Connect:Direct işlemi başlatan bir dosya aktarımı başlatmak için Managed File Transfer komutlarını kullanın.
4. Bir MFT dosya aktarma isteği göndermek için Connect:Direct işlemi kullanın.

Connect:Direct köprüsü dosyaları yalnızca Connect:Direct düğümlerine ya da düğümlerinden aktarabilir. Connect:Direct köprüsü, yalnızca bir Connect:Direct işlemi tarafından gönderilen bir aktarımın parçası olarak yerel dosya sistemine/sisteminden dosya aktarabilir.

 Connect:Direct köprüsünü, z/OS sistemindeki bir Connect:Direct düğümünde bulunan bir veri kümesine ya da veri kümesinden aktarmak için kullanabilirsiniz. Yalnızca Managed File Transfer araçlarını içeren veri kümesi aktarımlarıyla karşılaştırıldığında bazı davranış farklılıkları vardır. Daha fazla bilgi için bkz.  Veri kümelerinin Connect:Direct düğümlere/düğümlerden aktarılması.

Desteklenen platformlar

Connect:Direct köprüsü, bir MFT Connect:Direct köprü aracı ve bir Connect:Direct düğümünden yapılır. Aracı, x86-64 için Windows ve Linux sistemlerinde desteklenir. Düğüm, Windows için IBM Sterling Connect:Direct ve UNIX için IBM Sterling Connect:Direct için desteklenen platformlarda desteklenir. Aracının iletişim kurması için Connect:Direct köprü aracı oluşturma ve Connect:Direct düğümünü yapılandırmaya ilişkin yönergeler için [Connect:Direct köprüsünü yapılandırmabaşlıklı konuya](#) bakın.

Connect:Direct köprüsü, Windows için Connect:Direct ya da UNIX , ya da Connect:Direct for z/OS Hizmet kuruluşu için Connect:Direct içinde çalışan Connect:Direct düğümlerine ya da düğümlerinden dosya aktarabilir. Desteklenen Connect:Direct sürümlerinin ayrıntıları için [IBM MQ için Sistem Gereksinimleri](#) web sayfasına bakın.

Connect:Direct köprüsünü oluşturan aracı ve düğüm aynı sistemde olmalı ya da aynı dosya sistemine (örneğin, paylaşılan bir NFS bağlantısı aracılığıyla) erişmelidir. Bu dosya sistemi, Connect:Direct köprüsünü içeren dosya aktarımları sırasında dosyaları, **cdTmpDir** parametresi tarafından tanımlanan bir dizinde geçici olarak saklamak için kullanılır. Connect:Direct köprü aracı ve Connect:Direct köprü düğümü, aynı yol adını kullanarak bu dizini adresleyebilmelidir. Örneğin, aracı ve düğüm ayrı Windows sistemlerideyse, sistemlerin paylaşılan dosya sistemini bağlamak için aynı sürücü harfini kullanması gerekir. Aşağıdaki yapılandırmalar, aracının ve düğümün aynı yol adını kullanmasına izin verir:

- Aracı ve düğüm aynı sistemde; Windows ya da Linux for x86-64 çalıştırılıyor
- Aracı, x86-64 için Linux üzerinde ve düğüm AIX üzerdedir
- Aracı bir Windows sisteminde ve düğüm başka bir Windows sisteminde

Aşağıdaki yapılandırmalar, aracının ve düğümün aynı yol adını kullanmasına izin vermiyor:

- Aracı, x86-64 için Linux üzerinde ve düğüm Windows üzerdedir
- Aracı Windows üzerdedir ve düğüm UNIX üzerdedir

Connect:Direct köprüsünü kurmayı planlarken bu kısıtlamayı göz önünde bulundurun.

İlgili kavramlar

[“Connect:Direct düğümlerine/düğümlerinden aktarma için kurtarma ve yeniden başlatma” sayfa 333](#)

Managed File Transfer , bir aktarma sırasında IBM Sterling Connect:Direct düğümünüze bağlanamayabilir; örneğin, düğüm kullanılamaz duruma gelirse. Managed File Transfer , aktarımı kurtarmaya çalışır ya da aktarma başarısız olur ve bir hata iletisi üretilir.

[“Dosya aktarma isteğinden kullanıcı tanımlı bir Connect:Direct işleminin sunulması” sayfa 334](#)

Dosya aktarımının bir parçası olarak kullanıcı tanımlı bir Connect:Direct işlemini çağırarak Connect:Direct köprü aracısından geçen bir aktarma isteği gönderebilirsiniz.

[“Aktarma isteklerini Connect:Direct sunmak için Managed File Transfer süreçlerin kullanılması” sayfa 338](#)

Bir Connect:Direct işleminden Connect:Direct köprü aracısına bir aktarma isteği gönderebilirsiniz.

Managed File Transfer , Connect:Direct sürecindeki bir **RUN TASK** deyiminden çağrılabilen komutlar sağlar.

İlgili görevler

[Connect:Direct köprüsünün yapılandırılması](#)

[“Connect:Direct düğümüne dosya aktarılması” sayfa 326](#)

Connect:Direct köprüsünü kullanarak bir dosyayı Managed File Transfer aracısından Connect:Direct düğümüne aktarabilirsiniz. Hedef aracı olarak Connect:Direct köprü aracısını belirterek ve `connect_direct_node_name:file_path` formunda hedef dosyayı belirterek, aktarma hedefi olarak bir Connect:Direct düğümü belirtin.

[“Connect:Direct düğümünden dosya aktarılması” sayfa 327](#)

Connect:Direct köprüsünü kullanarak bir dosyayı Connect:Direct düğümünden Managed File Transfer Agent düğümüne aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprü aracısını belirterek ve `connect_direct_node_name:file_path` formunda kaynak belirtimini belirterek, aktarma kaynağı olarak bir Connect:Direct düğümü belirtebilirsiniz.

[“Birden çok dosyanın Connect:Direct düğümüne aktarılması” sayfa 329](#)

Connect:Direct köprüsünü kullanarak bir Managed File Transfer Agent düğümünden Connect:Direct düğümüne birden çok dosya aktarabilirsiniz. Birden çok dosya aktarımının hedefi olarak bir Connect:Direct düğümünü kullanmak için, hedef aracı olarak Connect:Direct köprü aracısını belirtin ve `connect_direct_node_name:directory_path` biçiminde hedef dizini belirtin.

[“Connect:Direct düğümünden birden çok dosyanın aktarılması” sayfa 330](#)

Connect:Direct köprüsünü kullanarak Connect:Direct düğümünden Managed File Transfer Agent düğümüne birden çok dosya aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprü aracısını belirterek ve formda bir ya da daha fazla kaynak belirtimini belirterek çoklu dosya aktarımının kaynağı olarak bir Connect:Direct düğümünü belirtebilirsiniz `connect_direct_node_name:file_path`.

[“Joker karakter kullanarak birden çok dosyayı Connect:Direct ' e aktarma” sayfa 331](#)

Bir Managed File Transfer aracısından Connect:Direct düğümüne birden çok dosya aktarmak için Connect:Direct köprüsünü kullanın. **fteCreateTransfer** komutuna sağladığınız kaynak belirtiminde genel arama karakterleri kullanabilirsiniz. Genel arama karakterlerini içeren tüm Managed File Transfer aktarımlarında olduğu gibi, dosya yolunun yalnızca son kısmı genel arama karakteri içerebilir. Örneğin, `/abc/def*` geçerli bir dosya yoludur ve `/abc*/def` geçerli değildir.

[Connect:Direct köprüsünde sorun giderme](#)

İlgili başvurular

[fteCreateCDAgent: Connect:Direct köprü aracısı oluşturma](#)

[Connect:Direct köprü aracısının kısıtlamaları](#)

Connect:Direct düğümüne dosya aktarılması

Connect:Direct köprüsünü kullanarak bir dosyayı Managed File Transfer aracısından Connect:Direct düğümüne aktarabilirsiniz. Hedef aracı olarak Connect:Direct köprü aracısını belirterek ve `connect_direct_node_name:file_path` formunda hedef dosyayı belirterek, aktarma hedefi olarak bir Connect:Direct düğümü belirtin.

Başlamadan önce

Bir dosyayı aktarmadan önce, Managed File Transfer bileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Daha fazla bilgi için [Connect:Direct köprüsünü yapılandırma](#) başlıklı konuya bakın.

Bu görev hakkında

Bu örnekte, Connect:Direct köprü aracı CD_BRIDGE olarak adlandırılır. Kaynak aracıya FTE_AGENT adı verilir ve WMQFTE 'nin herhangi bir sürümü olabilir. Hedef Connect:Direct düğümüne CD_NODE1 adı verilir. Aktarılacak dosya, FTE_AGENT 'in bulunduğu sistemde /home/helen/file.log dosya yolunda bulunur. Dosya, CD_NODE1 'in çalıştığı sistemdeki /files/data.log dosya yoluna aktarılır.

Yordam

1. fteCreateTransfer komutunu, *connect_direct_node_name:file_path* formundaki **-df** (hedef dosya) parametresi değeri ve Connect:Direct köprü aracısının adı olarak belirtilen **-da** (hedef aracı) parametresinin değeri ile birlikte kullanın.

Not: *connect_direct_node_name* ile belirtilen Connect:Direct düğümü, Connect:Direct köprüsünün bir parçası olarak çalışan Connect:Direct düğümüne değil, dosyanın aktarılmasını istediğiniz düğümdür.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE  
-df CD_NODE1:/files/data.log /home/helen/file.log
```

Daha fazla bilgi için bkz. **fteCreateTransfer**: yeni bir dosya aktarımı başlatma.

2. Kaynak aracı FTE_AGENT, dosyayı Connect:Direct köprü aracı CD_BRIDGE 'ye aktarır. Dosya, Connect:Direct köprü aracısının çalıştığı sistemde, cdTmpDir aracı özelliği tarafından tanımlanan konumda geçici olarak saklanır. Connect:Direct köprü aracı dosyayı Connect:Direct düğümüne CD_NODE1 aktarır.

İlgili kavramlar

[“Connect:Direct köprüsü” sayfa 324](#)

Var olan bir IBM Sterling Connect:Direct ağına/ağından dosya aktarabilirsiniz. Dosyaları MFT ile IBM Sterling Connect:Direct arasında aktarmak için Managed File Transfer bileşeni olan Connect:Direct köprüsünü kullanın.

İlgili görevler

[“Connect:Direct düğümünden dosya aktarılması” sayfa 327](#)

Connect:Direct köprüsünü kullanarak bir dosyayı Connect:Direct düğümünden Managed File Transfer Agent düğümüne aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprü aracısını belirterek ve *connect_direct_node_name:file_path* formunda kaynak belirtimini belirterek, aktarma kaynağı olarak bir Connect:Direct düğümü belirtebilirsiniz.

İlgili başvurular

[MFT agent.properties dosyası](#)

Connect:Direct düğümünden dosya aktarılması

Connect:Direct köprüsünü kullanarak bir dosyayı Connect:Direct düğümünden Managed File Transfer Agent düğümüne aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprü aracısını belirterek ve *connect_direct_node_name:file_path* formunda kaynak belirtimini belirterek, aktarma kaynağı olarak bir Connect:Direct düğümü belirtebilirsiniz.

Başlamadan önce

Bir dosyayı aktarmadan önce, Managed File Transfer bileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Bkz. [Connect:Direct köprüsünün yapılandırılması](#).

Bu görev hakkında

Bu örnekte, Connect:Direct köprü aracı CD_BRIDGE olarak adlandırılır. Hedef aracıya FTE_AGENT adı verilir ve herhangi bir Managed File Transfer sürümü olabilir. Kaynak Connect:Direct düğümüne CD_NODE1 adı verilir. Aktarılacak dosya, CD_NODE1 'in bulunduğu sistemdeki /home/brian/in.file dosya yolunda bulunur. Dosya, FTE_AGENT 'in çalıştığı sistemdeki /files/out.file dosya yoluna aktarılır.

Yordam

fteCreateTransfer komutunu, *connect_direct_node_name:file_path* formundaki kaynak belirtimine ilişkin deęerle ve Connect:Direct köprü aracısının adı olarak belirtilen **-sa** parametresinin deęeriyle birlikte kullanın.

Not: *connect_direct_node_name* ile belirtilen Connect:Direct düęümü, Connect:Direct köprüsünün bir parçası olarak çalışan Connect:Direct düęümü deęil, dosyanın aktarılmasını istedięiniz düęümdür. Örneęin:

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_AGENT  
-df /files/out.file CD_NODE1:/home/brian/in.file
```

Daha fazla bilgi için bkz. **fteCreateTransfer: yeni bir dosya aktarımı başlatma.**

Sonuçlar

Connect:Direct köprü aracı CD_BRIDGE, dosyayı Connect:Direct düęümünden CD_NODE1ister. Connect:Direct düęümü dosyayı Connect:Direct köprüsüne gönderir. Dosya Connect:Direct düęümünden aktarılırken, Connect:Direct köprüsü dosyayı geçici olarak cdTmpDir aracı özellięi tarafından tanımlanan konumda saklar. Dosya, Connect:Direct düęümünden Connect:Direct köprüsüne aktarma işlemini tamamladıęında, Connect:Direct köprüsü dosyayı hedef aracı FTE_AGENT ' e gönderir ve geçici konumdan dosyayı siler.

İlgili kavramlar

“Connect:Direct köprüsü” sayfa 324

Var olan bir IBM Sterling Connect:Direct aęına/aęından dosya aktarabilirsiniz. Dosyaları MFT ile IBM Sterling Connect:Directarasında aktarmak için Managed File Transferbileşeni olan Connect:Direct köprüsünü kullanın.

İlgili başvurular

MFT agent.properties dosyası

Veri kümesinin z/OS üzerinde Connect:Direct düęümüne aktarılması

Bir veri kümesini, Windows ya da Linux sisteminde bulunan bir Connect:Direct köprüsünü kullanarak z/OS üzerindeki bir Managed File Transfer aracısından z/OS üzerindeki bir Connect:Direct düęümüne aktarabilirsiniz.

Başlamadan önce

Bir dosyayı aktarmadan önce, Managed File Transferbileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Bkz. Connect:Direct köprüsünün yapılandırılması.

Bu görev hakkında

Bu örnekte, aktarma hedefini belirtmek için **-df** deęiştirgesi kullanılır. **-df** parametresi, aktarımın kaynak aracı herhangi bir Managed File Transfersürümü olduęunda kullanılabilir. Bunun yerine **-ds** parametresini kullanabilirsiniz. Kaynak aracı FTE_ZOS1 olarak adlandırılır ve bir Managed File Transfer aracıdır. Connect:Direct köprü aracı CD_BRIDGE olarak adlandırılır ve bir Linux sisteminde bulunur. Hedef Connect:Direct düęümüne CD_ZOS2adı verilir. Hem kaynak aracı hem de hedef Connect:Direct düęümü, z/OS sistemlerinde bulunur. Aktarılacak veri kümesi, FTE_ZOS1 ' in bulunduęu sistemde // FTEUSER . SOURCE . LIB dizininde bulunur. Veri kümesi, CD_ZOS2 ' nin bulunduęu sistemdeki // CDUSER . DEST . LIB veri kümesine aktarılır.

Yordam

1. fteCreateTransfer komutunu, řu formdaki **-df** parametresi deęeriyle birlikte kullanın:
connect_direct_node_name:data_set_name;attributes ve Connect:Direct köprü aracısının adı olarak belirtilen **-da** (hedef aracı) parametresinin deęerini.

`connect_direct_node_name` ile belirtilen Connect:Direct düğümü, Connect:Direct köprüsünün bir parçası olarak çalışan Connect:Direct düğümüne değil, veri kümesinin aktarılmasını istediğiniz düğümdür.

`data_set_name` tarafından belirtilen veri kümesi adı, görelî değil, mutlak olmalıdır. Connect:Direct , veri kümesi adına kullanıcının adıyla önek eklemeyiz.

```
fteCreateTransfer -sa FTE_ZOS1 -sm QM_ZOS
                  -da CD_BRIDGE -dm QM_BRIDGE
                  -df CD_ZOS2:/'CDUSER.DEST.LIB;BLKSIZE(8000);LRECL(80)'
                  //'FTEUSER.SOURCE.LIB'
```

Daha fazla bilgi için bkz. **fteCreateTransfer**: yeni bir dosya aktarımı başlatma.

2. Kaynak aracı FTE_ZOS1 , veri kümesindeki verileri Connect:Direct köprü aracısı CD_BRIDGE ' ye aktarır. Veriler, cdTmpDir aracısı özelliği tarafından tanımlanan konumda, Connect:Direct köprü aracısının çalıştığı sistemde geçici olarak düz bir dosya olarak depolanır. Connect:Direct Bridge Agent, verileri Connect:Direct düğümüne CD_ZOS2aktarır. Aktarım tamamlandığında, düz dosya Connect:Direct köprü aracısının çalıştığı sistemden silinir.

İlgili kavramlar


“Connect:Direct köprüsü” sayfa 324

Var olan bir IBM Sterling Connect:Direct ağına/ağından dosya aktarabilirsiniz. Dosyaları MFT ile IBM Sterling Connect:Directarasında aktarmak için Managed File Transferbileşeni olan Connect:Direct köprüsünü kullanın.

İlgili görevler

 [Veri kümelerinin Connect:Direct düğümlerine/düğümlerinden aktarılması](#)

İlgili başvurular

 [MFT ile kullanılmamanız gereken BPXWDYN özellikleri](#)

Birden çok dosyanın Connect:Direct düğümüne aktarılması

Connect:Direct köprüsünü kullanarak bir Managed File Transfer Agent düğümünden Connect:Direct düğümüne birden çok dosya aktarabilirsiniz. Birden çok dosya aktarımının hedefi olarak bir Connect:Direct düğümünü kullanmak için, hedef aracı olarak Connect:Direct köprü aracısını belirtin ve `connect_direct_node_name:directory_path`biçiminde hedef dizini belirtin.

Başlamadan önce

Dosyaları aktarmadan önce, Managed File Transferbileşeninin bir bileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Bkz. [Connect:Direct köprüsünün yapılandırılması](#).

Bu görev hakkında

Bu örnekte, kaynak aracıya FTE_AGENT adı verilir. Connect:Direct köprü aracısı CD_BRIDGE olarak adlandırılır. Hedef Connect:Direct düğümüne CD_NODE1adı verilir. Aktarılacak dosyalar, FTE_AGENT ' in bulunduğu sistemde /home/jack/data.log, /logs/log1.txtve /results/latest . Dosyalar, CD_NODE1 ' in çalıştığı sistemdeki /in/files dizinine aktarılır.

Yordam

fteCreateTransfer komutunu, `connect_direct_node_name:directory_path`formundaki **-dd** (hedef dizin) parametresi değeriyle birlikte kullanın. **-da** (hedef aracı) parametresinin değerini Connect:Direct köprü aracısının adı olarak belirtin.

Not: `connect_direct_node_name` ile belirtilen Connect:Direct düğümü, Connect:Direct köprüsünün bir parçası olarak çalışan Connect:Direct düğümüne değil, dosyaların aktarılmasını istediğiniz düğümdür.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/in/files /home/jack/data.log
                  /logs/log1.txt /results/latest
```

Daha fazla bilgi için bkz. [fteCreateTransfer](#): yeni bir dosya aktarımı başlatma.

Sonuçlar

Kaynak aracı FTE_AGENT, ilk dosyayı Connect:Direct köprü aracı CD_BRIDGE ' ye aktarır. Connect:Direct köprü aracı, dosyayı geçici olarak cdTmpDir özelliği tarafından tanımlanan konumda saklar. Dosya, kaynak aracıdan Connect:Direct köprüsüne tamamen aktarıldığında, Connect:Direct köprü aracı dosyayı cdNode aracı özelliği tarafından tanımlanan Connect:Direct düğümüne gönderir. Bu düğüm, dosyayı hedef Connect:Direct düğümü CD_NODE1 ' e gönderir. Connect:Direct Köprü aracı, iki Connect:Direct düğümü arasındaki aktarım tamamlandığında dosyayı geçici konumdan siler. Bu işlem, belirtilen her kaynak dosya için yinelenir.

İlgili kavramlar

[“Connect:Direct köprüsü” sayfa 324](#)

Var olan bir IBM Sterling Connect:Direct ağına/ağından dosya aktarabilirsiniz. Dosyaları MFT ile IBM Sterling Connect:Direct arasında aktarmak için Managed File Transfer bileşeni olan Connect:Direct köprüsünü kullanın.

İlgili görevler

[“Connect:Direct düğümüne dosya aktarılması” sayfa 326](#)

Connect:Direct köprüsünü kullanarak bir dosyayı Managed File Transfer aracısından Connect:Direct düğümüne aktarabilirsiniz. Hedef aracı olarak Connect:Direct köprü aracısını belirterek ve `connect_direct_node_name:file_path` formunda hedef dosyayı belirterek, aktarma hedefi olarak bir Connect:Direct düğümü belirtin.

[“Joker karakter kullanarak birden çok dosyayı Connect:Direct ' e aktarma” sayfa 331](#)

Bir Managed File Transfer aracısından Connect:Direct düğümüne birden çok dosya aktarmak için Connect:Direct köprüsünü kullanın. **fteCreateTransfer** komutuna sağladığınız kaynak belirtiminde genel arama karakterleri kullanabilirsiniz. Genel arama karakterlerini içeren tüm Managed File Transfer aktarımlarında olduğu gibi, dosya yolunun yalnızca son kısmı genel arama karakteri içerebilir. Örneğin, `/abc/def*` geçerli bir dosya yoludur ve `/abc*/def` geçerli değildir.

[“Connect:Direct düğümünden dosya aktarılması” sayfa 327](#)

Connect:Direct köprüsünü kullanarak bir dosyayı Connect:Direct düğümünden Managed File Transfer Agent düğümüne aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprü aracısını belirterek ve `connect_direct_node_name:file_path` formunda kaynak belirtimini belirterek, aktarma kaynağı olarak bir Connect:Direct düğümü belirtebilirsiniz.

[“Connect:Direct düğümünden birden çok dosyanın aktarılması” sayfa 330](#)

Connect:Direct köprüsünü kullanarak Connect:Direct düğümünden Managed File Transfer Agent düğümüne birden çok dosya aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprü aracısını belirterek ve formda bir ya da daha fazla kaynak belirtimini belirterek çoklu dosya aktarımının kaynağı olarak bir Connect:Direct düğümünü belirtebilirsiniz `connect_direct_node_name:file_path`.

İlgili başvurular

[MFT agent.properties dosyası](#)

Connect:Direct düğümünden birden çok dosyanın aktarılması

Connect:Direct köprüsünü kullanarak Connect:Direct düğümünden Managed File Transfer Agent düğümüne birden çok dosya aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprü aracısını belirterek ve formda bir ya da daha fazla kaynak belirtimini belirterek çoklu dosya aktarımının kaynağı olarak bir Connect:Direct düğümünü belirtebilirsiniz `connect_direct_node_name:file_path`.

Başlamadan önce

Bir dosyayı aktarmadan önce, Managed File Transfer bileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Bkz. [Connect:Direct köprüsünün yapılandırılması](#).

Bu görev hakkında

Bu örnekte, Connect:Direct köprü aracı CD_BRIDGE olarak adlandırılır. Hedef aracıya FTE_Z adı verilir ve z/OS sisteminde çalışır. Kaynak Connect:Direct düğümüne CD_NODE1 adı verilir. Aktarılabilecek kütükler,

CD_NODE1 ' in bulunduğu sistemde /in/file1, /in/file2ve /in/file3 kütük yollarında bulunur. Dosyalar, FTE_Z ' nin çalıştığı sistemdeki //OBJECT . LIB bölümlenmiş veri kümesine aktarılır.

Yordam

fteCreateTransfer komutunu, *connect_direct_node_name:file_path* formundaki kaynak belirtilmesine ilişkin değerlerle ve Connect:Direct köprü aracısının adı olarak belirtilen **-sa** parametresinin değeriyle birlikte kullanın.

Not: *connect_direct_node_name* ile belirtilen Connect:Direct düğümü, Connect:Direct köprüsünün bir parçası olarak çalışan Connect:Direct düğümü değil, dosyaların aktarılmasını istediğiniz düğümdür.

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_Z
                  -dp //OBJECT.LIB' CD_NODE1:/in/file1
                  CD_NODE1:/in/file2 CD_NODE1:/in/file3
```

Daha fazla bilgi için bkz. **fteCreateTransfer**: yeni bir dosya aktarımı başlatma.

Sonuçlar

Connect:Direct köprü aracı CD_BRIDGE, ilk dosyayı Connect:Direct düğümünden CD_NODE1 ister. Connect:Direct düğümü dosyayı Connect:Direct köprüsüne gönderir. Dosya Connect:Direct düğümünden aktarılırken, Connect:Direct köprüsü dosyayı geçici olarak cdTmpDir aracı özelliği tarafından tanımlanan konumda saklar. Dosya Connect:Direct düğümünden Connect:Direct köprüsüne aktarılmayı tamamladığında, Connect:Direct köprüsü dosyayı hedef aracı FTE_Z ' ye gönderir ve ardından geçici konumdan siler. Bu işlem, belirtilen her kaynak dosya için yinelenir.

İlgili kavramlar

“Connect:Direct köprüsü” sayfa 324

Var olan bir IBM Sterling Connect:Direct ağına/ağından dosya aktarabilirsiniz. Dosyaları MFT ile IBM Sterling Connect:Direct arasında aktarmak için Managed File Transfer bileşeni olan Connect:Direct köprüsünü kullanın.

İlgili başvurular

[MFT agent.properties dosyası](#)

Joker karakter kullanarak birden çok dosyayı Connect:Direct ' e aktarma

Bir Managed File Transfer aracısından Connect:Direct düğümüne birden çok dosya aktarmak için Connect:Direct köprüsünü kullanın. **fteCreateTransfer** komutuna sağladığınız kaynak belirtiminde genel arama karakterleri kullanabilirsiniz. Genel arama karakterlerini içeren tüm Managed File Transfer aktarımlarında olduğu gibi, dosya yolunun yalnızca son kısmı genel arama karakteri içerebilir. Örneğin, /abc/def* geçerli bir dosya yoludur ve /abc*/def geçerli değildir.

Başlamadan önce

Bir dosyayı aktarmadan önce, Managed File Transfer bileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Daha fazla bilgi için [Connect:Direct köprüsünü yapılandırmabaşlıklı](#) konuya bakın.

Bu görev hakkında

Bu örnekte, kaynak aracıya FTE_AGENT adı verilir ve Connect:Direct köprü aracı CD_BRIDGE olarak adlandırılır. Hedef Connect:Direct düğümüne CD_NODE1 adı verilir. Aktarılacak dosyalar, FTE_AGENT ' in bulunduğu sistemdeki /reports dizininde bulunur. Yalnızca report ile başlayan ve ardından iki karakter ve .log soneki gelen adlara sahip dosyalar aktarılır. Örneğin, /reports/report01.log kütüğü aktarılır, ancak /reports/report1.log kütüğü aktarılmaz. Dosyalar, CD_NODE1 ' in çalıştığı sistemdeki /home/fred dizinine aktarılır.

Yordam

1. `fteCreateTransfer` komutunu, `connect_direct_node_name:directory_path` formundaki **-dd** (hedef dizin) parametresi deęeriyle birlikte kullanın. **-da** (hedef aracı) parametresi için `Connect:Direct` köprü aracısını belirtin.

Not: `connect_direct_node_name` ile belirtilen `Connect:Direct` düęümü, `Connect:Direct` köprüsünün bir parçası olarak çalışan `Connect:Direct` düęümüne deęil, dosyaların aktarılmasını istedięiniz düęümdür.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE  
-dd CD_NODE1:/home/fred "/reports/report??.log"
```

Daha fazla bilgi için bkz. **fteCreateTransfer**: yeni bir dosya aktarımı başlatma.

2. Kaynak aracı `FTE_AGENT`, `/reports/report??.log` kalıbıyla eşleşen ilk dosyayı `Connect:Direct` köprü aracısı `CD_BRIDGE` 'ye aktarır. `Connect:Direct` köprü aracısı, dosyayı geçici olarak `cdTmpDir` özellięi tarafından tanımlanan konumda saklar. Dosya, kaynak araçından `Connect:Direct` köprüsüne tamamen aktarıldığında, `Connect:Direct` köprü aracısı dosyayı `cdNode` aracı özellięi tarafından tanımlanan `Connect:Direct` düęümüne gönderir. Bu düęüm, dosyayı hedef `Connect:Direct` düęümü `CD_NODE1` 'e gönderir. `Connect:Direct` Köprü aracısı, iki `Connect:Direct` düęümü arasındaki aktarım tamamlandığında dosyayı geçici konumdan siler. Bu işlem, genel arama karakteri kalıbıyla eşleşen her kaynak dosya için yinelenir `/reports/report??.log`.

Not: `/reports/report??.log` kalıbıyla eşleşen dosyaların listesi, kaynak aracı `FTE_AGENT` 'in bulunduğu sistemin işletim sistemine baęlı olarak deęişir.

- Kaynak aracı, Windows işletim sistemine sahip bir sistemde bulunuyorsa, kalıp eşleştirme büyük ve küçük harfe duyarlı deęildir. Örneğin, `/reports` dizinindeki tüm dosyaları `report` biçiminde bir dosya adıyla ve ardından iki karakter ve `.log` son ekiyle, harflerin bulunduğu büyük ve küçük harfe bakılmaksızın eşleştirir. Örneğin, `Report99.log` bir eşleştirmedir.
- Kaynak aracı Linux ya da UNIX işletim sistemine sahip bir sistemde bulunuyorsa, kalıp eşleştirme büyük ve küçük harfe duyarlıdır. Örneğin, yalnızca `/reports` dizinindeki dosya adı `report` olan dosyalarla ve ardından iki karakter ve `.log` son ekiyle eşleşir. Örneğin, `reportAB.log` bir eşleşmedir, ancak `reportAB.LOG` ve `Report99.log` eşleşmez.

İlgili kavramlar

[“Connect:Direct köprüsü” sayfa 324](#)

Var olan bir IBM Sterling Connect:Direct aęına/aęından dosya aktarabilirsiniz. Dosyaları MFT ile IBM Sterling Connect:Direct arasında aktarmak için Managed File Transfer bileşeni olan `Connect:Direct` köprüsünü kullanın.

İlgili görevler

[MFT ile genel arama karakterlerinin kullanılması](#)

[“Connect:Direct düęümüne dosya aktarılması” sayfa 326](#)

`Connect:Direct` köprüsünü kullanarak bir dosyayı Managed File Transfer aracısından `Connect:Direct` düęümüne aktarabilirsiniz. Hedef aracı olarak `Connect:Direct` köprü aracısını belirterek ve `connect_direct_node_name:file_path` formunda hedef dosyayı belirterek, aktarma hedefi olarak bir `Connect:Direct` düęümü belirtin.

[“Birden çok dosyanın Connect:Direct düęümüne aktarılması” sayfa 329](#)

`Connect:Direct` köprüsünü kullanarak bir Managed File Transfer Agent düęümünden `Connect:Direct` düęümüne birden çok dosya aktarabilirsiniz. Birden çok dosya aktarımının hedefi olarak bir `Connect:Direct` düęümünü kullanmak için, hedef aracı olarak `Connect:Direct` köprü aracısını belirtin ve `connect_direct_node_name:directory_path` biçiminde hedef dizini belirtin.

[“Connect:Direct düęümünden birden çok dosyanın aktarılması” sayfa 330](#)

`Connect:Direct` köprüsünü kullanarak `Connect:Direct` düęümünden Managed File Transfer Agent düęümüne birden çok dosya aktarabilirsiniz. Kaynak aracı olarak `Connect:Direct` köprü aracısını belirterek ve formda bir ya da daha fazla kaynak belirtimini belirterek çoklu dosya aktarımının kaynaęı olarak bir `Connect:Direct` düęümünü belirtebilirsiniz `connect_direct_node_name:file_path`.

İlgili başvurular

[MFT agent.properties dosyası](#)

Connect:Direct düğümüne/düğümünden aktarma için kurtarma ve yeniden başlatma

Managed File Transfer , bir aktarma sırasında IBM Sterling Connect:Direct düğümünüze bağlanamayabilir; örneğin, düğüm kullanılamaz duruma gelirse. Managed File Transfer , aktarımı kurtarmaya çalışır ya da aktarma başarısız olur ve bir hata iletisi üretilir.

Connect:Direct düğümü kullanılamazsa

Connect:Direct düğümü kullanılamaz duruma gelirse; örneğin, bir ağ ya da güç kesintisi nedeniyle, Managed File Transfer bir dosya aktarımını aşağıdaki şekillerde kurtarır:

- Managed File Transfer , daha önce bu aktarma isteğinin bir parçası olarak Connect:Direct düğümüne başarıyla bağlanmamışsa, aktarma işlemi **cdMaxConnectionRetries** ve **recoverableTransferRetryInterval properties** değerlerinin belirlediği bir süre boyunca yeniden denir. Bu özellikler, Connect:Direct köprü aracı için `agent.properties` dosyasında belirtilir. Aktarma başarısız olur ve başarısız deneme sayısı **cdMaxConnectionRetries property** değerine ulaştıktan sonra bir hata iletisi üretilir. Varsayılan olarak, aktarma girişimi süresiz olarak, girişimler arasında 60 saniye ile gerçekleştirilmeye çalışılır.
- Managed File Transfer , daha önce bu aktarma isteğinin bir parçası olarak Connect:Direct düğümüne başarıyla bağlandıysa, aktarma işlemi **cdMaxPartialWorkConnectionRetries** ve **recoverableTransferRetryInterval** özelliklerinin değerlerinin belirlediği bir süre boyunca yeniden denir. Aktarma başarısız olur ve başarısız girişim sayısı **cdMaxPartialWorkConnectionRetries** özelliğinin değerine ulaştıktan sonra bir hata iletisi üretilir. Varsayılan olarak, aktarma girişimi süresiz olarak, girişimler arasında 60 saniye ile gerçekleştirilmeye çalışılır.
- Belirli Connect:Direct düğüm hatası tiplerinde (örneğin, düğümün zorla durdurulması gibi), düğüm kurtarıldığında Connect:Direct işlemleri `HeId Due to Error (HE)` durumuna girer. After the node recovers, Managed File Transfer automatically resumes any Connect:Direct processes that are related to the file transfer and have a status of HE.
- Aktarma başarısız olursa, aktarıma ilişkin geçici dosyalar Connect:Direct köprüsünü barındıran sistemden silinir. Bu geçici dosyaların konumu, **cdTmpDir** özelliği tarafından tanımlanır.
- Aktarma Managed File Transfer 'dan Connect:Direct' a gerçekleşirse ve bir silme kaynağı belirtilirse, aktarma başarısız olursa kaynak dosyalar silinmez.

Connect:Direct düğümü kullanıcı kimlik bilgileri geçersizse

Managed File Transfer , kullanıcının kimlik bilgileri düğüm tarafından reddedildiği için Connect:Direct düğümüne bağlanamazsa, aktarma başarısız olur ve bir hata iletisi üretilir. Bu durumda, Connect:Direct düğümü için doğru kullanıcı kimlik bilgilerini sağladığınızı doğrulayın. Daha fazla bilgi için bkz. [Connect:Direct için kimlik bilgilerini eşleme](#).

Connect:Direct köprü aracı kullanılamaz duruma gelirse

Connect:Direct köprü aracı kullanılamaz duruma gelirse, devam eden dosya aktarımları standart Managed File Transfer aktarımlarla aynı şekilde kurtarılır. Daha fazla bilgi için bkz. [“MFT kurtarma ve yeniden başlatma” sayfa 340](#).

İlgili kavramlar

[“Connect:Direct köprüsü” sayfa 324](#)

Var olan bir IBM Sterling Connect:Direct ağına/ağından dosya aktarabilirsiniz. Dosyaları MFT ile IBM Sterling Connect:Direct arasında aktarmak için Managed File Transfer bileşeni olan Connect:Direct köprüsünü kullanın.

[“MFT kurtarma ve yeniden başlatma” sayfa 340](#)

Temsilciniz ya da kuyruk yöneticiniz herhangi bir nedenle (örneğin, bir güç ya da ağ arızası nedeniyle) kullanılamıyorsa, Managed File Transfer bu senaryolarda aşağıdaki gibi kurtarma gerçekleştirir:

İlgili görevler

[Connect:Direct köprüsünün yapılandırılması](#)

İlgili başvurular

[MFT agent.properties dosyası](#)

Dosya aktarma isteğinden kullanıcı tanımlı bir Connect:Direct işleminin sunulması

Dosya aktarımının bir parçası olarak kullanıcı tanımlı bir Connect:Direct işlemini çağıran Connect:Direct köprü aracısından geçen bir aktarma isteği gönderebilirsiniz.

Varsayılan olarak, Connect:Direct köprüsünden geçen bir aktarma için bir dosya aktarma isteği gönderdiğinizde, Connect:Direct köprü aracı dosyayı uzak Connect:Direct düğümüne ya da uzak düğümünden aktarmak için kullanılan Connect:Direct işlemini oluşturur.

Ancak, Connect:Direct köprü aracısını `ConnectDirectProcessDefinition.xml` dosyasını kullanarak kullanıcı tanımlı bir Connect:Direct işlemini çağırmak için yapılandırabilirsiniz.

ConnectDirectProcessDefinition.xml dosyası

fteCreateCDAgent komutu, aracı yapılandırma dizininde

`ConnectDirectProcessDefinitions.xml` dosyasını oluşturur `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name`. Connect:Direct köprü aracısından kullanıcı tanımlı Connect:Direct işlemlerini çağırmadan önce, bu dosyayı düzenleyerek süreç tanımlamalarını ayarlamanız gerekir.

Dosya, bir aktarımın parçası olarak çağrılan bir ya da daha fazla Connect:Direct işleminin konumunu içeren bir ya da daha fazla işlem kümesini tanımlar. Her süreç kümesi bir dizi koşul içerir. Aktarma, süreç kümesinin tüm koşullarını karşılıyorsa, aktarım tarafından hangi Connect:Direct işlemlerinin çağrılacağını belirtmek için süreç kümesi kullanılır. Daha fazla bilgi için bkz [“Connect:Direct işleminin ConnectDirectProcessDefinition.xml dosyasını kullanarak başlatılacağını belirtme” sayfa 335](#).

İç simgesel değişkenler

Managed File Transfer tarafından tanımlanan içsel sembolik değişkenleri, değerleri kullanıcı tanımlı Connect:Direct süreçlerine değiştirmek için kullanabilirsiniz. Connect:Direct adlandırma kuralını izlemek için, Managed File Transfer tarafından kullanılan tüm içsel simgesel değişkenlerin biçimi `%FTE`, onu izleyen beş büyük harfli alfasayısal karakterdir.

Dosyaları Connect:Direct düğümünden Connect:Direct köprü sistemine aktarmak için bir işlem oluştururken, Connect:Direct işleminde `TO FILE` değeri olarak `%FTETFILE` içsel değişkenini kullanmalısınız. Dosyaları Connect:Direct köprü sisteminden Connect:Direct düğümüne aktarmak için bir işlem yaratırken, Connect:Direct işleminde `%FTEFFILE` içsel değişkenini `FROM FILE` değeri olarak kullanmanız gerekir. Bu değişkenler, Connect:Direct Bridge aracısının Managed File Transfer ağına/ağından aktarma işlemleri için kullandığı geçici dosya yollarını içerir.

İçsel simgesel değişkenlerle ilgili daha fazla bilgi için Connect:Direct ürün belgelerine bakın.

Örnek Connect:Direct süreçleri

Managed File Transfer, örnek Connect:Direct işlemleri sağlar. Bu örnekler şu dizinde bulunur: `MQ_INSTALLATION_PATH/mqft/samples/ConnectDirectProcessTemplates`.

İlgili görevler

[“Connect:Direct işleminin ConnectDirectProcessDefinition.xml dosyasını kullanarak başlatılacağını belirtme” sayfa 335](#)

Managed File Transfer aktarımının bir parçası olarak başlatılacak Connect:Direct sürecini belirtin.

Managed File Transfer, süreç tanımlamalarını belirtmek için düzenleyebileceğiniz bir XML dosyası sağlar.

“Managed File Transfer tarafından çağrılan Connect:Direct süreçlerinde içsel sembolik değişkenlerin kullanılması” sayfa 336

Bir Managed File Transfer aktarma işleminden kullanıcı tanımlı bir Connect:Direct sürecini çağırabilir ve süreç tanımlamasında içsel simgesel değişkenleri kullanarak aktarma işleminden Connect:Direct sürecine bilgi aktarabilirsiniz.

İlgili başvurular

Connect:Direct süreç tanımlaması dosyası biçimi

Kullanıcı tanımlı Connect:Direct süreçleriyle kullanılacak yerine koyma değişkenleri

Connect:Direct işleminin ConnectDirectProcessDefinition.xml dosyasını kullanarak başlatılacağını belirtme

Managed File Transfer aktarımının bir parçası olarak başlatılacak Connect:Direct sürecini belirtin.

Managed File Transfer , süreç tanımlamalarını belirtmek için düzenleyebileceğiniz bir XML dosyası sağlar.

Bu görev hakkında

fteCreateCDAgent komutu, aracı yapılandırma dizininde

ConnectDirectProcessDefinitions.xml dosyasını oluşturur `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name`. Connect:Direct köprü aracısından kullanıcı tanımlı Connect:Direct işlemlerini çağırılmadan önce, bu dosyayı düzenleyerek süreç tanımlamalarını ayarlamanız gerekir.

Connect:Direct köprüsü üzerinden bir aktarma işleminin parçası olarak çağırılmak istediğiniz her işlem için aşağıdaki adımları gerçekleştirin:

Yordam

1. Connect:Direct köprü aracısının aktarma işleminin bir parçası olarak çağırmasını istediğiniz Connect:Direct işlemi tanımlayın ve süreç şablonunu bir dosyaya kaydedin.
2. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml` dosyasını bir metin düzenleyicisinde açın.
3. Bir `<processSet>` ögesi oluşturun.
4. `<processSet>` ögesinin içinde bir `<condition>` ögesi yaratın.
5. `<condition>` ögesinin içinde, 1. adımda tanımladığınız Connect:Direct işlemi çağırılmak için aktarma isteğinin eşleşmesi gereken bir koşulu tanımlayan bir ya da daha çok öge yaratın. Bu ögeler `<match>` ögeleri ya da `<defined>` ögeleri olabilir.
 - Bir değişkenin değerinin bir kalıpla eşleşmesi gerektiğini belirtmek için `<match>` ögesini kullanın. Aşağıdaki özniteliklerle `<match>` ögesini yaratın:
 - `variable` -değeri karşılaştırılan değişkenin adı. Değişken, içsel bir simgedir. Daha fazla bilgi için Kullanıcı tanımlı Connect:Direct süreçlerle kullanılacak yerine koyma değişkenleri başlıklı konuya bakın.
 - `value` -belirtilen değişkenin değeriyle karşılaştırılacak örüntü.
 - İsteğe bağlı: `pattern` - `value` özniteliğinin değeri tarafından kullanılan örüntü tipi. Bu örüntü tipi `wildcard` ya da `regex` olabilir. Bu öznitelik isteğe bağlıdır ve varsayılan değer `wildcard` dir.
 - Bir değişkenin tanımlı bir değeri olması gerektiğini belirtmek için `<defined>` ögesini kullanın. Aşağıdaki öznitelikle `<defined>` ögesini yaratın:
 - `variable` -bir değer tanımlanmış olması gereken değişkenin adı. Değişken, içsel bir simgedir. Daha fazla bilgi için Kullanıcı tanımlı Connect:Direct süreçlerle kullanılacak yerine koyma değişkenleri başlıklı konuya bakın.

`<condition>` ögesinde belirtilen koşullar mantıksal AND işaretiyle birleştirilir. Connect:Direct köprü aracısının bu `<processSet>` ögesi tarafından belirtilen süreci çağırması için tüm koşulların karşılanması gerekir. Bir `<condition>` ögesi belirtmezseniz, süreç kümesi tüm aktarımlarıyla eşleşir.

6. <processSet> ögesinin içinde bir <process> ögesi yaratın.

7. <process> ögesinin içinde bir <transfer> ögesi yaratın.

Aktarma ögesi, Connect:Direct köprü aracısının aktarımın bir parçası olarak çağırdığı Connect:Direct işlemini belirtir. Aşağıdaki öznitelikle <transfer> ögesini yaratın:

- process- -Adım 1 'de tanımladığınız Connect:Direct sürecinin konumu. Bu dosyanın konumu, mutlak bir yolla ya da *MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name* diziniyle görel olarak belirtilir.

Sonuçlar

Bir koşul eşleşmesini ararken, Connect:Direct köprü aracı dosyanın başından sonuna kadar arar. Bulunan ilk eşleşme, kullanılan eşleştirmesidir.

İlgili görevler

[Connect:Direct köprüsünün yapılandırılması](#)

İlgili başvurular

[Connect:Direct süreç tanımlaması dosyası biçimi](#)

[fteCreateCDAgent: Connect:Direct köprü aracı oluşturma](#)

Managed File Transfer tarafından çağrılan Connect:Direct süreçlerinde içsel sembolik değişkenlerin kullanılması

Bir Managed File Transfer aktarma işleminden kullanıcı tanımlı bir Connect:Direct sürecini çağırabilir ve süreç tanımlamasında içsel simgesel değişkenleri kullanarak aktarma işleminden Connect:Direct sürecine bilgi aktarabilirsiniz.

Bu görev hakkında

Bu örnek, Managed File Transfer aktarımdan kullanıcı tanımlı Connect:Direct bir sürece bilgi aktarmak için içsel sembolik değişkenleri kullanır. Managed File Transfer tarafından kullanılan içsel sembolik değişkenlerle ilgili daha fazla bilgi için [Kullanıcı tanımlı Connect:Direct süreçlerle kullanılacak yerine koyma değişkenleri](#) başlıklı konuya bakın.

Bu örnekte, dosya bir Managed File Transfer Agent düğümünden Connect:Direct köprü düğümüne aktarılır. Aktarımın ilk kısmı Managed File Transfer tarafından gerçekleştirilir. Aktarımın ikinci kısmı, kullanıcı tanımlı bir Connect:Direct işlemi tarafından gerçekleştirilir.

Yordam

1. İçsel sembolik değişkenleri kullanan bir Connect:Direct süreci oluşturun.

```
%FTEPNAME PROCESS
  SNODE=%FTESNODE
  PNODEID=(%FTEPUSER,%FTEPPASS)
  SNODEID=(%FTESUSER,%FTESPASS)

COPY001 COPY
  FROM (
    FILE=%FTEFFILE
    DISP=%FTEFDISP
  )
  TO (
    FILE=%FTETFILE
    DISP=%FTETDISP
  )
PEND
```

2. Bu işlemi şu konumdaki bir metin dosyasına kaydedin: *MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent/Example.cdp*

3. *ConnectDirectProcessDefinition.xml* dosyasını, Adım 1 'de oluşturduğunuz Connect:Direct sürecini çağıran bir kural içerecek şekilde düzenleyin.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:cdprocess xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
```



```

xsi:schemaLocation="http://wmqfte.ibm.com/
ConnectDirectProcessDefinitions ConnectDirectProcessDefinitions.xsd">

<tns:processSet>
  <tns:condition>
    <tns:match variable="%FTESNODE" value="TOBERMORY" pattern="wildcard" />
  </tns:condition>
  <tns:process>
    <tns:transfer process="Example.cdp" />
  </tns:process>
</tns:processSet>

</tns:cdprocess>

```

Bu örnekte, kaynak ya da hedef Connect:Direct düğümü TOBERMORY sahip Connect:Direct köprü aracısına bir aktarma isteği gönderildiyse, Example.cdp Connect:Direct işlemi çağrılır.

4. Adım 3 'teki ConnectDirectProcessDefinition.xml dosyasında tanımladığınız koşulları karşılayan bir dosya aktarma isteği gönderin.

Örneğin,

```

fteCreateTransfer -sa ORINOCO -da CD_BRIDGE
                  -sm QM_WIMBLEDON -dm QM_COMMON
                  -de overwrite -df TOBERMORY:/home/bulgaria/destination.txt
                  -sd leave c:\bungo\source.txt

```

Bu örnekte, hedef Connect:Direct düğümü TOBERMORY 'dir. Bu düğüm, aktarımda ikincil düğümdür ve %FTESNODE değeri TOBERMORY olarak ayarlanır. Bu komut, ConnectDirectProcessDefinition.xml dosyasında ayarlanan koşulla eşleşir.

5. Managed File Transfer , kaynak dosyayı Connect:Direct köprü aracısıyla aynı sistemdeki geçici bir konuma aktarır.
6. Connect:Direct köprü aracısı, aktarma isteği ve yapılandırma bilgilerindeki bilgilerden özgün sembolik değişkenlerin değerlerini ayarlar.

İçsel simgesel değişkenler aşağıdaki değerlere ayarlanır:

- %FTEPNAME=*process_name* -Bu değer, Connect:Direct köprü aracısı tarafından oluşturulan 8 karakterlik bir işlem adıdır.
- %FTESNODE=TOBERMORY-Bu değer, **fteCreateTransfer** komutunun **-df** değiştirgesinden ayarlanır.
- %FTEPUSER, =*primary_node_user* -Bu bilgi ConnectDirectCredentials.xml dosyasından alınır.
- %FTEPPASS=*primary_node_user_password* -Bu bilgiler ConnectDirectCredentials.xml dosyasından alınır.
- %FTESUSER, =*secondary_node_user* -Bu bilgi ConnectDirectCredentials.xml dosyasından alınır.
- %FTESPASS=*secondary_node_user_password* -Bu bilgi ConnectDirectCredentials.xml dosyasından alınır.
- %FTEFFILE =*temporary_location* -Bu değer, dosyanın Connect:Direct köprü aracısıyla aynı sistemdeki geçici konumudur.
- %FTEFDISP=*ayrıl*-Bu değer, **fteCreateTransfer** komutunun **-sd** değiştirgesinden belirlenir.
- %FTETFILE=/home/bulgaria/destination.txt -Bu değer, **fteCreateTransfer** komutunun **-df** değiştirgesinden belirlenir.
- %FTETDISP=*overwrite*-Bu değer, **fteCreateTransfer** komutunun **-de** değiştirgesinden ayarlanır.

7. Connect:Direct işlemi, Connect:Direct köprü düğümünde başlatılır. Connect:Direct , dosyayı Connect:Direct köprü sistemindeki geçici konumdan, Connect:Direct düğümünün TOBERMORY çalıştığı sistemdeki hedefe /home/bulgaria/destination.txt aktarır.

İlgili kavramlar

[“Dosya aktarma isteğinden kullanıcı tanımlı bir Connect:Direct işleminin sunulması” sayfa 334](#)

Dosya aktarımının bir parçası olarak kullanıcı tanımlı bir Connect:Direct işlemini çağıran Connect:Direct köprü aracısından geçen bir aktarma isteği gönderebilirsiniz.

İlgili başvurular

[Kullanıcı tanımlı Connect:Direct süreçleriyle kullanılacak yerine koyma değişkenleri](#)

Aktarma isteklerini Connect:Direct sunmak için Managed File Transfer süreçlerin kullanılması

Bir Connect:Direct işleminden Connect:Direct köprü aracısına bir aktarma isteği gönderebilirsiniz. Managed File Transfer , Connect:Direct sürecindeki bir **RUN TASK** deyiminden çağrılabilen komutlar sağlar.

Managed File Transfer , Connect:Direct işlemleriyle kullanılmak üzere aşağıdaki komutları sağlar:

ftetag

Bu komutu, aktarım için gerekli denetim bilgilerini yaratmak üzere **ftebxfer** ya da **ftecxfer** komutunun önündeki bir adımda belirtin. Bu komut, aktarmanın kaynak belirtimini parametre olarak alır. Kaynak belirtiminin biçimi hakkında bilgi için bkz. [fteCreateTransfer: yeni bir dosya aktarımı başlatma](#).

ftebxfer

Aktarma isteğinin sunulduğu kuyruk yöneticisi, komutu gönderen Connect:Direct düğümüyle aynı sistemde olduğunda dosya aktarma isteği yaratmak için bu komutu belirleyin. Bu komut, **fteCreateTransfer** komutuyla aynı parametreleri alır. Bu parametrelerle ilgili bilgi için bkz. [fteCreateTransfer: Yeni bir dosya aktarımı başlatma](#). Bu komutun ek bir parametresi de vardır:

-qmgrname

Gereklidir. Komutun gönderileceği kuyruk yöneticisinin adı.

ftecxfer

Aktarma isteğinin sunulduğu kuyruk yöneticisi, komutu gönderen Connect:Direct düğümüne farklı bir sistemde bulunduğunda dosya aktarma isteği yaratmak için bu komutu belirleyin. Bu komut, **fteCreateTransfer** komutuyla aynı parametreleri alır. Parametreler hakkında bilgi için bkz. [fteCreateTransfer: yeni bir dosya aktarımı başlatma](#). Bu komutun ayrıca üç ek parametresi vardır:

-qmgrname

Gereklidir. Komutun gönderileceği kuyruk yöneticisinin adı.

-connname

Gereklidir. IBM MQ CONNAME biçiminde belirtilen, komutun gönderileceği kuyruk yöneticisinin anasistemi ve kapısı. Örneğin, host.example.com(1337).

-channelname

İsteğe Bağlı. Komutun gönderileceği kuyruk yöneticisine bağlanmak için kullanılacak kanalın adı. Bu belirtilmezse, varsayılan değer olarak SYSTEM.DEF.SVRCONN kullanılır.

İlgili görevler

[“Connect:Direct İstekçisini kullanarak Managed File Transfer ' i çağıran bir Connect:Direct süreci oluşturma ve gönderme” sayfa 338](#)

Connect:Direct İstekte bulunan, Managed File Transfer' i çağıran bir Connect:Direct süreci yaratmak ve göndermek için kullanabileceğiniz bir grafik kullanıcı arabirimidir.

İlgili başvurular

[Örnek: MFT komutlarını çağıran bir Connect:Direct işlem dosyası](#)

Connect:Direct İstekçisini kullanarak Managed File Transfer ' i çağıran bir Connect:Direct süreci oluşturma ve gönderme

Connect:Direct İstekte bulunan, Managed File Transfer' i çağıran bir Connect:Direct süreci yaratmak ve göndermek için kullanabileceğiniz bir grafik kullanıcı arabirimidir.

Bu görev hakkında

Bu kısımda, Managed File Transfer **ftecxfer** komutunu ya da **ftebxfer** komutunu çağıran bir Connect:Direct işleminin nasıl oluşturulacağı açıklanmaktadır. Aktarma isteğinin sunulduğu kuyruk yöneticisi, komutu gönderen Connect:Direct düğümüne farklı bir sistemde bulunduğu **ftecxfer** komutunu kullanın. Aktarma isteğinin sunulduğu kuyruk yöneticisi, komutu gönderen Connect:Direct düğümüyle aynı sistemde bulunuyorsa **ftebxfer** komutunu kullanın. **ftecxfer** komutu, aktarma işleminin kaynak aracısının aracı kuyruk yöneticisiyle istemci bağlantısı kurar. **ftecxfer** komutunu çağırmadan önce, **ftetag** komutunu çağırmanız ve kaynak belirtimi bilgilerini iletmeniz gerekir. Bu, işlemin Managed File Transfer' den başlatılan aktarımlarla aynı şekilde günlüğe kaydedilmesini ve denetlenmesini sağlar.

Yordam

1. Connect:Direct İstekçiyi başlatın.
2. Panonun **Düğümler** sekmesinde, işlemin birincil düğümü olarak kullanılan Connect:Direct düğümünü seçin.
3. **Dosya > Yeni > Süreç** seçeneğini belirleyin. **Süreç Özellikleri** penceresi açılır.
4. **Ad:** alanında işlemin adını yazın.
5. **Snod** > **Ad:** listesinden ikincil düğümü seçin.
6. **Snod** > **Operating System:** (İşletim Sistemi) listesinden ikincil düğümün işletim sistemini seçin.
7. İsteğe bağlı: Bu pencerede gerek duyduğunuz diğer bilgileri tamamlayın.
8. **Tamam**'ı tıklatın. **Süreç Özellikleri** penceresi kapanır.
9. Managed File Transfer **ftetag** komutunu çalıştıran bir deyim oluşturun.
 - a) **End** deyiminin **Process** (Süreç) penceresini sağ tıklatın.
 - b) **Ekle** > **Görevi Çalıştır** seçeneğini belirleyin. **Görev Deyimini Çalıştır** penceresi açılır.
 - c) **Etiket:** alanına Tag yazın.
 - d) **İsteğe Bağlı Parametreler ya da Komutlar** alanına `pgm(MQ_INSTALLATION_PATH/bin/ftetag) args(source_specification)` yazın. *kaynak_belirtimi* biçimi hakkında daha fazla bilgi için bkz. **fteCreateTransfer:** yeni bir dosya aktarımı başlatma.
 - e) **Tamam**'ı tıklatın. **Görev Deyimini Çalıştır** penceresi kapanır.
10. Managed File Transfer **ftecxfer** ya da **ftebxfer** komutunu çalıştıran bir deyim oluşturun.
 - a) **End** deyiminin **Process** (Süreç) penceresini sağ tıklatın.
 - b) **Ekle** > **Görevi Çalıştır** seçeneğini belirleyin. **Görev Deyimini Çalıştır** penceresi açılır.
 - c) **Etiket:** alanına Transfer yazın.
 - d) **İsteğe Bağlı Parametreler ya da Komutlar** alanında, seçtiğiniz komuta bağlı olarak `pgm(MQ_INSTALLATION_PATH/bin/ftecxfer) args(parameters)` ya da `pgm(MQ_INSTALLATION_PATH/bin/ftebxfer) args(parameters)` yazın. **ftecxfer** ve **ftebxfer** komutları tarafından kullanılan değiştirgeler, **fteCreateTransfer** komutu tarafından kullanılan değiştirgelerle ve **ftecxfer** ve **ftebxfer** komutlarına özgü bazı ek değiştirgelerle aynıdır. Daha fazla bilgi için bkz. **fteCreateTransfer:** Yeni bir dosya aktarımı başlatma ve "Aktarma isteklerini Connect:Direct sunmak için Managed File Transfer süreçlerin kullanılması" sayfa 338.
 - e) **Tamam**'ı tıklatın. **Görev Deyimini Çalıştır** penceresi kapanır.
11. İsteğe bağlı: Gerek duyduğunuz ek deyimleri yaratın.
12. Süreci sunun.
 - a) **Süreç** penceresinde farenin sağ düğmesini tıklatın.
 - b) **Submit**(Gönder) seçeneğini belirleyin. **Connect:Direct Attach** penceresi açılır.
 - c) İşlemi çalıştırmak için kullanılacak kullanıcı adını ve parolayı girin.
 - d) **Tamam**'ı tıklatın.

İlgili kavramlar

“Aktarma isteklerini Connect:Direct sunmak için Managed File Transfer süreçlerin kullanılması” sayfa 338
Bir Connect:Direct işleminden Connect:Direct köprü aracısına bir aktarma isteği gönderebilirsiniz.
Managed File Transfer , Connect:Direct sürecindeki bir **RUN TASK** deyiminden çağrılabilen komutlar sağlar.

IBM Integration Bus içinden MFT ile çalışma

FTEOutput ve FTEInput düğümlerini kullanarak IBM Integration Bus içinden Managed File Transfer ile çalışabilirsiniz.

- Managed File Transfer komutunu kullanarak bir dosyayı ağ üzerinden aktarmak için FTEInput düğümünü kullanın ve bu dosyayı Integration Bus akışının bir parçası olarak işleyin.
- Integration Bus akışı tarafından çıkışı yapılan bir dosyayı ağdaki başka bir yere aktarmak için FTEOutput düğümünü kullanın.

Aracı aracısına ya da aracıdan dosya aktaran araçlar herhangi bir Managed File Transfer düzeyinde olabilir.

Daha fazla bilgi için [IBM Integration Bus ürün belgelerine](#) bakın.

MFT kurtarma ve yeniden başlatma

Temsilciniz ya da kuyruk yöneticiniz herhangi bir nedenle (örneğin, bir güç ya da ağ arızası nedeniyle) kullanılamıyorsa, Managed File Transfer bu senaryolarda aşağıdaki gibi kurtarma gerçekleştirir:

- Genellikle, dosya aktarılırken bir sorun ortaya çıkarsa, Managed File Transfer sorun onarıldıktan sonra bu dosya aktarımını kurtarır ve yeniden başlatır.
- Aracı ya da kuyruk yöneticisi kullanılmadığında, aktarılmakta olan bir dosya silinir ya da değiştirilirse, aktarma başarısız olur ve aktarma günlüğünde hataya ilişkin ayrıntıları belirten bir ileti alırsınız.
- Dosya aktarımı sırasında bir aracı işlemi başarısız olursa, aracıyı yeniden başlattığınızda aktarma devam eder.
- Bir aracı, aracı kuyruk yöneticisiyle bağlantısını kaybederse, aracı kuyruk yöneticisine yeniden bağlanmayı denerken bekler. Aracı kuyruk yöneticisine başarıyla yeniden bağlandığında, yürürlükteki aktarma devam eder.
- Aracı herhangi bir nedenle durdurulursa, bir aracıyla ilişkili kaynak izleyicileri yoklamayı durdurur. Aracı kurtarıldığında, izleme programları da yeniden başlatılır ve kaynak yoklaması devam eder.
- Kaynak yok etme kaynağı delete olan bir dosya aktarımı için, kaynak aracıdan hedef aracıya tüm veriler gönderildikten sonra kurtarma gerçekleşirse, silme işleminden önce kaynak dosyanın kilidi açılır. Bu kilit açma, kaynak dosyanın dosya silinmeden önce değiştirilebileceği anlamına gelir. Bu nedenle, kaynak dosyayı silmek güvenli değil olarak kabul edilir ve aşağıdaki uyarı görüntülenir:

```
BFGTR0075W: The source file has not been deleted because it is possible that the source file was modified after the source file was transferred.
```

Bu durumda, kaynak dosyanın içeriğinin değiştirilmediğini doğrulayın ve kaynak dosyayı el ile silin.

IBM MQ Explorer içinde aktarımlarınızın durumunu denetleyebilirsiniz. Herhangi bir aktarım Stalled olarak görüntülenirse, durdurulmuş durum, aracıda ya da aktarımda yer alan iki aracı arasında bir sorun olduğunu belirttiğinden, düzeltici işlem yapmanız gerekebilir.

İlgili görevler

“Durdurulmuş aktarımların kurtarılması için zamanaşımı ayarlanması” sayfa 341

Bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan, durdurulmuş dosya aktarımları için bir aktarım kurtarma zamanaşımı ayarlayabilirsiniz. Tek bir aktarma için aktarma kurtarma zamanaşımı da ayarlayabilirsiniz. Belirli bir süreyi saniye cinsinden ayarlarsanız, kaynak aracı durmuş bir dosya aktarımını kurtarmaya çalışır ve aracı zamanaşımına ulaştığında aktarma başarısız olur.

Durdurulmuş aktarımların kurtarılması için zamanaşımı ayarlanması

Bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan, durdurulmuş dosya aktarımları için bir aktarım kurtarma zamanaşımı ayarlayabilirsiniz. Tek bir aktarma için aktarma kurtarma zamanaşımı da ayarlayabilirsiniz. Belirli bir süreyi saniye cinsinden ayarlarsanız, kaynak aracı durmuş bir dosya aktarımını kurtarmaya çalışır ve aracı zamanaşımına ulaştığında aktarma başarısız olur.

Bu görev hakkında

IBM MQ 9.1' den, aracının `agent.properties` dosyasına bir aktarma kurtarma zamanaşımı parametresi ekleyerek bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan bir aktarma kurtarma zamanaşımı ayarlayabilirsiniz. Ayrıca, komut satırından ya da IBM MQ Explorer ile ya da Apache Ant görevlerini kullanarak tek bir aktarma için aktarma kurtarma zamanaşımını ayarlayabilirsiniz. `agent.properties` dosyasında bir aktarma kurtarma zamanaşımı değeri ayarlanmışsa, tek bir aktarma için aktarma kurtarma zamanaşımının ayarlanması `agent.properties` dosyasındaki değeri geçersiz kılar.

Aktarma kurtarma zamanaşımı için üç seçenek vardır:

- Aracı, başarıyla tamamlanincaya kadar durdurulmuş aktarımı kurtarmaya çalışmaya devam eder. Aktarma kurtarma zamanaşımı ayarlanmazsa, bu, aracının varsayılan davranışıyla aynıdır.
- Aracı, kurtarma işlemine girildikten hemen sonra aktarımı başarısız olarak işaretler.
- Aracı, aktarma başarısız olarak işaretlenmeden önce belirli bir süre boyunca durdurulmuş aktarımı yeniden denemeye devam eder.

Dosya aktarma kurtarma zamanaşımının ayarlanması isteğe bağlıdır. Ayarlamazsanız, aktarımlar varsayılan davranışı izler. Bu, bir Managed File Transfer kaynak aracısının IBM MQ 9.1 öncesinde olan varsayılan davranışıyla aynıdır; aracı, başarılı oluncaya kadar durdurulmuş bir aktarımı kurtarmaya çalışır.

İlgili kavramlar

“MFT kurtarma ve yeniden başlatma” sayfa 340

Temsilciniz ya da kuyruk yöneticiniz herhangi bir nedenle (örneğin, bir güç ya da ağ arızası nedeniyle) kullanılamıyorsa, Managed File Transfer bu senaryolarda aşağıdaki gibi kurtarma gerçekleştirir:

Aktarma kurtarma zamanaşımı kavramları

Bir kaynak aracısının durmuş bir dosya aktarımını kurtarmaya çalıştığı süreyi saniye cinsinden ayarlayabilirsiniz. Aracı yeniden deneme aralığına ilişkin zamanaşımına ulaştığında aktarma başarısız olursa, aktarma başarısız olur.

Kurtarma zamanaşımı önceliği

fteCreateTransfer, **fteCreateTemplate** ya da **fteCreateMonitor** komutlarıyla ya da IBM MQ Explorer kullanılarak ya da **fte:filespec** içiçe öğede belirtilen bir aktarma için aktarma kurtarma zamanaşımı değeri, kaynak aracıya ilişkin `agent.properties` dosyasında **transferRecoveryTimeout** parametresi için belirtilen değerden önceliklidir.

Örneğin, **fteCreateTransfer** komutu **-rt** parametre ve değer çifti olmadan başlatılırsa, kaynak aracı AGENT1, kurtarma zamanaşımı davranışını belirlemek için `agent.properties` dosyasını **transferRecoveryTimeout** değeri için denetler:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -df C:\import\transferredfile.txt  
C:\export\originalfile.txt
```

`agent.properties` dosyasındaki **transferRecoveryTimeout** parametresi ayarlanmazsa ya da **-1** olarak ayarlanırsa, aracı varsayılan davranışı izler ve başarılı oluncaya kadar aktarımı kurtarmaya çalışır.

Ancak **fteCreateTransfer** komutu **-rt** deęiřtirgesini içeriyorsa, bu deęiřtirgenin deęeri `agent.properties` kütüğündeki deęerden önceliklidir ve aktarma için kurtarma zamanařımı ayarı olarak kullanılır:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt  
C:\export\originalfile.txt
```

Kurtarma zamanařımı sayacı

Kurtarma zamanařımı sayacı, aktarım kurtarma durumuna girdiğinde başlar. SYSTEM.FTE konu dizesi `Log/agent_name/transfer_ID` ile TZE konusu. Aktarma, belirlenen yeniden deneme aralıęı içinde sürdürülür ve kurtarma zamanařımına (sayaç < =kurtarma zamanařımı) ulaşmazsa, sayaç 0 olarak sıfırlanır ve aktarma kurtarma işlemine girerse yeniden başlatılmaya hazırdır.

Sayaç, kurtarma zamanařımı (sayaç == kurtarma zamanařımı) için ayarlanan deęer üst sınırına ulařırsa, aktarma işlemi durdurulur ve kaynak aracı aktarımı başarısız olarak bildirir. Aktarma işleminin kurtarma zamanařımına ulaşması nedeniyle oluşan bu aktarma hatası tipi, RECOVERY TIMEOUT (69) ileti koduyla gösterilir. Aktarımın başarısız olduęunu ve bir ileti, dönüş kodu ve kaynak aracının olay günlüğünü içerdięini belirtmek için, SYSTEM.FTE konusunda `Log/agent_name/transfer_ID` konu dizisiyle başka bir aktarma günlüğü iletileri yayınlanır. Kurtarma sırasında ařağıdaki olaylardan biri olduęunda kaynak aracının olay günlüğü bir iletiyle güncellenir:

- Kurtarma zamanařımı parametresi -1 deęerinden büyük bir deęere ayarlandıęında, aktarma kurtarma işlemine girer. Aracının olay günlüğü, **TransferId** için kurtarma zamanlayıcısının başlatıldıęını ve kaynak aracının kurtarma zamanařımı işlemini başlatmadan önce bekleyeceęi süreyi gösterecek şekilde güncellenir.
- Kurtarma aktarımı sürdürüldüğünde, kaynak aracının olay günlüğü, kurtarmada olan **TransferId** ' un sürdürülmesini belirten yeni bir iletiyle güncellenir.
- Kurtarma işlemi zaman aşımına uğradıęında, kaynak aracının olay günlüğü, kurtarma zamanařımı nedeniyle kurtarma sırasında başarısız olan **TransferId** öęesini gösterecek şekilde güncellenir.

Bu günlük iletileri, kullanıcıların (aboneler ve kaydediciler), aktarma kurtarma zamanařımı nedeniyle başarısız olan aktarımları tanımlamalarını saęlar.

Kurtarma zamanařımı sayacı her zaman kaynak aracıdadır. Ancak, hedef aracı kaynak aracıdan zamanında bilgi alamazsa, kaynak aracıya aktarımı kurtarma işlemine sokması için bir istek gönderebilir. Kurtarma zamanařımı seęeneęinin belirlendięi bir aktarım için, kaynak aracı, isteęi hedef aracıdan aldıęında kurtarma zamanařımı sayacını başlatır.

Kurtarma zamanařımı seęeneęini, başarısız ve kısmen tamamlanmış aktarımları kullanmayan aktarımlar için el ile işleme yine de gereklidir.

Birden çok dosya için tek bir aktarma isteęi sunulduęu ve bazı dosyaların başarıyla tamamlandıęı, ancak bir tanesinin kısmen tamamlandıęı aktarım kümeleri için, aktarma işlemi beklendięi gibi tamamlanmadıęı için başarısız olarak işaretilenir. Kısmen tamamlanan dosya aktarılırken kaynak aracı zamanařımına uğramış olabilir.

Hedef aracının ve dosya sunucusunun hazır olduęundan ve dosya aktarımlarını kabul edecek durumda olduęundan emin olun.

Tüm küme için aktarma isteęini yeniden yayınlamanız gerekir; ancak, bazı dosyalar ilk aktarma girişiminden itibaren hedefte kaldıęı için, yeni isteęi varolan isteęin üzerine yaz seęeneęi belirtildiyse, yeni isteęi yayınlatabilirsiniz. Bu, önceki aktarma girişimindeki tamamlanmamış dosya kümesinin, dosyalar hedefe yeniden yazılmadan önce yeni aktarımın bir parçası olarak temizlenmesini saęlar.

IBM MQ 9.1.5' da, ilk aktarma girişimi başarısız olduktan sonra bir hedefte kalan kısım dosyalarının el ile kaldırılması gerekmez. Bir aktarma için aktarım kurtarma zamanařımı ayarlanırsa, aktarım kurtarma zamanařımına uğursa, kaynak aracı aktarımı [RecoveryTimedOut](#) (Zamanařımına uğradı) durumuna tařır. Aktarma yeniden eřitlendikten sonra, hedef aracı aktarma sırasında yaratılan kısım dosyalarını kaldırır ve kaynak aracıya bir tamamlanma iletileri gönderir.

İzlemeler ve iletiler

Tanılama amacıyla izleme noktaları eklenmiştir. Kurtarma zamanaşımı değeri, yeniden deneme aralığının başlangıcı, sürdürme döneminin başlangıcı ve sayaç sıfırlaması ve aktarmanın zamanaşımına uğrayıp uğramadığı ve başarısız olup olmadığı günlüğe kaydedilir. Bir sorun ya da beklenmeyen bir davranış olması durumunda, kaynak aracı çıkış günlüğü ve izleme dosyalarını toplayabilir ve sorun gidermeye yardımcı olması için IBM desteği tarafından istendiğinde bunları sağlayabilirsiniz.

Aşağıdaki durumlarda bildirilecek iletiler:

- Bir aktarma kurtarma işlemine girer ([BFGTR0081I](#))
- Kurtarma işlemi zamanaşımına uğradığı için aktarım sonlandırıldı ([BFGSS0081E](#))
- Kurtarma işleminden sonra aktarma devam eder ([BFGTR0082I](#))

İlgili kavramlar

“MFT kurtarma ve yeniden başlatma” sayfa 340

Temsilciniz ya da kuyruk yöneticiniz herhangi bir nedenle (örneğin, bir güç ya da ağ arızası nedeniyle) kullanılamıyorsa, Managed File Transfer bu senaryolarda aşağıdaki gibi kurtarma gerçekleştirir:

Bir kaynak aracı için tüm aktarımlar için aktarım kurtarma zamanaşımının ayarlanması

transferRecoveryTimeout parametresini `agent.properties` dosyasına ekleyerek, bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan bir aktarma kurtarma zamanaşımı ayarlayabilirsiniz.

Bu görev hakkında

Bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan bir kurtarma zamanaşımını ayarlamak için, **transferRecoveryTimeout** için parametre ve değer çiftini `agent.properties` dosyasına eklersiniz.

transferRecoveryTimeout parametresi için üç seçenek vardır:

-1

Aracı, aktarım başarılı oluncaya kadar durdurulmuş aktarımı kurtarmaya çalışmaya devam eder. Bu seçeneğin kullanılması, özellik ayarlanmadığında aracının varsayılan davranışının eşdeğeridir.

0

Aracı, kurtarma işlemine girer girmez dosya aktarımını durdurur.

>0

Aracı, belirtilen pozitif tamsayı değeriyle belirlendiği şekilde, saniye cinsinden süre boyunca durdurulmuş aktarımı kurtarma girişiminde bulunmaya devam eder.

`agent.properties` dosyasında yaptığınız değişiklikler yalnızca aracı yeniden başlatıldıktan sonra yürürlüğe girer.

Gerekirse, tek bir aktarma için `agent.properties` kütüğündeki aktarma kurtarma zamanaşımı değerini geçersiz kılabilirsiniz. Daha fazla bilgi için bkz [“Tek tek aktarımlar için aktarma kurtarma zamanaşımının ayarlanması” sayfa 344.](#)

Yordam

- Aracının, başarıyla tamamlanincaya kadar durdurulmuş aktarımı kurtarmaya devam edeceğini belirtmek için, aşağıdaki örnekte gösterildiği gibi **-1** aktarma kurtarma zamanaşımı değerini ayarlayın:

```
transferRecoveryTimeout=-1
```

- Kurtarma işlemi girdikten hemen sonra aracının aktarımı başarısız olarak işaretlediğini belirtmek için, aşağıdaki örnekte gösterildiği gibi, **0** aktarma kurtarma zamanaşımı değerini ayarlayın:

```
transferRecoveryTimeout=0
```

- Aracının, aktarma başarısız olarak işaretlenmeden önce belirli bir süre boyunca durdurulmuş bir aktarımı yeniden denemeye devam etmesini belirtmek için, aracının yeniden denemeye devam etmesini istediğiniz zaman için saniye cinsinden bir aktarım kurtarma zamanaşımı değeri ayarlayın.

Örneğin, aktarım kurtarma zamanaşımı değerinin 21600 olarak ayarlanması, aracının kurtarma işlemine girdiği andan itibaren altı saat boyunca aktarımı kurtarmaya çalıştığı anlamına gelir:

```
transferRecoveryTimeout=21600
```

Bu parametre için değer üst sınırı: 999999999.

Tek tek aktarımlar için aktarma kurtarma zamanaşımının ayarlanması

Komut satırından ya da IBM MQ Explorer ile ya da Apache Ant görevlerini kullanarak tek bir aktarma için aktarma kurtarma zamanaşımını ayarlayabilirsiniz. `agent.properties` dosyasında bir aktarma kurtarma zamanaşımı değeri ayarlanmışsa, tek bir aktarma için aktarma kurtarma zamanaşımının ayarlanması, `agent.properties` dosyasında ayarlanan değeri geçersiz kılar.

Bu görev hakkında

Aşağıdaki durumda tek bir aktarma için aktarma kurtarma zamanaşımı parametresini ayarlayabilirsiniz:

- **fteCreateTransfer** komutunu kullanarak ya da IBM MQ Explorer komutunu kullanarak aktarma yaratılması.
- **fteCreateTemplate** komutunu kullanarak ya da IBM MQ Explorer komutunu kullanarak aktarma şablonu yaratılması.
- **fteCreateMonitor** komutunu kullanarak ya da IBM MQ Explorer komutunu kullanarak kaynak izleyici yaratılması.
- `fte: filecopy` ya da `fte: filetaşıma` Ant görevlerini kullanarak dosyaları kopyalama ya da taşıma.

Tek bir aktarma için aktarma kurtarma zamanaşımı değeri belirlerseniz, bu değer `agent.properties` dosyasında ayarlanan aktarma kurtarma zamanaşımı değerini geçersiz kılar (bkz. [“Bir kaynak aracı için tüm aktarımlar için aktarım kurtarma zamanaşımının ayarlanması” sayfa 343](#)).

Yordam

- Aktarma kurtarma zamanaşımını ayarlamak üzere **fteCreateTransfer** ya da **fteCreateTemplate** komutunu kullanmak için **-rt** parametresi için uygun seçeneği belirleyin:

-1

Aracı, aktarım başarılı oluncaya kadar durdurulmuş aktarımı kurtarmaya çalışmaya devam eder. Bu seçeneğin kullanılması, özellik ayarlanmadığında aracının varsayılan davranışının eşdeğeri.

0

Aracı, kurtarma işlemine girer girmez dosya aktarımını durdurur.

>0

Aracı, belirtilen süre (saniye) boyunca durdurulmuş aktarımı kurtarma girişiminde bulunmaya devam eder.

fteCreateTransfer komutu örnekleri

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt -1 -df C:\import\transferredfile.txt  
C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 0 -df C:\import\transferredfile.txt  
C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt  
C:\export\originalfile.txt
```


fteCreateTemplate komutu örnekleri

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt -1 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 0 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 21600 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

fteCreateMonitor komutu için **-rt** parametresi yoktur. **-rt** değıştirgesini **fteCreateTransfer** komutuyla ayarlarsanız ve **-gt** değıştirgesini de ayarlarsanız, kurtarma zamanasını değıştirgesi **fteCreateTransfer** komutunu çalıştırdığınızda oluşturulan aktarma tanımlamasıyla birlikte XML belgesine eklenir. Kaynak izleyicisi, **fteCreateMonitor** komutunu çalıştırırken bu XML belgesini kullanır. Aşağıdaki örnekte, aktarma kurtarma zamanasını ayrıntıları `task.xml` kütüğüne eklenir:

```
fteCreateMonitor -ma AgentName -md C:\mqmft\monitors -mn Monitor_Name -mt task.xml -tr
"fileSize>=5MB,*.zip"
```

- Aktarma kurtarma zamanasını ayarlamak üzere IBM MQ Explorer Yeni Aktarma, Yeni İzleme Programı ya da Yeni Şablon sihirbazı sayfasını kullanmak için, **Aktarma Kurtarma Zamanasını** (saniye) alanında gerekli seçeneği belirleyin:

Kaynak Aracı Olarak

Kaynak Aracı Olarak seçeneğini belirlerseniz, `agent.properties` dosyasındaki **transferRecoveryTimeout** parametre değeri, ayarlanmışsa kullanılır; tersi durumda, aktarma kurtarma zamanasına ilişkin varsayılan davranış uygulanır.

Sayısal liste kutusu

Sayısal liste kutusuna saniye cinsinden bir süre girerseniz, aracı belirtilen süre boyunca durdurulmuş aktarımı kurtarma girişiminde bulunmaya devam eder.

Yok

Yok seçeneğini belirlerseniz, aktarma kurtarma zamanasını ayarlanmaz ve aracı, aktarma başarılı oluncaya kadar durdurulmuş aktarımı kurtarma girişiminde bulunmaya devam eder.

- Ant görevlerini kullanarak kurtarma zamanasını ayarlamak için. Dosyaları taşımak ya da kopyalamak için **fte:filecopy** ya da **fte:filemove** öğeleriyle birlikte **transferRecoveryTimeout** seçeneğini ve değerini ekleyin; örneğin:

Örnek: fte:filecopy

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="copy.result" transferRecoveryTimeout="0">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filecopy>
```

Örnek: fte:filemove

```
<fte:filemove cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="move.result" transferRecoveryTimeout="21600">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filemove>
```

İlgili kavramlar

[“MFT kurtarma ve yeniden başlatma” sayfa 340](#)

Temsilciniz ya da kuyruk yöneticiniz herhangi bir nedenle (örneğin, bir güç ya da ağ arızası nedeniyle) kullanılamıyorsa, Managed File Transfer bu senaryolarda aşağıdaki gibi kurtarma gerçekleştirir:

İlgili görevler

[“Yeni dosya aktarımı başlatılması” sayfa 233](#)

IBM MQ Explorer ' den ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir gruptaki tek bir dosyayı ya da birden çok dosyayı aktarmayı seçebilirsiniz.

[“IBM MQ Explorer kullanarak dosya aktarma şablonu yaratılması” sayfa 276](#)

IBM MQ Explorer içinden ya da komut satırından bir dosya aktarma şablonu yaratabilirsiniz. Daha sonra şablon ayrıntılarını kullanarak yeni dosya aktarımları oluşturmak için bu şablonu kullanabilir ya da dosya aktarımını başlatmak için şablonu gönderebilirsiniz.

[“MFT kaynaklarının izlenmesi” sayfa 244](#)

Managed File Transfer kaynaklarını izleyebilirsiniz; örneğin, bir kuyruk ya da dizin. Bu kaynaktaki bir koşul karşılandığında, kaynak izleme programı dosya aktarma gibi bir görevi başlatır. IBM MQ Explorer için Managed File Transfer eklentisindeki **fteCreateMonitor** komutunu ya da **İzleme Programları** görünümünü kullanarak bir kaynak izleme programı yaratabilirsiniz.

[“Aktarma Günlüğündeki Dosya Aktarımlarının Durumunun Görüntülenmesi” sayfa 242](#)

IBM MQ Explorer içindeki **Aktarma Günlüğü** ' nü kullanarak dosya aktarımlarının ayrıntılarını görüntüleyebilirsiniz. Bunlar, komut satırından ya da IBM MQ Explorer' den başlatılan aktarımlar olabilir. **Aktarma Günlüğü** ' nde görüntülenenleri de uyarlayabilirsiniz.

İlgili başvurular

[MFT agent.properties dosyası](#)

fteCreateTransfer: yeni bir dosya aktarımı başlatır

fteCreateTemplate: yeni dosya aktarma şablonu yarat

fteCreateMonitor: MFT kaynak izleme programı yaratılması

fte: filecopy Ant görevi

fte: filetaşıma Ant görevi

Linux

Windows

AIX

YönetmeMQ Telemetry

MQ Telemetry , IBM MQ Explorer kullanılarak ya da bir komut satırında yönetilir. Telemetry kanallarını yapılandırmak, telemetry hizmetini denetlemek ve IBM MQ' e bağlı MQTT istemcilerini izlemek için gezgini kullanın. MQ Telemetry güvenliğini JAAS, TLS ve IBM MQ nesne yetki yöneticisini kullanarak yapılandırın.

IBM MQ Explorer kullanarak yönetme

Telemetry kanallarını yapılandırmak, telemetry hizmetini denetlemek ve IBM MQ' e bağlı MQTT istemcilerini izlemek için gezgini kullanın. MQ Telemetry güvenliğini JAAS, TLS ve IBM MQ nesne yetki yöneticisini kullanarak yapılandırın.

Komut satırını kullanarak yönetme

MQ Telemetry , [MQSC komutları](#) kullanılarak komut satırında tam olarak denetlenir.

MQ Telemetry belgelerinde, IBM MQ Telemetry Transport v3 Client uygulamasının temel kullanımını gösteren örnek komut dosyaları da vardır.

Kullanmadan önce [IBM MQ Telemetry Transport örnek programlarındaki örnekleri](#) okuyun ve anlayın.

İlgili kavramlar

[MQ Telemetry](#)

İlgili başvurular

[MQXR özellikleri](#)

Linux ve AIX üzerinde telemetri için bir kuyruk yöneticisinin yapılandırılması

MQ Telemetry ' yi el ile yapılandırmak için aşağıdaki adımları izleyin. Yalnızca konuk kullanıcı kimliğini kullanan basit bir yapılandırmaya gereksinim duyarsanız, bunun yerine IBM MQ Exploreriçinde MQ Telemetry destek sihirbazını çalıştırabilirsiniz.

Başlamadan önce

Yalnızca basit bir yapılandırmaya gereksinim duyarsanız, IBM MQ Exploreriçinde MQ Telemetry desteğini kullanmayı düşünün. Bu destek bir sihirbaz ve örnek bir komut yordamı içerir `sampleMQM`. Bu kaynaklar, konuk kullanıcı kimliğini kullanarak bir ilk yapılandırma ayarlar. Bkz. [IBM MQ Explorer komutunu kullanarak MQ Telemetry kuruluşunun doğrulanması](#) ve [IBM MQ Telemetry Transport örnek programları](#).

Farklı bir kimlik doğrulama yöntemi kullanan daha karmaşık bir yapılandırmaya gereksinim duyarsanız, bu görevdeki adımları kullanın. Aşağıdaki ilk adımlarla başlayın:

1. IBM MQve MQ Telemetry özelliğinin nasıl kurulacağına ilişkin bilgi için bkz. [MQ Telemetry için kuruluşla ilgili önemli noktalar](#) .
2. Bir kuyruk yöneticisi yaratın ve başlatın. Bu görevde kuyruk yöneticisine `qMgr` adı verilir.
3. Bu görevin bir parçası olarak telemetri (MQXR) hizmetini yapılandırırınız. MQXR özellik ayarları, platforma özgü özellikler dosyasında saklanır: `mqxr_win.properties`. MQXR özellikleri dosyasını doğrudan düzenlemeniz gerekmez; hemen hemen tüm ayarlar MQSC denetim komutları ya da IBM MQ Explorerile yapılandırılabilir. Dosyayı doğrudan düzenlemeye karar verirseniz, değişiklikleri yapmadan önce kuyruk yöneticisini durdurun. Bkz. [MQXR özellikleri](#).

Bu görev hakkında

Farklı yetki şemaları kullanarak MQ Telemetry ' i el ile yapılandırmak için bu görevdeki adımları izleyin.

Yordam

1. Telemetri örnekleri dizininde bir komut penceresi açın.

Telemetri örnekleri dizini: `/opt/mqm/mqxr/samples`.

2. Telemetri iletim kuyruğunu oluşturun.

SYSTEM.MQTT.TRANSMIT.QUEUE yoksa, telemetri (MQXR) hizmeti ilk başlatıldığında otomatik olarak yaratılır ve konuk kullanıcı kimliğini kullanacak şekilde ayarlanır. Ancak bu görev, MQ Telemetry ' yi farklı bir yetki şeması kullanacak şekilde yapılandırır. Bu görev için telemetri (MQXR) hizmetini başlatmadan önce SYSTEM.MQTT.TRANSMIT.QUEUEögesini yaratır ve bu göreve erişimi yapılandırırınız.

Aşağıdaki komutu çalıştırın:

```
echo "DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000)" | runmqsc qMgr
```

3. Varsayılan iletim kuyruğunu belirler.

Varsayılan iletim kuyruğu SYSTEM.MQTT.TRANSMIT.QUEUE ise, iletilerin doğrudan MQTT istemcilerine gönderilmesi daha kolaydır. Ters durumda, IBM MQ iletilerini alan her istemci için bir uzak kuyruk tanımlaması eklemeniz gerekir; bkz. [“Bir istemciye doğrudan ileti gönderilmesi” sayfa 352](#). Varsayılan iletim kuyruğunun değiştirilmesi, varolan konfigürasyonunuzu etkileyebilir.

Telemetry (MQXR) hizmeti ilk kez başlatıldığında, SYSTEM.MQTT.TRANSMIT.QUEUE kuyruk yöneticisi için varsayılan iletim kuyruğu olarak ayarlanmaz. Bu ayarın konfigürasyonunu tanımlamak için varsayılan iletim kuyruğu özelliğini değiştirmeniz gerekir. Bunu IBM MQ Explorerkomutunu kullanarak ya da aşağıdaki komutu çalıştırarak yaparsınız:

```
echo "ALTER QMGR DEFEXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') " | runmqsc qMgr
```

4. Bir ya da daha çok kullanıcı kimliği yaratmak için [“MQTT istemcilerinin IBM MQ nesnelere erişmesi için yetki verilmesi” sayfa 354](#) içindeki bir yordamı izleyin. Kullanıcı kimlikleri, yayınları MQTT istemcilerine yayınlama, abone olma ve gönderme yetkisine sahip olur.
5. Telemetry (MQXR) hizmetini kurun.

```
cat /opt/<install_dir>/mqxr/samples/installMQXRService_unix.mqsc | runmqsc queue_manager
```

[“SYSTEM.MQXR.SERVICE oluşturma” sayfa 348](#) içindeki örnek koda da bakın.

6. Hizmeti başlatın.

```
echo "START SERVICE(SYSTEM.MQXR.SERVICE) " | runmqsc qMgr
```

Kuyruk yöneticisi başlatıldığında telemetry (MQXR) hizmeti otomatik olarak başlatılır. Kuyruk yöneticisi zaten çalıştığı için, bu görevde el ile başlatılır.

7. IBM MQ Explorer kullanarak MQTT istemcilerinden gelen bağlantıları kabul etmek için telemetry kanallarını yapılandırın.

Telemetry kanalları, kimlikleri [“4” sayfa 348](#). adımda tanımlanan kullanıcı kimliklerinden biri olacak şekilde yapılandırılmalıdır.

Ayrıca bkz. DEFINE CHANNEL (MQTT).

8. Örnek istemciyi çalıştırarak yapılandırmayı doğrulayın.

Örnek istemcinin telemetry kanalınızla çalışması için, kanalın istemciye yayınları yayınlama, abone olma ve yayınları alma yetkisi vermesi gerekir. Örnek istemci, varsayılan olarak 1883 kapısındaki telemetry kanalına bağlanır. Ayrıca bkz. [IBM MQ Telemetry Transport örnek programları](#).

SYSTEM.MQXR.SERVICE oluşturma

SYSTEM.MQXR.SERVICE oluşturmak için **runMQXRService** komutunu kullanın.

V 9.3.0

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+/mqxr/bin/runMQXRService.sh') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+" -g "+MQ_DATA_PATH+" -sf "[DEFAULT]"') +
STOPCMD('+MQ_INSTALL_PATH+/mqxr/bin/endMQXRService.sh') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+/mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+/mqxr.stderr')
```

Not: **V 9.3.0** IBM MQ 9.3.0 içinden **-sf** işareti, TLS kanal şifrelerini şifrelemek için kullanılacak anahtar içeren kimlik bilgileri anahtar dosyası adını sağlar. Daha fazla bilgi için bkz. [MQTT TLS kanalları için geçiş tümceleri şifrelemesi](#).

Windows üzerinde telemetry için bir kuyruk yöneticisinin yapılandırılması

MQ Telemetry 'yi el ile yapılandırmak için aşağıdaki adımları izleyin. Yalnızca konuk kullanıcı kimliğini kullanan basit bir yapılandırmaya gereksinim duyarsanız, bunun yerine IBM MQ Explorer içinde MQ Telemetry destek sihirbazını çalıştırabilirsiniz.

Başlamadan önce

Yalnızca basit bir yapılandırmaya gereksinim duyarsanız, IBM MQ Explorer'inde MQ Telemetry desteğini kullanmayı düşünün. Bu destek bir sihirbaz ve örnek bir komut yordamı içerir `sampleMQM`. Bu kaynaklar, konuk kullanıcı kimliğini kullanarak bir ilk yapılandırma ayarlar. Bkz. [IBM MQ Explorer komutunu kullanarak MQ Telemetry kuruluşunun doğrulanması](#) ve [IBM MQ Telemetry Transport örnek programları](#).

Farklı bir kimlik doğrulama yöntemi kullanan daha karmaşık bir yapılandırmaya gereksinim duyarsanız, bu görevdeki adımları kullanın. Aşağıdaki ilk adımlarla başlayın:

1. IBM MQ ve MQ Telemetry özelliğinin nasıl kurulacağına ilişkin bilgi için bkz. [MQ Telemetry için kuruluşla ilgili önemli noktalar](#).
2. Bir kuyruk yöneticisi yaratın ve başlatın. Bu görevde kuyruk yöneticisine `qMgr` adı verilir.
3. Bu görevin bir parçası olarak telemetri (MQXR) hizmetini yapılandırabilirsiniz. MQXR özellik ayarları, platforma özgü özellikler dosyasında saklanır: `mqxr_win.properties`. MQXR özellikleri dosyasını doğrudan düzenlemeniz gerekmez; hemen hemen tüm ayarlar MQSC denetim komutları ya da IBM MQ Explorer ile yapılandırılabilir. Dosyayı doğrudan düzenlemeye karar verirseniz, değişiklikleri yapmadan önce kuyruk yöneticisini durdurun. Bkz. [MQXR özellikleri](#).

Bu görev hakkında

Farklı yetki şemaları kullanarak MQ Telemetry 'i el ile yapılandırmak için bu görevdeki adımları izleyin.

Yordam

1. Telemetri örnekleri dizininde bir komut penceresi açın.

Telemetri örnekleri dizini: `WMQ program installation directory\mqxr\samples`.

2. Telemetri iletim kuyruğunu oluşturun.

SYSTEM.MQTT.TRANSMIT.QUEUE yoksa, telemetri (MQXR) hizmeti ilk başlatıldığında otomatik olarak yaratılır ve konuk kullanıcı kimliğini kullanacak şekilde ayarlanır. Ancak bu görev, MQ Telemetry 'yi farklı bir yetki şeması kullanacak şekilde yapılandırır. Bu görev için telemetri (MQXR) hizmetini başlatmadan önce SYSTEM.MQTT.TRANSMIT.QUEUE ögesini yaratır ve bu göreve erişimi yapılandırabilirsiniz.

Aşağıdaki komutu çalıştırın:

```
echo DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000) | runmqsc qMgr
```

3. Varsayılan iletim kuyruğunu belirler.

Varsayılan iletim kuyruğu SYSTEM.MQTT.TRANSMIT.QUEUE ise, iletilerin doğrudan MQTT istemcilerine gönderilmesi daha kolaydır. Ters durumda, IBM MQ iletilerini alan her istemci için bir uzak kuyruk tanımlaması eklemeniz gerekir; bkz. [“Bir istemciye doğrudan ileti gönderilmesi” sayfa 352](#). Varsayılan iletim kuyruğunun değiştirilmesi, varolan konfigürasyonunuzu etkileyebilir.

Telemetry (MQXR) hizmeti ilk kez başlatıldığında, SYSTEM.MQTT.TRANSMIT.QUEUE kuyruk yöneticisi için varsayılan iletim kuyruğu olarak ayarlanmaz. Bu ayarın konfigürasyonunu tanımlamak için varsayılan iletim kuyruğu özelliğini değiştirmeniz gerekir. Bunu IBM MQ Explorer komutunu kullanarak ya da aşağıdaki komutu çalıştırarak yaparsınız:

```
echo ALTER QMGR DEFQMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```

4. Bir ya da daha çok kullanıcı kimliği yaratmak için [“MQTT istemcilerinin IBM MQ nesnelere erişmesi için yetki verilmesi” sayfa 354](#) içindeki bir yordamı izleyin. Kullanıcı kimlikleri, yayınları MQTT istemcilerine yayınlama, abone olma ve gönderme yetkisine sahip olur.
5. Telemetry (MQXR) hizmetini kurun.

```
type installMQXRService_win.mqsc | runmqsc qMgr
```

“Oluşturma SYSTEM.MQXR.SERVICE” sayfa 350 içindeki örnek koda da bakın.

6. Hizmeti başlatın.

```
echo START SERVICE(SYSTEM.MQXR.SERVICE) | runmqsc qMgr
```

Kuyruk yöneticisi başlatıldığında telemetry (MQXR) hizmeti otomatik olarak başlatılır. Kuyruk yöneticisi zaten çalıştığı için, bu görevde el ile başlatılır.

7. IBM MQ Explorer kullanarak MQTT istemcilerinden gelen bağlantıları kabul etmek için telemetry kanallarını yapılandırın.

Telemetry kanalları, kimlikleri “4” sayfa 349. adımda tanımlanan kullanıcı kimliklerinden biri olacak şekilde yapılandırılmalıdır.

Ayrıca bkz. [DEFINE CHANNEL \(MQTT\)](#).

8. Örnek istemciyi çalıştırarak yapılandırmayı doğrulayın.

Örnek istemcinin telemetry kanalınızla çalışması için, kanalın istemciye yayınları yayınlama, abone olma ve yayınları alma yetkisi vermesi gerekir. Örnek istemci, varsayılan olarak 1883 kapısındaki telemetry kanalına bağlanır. Ayrıca bkz. [IBM MQ Telemetry Transport örnek programları](#).

Oluşturma SYSTEM.MQXR.SERVICE

SYSTEM.MQXR.SERVICE oluşturmak için **runMQXRService** komutunu kullanın.

V9.3.0

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +  
CONTROL(QMGR) +  
DESCR('Manages clients using MQXR protocols such as MQTT') +  
SERVTYPE(SERVER) +  
STARTCMD('+MQ_INSTALL_PATH+mqxr\bin\runMQXRService.bat') +  
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\" -g "+MQ_DATA_PATH+\" -sf "[DEFAULT]"') +  
STOPCMD('+MQ_INSTALL_PATH+mqxr\bin\endMQXRService.bat') +  
STOPARG('-m +QMNAME+') +  
STDOUT('+MQ_Q_MGR_DATA_PATH+mqxr.stdout') +  
STDERR('+MQ_Q_MGR_DATA_PATH+mqxr.stderr')
```

Not: V9.3.0 IBM MQ 9.3.0 içinden **-sf** işareti, TLS kanal şifrelerini şifrelemek için kullanılacak anahtarları içeren kimlik bilgileri anahtar dosyası adını sağlar. Daha fazla bilgi için bkz. [MQTT TLS kanalları için geçiş tümceleri şifrelemesi](#).

Linux

Windows

AIX

Dağıtılmış kuyruğa alma özelliğinin MQTT istemcilerine ileti gönderecek şekilde yapılandırılması

IBM MQ uygulamaları, bir istemci tarafından oluşturulan aboneliğe yayınlayarak ya da doğrudan ileti göndererek MQTT v3 istemcisi iletileri gönderebilir. Hangi yöntem kullanılırsa kullanılsın, ileti SYSTEM.MQTT.TRANSMIT.QUEUE üzerine yerleştirilir ve telemetry (MQXR) hizmeti tarafından istemciye gönderilir. SYSTEM.MQTT.TRANSMIT.QUEUE' e ileti yerleştirmenin çeşitli yolları vardır.

MQTT istemci aboneliğine yanıt olarak ileti yayınlanması

Telemetry (MQXR) hizmeti, MQTT istemcisi adına bir abonelik oluşturur. İstemci, istemci tarafından gönderilen abonelikte eşleşen tüm yayınların hedefidir. Telemetry hizmetleri, eşleşen yayınları müşteriye geri iletir.

Bir MQTT istemcisi, kuyruk yöneticisi adı ClientIdentifier olarak ayarlanmış bir kuyruk yöneticisi olarak IBM MQ' e bağlanır. İstemciye gönderilecek yayınların hedefi bir iletim kuyruğudur (SYSTEM.MQTT.TRANSMIT.QUEUE). Telemetry hizmeti, hedef kuyruk yöneticisi adını belirli bir istemcinin anahtarı olarak kullanarak SYSTEM.MQTT.TRANSMIT.QUEUE üzerindeki iletileri MQTT istemcilerine iletir.

Telemetry (MQXR) hizmeti, kuyruk yöneticisi adı olarak *ClientIdentifier* kullanılarak iletim kuyruğunu açar. Telemetry (MQXR) hizmeti, istemci aboneliğiyle eşleşen yayınları iletmek için kuyruğun nesne tanıtıcısını MQSUB çağırısına iletir. Nesne adı çözümünde, *ClientIdentifier* uzak kuyruk yöneticisi adı olarak yaratılır ve iletim kuyruğunun SYSTEM.MQTT.TRANSMIT.QUEUE olarak çözülmesi gerekir. Standart IBM MQ nesne adı çözümlemesi kullanılarak *ClientIdentifier* aşağıdaki gibi çözümlenir; bkz. Çizelge 16 sayfa 351.

1. *ClientIdentifier* hiçbir şey ile eşleşmez.

ClientIdentifier bir uzak kuyruk yöneticisi adıdır. Yerel kuyruk yöneticisi adı, kuyruk yöneticisi diğer adı ya da iletim kuyruğu adıyla eşleşmiyor.

Kuyruk adı tanımlanmadı. Şu anda telemetry (MQXR) hizmeti, kuyruğun adı olarak SYSTEM.MQTT.PUBLICATION.QUEUE değerini ayarlar. MQTT v3 istemcisi kuyrukları desteklemediği için, çözülen kuyruk adı istemci tarafından yoksayılır.

Yerel kuyruk yöneticisi özelliği Varsayılan iletim kuyruğu, yayının istemciye gönderilmesi için SYSTEM.MQTT.TRANSMIT.QUEUE üzerine konması için SYSTEM.MQTT.TRANSMIT.QUEUE olarak ayarlanmalıdır.

2. *ClientIdentifier*, *ClientIdentifier* adlı bir kuyruk yöneticisi diğer adıyla eşleşir.

ClientIdentifier bir uzak kuyruk yöneticisi adıdır. Bir kuyruk yöneticisi diğer adının adıyla eşleşir.

Kuyruk yöneticisi diğer adı, uzak kuyruk yöneticisi adı olarak *ClientIdentifier* ile tanımlanmalıdır.

Kuyruk yöneticisi diğer adı tanımında iletim kuyruğu adı ayarlandığında, varsayılan iletimin SYSTEM.MQTT.TRANSMIT.QUEUE olarak ayarlanması gerekmez.

Çizelge 16. MQTT kuyruk yöneticisi diğer adının çözülmesi

<i>ClientIdentifier</i>	Giriş		Çıktı		
	Kuyruk yöneticisi adı	Kuyruk adı	Kuyruk yöneticisi adı	Kuyruk adı	İletim kuyruğu
Hiçbir şey ile eşleşmez	<i>ClientIdentifier</i>	<i>tanımsız</i>	<i>ClientIdentifier</i>	<i>tanımsız</i>	Varsayılan iletim kuyruğu. SYSTEM.MQTT.TRANSMIT.QUEUE
<i>ClientIdentifier</i> adlı bir kuyruk yöneticisi diğer adıyla eşleşir	<i>ClientIdentifier</i>	<i>tanımsız</i>	<i>ClientIdentifier</i>	<i>tanımsız</i>	SYSTEM.MQTT.TRANSMIT.QUEUE

Ad çözümlemesi hakkında daha fazla bilgi için bkz. [Ad çözümlemesi](#).

Herhangi bir IBM MQ programı aynı konuda yayınlayabilir. Yayın, konu aboneliği olan MQTT v3 istemcileri de içinde olmak üzere abonelerine gönderilir.

CLUSTER(*clusterName*) özniteliğiyle bir kümede yönetim konusu yaratıldıysa, kümedeki herhangi bir uygulama istemciye yayınlayabilir; örneğin:

```
echo DEFINE TOPIC('MQTTExamples') TOPICSTR('MQTT Examples') CLUSTER(MQTT) REPLACE | runmqsc qMgr
```

Not: SYSTEM.MQTT.TRANSMIT.QUEUE ' e bir küme özniteliği vermeyin.

MQTT istemci aboneleri ve yayıncıları farklı kuyruk yöneticilerine bağlanabilir. Aboneler ve yayıncılar aynı kümenin parçası olabilir ya da bir yayınlama/abone olma sıradüzeniyle bağlanabilir. Yayın, IBM MQ kullanılarak yayıncıdan aboneye teslim edilir.

Bir istemciye doğrudan ileti gönderilmesi

Abonelik oluşturan ve abonelik konusuyla eşleşen bir yayın alan bir istemciye alternatif olarak, MQTT v3 istemcisine doğrudan bir ileti gönderin. MQTT V3 istemci uygulamaları doğrudan ileti gönderemez, ancak IBM MQ uygulamaları gibi diğer uygulamalar gönderebilir.

IBM MQ uygulaması, MQTT v3 istemcisinin `ClientIdentifier` değerini bilmelidir. MQTT v3 istemcilerinin kuyrukları olmadığı için, hedef kuyruk adı MQTT v3 uygulama istemcisi `messageArrived` yöntemine konu adı olarak geçirilir. Örneğin, bir MQI programında, istemciyle `ObjectQmgrAdı` olarak bir nesne tanımlayıcısı yaratın:

```
MQOD.ObjectQmgrName = ClientIdentifier ;
MQOD.ObjectName = name ;
```

Uygulama JMSkullanılarak yazıldıysa, bir noktadan noktaya iletişim hedefi oluşturun; örneğin:

```
V9.3.0 V9.3.0 JM 3.0
jakarta.jms.Destination jmsDestination =
(jakarta.jms.Destination)jmsFactory.createQueue
("queue://ClientIdentifier/name");
```

```
JMS 2.0
javax.jms.Destination jmsDestination =
(javax.jms.Destination)jmsFactory.createQueue
("queue://ClientIdentifier/name");
```

MQTT istemcisine istenmeyen bir ileti göndermek için uzak kuyruk tanımı kullanın. Uzak kuyruk yöneticisi adı, istemcinin `ClientIdentifier` olarak çözülmelidir. İletim kuyruğunun şu şekilde çözülmesi gerekir: `SYSTEM.MQTT.TRANSMIT.QUEUE`; bkz. [Çizelge 17 sayfa 352](#). Uzak kuyruk adı herhangi bir şey olabilir. İstemci bunu konu dizgisi olarak alır.

Çizelge 17. MQTT istemcisi uzak kuyruk tanımlamasının ad çözümlemesi

Giriş		Çıktı		
Kuyruk adı	Kuyruk yöneticisi adı	Kuyruk adı	Kuyruk yöneticisi adı	İletim kuyruğu
Uzak kuyruk tanımlamasının adı	Boş ya da yerel kuyruk yöneticisi adı	Konu dizgisi olarak kullanılan uzak kuyruk adı	<code>ClientIdentifier</code>	<code>SYSTEM.MQTT.TRANSMIT.QUEUE</code>

İstemci bağlıysa, ileti doğrudan `messageArrived` yöntemini çağıran MQTT istemcisine gönderilir; bkz. [messageArrived yöntemi](#).

İstemci kalıcı bir oturumla bağlantısını kestiye, ileti `SYSTEM.MQTT.TRANSMIT.QUEUE`; Bkz. [MQTT durumsuz ve durumlu oturumlar](#). İstemci oturuma yeniden bağlandığında istemciye iletilir.

Kalıcı olmayan bir ileti gönderirseniz, istemciye en fazla hizmet kalitesi `QoS=0` ile gönderilir. Kalıcı bir iletiyi doğrudan bir istemciye gönderirseniz, varsayılan olarak ileti tam olarak bir kez hizmet kalitesi `QoS=2` ile gönderilir. İstemcinin kalıcı saklama mekanizması olmayabileceğinden, istemci doğrudan gönderilen iletiler için kabul ettiği hizmet kalitesini azaltabilir. Doğrudan bir istemciye gönderilen iletilere ilişkin hizmet kalitesini azaltmak için `DEFAULT.QoS` konusuna abonelik yapın. İstemcinin destekleyebileceği hizmet kalitesi üst sınırını belirtin.

Linux Windows AIX MQTT müşteri kimliği, yetkilendirme ve kimlik doğrulaması

Telemetry (MQXR) hizmeti, MQTT kanallarını kullanarak MQTT istemcileri adına IBM MQ konularını yayınlar ya da bu konulara abone olur. IBM MQ yöneticisi, IBM MQ yetkilendirmesi için kullanılan MQTT

kanal kimliğini yapılandırır. Yönetici, kanal için ortak bir kimlik tanımlayabilir ya da kanala bağlı bir istemcinin `Kullanıcı Adı` ya da `ClientIdentifier` değerini kullanabilir.

Telemetry (MQXR) hizmeti, istemci tarafından sağlanan `Kullanıcı Adı` 'nı kullanarak ya da bir istemci sertifikası kullanarak istemcinin kimliğini doğrulayabilir. `Kullanıcı adı` kimliği, istemci tarafından sağlanan bir parola kullanılarak doğrulanır.

Özetlemek gerekirse: Müşteri kimliği, müşteri kimliği seçimidir. Bağlama bağlı olarak istemci, `ClientIdentifier`, `Username`, yönetici tarafından oluşturulan ortak bir istemci kimliği ya da bir istemci sertifikası ile tanımlanır. Kimlik denetimi için kullanılan istemci tanıtıcısının, yetki için kullanılan tanıtıcıyla aynı olması gerekmez.

MQTT istemci programları, MQTT kanalı kullanılarak sunucuya gönderilen `Kullanıcı Adı` ve `Parola` değerlerini ayarlar. Bağlantıyı şifrelemek ve doğrulamak için gereken TLS özelliklerini de ayarlayabilirler. Yönetici, MQTT kanalının kimliğini doğrulayıp doğrulamayacağına ve kanalın kimliğini nasıl doğrulayacağına karar verir.

Bir MQTT istemcisine IBM MQ nesnelere erişim yetkisi vermek, istemcinin `ClientIdentifier` ya da `Username` kimliğine yetki vermek ya da ortak bir istemci kimliğine yetki vermek için. Bir istemcinin IBM MQ'e bağlanmasına izin vermek için `Kullanıcı Adı` 'nı doğrulayın ya da bir istemci sertifikası kullanın. JAAS 'ı `Kullanıcı Adı` 'nı doğrulamak için yapılandırın ve TLS 'yi bir istemci sertifikasının kimliğini doğrulamak üzere yapılandırın.

İstemcide bir `Parola` belirlerseniz, VPN 'yi kullanarak bağlantıyı şifreleyin ya da MQTT kanalını, parolayı özel tutmak için TLS kullanacak şekilde yapılandırın.

İstemci sertifikalarını yönetmek zordur. Bu nedenle, parola kimlik doğrulamasıyla ilişkili riskler kabul edilebilir ise, genellikle istemcilerin kimliğini doğrulamak için parola kimlik doğrulaması kullanılır.

İstemci sertifikasını yönetmenin ve saklamanın güvenli bir yolu varsa, sertifika kimlik doğrulamasına güvenebilirsiniz. Bununla birlikte, sertifikaların telemetrinin kullanıldığı ortam tiplerinde güvenli bir şekilde yönetilebilmesi nadiren söz konusu olur. Bunun yerine, istemci sertifikalarını kullanan aygıtların kimlik doğrulaması, sunucuda istemci parolalarının doğrulanmasıyla tamamlanır. Ek karmaşıklık nedeniyle, istemci sertifikalarının kullanımı son derece hassas uygulamalarla sınırlıdır. İki tür kimlik doğrulama kullanımı, iki faktörlü kimlik doğrulama olarak adlandırılır. Parola gibi etmenlerden birini bilmeniz ve sertifika gibi diğer etmenlere sahip olmanız gerekir.

Çip ve iğne aygıtı gibi oldukça hassas bir uygulamada, cihaz, dahili donanım ve yazılıma müdahale edilmesini önlemek için üretim sırasında kilitletir. Güvenilir, zaman sınırlı, istemci sertifikası aygıtta kopyalanır. Aygıt, kullanılacağı yere konuşlandırılır. Aygıt her kullanıldığında, bir parola ya da akıllı karttan alınan başka bir sertifika kullanılarak daha fazla kimlik doğrulaması gerçekleştirilir.

Linux Windows AIX MQTT istemci kimliği ve yetkilendirmesi

IBM MQ nesnelere erişim yetkisi için istemci kimliğini (`Kullanıcı adı`) ya da ortak istemci kimliğini kullanın.

IBM MQ yöneticisi, MQTT kanalının kimliğini seçmek için üç seçeneğine sahiptir. Yönetici, istemci tarafından kullanılan MQTT kanalını tanımlarken ya da değiştirirken seçim yapar. Kimlik, IBM MQ konularına erişimi yetkilendirmek için kullanılır. Seçim aşağıdaki sırayla yapılır:

1. İstemci tanıtıcısı (bkz. [USECLNTID](#)).
2. Yöneticinin kanal için sağladığı bir kimlik (kanalın `MCAUSER`). Bkz. [MCAUSER](#)).
3. Önceki seçeneklerden hiçbiri geçerli değilse, MQTT istemcisinden geçirilen `Kullanıcı adı` (`Kullanıcı adı`), `MqttConnectOptions` sınıfının bir özneliğidir. İstemci hizmete bağlanmadan önce ayarlanmalıdır. Varsayılan değeri boş değerdir.

Beladan kaçının: Bu işlem tarafından seçilen kimliğe, örneğin, `DISPLAY CHSTATUS (MQTT)` komutu tarafından istemcinin `MCAUSER` değeri olarak başvurulur. Bunun, seçimde atıfta bulunulan kanalın `MCAUSER` 'i ile aynı kimlik olması gerekmediğini unutmayın (2).

Hangi nesnelerin ve hangi işlemlerin MQTT kanalıyla ilişkili kimlik tarafından kullanılma yetkisine sahip olduğunu seçmek için IBM MQ **setmqaut** komutunu kullanın. Örneğin, aşağıdaki kod, QM1kuyruk yöneticisi tarafından sağlanan MQTTClientkanal kimliğini yetkilendirir:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

Linux

Windows

AIX

MQTT istemcilerinin IBM MQ nesnelere erişmesi

İçin yetki verilmesi

MQTT istemcilerine IBM MQ nesnelere yayınlama ve bunlara abone olma yetkisi vermek için bu adımları izleyin. Adımlar, dört alternatif erişim denetimi örüntülerini izler.

Başlamadan önce

MQTT istemcileri, bir telemetri kanalına bağlandıklarında bir kimliğe atanarak IBM MQ içindeki nesnelere erişme yetkisine sahiptir. IBM MQ Yöneticisi, IBM MQ Explorer 'ı kullanarak telemetri kanalını yapılandırır ve bir istemciye şu üç kimlik tipinden birini verir:

1. ClientIdentifier
2. Kullanıcı adı
3. Yöneticinin kanala atadığı ad.

Hangi tip kullanılırsa kullanılsın, kimlik, kurulu yetkilendirme hizmeti tarafından birincil kullanıcı olarak IBM MQ olarak tanımlanmalıdır. Windows ya da Linux üzerindeki varsayılan yetkilendirme hizmeti, Nesne Yetki Yöneticisi (OAM) olarak adlandırılır. OAM kullanıyorsanız, kimlik bir kullanıcı kimliği olarak tanımlanmalıdır.

Bir istemciye ya da istemci derlemine IBM MQ içinde tanımlanan konuları yayınlama ya da bunlara abone olma izni vermek için kimliği kullanın. Bir MQTT istemcisi bir konuya abone olursa, sonuçta elde edilen yayınları alması için bu kimliği kullanın.

Her biri ayrı ayrı erişim izni gerektiren on binlerce MQTT istemcisi olan bir sistemi yönetmek zordur. Bir çözüm, ortak kimlikleri tanımlamak ve tek tek MQTT istemcilerini ortak kimliklerden biriyle ilişkilendirmektir. Farklı izin birleşimleri tanımlamak için gereken sayıda ortak kimlik tanımlayın. Başka bir çözüm, binlerce kullanıcıyla işletim sisteminden daha kolay başa çıkabilen kendi yetkilendirme hizmetinizi yazmaktır.

OAM kullanarak MQTT istemcilerini iki şekilde ortak kimliklerde birleştirebilirsiniz:

1. Her biri, yöneticinin IBM MQ Explorer kullanarak ayırdığı farklı bir kullanıcı kimliğine sahip birden çok telemetri kanalı tanımlayın. Farklı TCP/IP kapı numaralarını kullanan istemciler farklı telemetri kanallarıyla ilişkilendirilir ve farklı kimliklere atanır.
2. Tek bir telemetri kanalı tanımlayın, ancak her istemcinin küçük bir kullanıcı kimliği kümesinden Kullanıcı adı seçmesini sağlayın. Yönetici, Kullanıcı Adı istemcisini kimliği olarak seçecek şekilde telemetri kanalını yapılandırır.

Bu görevde, nasıl ayarlandığına bakılmaksızın, telemetri kanalının kimliği *mqttUser* olarak adlandırılır. İstemci derlemleri farklı kimlikler kullanıyorsa, her istemci derlemi için bir tane olmak üzere birden çok *mqttUsers* kullanın. Görev OAM kullandığından, her *mqttUser* bir kullanıcı kimliği olmalıdır.

Bu görev hakkında

Bu görevde, belirli gereksinimlere göre uyarlayabileceğiniz dört erişim denetimi kalıbı seçeneğiniz vardır. Kalıplar, erişim denetiminin ayrıntı düzeylerinde farklılık gösterir.

- [“Erişim denetimi yok” sayfa 355](#)
- [“Kaba taneli erişim denetimi” sayfa 355](#)
- [“Orta taneli erişim denetimi” sayfa 355](#)
- [“Ayrıntılı erişim denetimi” sayfa 355](#)

Modellerin sonucu, IBM MQ'yi yayınlamak ve bunlara abone olmak için *mqttUsers* izin kümeleri atamak ve IBM MQ' den yayınlar almaktır.

Erişim denetimi yok

MQTT istemcilerine IBM MQ denetim yetkisi verilir ve herhangi bir nesne üzerinde herhangi bir işlem gerçekleştirilebilir.

Yordam

1. Tüm MQTT istemcilerinin kimliği olarak işlev görecektir bir *mqttUser* kullanıcı kimliği oluşturun.
2. *mqm* grubuna *mqttUser* ekleyin; bkz. [Windows üzerinde bir gruba kullanıcı eklenmesi](#) ya da [Linux üzerinde grup oluşturma ve yönetme](#)

Kaba taneli erişim denetimi

MQTT istemcilerinin yayınlama ve abone olma ve MQTT istemcilerine ileti gönderme yetkileri vardır. Diğer işlemleri gerçekleştirme ya da diğer nesnelere erişme yetkileri yoktur.

Yordam

1. Tüm MQTT istemcilerinin kimliği olarak işlev görecektir bir *mqttUser* kullanıcı kimliği oluşturun.
2. *mqttUser* ' e tüm konuları yayınlama ve bunlara abone olma ve yayınları MQTT istemcilerine gönderme yetkisi verin.

```
setmqaut -m qMgr -t topic -n SYSTEM.BASE.TOPIC -p mqttUser -all +pub +sub  
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqttUser -all +put
```

Orta taneli erişim denetimi

MQTT istemcileri, farklı konu kümelerini yayınlamak ve bunlara abone olmak ve MQTT istemcilerine ileti göndermek için farklı gruplara ayrılır.

Yordam

1. Yayınlama/abone olma konu ağacında birden çok kullanıcı kimliği, *mqttUsers* ve birden çok yönetim konusu oluşturun.
2. Farklı *mqttUsers* konularına yetki verin.

```
setmqaut -m qMgr -t topic -n topic1 -p mqttUserA -all +pub +sub  
setmqaut -m qMgr -t topic -n topic2 -p mqttUserB -all +pub +sub
```

3. Bir *mqtt* grubu oluşturun ve tümünü *mqttUsers* gruba ekleyin.
4. *mqtt* ' e MQTT istemcilerine konu göndermesi için yetki verin.

```
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

Ayrıntılı erişim denetimi

MQTT istemcileri, gruplara nesnelere üzerinde işlem gerçekleştirme yetkisi veren var olan bir erişim denetimi sistemiyle birleştirilir.

Bu görev hakkında

Kullanıcı kimliği, gerektirdiği yetkilere bağlı olarak bir ya da daha çok işletim sistemi grubuna atanır. IBM MQ uygulamaları yayınlıyor ve MQTT istemcileriyle aynı konu alanına abone yapıyorsa, bu modeli kullanın. Gruplara Publish X, Subscribe Y ve mqtt denir

Publish X

Publish X gruplarının üyeleri *topicX* ' e yayınlatabilir.

Subscribe Y

Subscribe Y gruplarının üyeleri *topicY* e abone olabilir.

mqtt

mqtt grubunun üyeleri, yayınları MQTT istemcilerine gönderebilir.

Yordam

1. Yayınlama/abone olma konu ağacında birden çok yönetim konusuna ayrılan birden çok grup Publish X ve Subscribe Y oluşturun.
2. Bir grup oluşturun *mqtt*.
3. Birden çok kullanıcı kimliği (*mqttUsers*) oluşturun ve kullanıcıları, ne yapma yetkisine sahip olduklarına bağlı olarak gruplardan herhangi birine ekleyin.
4. Farklı Publish X ve Subscribe X gruplarına farklı konular için yetki verin ve *mqtt* grubuna MQTT istemcilerine ileti gönderme yetkisi verin.

```
setmqaut -m qMgr -t topic -n topic1 -p Publish X -all +pub
setmqaut -m qMgr -t topic -n topic1 -p Subscribe X -all +pub +sub
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

Linux

Windows

AIX

Parola kullanarak MQTT istemci kimlik

doğrulaması

İstemci parolasını kullanarak Kullanıcı Adı 'nı doğrulayın. İstemciye yayınlama ve konulara abone olma yetkisi vermek için kullanılan kimliğe farklı bir kimlik kullanarak istemcinin kimliğini doğrulayabilirsiniz.

Telemetry (MQXR) hizmeti, Kullanıcı Adı istemcisinin kimliğini doğrulamak için JAAS kullanır. JAAS , MQTT istemcisi tarafından sağlanan Parola değerini kullanır.

IBM MQ yöneticisi, bir istemcinin bağlandığı MQTT kanalını yapılandırarak Kullanıcı Adı 'nı doğrulayıp doğrulamayacağına karar verir. Müşteriler farklı kanallara atanabilir ve her kanal, istemcilerinin kimliğini farklı şekillerde doğrulayabilecek şekilde yapılandırılabilir. JAAS' ı kullanarak, hangi yöntemlerin istemcinin kimliğini doğrulamasını ve isteğe bağlı olarak istemcinin kimliğini doğrulamasını gerektiğini yapılandırabilirsiniz.

Kimlik doğrulama için kimlik seçimi, yetkilendirme için kimlik seçimini etkilemez. Yönetim kolaylığı için yetki için ortak bir kimlik oluşturmak isteyebilirsiniz, ancak her kullanıcının bu kimliği kullanması için kimlik doğrulaması gerçekleştirin. Aşağıdaki yordamda, ortak bir kimlik kullanmak için tek tek kullanıcıların kimliğini doğrulama adımları özetmektedir:

1. IBM MQ yöneticisi, IBM MQ Explorer 'ı kullanarak MQTT kanal kimliğini MQTTClientUser gibi herhangi bir ada ayarlar.
2. IBM MQ yöneticisi, MQTTClient ' e herhangi bir konuyu yayınlama ve bu konuya abone olma yetkisi verir:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

3. MQTT istemci uygulaması geliştiricisi bir MqttConnectOptions nesnesi yaratır ve sunucuya bağlanmadan önce Kullanıcı adı ve Parola değerini ayarlar.
4. Güvenlik geliştiricisi, Parola ile Kullanıcı Adı 'nı doğrulamak için bir JAAS LoginModule yaratır ve bunu JAAS yapılandırma dosyasına dahil eder.
5. IBM MQ yöneticisi, JAAS kullanarak istemcinin Username kimliğini doğrulamak için MQTT kanalını yapılandırır.

doğrulaması

MQTT istemcisi ile kuyruk yöneticisi arasındaki bağlantılar her zaman MQTT istemcisi tarafından başlatılır. MQTT istemcisi her zaman SSL istemcisidir. MQTT istemcisinin sunucu ve sunucu kimlik doğrulaması için istemci kimlik doğrulaması isteğe bağlıdır.

İstemciye özel imzalı bir sayısal sertifika sağlayarak, MQTT istemcisinin kimliğini WebSphere MQ' da doğrulayabilir. WebSphere MQ Denetimcisi, MQTT istemcilerini TLS kullanarak kuyruk yöneticisinde kendilerinin kimliğini doğrulamaya zorlayabilir. Yalnızca karşılıklı kimlik doğrulamasının bir parçası olarak istemci kimlik doğrulaması isteyebilirsiniz.

SSL kullanımına alternatif olarak, IPsec gibi bazı Sanal Özel Ağ (VPN) türleri, bir TCP/IP bağlantısının uç noktalarını doğrular. VPN, ağ üzerinden akan her bir IP paketini şifreler. Böyle bir VPN bağlantısı kurulduktan sonra, güvenilir bir ağ oluşturmuş olunuz. VPN ağı üzerinden TCP/IP ' yi kullanarak MQTT istemcilerini telemetri kanallarına bağlayabilirsiniz.

TLS kullanan istemci kimlik doğrulaması, istemcinin bir güvenlik dizgisi olmasına dayanır. Güvenlik dizgisi, kendinden imzalı bir sertifika durumunda istemcinin özel anahtarı ya da sertifika yetkilisi tarafından sağlanan bir anahtardır. Anahtar, istemcinin sayısal sertifikasını imzalamak için kullanılır. İlgili ortak anahtara sahip olan herkes dijital sertifikayı doğrulayabilir. Sertifikalar güvenilir olabilir ya da zincirlenmişse, bir sertifika zinciri üzerinden güvenilir bir kök sertifikaya kadar izlenebilir. İstemci doğrulaması, istemci tarafından sağlanan sertifika zincirindeki tüm sertifikaları sunucuya gönderir. Sunucu, güvendiği bir sertifika buluncaya kadar sertifika zincirini denetler. Güvenilir sertifika, kendinden onaylı bir sertifikadan oluşturulan genel sertifika ya da genellikle bir sertifika yetkilisi tarafından verilen kök sertifika olabilir. Son, isteğe bağlı bir adım olarak, güvenilir sertifika "canlı" sertifika iptal listesiyle karşılaştırılabilir.

Güvenilen sertifika, bir sertifika yetkilisi tarafından verilmiş ve JRE sertifika deposuna önceden eklenmiş olabilir. Kendinden onaylı bir sertifika ya da telemetri kanalı anahtar deposuna güvenilen bir sertifika olarak eklenen herhangi bir sertifika olabilir.

Not: Telemetri kanalında, hem bir ya da daha çok telemetri kanalının özel anahtarlarını, hem de istemcilerin kimliğini doğrulamak için gereken genel sertifikaları içeren birleşik bir anahtar deposu/güvenli depo bulunur. Bir SSL kanalının bir anahtar deposu olması gerektiğinden ve bu dosya kanal güvenli deposuyla aynı olduğundan, JRE sertifika deposuna hiçbir zaman başvurulmaz. Bunun sonucu olarak, bir istemcinin kimlik doğrulaması için bir CA kök sertifikası gerekiyorsa, CA kök sertifikası JRE sertifika deposunda olsa bile, kök sertifikayı kanala ilişkin anahtar deposuna yerleştirmeniz gerekir. JRE sertifika deposuna hiçbir zaman başvurulmaz.

Müşteri kimlik doğrulamasının karşı koyacağı tehditleri ve istemci ve sunucunun tehditlere karşı mücadele etme konusunda oynadığı rolleri düşünün. İstemci sertifikasının tek başına doğrulanması, bir sisteme yetkisiz erişimi önlemek için yeterli değildir. Başka bir kullanıcı istemci aygıtını ele aldıysa, istemci aygıtının sertifika sahibinin yetkisiyle hareket etmemesi gerekir. İstenmeyen saldırılara karşı tek bir savunmaya asla güvenme. En azından iki faktörlü bir kimlik doğrulama yaklaşımı ve özel bilgi bilgisiyle bir sertifikaya sahip olma özelliğini kullanın. Örneğin, JAAS' ı kullanın ve sunucu tarafından verilen bir parolayı kullanarak istemcinin kimliğini doğrulayın.

İstemci sertifikası için birincil tehdit, yanlış ellere geçmesi. Sertifika, istemcideki parola korumalı bir anahtar deposunda tutulur. Anahtar deposuna nasıl yerleştirilir? MQTT istemcisi anahtar deposuna parolayı nasıl alır? Şifre koruması ne kadar güvenli? Telemetri cihazlarını çıkarmak genellikle kolaydır ve daha sonra özel olarak hacklenebilir. Cihaz donanımının kurcalamaya dayanıklı olması şart mı? İstemci tarafı sertifikalarının dağıtılması ve korunması zor olarak kabul edilir; buna anahtar yönetimi sorunu denir.

İkincil tehdit, aygıtın sunuculara istenmeyen yollarla erişmek için yanlış kullanılmasıdır. Örneğin, MQTT uygulaması kurcalandıysa, kimliği doğrulanmış istemci kimliğini kullanarak sunucu yapılandırmasında bir zayıflık kullanılması mümkün olabilir.

SSL kullanarak bir MQTT istemcisinin kimliğini doğrulamak için telemetri kanalını ve istemciyi yapılandırın.

İlgili kavramlar

[“TLS kullanarak MQTT istemci kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 358](#)

IBM MQ yöneticisi, sunucuda telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasında TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar dosyalarına parola korumalı erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcesi ya da anahtar dosyası olmadan tanımlanırsa, kanal TLS bağlantılarını kabul etmez.

[TLS kullanarak istemci kimlik doğrulaması için MQTT istemci yapılandırması](#)

TLS kullanarak MQTT istemci kimlik doğrulaması için telemetri kanalı yapılandırması

IBM MQ yöneticisi, sunucuda telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasında TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar dosyalarına parola korumalı erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcesi ya da anahtar dosyası olmadan tanımlanırsa, kanal TLS bağlantılarını kabul etmez.

Bir TLS telemetri kanalının `com.ibm.mq.MQTT.ClientAuth` özelliğini `REQUIRED` olarak ayarlayarak, bu kanalda bağlantı kuran tüm istemcilerin dijital sertifikaları doğruladıklarına dair kanıt sağlamalarını sağlayın. İstemci sertifikalarının kimliği, sertifika yetkililerinden gelen sertifikalar kullanılarak doğrulanır ve bu da güvenilir bir kök sertifikaya yol açar. İstemci sertifikası kendinden onaylı ya da bir sertifika yetkilisinden alınan bir sertifika tarafından imzalanmışsa, istemcinin ya da sertifika yetkilisinin genel olarak imzalı sertifikalarının sunucuda güvenli bir şekilde saklanması gerekir.

Genel olarak imzalanmış istemci sertifikasını ya da sertifika yetkilisinden alınan sertifikayı telemetri kanalı anahtar deposuna yerleştirin. Sunucuda, genel olarak imzalı sertifikalar ayrı bir güvenli depoda değil, özel olarak imzalanmış sertifikalarla aynı anahtar dosyasında saklanır.

Sunucu, sahip olduğu tüm genel sertifikaları ve şifreleme takımlarını kullanarak gönderildiği istemci sertifikalarının imzasını doğrular. Sunucu, anahtar zincirini doğrular. Kuyruk yöneticisi, sertifikayı sertifika iptal listesine karşı sınavabilecek şekilde yapılandırılabilir. Kuyruk yöneticisi iptal `namelist` özelliği: `SSLCRLNL`.

Bir istemcinin gönderdiği sertifikalardan herhangi biri, sunucu anahtar deposundaki bir sertifika tarafından doğrulanır ve istemcinin kimliği doğrulanır.

IBM MQ yöneticisi aynı telemetri kanalını, istemci Parola ile istemcinin `UserName` ya da `ClientIdentifier` değerini denetlemek için JAAS kullanacak şekilde yapılandırılabilir.

Birden çok telemetri kanalı için aynı anahtar deposunu kullanabilirsiniz.

Aygıttaki parola korumalı istemci anahtar deposundaki en az bir sayısal sertifikanın doğrulanması, istemcinin sunucuda kimliğini doğrular. Sayısal sertifika yalnızca IBM MQ tarafından kimlik doğrulaması için kullanılır. İstemcinin TCP/IP adresini doğrulamak ya da yetki ya da hesap için istemcinin kimliğini belirlemek için kullanılmaz. Sunucu tarafından benimsenen istemcinin kimliği, istemcinin `ClientIdentifier` ya da `IBM MQ yöneticisi tarafından yaratılan bir kimliktir`.

İstemci kimlik doğrulaması için TLS şifreleme takımlarını da kullanabilirsiniz. SHA-2 şifreleme takımlarını kullanmayı planlıyorsanız, bkz. [“SHA-2 şifreleme takımlarının MQTT kanallarıyla kullanılmasına ilişkin sistem gereksinimleri” sayfa 362](#).

İlgili kavramlar

[“TLS kullanarak kanal kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 359](#)

IBM MQ yöneticisi, sunucuda telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasında TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar dosyalarına parola korumalı erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcesi ya da anahtar dosyası olmadan tanımlanırsa, kanal TLS bağlantılarını kabul etmez.

[CipherSpecs ve CipherSuites](#)

İlgili başvurular

[KANAL TANIMLAYIN \(MQTT\)](#)

[DEĞİŞTİRME KANALI \(MQTT\)](#)

doğrulaması

MQTT istemcisi ile kuyruk yöneticisi arasındaki bağlantılar her zaman MQTT istemcisi tarafından başlatılır. MQTT istemcisi her zaman SSL istemcisidir. MQTT istemcisinin sunucu ve sunucu kimlik doğrulaması için istemci kimlik doğrulaması isteğe bağlıdır.

İstemci, anonim bağlantıyı destekleyen bir CipherSpec kullanacak şekilde yapılandırılmadıysa, istemci her zaman sunucunun kimliğini doğrulamayı dener. Kimlik doğrulaması başarısız olursa, bağlantı kurulmaz.

SSL kullanımına alternatif olarak, IPsec gibi bazı Sanal Özel Ağ (VPN) türleri, bir TCP/IP bağlantısının uç noktalarını doğrular. VPN, ağ üzerinden akan her bir IP paketini şifreler. Böyle bir VPN bağlantısı kurulduktan sonra, güvenilir bir ağ oluşturmuş olunuz. VPN ağı üzerinden TCP/IP 'yi kullanarak MQTT istemcilerini telemetri kanallarına bağlayabilirsiniz.

SSL kullanarak sunucu kimlik doğrulaması, gizli bilgileri göndermek üzere olduğunuz sunucunun kimliğini doğrular. İstemci, sunucudan gönderilen sertifikalarla, güvenli deposuna ya da JRE cacerts deposuna yerleştirilen sertifikalarla eşleşen denetimleri gerçekleştirir.

JRE sertifika deposu bir JKS dosyasıdır, cacerts. JRE InstallPath\lib\security\içinde bulunur. changeit varsayılan parolasıyla kurulur. Güvendiğiniz sertifikaları JRE sertifika deposunda ya da istemci güvenli deposunda saklayabilirsiniz. Her iki mağazaları da kullanamazsınız. İstemcinin güvendiği genel sertifikaları diğer Java uygulamalarının kullandığı sertifikalardan ayrı tutmak istiyorsanız istemci güvenli deposunu kullanın. İstemcide çalışan tüm Java uygulamaları için ortak bir sertifika deposu kullanmak istiyorsanız JRE sertifika deposunu kullanın. JRE sertifika deposunu kullanmaya karar verirseniz, içerdiği sertifikaları gözden geçirip bunlara güvendiğinizden emin olun.

Farklı bir güven sağlayıcısı sağlayarak JSSE yapılandırmasını değiştirebilirsiniz. Bir sertifikada farklı denetimler gerçekleştirmek için bir güven sağlayıcısını özelleştirebilirsiniz. MQTT istemcisini kullanan bazı OGSi ortamlarında, ortamı farklı bir güven sağlayıcısı sağlar.

TLS kullanarak telemetri kanalının kimliğini doğrulamak için sunucuyu ve istemciyi yapılandırın.

telemetri kanalı yapılandırması

IBM MQ yöneticisi, sunucuda telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasında TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar dosyalarına parola korumalı erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcesi ya da anahtar dosyası olmadan tanımlanırsa, kanal TLS bağlantılarını kabul etmez.

Özel anahtarıyla imzalanan sunucunun sayısal sertifikasını, telemetri kanalının sunucuda kullanacağı anahtar deposunda saklayın. Anahtar zincirini istemciye iletmek istiyorsanız, anahtar zincirindeki sertifikaları anahtar deposunda saklayın. TLS 'yi kullanmak için IBM MQ Explorer 'ı kullanarak telemetri kanalını yapılandırın. Anahtar deposu yolunu ve anahtar deposuna erişmek için parolayı belirtin. Kanalın TCP/IP kapı numarasını ayarlamazsanız, TLS telemetri kanalı kapı numarası varsayılan olarak 8883 olur.

Kanal kimlik doğrulaması için TLS şifre takımlarını da kullanabilirsiniz. SHA-2 şifreleme takımlarını kullanmayı planlıyorsanız, bkz. [“SHA-2 şifreleme takımlarının MQTT kanallarıyla kullanılmasına ilişkin sistem gereksinimleri” sayfa 362.](#)

İlgili kavramlar

[“TLS kullanarak MQTT istemci kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 358](#)

IBM MQ yöneticisi, sunucuda telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasında TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar dosyalarına parola korumalı erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcesi ya da anahtar dosyası olmadan tanımlanırsa, kanal TLS bağlantılarını kabul etmez.

[CipherSpecs ve CipherSuites](#)

İlgili başvurular

[KANAL TANIMLAYIN \(MQTT\)](#)

ALTER CHANNEL (MQTT)

TLS kimlik doğrulamasını kullanan örnek MQTT kanal yapılandırması

Bu örnek, TLS kimlik doğrulamasını kullanan bir MQTT kanalını yapılandırma örneği boyunca size yol sağlar.

Bu örnek, MQTT ile Mosquitto arasında bir kanal yapılandırır.

Bu örnek, CentOS üzerinde hem Red Hat Enterprise Linux üzerinde IBM MQ hem de Mosquitto için Docker kapsayıcısını kullanır, ancak herhangi bir sunucu tipi için geçerlidir. (CentOS , kayıt defteri yetkileri nedeniyle Mosquitto için kullanılmıştır.)

IBM MQ anahtar deposunu ve kanalı tek yönlü TLS için yapılandır

Aşağıdaki adımları izleyin:

1. IBM MQ anahtar deposunu yaratın:

```
runmqakm -keydb -create -db mqtt.kdb -stash -pw "passw0rd"
```

2. Kişisel sertifika yarat:

```
runmqakm -cert -create -db mqtt.kdb -stashed -size 2048 -dn "CN= 172.17.0.2, OU=MQTest, O=MQSuppor, C=US" -sig_alg SHA256_WITH_RSA -label ibmwebspheremqmqm
```

Sertifikanın yaratılmasını onaylamak için aşağıdaki komutu kullanabilirsiniz:

```
runmqakm -cert -list -v -db mqtt.kdb -stashed
```

3. Runmqsc komut isteminde aşağıdaki komutu girerek MQTT kanalını oluşturun:

```
DEFINE CHANNEL(MQTTDEMO) CHLTYPE(MQTT) BACKLOG(4096) PORT(8883) MCAUSER('mqm')  
PROTOCOL(MQTTV311,MQTTV3,HTTP) SSLCAUTH(OPTIONAL) SSLCIPH('SSL_RSA_WITH_AES_256_CBC_SHA256')  
SSLKEYR('/var/mqm/mqtt/mqtt.kdb') TRPTYPE(TCP)
```

Kanalın Java şifre eşlemelerini kullandığını unutmayın, bkz. [JMS için IBM MQ sınıflarında TLS CipherSpecs ve CipherSuites](#).

4. Sertifikayı çıkar:

```
runmqakm -cert -extract -db mqtt.kdb -stashed -label ibmwebspheremqmqm -target serverCert.pem
```

Docker kapsayıcısında CentOS üzerine Mosquitto 'yu kurun

CentOS üzerinde çalışan Mosquitto ile bir Docker kapsayıcısı oluşturmak için aşağıdaki adımları tamamlayın:

1. docker pull centos
2. docker run -it centos /bin/bash
3. yum -y install epel-release
4. yum -y install mosquitto

İmzalayıcı sertifikasını Mosquitto 'ya taşı

IBM MQ içinde oluşturduğunuz sertifikayı Mosquitto 'ya taşımak için aşağıdaki adımları tamamlayın. Bu adımlar, Docker anasistem makinesinde çalıştırılır.

1. Docker' da taşıyıcı tanıtıcılarını görüntüleyin:

```
docker container ls
```

2. Docker kapsayıcısından yerel sistem docker'ınıza dosyayı kopyalayın


```
cp MQ_Container_ID:/var/mqm/mqtt/serverCert.pem serverCert.pem
```

3. Dosyayı yerel makinenizden CentOS makinesindeki kök dizine kopyalayın:

```
docker cp serverCert.pem CentOS_ContainerID:/serverCert.pem
```

Sivrisinekle Yayınla

Aşağıdaki komutu kullanarak Mosquitto 'da bir test iletisi yayınlayın:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d
```

Komut bağımsız değişkenleri aşağıdaki anlamlara sahiptir:

-h

Red Hat Enterprise Linux anasistem IP adresi (**nslookup** kullanılarak bulunabilir).

-- cafile

İmzalayıcı sertifikasını içeren dosya.

-- güvenli olmayan

Bu seçenek, örnek kendinden onaylı bir sertifika kullandığı için belirtilir. Gerçek CA sertifikalarını kullanırken bu seçeneği kullanmayın.

-p

Kapı numarası.

-i

Müşteri Kimliği.

-t

Yayınlanmakta olan konu.

-m

Yayınlanmakta olan ileti.

-d

Hata ayıklama iletilerini etkinleştirin.

Karşılıklı TLS kimlik doğrulaması için MQTT kanalını yapılandırma

MQTT kanalını SSLCAUTH (REQUIRED) olarak yeniden yapılandırmak için aşağıdaki komutu girin.

```
ALTER CHANNEL(MQTTDEMO) CHLTYPE(MQTT) SSLCAUTH(REQUIRED)
```

Mosquitto sunucusunda bir anahtar/sertifika çifti oluşturun ve IBM MQ ' a ekleyin

Mosquitto 'da anahtar/sertifika çiftini oluşturmak için aşağıdaki komutları girin:

1. Mosquitto için anahtar/sertifika çifti oluşturmak için **openssl** kullanın:

```
openssl req -x509 -newkey rsa:4096 -keyout mosquittoKey.pem -out mosquittoCert.pem -subj "/CN=Mosquitto"
```

2. Taşıyıcılara ilişkin taşıyıcı tanıtıcılarını listele:

```
docker container ls
```

3. Sivrisinek sertifikasını yerel sistem docker'ınıza kopyalayın:

```
docker cp CentOS_ContainerID:mosquittoCert.pem .
```

4. Sivrisinek sertifikasını IBM MQ' e kopyalayın:

```
docker cp mosquittoCert.pem MQ_Container_ID:/var/mqm/mqtt
```

5. Sertifikayı IBM MQ anahtar deposuna ekleyin:

```
runmqakm -cert -add -db mqtt.kdb -stashed -file mosquitoCert.pem
```

6. MQTT kanalını yeniden başlatın.

Sivrisineklerle ve karşılıklı kimlik doğrulamasıyla yayınla

Karşılıklı kimlik doğrulamasını kullanarak Mosquitto ile yayınlamak için aşağıdaki adımları tamamlayın.

1. Aşağıdaki komut bir sınıma iletisini başarıyla yayınlamalıdır:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d --cert mosquitoCert.pem --key mosquitoKey.pem
```

2. Aşağıdaki komut, Mosquitto 'dan kişisel sertifika göndermediği için bir test iletisi yayınlamamalı ve bir hata iletisi oluşturmamalıdır:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d /var/mqm/qmgrs/mqttDemoQM/errors/ mqxr_0.log
```

İlgili bilgiler

[Anahtarları ve sertifikaları yönetme](#)

Linux

Windows

AIX

SHA-2 şifreleme takımlarının MQTT kanallarıyla kullanılmasına ilişkin sistem gereksinimleri

Java ' in SHA-2 şifreleme takımlarını destekleyen bir sürümünü kullanıyorsanız, MQTT (telemetri) kanallarınızı ve istemci uygulamalarınızı korumak için bu takımları kullanabilirsiniz.

Telemetri (MQXR) hizmetini içeren IBM MQ 8.0 için en düşük Java sürümü Java 7 from IBM , SR6' dir. SHA-2 şifreleme takımları, varsayılan olarak Java 7 from IBM, SR4 içinde desteklenir. Bu nedenle, MQTT (telemetri) kanallarınızı korumak için SHA-2 şifreleme takımlarını telemetri (MQXR) hizmetiyle kullanabilirsiniz.

MQTT istemcisini farklı bir JRE ile çalıştırıyorsanız, bunun SHA-2 şifreleme takımlarını da desteklediğinden emin olmanız gerekir.

İlgili kavramlar

[Telemetry \(MQXR\) hizmeti](#)

“TLS kullanarak kanal kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 359

IBM MQ yöneticisi, sunucuda telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasında TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar dosyalarına parola korumalı erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcesi ya da anahtar dosyası olmadan tanımlanırsa, kanal TLS bağlantılarını kabul etmez.

İlgili başvurular

[KANAL TANIMLAYIN \(MQTT\)](#)

[DEĞİŞTİRME KANALI \(MQTT\)](#)

Linux

Windows

AIX

Telemetri kanallarında yayın gizliliği

Telemetri kanalları üzerinden her iki yönde gönderilen MQTT yayınlarının gizliliği, bağlantı üzerinden iletileri şifrelemek için TLS kullanılarak korunmaktadır.

Telemetri kanallarına bağlanan MQTT istemcileri, simetrik anahtar şifrelemesi kullanarak kanalda iletilen yayınların gizliliğini korumak için TLS ' yi kullanır. Uç noktaların kimliği doğrulanmadığından, kanal şifrelemesine tek başına güvenemezsiniz. Güvenli gizliliği sunucu veya karşılıklı kimlik doğrulamasıyla birleştirin.

SSL kullanımına alternatif olarak, IPsec gibi bazı Sanal Özel Ağ (VPN) türleri, bir TCP/IP bağlantısının uç noktalarını doğrular. VPN, ağ üzerinden akan her bir IP paketini şifreler. Böyle bir VPN bağlantısı

kurulduktan sonra, güvenilir bir ağ oluşturmuş olunuz. VPN ağı üzerinden TCP/IP ' yi kullanarak MQTT istemcilerini telemetri kanallarına bağlayabilirsiniz.

Kanalı şifreleyen ve sunucuyu doğrulayan tipik bir yapılandırma için bkz. [“TLS kullanarak telemetri kanalı kimlik doğrulaması” sayfa 359.](#)

Sunucunun kimliğini doğrulamadan TLS bağlantılarının şifrlenmesi, ortadaki adam saldırılarına bağlantıyı gösterir. Değiştirdiğiniz bilgiler gizlice dinlenmeye karşı korunsada, kimle değiş tokuş yaptığınızı bilmiyorsunuz. Ağı denetlemezseniz, IP iletimlerinizi alan ve uç nokta olarak gizleyen bir kişiye maruz kalırsınız.

Anonim TLS ' yi destekleyen bir Diffie-Hellman anahtar değiş tokuşu CipherSpec kullanarak, sunucunun kimliğini doğrulamadan şifrenmiş bir TLS bağlantısı oluşturabilirsiniz. İstemci ve sunucu arasında paylaşılan ve TLS iletimini şifrelemek için kullanılan ana güvenlik dizgisi, özel olarak imzalanmış bir sunucu sertifikası değiş tokuş edilmeden oluşturulur.

Anonim bağlantılar güvenli olmadığından, çoğu TLS uygulaması varsayılan olarak anonim CipherSpecs' i kullanmaz. TLS bağlantısı için bir istemci isteği bir telemetri kanalı tarafından kabul edilirse, kanalın bir geçiş tümcecığı tarafından korunan bir anahtar deposu olması gerekir. Varsayılan olarak TLS uygulamaları anonim CipherSpecskullanmadığı için anahtar deposu, istemcinin kimlik doğrulayabileceği özel olarak imzalanmış bir sertifika içermelidir.

Anonim CipherSpecskullanıyorsanız, sunucu anahtar deposu var olmalıdır, ancak özel olarak imzalanmış sertifika içermesi gerekmez.

Şifrenmiş bir bağlantı oluşturmanın başka bir yolu da istemcideki güven sağlayıcıyı kendi uygulamanızla değiştirmektir. Güven sağlayıcınız sunucu sertifikasının kimliğini doğrulamaz, ancak bağlantı şifrenilir.



Uyarı: TLS ' yi MQTT ile kullanırken büyük iletileri kullanabilirsiniz, ancak bunu yaparken olası bir performans etkisi olabilir. MQTT , küçük iletileri işlemek için optimize edilmiştir (genellikle 1KB ile 1MB arasında).

Linux

Windows

AIX

MQTT Java istemcilerinin ve telemetri

kanallarının TLS yapılandırması

Telemetri kanalını ve MQTT Java istemcisini doğrulamak için TLS ' yi yapılandırın ve bunlar arasındaki iletilerin aktarımını şifreleyin. MQTT Java istemcileri, TLS kullanarak telemetri kanallarını bağlamak için Java Secure Socket Extension (JSSE) olanağını kullanır. SSL kullanımına alternatif olarak, IPsec gibi bazı Sanal Özel Ağ (VPN) türleri, bir TCP/IP bağlantısının uç noktalarını doğrular. VPN, ağ üzerinden akan her bir IP paketini şifreler. Böyle bir VPN bağlantısı kurulduktan sonra, güvenilir bir ağ oluşturmuş olunuz. VPN ağı üzerinden TCP/IP ' yi kullanarak MQTT istemcilerini telemetri kanallarına bağlayabilirsiniz.

TCP/IP üzerinden TLS iletişim kuralını kullanmak için bir Java MQTT istemcisi ile bir telemetri kanalı arasındaki bağlantıyı yapılandırabilirsiniz. Güvenli olan, TLS 'yi JSSE' yi kullanacak şekilde nasıl yapılandırıdığınıza bağlıdır. En güvenli yapılandırmayla başlayarak, üç farklı güvenlik düzeyi yapılandırabilirsiniz:

1. Yalnızca güvenilir MQTT istemcilerinin bağlanmasına izin verin. Bir MQTT istemcisini yalnızca güvenilir bir telemetri kanalına bağlayın. İstemci ile kuyruk yöneticisi arasındaki iletileri şifreleyin; bkz. [“TLS kullanarak MQTT istemci kimlik doğrulaması” sayfa 357](#)
2. Bir MQTT istemcisini yalnızca güvenilir bir telemetri kanalına bağlayın. İstemci ile kuyruk yöneticisi arasındaki iletileri şifreleyin; bkz. [“TLS kullanarak telemetri kanalı kimlik doğrulaması” sayfa 359.](#)
3. İstemci ile kuyruk yöneticisi arasındaki iletileri şifreleyin; bkz. [“Telemetri kanallarında yayın gizliliği” sayfa 362.](#)

JSSE yapılanış değiştirgeleri

TLS bağlantısının yapılandırılmasını değiştirmek için JSSE değiştirgelerini değiştirin. JSSE yapılanış değiştirgeleri üç küme halinde düzenlenir:

1. [MQ Telemetry kanal](#)

2. MQTT Java istemcisi

3. JRE

IBM MQ Explorer 'ı kullanarak telemetri kanal parametrelerini yapılandırın. `MqttConnectionOptions.SSLProperties` özniteliğinde MQTT Java Client parametrelerini ayarlayın. JRE güvenlik deęiřtirgelerini deęiřtirmek için, istemcinin ve sunucunun JRE güvenlik dizinindeki dosyaları düzenleyin.

MQ Telemetry channel

IBM MQ Explorer 'ı kullanarak tüm telemetri kanalı TLS parametrelerini ayarlayın.

ChannelName

`ChannelName` , tüm kanallarda gerekli bir parametredir.

Kanal adı, belirli bir kapı numarasıyla ilişkili kanalı tanımlar. MQTT istemcisi kümelerini yönetmenize yardımcı olacak ad kanalları.

PortNumber

`PortNumber` , tüm kanallarda isteęe baęlı bir parametredir. Varsayılan olarak TCP kanalları için 1883 ve TLS kanalları için 8883 deęerine ayarlanır.

Bu kanalla ilişkili TCP/IP kapı numarası. MQTT istemcileri, kanal için tanımlanan kapı belirtilerek bir kanala baęlanır. Kanal TLS özelliklerine sahipse, istemcinin TLS iletişim kuralını kullanarak baęlanması gerekir; örneęin:

```
MQTTClient mqttClient = new MqttClient( "ssl://www.example.org:8884", "clientId");
mqttClient.connect();
```

KeyFileAdı

`KeyFileName` , TLS kanalları için gerekli bir parametredir. TCP kanalları için atlanmalıdır.

`KeyFileAdı` , saęladığınız sayısal sertifikaları içeren Java anahtar deposunun yoludur. Sunucudaki anahtar deposu tipi olarak JKS, JCEKS ya da PKCS12 kullanın.

Ařaęıdaki dosya uzantılarından birini kullanarak anahtar deposu tipini saptayın:

- .jks
- .jceks
- .p12
- .pkcs12

Bařka bir dosya uzantısına sahip bir anahtar deposunun JKS anahtar deposu olduęu varsayılır.

Sunucudaki bir anahtar deposu tipini istemcideki dięer anahtar deposu tipleriyle birleřtirebilirsiniz.

Sunucunun özel sertifikasını anahtar deposuna yerleřtirin. Sertifika, sunucu sertifikası olarak bilinir. Sertifika, kendinden imzalı ya da bir imza yetkilisi tarafından imzalanmış bir sertifika zincirinin parçası olabilir.

Bir sertifika zinciri kullanıyorsanız, ilişkili sertifikaları sunucu anahtar deposuna yerleřtirin.

Sunucu sertifikası ve sertifika zincirindeki tüm sertifikalar, sunucunun kimlięini doęrulamak için istemcilere gönderilir.

`ClientAuth` deęerini `Required` olarak ayarladıysanız, anahtar deposu, istemcinin kimlięini doęrulamak için gereken tüm sertifikaları içermelidir. İstemci kendinden onaylı bir sertifika ya da bir sertifika zinciri gönderir ve istemcinin kimlięi, bu malzemenin anahtar deposundaki bir sertifikaya karřı ilk doęrulamasıyla doęrulandır. Bir sertifika zinciri kullanarak, bir sertifika, farklı istemci sertifikalarıyla verilmiş olsa da, birçok istemciyi doęrulayabilir.

PassPhrase

`PassPhrase` , TLS kanalları için gerekli bir parametredir. TCP kanalları için atlanmalıdır.

Anahtar deposunu korumak için parola kullanılır.

ClientAuth

ClientAuth isteğe bağlı bir TLS parametresidir. Varsayılan olarak istemci kimlik doğrulaması yoktur. TCP kanalları için atlanmalıdır.

İstemcinin telemetri kanalına bağlanmasına izin vermeden önce telemetri (MQXR) hizmetinin istemcinin kimliğini doğrulamasını istiyorsanız, ClientAuth değerini ayarlayın.

ClientAuthayarını tanımlarsanız, istemcinin TLS kullanarak sunucuya bağlanması ve sunucunun kimliğini doğrulaması gerekir. ClientAuthayarını tanımlamaya yanıt olarak istemci, sayısal sertifikasını sunucuya ve anahtar deposundaki diğer sertifikalara gönderir. Sayısal sertifikası, istemci sertifikası olarak bilinir. Bu sertifikaların kimliği, kanal anahtar deposunda ve JRE cacerts deposunda tutulan sertifikalara göre doğrulanır.

CipherSuite

CipherSuite isteğe bağlı bir TLS parametresidir. Varsayılan değer olarak, etkinleştirilen tüm CipherSpecs' ı dener. TCP kanalları için atlanmalıdır.

Belirli bir CipherSpeckullanmak istiyorsanız, TLS bağlantısını kurmak için kullanılması gereken CipherSpec ' in adına CipherSuite değerini ayarlayın.

Telemetri hizmeti ve MQTT istemcisi, her bir uçta etkinleştirilen tüm CipherSpecs ürünlerinden ortak bir CipherSpec anlaşması sağlar. Bağlantının uçlarından birinde ya da her ikisinde belirli bir CipherSpec belirtilirse, diğer uçta CipherSpec ile eşleşmelidir.

JSE ' ye ek sağlayıcılar ekleyerek ek şifrelemeler kurun.

Federal Bilgi İşleme Standartları (FIPS)

FIPS isteğe bağlı bir ayardır. Varsayılan olarak ayarlanmaz.

Kuyruk yöneticisinin özellikler panosunda ya da **runmqsc**komutunu kullanarak SSLFIPSayarını yapın. SSLFIPS , yalnızca FIPS onaylı algoritmaların kullanılıp kullanılmayacağını belirtir.

İptal ad listesi

İptal namelist, isteğe bağlı bir ayardır. Varsayılan olarak ayarlanmaz.

Kuyruk yöneticisinin özellikler panosunda ya da **runmqsc**komutunu kullanarak SSLCRLNLdeğerini ayarlayın. SSLCRLNL , sertifika iptal konumlarını sağlamak için kullanılan kimlik doğrulama bilgileri nesnelere ad bilgilerini belirtir.

TLS özelliklerini ayarlanan başka kuyruk yöneticisi parametresi kullanılmaz.

MQTT Java istemcisi

MqttConnectionOptions.SSLProperties içinde Java istemcisi için TLS özelliklerini ayarlama; örneğin:

```
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.keyStoreType", "JKS");
com.ibm.micro.client.mqttv3.MqttConnectOptions conOptions = new MqttConnectOptions();
conOptions.setSSLProperties(sslClientProperties);
```

Belirli özelliklerin adları ve değerleri MqttConnectOptions sınıfında açıklanır. MQTT istemci kitaplıklarına ilişkin istemci API belgelerine ilişkin bağlantılar için bkz. [MQTT istemci programlama başvurusu](#).

Protokol

Protokol isteğe bağlıdır.

Protokol, telemetri sunucusu ile anlaşma için seçilir. Belirli bir iletişim kuralına gereksinim duyarsanız, bir iletişim kuralı seçebilirsiniz. Telemetri sunucusu protokolü desteklemiyorsa, bağlantı başarısız olur.

ContextProvider

ContextProvider isteğe bağlıdır.

KeyStore

KeyStore isteğe bağlıdır. İstemcinin kimlik doğrulamasını zorlamak için sunucuda ClientAuth ayarlandıysa bunu yapılandırın.

İstemcinin özel anahtarı kullanılarak imzalanan sayısal sertifikasını anahtar deposuna yerleştirin. Anahtar deposu yolunu ve parolasını belirtin. Tip ve sağlayıcı isteğe bağlıdır. Varsayılan tip JKS ve varsayılan sağlayıcı IBMJCE 'dir.

Yeni bir anahtar deposu sağlayıcısı ekleyen bir sınıfa gönderme yapmak için farklı bir anahtar deposu sağlayıcısı belirtin. Anahtar deposu sağlayıcısı tarafından anahtar yöneticisi adını ayarlayarak KeyManagerÜreticisi ' ni somutlaştırmak için kullanılan algoritmanın adını geçirin.

TrustStore

TrustStore isteğe bağlıdır. Güvendiğiniz tüm sertifikaları JRE cacerts deposuna koyabilirsiniz.

İstemci için farklı bir güvenli depoya sahip olmak istiyorsanız güvenli depoyu yapılandırın. Sunucu, kök sertifikasını zaten cacerts içinde saklamış olan bilinen bir sertifika kuruluşu tarafından verilen bir sertifikayı kullanıyorsa, güvenli depoyu yapılandırmayabilirsiniz.

Sunucunun ya da kök sertifikanın genel olarak imzalanmış sertifikasını güvenli depoya ekleyin ve güvenli depo yolunu ve parolayı belirtin. Varsayılan tip JKS ve varsayılan sağlayıcı IBMJCE 'dir.

Yeni bir güvenli depo sağlayıcısı ekleyen bir sınıfa gönderme yapmak için farklı bir güvenli depo sağlayıcısı belirtin. Güvenilirlik yöneticisi adını ayarlayarak TrustManagerFactory ögesini somutlaştırmak için güvenilirlik deposu sağlayıcısı tarafından kullanılan algoritmanın adını iletin.

JRE

Java güvenliğinin hem istemci hem de sunucu üzerindeki TLS davranışını etkileyen diğer yönleri JRE ' de yapılandırılır. Windows üzerindeki yapılandırma dosyaları *Java Installation Directory*\jre\lib\security içinde yer almaktadır. IBM MQ ile verilen JRE ' yi kullanıyorsanız, yol aşağıdaki çizelgede gösterildiği gibidir:

Hizmet olarak sunulan	filePath
Windows	<i>WMQ Installation Directory</i> \java\jre\lib\security
AIX and Linux Platformlar	<i>WMQ Installation Directory</i> /java/jre64/jre/lib/security

Tanınmış sertifika yetkilileri

cacerts dosyası, tanınan sertifika yetkililerinin kök sertifikalarını içerir. Bir güvenli depo belirtmezseniz, varsayılan olarak cacerts kullanılır. cacerts deposunu kullanıyorsanız ya da bir güvenli depo sağlamazsanız, güvenlik gereksinimlerinizi karşılamak için cacerts içindeki imzalayanlar listesini gözden geçirmeniz ve düzenlemeniz gerekir.

cacerts komutunu, IBM Key Management yardımcı programını çalıştıran IBM MQ komutu stmqikm kullanarak açabilirsiniz. changeit parolasını kullanarak cacerts dosyasını JKS dosyası olarak açın. Dosyayı korumak için parolayı değiştirin.

Güvenlik sınıflarının yapılandırılması

Ek güvenlik sağlayıcılarını ve diğer varsayılan güvenlik özelliklerini kaydetmek için java.security dosyasını kullanın.

İzinler

Kaynaklara verilen izinleri değiştirmek için `java.policy` dosyasını kullanın. `javaws.policy` izin verir `javaws.jar`

Şifreleme gücü

Bazı JRE 'ler düşük güçlü şifrelemeyle birlikte gönderiliyor. Anahtarları anahtar depolarına aktaramıyorsanız, daha az güçlü şifreleme bunun nedeni olabilir. **ikeyman** ' i **strmqikm** komutunu kullanarak başlatmayı deneyin ya da IBM geliştirici takımlarından, Güvenlik bilgilerindengüçlü, ancak sınırlı yargı yetkisi alanı dosyalarını karşıdan yükleyin.

Önemli: Kaynak ülkenizde, şifreleme yazılımının ithal, zilyedlik, kullanım ya da başka bir ülkeye yeniden ihracatına ilişkin kısıtlamalar olabilir. Sınırlanmamış ilke dosyalarını karşıdan yüklemeyen ya da kullanmadan önce, ülkenizin yasalarını kontrol etmelisiniz. İzin verilip verilmediğini belirlemek için şifreleme yazılımının ithalatına, zilyedliğine, kullanımına ve yeniden ihracatına ilişkin düzenlemelerini ve ilkelerini kontrol edin.

İstemcinin herhangi bir sunucuya bağlanmasına izin vermek için güven sağlayıcısını değiştirin

Bu örnek, bir güven sağlayıcısının nasıl ekleneceğini ve MQTT istemci kodundan bu sağlayıcıya nasıl başvuruda bulunacağını gösterir. Örnek, istemci ya da sunucu için kimlik doğrulaması gerçekleştirmez. Sonuçta elde edilen TLS bağlantısı, kimliği doğrulanmadan şifrelenir.

Şekil 16 sayfa 367 içindeki kod parçacığı, MQTT istemcisi için `AcceptAllProviders` güven sağlayıcısını ve güvenilirlik yöneticisini ayarlar.

```
java.security.Security.addProvider(new AcceptAllProvider());
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.trustManager", "TrustAllCertificates");
sslClientProperties.setProperty("com.ibm.ssl.trustStoreProvider", "AcceptAllProvider");
conOptions.setSSLProperties(sslClientProperties);
```

Şekil 16. MQTT İstemci kodu parçacığı

```
package com.ibm.mq.id;
public class AcceptAllProvider extends java.security.Provider {
private static final long serialVersionUID = 1L;
public AcceptAllProvider() {
super("AcceptAllProvider", 1.0, "Trust all X509 certificates");
put("TrustManagerFactory.TrustAllCertificates",
AcceptAllTrustManagerFactory.class.getName());
}
}
```

Şekil 17. `AcceptAllProvider.java`

```
protected static class AcceptAllTrustManagerFactory extends
javax.net.ssl.TrustManagerFactorySpi {
public AcceptAllTrustManagerFactory() {}
protected void engineInit(java.security.KeyStore keystore) {}
protected void engineInit(
javax.net.ssl.ManagerFactoryParameters parameters) {}
protected javax.net.ssl.TrustManager[] engineGetTrustManagers() {
return new javax.net.ssl.TrustManager[] { new AcceptAllX509TrustManager() };
}
}
```

Şekil 18. `AcceptAllTrustManagerFactory.java`

```

protected static class AcceptAllX509TrustManager implements
javax.net.ssl.X509TrustManager {
public void checkClientTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Client authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public void checkServerTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Server authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public java.security.cert.X509Certificate[] getAcceptedIssuers() {
return new java.security.cert.X509Certificate[0];
}
private static void report(String string) {
System.out.println(string);
}
}
}

```

Şekil 19. AcceptAllX509TrustManager.java

Linux

Windows

AIX

Telemetri kanalı JAAS yapılandırması

JAAS ' ı istemci tarafından gönderilen Username kimliğini doğrulayacağı şekilde yapılandırın.

IBM MQ yöneticisi, JAASKullanarak istemci kimlik doğrulaması gerektiren MQTT kanallarını yapılandırır. JAAS kimlik doğrulamasını gerçekleştirecek her kanal için JAAS yapılandırmasının adını belirtin. Kanalların tümü aynı JAAS yapılandırmasını kullanabilir ya da farklı JAAS yapılandırmaları kullanabilir. Yapılandırmalar *WMQData directory\qmgrs\qMgrName\mqxr\jaas.config* içinde tanımlanır.

jaas.config dosyası, JAAS yapılandırma adına göre düzenlenir. Her yapılandırma adının altında Oturum Açma yapılandırmalarının bir listesi bulunur; bkz. "[Örnek jaas.config dosyası](#)" sayfa 369.

JAAS , dört standart Oturum Açma modülü sağlar. Standart NT ve UNIX Oturum açma modülleri sınırlı değere sahiptir.

JndiLoginModülü

JNDI (Java Naming and Directory Interface; Adlandırma ve Dizin Arabirimi) altında yapılandırılan bir dizin hizmetine ilişkin kimlik doğrulamasını yapar.

Krb5LoginModule

Kerberos iletişim kurallarını kullanarak kimlik doğrular.

NTLoginModule

Geçerli kullanıcı için NT güvenlik bilgilerini kullanarak kimlik doğrular.

UnixLoginModülü

Geçerli kullanıcı için UNIX güvenlik bilgilerini kullanarak kimlik doğrular.

NTLoginModule ya da UnixLoginModule kullanımıyla ilgili sorun, telemetri (MQXR) hizmetinin MQTT kanalının kimliğiyle değil, mqm kimliğiyle çalışmasıdır. mqm , istemcinin kimliği değil, kimlik doğrulaması için NTLoginModule ya da UnixLoginModule ' e iletilen kimliktir.

Bu sorunu çözmek için kendi Oturum Açma modülünüzü yazın ya da diğer standart Oturum Açma modüllerini kullanın. MQ Telemetry ile birlikte bir örnek JAASLoginModule .java sağlanır. Bu, javax.security.auth.spi.LoginModule arabiriminin bir uygulamasıdır. Kendi kimlik doğrulama yönteminizi geliştirmek için bunu kullanın.

Sağladığınız yeni LoginModule sınıfları, telemetri (MQXR) hizmetinin sınıf yolunda olmalıdır. Sınıflarınızı, sınıf yolundaki IBM MQ dizinlerine yerleştirmeyin. Kendi dizinlerinizi yaratın ve telemetri (MQXR) hizmeti için tüm sınıf yolunu tanımlayın.

service.env dosyasında sınıf yolunu ayarlayarak telemetri (MQXR) hizmeti tarafından kullanılan sınıf yolunu genişletebilirsiniz. CLASSPATH büyük harfle yazılmalı ve sınıf yolu deyimi yalnızca hazır bilgi içerebilir. CLASSPATH içinde değişkenleri kullanamazsınız; örneğin, CLASSPATH=%CLASSPATH% yanlış. Telemetri (MQXR) hizmeti kendi sınıf yolunu ayarlar. service.env içinde tanımlanan CLASSPATH buna eklenir.

Telemetry (MQXR) hizmeti, MQTT kanalına bağlı bir istemci için Kullanıcı adı ve Parola değerlerini döndüren iki geri çağrı sağlar. Kullanıcı adı ve Parola, MqttConnectOptions nesnesinde ayarlanır. Kullanıcı adı ve Parola'ya nasıl erişileceğine ilişkin bir örnek için bkz. "[Örnek JAASLoginModule.Login\(\) yöntemi](#)" sayfa 369.

Örnek jaas.config dosyası

Alandırılmış bir yapılandırmaya sahip bir JAAS yapılandırma dosyası örneği, MQXRConfig

```
MQXRConfig {
samples.JAASLoginModule required debug=true;
//com.ibm.security.auth.module.NTLoginModule required;
//com.ibm.security.auth.module.Krb5LoginModule required
//    principal=principal@your_realm
//    useDefaultCcache=TRUE
//    renewTGT=true;
//com.sun.security.auth.module.NTLoginModule required;
//com.sun.security.auth.module.UnixLoginModule required;
//com.sun.security.auth.module.Krb5LoginModule required
//    useTicketCache="true"
//    ticketCache="$${user.home}/${}tickets";
};
```

Örnek JAASLoginModule.Login() yöntemi

Bir MQTT istemcisi tarafından sağlanan Kullanıcı Adı ve Parola değerlerini almak için kodlanmış JAAS oturum açma birimi örneği.

```
public boolean login()
throws javax.security.auth.login.LoginException {
    javax.security.auth.callback.Callback[] callbacks =
    new javax.security.auth.callback.Callback[2];
    callbacks[0] = new javax.security.auth.callback.NameCallback("NameCallback");
    callbacks[1] = new javax.security.auth.callback.PasswordCallback(
    "PasswordCallback", false);
    try {
        callbackHandler.handle(callbacks);
        String username = ((javax.security.auth.callback.NameCallback) callbacks[0])
        .getName();
        char[] password = ((javax.security.auth.callback.PasswordCallback) callbacks[1])
        .getPassword();
        // Accept everything.
        if (true) {
            loggedIn = true;
        } else
        {
            throw new javax.security.auth.login.FailedLoginException("Login failed");
        }
        principal= new JAASPrincipal(username);
    } catch (java.io.IOException exception) {
        throw new javax.security.auth.login.LoginException(exception.toString());
    } catch (javax.security.auth.callback.UnsupportedCallbackException exception) {
        throw new javax.security.auth.login.LoginException(exception.toString());
    }
}

return loggedIn;
}
```

İlgili görevler

Sorun çözülüyor: JAAS oturum açma modülü telemetri hizmeti tarafından çağrılmadı

İlgili başvurular

[AuthCallback MQXR sınıfı](#)

AMQP istemcisinin yönetilmesi

Bir AMQP istemcisini IBM MQ Explorer komutunu kullanarak ya da bir komut satırında denetleyebilirsiniz. IBM MQ' e bağlı AMQP istemcilerini izlemek ve kanalları yapılandırmak için Explorer 'ı kullanın. TLS ve JAASKullanarak AMQP istemcilerinin güvenliğini yapılandırın.

Başlamadan önce

AMQP ' nin platformunuza kurulmasıyla ilgili bilgi için [Ne kurulacağını seçme](#) başlıklı konuya bakın.

IBM MQ Explorer kullanarak yönetme

AMQP kanallarını yapılandırmak ve IBM MQ' e bağlı AMQP istemcilerini izlemek için Explorer 'ı kullanın. TLS ve JAASKullanarak AMQP istemcilerinin güvenliğini yapılandırabilirsiniz.

Komut satırını kullanarak yönetme

Bir AMQP istemcisini [MQSC komutlarını](#) kullanarak komut satırında denetleyebilirsiniz.

AMQP istemcileri tarafından kullanılan IBM MQ nesnelerini görüntüleme

Bağlantılar ve abonelikler gibi AMQP istemcileri tarafından kullanılan farklı IBM MQ kaynaklarını görüntüleyebilirsiniz.

Bağlantılar

AMQP hizmeti başlatıldığında, yeni Hconns yaratılır ve kuyruk yöneticisine bağlanır. Bu Hconn havuzu AMQP istemcileri ileti yayınlarken kullanılır. Hconns 'u **DISPLAY CONN** komutunu kullanarak görüntüleyebilirsiniz. Örneğin:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (APPLDESC LK 'IBM MQ Advanced Message Queuing Protocol*')
```

Bu komut, istemciye özgü Hconnları da gösterir. Boş istemci tanıtıcısı özneliği olan Hconn 'lar, havuzda kullanılan Hconns değerleridir.

Bir AMQP istemcisi bir AMQP kanalına bağlandığında, kuyruk yöneticisine yeni bir Hconn bağlanır. Bu Hconn, AMQP istemcisinin yarattığı abonelikler için iletileri zamanuyumsuz olarak kullanmak için kullanılır. Belirli bir AMQP istemcisi tarafından kullanılan Hconn 'u **DISPLAY CONN** komutunu kullanarak görüntüleyebilirsiniz. Örneğin:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_abcd1234')
```

İstemciler tarafından oluşturulan abonelikler

Bir AMQP istemcisi bir konuya abone olduğunda, yeni bir IBM MQ aboneliği oluşturulur. Abonelik adı aşağıdaki bilgileri içerir:

- İstemcinin adı. İstemci paylaşılan bir aboneliğe katıldıysa, paylaşımın adı kullanılır
- İstemcinin abone olduğu konu kalıbı
- Bir önek. İstemci paylaşılmayan bir abonelik oluşturduysa `private` öneki ya da istemci paylaşılan bir aboneliğe katıldıysa `share` önekidir.

Belirli bir AMQP istemcisi tarafından kullanılan abonelikleri görüntülemek için, **DISPLAY SUB** komutunu çalıştırın ve `private` önekinde süzün:

```
DISPLAY SUB(':private:*')
```

Birden çok istemci tarafından kullanılan paylaşılan abonelikleri görüntülemek için **DISPLAY SUB** komutunu çalıştırın ve share önekinde süzün:

```
DISPLAY SUB('/:share:*')
```

Paylaşılan abonelikler birden çok AMQP istemcisi tarafından kullanılabilir olduğundan, paylaşılan abonelikten gelen iletileri kullanmakta olan istemcileri görüntülemek isteyebilirsiniz. Bunu, şu anda abonelik kuyruğunda açık bir tanıtıcısı olan Hconnları listeleterek yapabilirsiniz. Şu anda bir paylaşım kullanan istemcileri görüntülemek için aşağıdaki adımları tamamlayın:

1. Paylaşılan aboneliğin hedef olarak kullandığı kuyruk adını bulun. Örneğin:

```
DISPLAY SUB('/:private:recv_e298452:public') DEST
5 : DISPLAY SUB('/:private:recv_e298452:public') DEST
AMQ8096: WebSphere MQ subscription inquired.
SUBID(414D5120514D3120202020202020202020707E0A565C2D0020)
SUB(/:private:recv_e298452:public)
DEST(SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
```

2. Bu kuyrukta açık tutamaçları bulmak için **DISPLAY CONN** komutunu çalıştırın:

```
DISPLAY CONN(*) TYPE(HANDLE) WHERE (OBJNAME
EQ SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
21 : DISPLAY CONN(*) TYPE(HANDLE) WHERE (OBJNAME EQ
SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)

AMQ8276: Display Connection details.
CONN(707E0A56642B0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(HANDLE)

OBJNAME(SYSTEM.BASE.TOPIC) OBJTYPE(TOPIC)

OBJNAME(SYSTEM.MANAGED.DURABLE.560A7E7020002961)
OBJTYPE(QUEUE)
```

3. Tutamaçların her biri için, tutamacı açık olan AMQP istemci tanıtıcısını görüntüleyin:

```
DISPLAY CONN(707E0A56642B0020) CLIENTID
23 : DISPLAY CONN(707E0A56642B0020) CLIENTID

AMQ8276: Display Connection details.
CONN(707E0A56642B0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_8f02c9d)
DISPLAY CONN(707E0A565F290020) CLIENTID
24 : DISPLAY CONN(707E0A565F290020) CLIENTID
AMQ8276: Display Connection details.
CONN(707E0A565F290020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_86d8888)
```

AMQP istemci tanıtıcısı, yetkilendirme ve kimlik doğrulaması

Diğer IBM MQ istemci uygulamaları gibi, AMQP bağlantılarını çeşitli şekillerde güvenceye alabilirsiniz.

IBM MQAMQP bağlantılarının güvenliğini sağlamak için aşağıdaki güvenlik özelliklerini kullanabilirsiniz:

- [Kanal kimlik doğrulama kayıtları](#)
- [Bağlantı kimlik doğrulaması](#)
- Kanal MCA kullanıcı yapılandırması
- IBM MQ yetki tanımları
- [TLS bağlantırlığı](#)

Güvenlik açısından, bağlantı kurulması aşağıdaki iki adımdan oluşur:

- Bağlantının devam edip etmeyeceğine karar verilmesi
- Uygulamanın sonraki yetki denetimleri için hangi IBM MQ kimliğinin varolduğuna karar verilmesi

Aşağıdaki bilgiler, farklı IBM MQ yapılandırmalarını ve AMQP istemcisi bağlantı kurmaya çalışıldığında üzerinde çalışılacak adımları özetler. Her IBM MQ yapılandırması açıklanan adımların tümünü kullanmaz. Örneğin, bazı yapılandırmalar şirket güvenlik duvarı içindeki bağlantılar için TLS kullanmaz ve bazı yapılandırmalar TLS kullanır, ancak kimlik doğrulaması için istemci sertifikaları kullanmaz. Birçok ortam özel ya da özel JAAS birimleri kullanmaz.

Bağlantı kurulması

Aşağıdaki adımlarda, bir AMQP istemcisi tarafından bağlantı kurulduğunda ne olduğu açıklanmaktadır. Adımlar, bağlantının devam edip etmeyeceğini ve uygulamanın yetki denetimi için hangi IBM MQ kimliğinin varsayıldığını belirler:

1. İstemci IBM MQ ile bir TLS bağlantısı açar ve bir sertifika sağlarsa, kuyruk yöneticisi istemci sertifikasını doğrulamayı dener.
2. İstemci kullanıcı adı ve parola kimlik bilgileri sağlarsa, kuyruk yöneticisi tarafından bir AMQP SASL çerçevesi alınır ve MQ CONNAUTH yapılandırması denetlenir.
3. MQ kanal kimlik doğrulama kuralları denetlenir (örneğin, IP adresi ve TLS sertifika DN ' inin geçerli olup olmadığı)
4. Kanal kimlik doğrulama kuralları aksini belirlemedikçe, MCAUSER kanalı bildirilir.
5. Bir JAAS modülü yapılandırıldıysa, çağrılır
6. MQ Sonuçtaki MQ kullanıcı kimliğine CONNECT yetki denetimi uygulandı.
7. Varsayılan bir IBM MQ kimliğiyle bağlantı kuruldu.

İleti yayınlanması

Aşağıdaki adımlarda, bir ileti AMQP istemcisi tarafından yayınlandığında ne olduğu açıklanmaktadır. Adımlar, bağlantının devam edip etmeyeceğini ve uygulamanın yetki denetimi için hangi IBM MQ kimliğinin varsayıldığını belirler:

1. AMQP bağlantı ekleme çerçevesi kuyruk yöneticisine ulaşır. Bağlantı sırasında oluşturulan MQ kullanıcı kimliği için, belirtilen konu dizgisi için IBM MQ yayınlama yetkisi denetlenir.
2. İleti, belirtilen konu dizgisine yayınlanır.

Konu örüntüsüne abone olma

Aşağıdaki adımlarda, AMQP istemcisi bir konu örüntüsüne abone olduğunda ne olacağı açıklanmaktadır. Adımlar, bağlantının devam edip etmeyeceğini ve uygulamanın yetki denetimi için hangi IBM MQ kimliğinin varsayıldığını belirler:

1. AMQP bağlantı ekleme çerçevesi kuyruk yöneticisine ulaşır. Bağlantı sırasında oluşturulan MQ kullanıcı kimliği için, belirtilen konu örüntüsü için IBM MQ abone olma yetkisi denetlenir.
2. Abonelik oluşturuldu.

AMQP istemci kimliği ve yetkilendirmesi

IBM MQ nesnelere erişim yetkisi için AMQP istemci kimliğini, AMQP kullanıcı adını ya da kanalda ya da bir kanal kimlik doğrulama kuralında tanımlanan ortak istemci kimliğini kullanın.

Yönetici, AMQP kanalını tanımlarken ya da değiştirirken, kuyruk yöneticisi CONNAUTH ayarını yapılandırarak ya da kanal kimlik doğrulama kurallarını tanımlayarak seçim yapar. Kimlik, IBM MQ konularına erişimi yetkilendirmek için kullanılır. Seçim, aşağıdakilere dayalı olarak yapılır:

1. Kanal USECLNTID özniteliği.

2. Kuyruk yöneticisi CONNAUTH kuralının ADOPTCTX özniteliği.
3. Kanalda tanımlanan MCAUSER özniteliği.
4. Eşleşen bir kanal kimlik doğrulama kuralının USERSRC özniteliği.

Beladan kaçının: Bu işlem tarafından seçilen kimliğe, örneğin, DISPLAY CHSTATUS (AMQP) komutu tarafından istemcinin MCAUSER değeri olarak atıfta bulunulur. Bunun, seçimde atıfta bulunulan kanalın MCAUSER ' i ile aynı kimlik olması gerekmediğini unutmayın (2).

AMQP kanalıyla ilişkili kimlik tarafından hangi nesnelere ve hangi işlemlerin kullanılacağı seçmek için IBM MQ **setmqaut** komutunu kullanın. Örneğin, aşağıdaki komutlar QM1kuyruk yöneticisi tarafından sağlanan AMQPClientkanal kimliğini yetkilendirir:

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPClient -all +pub +sub
```

ve

```
setmqaut -m QM1 -t qmgr -p AMQPClient -all +connect
```

Parola kullanarak AMQP istemci kimlik doğrulaması

İstemci parolasını kullanarak AMQP istemcisi kullanıcı adını doğrulayın. İstemciye yayınlama ve konulara abone olma yetkisi vermek için kullanılan kimlikten farklı bir kimlik kullanarak istemcinin kimliğini doğrulayabilirsiniz.

AMQP hizmeti, istemci kullanıcı adının kimliğini doğrulamak için MQ CONNAUTH ya da JAAS ' ı kullanabilir. Bunlardan biri yapılandırılırsa, istemci tarafından sağlanan parola MQ CONNAUTH yapılandırması ya da JAAS modülü tarafından doğrulanır.

Aşağıdaki yordamda, yerel işletim sistemi kullanıcılarına ve parolalara karşı tek tek kullanıcıların kimliğini doğrulamak ve başarılı olursa, AMQPUserortak kimliğini benimsemek için atılacak örnek adımlar açıklanır:

1. IBM MQ yöneticisi, AMQP kanalı MCAUSER kimliğini IBM MQ Explorer ' ı kullanarak AMQPUsergibi herhangi bir ada ayarlar.
2. IBM MQ yöneticisi, AMQPUser ' e herhangi bir konuyu yayınlama ve bu konuya abone olma yetkisi verir:

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPUser -all +pub +sub +connect
```

3. IBM MQ yöneticisi, istemci tarafından sunulan kullanıcı adını ve parolayı denetlemek için bir IDPWOS CONNAUTH kuralı yapılandırır. CONNAUTH kuralı CHECKCLNT (REQUIRED) ve ADOPTCTX (NO) belirlemelidir.

Not: Kanal kimlik doğrulama kurallarını kullanmanız ve MCAUSER kanal özniteliğini, kuyruk yöneticisine yönelik bağlantılar üzerinde daha fazla denetime izin vermek için ayrıcalığı olmayan bir kullanıcıya ayarlamanız önerilir.

Kanallarda yayın gizliliği

AMQP kanalları üzerinden her iki yönde gönderilen AMQP yayınlarının gizliliği, bağlantı üzerinden iletileri şifrelemek için TLS kullanılarak korunmaktadır.

AMQP kanallarına bağlanan AMQP istemcileri, simetrik anahtar şifrelemesi kullanarak kanalda iletilen yayınların gizliliğini korumak için TLS kullanır. Uç noktaların kimliği doğrulanmadığından, kanal şifrelemesine tek başına güvenemezsiniz. Güvenli gizliliği sunucu veya karşılıklı kimlik doğrulamasıyla birleştirin.

TLS kullanımına alternatif olarak, IPsec gibi bazı Sanal Özel Ağ (VPN) türleri, bir TCP/IP bağlantısının uç noktalarını doğrular. VPN, ağ üzerinden akan her bir IP paketini şifreler. Böyle bir VPN bağlantısı kurulduktan sonra, güvenilir bir ağ oluşturmuşsunuz. VPN ağı üzerinden TCP/IP ' yi kullanarak AMQP istemcilerini AMQP kanallarına bağlayabilirsiniz.

Sunucunun kimliğini doğrulamadan TLS bağlantılarının şifrenmesi, ortadaki adam saldırılarına bağlantıyı gösterir. Değiştirdiğiniz bilgiler gizlice dinlenmeye karşı korunsun da, kimle değiş tokuş yaptığınızı bilmiyorsunuz. Ağ denetlemezseniz, IP iletimlerinizi alan ve uç nokta olarak gizleyen bir kişiye maruz kalırsınız.

Anonim TLS 'yi destekleyen bir Diffie-Hellman anahtar değiş tokuşu CipherSpec kullanarak, sunucunun kimliğini doğrulamadan şifrenmiş bir TLS bağlantısı oluşturabilirsiniz. İstemci ve sunucu arasında paylaşılan ve TLS iletimini şifrelemek için kullanılan ana güvenlik dizgisi, özel olarak imzalanmış bir sunucu sertifikası değiş tokuş edilmeden oluşturulur.

Anonim bağlantılar güvenli olmadığından, çoğu TLS uygulaması varsayılan olarak anonim CipherSpecs' i kullanmaz. TLS bağlantısı için bir istemci isteği bir AMQP kanalı tarafından kabul edilirse, kanalın bir geçiş tümceciği tarafından korunan bir anahtar deposu olması gerekir. Varsayılan olarak TLS uygulamaları anonim CipherSpecskullanmadığı için anahtar deposu, istemcinin kimlik doğrulayabileceği özel olarak imzalanmış bir sertifika içermelidir.

Anonim CipherSpecskullanıyorsanız, sunucu anahtar deposu var olmalıdır, ancak özel olarak imzalanmış sertifika içermesi gerekmez.

Şifrenmiş bir bağlantı oluşturmanın başka bir yolu da istemcideki güven sağlayıcıyı kendi uygulamanızla değiştirmektir. Güven sağlayıcınız sunucu sertifikasının kimliğini doğrulamaz, ancak bağlantı şifrenir.

TLS ile AMQP istemcilerini yapılandırma

AMQP istemcilerini, ağ üzerinden akan verileri korumak ve istemcinin bağlandığı kuyruk yöneticisinin kimliğini doğrulamak için TLS kullanacak şekilde yapılandırabilirsiniz.

Bir AMQP istemcisinden AMQP kanalına bağlantı için TLS kullanmak üzere kuyruk yöneticisinin TLS olarak yapılandırıldığından emin olmanız gerekir. [Kuyruk yöneticisinde TLS yapılandırılması](#) , bir kuyruk yöneticisinin TLS sertifikalarını okuduğu anahtar deposunun nasıl yapılandırılacağını açıklar.

Kuyruk yöneticisi bir anahtar deposuyla yapılandırıldığında, istemcilerin bağlanacağı AMQP kanalında TLS özniteliklerini yapılandırmanız gerekir. AMQP kanallarının TLS yapılandırmasıyla ilgili dört özniteliği vardır:

SSLCAUTH

SSLCAUTH özniteliği, kuyruk yöneticisinin kimliğini doğrulamak için bir istemci sertifikası sunması için bir AMQP istemcisi gerektirip gerektirmeyeceğini belirtmek için kullanılır.

SSLCIPH

SSLCIPH özniteliği, TLS akışındaki verileri kodlamak için kanalın kullanması gereken şifreyi belirtir.

V 9.3.1 IBM MQ 9.3.1'den AMQP kanalları ANY* soysal CipherSpecs' i destekler. CipherSpecsile ilgili daha fazla bilgi için bkz. [CipherSpecs'](#) in etkinleştirilmesi.

SSLPEER

Bir bağlantıya izin veriliyorsa, istemci sertifikasının eşleşmesi gereken ayırt edici adı (DN) belirtmek için SSLPEER özniteliği kullanılır.

CERTLABL

CERTLABL, kuyruk yöneticisinin istemciye sunması gereken sertifikayı belirtir. Kuyruk yöneticisinin anahtar deposu birden çok sertifika içerebilir. Bu öznitelik, bu kanala yönelik bağlantılar için kullanılacak sertifikayı belirtmenizi sağlar. CERTLABL belirtilmezse, kuyruk yöneticisi anahtar havuzundaki sertifika, kuyruk yöneticisi CERTLABL özniteliğine karşılık gelen etiketle birlikte kullanılır.

AMQP kanalınızı TLS öznitelikleriyle yapılandırdığınızda, AMQP hizmetini aşağıdaki komutu kullanarak yeniden başlatmanız gerekir:

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

Bir AMQP istemcisi TLS ile korunan bir AMQP kanalına bağlandığında, istemci kuyruk yöneticisi tarafından sunulan sertifikanın kimliğini doğrular. Bunu yapmak için AMQP istemcinizi kuyruk yöneticisinin sertifikasını içeren bir güvenli depoyla yapılandırmanız gerekir. Bunu yapma adımları, kullandığınız AMQP istemcisine bağlı olarak değişir. Çeşitli AMQP istemcileri ve API 'leri hakkında bilgi için ilgili AMQP istemcisi belgelerine bakın.

İlgili başvurular

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

[Çoklu Platformda STOP SERVICE \(hizmeti durdur\)](#)

[START SERVICE \(hizmeti başlat\)-Çoklu Platformlar üzerinde](#)

AMQP istemcilerinin kuyruk yöneticisiyle bağlantısı kesiliyor

AMQP istemcilerinin kuyruk yöneticisiyle bağlantısını kesmek istiyorsanız, PURGE CHANNEL komutunu çalıştırın ya da AMQP istemcisiyle bağlantıyı durdurun.

- **PURGE CHANNEL** komutunu çalıştırın. Örneğin:

```
PURGE CHANNEL (MYAMQP) CLIENTID('recv_28dbb7e')
```

- Diğer bir seçenek olarak, aşağıdaki adımları tamamlayarak istemcinin bağlantısını kesmek için AMQP istemcisinin kullandığı bağlantıyı durdurun:

1. **DISPLAY CONN** komutunu çalıştırarak istemcinin kullandığı bağlantıyı bulun. Örneğin:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_28dbb7e')
```

Komut çıkışı aşağıdaki gibidir:

```
DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
40 : DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
AMQ8276: Display Connection details.
CONN(707E0A565F2D0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE (CONN)
CLIENTID(recv_28dbb7e)
```

2. Bağlantıyı durdurun. Örneğin:

```
STOP CONN(707E0A565F2D0020)
```

Çoklu yayın yönetimi

Çoklu yayın iletilerinin boyutunu küçültme ve veri dönüştürmeyi etkinleştirme gibi IBM MQ Multicast yönetim görevleri hakkında bilgi edinmek için bu bilgileri kullanın.

Çoklu yayın ile çalışmaya başlama

IBM MQ Multicast konularına ve iletişim bilgileri nesnelere başlamak için bu bilgileri kullanın.

Bu görev hakkında

IBM MQ Çok hedefli ileti sistemi, konuları grup adresleriyle eşleyerek iletileri teslim etmek için ağı kullanır. Aşağıdaki görevler, gerekli IP adresinin ve kapının çoklu yayın ileti sistemi için doğru şekilde yapılandırılıp yapılandırılmadığını sınamanın hızlı bir yoludur.

Çoklu yayın için COMMINFO nesnesi yaratılması

İletişim bilgileri (COMMINFO) nesnesi, çoklu yayın iletimi ile ilişkili öznitelikleri içerir. COMMINFO nesne parametreleriyle ilgili ek bilgi için [DEFINE COMMINFO](#) başlıklı konuya bakın.

Çoklu yayın için bir COMMINFO nesnesi tanımlamak üzere aşağıdaki komut satırı örneğini kullanın:

```
DEFINE COMMINFO(MC1) GRPADDR(group address) PORT(port number)
```

Burada *MC1* COMMINFO nesnenizin adıdır, *grup adresi* grubunuz çok hedefli IP adresi ya da DNS adıdır ve *kapı numarası* iletimin açık olduğu kapıdır (Varsayılan değer 1414 'tür).

MC1 adlı yeni bir COMMINFO nesnesi yaratılır; bu ad, sonraki örnekte bir TOPIC nesnesi tanımlarken belirtmeniz gereken addır.

Çoklu yayın için TOPIC nesnesi oluşturma

Konu, yayınlama/abone olma iletilisinde yayınlanan bilgilerin konusudur ve konu, bir TOPIC nesnesi yaratılarak tanımlanır. TOPIC nesnelerinin, çok hedefli kullanılıp kullanılmayacağını tanımlayan iki parametresi vardır. Bu parametreler: **COMMINFO** ve **MCAST**.

- **COMMINFO** Bu değiştirge, çok hedefli iletişim bilgileri nesnesinin adını belirtir. COMMINFO nesne parametreleriyle ilgili ek bilgi için [DEFINE COMMINFO](#) başlıklı konuya bakın.
- **MCAST** Bu parametre, konu ağacında bu konumda çoklu yayına izin verilip verilmeyeceğini belirtir.

Çoklu yayın için bir TOPIC nesnesi tanımlamak üzere aşağıdaki komut satırı örneğini kullanın:

```
DEFINE TOPIC(ALLSPORTS) TOPICSTR('Sports') COMMINFO(MC1) MCAST(ENABLED)
```

ALLSPORTS adlı yeni bir TOPIC nesnesi yaratılır. Sports konu dizisi vardır, ilgili iletişim bilgileri nesnesi MC1 olarak adlandırılır (önceki örnekte COMMINFO nesnesi tanımlanırken belirttiğiniz ad) ve çoklu yayın etkinleştirilir.

Çok hedefli yayınlama/abone olma sınaması

TOPIC ve COMMINFO nesneleri yaratıldıktan sonra, bunlar amqsubc örneği ve amqssubc örneği kullanılarak sınanabilir. Bu örneklerle ilgili daha fazla bilgi için bkz. [Yayınlama/Abone Olma örnek programları](#).

1. İki komut satırı penceresi açın; ilk komut satırı amqsubc yayınlama örneği içindir ve ikinci komut satırı amqssubc abone olma örneği içindir.
2. Komut satırı 1 'de aşağıdaki komutu girin:

```
amqsubc Sports QM1
```

Burada Spor , daha önceki bir örnekte tanımlanan TOPIC nesnesinin konu dizisidir ve QM1 kuyruk yöneticisinin adıdır.

3. Komut satırı 2 'de aşağıdaki komutu girin:

```
amqssubc Sports QM1
```

Burada Sports ve QM1 , adım "2" sayfa 376' de kullanımıyla aynıdır.

4. Komut satırı 1 'de Hello world yazın. COMMINFO nesnesinde belirtilen kapı ve IP adresi doğru yapılandırıldıysa; belirtilen adresteki yayınlar için kapıda dinleme yapan amqssubc örneği Hello world komut satırı 2 'de çıkar.

IBM MQ Çok hedefli konu topolojisi

IBM MQ Çok hedefli konu topolojisini anlamak için bu örneği kullanın.

IBM MQ Çoklu yayın desteği, her alt ağacın toplam sıradüzeni içinde kendi çoklu yayın grubuna ve veri akışına sahip olmasını gerektirir.

Sınıf ağ IP adresleme şemasında, çok hedefli adres için belirlenmiş adres alanı vardır. IP adresinin tam çoklu yayın aralığı 224.0.0.0 - 239.255.255.255 arasındadır, ancak bu adreslerden bazıları ayrılmıştır. Ayrılmış adreslerin listesi için sistem yöneticinize başvurun ya da daha fazla bilgi için bkz. <https://www.iana.org/assignments/multicast-addresses> . 239.0.0.0 - 239.255.255.255 aralığında yerel kapsamlı çoklu yayın adresini kullanmanız önerilir.

Aşağıdaki çizgede, iki olası çoklu yayın veri akışı vardır:

```
DEF COMMINFO(MC1) GRPADDR(239.XXX.XXX.XXX
)
```

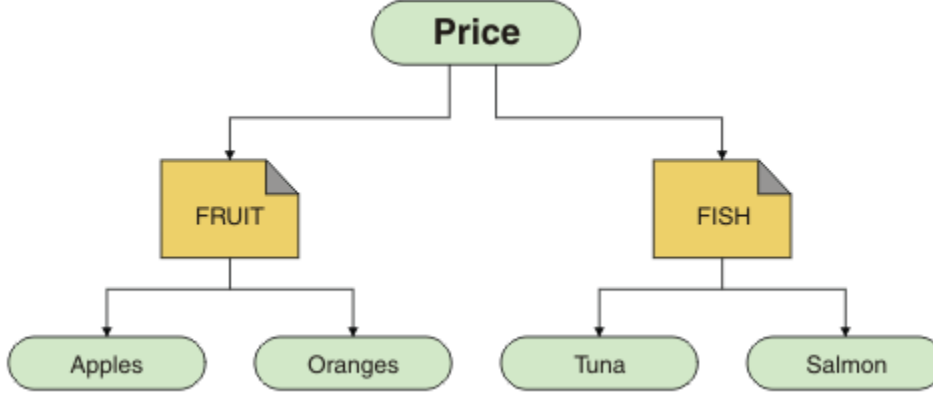


```
DEF COMMINFO(MC2) GRPADDR(239.YYY.YYY.YYY)
```

burada 239.XXX.XXX.XXX ve 239.YYY.YYY.YYY geçerli çoklu yayın adresleridir.

Bu konu tanımlamaları, aşağıdaki çizgede gösterildiği gibi bir konu ağacı yaratmak için kullanılır:

```
DEFINE TOPIC(FRUIT) TOPICSTRING('Price/FRUIT') MCAST(ENABLED) COMMINFO(MC1)  
DEFINE TOPIC(FISH) TOPICSTRING('Price/FISH') MCAST(ENABLED) COMMINFO(MC2)
```



Her çok hedefli iletişim bilgisi (COMMINFO) nesnesi, grup adresleri farklı olduğu için farklı bir veri akışını temsil eder. Bu örnekte FRUIT konusu COMMINFO nesnesini kullanmak üzere tanımlanmıştır MC1 , FISH konusu COMMINFO nesnesini kullanmak için tanımlanmıştır MC2 ve Price düğümünün çoklu yayın tanımlamaları yoktur.

IBM MQ Çoklu yayın, konu dizgileri için 255 karakter sınırına sahiptir. Bu sınırlama, ağaçtaki düğümlerin ve yaprak düğümlerin adlarıyla dikkat edilmesi gerektiği anlamına gelir; düğümlerin ve yaprak düğümlerin adları çok uzunsa, konu dizgisi 255 karakteri aşabilir ve [2425 \(0979\) \(RC2425\): MQRC_TOPIC_STRING_ERROR](#) neden kodunu döndürür. Daha uzun konu dizelerinin performans üzerinde zararlı bir etkisi olabileceğinden, konu dizgilerinin mümkün olduğunca kısa olması önerilir.

Çok hedefli iletilerin boyutunun denetlenmesi

IBM MQ ileti biçimi hakkında bilgi edinmek ve IBM MQ iletilerinin boyutunu azaltmak için bu bilgileri kullanın.

IBM MQ iletileriyle ilişkilendirilmiş, ileti tanımlayıcısında bulunan bazı öznitelikler vardır. Küçük iletiler için, bu öznitelikler veri trafiğinin çoğunu temsil edebilir ve iletim hızı üzerinde önemli bir zararlı etkisi olabilir. IBM MQ Multicast, kullanıcının bu özniteliklerden hangilerinin iletiyle birlikte iletildiğini yapılandırmasına olanak sağlar.

Konu dizesi dışındaki ileti özniteliklerinin varlığı, COMMINFO nesnesinin gönderilmesi gerekip gerekmediğine bağlıdır. Bir öznitelik iletilmezse, alan uygulama varsayılan bir değer uygular. Varsayılan MQMD değerleri her zaman MQMD_DEFAULT değeriyle aynı değildir ve [Çizelge 19 sayfa 378](#) içinde açıklanmıştır.

COMMINFO nesnesi, iletiyle birlikte MQMD alanlarının ve kullanıcı özelliklerinin kaç tanesinin akacağını denetleyen MCPROP özniteliğini içerir. Bu özniteliğin değerini uygun bir düzeye ayarlayarak, IBM MQ Çoklu yayın iletilerinin boyutunu denetleyebilirsiniz:

MCPROP

Çok hedefli özellikler, iletiyle birlikte MQMD özelliklerinin ve kullanıcı özelliklerinin kaç tanesinin akışını denetler.

TÜMÜ

Tüm kullanıcı özellikleri ve MQMD ' nin tüm alanları iletilir.

Yanıt

Yalnızca kullanıcı özellikleri ve iletilerin yanıtlanmasıyla ilgili MQMD alanları iletilir. Bu özellikler şunlardır:

- MsgType
- MessageId
- CorrelId
- ReplyToQ
- ReplyToQmgr

KULLANICI



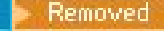
Yalnızca kullanıcı özellikleri iletilir.

YOK

Hiçbir kullanıcı özelliği ya da MQMD alanı iletilmez.

ŞİRKET

Bu değer, iletinin RMMuyumlu kipte iletilmesine neden olur; bu, yürürlükteki XMS uygulamaları ve IBM Integration Bus RMM uygulamalarıyla birlikte bazı işlemler gerçekleştirilmesine olanak sağlar.

   XMS .NET Çok hedefli ileti sistemi (RMMkullanılarak) IBM MQ 9.2 ' den kullanımdan kaldırılmıştır ve şu adresten kaldırılmıştır: IBM MQ 9.3.

Çok hedefli ileti öznitelikleri

İleti öznitelikleri, MQMD, MQRFH2içindeki alanlar ve ileti özellikleri gibi çeşitli yerlerden gelebilir.

Aşağıdaki tablo, iletiler MCPROP (bu bölümde daha önce açıklanmıştır) değerine bağlı olarak gönderildiğinde ne olduğunu ve bir öznitelik gönderilmediğinde kullanılan varsayılan değeri gösterir.

Öznitelik	Çoklu yayın kullanırken yapılacak işlem	İletilemezse varsayılan değer
TopicString	Her Zaman Dahil Edilen	Geçerli değildir
MQMQ StrucId	İletilenmedi	Geçerli değildir
MQMD Sürümü	İletilenmedi	Geçerli değildir
Rapor	Varsayılan değilse dahil edilir	0
MsgType	Varsayılan değilse dahil edilir	MQMT_DATAGRAM
Son kullanma tarihi	Varsayılan değilse dahil edilir	0
Geri bildirim	Varsayılan değilse dahil edilir	0
Kodlama	Varsayılan değilse dahil edilir	MQENC_NORMAL (eşdeğeri)
CodedCharSetId	Varsayılan değilse dahil edilir	1208
Biçim	Varsayılan değilse dahil edilir	MQRFH2
Öncelik	Varsayılan değilse dahil edilir	4
Kalıcılık	Varsayılan değilse dahil edilir	MQPER_NOT_KALICI
MsgId	Varsayılan değilse dahil edilir	Boş Değerli
CorrelId	Varsayılan değilse dahil edilir	Boş Değerli
BackoutCount	Varsayılan değilse dahil edilir	0
ReplyToQ	Varsayılan değilse dahil edilir	Boş

Çizelge 19. İleti sistemi öznelikleri ve bunların çoklu yayınlı ilişkisi (devamı var)

Öznelik	Çoklu yayın kullanırken yapılacak işlem	İletilemezse varsayılan değer
ReplyToQMgr	Varsayılan değilse dahil edilir	Boş
UserIdentifier	Varsayılan değilse dahil edilir	Boş
AccountingToken	Varsayılan değilse dahil edilir	Boş Değerli
PutAppIType	Varsayılan değilse dahil edilir	MQAT_JAVA
PutAppIName	Varsayılan değilse dahil edilir	Boş
PutDate	Varsayılan değilse dahil edilir	Boş
PutTime	Varsayılan değilse dahil edilir	Boş
ApplOriginVerileri	Varsayılan değilse dahil edilir	Boş
GroupID	Çıkarıldı	Geçerli değildir
MsgSeqNumarası	Çıkarıldı	Geçerli değildir
Görelî Konum	Çıkarıldı	Geçerli değildir
MsgFlags	Çıkarıldı	Geçerli değildir
OriginalLength	Çıkarıldı	Geçerli değildir
UserProperties	Dahildir	Geçerli değildir

İlgili başvurular

Multi ALTER COMMINFO (COMMINFO)

COMMINFO TANIMLAYIN

Çoklu yayın ileti sistemi için veri dönüştürmeyi etkinleştirme

Veri dönüştürmenin IBM MQ Multicast ileti sistemi için nasıl çalıştığını anlamak için bu bilgileri kullanın.

IBM MQ Multicast, paylaşılan, bağlantısız bir iletişim kuralıdır ve her bir istemcinin veri dönüştürme için belirli isteklerde bulunması mümkün değildir. Aynı çoklu yayın akımına abone olan her istemci aynı ikili verileri alır; bu nedenle, IBM MQ veri dönüştürmesi gerekiyorsa, dönüştürme her istemcide yerel olarak gerçekleştirilir.

Karma bir platform kuruluşunda, istemcilerin çoğu verileri, iletme uygulamasının yerel biçimi olmayan bir biçimde gerektirebilir. Bu durumda, verimlilik için ileti iletiminin kodlamasını tanımlamak üzere çok hedefli COMMINFO nesnesinin **CCSID** ve **ENCODING** değerleri kullanılabilir.

IBM MQ Multicast, aşağıdaki yerleşik biçimler için ileti bilgi yükünün veri dönüşümünü destekler:

- MQADMIN
- MQEVENT
- MQPCF
- MQRFH
- MQRFH2
- MQSTR

Bu biçimlere ek olarak, kendi biçimlerinizi de tanımlayabilir ve MQDXP-Data-conversion exit parameter veri dönüştürme çıkışı kullanabilirsiniz.

Programlama verileri dönüştürmeleri hakkında bilgi için bkz. Çok hedefli ileti sistemi için MQI 'da veri dönüştürme.

Veri dönüştürme hakkında daha fazla bilgi için bkz. [Veri dönüştürme](#).

Veri dönüştürme çıktıları ve ClientExitPath ile ilgili daha fazla bilgi için bkz. [ClientExitIstemci yapılandırma dosyasının yol kısmı](#).

Çok hedefli uygulama izleme

IBM MQ Multicast olanağını yönetme ve izleme hakkında bilgi edinmek için bu bilgileri kullanın.

Çok hedefli trafiğe ilişkin geçerli yayıncıların ve abonelerin durumu (örneğin, gönderilen ve alınan iletilerin sayısı ya da kaybedilen iletilerin sayısı) belirli aralıklarla istemciden sunucuya iletilir. Durum alındığında, COMMINFO nesnesinin [COMMEV](#) özniteliği, kuyruk yöneticisinin SYSTEM.ADMIN.PUBSUB.EVENT. Olay ileti, alınan durum bilgilerini içerir. Bu bilgi, bir sorunun kaynağını bulmada paha biçilemez bir tanılama yardımcısıdır.

Kuyruk yöneticisine bağlı uygulamalara ilişkin bağlantı bilgilerini görüntülemek için **DISPLAY CONN** MQSC komutunu kullanın. **DISPLAY CONN** komutuna ilişkin ek bilgi için [DISPLAY CONN](#) başlıklı konuya bakın.

Yayıncılarınızın ve abonelerinizin durumunu görüntülemek için **DISPLAY TPSTATUS** MQSC komutunu kullanın. **DISPLAY TPSTATUS** komutuna ilişkin ek bilgi için [DISPLAY TPSTATUS](#) başlıklı konuya bakın.

COMMEV ve çok hedefli ileti güvenilirliği göstergesi

COMMINFO nesnesinin **COMMEV** özniteliğiyle birlikte kullanılan *güvenilirlik göstergesi*, IBM MQ Multicast yayıncıları ve abonelerinin izlenmesinde önemli bir öğedir. Güvenilirlik göstergesi (Yayınla ya da Abone Ol durum komutlarında döndürülen **MSGREL** alanı), hata olmayan iletimlerin yüzdesini gösteren bir IBM MQ göstergesidir. Bazen iletilerin, **MSGREL** değerine yansıyan bir iletim hatası nedeniyle yeniden iletilmesi gerekir. İletim hatalarının olası nedenleri arasında yavaş aboneler, meşgul ağlar ve ağ kesintileri yer alır. **COMMEV**, COMMINFO nesnesi kullanılarak yaratılan çoklu yayın tanıtıcıları için olay iletilerinin oluşturulup oluşturulmayacağını ve üç olası değerden birine ayarlanıp ayarlanmayacağını denetler:

DEVRE DIŞI

Olay iletileri yazılmaz.

Etkinleştirildi

Olay iletileri her zaman, COMMINFO **MONINT** parametresinde tanımlanan bir sıklıkla yazılır.

KURAL DIŞI DURUM

İleti güvenilirliği güvenilirlik eşliğinin altındaysa olay iletileri yazılır. %90 ya da daha düşük bir ileti güvenilirlik düzeyi, ağ yapılandırmasında bir sorun olabileceğini ya da Yayınla/Abone Ol uygulamalarından birinin ya da daha fazlasının çok yavaş çalıştığını gösterir:

- **MSGREL (100, 100)** değeri, kısa vadede ya da uzun vadede herhangi bir sorun olmadığını gösterir.
- **MSGREL (80, 60)** değeri, iletilerin %20 'sinin şu anda sorun yaşadığını, ancak 60 'ın uzun vadeli değerinde de bir iyileşme olduğunu gösterir.

Kuyruk yöneticisine tek hedefli bağlantı kesildiğinde bile, istemciler çok hedefli trafiği iletip almaya devam edebilir, bu nedenle veriler güncel olmayabilir.

Çok hedefli ileti güvenilirliği

IBM MQ Multicast aboneliği ve ileti geçmişi nasıl ayarlayacağınızı öğrenmek için bu bilgileri kullanın.

Çok hedefli iletim hatasının üstesinden gelmenin önemli bir öğesi, IBM MQ tarafından iletilen verilerin (bağlantının iletim ucunda tutulacak iletilerin geçmişi) arabelleğe alınmasıdır. Bu işlem, IBM MQ güvenilirliği sağladığından, koyma uygulaması işleminde iletilerin arabelleğe alınması gerekmediği anlamına gelir. Bu geçmişi büyüklüğü, aşağıdaki bilgilerde açıklandığı gibi, iletişim bilgileri (COMMINFO) nesnesiyle yapılandırılır. Daha büyük bir iletim arabelleği, gerekirse yeniden iletilecek daha fazla iletim geçmişi olduğu anlamına gelir, ancak çoklu yayın doğası nedeniyle %100 garantili teslim desteklenemez.

IBM MQ Multicast ileti geçmişi, iletişim bilgileri (COMMINFO) nesnesinde **MSGHIST** özniteliğiyle denetlenir:

MSGHIST

Bu deęer, NACKs (olumsuz alındı bildirimleri) durumunda yeniden iletimleri işlemek için sistem tarafından tutulan kilobayt cinsinden ileti geęmiři miktarıdır.

0 deęeri, en düşük güvenilirlik düzeyini verir. Varsayılan deęer 100 KB 'dir.

IBM MQ Çok hedefli yeni abonelik geęmiři, iletişim bilgileri (COMMINFO) nesnesinde **NSUBHIST** öznitelięiyle denetlenir:

NSUBHIST

Yeni abone geęmiři, bir yayın akıřına katılan bir abonenin řu anda kullanılabilir olduęu kadar fazla veri alıp almayacaęını ya da yalnızca abonelik zamanından itibaren yapılan yayınları alıp almayacaęını denetler.

YOK

NONE deęeri, ileticinin yalnızca abonelik zamanından itibaren yayınlanan yayınları ilemesine neden olur. NONE varsayılan deęerdir.

TÜMÜ

ALL deęeri, ileticinin konuya iliřkin bilinen çok fazla geęmiři yeniden ilemesine neden olur. Bazı durumlarda bu durum, tutulan yayınlara benzer bir davranıř verebilir.

Not: ALL deęerinin kullanılması, tüm konu geęmiři yeniden iletiildięi için büyük bir konu geęmiři varsa, performans üzerinde zararlı bir etkiye sahip olabilir.

İlgili bařvurular

[COMMINFO TANıMLAYIN](#)

Multi

[ALTER COMMINFO \(COMMINFO\)](#)

Geliřmiř çoklu yayın görevleri

.ini dosyalarını yapılandırma ve IBM MQ LLM ile birlikte çalışabilirlik gibi geliřmiř IBM MQ Çok hedefli yönetim görevleri hakkında bilgi edinmek için bu bilgileri kullanın.

Çoklu yayın kuruluşunda güvenlikle ilgili dikkat edilmesi gereken noktalar için bkz. [Çok hedefli güvenlik](#).

Çok hedefli ve çok hedefli olmayan yayınlama/abone olma etki alanları arasında köprü

Çok hedefli olmayan bir yayınlama IBM MQ Çok hedefli etkin bir konuda yayınladığında ne olacaęını anlamak için bu bilgileri kullanın.

Çok hedefli olmayan bir yayınlama **MCAST** etkin ve **BRIDGE** etkin olarak tanımlanan bir konuyu yayınlarsa, kuyruk yöneticisi iletiyi çoklu yayın üzerinden doğrudan dinleyen abonelere iletir. Çok hedefli bir yayınlama, çok hedefli olmayan konulara yayınlamaz.

Var olan konular, bir konu nesnesinin **MCAST** ve **COMMINFO** parametreleri ayarlanarak çok hedefli olarak etkinleştirilebilir. Bu deęiřtirgelerle ilgili ek bilgi için [İlk çoklu yayın kavramları](#) başlıklı konuya bakın.

COMMINFO nesnesi **BRIDGE** öznitelięi, çoklu yayın kullanmayan uygulamalardaki yayınları denetler. **BRIDGE** ENABLED olarak ayarlanırsa ve konunun **MCAST** parametresi de ENABLED olarak ayarlanırsa, çoklu yayın kullanmayan uygulamalardaki yayınlar, bunu yapan uygulamalara köprülenmiř olur. **BRIDGE** deęiřtirgesiyle ilgili ek bilgi için [DEFINE COMMINFO](#) başlıklı konuya bakın.

.ini dosyalarını Multicast için yapılandırma

.ini dosyalarındaki IBM MQ Multicast alanlarını anlamak için bu bilgileri kullanın.

Bir ini dosyasında ek IBM MQ Multicast yapılandırması yapılabilir. Kullanmanız gereken belirli ini dosyası, uygulamaların tipine baęlıdır:

- İstemci: `MQ_DATA_PATH/mqclient.ini` dosyasını yapılandırın.
- Kuyruk yöneticisi: `MQ_DATA_PATH/qmgrs/QMNAME/qm.ini` dosyasını yapılandırın.

Burada `MQ_DATA_PATH` , IBM MQ veri dizininin yeri (`/var/mqm/mqclient.ini`) ve `QMNAME` , `.ini` dosyasının geçerli olduğu kuyruk yöneticisinin adıdır.

`.ini` dosyası, IBM MQ Çok hedefli: davranışını ayarlamak için kullanılan alanları içerir.

```
Multicast:
Protocol      = IP | UDP
IPVersion     = IPv4 | IPv6 | ANY | BOTH
LimitTransRate = DISABLED | STATIC | DYNAMIC
TransRateLimit = 100000
SocketTTL     = 1
Batch         = NO
Loop          = 1
Interface     = <IPAddress>
FeedbackMode  = ACK | NACK | WAIT1
HeartbeatTimeout = 20000
HeartbeatInterval = 2000
```

Protokol

UDP

Bu kipte, paketler UDP protokolü kullanılarak gönderilir. Ancak, IP kipinde olduğu gibi, ağ öğeleri çoklu yayın dağıtımında yardım sağlayamaz. Paket biçimi PGM ile uyumlu olmaya devam eder. Bu varsayılan değerdir.

IP

Bu kipte, verici işlenmemiş IP paketleri gönderir. PGM desteğine sahip ağ öğeleri, güvenilir çok hedefli paket dağıtımında yardımcı olur. Bu kip, PGM standardıyla tam olarak uyumludur.

IPVersion

IPv4

Yalnızca IPv4 iletişim kuralını kullanarak iletişim kurun. Bu varsayılan değerdir.

IPv6

Yalnızca IPv6 iletişim kuralını kullanarak iletişim kurun.

Fark Etmez

Hangi iletişim kuralının kullanılabilir olduğuna bağlı olarak IPv4, IPv6ya da her ikisini kullanarak iletişim kurun.

HER İKİSİ

IPv4 ve IPv6 kullanarak iletişimi destekler.

LimitTransHızı

DEVRE DIŞI

İletim hızı kontrolü yok. Bu varsayılan değerdir.

STATIC

Statik iletim hızı denetimini uygular. Verici, TransRateLimit parametresi tarafından belirlenen hızı aşan bir hızda iletmez.

Devingen

Verici, alıcılardan aldığı geribildirime göre iletim hızını uyarlar. Bu durumda iletim hızı sınırı, TransRateLimit parametresi tarafından belirlenen değerden fazla olamaz. Verici en uygun iletim hızına ulaşmaya çalışıyor.

TransRateSınırı

Kbps olarak iletim hızı sınırı.

SocketTTL

SocketTTL değeri, çok hedefli trafiğin bir yönlendiriciden geçip geçemeyeceğini ya da geçebildiği yöneltici sayısını belirler.

Toplu

İletilerin toplu olarak mı gönderileceğini yoksa hemen mi gönderileceğini denetler 2 olası değer vardır:

- *HAYIR* İletiler toplu olarak gönderilmez, hemen gönderilir.

- *EVET* İletiler toplu olarak gönderiliyor.

Döngü

Çoklu yayın döngüsünü etkinleştirmek için değeri 1 olarak ayarlayın. Çoklu yayın döngüsü, gönderilen verilerin anasisteme geri dönüp dönmeyeceğini tanımlar.

Arabirim

Çok hedefli trafiğin aktığı arabirimin IP adresi. Daha fazla bilgi ve sorun giderme için bkz: [Birden çok noktaya yayın olmayan bir ağda çoklu yayın uygulamalarının test edilmesi](#) ve [Çok noktaya yayın trafiği için uygun ağın ayarlanması](#)

FeedbackMode

NACK.

Olumsuz onaylarla geri bildirim. Bu varsayılan değerdir.

ACK.

Olumlu onaylarla geri bildirim.

WAIT1

Alıcının herhangi bir alıcıdan yalnızca 1 ACK beklediği pozitif onaylarla geri bildirim.

HeartbeatTimeout

Milisaniye cinsinden sağlıklı işletim bildirim zaman aşımı. 0 değeri, sağlıklı işletim bildirim zaman aşımı olaylarının konunun alıcısı ya da alıcıları tarafından yükseltilmediğini gösterir. Varsayılan değer 20000 'dir.

HeartbeatInterval

Milisaniye cinsinden sağlıklı işletim bildirim aralığı. 0 değeri, sağlıklı işletim bildirim gönderilmediğini gösterir. Yanlış sağlıklı işletim bildirim zaman aşımı olaylarını önlemek için sağlıklı işletim bildirim aralığı **HeartbeatTimeout** değerinden önemli ölçüde küçük olmalıdır. Varsayılan değer 2000 'dir.

IBM MQ Düşük Gecikmeli İleti Sistemi ile çok hedefli birlikte çalışabilirlik

IBM MQ Multicast ve IBM MQ Low Latency Messaging (LLM) arasındaki birlikte çalışabilirliği anlamak için bu bilgileri kullanın.

Temel bilgi yükü aktarımı, her iki yönde de ileti değiş tokuşu için çoklu yayın kullanan başka bir uygulama ile LLM kullanan bir uygulama için mümkündür. Çoklu yayın LLM teknolojisini kullansa da, LLM ürününün kendisi yerleşik değildir. Bu nedenle, hem LLM hem de IBM MQ Multicast ürününü kurmak ve bu iki ürüne ayrı ayrı hizmet vermek mümkündür.

Çoklu yayın ile iletişim kuran LLM uygulamalarının ileti özelliklerini göndermesi ve alması gerekebilir. IBM MQ ileti özellikleri ve MQMD alanları, aşağıdaki çizelgede gösterildiği gibi belirli LLM iletileri özellik kodlarına sahip LLM ileti özellikleri olarak iletilir:

<i>Çizelge 20. IBM MQ ileti özellikleri- IBM MQ LLM özellik eşlemeleri</i>			
IBM MQ özellik	IBM MQ LLM özellik tipi	LLM özellik türü	LLM özellik kodu
MQMD.Report	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1001
MQMD.MsgType	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1002
MQMD.Expiry	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1003
MQMD.Feedback	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1004
MQMD.Encoding	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1005
MQMD.CodedCharSetId	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1006
MQMD.Format	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1007
MQMD.Priority	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1008
MQMD.Persistence	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1009

Çizelge 20. IBM MQ ileti özellikleri- IBM MQ LLM özellik eşlemeleri (devamı var)

IBM MQ özellik	IBM MQ LLM özellik tipi	LLM özellik türü	LLM özellik kodu
MQMD.MsgId	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_ByteArray	-1010
MQMD.BackoutCount	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1012
MQMD.ReplyToQ	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1013.
MQMD.ReplyToQMger	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1014
MQMD.PutDate	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1020
MQMD.PutTime	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1021
MQMD.ApplOriginData	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1022
MQPubOptions	RMM_MSG_PROP_INT32	LLM_PROP_KIND_int32	-1053.

LLM ile ilgili ek bilgi için LLM ürün belgelerine bakın: [IBM MQ Low Latency Messaging](#).

IBM i Yönetme IBM MQ for IBM i

CL komutları, IBM üzerinde IBM MQ ' i yönetmek için tercih edilen yöntemdir. MQSC komutlarını, PCF komutlarını, denetim komutlarını ve uzak denetimi de kullanabilirsiniz.

Bu görev hakkında

Yönetim görevleri arasında kümelerin, süreçlerin ve IBM MQ nesnelерinin yaratılması, başlatılması, değiştirilmesi, görüntülenmesi, durdurulması ve silinmesi (kuyruk yöneticileri, kuyruklar, ad alanları, süreç tanımlamaları, kanallar, istemci bağlantısı kanalları, dinleyiciler, hizmetler ve kimlik doğrulama bilgileri nesneleri) yer alır.

IBM MQ for IBM i' in yönetilmesine ilişkin ayrıntılar için aşağıdaki bağlantılara bakın.

İlgili kavramlar

[IBM MQ for IBM i kuyruk yöneticisi kitaplık adlarını anlama](#)

[IBM i üzerine kurulabilir hizmetler ve bileşenler](#)

İlgili görevler

[Çoklu Platformlar üzerinde IBM MQ yapılandırma bilgilerinin değiştirilmesi](#)

[IBM i üzerinde güvenliğin ayarlanması](#)

[“IBM i üzerinde gönderilmeyen ileti kuyruğu işleyicisinin çağırılması” sayfa 183](#)

IBM MQ for IBM i' da, **STRMQMDLQ** komutunu ayarlayarak DLQ işleyicisini çağırıyorsunuz.

[IBM MQ for IBM i uygulamalarıyla ilgili sorunların belirlenmesi](#)

İlgili başvurular

[Sistem ve varsayılan nesnelер](#)

IBM i CL komutlarını kullanarak IBM MQ for IBM i yönetimi

IBM MQ IBM i komutlarını anlamak için bu bilgileri kullanın.

Kuyruk yöneticileri, kuyruklar, konular, kanallar, ad listesi, süreç tanımlamaları ve kimlik doğrulama bilgileri nesneleri de içinde olmak üzere, IBM MQ komut gruplarının çoğuna ilgili **WRK*** komutu kullanılarak erişilebilir.

Kümedeki asıl komut **WRKMQM**. Bu komut, örneğin, durum bilgileriyle birlikte sistemdeki tüm kuyruk yöneticilerinin listesini görüntülemenizi sağlar. Diğer bir yöntem olarak, her bir girişe ilişkin çeşitli seçenekleri kullanarak kuyruk yöneticisine özgü tüm komutları işleyebilirsiniz.

WRKMQM komutundan, her kuyruk yöneticisinin belirli alanlarını seçebilirsiniz; örneğin, kanallarla, konularla ya da kuyruklarla çalışma ve buradan tek tek nesneleri seçebilirsiniz.

IBM MQ uygulama tanımlarının kaydedilmesi

IBM MQ uygulamalarını oluşturduğunuzda ya da özelleştirdiğinizde, oluşturulan tüm IBM MQ tanımlarının kaydını tutmak yararlı olur. Bu kayıt aşağıdakiler için kullanılabilir:

- Kurtarma amaçları
- Bakım
- IBM MQ uygulamaları kullanıma alınıyor

IBM MQ uygulama tanımlarını 1/2 yöntemle kaydedebilirsiniz:

1. Sunucuya ilişkin IBM MQ tanımlamalarınızı oluşturmak için CL programları yaratılıyor.
2. IBM MQ tanımlamalarınızı altyapılar arası IBM MQ komut dilini kullanarak oluşturmak için SRC üyeleri olarak MQSC metin dosyaları yaratılıyor.

Kuyruk nesnelerinin tanımlanmasıyla ilgili daha fazla ayrıntı için bkz. [“IBM MQ ' nin MQSC komutlarını kullanarak denetlenmesi” sayfa 12](#) ve [“IBM MQ Programlanabilir Komut Biçimlerini Kullanma” sayfa 26](#).

İlgili başvurular

[IBM MQ for IBM i CL komutları başvurusu](#)

IBM i CL komutlarını kullanarak IBM MQ for IBM i komutunu kullanmaya başlamadan önce

IBM MQ altsistemini başlatmak ve yerel bir kuyruk yöneticisi yaratmak için bu bilgileri kullanın.

Başlamadan önce

IBM MQ altsistemini çalıştırdığınızdan (STRSBS QMQM/QMQM komutunu kullanarak) ve bu altsistemle ilişkili iş kuyruğunun tutulmadığından emin olun. Varsayılan olarak, IBM MQ altsistem ve iş kuyruğunun her ikisi de QMQM kitaplığında QMQM olarak adlandırılır.

Bu görev hakkında

Kuyruk yöneticisini başlatmak için IBM i komut satırının kullanılması

Yordam

1. Bir IBM i komut satırından CRTMQM komutunu çalıştırarak yerel bir kuyruk yöneticisi yaratın.
Bir kuyruk yöneticisi yarattığınızda, o kuyruk yöneticisini varsayılan kuyruk yöneticisi yapma seçeneğiniz vardır. Varsayılan kuyruk yöneticisi (yalnızca biri olabilir), kuyruk yöneticisi adı değişikliği (MQMNAME) atlanırsa, CL komutunun uygulandığı kuyruk yöneticisidir.
2. Bir IBM i komut satırından STRMQM komutunu çalıştırarak yerel bir kuyruk yöneticisini başlatın.
Kuyruk yöneticisi başlatma işlemi birkaç saniyeden uzun sürerse IBM MQ , başlatma işleminin ilerleyişini aralıklı olarak ayrıntılı olarak gösteren durum iletileri görüntüler. Bu iletilere ilişkin ek bilgi için [İletiler ve neden kodları](#) konusuna bakın.

Sonraki adım

Bir kuyruk yöneticisini durdurmak için, IBM i komut satırından ENDMQM komutunu verebilir ve IBM i komut satırından diğer IBM MQ komutlarını vererek bir kuyruk yöneticisini denetleyebilirsiniz.

Uzak kuyruk yöneticileri uzaktan başlatılamaz, ancak yerel işlemler tarafından kendi sistemlerinde yaratılmalı ve başlatılmalıdır. Bu tür işlemleri etkinleştirmek için uzak işletim olanaklarının (IBM MQ for IBM i'de) bulunduğu bir kural dışı durum vardır.

Yerel kuyruk yöneticisi uzak bir kuyruk yöneticisini durduramıyor.

Not: IBM MQ sistemini susturmanın bir parçası olarak etkin kuyruk yöneticilerini susturmanız gerekir. Bu, [“IBM MQ for IBM i ' i susturma” sayfa 453](#) içinde açıklanmıştır.

IBM için IBM MQ nesneleri oluşturmaya ilişkin yöntemleri anlamak için bu bilgileri kullanın.

Başlamadan önce

Aşağıdaki görevler, komut satırından IBM MQ for IBM i ' i kullanabilmeniz için çeşitli yollar önerir.

Bu görev hakkında

IBM MQ nesneleri oluşturmak için iki çevrimiçi yöntem vardır:

Yordam

1. Create komutunu kullanarak, örneğin: **Create MQM Queue** komutu: **CRTMQMQ**
2. MQM nesnesiyle çalış komutunu ve ardından F6komutunu kullanarak, örneğin: **Work with MQM Queues** komutu: **WRKMQM**

Sonraki adım

Tüm komutların bir listesi için bkz. [IBM MQ for IBM i CL komutları](#).

Not: Tüm MQM komutları, İleti Kuyruğu Yöneticisi Komutları menüsünden sunulabilir. Bu menüyü görüntülemek için komut satırına GO CMDMQM yazın ve Enter tuşuna basın.

Bu menüden bir komut seçtiğinizde, sistem bilgi istemi panosunu otomatik olarak görüntüler. Komut satırına doğrudan yazdığınız bir komuta ilişkin bilgi istemi panosunu görüntülemek için Enter tuşuna basmadan önce F4 tuşuna basın.

CRTMQMQ komutunu kullanarak yerel kuyruk yaratılması

Yordam

1. Komut satırına CHGMQM yazın ve F4 tuşuna basın.
2. **MQM Kuyruğu Yarat panosunda**, Queue name alanında yaratmak istediğiniz kuyruğun adını yazın. Büyük ve küçük harf karışık bir ad belirtmek için adı tırnak içine almanız gerekir.
3. Queue type alanına *LCL yazın.
4. Varsayılan kuyruk yöneticisini kullanmıyorsanız, bir kuyruk yöneticisi adı belirtin ve Enter tuşuna basın. Yeni bir değerle değerlerin üzerine yazabilirsiniz. Diğer alanları görmek için ileri doğru kaydırın. Kümeler için kullanılan seçenekler, seçenekler listesinin sonunda yer alır.
5. Herhangi bir değeri değiştirdiğinizde, kuyruğu yaratmak için Enter tuşuna basın.

WRKMQM komutunu kullanarak yerel kuyruk yaratılması

Yordam

1. Komut satırına WRKMQM yazın.
2. Bir kuyruk yöneticisinin adını girin.
3. Bilgi istemi panosunu görüntülemek istiyorsanız F4tuşuna basın. Bilgi istemi panosu, soysal bir kuyruk adı ya da kuyruk tipi belirleyerek görüntülenen kuyruk sayısını azaltmak için kullanışlıdır.
4. Enter tuşuna bastığınızda **MQM Kuyruklarıyla Çalışma panosu** görüntülenir. Yeni bir değerle değerlerin üzerine yazabilirsiniz. Diğer alanları görmek için ileri doğru kaydırın. Kümeler için kullanılan seçenekler, seçenekler listesinin sonunda yer alır.
5. Yeni bir kuyruk yaratmak için F6 tuşuna basın; bu işlem sizi **CRTMQMQ** panosuna götürür. Kuyruğun yaratılmasına ilişkin yönergeler için bkz. "[CRTMQMQ komutunu kullanarak yerel kuyruk yaratılması](#)" sayfa 386 . Kuyruğu yarattığınızda, **MQM Kuyruklarıyla Çalışma panosu** yeniden görüntülenir. F5=Refresh tuşuna bastığınızda yeni kuyruk listeye eklenir.

Kuyruk yöneticisi özneliklerini deęiřtirme

Bu görev hakkında

CHGMQM komutunda belirtilen kuyruk yöneticisinin özneliklerini, deęiřtirmek istedięiniz öznelikleri ve deęerleri belirterek deęiřtirmek için. Örneęin, `jupiter.queue.managerözneliklerini` deęiřtirmek için ařaęıdaki seęenekleri kullanın:

Yordam

Komut satırına **CHGMQM** yazın ve F4 tuřuna basın.

Sonuęlar

Komut, kullanılan kullanılmayan harf kuyruęunu deęiřtirir ve olayları engeller.

IBM i IBM i üzerinde yerel kuyruklarla ęalıřma

Bu bölümde, yerel kuyrukları yönetmek için kullanabileceęiniz bazı komutlara iliřkin örnekler yer alır. Gösterilen tüm komutlar, **WRKMQM komut panosundaki** seęenekler kullanılarak da kullanılabilir.

Yerel kuyruk tanımlanması

Bir uygulama için, yerel kuyruk yöneticisi uygulamanın baęlı olduęu kuyruk yöneticisidir. Yerel kuyruk yöneticisi tarafından yönetilen kuyrukların, o kuyruk yöneticisi için yerel olduęu söylenen kuyruklar.

Yerel bir kuyruk tanımlaması yaratmak ve kuyruk adı verilen veri yapısını yaratmak için **CRTMQMQ QTYPE *LCL** komutunu kullanın. Varsayılan yerel kuyruktaki kuyruk özelliklerini de deęiřtirebilirsiniz.

Bu örnekte, tanımladıęımız kuyruk (`orange.local.queue`) ařaęıdaki özelliklere sahip olacak řekilde belirtilir:

- Bu özellik, ilk giren ilk ıkar (FIFO) temelinde ulařır, giriřler için devre dıřı bırakılır ve alıřır.
- Bu bir *olaęan* kuyruktur; bařlatma kuyruęu ya da iletim kuyruęu deęildir ve tetikleyici iletileri üretmez.
- Kuyruk derinlięi üst sınırı 1000 iletidir; ileti uzunluęu üst sınırı 2000 bayttır.

Varsayılan kuyruk yöneticisinde ařaęıdaki komut bunu yapar:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL)
TEXT('Queue for messages from other systems')
PUTENBL(*NO)
GETENBL(*YES)
TRGENBL(*NO)
MSGDLYSEQ(*FIFO)
MAXDEPTH(1000)
MAXMSGLEN(2000)
USAGE(*NORMAL)
```

Not:

1. USAGE *NORMAL , bu kuyruęun bir iletim kuyruęu olmadıęını gösterir.
2. Aynı kuyruk yöneticisinde `orange.local.queue` adlı bir yerel kuyruęunuz varsa, bu komut bařarısız olur. Bir kuyruęun var olan tanımının üzerine yazılmasını istiyorsanız REPLACE *YES öznelięini kullanın, ancak ayrıca bkz. [“Yerel kuyruk özneliklerinin deęiřtirilmesi” sayfa 388.](#)

Bir teslim mektubu kuyruęunun tanımlanması

Doęru hedeflerine teslim edilemeyen iletilerin daha sonra alınmak üzere saklanabilmesi için, her kuyruk yöneticisinin kullanılmayan bir kuyruk olarak yerel bir kuyruęu olmalıdır. Kuyruk yöneticisine, gönderilmeyen iletiler kuyruęundan açıkça bahsetmeniz gerekir. Bunu, **CRTMQMQ** komutunda bir teslim edilmeyen ileti kuyruęu belirterek yapabilir ya da daha sonra bir tane belirlemek için **CHGMQM** komutunu kullanabilirsiniz. Kullanılmadan önce, teslim edilmeyen iletiler kuyruęunu da tanımlamanız gerekir.

Ürünle birlikte SYSTEM . DEAD . LETTER . QUEUE adlı örnek bir ileti kuyruğu sağlanır. Bu kuyruk, kuyruk yöneticisini yarattığınızda otomatik olarak yaratılır. Gerekirse bu tanımı değiştirebilirsiniz. Yeniden adlandırmanıza gerek yoktur, ancak isterseniz yeniden adlandırabilirsiniz.

Bir teslim mektubu kuyruğunun aşağıdakiler dışında özel bir gereksinimi yoktur:

- Yerel bir kuyruk olmalıdır.
- MAXMSGL (ileti uzunluğu üst sınırı) özniteliği, kuyruk yöneticisinin işlemesi gereken en büyük iletilerin **artı** teslim mektubu üstbilgisinin (MQDLH) büyüklüğünü kapsamasını sağlamalıdır.

IBM MQ , bir teslim edilmeyen ileti kuyruğunda bulunan iletilerin nasıl işleneceğini ya da kaldırılacağını belirlemenizi sağlayan bir ileti kuyruğu işleyicisi sağlar. Daha fazla bilgi için bkz. [“IBM i üzerinde gönderilmeyen ileti kuyruğu işleyicisinin çağırılması” sayfa 183.](#)

Varsayılan nesne özniteliklerinin görüntülenmesi

Bir IBM MQ nesnesi tanımladığınızda, varsayılan nesneden belirtmediğiniz öznitelikleri alır. Örneğin, bir yerel kuyruk tanımladığınızda, kuyruk tanımlamada atladığınız öznitelikleri SYSTEM . DEFAULT . LOCAL . QUEUE olarak adlandırılan varsayılan yerel kuyruktan devralır. Bu özniteliklerin tam olarak ne olduğunu görmek için aşağıdaki komutu kullanın:

```
DSPMQMQ QNAME(SYSTEM.DEFAULT.LOCAL.QUEUE) MQMNAME(MYQUEUEMANAGER)
```

Yerel kuyruk tanımlamasının kopyalanması

CPYMQMQ komutunu kullanarak bir kuyruk tanımlamasını kopyalayabilirsiniz. Örneğin:

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Bu komut, sistemin varsayılan yerel kuyruğuyla değil, özgün kuyruğumuzla orange . local . queue aynı özniteliklere sahip bir kuyruk yaratır.

Bir kuyruk tanımlamasını kopyalamak için **CPYMQMQ** komutunu kullanabilirsiniz, ancak özgün özniteliklerin yerine bir ya da daha çok değişiklik kullanabilirsiniz. Örneğin:

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('third.queue') MQMNAME(MYQUEUEMANAGER)  
MAXMSGLEN(1024)
```

Bu komut, orange . local . queue kuyruğunun özniteliklerini kuyruğa third . queue kopyalar, ancak yeni kuyruktaki ileti uzunluğu üst sınırının 2000 değil 1024 bayt olacağını belirtir.

Not: **CPYMQMQ** komutunu kullandığınızda, kuyruktaki iletileri değil, yalnızca kuyruk özniteliklerini kopyalayabilirsiniz.

Yerel kuyruk özniteliklerinin değiştirilmesi

CHGMQMQ komutunu ya da **CPYMQMQ** komutunu REPLACE *YES özniteliğiyle kullanarak kuyruk özniteliklerini iki şekilde değiştirebilirsiniz. [“Yerel kuyruk tanımlanması” sayfa 387](#) içinde, orange . local . queue kuyruğunu tanımladınız. Örneğin, bu kuyruktaki ileti uzunluğu üst sınırını 10.000 bayta yükseltmeniz gerekir.

- **CHGMQMQ** komutunu kullanarak:

```
CHGMQMQ QNAME('orange.local.queue') MQMNAME(MYQUEUEMANAGER) MAXMSGLEN(10000)
```

Bu komut, ileti uzunluğu üst sınırı olan tek bir özniteliği değiştirir; diğer tüm öznitelikler aynı kalır.

- **CRTMQMQ** komutunun REPLACE *YES seçeneğiyle kullanılması, örneğin:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL) MQMNAME(MYQUEUEMANAGER)
MAXMSGLEN(10000) REPLACE(*YES)
```

Bu komut yalnızca ileti uzunluğu üst sınırını değil, varsayılan değerleri verilen diğer tüm öznitelikleri de değiştirir. Kuyruk şimdi geçerli kılınırken, daha önce engellenmişti. Değiştirmedişiniz sürece, SYSTEM.DEFAULT.LOCAL.QUEUEkuyruğu tarafından belirlendiğinde, snr geçerli klnd § varsayılan değerlerdir.

Var olan bir kuyruқта ileti uzunluğu üst sınırını **azaltırsanız**, var olan iletiler etkilenmez. Ancak yeni iletiler, yeni ölçütleri karşılamalıdır.

Yerel kuyruğun temizlenmesi

magenta.queueadlı yerel bir kuyruktan tüm iletileri silmek için aşağıdaki komutu kullanın:

```
CLRMQMQ QNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Aşağıdaki durumda bir kuyruğu temizleyemezsiniz:

- Kuyruğa eşitleme noktası altında konan kesinleştirilmemiş iletiler var.
- Bir uygulamanın kuyruğu açık.

Yerel kuyruğun silinmesi

Yerel bir kuyruğu silmek için **DLTMQMQ** komutunu kullanın.

Kuyruқта kesinleştirilmemiş iletiler varsa ya da kullanıyorsa, kuyruk silinemez.

Büyük kuyrukların etkinleştirilmesi

IBM MQ, 2 GB 'den büyük kuyrukları destekler. Büyük dosyaları desteklemek üzere IBM i 'in nasıl etkinleştirileceğine ilişkin bilgi için işletim sistemi belgelerinize bakın.

IBM i ürün bilgilerini [IBM Documentation](#) içinde bulabilirsiniz.

Bazı yardımcı programlar 2 GB 'den büyük dosyalarla başa çıkamayabilir. Büyük dosya desteğini etkinleştirmeden önce, işletim sistemi belgelerinizde bu tür bir desteğe ilişkin sınırlamalara ilişkin bilgi olup olmadığını denetleyin.

IBM i üzerindeki diğer ad kuyruklarıyla çalışma

Bu bölümde, diğer ad kuyruklarını yönetmek için kullanabileceğiniz bazı komutlara ilişkin örnekler bulunur. Gösterilen tüm komutlar, **WRKMQMQ komut panosundaki** seçenekler kullanılarak da kullanılabilir.

Diğer ad kuyruğu (bazen kuyruk diğer adı olarak da bilinir), MQI çağrılarını yeniden yönlendirme yöntemi sağlar. Diğer ad kuyruğu gerçek bir kuyruk değil, gerçek bir kuyruğa çözülen bir tanımdır. Diğer ad kuyruğu tanımlaması, TGTQNAME özniteliğiyle belirtilen bir hedef kuyruk adı içeriyor.

Bir uygulama MQI çağrısında bir diğer ad kuyruğu belirttiğinde, kuyruk yöneticisi yürütme sırasında gerçek kuyruk adını çözer.

Örneğin, iletileri my.alias.queueadlı bir kuyruğa koymak için bir uygulama geliştirilmiştir. Bu kuyruk, bir **MQOPEN** isteği gönderdiğinde ve dolaylı olarak bu kuyruğa bir ileti koyarsa, bu kuyruğun adını belirtir. Uygulama, kuyruğun bir diğer ad kuyruğu olduğunu farkında değil. Bu diğer adı kullanan her MQI çağrısı için, kuyruk yöneticisi gerçek kuyruk adını çözer; bu, yerel bir kuyruk ya da bu kuyruk yöneticisinde tanımlanmış bir uzak kuyruk olabilir.

TGTQNAME özniteliğinin değerini değiştirerek, MQI çağrılarını büyük olasılıkla başka bir kuyruk yöneticisinde başka bir kuyruğa yeniden yönlendirebilirsiniz. Bu, bakım, geçiş ve yük dengeleme için yararlıdır.

Diğer ad kuyruğu tanımlanması

Aşağıdaki komut bir diğer ad kuyruğu yaratır:

```
CRTMQMQ QNAME('my.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
MQMNAME(MYQUEUEEMANAGER)
```

Bu komut, `my.alias.queue` değerini belirten MQI çağrılarını `yellow.queue` kuyruğuna yönlendirir. Komut hedef kuyruğu yaratmaz; yürütme sırasında `yellow.queue` kuyruğu yoksa MQI çağrıları başarısız olur.

Diğer ad tanımlamasını değiştirirseniz, MQI çağrılarını başka bir kuyruğa yönlendirebilirsiniz. Örneğin:

```
CHGMQMQ QNAME('my.alias.queue') TGTQNAME('magenta.queue') MQMNAME(MYQUEUEEMANAGER)
```

Bu komut, MQI çağrılarını başka bir kuyruğa (`magenta.queue`) yönlendirir.

Tek bir kuyruğun (hedef kuyruk) farklı uygulamalara ilişkin farklı özniteliklere sahip görünmesini sağlamak için diğer ad kuyruklarını da kullanabilirsiniz. Bunu, her uygulama için bir tane olmak üzere iki diğer ad tanımlayarak yaparsınız. İki uygulama olduğunu varsayalım:

- ALPHA uygulaması `yellow.queue` a ileti gönderebilir, ancak bu uygulamadan ileti almasına izin verilmez.
- Uygulama BETA, `yellow.queue` dan ileti alabilir, ancak üzerine ileti koymasına izin verilmez.

Bunu yapmak için aşağıdaki komutları kullanabilirsiniz:

```
/* This alias is put enabled and get disabled for application ALPHA */
CRTMQMQ QNAME('alphas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*YES) GETENBL(*NO) MQMNAME(MYQUEUEEMANAGER)

/* This alias is put disabled and get enabled for application BETA */
CRTMQMQ QNAME('betas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*NO) GETENBL(*YES) MQMNAME(MYQUEUEEMANAGER)
```

ALPHA, MQI çağrılarında `alphas.alias.queue` kuyruk adını kullanır; BETA, `betas.alias.queue` kuyruk adını kullanır. İkisi de aynı kuyruğa farklı yollarla erişiyor.

Diğer ad kuyruklarını tanımlarken REPLACE *YES özniteliğini, bu öznitelikleri yerel kuyruklarla aynı şekilde kullanabilirsiniz.

Diğer komutların diğer ad kuyruklarıyla kullanılması

Diğer ad kuyruğu özniteliklerini görüntülemek ya da değiştirmek için uygun komutları kullanabilirsiniz. Örneğin:

```
* Display the alias queue's attributes */
DSPMQMQ QNAME('alphas.alias.queue') MQMNAME(MYQUEUEEMANAGER)

/* ALTER the base queue name, to which the alias resolves. */
/* FORCE = Force the change even if the queue is open. */

CHQMCMQ QNAME('alphas.alias.queue') TGTQNAME('orange.local.queue') FORCE(*YES)
MQMNAME(MYQUEUEEMANAGER)
```

Bu bölümde, model kuyruklarını yönetmek için kullanabileceğiniz bazı komutlara ilişkin örnekler yer alır. Gösterilen tüm komutlar, **WRKMQMQ komut panosundaki** seçenekler kullanılarak da kullanılabilir.

Bir kuyruk yöneticisi, bir uygulamadan model kuyruğu olarak tanımlanmış bir kuyruk adını belirten bir MQI çağırısı alırsa dinamik bir kuyruk yaratır. Yeni dinamik kuyruğun adı, kuyruk yaratıldığında kuyruk yöneticisi tarafından oluşturulur. Model kuyruğu, bu kuyruktan yaratılan dinamik kuyrukların özniteliklerini belirleyen bir şablondur.

Model kuyrukları, uygulamaların gerektiğinde kuyruk yaratmaları için uygun bir yöntem sağlar.

Model kuyruğu tanımlanması

Bir model kuyruğunu, yerel bir kuyruk tanımladığınız şekilde bir öznitelikler kümesi ile tanımlarsınız. Model kuyrukları ve yerel kuyruklar aynı öznitelik kümesine sahiptir; ancak, model kuyruklarında yaratılan dinamik kuyrukların geçici mi, yoksa kalıcı mı olduğunu belirleyebilirsiniz. (Kalıcı kuyruklar kuyruk yöneticisi yeniden başlatma işlemleri boyunca tutulur, geçici kuyruklar tutulmaz). Örneğin:

```
CRTMQMQ QNAME('green.model.queue') QTYPE(*MDL) DFNTYPE(*PERMDYN)
```

Bu komut bir model kuyruğu tanımlaması yaratır. DFNTYPE özniteliğinden, bu şablondan yaratılan gerçek kuyruklar kalıcı dinamik kuyruklardır. Belirtilmeyen öznitelikler, SYSYTEM.DEFAULT.MODEL.QUEUE varsayılan kuyruğundan otomatik olarak kopyalanır.

Model kuyruklarını, yerel kuyruklarıyla aynı şekilde tanımlarken REPLACE *YES özniteliğini kullanabilirsiniz.

Model kuyruklarıyla diğer komutların kullanılması

Bir model kuyruğunun özniteliklerini görüntülemek ya da değiştirmek için uygun komutları kullanabilirsiniz. Örneğin:

```
/* Display the model queue's attributes */
DSPMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('green.model.queue')

/* ALTER the model queue to enable puts on any */
/* dynamic queue created from this model. */
CHGMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('blue.model.queue') PUTENBL(*YES)
```

Tetikleme ve süreç tanımlamalarıyla ilgili bilgi edinmek için bu bilgileri kullanın.

IBM MQ , bir kuyruktaki belirli koşullar karşılandığında uygulamayı otomatik olarak başlatmak için bir olanak sağlar. Bu koşullara bir örnek, kuyruktaki ileti sayısının belirli bir sayıya ulaşması olabilir. Bu olanağa *triggering* adı verilir ve [Triggering \(Tetikleme\) kanallarında](#) ayrıntılı olarak açıklanır.

Tetikleyen nedir?

Kuyruk yöneticisi, belirli koşulları tetikleyici olayları oluşturan olarak tanımlar. Bir kuyruk için tetikleme etkinleştirilirse ve bir tetikleme olayı ortaya çıkarsa, kuyruk yöneticisi başlatma kuyruğu adı verilen bir kuyruğa tetikleyici ileti gönderir. Tetikleyici iletisinin başlatma kuyruğunda bulunması, bir tetikleyici olayının oluştuğunu gösterir.

Kuyruk yöneticisi tarafından oluşturulan tetikleyici iletileri kalıcı değil. Bu, günlüğe kaydetmeyi azaltma (böylece performansı artırma) ve yeniden başlatma sırasında yinelemeleri en aza indirmeye etkisi yaratarak yeniden başlatma süresini kısaltır.

Tetikleyici monitörü nedir?

Başlatma kuyruğunu işleyen programa tetikleyici izleme uygulaması adı verilir ve bu programın işlevi tetikleme iletisini okumak ve tetikleme iletisinde bulunan bilgilere dayalı olarak uygun işlemi gerçekleştirmektir. Olağan durumda bu işlem, tetikleme iletisinin oluşturulmasına neden olan kuyruğu işlemek için başka bir uygulama başlatılmasını sağlar. Kuyruk yöneticisinin bakış açısından, tetikleyici izleme uygulamasıyla ilgili özel bir şey yoktur; bu, kuyruktaki (başlatma kuyruğu) iletileri okuyan başka bir uygulamadır.

Tetikleyici izleme programının iş sunma özniteliklerinin değiştirilmesi

STRMQMTRM komutu olarak sağlanan tetikleyici izleme programı, sistemin varsayılan iş tanımlamasını (QDFTJOBBD) kullanarak her tetikleyici iletisi için bir iş sunar. Bu, gönderilen işlerin her zaman QDFTJOBBD olarak adlandırılması ve kitaplık listesi de içinde olmak üzere *SYSVALvarsayılan iş tanımlamasının özniteliklerine sahip olması konusunda sınırlamalar içerir. IBM MQ , bu öznitelikleri geçersiz kılmaya ilişkin bir yöntem sağlar. Örneğin, gönderilen işler aşağıdaki gibi daha anlamlı iş adlarına sahip olacak şekilde uyarlanabilir:

1. İş tanımlamasında, istediğiniz tanımlamayı belirleyin; örneğin, günlük kaydı değerleri.
2. Tetikleme işleminde kullanılan süreç tanımlamasının ortam verilerini belirtin:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)')
```

Tetikleyici İzleme Programı, belirtilen tanımlamayı kullanarak bir SBMJOB işlemi gerçekleştirir.

Süreç tanımlamasının ortam verilerinde uygun anahtar sözcüğü ve değeri belirterek, SBMJOB ' in diğer özniteliklerini geçersiz kılabilirsiniz. Bu öznitelik tetikleyici izleme programı tarafından doldurulduğundan, bunun tek kural dışı durumu CMD anahtar sözcüğüdür. İş adının ve tanımının değiştirileceği süreç tanımlamasına ilişkin ortam verilerini belirten bir komut örneği:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)
JOB(TRIGGER)')
```

Tetikleme için uygulama kuyruğu tanımlanması

Uygulama kuyruğu, MQI aracılığıyla ileti alışverişi için uygulamalar tarafından kullanılan yerel bir kuyruktur. Tetikleme, uygulama kuyruğunda bir dizi kuyruk özniteliğinin tanımlanmasını gerektirir. Tetikleme TRGENBL özniteliği tarafından etkinleştirilir.

Bu örnekte, motor.insurance.queueyerel kuyruğunda öncelik düzeyi 5 ya da daha yüksek 100 ileti varsa, aşağıdaki gibi bir tetikleme olayı oluşturulur:

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.insurance.queue') QTYPE(*LCL)
PRCNAME('motor.insurance.quote.process') MAXMSGLEN(2000)
DFTMSGPST(*YES) INITQNAME('motor.ins.init.queue')
TRGENBL(*YES) TRGTYPE(*DEPTH) TRGDEPTH(100) TRGMSGPTY(5)
```

burada parametreler şunlardır:

MQMNAME (MYQUEUEMANAGER)

Kuyruk yöneticisinin adı.

QNAME ('motor.insurance.queue')

Tanımlanmakta olan uygulama kuyruğunun adı.

PRCNAME ('motor.insurance.quote.process')

Tetikleyici izleme programı tarafından başlatılacak uygulamanın adı.

MAXMSGLEN (2000)

Kuyruktaki iletilerin uzunluk üst sınırı.

DFTMSGPST(*YES)

Bu kuyruktaki iletiler varsayılan olarak kalıcıdır.

INITQNAME('motor.ins.init.queue')

Kuyruk yöneticisinin tetikleme iletilerini koyacağı başlatma kuyruğunun adı.

TRGENBL(*YES)

Tetikleyici özniteliği değeri.

TRGTYPE(*DEPTH)

Gerekli önceliğe sahip iletilerin sayısı (**TRGMSGPTY**) **TRGDEPTH** içinde belirtilen sayıya ulaşır.

TRGDEPTH(100)

Bir tetikleyici olayı oluşturmak için gereken ileti sayısı.

TRGMSGPTY(5)

Bir tetikleyici olayı oluşturulup oluşturulmayacağına karar vermede kuyruk yöneticisi tarafından sayılacak iletilerin önceliği. Yalnızca 5 ya da daha yüksek önceliğe sahip iletiler sayılır.

Başlatma kuyruğu tanımlanması

Bir tetikleme olayı oluşturduğunda, kuyruk yöneticisi uygulama kuyruğu tanımlamasında belirtilen başlatma kuyruğuna bir tetikleyici iletilerini koyar. Başlatma kuyruklarının özel ayarları yoktur, ancak yol göstermek için `motor.ins.init.queue` yerel kuyruğunun aşağıdaki tanımını kullanabilirsiniz:

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.ins.init.queue') QTYPE(*LCL)
GETENBL(*YES) SHARE(*NO) TRGTYPE(*NONE)
MAXMSGL(2000)
MAXDEPTH(1000)
```

Süreç tanımlaması yaratılması

Bir süreç tanımlaması yaratmak için **CRTMQMPRC** komutunu kullanın. Süreç tanımlaması, bir uygulama kuyruğunu, kuyruktaki iletileri işleyecek uygulamayla ilişkilendirir. Bu, uygulama kuyruğundaki PRCNAME özniteliği `motor.insurance.queue` aracılığıyla gerçekleştirilir. Aşağıdaki komut, bu örnekte tanımlanan gerekli işlemi (`motor.insurance.quote.process`) yaratır:

```
CRTMQMPRC MQMNAME(MYQUEUEMANAGER) PRCNAME('motor.insurance.quote.process')
TEXT('Insurance request message processing')
APPATYPE(*OS400) APPID(MQTEST/TESTPROG)
USRDATA('open, close, 235')
```

burada parametreler şunlardır:

MQMNAME(MYQUEUEMANAGER)

Kuyruk yöneticisinin adı.

PRCNAME('motor.insurance.quote.process')

Süreç tanımlamasının adı.

TEXT('Insurance request message processing')

Bu tanımın ilişkili olduğu uygulama programının açıklaması. Bu metin, **DSPMQMPRC** komutunu kullandığınızda görüntülenir. Bu, sürecin ne yaptığını tanımlamanıza yardımcı olabilir. Dizgide boşluk kullanırsanız, dizgiyi tek tırnak içine almanız gerekir.

APPATYPE(*OS400)

Başlatılacak uygulamanın tipi.

APPID(MQTEST/TESTPROG)

Tam olarak nitelenmiş bir dosya adı olarak belirtilen, uygulama yürütülür dosyasının adı.

USRDATA('open, close, 235')

Uygulama tarafından kullanılabilen kullanıcı tanımlı veriler.

Süreç tanımlamanızı görüntüleme

Tanımınızın sonuçlarını incelemek için **DSPMQMPCRC** komutunu kullanın. Örneğin:

```
MQMNAME(MYQUEUEMANAGER) DSPMQMPCRC('motor.insurance.quote.process')
```

Var olan bir süreç tanımlamasını değiştirmek için **CHGMQMPCRC** komutunu ve bir süreç tanımlamasını silmek için **DLTMQMPCRC** komutunu kullanabilirsiniz.

IBM i

IBM i üzerinde iki IBM MQ sistemi arasında iletişim kurma

Bu kodlama örneği, birbirleriyle iletişim kurabilmeleri için CL komutlarını kullanarak iki IBM MQ for IBM i sisteminin nasıl ayarlanacağını gösterir.

Sistemler SYSTEMA ve SYSTEMB olarak adlandırılır ve kullanılan iletişim protokolü TCP/IP ' dir.

Aşağıdaki yordamı gerçekleştirin:

1. SYSTEMA üzerinde QMGRA1 adını kullanarak bir kuyruk yöneticisi oluşturun.

```
CRTMQM  MQMNAME(QMGRA1) TEXT('System A - Queue +  
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

2. Bu kuyruk yöneticisini başlatın.

```
STRMQM  MQMNAME(QMGRA1)
```

3. SYSTEMA üzerinde, SYSTEMB üzerindeki bir kuyruk yöneticisine ileti göndermeniz için gereken IBM MQ nesnelerini tanımlayın.

```
/* Transmission queue */  
CRTMQMQ  QNAME(XMITQ.TO.QMGRB1) QTYPE(*LCL) +  
MQMNAME(QMGRA1) TEXT('Transmission Queue +  
to QMGRB1') MAXDEPTH(5000) USAGE(*TMQ)  
  
/* Remote queue that points to a queue called TARGETB */  
/* TARGETB belongs to queue manager QMGRB1 on SYSTEMB */  
CRTMQMQ  QNAME(TARGETB.ON.QMGRB1) QTYPE(*RMT) +  
MQMNAME(QMGRA1) TEXT('Remote Q pointing +  
at Q TARGETB on QMGRB1 on Remote System +  
SYSTEMB') RMTQNAME(TARGETB) +  
RMTMQMNAME(QMGRB1) TMQNAME(XMITQ.TO.QMGRB1)  
  
/* TCP/IP sender channel to send messages to the queue manager on SYSTEMB*/  
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*SDR) +  
MQMNAME(QMGRA1) TRPTYPE(*TCP) +  
TEXT('Sender Channel From QMGRA1 on +  
SYSTEMA to QMGRB1 on SYSTEMB') +  
CONNNAME(SYSTEMB) TMQNAME(XMITQ.TO.QMGRB1)
```

4. SYSTEMB üzerinde QMGRB1 adını kullanarak bir kuyruk yöneticisi oluşturun.

```
CRTMQM  MQMNAME(QMGRB1) TEXT('System B - Queue +  
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

5. SYSTEMB üzerinde kuyruk yöneticisini başlatın.

```
STRMQM  MQMNAME(QMGRB1)
```

6. SYSTEMA üzerindeki kuyruk yöneticisinden ileti almak için gereksinim duyduğunuz IBM MQ nesnelerini tanımlayın.

```
/* Local queue to receive messages on */  
CRTMQMQ  QNAME(TARGETB) QTYPE(*LCL) MQMNAME(QMGRB1) +
```

```

TEXT('Sample Local Queue for QMGRB1')

/* Receiver channel of the same name as the sender channel on SYSTEMA */
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*RCVR) +
MQMNAME(QMGRB1) TRPTYPE(*TCP) +
TEXT('Receiver Channel from QMGRA1 to +
QMGRB1')

```

7. Son olarak, SYSTEMB üzerinde bir TCP/IP dinleyicisi başlatın; böylece kanal başlatılabilir. Bu örnek, varsayılan kapı olan 1414 'ü kullanır.

```
STRMQMLSR MQMNAME(QMGRB1)
```

Artık SYSTEMA ile SYSTEMB arasında test iletileri göndermeye hazırsınız. Sağlanan örneklerden birini kullanarak, SYSTEMA üzerindeki uzak kuyruğunuza bir dizi ileti koyun.

Kanalı SYSTEMA üzerinden **STRMQMCHL** komutunu kullanarak ya da **WRKMQMCHL** komutunu kullanarak ve gönderen kanalı için bir başlatma isteği (Seçenek 14) girerek başlatın.

Kanal RUNNING durumuna geçmeli ve iletiler SYSTEMB üzerindeki TARGETB kuyruğuna gönderilmelidir.

Şu komutu vererek iletilerinizi denetleyin:

```
WRKMQMMSG QNAME(TARGETB) MQMNAME(QMGRB1).
```

IBM i

IBM i üzerinde örnek kaynak tanımlamaları

Bu örnek, AMQSAMP4 örnek IBM i CL programını içerir.

```

/*****/
/*
/* Program name: AMQSAMP4
/*
/* Description: Sample CL program defining MQM queues
/* to use with the sample programs
/* Can be run, with changes as needed, after
/* starting the MQM
/*
/* <N_OCO_COPYRIGHT> */
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 1993, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* <NOC_COPYRIGHT> */
/*
/*****/
/*
/* Function:
/*
/* AMQSAMP4 is a sample CL program to create or reset the
/* MQI resources to use with the sample programs.
/*
/* This program, or a similar one, can be run when the MQM
/* is started - it creates the objects if missing, or resets
/* their attributes to the prescribed values.
/*
/*
/*
/* Exceptions signaled: none
/* Exceptions monitored: none
/*
/* AMQSAMP4 takes a single parameter, the Queue Manager name
/*
/*****/
QSYS/PGM PARM(&QMGRNAME)

```

```

/*****
/* Queue Manager Name Parameter */
/*****
QSYS/DCL VAR(&QMGRNAME) TYPE(*CHAR)

/*****
/*   EXAMPLES OF DIFFERENT QUEUE TYPES   */
/*   */
/*   Create local, alias and remote queues   */
/*   */
/*   Uses system defaults for most attributes   */
/*   */
/*****
/* Create a local queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.LOCAL') +
MQMNAME (&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Sample local queue') /* description */+
SHARE(*YES) /* Shareable */+
DFTMSGPST(*YES) /* Persistent messages OK */

/* Create an alias queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.ALIAS') +
MQMNAME (&QMGRNAME) +
QTYPE(*ALS) REPLACE(*YES) +
+
TEXT('Sample alias queue') +
DFTMSGPST(*YES) /* Persistent messages OK */+
TGTQNAME('SYSTEM.SAMPLE.LOCAL')

/* Create a remote queue - in this case, an indirect reference */
/* is made to the sample local queue on OTHER queue manager */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REMOTE') +
MQMNAME (&QMGRNAME) +
QTYPE(*RMT) REPLACE(*YES) +
+
TEXT('Sample remote queue')/* description */+
DFTMSGPST(*YES) /* Persistent messages OK */+
RMTQNAME('SYSTEM.SAMPLE.LOCAL') +
RMTQMNAME(OTHER) /* Queue is on OTHER */

/* Create a transmission queue for messages to queues at OTHER */
/* By default, use remote node name */
CRTMQMQ QNAME('OTHER') /* transmission queue name */+
MQMNAME (&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
TEXT('Transmission queue to OTHER') +
USAGE(*TMQ) /* transmission queue */

/*****
/*   SPECIFIC QUEUES AND PROCESS USED BY SAMPLE PROGRAMS   */
/*   */
/*   Create local queues used by sample programs   */
/*   Create MQI process associated with sample initiation queue   */
/*   */
/*****
/* General reply queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REPLY') +
MQMNAME (&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('General reply queue') +
DFTMSGPST(*NO) /* Not Persistent */

/* Queue used by AMQSINQ4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.INQ') +
MQMNAME (&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSINQ4') +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO) /* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.INQPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSSET4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.SET') +
MQMNAME (&QMGRNAME) +

```

```

QTYPE(*LCL) REPLACE(*YES)      +
+
TEXT('Queue for AMQSSET4')      +
SHARE(*YES) /* Shareable */ +
DFTMSGPST(*NO)/* Not Persistent */ +
+
TRGENBL(*YES) /* Trigger control on */ +
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.SETPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSECH4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.ECHO') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSECH4') +
SHARE(*YES) /* Shareable */ +
DFTMSGPST(*NO)/* Not Persistent */ +
+
TRGENBL(*YES) /* Trigger control on */ +
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Initiation Queue used by AMQSTRG4, sample trigger process */
CRTMQMQ QNAME('SYSTEM.SAMPLE.TRIGGER') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
TEXT('Trigger queue for sample programs')

/* MQI Processes associated with triggered sample programs */
/*
/***** Note - there are versions of the triggered samples *****/
/***** in different languages - set APPID for these *****/
/***** process to the variation you want to trigger *****/
/*
CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.INQPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
+
TEXT('Trigger process for AMQSINQ4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSINQ4') /* C */ +
/* APPID('QMOM/AMQ0INQ4') /* COBOL */ +
/* APPID('QMOM/AMQ3INQ4') /* RPG - ILE */

CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.SETPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
+
TEXT('Trigger process for AMQSSET4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSSET4') /* C */ +
/* APPID('QMOM/AMQ0SET4') /* COBOL */ +
/* APPID('QMOM/AMQ3SET4') /* RPG - ILE */

CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
+
TEXT('Trigger process for AMQSECH4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSECH4') /* C */ +
/* APPID('QMOM/AMQ0ECH4') /* COBOL */ +
/* APPID('QMOM/AMQ3ECH4') /* RPG - ILE */

/*****/
/*
/* Normal return.
/*
/*****/
SNDPGMMMSG MSG('AMQSAMP4 Completed creating sample +
objects for ' *CAT &QMGRNAME)
RETURN
ENDPGM

/*****/
/*

```

IBM i IBM MQ for IBM i ' i yönetmenin alternatif yolları

CL komutlarının kullanılması, IBM MQ for IBM i yönetimi için tercih edilen yöntemdir. Ancak, MQSC komutları, PCF komutları, denetim komutları ve uzak denetim de içinde olmak üzere çeşitli denetim yöntemlerini kullanabilirsiniz.

Bu görev hakkında

Olağan durumda IBM i Denetim dili (CL) komutlarını IBM MQ for IBM i kullanırsınız. Bu komutlara genel bakış için bkz. [“CL komutlarını kullanarak IBM MQ for IBM i yönetimi” sayfa 384.](#)

Alt konularda açıklandığı gibi MQSC komutlarını ve PCF komutlarını kullanabilir ve denetim komutlarını [“IBM MQ for Multiplatforms ' i denetim komutlarını kullanarak yönetme” sayfa 10](#) konusunda açıklandığı gibi kullanabilirsiniz.

Kuyruk yöneticilerinin çalışmasını izlemek için IBM MQ izleme kodu ekleme olaylarını kullanabilirsiniz. IBM MQ özel işlem denetim geçirme olayları ve bunların nasıl kullanılacağına ilişkin bilgi için [Özel İşlem Denetim Geçirme Olayları](#) konusuna bakın.

IBM i CL komutlarını kullanmaya alternatif olarak aşağıdaki alt konularda açıklanan yönetim yöntemlerinden herhangi birini kullanın:

IBM i IBM i üzerinde yerel ve uzak yönetim

IBM MQ for IBM i nesnelerini yerel olarak ya da uzaktan yönetebilirsiniz.

Bu görev hakkında

Yerel denetim , yerel sisteminizde tanımladığınız kuyruk yöneticilerine ilişkin denetim görevlerini gerçekleştirmek anlamına gelir. IBM MQ' da bunu şu işte işi yerel yönetim olarak düşünebilirsiniz; IBM MQ kanalları bulunmaz, yani iletişim işletim sistemi tarafından yönetilir. Bu tip bir görevi gerçekleştirmek için uzak sistemde oturum açıp komutları buradan vermeli ya da komutları sizin için yayınlayabilecek bir işlem yaratmalısınız.

IBM MQ , *uzaktan yönetim* olarak bilinen tek bir noktadan yönetimi destekler. Uzak denetim, hedef kuyruk yöneticisindeki SYSTEM . ADMIN . COMMAND . QUEUE ' e programlanabilir komut biçimi (PCF) denetim iletileri gönderilmesini içerir.

PCF iletileri oluşturmanın birçok yolu vardır. Bunlar aşağıdaki adımlarda açıklanır.

Yordam

- PCF iletilerini kullanarak bir program yazın. Bkz. [“IBM i üzerinde PCF komutlarını kullanarak denetim” sayfa 400.](#)
- PCF iletileri gönderen MQAI kullanarak bir program yazın. Bkz. [“PCF 'lerin kullanımını basitleştirmek için MQAI' nin kullanılması” sayfa 37.](#)
- IBM MQ for Windows ile birlikte sağlanan IBM MQ Explorer ' ı kullanın; bu, grafik kullanıcı arabirimi (GUI) kullanmanıza ve doğru PCF iletilerini oluşturmanıza olanak sağlar. Bkz. [“IBM MQ Explorer ' in IBM MQ for IBM i ile kullanılması” sayfa 400.](#)
- Uzak bir kuyruk yöneticisine dolaylı olarak komut göndermek için **STRMQMQSC** komutunu kullanın. Bkz. [“IBM i üzerinde MQSC komutlarını kullanarak yönetme” sayfa 399.](#)

Örneğin, uzak bir kuyruk yöneticisindeki bir kuyruk tanımlamasını değiştirmek için uzak bir komut verebilirsiniz.

Bazı komutlar, kuyruk yöneticikullanılacak sonra mi başlatılsın, yoksa kuyruk yöneticilerini ben; olamaz de de bu olur bu olur bu de bu de. Bu tip bir görevi gerçekleştirmek için, uzak sistemde oturum açıp komutları oradan vermeniz ya da komutları sizin için yayınlayabilecek bir işlem yaratmanız gerekir.

IBM i

IBM i üzerinde MQSC komutlarını kullanarak yönetme

IBM i' ta, Komut dosyasında bir komut listesi oluşturup dosyayı **STRMQMMQSC** komutunu kullanarak çalıştırıbelirtin-! Kuyruk yöneticisi, kuyruklar, süreç tanımlamaları, ad alanları, kanallar, istemci bağlantısı kanalları, dinleyiciler, hizmetler, konular ve kimlik doğrulama bilgileri nesnelere de içinde olmak üzere kuyruk yöneticisi nesnelere yönetmek için MQSC komutlarını kullanabilirsiniz.

Bu görev hakkında

IBM MQ komut dosyası (MQSC) komutları, EBCDIC metni içinde, insan tarafından okunabilir biçimde yazılır. **STRMQMMQSC** IBM MQ CL komutunu kullanarak bir kuyruk yöneticisine MQSC komutları yayınlayın. Bu yöntem, girişini sunucu kitaplık sistemindeki bir kaynak fiziksel dosyadan alan bir toplu iş yöntemidir. Bu kaynak fiziksel dosyanın varsayılan adı QMQSC' dir.



Uyarı: QTEMP kitaplığının kullanımı sınırlı olduğundan, QTEMP kitaplığını STRMQMMQSC kaynak kitaplığı olarak kullanmayın. Komutun giriş dosyası olarak başka bir kitaplık kullanmalısınız.

IBM MQ ortamları arasında taşınabilirlik için, MQSC komut dosyalarındaki satır uzunluğunu 72 karakterle sınırlayın. Komutun sonraki satırda devam ettiğini belirtmek için artı işareti kullanın.

MQSC ' de belirtilen nesne öznitelikleri büyük harfle (örneğin, QMNAME) gösterilir, ancak bunlar büyük/küçük harfe duyarlı değildir.

Not:

1. MQSC dosyasının biçimi, dosya sistemindeki konumuna bağlı değildir.
2. MQSC öznitelik adları sekiz karakterle sınırlıdır.
3. MQSC komutları, z/OSda dahil olmak üzere diğer platformlarda kullanılabilir.

Her MQSC komutunun tanımı ve sözdizimi için [MQSC komutları](#) konusuna bakın.

Yordam

1. QMQSC kaynak dosyasını yaratın.

IBM MQ for IBM i , QMQSCadlı bir kaynak dosya sağlamaz. MQSC komutlarını işlemek için, QMQSC kaynak dosyasını seçtiğiniz bir kitaplıkta yaratmanız gerekir; bunun için aşağıdaki komutu verin:

```
CRTSRCPF FILE(MYLIB/QMQSC) RCDLEN(240) TEXT(' IBM MQ - MQSC Source')
```

2. Üyelerle çalışın.

MQSC kaynağı, bu kaynak dosya içindeki üyelerde tutuluyor. Üyelerle çalışmak için aşağıdaki komutu girin:

```
WRKMBRPDM MYLIB/QMQSC
```

Artık yeni üyeler ekleyebilir ve var olanları koruyabilirsiniz.

```

.
.
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +
DESCR(' ') +
PUT(ENABLED) +
DEFPRTY(0) +
DEFPSIST(NO) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(1024) +
DEFSOPT(SHARED) +
NOHARDENBO +
USAGE(NORMAL) +
NOTRIGGER;
.
.

```

Şekil 20. Öznitelikleri olan bir MQSC komutunu (DEFINE QLOCAL) gösteren bir MQSC komut dosyasından (myprog.in) alın.

İlgili bilgiler

[IBM MQ ' nin MQSC komutlarını kullanarak denetlenmesi](#)

IBM i

IBM i üzerinde PCF komutlarını kullanarak denetim

IBM MQ programlanabilir komut biçimi (PCF) komutlarının amacı, denetim görevlerinin bir denetim programına programlanmasına izin vermektir. Bu şekilde, bir programdan kuyruklar ve süreç tanımlamaları yaratabilir ve kuyruk yöneticilerini değiştirebilirsiniz.

PCF komutları, MQSC komutları tarafından sağlanan aynı işlev aralığını kapsar. Ancak, MQSC komutlarından farklı olarak, PCF komutları ve yanıtları okuyabileceğiniz bir metin biçiminde değildir.

Ağdaki herhangi bir kuyruk yöneticisine tek bir düğümden PCF komutları yayınlamak için bir program yazabilirsiniz. Bu şekilde, yönetim görevlerini hem merkezileştirir hem de otomatikleştirebilirsiniz.

Her PCF komutu, IBM MQ iletilisinin uygulama verileri bölümüne gömülü bir veri yapısıdır. Her komut, MQI işlevi MQPUT diğer iletilerle aynı şekilde kullanılarak hedef kuyruk yöneticisine gönderilir. İletiyi alan kuyruk yöneticisindeki komut sunucusu, iletiyi bir komut iletilisi olarak yorumlar ve komutu çalıştırır. Yanıtları almak için uygulama bir MQGET çağırısı yayınlamalı ve yanıt verileri başka bir veri yapısında döndürülür. Daha sonra uygulama yanıtı işleyebilir ve buna göre işlem yapabilir.

Kısaca, uygulama programcısının PCF komut iletilisi yaratmak için belirtmesi gereken bazı konular şunlardır:

İleti tanımlayıcı

Bu, aşağıdaki özelliklere sahip standart bir IBM MQ ileti tanımlayıcısıdır:

- İleti tipi (*MsgType*) MQMT_REQUEST.
- İleti biçimi (*Format*) MQFMT_ADMIN.

Uygulama Verileri

PCF üstbilgisini içeren PCF iletilisini içerir. Bu iletide aşağıdakiler yer alır:

- PCF ileti tipi (*Type*) MQCFT_COMMAND belirtiyor.
- Komut tanıtıcısı komutu belirtir; örneğin, *Change Queue* (MQCMD_CHANGE_Q).

Çıkış PCF ' leri, ileti metni içinde MQSC komutlarını içeren PCF komutlarıdır. Uzak kuyruk yöneticisine komut göndermek için PCF ' leri kullanabilirsiniz. Daha fazla bilgi için bkz. [“PCF ' lerin kullanımını basitleştirmek için MQAI' nin kullanılması” sayfa 37](#) .

PCF veri yapılarının tam açıklaması ve bunların nasıl uygulanacağına ilişkin bilgi için [Komutlar ve yanıtlar için yapılarbaşlıklı konuya](#) bakın.

IBM i

IBM MQ Explorer ' in IBM MQ for IBM i ile kullanılması

IBM MQ Explorer kullanarak IBM MQ for IBM i ' ı yönetmek için bu bilgileri kullanın.

IBM MQ for Windows (x86 platformu) ve IBM MQ for Linux (x86 ve x86-64 platformları), denetim görevlerini CL, denetim ya da MQSC komutlarının kullanılmasına alternatif olarak gerçekleştirmek için IBM MQ Explorer adı verilen bir yönetim arabirimi sağlar.

IBM MQ Explorer , IBM MQ Explorer ürününü ilgilendiginiz kuyruk yöneticilerine ve kümelere yönlendirerek Windows (x86 platformu) ya da Linux (x86 ve x86-64 platformları) çalıştıran bir bilgisayardan ağınızın yerel ya da uzaktan yönetimini gerçekleştirmenizi sağlar.

IBM MQ Explorer ile aşağıdakileri yapabilirsiniz:

- Bir kuyruk yöneticisini başlatın ve durdurun (yalnızca yerel makinenizde).
- Kuyruklar, konular ve kanallar gibi IBM MQ nesnelерinin tanımlarını tanımlayın, görüntüleyin ve değiştirin.
- Kuyruktaki iletilere göz atın.
- Bir kanalı başlat ve durdur.
- Bir kanalla ilgili durum bilgilerini görüntüleyin.
- Bir kümedeki kuyruk yöneticilerini görüntüleyin.
- Hangi uygulamaların, kullanıcıların ya da kanalların belirli bir kuyruğun açık olduğunu denetleyin.
- **Yeni Küme Yarat** sihirbazını kullanarak yeni bir kuyruk yöneticisi kümesi yaratın.
- **Kümeye Kuyruk Yöneticisi Ekle** sihirbazını kullanarak bir kümeye kuyruk yöneticisi ekleyin.
- TLS (Transport Layer Security; İletim Katmanı Güvenliği) kanal güvenliği ile kullanılan kimlik doğrulama bilgileri nesnesini yönetin.

Çevrimiçi kılavuzu kullanarak şunları yapabilirsiniz:

- Kuyruk yöneticileri, kuyruklar, kanallar, süreç tanımlamaları, istemci bağlantısı kanalları, dinleyiciler, konular, hizmetler, ad listesi ve kümeler de dahil olmak üzere çeşitli kaynakları tanımlayın ve denetleyin.
- Bir kuyruk yöneticisini ve ilişkili işlemlerini başlatır ya da durdurur.
- İş istasyonunuzdaki ya da diğer iş istasyonlarındaki kuyruk yöneticilerini ve ilişkili nesnelерini görüntüleyin.
- Kuyruk yöneticilerinin, kümelerin ve kanalların durumunu denetleyin.

Bir sunucu makinesinde IBM MQ Explorer yönetmek üzere IBM MQ ' i kullanmayı denemeden önce aşağıdaki gereksinimleri karşıladığınızdan emin olun. Aşağıdakileri denetleyin:

1. **STRMQCSVR** Denetim dili (CL) komutuyla sunucuda başlatılan, denetlenmekte olan herhangi bir kuyruk yöneticisi için bir komut sunucusu çalışıyor.
2. Her uzak kuyruk yöneticisi için uygun bir TCP/IP dinleyicisi vardır. Bu, **STRMQLSR** komutuyla başlatılan IBM MQ dinleyicidir.
3. SYSTEM.ADMIN.SVRCONN adlı sunucu bağlantısı kanalı, her uzak kuyruk yöneticisinde bulunur. Bu kanalı kendiniz yaratmalısınız. Bu, denetlenmekte olan her uzak kuyruk yöneticisi için zorunludur. Bu olmadan, uzaktan yönetim mümkün değildir.
4. SYSTEM.MQEXPLORER.REPLY.MODEL kuyruğunun var olduğunu doğrulayın.

IBM i **IBM i üzerinde uzaktan yönetim için komut sunucusunun yönetilmesi**

IBM MQ for IBM i komut sunucusunun uzaktan yönetimi hakkında bilgi edinmek için bu bilgileri kullanın.

Her kuyruk yöneticisiyle ilişkilendirilmiş bir komut sunucusu olabilir. Komut sunucusu, uzak kuyruk yöneticilerinden gelen komutları ya da uygulamalardan gelen PCF komutlarını işler. Kuyruk yöneticisine işlenmek üzere komutlar sunar ve komutun kaynağına bağlı olarak bir tamamlanma kodu ya da işletmen iletisi döndürür.

Komut sunucusu, PCF 'leri, MQAI' yi ve uzak yönetimi içeren tüm yönetim için zorunludur.

Not: Uzak yönetim için, hedef kuyruk yöneticisinin çalıştığından emin olmanız gerekir. Tersi durumda, komutlar içeren iletiler, yayınlandıkları kuyruk yöneticisinden çıkamaz. Bunun yerine, bu iletiler uzak kuyruk yöneticisine hizmet veren yerel iletim kuyruğunda kuyruğa alınır. Mümkünse bu durumdan kaçınınız.

Komut sunucusunu başlatmak ve durdurmak için ayrı denetim komutları vardır. IBM MQ Gezgini 'ni kullanarak aşağıdaki bölümlerde açıklanan işlemleri gerçekleştirebilirsiniz.

Komut sunucusunun başlatılması ve durdurulması

Komut sunucusunu başlatmak için şu CL komutunu kullanınız:

```
STRMQMSVR MQMNAME('saturn.queue.manager')
```

Burada `saturn.queue.manager`, komut sunucusunun başlatıldığı kuyruk yöneticisidir.

Komut sunucusunu durdurmak için aşağıdaki CL komutlarından birini kullanınız:

1.

```
ENDMQMSVR MQMNAME('saturn.queue.manager') OPTION(*CNTRL)
```

Burada `saturn.queue.manager`, komut sunucusunun durdurulmakta olduğu kuyruk yöneticisidir. Bu, `OPTION(*CNTRL)` öğesinin atlanabileceği anlamına gelen varsayılan seçenektir.

2.

```
ENDMQMSVR MQMNAME('saturn.queue.manager') OPTION(*IMMED)
```

Burada `saturn.queue.manager`, komut sunucusunun durdurulmakta olduğu kuyruk yöneticisidir.

Komut sunucusunun durumunun görüntülenmesi

Uzak yönetim için, hedef kuyruk yöneticisindeki komut sunucusunun çalıştığını doğrulayınız. Çalışmıyorsa, uzak komutlar işlenemez. Komutları içeren iletiler, hedef kuyruk yöneticisinin komut kuyruğunda (`SYSTEM.ADMIN.COMMAND.QUEUE`) kuyruğa alınır.

Burada `saturn.queue.manager` olarak adlandırılan bir kuyruk yöneticisine ilişkin komut sunucusunun durumunu görüntülemek için CL komutu şöyledir:

```
DSPMQMSVR MQMNAME('saturn.queue.manager')
```

Hedef makinede bu komutu verin. Komut sunucusu çalışıyorsa, [Şekil 21 sayfa 402](#) içinde gösterilen pano görüntülenir:

```
Display MQM Command Server (DSPMQMSVR)
```

```
Queue manager name . . . . . > saturn.queue.manager
```

```
MQM Command Server Status. . . . > RUNNING
```

```
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys
```

Şekil 21. MQM Komut Sunucusu Panosunu Görüntüle

Web konsoluyla ilgili Qshell komutlarının IBM MQ for IBM üzerinde düzgün çalışması için ortamınızı aşağıdaki metinde açıklandığı gibi yapılandırmanız gerekir.

Bu görev hakkında

Qshell başlatıldığında, işin CCSID 'sine dayalı olarak komutları işlemek üzere iç çizelgeleri başlatır. Web konsoluyla ilgili Qshell komutlarının IBM üzerinde doğru çalışması için ortamınızı yapılandırmanız gerekir.

Yerel ayar, LANG ortam değişkeninin bir yerel ayar nesnesine yol adı olarak ayarlanmasıyla belirlenir. Örneğin, US English ülke değerini ayarlamak için LANG ortam değişkeni aşağıdaki gibi ayarlanır:

```
LANG=/QSYS.LIB/EN_US.LOCALE
```

Qshell 'de, tüm ortam değişkenlerini listelemek için komut kümesini yayınlayarak ayarı denetleyebilirsiniz. Genellikle, yürütme ortamının ülke değerini etkileyebilecek LANG ' dir. LC_ALL da olabilir.

Qshell komutlarını doğru çalıştırmak için, yerel ayar ortamı ayarının iş ayarınızla tutarlı olması gerekir.

Yordam

DSPJOB JOB (JobNumber/USERProfile/JobName) CL komutunu kullanın.

- İş tanımı özniteliklerini görüntülemek için seçenek 2 'yi belirleyin.
- Aşağıdaki öznitelikler LANG ya da LC_ALL ortam ayarıyla tutarlı olmalıdır

- Dil tanıtıcısı
- Ülke ya da bölge tanıtıcısı
- Kodlanmış karakter takımı tanıtıcısı

Örneğin,

```
LANG=/QSYS.LIB/FR_FR.LOCALE
```

iş öznitelikleriniz şunlar olmalıdır:

- Dil Tanıtıcısı..... fra
- Ülke ya da bölge tanıtıcısı.... FR
- Kodlanmış karakter takımı tanıtıcısı... 297

Sonraki adım

Ulusal dil desteğine ilişkin ek bilgi için IBM Documentation [Ulusal dil desteği \(NLS\) ile ilgili önemli noktalar](#) başlıklı konuya bakın.

Bu bilgilerde, IBM MQ ' in iş isteklerini nasıl işleyeceği ve IBM MQ ile ilişkili işlerin önceliklerinin belirlenmesi ve denetlenmesi için kullanılacak seçenekler açıklanır.

Uyarı

IBM i ve IBM MQ iş yönetimi kavramlarını tam olarak anlamadığınız sürece IBM MQ iş yönetimi nesnelere değiştirmeyin.

Altsistemlere ve iş tanımlarına ilişkin ek bilgileri IBM i ürün belgelerinde [İş Yönetimi](#) altında bulabilirsiniz. [İşleri başlatma](#) ve [Toplu işler](#) bölümlerindeki bölümlere özellikle dikkat edin.

IBM MQ for IBM i , IBM i UNIX ortamını ve IBM i iş parçacıklarını birleştirir. Integrated File System (IFS) içindeki nesnelere değişiklik **yapmayın** .

Olağan işlemler sırasında, IBM MQ kuyruk yöneticisi farklı görevleri gerçekleştirmek için bir dizi toplu iş başlatır. Varsayılan olarak bu toplu işler, IBM MQ kurulduğunda yaratılan QMQM altsisteminde çalışır.

İş yönetimi, sisteminizden en iyi performansı elde etmek ya da yönetimi basitleştirmek için IBM MQ görevlerinin uyarlanması işlemi ifade eder.

Örneğin, şunları yapabilirsiniz:

- Bir kuyruk yöneticisini diğerinden daha duyarlı hale getirmek için işlerin çalıştırma önceliğini değiştirin.
- Bir dizi işin çıkışını belirli bir çıkış kuyruğuna yeniden yönlendirin.
- Belirli bir tipteki tüm işlerin belirli bir altsistemde çalıştırılmasını sağlar.
- Hataları bir altsistemde yalıtın.

İş yönetimi, IBM MQ işleriyle ilişkili iş tanımlamaları yaratılarak ya da değiştirilerek gerçekleştirilir. İş yönetimini aşağıdakiler için yapılandırabilirsiniz:

- Tam bir IBM MQ kuruluşu.
- Tek tek kuyruk yöneticileri.
- Tek tek kuyruk yöneticileri için tek tek işler.

IBM i IBM MQ görevleri IBM i

Bu, IBM MQ for IBM i işlerini içeren bir çizelgedir ve her birinin kısa bir tanımlamasıdır.

Bir kuyruk yöneticisi çalışırken, IBM MQ altsisteminde QMQM kullanıcı tanıtımı altında çalışan aşağıdaki toplu işlerin bazılarını ya da tümünü görürsünüz. İşler [Çizelge 21 sayfa 404](#) içinde kısaca açıklanır.

Bir kuyruk yöneticisine bağlı tüm işleri, **Kuyruk Yöneticisiyle Çalışma** (WRKMQM) panosunda seçenek 22 'yi kullanarak görüntüleyebilirsiniz. WRKMQLSR komutunu kullanarak dinleyicileri görüntüleyebilirsiniz.

İş adı	İşlev
AMQZMUC0	Yardımcı program yöneticisi. Bu iş, günlük zinciri yöneticisi gibi kritik kuyruk yöneticisi yardımcı programlarını yürütür.
AMQZXMA0	Kuyruk yöneticisi tarafından başlatılan ilk iş olan yürütme denetleyicisi. MQCONN isteklerini işler ve IBM MQ API çağrılarını işlemek için aracı işlemlerini başlatır.
AMQZFUMA	Nesne yetki yöneticisi (OAM).
AMQZLAA0	MQCNO_STANDARD_BINDING ile kuyruk yöneticisine bağlanan uygulamalar için işin çoğunu gerçekleştiren kuyruk yöneticisi araçları.
AMQZLSA0	Kuyruk yöneticisi aracı.
AMQZMUFO	Yardımcı Program Yöneticisi
AMQZMGRO	İşlem denetleyicisi. Bu iş, dinleyicileri ve hizmetleri başlatmak ve yönetmek için kullanılır.
AMQZMUR0	Yardımcı program yöneticisi. Bu iş, günlük zinciri yöneticisi gibi kritik kuyruk yöneticisi yardımcı programlarını yürütür.
AMQFQPUB	Kuyruğa alınan yayınlama/abone olma cini.
AMQFCXBA	Aracı işçi işi.
RUNMQBRK	Aracı denetim işi.
AMQRMPPA	Kanal işlemi havuzlama işi.
AMQCRSTA	TCP/IP ' yle çağrılan kanal yanıtlayıcı.
AMQCRS6B	LU62 alıcı kanalı ve istemci bağlantısı (nota bakın).

Çizelge 21. IBM MQ Görevler. (devamı var)

İş adı	İşlev
AMQRRMFA	Kümelere için havuz yöneticisi.
AMQCLMAA	İş parçacıklı olmayan TCP/IP dinleyicisi.
AMQPCSEA	PCF ve uzak denetim isteklerini işleyen PCF komut işlemcisi.
RUNMQTRM	Tetikleyici.
RUNMQDLQ	Gönderilmeyen ileti kuyruğu işleyicisi.
RUNMQCHI	Kanal başlatıcı.
RUNMQCHL	Her gönderen kanalı için başlatılan gönderen kanalı işi.
RUNMQLSR	İş parçacıklı TCP/IP dinleyicisi.
AMQRCMLA	Kanal MQSC ve PCF komut işlemcisi.

Not: LU62 günlük nesnesi işi, iletişim altsisteminde çalışır ve çalıştırma zamanı özelliklerini, işi başlatmak için kullanılan yönlendirme ve iletişim girişlerinden alır. Ek bilgi için [Initiated end \(Receiver\)](#) başlıklı konuya bakın.

IBM i IBM i üzerinde iş yönetimi nesneleri

IBM MQ kurulduğunda, iş yönetimine yardımcı olmak için QMQM kitaplığında çeşitli nesnelere sağlanır. Bu nesnelere, IBM MQ işlerinin kendi altsisteminde çalışması için gerekli nesnelere dir.

Örnek iş tanımlamaları, IBM MQ toplu işlerinden ikisi için sağlanır. Bir IBM MQ işi için belirli bir iş tanımlama sağlanmazsa, varsayılan iş tanımlama QMQMJOB ile çalışır.

IBM MQ kurulurken sağlanan iş yönetimi nesnelere Çizelge 22 sayfa 405 içinde listelenir ve bir kuyruk yöneticisi için oluşturulan nesnelere Çizelge 23 sayfa 406 içinde listelenir.

Not: İş yönetimi nesnelere QMQM kitaplığında bulunabilir ve kuyruk yöneticisi nesnelere kuyruk yöneticisi kitaplığında bulunabilir.

Çizelge 22. İş yönetimi nesnelere

Ad	Tip	Açıklama
AMQZLAA0	*JOB	IBM MQ aracı işlemlere tarafından kullanılan iş tanımlaması
AMQZLSA0	*JOB	Yalıtılmış bağ tanımlaması kuyruk yöneticisi aracısı
AMQZXMA0	*JOB	IBM MQ yürütme denetleyicilere tarafından kullanılan iş tanımlaması
QMQM	*SBS	Tüm IBM MQ işlerinin çalıştırıldığı altsistem
QMQM	*JOB	Sağlanan altsisteme bağlı iş kuyruğu
QMQMJOB	*JOB	Bir işe ilişkin belirli bir iş tanımlaması yoksa, varsayılan IBM MQ iş tanımlaması
QMQMMSG	*MSG	IBM MQ işlere için varsayılan ileti kuyruğu.
QMQMRUN20	*CLS	Yüksek öncelikli IBM MQ işlere ilişkin sınıf tanımlaması
QMQMRUN35	*CLS	Orta öncelikli IBM MQ işlere ilişkin sınıf tanımlaması
QMQMRUN50	*CLS	Düşük öncelikli IBM MQ işlere için sınıf tanımlaması

Çizelge 23. Bir kuyruk yöneticisi için yaratılan iş yönetimi nesneleri		
Ad	Tip	Açıklama
AMQA000000	*JRNRCV	Yerel günlük nesnesi
AMQAJRN	*JRN	Yerel günlük
AMQJRNINF	*USRSPC	Bir kuyruk yöneticisinin başlatılması ve ortam kurtarması için gereken en son günlük nesneleriyle güncellenen kullanıcı alanı. Bu kullanıcı alanı, hangi günlük nesnelerinin arşivlenmesi gerektiğini ve hangilerinin güvenli bir şekilde silinebildiğini belirlemek için bir uygulama tarafından sorgulanabilir.
AMQAJRNMSG	*MSGQ	Yerel günlük ileti kuyruğu
AMQCRC6B	*PGM	LU6.2 bağlantısını başlatmak için program
AMQRFOLD	*Kütük	Geçirilen kuyruk yöneticisi kanal tanımlama dosyası
QMQMMSG	*MSGQ	Kuyruk yöneticisi ileti kuyruğu

IBM i IBM MQ , IBM i üzerinde iş yönetimi nesneleri nasıl kullanılır?

Bu bilgiler, IBM MQ ' in iş yönetimi nesneleri nasıl kullandığını açıklar ve yapılandırma örnekleri sağlar.



Uyarı: QMQM altsistemindeki iş kuyruğu giriş ayarlarını, altsistemde izin verilen iş sayısını önceliğe göre sınırlayacak şekilde değiştirmeyin. Bunu yapmayı denerseniz, temel IBM MQ işlerinin sunulduktan sonra çalışmasını durdurabilir ve kuyruk yöneticisi başlatma işleminin başarısız olmasına neden olabilirsiniz.

İş yönetiminin nasıl yapılandırılacağını anlamak için öncelikle IBM MQ ' e ilişkin iş tanımlarının nasıl kullanıldığını anlamamız gerekir.

İş başlatmak için kullanılan iş tanımlaması, işin birçok özneliğini denetler. Örneğin:

- İşin kuyruğa alındığı ve işin hangi altsistemde çalıştırıldığı iş kuyruğu.
- İşin çalıştırma zamanı parametreleri için kullandığı işi ve sınıfı başlatmak için kullanılan yöneltme verileri.
- İşin yazdırma dosyaları için kullandığı çıkış kuyruğu.

IBM MQ işini başlatma işlemi üç adımda düşünülebilir:

1. IBM MQ bir iş tanımlaması seçer.

IBM MQ , bir toplu iş için hangi iş tanımlamasının kullanılacağını belirlemek için aşağıdaki tekniği kullanır:

- a. Kuyruk yöneticisi kitaplığında, işle aynı adı taşıyan bir iş tanımlaması arayın. Kuyruk yöneticisi kitaplığına ilişkin ek ayrıntılar için [IBM MQ for IBM i kuyruk yöneticisi kitaplık adlarının anlaşılması](#) başlıklı konuya bakın.
- b. QMQMJOBID varsayılan iş tanımı için kuyruk yöneticisi kitaplığına bakın.
- c. QMQM kitaplığında işle aynı adı taşıyan bir iş tanımlaması arayın.
- d. QMQM kitaplığındaki varsayılan iş tanımlamasını (QMQMJOBID) kullanın.

2. İş, iş kuyruğuna gönderilir.

IBM MQ ile sağlanan iş tanımlamaları, varsayılan olarak QMQM kitaplığındaki QMQM iş kuyruğuna iş yerleştirilecek şekilde ayarlandı. QMQM iş kuyruğu, sağlanan QMQM altsistemine bağlı olduğundan, varsayılan olarak işler QMQM altsisteminde çalışmaya başlar.

3. İş altsisteme girer ve yöneltme adımlarından geçer.

İş altsisteme girdiğinde, işe ilişkin yöneltme girişlerini bulmak için iş tanımlamasında belirlenen yöneltme verileri kullanılır.

Yöneltme verileri, QMQM altsisteminde tanımlanan yöneltme girişlerinden biriyle eşleşmelidir ve bu, iş tarafından sağlanan sınıflardan (QMQRUN20, QMQRUN35ya da QMQRUN50) hangisinin kullanıldığını tanımlar.

Not: IBM MQ işleri başlatıyor gibi görünmüyorsa, altsistemin çalıştığından ve iş kuyruğunun tutulmadığından emin olun.

IBM MQ iş yönetimi nesnelere değiştirdiyse, her şeyin doğru şekilde ilişkilendirildiğinden emin olun. Örneğin, iş tanımında QMQM/QMQM dışında bir iş kuyruğu belirtirseniz, altsistem için (QMQRUN) bir ADDJOBQE gerçekleştirildiğinden emin olun.

Aşağıdaki çalışma sayfasını örnek olarak kullanarak [Çizelge 21 sayfa 404](#) içinde belirlenen her iş için bir iş tanımlaması oluşturabilirsiniz:

```
What is the queue manager library name? _____
Does job description AMQZXMA0 exist in the queue manager library? Yes No
Does job description QMQMJOB exist in the queue manager library? Yes No
Does job description AMQZXMA0 exist in the QMQM library? Yes No
Does job description QMQMJOB exist in the QMQM library? Yes No
```

Tüm bu sorulara Hayır yanıtını verdiyseniz, QMQM kitaplığında bir genel iş tanımı QMQMJOB yaratın.

IBM MQ ileti kuyruğu

Her kuyruk yöneticisi kitaplığında bir IBM MQ ileti kuyruğu (QMQRMSG) yaratılır. Kuyruk yöneticisi işleri sona erdiğinde ve IBM MQ kuyruğa ileti gönderdiğinde, işletim sistemi iletileri bu kuyruğa gönderilir. Örneğin, başlatma sırasında hangi günlük nesnelere gerekli olduğunu bildirmek için. Bu ileti kuyruğundaki ileti sayısını, izlenmesini kolaylaştırmak için yönetilebilir bir boyutta tutun.

IBM i IBM i için varsayılan sistem örnekleri

Bu örnekler, standart işlerin bazıları kuyruk yöneticisi başlatma sırasında sunulduğunda, değiştirilmemiş bir IBM MQ kuruluşunun nasıl çalıştığını gösterir.

Önce, AMQZXMA0 yürütme denetleyicisi işi başlatılır.

1. Kuyruk yöneticisi TESTQM için **STRMQM** komutunu verin.
2. IBM MQ , önce AMQZXMA0 iş tanımı, daha sonra QMQMJOB iş tanımı için QMTESTQM kuyruk yöneticisi kitaplığını arar.

Bu iş tanımlamalarının hiçbiri bulunmamaktadır; bu nedenle IBM MQ , QMQM ürün kitaplığında AMQZXMA0 iş açıklamasını arar. Bu iş tanımlaması var, bu nedenle işi sunmak için kullanılır.
3. İş tanımlaması IBM MQ varsayılan iş kuyruğunu kullandığından, iş QMQM/QMQM iş kuyruğuna gönderilir.
4. AMQZXMA0 iş tanımındaki yöneltme verileri QMQRUN20 olduğundan, sistem bu verilerle eşleşen altsistem yöneltme girişlerini arar.

Varsayılan olarak, sıra numarası 9900 olan yöneltme girişinde QMQRUN20 ile eşleşen karşılaştırma verileri vardır; bu nedenle, iş o yöneltme girişinde tanımlanan sınıfla (QMQRUN20 olarak da adlandırılır) başlatılır.
5. QMQM/QMQRUN20 sınıfının çalışma önceliği 20 olarak ayarlandığı için AMQZXMA0 işi, sistemdeki en etkileşimli işlerle aynı önceliğe sahip QMQM altsisteminde çalışır.

IBM i IBM i üzerinde iş yönetimi örneklerinin yapılandırılması

IBM MQ işlerinin çalıştırma zamanı özniteliklerini değiştirmek için IBM MQ iş tanımlamalarını nasıl değiştirebileceğinizi ve yaratabileceğinizi öğrenmek için bu bilgileri kullanın.

IBM MQ iş yönetiminin esnekliğinin anahtarı, IBM MQ ' in iş tanımlarını aradığı iki katmanlı yöntemle bulunur:

- Bir kuyruk yöneticisi kitaplığında iş tanımlamaları yaratır ya da değiştirirseniz, bu değişiklikler QMQM ' deki genel iş tanımlamalarını geçersiz kılar, ancak değişiklikler yereldir ve yalnızca o kuyruk yöneticisini etkiler.
- QMQM kitaplığında genel iş tanımlamaları yaratır ya da değiştirirseniz, bu iş tanımlamaları, tek tek kuyruk yöneticileri için yerel olarak geçersiz kılınmadıkça, sistemdeki tüm kuyruk yöneticilerini etkiler.

1. Aşağıdaki örnek, tek bir kuyruk yöneticisine ilişkin kanal denetimi işlerinin önceliğini artırır.

Havuz yöneticisi ve kanal başlatıcı işlerini (AMQRRMFA ve RUNMQCHI), kuyruk yöneticisi TESTQM için mümkün olan en hızlı şekilde çalıştırmak üzere aşağıdaki adımları gerçekleştirin:

- a. Kuyruk yöneticisi kitaplığında denetlemek istediğiniz IBM MQ işlemlerinin adlarıyla QMQM/ QMQMJOBDB iş tanımının yerel yinemesini yaratın. Örneğin:

```
CRTDUPOBJ OBJ(QMQMJOBDB) FROMLIB(QMQM) OBJTYPE(*JOBDB) TOLIB(QMTESTQM)
NEWOBJ(RUNMQCHI)
CRTDUPOBJ OBJ(QMQMJOBDB) FROMLIB(QMQM) OBJTYPE(*JOBDB) TOLIB(QMTESTQM)
NEWOBJ(AMQRRMFA)
```

- b. İşlerin QMQMRUN20 sınıfını kullandığından emin olmak için iş tanımındaki yönlendirme verileri değiştirilmesini değiştirin.

```
CHGJOBDB JOBDB(QMTESTQM/RUNMQCHI) RTGDTA('QMQMRUN20')
CHGJOBDB JOBDB(QMTESTQM/AMQRRMFA) RTGDTA('QMQMRUN20')
```

Şu anda kuyruk yöneticisi TESTQM için AMQRRMFA ve RUNMQCHI işleri:

- Kuyruk yöneticisi kitaplığındaki yeni yerel iş tanımlamalarını kullan
- İşler altsisteme girdiğinde QMQMRUN20 sınıfı kullanıldığından, öncelik 20 ile çalıştırın.

2. Aşağıdaki örnek, QMQM altsistemi için yeni bir çalışma önceliği sınıfını tanımlar.

- a. Diğer kuyruk yöneticilerinin sınıfa erişmesine izin vermek için QMQM kitaplığında yinelenen bir sınıf yaratın:

```
CRTDUPOBJ OBJ(QMQMRUN20) FROMLIB(QMQM) OBJTYPE(*CLS) TOLIB(QMQM)
NEWOBJ(QMQMRUN10)
```

- b. Aşağıdaki komutu vererek sınıfı yeni çalışma önceliğine sahip olacak şekilde değiştirin:

```
CHGCLS CLS(QMQM/QMQMRUN10) RUNPTY(10)
```

- c. Aşağıdaki komutu vererek yeni sınıf tanımlamasını altsisteme ekleyin:

```
ADDRTGE SBSDB(QMQM/QMQM) SEQNBR(8999) CMPVAL('QMQMRUN10') PGM(QSYS/QCMD)
CLS(QMQM/QMQMRUN10)
```

Not: Yönlendirme sıra numarası için herhangi bir sayısal değer belirtebilirsiniz, ancak değerlerin sıralı olması gerekir. Bu sıra numarası, altsisteme yönlendirme girişlerinin bir yönlendirme verileri eşleşmesi için aranacağı sırayı bildirir.

- d. Aşağıdaki komutu vererek, yerel ya da genel iş tanımlamasını yeni öncelik sınıfını kullanacak şekilde değiştirin:

```
CHGJOBDB JOBDB(QMQMlibname/QMQMJOBDB) RTGDTA('QMQMRUN10')
```

Artık QMLibraryname ile ilişkili tüm kuyruk yöneticisi işleri 10 çalışma önceliğini kullanıyor.

3. Aşağıdaki örnek, kendi altsisteminde bir kuyruk yöneticisini çalıştırır

QBATCH altsisteminde TESTQM kuyruk yöneticisine ilişkin tüm işlerin çalışmasını sağlamak için aşağıdaki adımları gerçekleştirin:

- a. Komutla kuyruk yöneticisi kitaplığında QMQM/QMQMJOBDB iş tanımının yerel bir kopyasını yaratın.

```
CRTDUPOBJ OBJ(QMQMJOBDB) FROMLIB(QMQM) OBJTYPE(*JOBDB) TOLIB(QMTESTQM)
```

- b. İşlerin QBATCH iş kuyruğunu kullandığından emin olmak için iş tanımlamasındaki iş kuyruğu parametresini değiştirin.

```
CHGJOBDB JOBDB(QMTESTQM/QMQMJOBDB) JOBQ(*LIBL/QBATCH)
```

Not: İş kuyruğu, altsistem tanımlamasıyla ilişkilendirilir. İşlerin iş kuyruğunda kaldığını bulursanız, iş kuyruğu tanımının SBSDB 'de tanımlı olduğunu doğrulayın. Altsistem için DSPSBSDB komutunu kullanın ve seçenek 6, İş kuyruğu girişlerini alın.

TESTQM kuyruk yöneticisine ilişkin tüm işler şimdi:

- Kuyruk yöneticisi kitaplığındaki yeni yerel varsayılan iş tanımlamasını kullan
- QBATCH iş kuyruğuna gönderilir.

İşlerin doğru şekilde yönlendirildiğinden ve önceliklendirildiğinden emin olmak için:

- QBATCH altsistemindeki IBM MQ işleri için yöneltme girişleri yaratın ya da
- Hangi yöneltme verilerinin kullanıldığından bağımsız olarak, QCMD 'yi çağıran bir tümünü yakala yöneltme girişine güvenin.

Bu seçenek yalnızca, QBATCH iş kuyruğuna ilişkin etkin iş sayısı üst sınırı seçeneği *NOMAX olarak ayarlandığında çalışır. Sistemin varsayılan değeri 1 'dir.

4. Aşağıdaki örnek başka bir IBM MQ altsistemi yaratır

- a. Aşağıdaki komutu vererek QMQM kitaplığında yinelenen bir altsistem yaratın:

```
CRTDUPOBJ OBJ(QMQM) FROMLIB(QMQM) OBJTYPE(*SBSDB) TOLIB(QMQM) NEWOBJ(QMQM2)
```

- b. Aşağıdaki komutu vererek QMQM iş kuyruğunu kaldırın:

```
RMVJOBQE SBSDB(QMQM/QMQM2) JOBQ(QMQM/QMQM)
```

- c. Aşağıdaki komutu vererek altsistem için yeni bir iş kuyruğu yaratın:

```
CRTJOBQ JOBQ(QMQM/QMQM2) TEXT('Job queue for IBM MQ Queue Manager')
```

- d. Aşağıdaki komutu vererek altsisteme bir iş kuyruğu girişi ekleyin:

```
ADDJOBQE SBSDB(QMQM/QMQM2) JOBQ(QMQM/QMQM2) MAXACT(*NOMAX)
```

- e. Aşağıdaki komutu vererek, kuyruk yöneticisi kitaplığında yinelenen bir QMQMJOBDB yaratın:

```
CRTDUPOBJ OBJ(QMQMJOBDB) FROMLIB(QMQM) OBJTYPE(*JOBDB) TOLIB(QMlibraryname)
```

- f. Aşağıdaki komutu vererek, iş tanımlamasını yeni iş kuyruğunu kullanacak şekilde değiştirin:

```
CHGJOBDB JOBDB(QMlibraryname/QMQMJOBDB) JOBQ(QMQM/QMQM2)
```

- g. Aşağıdaki komutu vererek altsistemi başlatın:

```
STRSBS SBSDB(QMQM/QMQM2)
```

Not:

- Altsistemi herhangi bir kitaplıkta belirleyebilirsiniz. Herhangi bir nedenle ürün yeniden kurulursa ya da QMQM kitaplığı değiştirilirse, yaptığınız değişiklikler kaldırılır.
- QMLibraryname ile ilişkili tüm kuyruk yöneticisi işleri artık QMQM2altsistemi altında çalışır.

IBM i IBM i üzerinde kullanılabilirlik, yedekleme, kurtarma ve yeniden başlatma

IBM MQ for IBM i 'in yedekleme ve geri yükleme stratejisine yardımcı olmak için IBM i günlük kaydı desteğini nasıl kullandığını anlamak için bu bilgileri kullanın.

Bu bölümü okumadan önce, standart IBM i yedekleme ve kurtarma yöntemlerini ve IBM i üzerindeki günlüklerin ve bunlarla ilişkili günlük nesnelere kullanımını bilmeniz gerekir. Bu konularla ilgili bilgi için [Yedekleme ve kurtarmabaşlıklı](#) konuya bakın.

Yedekleme ve kurtarma stratejisini anlamak için öncelikle IBM MQ for IBM i 'un verilerini IBM i dosya sisteminde ve tümleşik dosya sisteminde (IFS) nasıl düzenlediğini anlamanız gerekir.

IBM MQ for IBM i , verilerini her kuyruk yöneticisi eşgörunümü için ayrı bir kitaplıkta ve IFS dosya sistemindeki akış dosyalarında tutar.

Kuyruk yöneticisine özgü kitaplıklar, kuyruk yöneticisinin iş yönetimini denetlemek için gereken günlükleri, günlük nesnelere ve nesnelere içerir. IFS dizinleri ve dosyaları, IBM MQ yapılandırma dosyalarını, IBM MQ nesnelere açıklamalarını ve bunların içerdiği verileri içerir.

Bir sistem arızasında kurtarılabilir olan bu nesnelere yapılan her değişiklik, uygun nesneye uygulanmadan önce bir günlüğe kaydedilir. Bu, günlükte kaydedilen bilgiler yeniden yürütülerek bu tür değişikliklerin kurtarılabilmesi için bir etkiye sahiptir.

IBM MQ for IBM i 'i farklı sunucularda birden çok kuyruk yöneticisi yönetim ortamı kullanacak şekilde yapılandırılabilir ve bir sunucu ya da kuyruk yöneticisi arızası durumunda daha fazla kuyruk yöneticisi kullanılabilirliği ve kurtarmayı hızlandırabilirsiniz.

IBM i IBM i üzerindeki kuyruk yöneticisi günlükleri

IBM MQ for IBM i 'in yerel nesnelere ilişkin güncellemeleri denetlemek için işletimdeki günlükleri nasıl kullandığını anlamak için bu bilgileri kullanın.

Her kuyruk yöneticisi kitaplığı o kuyruk yöneticisine ilişkin bir günlük içerir; günlük QM *GRLIB/AMQ A* JRNadını taşır; burada QM *GRLIB* , kuyruk yöneticisi kitaplığının adıdır; *A* , tek bir eşgörunüm kuyruk yöneticisi söz konusu olduğunda *A* , kuyruk yöneticisi eşgörunümü için benzersiz olan bir harftir.

QM *GRLIB* , QMadını alır ve ardından benzersiz bir formda kuyruk yöneticisinin adını alır. Örneğin, TEST adlı bir kuyruk yöneticisinin QMTESTadlı bir kuyruk yöneticisi kitaplığı vardır. Kuyruk yöneticisi kitaplığı, **CRTMQM** komutu kullanılarak bir kuyruk yöneticisi yaratılırken belirtilebilir.

Günlüklerin, günlüğe kaydedilmekte olan bilgileri içeren ilişkili günlük nesnelere vardır. Günlük nesnelere, bilgilerin yalnızca sonuna eklenebildiği ve sonunda doldurulabileceği nesnelere.

Günlük nesnelere, güncel olmayan bilgilerle değerli disk alanı kullanır. Ancak, bu sorunu en aza indirmek için bilgileri kalıcı depolamaya yerleştirebilirsiniz. Bir günlük nesnesi belirli bir zamanda günlüğe eklenir. Günlük nesnesi önceden belirlenen eşik büyüklüğüne ulaşırsa, bu nesne ayrılır ve yeni bir günlük nesnesiyle değiştirilir. **CRTMQM** ve **THRESHOLD** değiştirgesini kullanarak bir kuyruk yöneticisi yarattığınızda günlük nesnelere eşikini belirleyebilirsiniz.

Yerel IBM MQ for IBM i günlüğüyle ilişkilendirilmiş günlük nesnelere her kuyruk yöneticisi kitaplığında bulunur ve aşağıdaki gibi bir adlandırma kuralı benimser:

```
AMQ Arnnnnn
```

burada:

A

bir harf A-Z. Bu, tek eşgörünüm kuyruk yöneticileri için A ' dir. Çok eşgörünümlü bir kuyruk yöneticisinin farklı eşgörünümlerine göre değişir.

nnnnn

Sıradaki sonraki günlük için 1 artırılan ondalık 00000 to 99999 değeridir.

:NONE.

Bir günlük nesnesinin her geri yüklenişinde 1 artırılan ondalık 0 to 9değeridir.

Günlüklerin sırası tarihe dayanır. Ancak, sonraki günlüğün adlandırılması aşağıdaki kurallara bağlıdır:

1. AMQAınnnnn , AMQAı (nnnnn+1) adresine gider ve nnnnn , 99999' e ulaştığında kayar. Örneğin, AMQA099999 , AMQA000000adresine gider ve AMQA999999 , AMQA900000adresine gider.
2. Kural 1 tarafından oluşturulan bir günlük zaten varsa, CPI70E3 iletisi QSYSOPR iletisi kuyruğuna gönderilir ve otomatik günlük nesnesi geçişi durur.

Şu anda bağlı olan günlük nesnesi, siz sorunu inceleyinceye ve yeni bir günlük nesnesini el ile bağlanmaya kadar kullanılmaya devam eder.

3. Sırada yeni bir ad yoksa (yani, sistemde tüm olası günlük adları varsa), aşağıdakilerden ikisini de yapmanız gerekir:

- a. Artık gerekli olmayan günlükleri silin (bkz. ["IBM i üzerinde günlük yönetimi"](#) sayfa 415).
- b. Günlük değişikliklerini en son günlük nesnesine kaydetmek için: (**RCDMQMIMG**) ve daha sonra, önceki adımı yineleyin. Bu, eski günlük nesnesi adlarının yeniden kullanılmasını sağlar.

AMQAJRN günlüğü, eşığe ulaşıldığında işletim sisteminin günlük nesnelerini otomatik olarak değiştirmesini sağlamak için MNGRCV (*SYSTEM) seçeneğini kullanır. Sistemin günlük nesnelerini nasıl yönettiğine ilişkin daha fazla bilgi için bkz. *IBM i Yedekleme ve Kurtarma*.

Günlük nesnesinin varsayılan eşik değeri 100.000 KB 'dir. Kuyruk yöneticisini yaratırken bu değeri daha büyük bir değere ayarlayabilirsiniz. LogReceiverSize öznitelikinin başlangıç değeri, mq.s.ini dosyasının LogDefaults kısmına yazılır.

Bir günlük nesnesi belirlenen eşığı aştığında, nesne ayrılır ve yeni bir günlük nesnesi yaratılır; bu nesne, öznitelikleri önceki günlük nesnesinden devralır. Bir kuyruk yöneticisi yaratıldıktan sonra LogReceiverSize ya da LogASP özniteliklerinde yapılan değişiklikler, sistem otomatik olarak yeni bir günlük nesnesi eklediğinde yoksayılır.

Sistemi yapılandırmaya ilişkin daha fazla ayrıntı için bkz. [Çoklu Platformda IBM MQ yapılandırma bilgilerinin değiştirilmesi](#) .

Kuyruk yöneticisi yaratıldıktan sonra günlük nesnelerinin büyüklüğünü değiştirmeniz gerekirse, yeni bir günlük nesnesi yaratın ve aşağıdaki komutları kullanarak günlük nesnesinin iyesini QMQM olarak ayarlayın:

```
CRTJRNRCV JRNRCV(QM GRLIB/AMQ Arnnnnn) THRESHOLD(xxxxxx) +
TEXT('MQM LOCAL JOURNAL RECEIVER')
CHGOBJOWN OBJ(QM GRLIB/AMQ Arnnnnn) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
```

burada:

QMGRLIB

Kuyruk yöneticisi kitaplığınızın adı

A

Eşgörünüm tanıtıcısıdır (genellikle A).

rnnnnn

Daha önce açıklanan adlandırma sırasındaki sonraki günlük nesnesi

xxxxxx

Yeni günlük nesnesi eşığı (KB)

Not: Alıcının büyüklük üst sınırı işletim sistemi tarafından yönetilir. Bu değeri denetlemek için **CRTJRNRCV** komutundaki THRESHOLD anahtar sözcüğüne bakın.

Yeni günlük nesnesini aşağıdaki komutla AMQAJRN günlüğüne ekleyin:

```
CHGJRN JRN(QMGLIB/AMQ A JRN) JRNRCV(QMGLIB/AMQ Annnnnn)
```

Bu günlük nesnelere ilişkin ayrıntılar için bkz. [“IBM i üzerinde günlük yönetimi” sayfa 415](#).

IBM i **IBM i üzerinde kuyruk yöneticisi günlük kullanımı**

IBM MQ for IBM i 'in yerel nesnelere ilişkin güncellemeleri denetlemek için işletimdeki günlükleri nasıl kullandığını anlamak için bu bilgileri kullanın.

İleti kuyruklarında kalıcı güncellemeler iki aşamada gerçekleşir. Güncellemeyi gösteren kayıtlar önce günlüğe yazılır, daha sonra kuyruk dosyası güncellenir.

Bu nedenle günlük nesnelere, kuyruk dosyalarından daha güncel olabilir. Yeniden başlatma işleminin tutarlı bir noktadan başladığından emin olmak için IBM MQ denetim noktalarını kullanın.

Denetim noktası, günlükte açıklanan kaydın kuyruktaki kayıtla aynı olduğu bir zaman noktasıdır. Denetim noktasının kendisi, kuyruk yöneticisini yeniden başlatmak için gereken günlük kayıtları dizisinden oluşur. Örneğin, denetim noktası sırasında etkin olan tüm işlemlerin (yani iş birimlerinin) durumu.

Denetim noktaları IBM MQ tarafından otomatik olarak oluşturulur. Bunlar, kuyruk yöneticisi başlatıldığında ve kapandığında ve belirli sayıda işlem günlüğe kaydedildikten sonra alınır.

Bir kuyruk yöneticisini, bir kuyruk yöneticisindeki tüm nesnelere için RCDMQMIMG komutunu vererek ve sonuçları aşağıdaki gibi görüntüleyerek denetim noktası almaya zorlayabilirsiniz:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNTA(*YES)
```

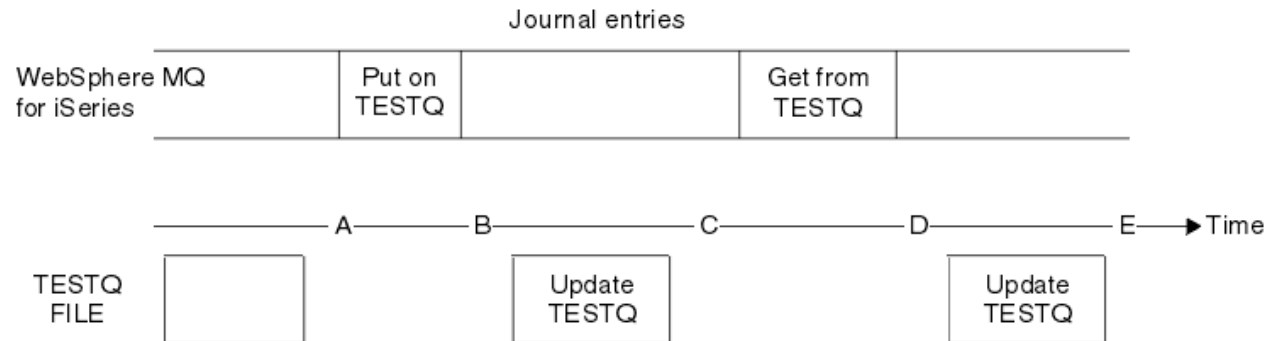
Kuyruklar daha fazla iletiyi işlerken, denetim noktası kaydı kuyrukların yürürlükteki durumuyla tutarsız olur.

IBM MQ yeniden başlatıldığında, günlükteki en son denetim noktası kaydını bulur. Bu bilgiler, her denetim noktasının sonunda güncellenen denetim noktası dosyasında tutulur. Denetim noktası kaydı, günlük ile veriler arasındaki en son tutarlılık noktasını gösterir. Bu denetim noktasındaki veriler, kuyrukları denetim noktası zamanında olduğu gibi yeniden oluşturmak için kullanılır. Kuyruklar yeniden yaratıldığında, kuyrukları sistem arızasından ya da kapanmadan önce buldukları duruma geri getirmek için günlük ileriye doğru yürütülür.

IBM MQ 'in günlüğü nasıl kullandığını anlamak için, kuyruk yöneticisinde TESTQ adlı yerel bir kuyruğun durumunu göz önünde bulundurun TEST. Bu, IFS dosyası tarafından temsil edilir:

```
/QIBM/UserData/mqm/qmgrs/TEST/queues
```

Belirtilen bir ileti bu kuyruğa konup kuyruktan alınırsa, gerçekleşen işlemler Şekil 22 sayfa 412’de gösterilir.



Şekil 22. MQM nesnelere güncellenirken olayların sırası

Çizgede gösterilen beş nokta (A-E), aşağıdaki durumları tanımlayan zaman noktalarını gösterir:

- A** Kuyruğun IFS dosyası gösterimi, günlükte bulunan bilgilerle tutarlı.
- B** Günlük girişi, kuyruktaki bir Put işlemini tanımlayan günlüğe yazılır.
- C** Kuyruk için uygun güncelleme yapılır.
- D** Günlük girişi, kuyruktan bir Get işlemini tanımlayan günlüğe yazılır.
- E** Kuyruk için uygun güncelleme yapılır.

IBM MQ for IBM i kurtarma yeteneklerinin anahtarı, kullanıcının TESTQ 'nun IFS dosyası gösterimini Azamanında olduğu gibi kaydedebilmesidir ve daha sonra, kaydedilen nesneyi geri yükleyerek ve A zaman içinde günlükteki girişleri yeniden yürüterek TESTQ' nun IFS dosyası gösterimini Ezamanında olduğu gibi kurtarabilmesidir.

Bu strateji, sistem hatasından sonra kalıcı iletileri kurtarmak için IBM MQ for IBM i tarafından kullanılır. IBM MQ , günlük nesnelere belirli bir giriş hatırlar ve başlatma sırasında günlüklerdeki girişleri bu noktadan itibaren yeniden çalmasını sağlar. Bu başlangıç girdisi, IBM MQ ' in yalnızca sonraki başlatma sırasında gerekli en düşük kaydı gerçekleştirilmesi gerektiği şekilde düzenli olarak yeniden hesaplanır.

IBM MQ , nesnelere tek tek kurtarılmasını sağlar. Bir nesneyle ilgili tüm kalıcı bilgiler yerel IBM MQ for IBM i günlüklerine kaydedilir. Zarar gören ya da bozulmuş herhangi bir IBM MQ nesnesi, günlükte tutulan bilgilerden tamamen yeniden oluşturulabilir.

Sistemin günlük nesnelere nasıl yönettiğine ilişkin ek bilgi için bkz. [“IBM i üzerinde kullanılabilirlik, yedekleme, kurtarma ve yeniden başlatma” sayfa 410.](#)

IBM i *IBM i üzerindeki ortam görüntüleri*

IBM i üzerinde ortam görüntüsü, günlüğe kaydedilen IBM MQ nesnesinin tam bir kopyasıdır. Bazı bozuk ya da hasarlı nesnelere, ortam görüntülerinden otomatik olarak kurtarılabilir.

Uzun süreli bir IBM MQ nesnesi, yaratıldığı noktaya geri dönen çok sayıda günlük girişini temsil edebilir. Bunu önlemek için IBM MQ for IBM i , bir nesnenin ortam görüntüsü kavramına sahiptir.

Bu ortam görüntüsü, günlüğe kaydedilen IBM MQ nesnesinin tam bir kopyasıdır. Bir nesnenin resmi alınır, nesne, bu görüntüden başlayarak günlük girişleri yeniden oynanarak yeniden oluşturulabilir. Günlükte, her IBM MQ nesnesi için yeniden yürütme noktasını gösteren giriş ortam kurtarma girişi denir. IBM MQ aşağıdaki işlemleri izler:

- Her kuyruk yöneticisi nesnesi için ortam kurtarma girişi.
- Bu kümedeki en eski giriş (ayrıntılar için [“IBM i üzerinde günlük yönetimi” sayfa 415](#) içindeki AMQ7462 hata ilemesine bakın).

*CTLG nesnesinin ve *MQM nesnesinin görüntüleri, kuyruk yöneticisinin yeniden başlatılması için çok önemli olduğundan düzenli olarak alınır.

Uygun olduğunda diğer nesnelere resimleri alınır. Varsayılan olarak, bir kuyruk yöneticisi ENDCCTJOB (*YES) parametresiyle **ENDMQM** komutu kullanılarak kapatıldığında tüm nesnelere görüntüleri alınır. Bu işlem çok büyük kuyruk yöneticileri için çok uzun sürebilir. Hızlı bir şekilde sona erdirmeniz gerekiyorsa, ENDCCTJOB (*YES) parametresiyle RCDMQMIMG (*NO) parametresini belirleyin. Bu tür durumlarda, kuyruk yöneticisi yeniden başlatıldıktan sonra günlüklere tam bir ortam görüntüsü kaydetmeniz önerilir. Bu durumda, aşağıdaki komutu kullanarak:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME)
```

IBM MQ , bir nesnenin resmini, bir nesnenin günlükteki küçük bir girişle tam olarak tanımlanabileceği uygun bir nokta bulursa otomatik olarak kaydeder. Ancak bu, bazı nesnelere (örneğin, sürekli olarak çok sayıda ileti içeren kuyruklar) için hiçbir zaman gerçekleşmeyebilir.

En eski ortam kurtarma girişinin tarihinin gereksiz yere uzun bir süre devam etmesine izin vermek yerine, seçilen nesnelere görüntüsünü el ile almanızı sağlayan IBM MQ komutu RCDMQMIMG kullanın.

Ortam görüntülerinden kurtarma

IBM MQ , bozuk ya da hasarlı olduğu saptandığında ortam görüntülerinden bazı nesnelere otomatik olarak kurtarır. Bu, olağan kuyruk yöneticisi başlatmasının bir parçası olarak özel *MQM ve *CTLG nesnelere için geçerlidir. Kuyruk yöneticisinin son kapatıldığı sırada herhangi bir eşitleme noktası hareketi tamamlanmamışsa, başlatma işlemini tamamlamak için etkilenen herhangi bir kuyruk da otomatik olarak kurtarılır.

IBM MQ komutunu RCRMQMOBJK kullanarak diğer nesnelere el ile kurtarmanız gerekir. Bu komut, IBM MQ nesnesini yeniden yaratmak için günlükteki girişleri yeniden yürütür. Bir IBM MQ nesnesi zarar görürse, geçerli işlemler yalnızca nesneyi silmek ya da bu yöntemle yeniden yaratmak olur. Ancak, kalıcı olmayan iletilerin bu şekilde kurtarılamayacağını unutmayın.

IBM i **IBM MQ for IBM i üzerindeki denetim noktaları**

Denetim noktaları, kurtarma için bilinen tutarlı bir başlangıç noktası sağlamak üzere çeşitli zamanlarda alınır.

AMQZMUC0 işlemi içindeki denetim noktası iş parçacığı, aşağıdaki noktalarda denetim noktasını almaktan sorumludur:

- Kuyruk yöneticisi başlatma (STRMQM).
- Kuyruk yöneticisi kapatma (ENDMQM).
- Son denetim noktasından (varsayılan dönem 30 dakikadır) ve önceki denetim noktasından (varsayılan değer 100 'dür) bu yana günlük kaydı sayısı alt sınırı yazıldıktan sonra bir süre geçtikten sonra.
- Bir dizi günlük kaydı yazıldıktan sonra. Varsayılan değer 10 000 'dir.
- Günlük eşiği büyüklüğü aşıldıktan ve yeni bir günlük nesnesi otomatik olarak yaratıldıktan sonra.
- Tam ortam görüntüsü aşağıdakilerle alındığında:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNDTA(*YES)
```

IBM i **IBM MQ for IBM i verilerinin yedeklenmesi**

Her bir kuyruk yöneticisine ilişkin iki IBM MQ yedekleme tipini anlamak için bu bilgileri kullanın.

Her kuyruk yöneticisi için dikkate alınması gereken iki tip IBM MQ yedekleme vardır:

- Veri ve günlük yedekleme.

Her iki veri kümesinin de tutarlı olduğundan emin olmak için, bunu yalnızca kuyruk yöneticisini kapattıktan sonra yapın.

- Günlük yedekleme.

Kuyruk yöneticisi etkinken bunu yapabilirsiniz.

Her iki yöntem için de, kuyruk yöneticisi IFS dizininin ve kuyruk yöneticisi kitaplığının adlarını bulmanız gerekir. Bunları IBM MQ yapılandırma dosyasında (mqc.ini) bulabilirsiniz. Daha fazla bilgi için bkz. [Çoklu Platformlar 'da IBM MQ yapılandırma bilgilerini değiştirme](#).

Her iki tip yedekleme için aşağıdaki yordamları kullanın:

Belirli bir kuyruk yöneticisinin veri ve günlük yedeklemesi

Not: Kuyruk yöneticisi çalışırken etkinken saklama isteği kullanmayın. Bekleyen değişiklikleri olan tüm kesinleştirme tanımları kesinleştirilmedikçe ya da geriye işlenmedikçe, böyle bir istek

tamamlanamaz. Kuyruk yöneticisi etkinken bu komut kullanılırsa, kanal bağlantıları olağan şekilde sona ermeyebilir. Her zaman aşağıdaki yordamı kullanın.

1. Şu komutu kullanarak boş bir günlük nesnesi yaratın:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. **RCDMQMIMG** komutunu kullanarak tüm IBM MQ nesneleri için bir MQM görüntüsü kaydedin ve sonra şu komutu kullanarak bir denetim noktası zorlayın:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. Kanalları sona erdirin ve kuyruk yöneticisinin çalışmadığından emin olun. Kuyruk yöneticiniz çalışıyorsa, **ENDMQM** komutuyla durdurun.
4. Aşağıdaki komutu vererek kuyruk yöneticisi kitaplığını yedekleyin:

```
SAVLIB LIB(QMTEST)
```

5. Aşağıdaki komutu vererek kuyruk yöneticisi IFS dizinlerini yedekleyin:

```
SAV DEV(...) OBJ(('QIBM/UserData/mqm/qmgrs/test'))
```

Belirli bir kuyruk yöneticisinin günlük yedeklemesi

Tüm ilgili bilgiler günlüklerde tutulduğundan, bir süre tam saklama işlemi gerçekleştirdiğiniz sürece, günlük nesnelere saklanmasıyla kısmi yedekleme gerçekleştirilebilir. Bu, tam yedekleme zamanından bu yana yapılan tüm değişiklikleri kaydeder ve aşağıdaki komutlar çalıştırılarak gerçekleştirilir:

1. Şu komutu kullanarak boş bir günlük nesnesi yaratın:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. **RCDMQMIMG** komutunu kullanarak tüm IBM MQ nesneleri için bir MQM görüntüsü kaydedin ve sonra şu komutu kullanarak bir denetim noktası zorlayın:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. Günlük nesnelere saklamak için aşağıdaki komutu kullanın:

```
SAVOBJ OBJ(AMQ*) LIB(QMTEST) OBJTYPE(*JRNRCV) .....
```

Basit bir yedekleme stratejisi, IBM MQ kitaplıklarının her hafta tam olarak yedeklenmesini ve günlük yedekleme gerçekleştirilmesini sağlar. Bu, elbette, kuruluşunuz için yedekleme stratejinizi nasıl oluşturduğuna bağlıdır.

IBM i *IBM i üzerinde günlük yönetimi*

Yedekleme stratejinizin bir parçası olarak günlük nesnelere ilgilenin. Çeşitli nedenlerden ötürü IBM MQ kitaplıklarından günlük nesnelere kaldırmak yararlı olur:

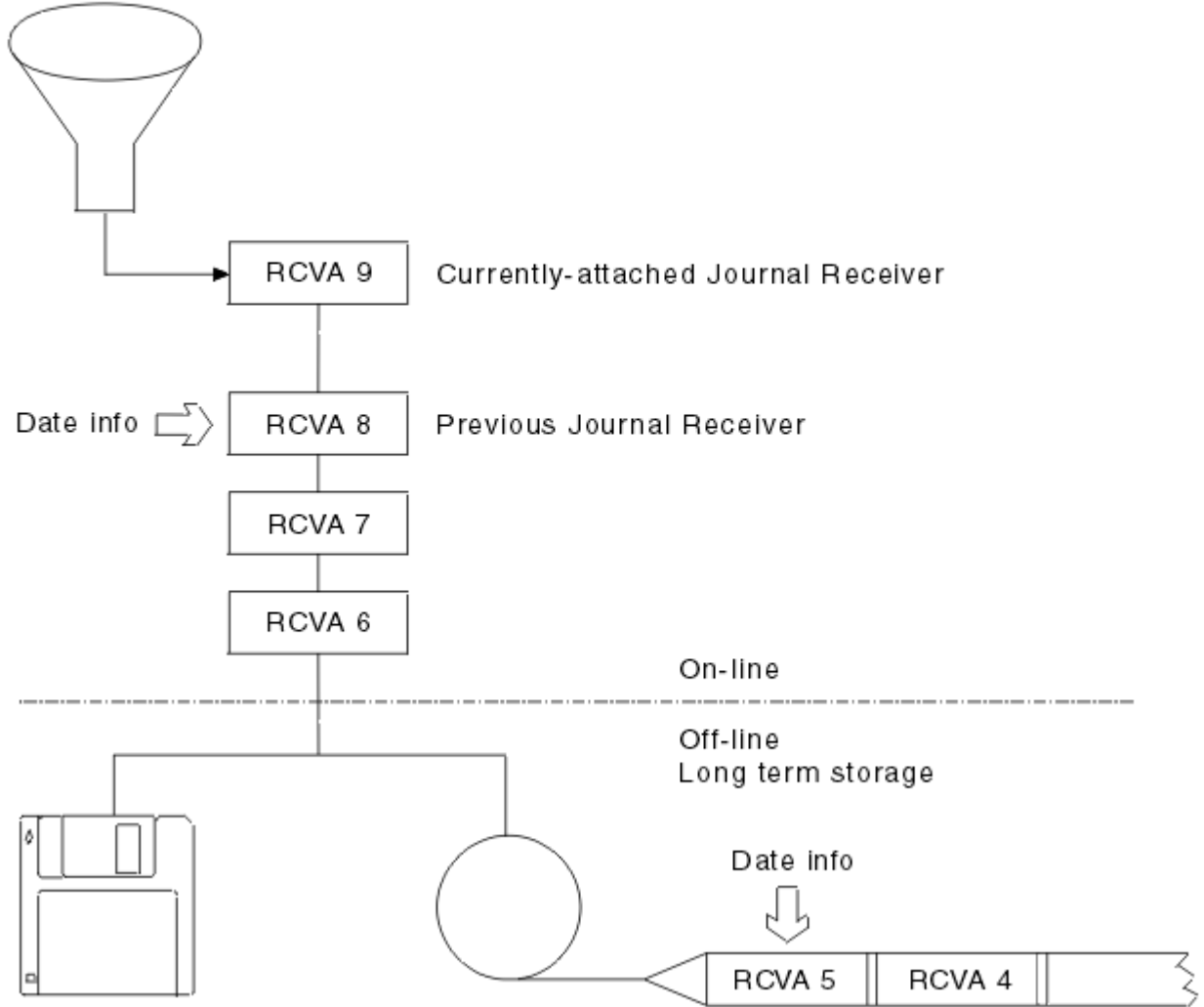
- Alanı serbest bırakmak için; bu, tüm günlük nesnelere için geçerlidir
- Başlatılırken performansı artırmak için (STRMQM)
- Nesnelere yeniden yaratma başarımını artırmak için (RCRMQMOBJ)

Bir günlük nesnesini silmeden önce, bir yedek kopyanızı almanız ve günlük nesnesine artık gerek duymamanız gerekir.

Günlük nesneleri, günlüklerden ayrıldıktan sonra kuyruk yöneticisi kitaplığından kaldırılabilir ve kurtarma işlemi için gerekirse geri yüklenebilmeleri koşuluyla saklanabilirler.

Günlük yönetimi kavramı Şekil 23 sayfa 416 içinde gösterilmektedir.

WebSphere MQ for ISeries Journal



Şekil 23. IBM i üzerinde günlük kaydı

Yedeklenen bir günlük nesnesinin kuyruk yöneticisi kitaplığından ne zaman kaldırılabilceğini ve yedeklemenin kendisinin ne zaman atılabileceğini belirlemek için IBM MQ günlüklerinde ne kadar geriye gitmesi gerektiğini bilmek önemlidir.

IBM MQ , bu süreyi belirlemeye yardımcı olmak için kuyruk yöneticisi ileti kuyruğuna (kuyruk yöneticisi kitaplığındaki QMQMSG) iki ileti gönderir. Bu iletiler başlatıldığında, yerel günlük nesnesini değiştirdiğinde ve denetim noktasını zorlamak için RCDMQIMG ' yi kullandığınızda yayınlanır. İki ileti şunlardır:

AMQ7460

Başlangıç kurtarma noktası. Bu ileti, başlatma kurtarma geçişi başarılı olduğunda IBM MQ ' un günlüğü yeniden kayıttan başlatması için kullanılacak başlangıç girişinin tarihini ve saatini tanımlar. Bu kaydı içeren günlük nesnesi IBM MQ kitaplıklarında varsa, bu ileti kaydı içeren günlük nesnesinin adını da içerir.

- AMQ7462 içindeki günlük nesnesi gelişmiş değilse, bir ya da daha çok zarar görmüş nesne olduğunu gösterir.

IBM i **IBM i üzerinde tam bir kuyruk yöneticisinin (veri ve günlükler) geri yüklenmesi**

Bir ya da daha çok kuyruk yöneticisi bir yedekten ya da uzak bir makineden geri yüklemek için bu bilgileri kullanın.

Bir ya da daha çok IBM MQ kuyruk yöneticisini bir yedekten kurtarmanız gerekirse, aşağıdaki adımları gerçekleştirin.

1. IBM MQ kuyruk yöneticilerini susturun.
2. En son tam yedekleme ve daha sonra yedeklenen günlük nesnelere oluşan en son yedekleme kümesini bulun.
3. IBM MQ veri kitaplıklarını tam yedekleme sırasında geri yüklemek için, tam yedeklemeden bir RSTLIB işlemi gerçekleştirin ve aşağıdaki komutları verin:

```
RSTLIB LIB(QMQLIB1) .....
RSTLIB LIB(QMQLIB2) .....
```

Günlük nesnesi kısmen bir günlük yedekinde saklandıysa ve sonraki bir yedekte tam olarak saklandıysa, yalnızca tam olarak saklanana geri yükleyin. Günlükleri kronolojik sırayla tek tek geri yükleyin.

4. Aşağıdaki komutu kullanarak IBM MQ IFS dizinlerini IFS dosya sistemine geri yüklemek için bir RST işlemi gerçekleştirin:

```
RST DEV(...) OBJ(('QIBM/UserData/mqm/qmgrs/testqm')) ...
```

5. İleti kuyruğu yöneticisini başlatın. Bu işlem, tam yedeklemeden bu yana yazılan tüm günlük kayıtlarını yeniden oynatır ve günlük yedekleme sırasında tüm IBM MQ nesnelere tutarlı duruma geri yükler.

Tam bir kuyruk yöneticisini farklı bir makineye geri yüklemek istiyorsanız, kuyruk yöneticisi kitaplığındaki her şeyi geri yüklemek için aşağıdaki yordamı kullanın. (Örnek kuyruk yöneticisi adı olarak TEST kullanılır.)

1. CRTMQM TEST
2. DLTLIB LIB(QMTEST)
3. RSTLIB SAVLIB(QMTEST) DEV(*SAVF) SAVF(QMGRLIBSAV)
4. Şu IFS dosyalarını sil:

```
/QIBM/UserData/mqm/qmgrs/TEST/QMQMCHKPT
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMQMOBJCAT
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMANAGER
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.AUTH.DATA.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CHANNEL.INITQ/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.COMMAND.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.REPOSITORY.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.TRANSMIT.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.PENDING.DATA.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.ADMIN.COMMAND.QUEUE/q
```

5. STRMQM TEST
6. RCRMQMOBJ OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(TEST)

IBM i **IBM i üzerinde belirli bir kuyruk yöneticisine ilişkin günlük nesnelere geri yüklenmesi**

Günlük nesnelere geri yüklemenin farklı yollarını anlamak için bu bilgileri kullanın.

En yaygın işlem, sonraki bir kurtarma işlevi için kaldırılan bir günlük nesnesinin yeniden gerekmesi durumunda, yedeklenen günlük nesnesinin bir kuyruk yöneticisi kitaplığına geri yüklenmesini sağlar.

Bu basit bir görevdir ve günlük nesnelerinin standart IBM i RSTOBJ komutu kullanılarak geri yüklenmesini gerektirir:

```
RSTOBJ OBJ(QMQMDATA/AMQA000005) OBJTYPE(*JRNRCV) .....
```

Bir dizi günlük nesnesinin tek bir günlük nesnesi yerine geri yüklenmesi gerekebilir. Örneğin, AMQA000007 , IBM MQ kitaplıklarındaki en eski günlük nesnesidir ve hem AMQA000005 hem de AMQA000006 nesnesinin geri yüklenmesi gerekir.

Bu durumda, günlük nesnelerini ters kronolojik sırada tek tek geri yükleyin. Bu her zaman gerekli değildir, ama iyi bir uygulamadır. Önemli durumlarda, geri yüklenen günlük nesnelerini günlükle ilişkilendirmek için IBM i komutu WRKJRNA kullanmanız gerekebilir.

Günlükleri geri yüklerken sistem, günlük nesnesi sırasına yeni bir ad eklenmiş bir günlük nesnesini otomatik olarak yaratır. Ancak, oluşturulan yeni ad, geri yüklemeniz gereken bir günlük nesnesiyle aynı olabilir. Bu sorunun üstesinden gelmek için el ile müdahale gerekir; sırayla yeni bir ad günlüğü nesnesi ve günlük nesnesini geri yüklemeyen önce yeni bir günlük nesnesi yaratmak için.

Örneğin, kaydedilen günlük AMQAJRN ve aşağıdaki günlük nesneleriyle ilgili sorunu göz önünde bulundurun:

- AMQA000000
- AMQA100000
- AMQA200000
- AMQA300000
- AMQA400000
- AMQA500000
- AMQA600000
- AMQA700000
- AMQA800000
- AMQA900000

AMQAJRN günlüğünü bir kuyruk yöneticisi kitaplığına geri yüklerken, sistem AMQA000000günlük nesnesini otomatik olarak yaratır. Otomatik olarak oluşturulan bu günlük nesnesi, geri yükleyemeyeceğiniz varolan günlük nesnelere (AMQA000000) biriyle çakışıyor.

Çözüm:

1. Sonraki günlük nesnesini el ile yarat (bkz. [“IBM i üzerindeki kuyruk yöneticisi günlükleri” sayfa 410](#)):

```
CRTJRNRCV JRNRCV(QMQLIB/AMQA900001) THRESHOLD(XXXXX)
```

2. Günlüğü günlük nesnesiyle el ile yaratın:

```
CRTJRN JRN(QMQLIB/AMQAJRN) MNGRCV(*SYSTEM) +  
JRNRCV(QMQLIB/AMQA900001) MSGQ(QMQLIB/AMQAJRNMSG)
```

3. AMQA000000 yerel günlük nesnelerini AMQA900000olarak geri yükleyin.

IBM i

IBM i üzerinde çok eşgörünümlü kuyruk yöneticileri

Çok eşgörünümlü kuyruk yöneticileri, etkin sunucu başarısız olursa otomatik olarak yedek sunucuya geçerek kullanılabilirliği artırır. Etkin ve yedek sunucular aynı kuyruk yöneticisinin birden çok örneğidir; aynı kuyruk yöneticisi verilerini paylaşırlar. Etkin yönetim ortamı başarısız olursa, kuyruk yöneticisinin kuyruklarını yeniden oluşturması için günlüğü yedeğe aktarmanız gerekir.

Çok eşgörünümlü kuyruk yöneticilerini çalıştırdığınız IBM i sistemlerini, etkin kuyruk yöneticisi yönetim ortamı başarısız olursa, kullanmakta olduğu günlüğün devralan yedek yönetim ortamı tarafından kullanılabilmesi için yapılandırın. Günlüğü etkin eşgörünümünden devralan eşgörünümün kullanımına sunmak için kendi konfigürasyon ve denetim görevlerinizi tasarlayabilirsiniz. İletileri kaybetmek istemiyorsanız, tasarımınız yedek günlüğün hata noktasında etkin günlükle tutarlı olmasını sağlamalıdır. Tasarımınızı, tutarlılığı sağlayan sonraki konulardaki örneklerle açıklanan iki yapılandırmadan birinden uyarlayabilirsiniz.

1. Günlüğü, etkin kuyruk yöneticisi yönetim ortamını çalıştıran sistemden, yedek yönetim ortamlarını çalıştıran sistemlere ikizleyebilirsiniz.
2. Günlüğü, etkin eşgörünümü çalıştıran sistemden yedek eşgörünümü aktarılabilir bir Bağımsız Yardımcı Bellek Havuzu (IASP) içine yerleştirin.

İlk çözüm, temel ASP ' leri kullandığı için ek donanım veya yazılım gerektirmez. İkinci çözüm, ayrı olarak fiyatlandırılan IBM i Lisans Ürünü 5761-SS1 Seçenek 41 olarak kullanılabilen IBM i kümeleme desteğine gereksinim duyan değişimli kullanılabilir IASP ' leri gerektirir.

IBM i **IBM i üzerinde güvenilirlik ve kullanılabilirlik**

Çok eşgörünümlü kuyruk yöneticileri, uygulamaların kullanılabilirliğini artırmayı amaçlar. Teknolojik ve fiziksel kısıtlamalar, olağanüstü durum kurtarma, kuyruk yöneticilerini yedekleme ve sürekli çalışma taleplerini karşılamak için farklı çözümlere gereksinim duyacağınız anlamına gelir.

Güvenilirlik ve kullanılabilirlik için yapılandırma yapılırken çok sayıda faktörü devre dışı bırakarak dört farklı tasarım noktası elde edin:

Olağanüstü durum kurtarma

Tüm yerel varlıklarınızı yok eden büyük bir olağanüstü durumdan sonra kurtarma için optimize edilmiştir.

IBM i üzerinde olağanüstü durumdan kurtarma genellikle IASP ' nin coğrafi ikizlemesine dayalıdır.

Yedekleme

Yerleştirilmiş bir hatadan, genellikle bir insan hatasından ya da öngörülemeyen bir teknik sorundan sonra kurtarma için optimize edilmiştir.

IBM MQ , yedekleme kuyruğu yöneticilerinin düzenli aralıklarla kuyruk yöneticilerini yedeklemesini sağlar. Yedeklemenin para birimini geliştirmek için kuyruk yöneticisi günlüklerinin zamanuyumsuz eşlemesini de kullanabilirsiniz.

Kullanılabilirlik

Sunucu ya da disk arızası gibi öngörülebilir teknik hataların ardından neredeyse kesintisiz bir hizmetin görünümünü hızla veren işlemleri geri yüklemek için optimize edilmiştir.

Kurtarma genellikle dakika cinsinden ölçülür ve saptama bazen kurtarma işleminden daha uzun sürer. Çok eşgörünümlü bir kuyruk yöneticisi, *kullanılabilirlik* için yapılandırmada size yardımcı olur.

Sürekli çalışma

Kesintisiz bir hizmet sağlamak için eniyilenmiştir.

Sürekli operasyon çözümleri, algılama sorununu çözmek zorunda kalır ve neredeyse her zaman aynı işin birden fazla sistem üzerinden sunulmasını ve ilk sonucun kullanılmasını ya da doğruluğun en az iki sonucu karşılaştırarak önemli bir önem taşıdığını içerir.

Çok eşgörünümlü bir kuyruk yöneticisi, *kullanılabilirlik* için yapılandırmada size yardımcı olur. Kuyruk yöneticisinin bir örneği aynı anda etkin. Sistemin nasıl yapılandırıldığına, yüklendiğine ve ayarlandığına bağlı olarak, yedek bir örneğe geçiş on saniyeden on dakikadan on beş dakikaya ya da daha uzun sürer.

Çok eşgörünümlü bir kuyruk yöneticisi, uygulama programı bir kuyruk yöneticisi kesintisinden haberdar olmadan işlemeye devam edebilen IBM MQ MQI clientsile birlikte kullanıldığında neredeyse kesintisiz bir hizmetin görünümünü verebilir; Otomatik istemci yeniden bağlantısı konusuna bakın.

Kuyruk yöneticisi verileri, günlük eşleme ya da kuyruk yöneticisi günlükleri için güçlü IASP depolaması sağlayarak ve yeniden başlatılabilir kuyruk yöneticisi hizmetleri olarak yapılandırılan uygulamaların yeniden bağlanabilir istemcilerini kullanarak çok eşgörünümlü kuyruk yöneticilerini kullanarak yüksek düzeyde kullanılabilirlik çözümü oluşturun.

Çok eşgörünümlü bir kuyruk yöneticisi, başka bir sunucuda başka bir kuyruk yöneticisi yönetim ortamının başlatılmasını sürdürerek kuyruk yöneticisi hatasının saptanmasına tepki verir. Başlatma işlemini tamamlamak için, yönetim ortamının ağa bağlı saklama alanındaki paylaşılan kuyruk yöneticisi verilerine ve yerel kuyruk yöneticisi günlüğünün kopyasına erişmesi gerekir.

Yüksek kullanılabilirlikli bir çözüm yaratmak için, kuyruk yöneticisi verilerinin kullanılabilirliğini, yerel kuyruk yöneticisi günlüğünün para birimini yönetmeniz ve yeniden bağlanabilir istemci uygulamaları oluşturmanız ya da uygulamalarınızı kuyruk yöneticisi yeniden başlatıldığında otomatik olarak yeniden başlatmak üzere kuyruk yöneticisi hizmetleri olarak konuşturmanız gerekir. Otomatik istemci yeniden bağlantısı IBM MQ classes for Javatafından desteklenmez.

Kuyruk yöneticisi verileri

Kuyruk yöneticisi verilerini, büyük olasılıkla RAID düzey 1 diskleri ya da daha yüksek diskler kullanılarak paylaşılan, yüksek düzeyde kullanılabilir ve güvenilir olan ağa bağlı depolamaya yerleştirin. Dosya sisteminin çok eşgörünümlü kuyruk yöneticileri için paylaşılan bir dosya sistemi gereksinimlerini karşılaması gerekir; paylaşılan dosya sistemlerine ilişkin gereksinimler hakkında daha fazla bilgi için bkz. [Paylaşılan dosya sistemlerine ilişkin gereksinimler](#). Ağ Dosya Sistemi 4 (NFS4), bu gereksinimleri karşılayan bir iletişim kuralıdır.

Kuyruk yöneticisi günlükleri

Yedek yönetim ortamının kuyruk yöneticisi verilerini tutarlı duruma geri yükleyebilmesi için, kuyruk yöneticisi yönetim ortamları tarafından kullanılan IBM i günlüklerini de yapılandırmanız gerekir. Kesintisiz hizmet için, etkin yönetim ortamı başarısız olduğunda günlükleri geri yüklemeniz gerektiği anlamına gelir. Yedekleme veya olağanüstü durumdan kurtarma çözümlerinden farklı olarak, günlüklerin daha önceki bir denetim noktasına geri yüklenmesi yeterli değildir.

Ağ üzerinden depolamada birden çok IBM i sistemi arasında fiziksel olarak günlük paylaşamazsınız. Kuyruk yöneticisi günlüklerini hata noktasında tutarlı duruma geri yüklemek için, hata sırasında etkin kuyruk yöneticisi yönetim ortamına yerel olan fiziksel günlüğü etkinleştirilen yeni yönetim ortamına aktarmanız ya da çalışan yedek yönetim ortamlarında günlüğün ikizlemesini sürdürmeniz gerekir. İkizlenmiş günlük, başarısız olan eşgörünüme ait yerel günlükle tam olarak eşitlenmiş bir uzak günlük eşlemez.

Çok eşgörünümlü bir kuyruk yöneticisine ilişkin günlükleri nasıl yöneteceğinizi tasarlamak için üç yapılanış vardır:

1. Etkin eşgörünüm ASP 'sinden eşzamanlanmış günlük kopyalama (günlük ikizleme) yardımcı programının yedek eşgörünüm ASP 'lerine kullanılması.
2. Kuyruk yöneticisi günlüğünü tutmak üzere yapılandırıdığınız bir IASP ' nin etkin eşgörünümünden etkin eşgörünüm olarak devralınan yedek eşgörünüme aktarılması.
3. Eşitlenmiş ikincil IASP ikizlemeleri kullanılıyor.

IBM MQ IBM i CRTMQM komutunda, kuyruk yöneticisi verilerini IASP' ye yerleştirmeye ilişkin ek bilgi için [ASP seçeneklerine](#) bakın.

Ayrıca, IBM Documentationindeki IBM i bilgilerinde [High availability](#) (Yüksek kullanılabilirlik) başlıklı konuya bakın.

Uygulamalar

Yedek kuyruk yöneticisi devam ettiğinde kuyruk yöneticisine otomatik olarak yeniden bağlanmak üzere bir istemci oluşturmak için, uygulamanızı MQCONNX komutunu kullanarak kuyruk yöneticisine bağlayın

ve **MQCNO** Seçenekler alanında MQCNO_RECONNECT_Q_MGR değerini belirtin. Yeniden bağlanabilir istemcileri kullanan üç örnek program için [High availability sample programs](#) (Yüksek Kullanılabilirlik örnek programları) ve kurtarma için istemci uygulamalarının tasarlanmasına ilişkin bilgi için [Application recovery](#) (Uygulama kurtarma) başlıklı konuya bakın.

IBM i IBM i üzerinde NetServer kullanılarak kuyruk yöneticisi verileri için ağ paylaşımı yaratılması Kuyruk yöneticisi verilerini saklamak için IBM i sunucusunda bir ağ paylaşımı yaratın. Ağ paylaşımına erişmek için anasistem kuyruk yöneticisi yönetim ortamlarına gidecek iki sunucudan bağlantı ayarlayın.

Başlamadan önce

- Bu görev için üç IBM i sunucusu gerekir. Ağ paylaşımı, sunuculardan birinde, GAMMA ' da tanımlanır. Diğer iki sunucu, ALPHA ve BETA, GAMMA ' ya bağlanacak.
- IBM MQ ürününü üç sunucuya da kurun.
- System i Navigator' ı kurun; bkz. [System i Navigator](#).

Bu görev hakkında

- GAMMA ' da kuyruk yöneticisi dizinini oluşturun ve QMQM ve QMQMADM kullanıcı profilleri için doğru sahipliği ve izinleri ayarlayın. Dizin ve izin, IBM MQ GAMMA üzerine kurularak kolayca oluşturulur.
- GAMMA 'da kuyruk yöneticisi veri dizini için bir paylaşım oluşturmak üzere System i Navigator ' i kullanın.
- ALPHA ve BETA ' da paylaşım işaret eden dizinler oluşturun.

Yordam

1. GAMMA ' da, kuyruk yöneticisi verilerini sahip olarak QMQM kullanıcı profili ve birincil grup olarak QMQMADM ile barındıracak dizini oluşturun.

İpucu:

Dizini doğru izinlerle oluşturmanın hızlı ve güvenilir bir yolu, IBM MQ ürününü GAMMA üzerine kurmaktır.

Daha sonra, IBM MQ programını GAMMA üzerinde çalıştırmak istemiyorsanız, IBM MQ ürününü kaldırın. Kaldırma işleminden sonra /QIBM/UserData/mqm/qmgrs dizini, sahip QMQM kullanıcı profili ve QMQMADM birincil grubu ile birlikte GAMMA üzerinde kalır.

Görev, paylaşım için GAMMA üzerindeki /QIBM/UserData/mqm/qmgrs dizinini kullanır.

2. System i Navigator **Bağlantı ekle** sihirbazını başlatın ve GAMMA sistemine bağlanın.
 - a) Windows masaüstünüzdeki **System i Navigator** simgesini çift tıklayın.
 - b) Bağlantı yaratmak için **Evet** düğmesini tıklayın.
 - c) **Add Connection** (Bağlantı Ekle) sihirbazındaki yönergeleri izleyin ve IBM i sisteminden GAMMA ' ya bir bağlantı oluşturun.

GAMMA bağlantısı **Bağlantılarım**' a eklenir.
3. GAMMA ' da yeni bir dosya paylaşımı ekleyin.
 - a) **System i Navigator** penceresinde My Connections/GAMMA/File Systemsindeki File Shares klasörünü tıklayın.
 - b) **Görevlerim** penceresinde **IBM i NetServer paylaşımlarını yönet** seçeneğini tıklayın.

Masaüstünüzde **IBM i NetServer -GAMMA** adlı yeni bir pencere açılır ve paylaşılan nesnelere gösterir.
 - c) Shared Objects klasör > **Dosya** > **Yeni** > **Dosya** öğelerini sağ tıklayın.

IBM i NetServer Dosya Paylaşımı-GAMMA adlı yeni bir pencere açılır.
 - d) Paylaşım adını WMQ gibi bir ad verin.

e) Erişim denetimini Read/Writeolarak ayarlayın.

f) Daha önce oluşturduğunuz /QIBM/UserData/mqm/qmgrs dizinine göz atarak **Yol adı** 'nı seçin ve **Tamam** 'ı tıklatın.

IBM i NetServer Dosya Paylaşımı-GAMMA penceresi kapanır ve WMQ , paylaşılan nesnelere penceresinde listelenir.

4. Paylaşılan nesnelere penceresinde **WMQ** öğesini sağ tıklatın. **Dosya > İzinler**seçeneğini tıklatın.

/QIBM/UserData/mqm/qmgrsnesnesi için **Qmgrs Permissions-GAMMA**(Qmgrs İzinleri-GAMMA) penceresi açılır.

a) Önceden ayarlanmamışsa, QMQMiçin aşağıdaki izinleri denetleyin:

Read
Write
Execute
Management
Existence
Alter
Reference

b) Önceden ayarlanmamışsa, QMQMADMiçin aşağıdaki izinleri denetleyin:

Read
Write
Execute
Reference

c) /QIBM/UserData/mqm/qmgrs' e izin vermek istediğiniz diğer kullanıcı profillerini ekleyin.

Örneğin, /QIBM/UserData/mqm/qmgrs için varsayılan kullanıcı profiline (Genel) Read ve Execute izinlerine izin verebilirsiniz.

5. GAMMA üzerinde /QIBM/UserData/mqm/qmgrs erişimi verilen tüm kullanıcı profillerinin, GAMMA ' ya erişen sunucularla aynı parolaya sahip olup olmadığını denetleyin.

Özellikle, paylaşım erişecek diğer sunuculardaki QMQM kullanıcı profillerinin, GAMMA üzerindeki QMQM kullanıcı profiliyle aynı parolaya sahip olduğundan emin olun.

İpucu: Parolaları ayarlamak için System i Navigator ' nde My Connections/GAMMA/Users and Groups klasörünü tıklatın. Alternatif olarak, **CHFUSRPRF** ve **CHGPWD** komutlarını kullanın.

Sonuçlar

Paylaşımı kullanarak diğer sunuculardan GAMMA ' ya erişebildiğinizi kontrol edin. Diğer görevleri yapıyorsanız, /QNTC/GAMMA/WMQyolunu kullanarak ALPHA ve BETA 'dan GAMMA' ya erişebildiğinizi kontrol edin. /QNTC/GAMMA dizini ALPHA ya da BETA üzerinde yoksa, dizini oluşturmanız gerekir. NetServer etki alanına bağlı olarak, dizini yaratmadan önce IPL ALPHA ya da BETA işlemini gerçekleştirmeniz gerekebilir.

```
CRTDIR DIR('/QNTC/GAMMA')
```

ALPHA ya da BETA ' dan /QNTC/GAMMA/WMQ erişimine sahip olup olmadığını kontrol ettiğinizde CRTMQM MQMNAME('QM1') MQMDIRP('/QNTC/GAMMA/WMQ') , GAMMA üzerinde /QIBM/UserData/mqm/qmgrs/QM1 oluşturur.

Sonraki adım

[“IBM i üzerinde NetServer ve günlük ikizleme kullanılarak çok eşgörunümlü kuyruk yöneticisinin yaratılması” sayfa 434](#) ya da [“NetServer ve IBM i üzerinde günlük ikizleme kullanılarak tek bir eşgörunüm kuyruk yöneticisinin çok eşgörunümlü bir kuyruk yöneticisine dönüştürülmesi” sayfa 438](#) görevlerinden birindeki adımları izleyerek çok eşgörunümlü bir kuyruk yöneticisi yaratın.

Bir kuyruk yöneticisi eşgörünümünün saptanması için geçen süre başarısız oldu ve sonra bir yedek veritabanında işlemenin sürdürülmesi, yapılandırmaya bağlı olarak on beş dakika ya da daha fazla saniye arasında değişebilir. Yüksek düzeyde kullanılabilirlik çözümünün tasarlanmasında ve test edilmesinde performansın önemli bir dikkat edilmesi gerekir.

Çok eşgörünümlü bir kuyruk yöneticisinin konfigürasyonunu günlük eşlemesini kullanacak şekilde mi, yoksa IASP kullanacak şekilde mi tanımlanacağına karar vermenin yararları ve dezavantajları vardır. İkizleme, kuyruk yöneticisinin uzak bir günlüğe zamanuyumlu olarak yazmasını gerektirir. Donanım açısından bakıldığında, bunun başarımı etkilemesi gerekmez, ancak yazılım açısından bakıldığında uzak bir günlüğe yazılırken yalnızca yerel bir günlükten daha fazla yol uzunluğu söz konusu olabilir ve bunun, çalışan bir kuyruk yöneticisinin performansını bir dereceye kadar düşürmesi beklenir. Ancak, yedek kuyruk yöneticisi devraldığında, yerel günlüğün başarısız olmadan önce etkin eşgörünüm tarafından tutulan uzak günlükten eşzamanlanmasının gecikmesi, genellikle IBM i 'un IASP' yi algılaması ve kuyruk yöneticisinin yedek eşgörünümünü çalıştıran sunucuya aktarması için geçen süreye kıyasla küçüktür. IASP aktarım süreleri, saniyeler içinde tamamlanmak yerine on ila on beş dakika arasında olabilir. IASP aktarma süresi, IASP yedek sisteme ve birleştirilmesi gereken erişim yollarının ya da izinlerin büyüklüğüne aktarıldığında *kullanılabilir kılınması* gereken nesne sayısına bağlıdır.

Yedek kuyruk yöneticisi devraldığında, yerel günlüğün başarısız olmadan önce etkin eşgörünüm tarafından tutulan uzak günlükten eşzamanlanmasının gecikmesi, IBM i 'in bağımsız ASP' yi algılaması ve kuyruk yöneticisinin yedek eşgörünümünü çalıştıran sunucuya aktarması için geçen süreye kıyasla genellikle azdır. Bağımsız ASP aktarım süreleri, saniyeler içinde tamamlanmak yerine on ile on beş dakika arasında olabilir. Bağımsız ASP aktarım süresi, bağımsız ASP yedek sisteme aktarıldığında ve birleştirilmesi gereken erişim yollarının ya da izinlerin boyutuna *kullanılabilir* olması gereken nesne sayısına bağlıdır.

Ancak, yedek eşgörünümün tam olarak devam etmesi için gereken süreyi etkileyen tek etken günlüğün aktarılması değildir. Ayrıca, ağ dosya sisteminin, yedek yönetim ortamına başlatma işlemine devam etme sinyali veren kuyruk yöneticisi verilerindeki kilidi serbest bırakması için gereken süreyi ve ayrıca, yönetim ortamının iletileri yeniden işlemeye başlayabilmesi için günlükten kuyrukları kurtarma süresini de göz önünde bulundurmanız gerekir. Bu diğer gecikme kaynakları, yedek bir yönetim ortamını başlatmak için gereken süreyi ekler. Geçiş için toplam süre aşağıdaki bileşenlerden oluşur:

Arıza algılama süresi

NFS ' nin kuyruk yöneticisi verilerindeki kilidi serbest bırakması ve yedek yönetim ortamının başlatma işlemine devam etmesi için geçen süre.

Aktarım süresi

Bir HA kümesi söz konusu olduğunda, IASP ' nin etkin yönetim ortamını barındıran sistemden yedek yönetim ortamına aktarılması için IBM i geçen süre ve günlük eşleme durumunda, yedekteki yerel günlüğün uzak eşlemedeki verilerle güncellenmesi için geçen süre.

Yeniden başlatma zamanı

Yeni etkin kuyruk yöneticisi eşgörünümünün, geri yüklenen günlüğündeki en son denetim noktasından kuyruklarını yeniden oluşturması ve iletileri işlemeyi sürdürmesi için gereken süre.

Not:

Devralınan yedek yönetim ortamı, daha önce etkin olan yönetim ortamına zamanuyumlu olarak eşlenecek şekilde yapılandırıldıysa, başlatma gecikebilir. Yeni etkinleştirilen örnek, uzak günlük, daha önce etkin olan yönetim ortamını bulandıran sunucuysa ve sunucu başarısız olduysa, uzak günlüğe eşlenemeyebilir.

Zamanuyumlu bir yanıt için varsayılan bekleme süresi bir dakikadır. Eşleme zamanışımına uğramadan önce gecikme üst sınırını yapılandırabilirsiniz. Diğer bir seçenek olarak, yedek yönetim ortamlarını, başarısız olan etkin yönetim ortamına zamanuyumsuz kopyalamayı kullanmaya başlayacak şekilde yapılandırabilirsiniz. Daha sonra, başarısız olan yönetim ortamı yedek veritabanında yeniden çalışırken zamanuyumlu eşlemeye geçersiniz. Aynı değer, zamanuyumlu bağımsız ASP ikizlerinin kullanılması için de geçerlidir.

Bu bileşenler için ayrı temel çizgi ölçümleri yaparak, hata durumunda yedek sisteme geçiş için genel süreyi değerlendirmenize ve hangi yapılandırma yaklaşımının kullanılacağını kararınıza katmanıza

yardımcı olabilir. En iyi konfigürasyon kararını verirken, aynı sunucudaki diğer uygulamaların nasıl yedek sisteme geçeceğini ve IASP 'yi zaten kullanan yedekleme ya da olağanüstü durumdan kurtarma işlemlerinin olup olmadığını da göz önünde bulundurmanız gerekir.

IASP aktarım süreleri, küme yapılandırmanız ayarlanarak kısaltılabilir:

1. Kullanılır kılma işleminin UUID ve GID 'leri değiştirmesi gereksinimini ortadan kaldırmak için kümedeki sistemler arasındaki kullanıcı tanımları aynı GID ve UID' ye sahip olmalıdır.
2. Disk havuzu grubuna ilişkin çapraz başvuru çizelgesini yaratmak için bunların birleştirilmesi gerektiğinden, sistemdeki veri tabanı nesnelere ve temel kullanıcı disk havuzlarının sayısını en aza indirin.
3. Diğer performans ipuçlarını IBM Redbook, *Implementing PowerHA for IBM i, SG24-7405* içinde bulabilirsiniz.

Temel ASP 'leri, günlük ikizlemeyi ve küçük bir yapılandırmayı kullanan bir yapılandırma, on saniye içinde geçiş yapmalıdır.

IBM i **IBM i kümeleme yeteneklerinin IBM MQ kümeleme ile birleştirilmesine genel bakış**

IBM üzerinde IBM MQ 'in çalıştırılması ve IBM i kümeleme yeteneklerinden yararlanılması, yalnızca IBM MQ kümelemesinden daha kapsamlı bir Yüksek Düzeyde Kullanılabilirlik çözümü sağlayabilir.

Bu yeteneğe sahip olmak için aşağıdakileri ayarlamamız gerekir:

1. IBM i makinenizdeki kümeler; bkz. [“IBM i Kümeler” sayfa 425](#)
2. Kuyruk yöneticisini taşıdığınız bağımsız yardımcı bellek havuzu (IASP); bkz. [“Bağımsız yardımcı bellek havuzları \(IASP\)” sayfa 425](#)
3. Bir küme kaynağı grubu (CRG); aşağıdakileri tanımladığınız [“Aygıt kümesi kaynak grupları” sayfa 426](#) başlıklı konuya bakın:
 - Kurtarma etki alanı
 - IASP
 - Çıkış programı; bkz. [“Aygıt CRG çıkış programı” sayfa 426](#)

IBM i Kümeler

IBM i kümesi, mantıksal olarak birbirine bağlanan IBM i bilgisayarları ya da bölümleri olan yönetim ortamlarının bir toplamasıdır.

Bu gruplamanın amacı, tek bir hata noktasını ortadan kaldırarak ve uygulama ve veri esnekliğini artırarak her eşgörünümün yedeklenmesine izin vermektir. Bir küme yaratıldığında, çeşitli küme kaynağı grubu (CRG) tipleri, kümedeki uygulamaları, verileri ve aygıtları yönetecek şekilde yapılandırılabilir.

Ek bilgi için [Küme yaratılması](#) ve [Küme yaratılması \(CRTCLU\)](#) komutuna bakın.

Bağımsız yardımcı bellek havuzları (IASP)

IASP, tek düzeyli saklama alanının uzantısı olarak hizmet veren bir kullanıcı ASP tipidir. Sistem depolamasından bağımsızlığı nedeniyle, sistemi IPL etmek zorunda kalmadan kolayca manipüle edilebilen bir depolama alanıdır.

Bir IASP, başka bir işletim sistemi örneğine kolayca geçilebilir ya da başka bir işletim sistemi örneğinde hedef IASP 'ye eşlenebilir. Bir IASP 'yi eşgörünüm arasında değiştirmek için iki yöntem kullanılabilir:

- İlk yöntem, kümedeki tüm bilgisayarların ve IASP 'yi içeren değişimli kullanılabilir disk biriminin bir HSL (High Speed Link; Yüksek Hızlı Bağlantı) döngüsü kullanılarak bağlanmasını gerektirir.
- İkinci yöntem, işletim sistemi yönetim ortamlarının, giriş/çıkış işlemcilerinin (IOP 'ler) bölümler arasında değiştirilebileceği aynı IBM i bilgisayarda yer almalarını gerektirir. Bir IASP 'yi eşleyebilmek için özel donanım gerekmez. Eşleme, ağ üzerinden TCP/IP kullanılarak gerçekleştirilir.

Ek bilgi için [Configure Device ASP \(CFGDEVASP\)](#) (Aygıt ASP 'si konfigürasyonunun tanımlanması) komutuna bakın.

Aygıt kümesi kaynak grupları

Birkaç tip küme kaynağı grubu (CRG) vardır. Kullanılabilir farklı CRG tipleri hakkında daha fazla bilgi için bkz. [Küme kaynağı grubu](#).

Bu konu, bir aygıt CRG 'si üzerinde yoğunlaşır. Bir aygıt CRG 'si:

- Bağımsız yardımcı depolama havuzları (IASP) gibi aygıt kaynaklarını açıklar ve yönetir.
- Küme düğümlerinin kurtarma etki alanını tanımlar
- Bir aygıt atar ve
- Küme olaylarını işleyecek çıkış programını atar.

Kurtarma etki alanı, hangi küme düğümünün birincil düğüm olarak kabul edileceğini belirtir. Düğümlerin geri kalanı yedek olarak kabul edilir. Yedek düğümler, kurtarma etki alanında kaç düğüm olduğuna bağlı olarak, hangi düğümün ilk yedek, ikinci yedek olduğunu belirterek kurtarma etki alanında da sıralanır.

Birincil düğüm arızası durumunda, çıkış programı kurtarma etki alanındaki tüm düğümlerde çalıştırılır. Daha sonra, ilk yedeklemede çalışan çıkış programı, bu düğümü yeni birincil düğüm yapmak için gerekli kullanıma hazırlamaları yapabilir.

Daha fazla bilgi için [Creating device CRG](#) (Aygıt CRG 'leri oluşturma) ve [Create Cluster Resource Group \(CRTCRG\)](#) (Küme Kaynağı Grubu Oluştur) komutuna bakın.

Aygıt CRG çıkış programı

İşletim sistemi küme kaynağı hizmeti, kurtarma etki alanının tanımladığı düğümlerden birinde bir olay ortaya çıktığında bir aygıt CRG çıkış programını çağırır; örneğin, bir hata durumunda yedek sisteme geçiş ya da yedek sisteme geçiş olayı.

Bir yedek sisteme geçiş olayı, kümenin birincil düğümü başarısız olduğunda ve CRG 'ler yönettikleri tüm kaynaklarla değiştirildiğinde ve belirli bir CRG, birincil düğümden yedek düğüme el ile geçildiğinde ortaya çıkar.

Her iki yöntemle de, çıkış programı, ilk yedek düğümü yeni birincil düğüme dönüştüren önceki birincil düğümden çalışan tüm programları başlatma ve başlatma sorumluluğunu üstlenir.

Örneğin, IBM MQ ile, çıkış programı IBM MQ altsistemini (QMOM) ve kuyruk yöneticilerini başlatmakla sorumlu olmalıdır. Kuyruk yöneticileri, dinleyicileri ve hizmetleri (tetikleyici izleme programları gibi) otomatik olarak başlatabilecekler şekilde yapılandırılmalıdır.

Örnek bir çıkış programı olan AMQSCRG4, IBM i üzerinde IBM MQ 9.1 adresinden bulunabilir.

Değişimli kullanılabilir IASP konfigürasyonu

IBM MQ , IBM ikümeleme yeteneklerinden yararlanacak şekilde ayarlanabilir. Bunu yapmak için:

1. Veri merkezi sistemleri arasında bir IBM i kümesi oluşturma
2. Kuyruk yöneticisini bir IASP 'ye taşıyın.

[“Bir kuyruk yöneticisinin bağımsız yardımcı bellek havuzuna ya da havuzdan taşınması ya da kaldırılması” sayfa 427](#) , bu işlemi gerçekleştirmenize yardımcı olacak bazı örnek kodları içerir.

3. Kurtarma etki alanını, IASP 'yi ve çıkış programını tanımlayan bir CRG yaratmanız gerekir.

[“Aygıt kümesi kaynak grubunun yapılandırılması” sayfa 427](#) , bu işlemi gerçekleştirmenize yardımcı olacak bazı örnek kodları içerir.

İlgili kavramlar

[“Bağımsız ASP 'ler ve yüksek düzeyde kullanılabilirlik” sayfa 447](#)

Bağımsız ASP 'ler, uygulamaların ve verilerin sunucular arasında taşınmasını sağlar. Bağımsız ASP 'lerin esnekliği, bunların bazı IBM i yüksek düzeyde kullanılabilirlik çözümlerinin temeli olduğu anlamına gelir. Kuyruk yöneticisi günlüğü için bir ASP mi, yoksa bağımsız ASP mi kullanacağınızı göz önünde bulundurarak, bağımsız ASP 'lere dayalı olarak diğer yüksek kullanılabilirlikli konfigürasyonları göz önünde bulundurmanız gerekir.

IBM i *Aygıt kümesi kaynak grubunun yapılandırılması*

Bir aygıt kümesi kaynak grubu (CRG) kurmak için örnek bir program.

Bu görev hakkında

Aşağıdaki örnekte şunu göz önünde bulundurun:

- [PRIMARY SITE NAME] ve [BACKUP SITE NAME] en çok sekiz karakterden oluşan iki ayrı dizgi olabilir.
- [PRIMARY IP] ve [BACKUP IP], ikizleme için kullanılacak IP 'lerdir.

Yordam

1. Kümenin adını tanımlayın.
2. CRG çıkış programı adını ve kitaplığını tanımlayın.
3. Bu CRG tarafından tanımlanacak birincil düğümün ve yedek düğümlerin adını belirleyin.
4. Bu CRG tarafından yönetilecek IASP 'yi tanımlayın ve birincil düğüm altında yaratıldığından emin olun.
5. Şu komutu kullanarak yedek düğümlerde bir aygıt tanımlaması oluşturun:

```
CRTDEVASP DEVD([IASP NAME]) RSRNAME([IASP NAME])
```

6. Devralınan IP adresini şu komutu kullanarak tüm düğümlere ekleyin:

```
ADDTCPIFC INTNETADR(' [TAKEOVER IP]') LIND([LINE DESC])  
SUBNETMASK(' [SUBNET MASK]') AUTOSTART(*NO)
```

7. Devralınan IP adresini yalnızca birincil düğümde şu komutu kullanarak başlatın:

```
STRTCPIFC INTNETADR(' [TAKEOVER IP]')
```

8. İsteğe bağlı: IASP 'iniz değişimli kullanılabilir ise, şu komutu çağırın:

```
CRTCRG CLUSTER([CLUSTER NAME]) CRG([CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT  
NAME])  
USRPRF([EXIT PROFILE]) RCYDMN(([PRIMARY NODE] *PRIMARY) ([BACKUP NAME] *BACKUP))  
EXITPGMFMT(EXTP0200) CFGOBJ([IAPS NAME] *DEVD *ONLINE '[TAKEOVER IP]')
```

9. İsteğe bağlı: IASP 'iniz ikizlenecek ise, şu komutu çağırın:

```
CRTCRG CLUSTER([CLUSTER NAME]) CRG([CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT NAME])  
USRPRF([EXIT PROFILE]) RCYDMN(([PRIMARY NODE] *PRIMARY *LAST [PRIMARY SITE NAME] (' [PRIMARY  
IP] '))  
[BACKUP NAME] *BACKUP *LAST [BACKUP SITE NAME] (' [BACKUP IP] '))) EXITPGMFMT(EXTP0200)  
CFGOBJ([IAPS NAME] *DEVD *ONLINE '[TAKEOVER IP]')
```

IBM i *Bir kuyruk yöneticisinin bağımsız yardımcı bellek havuzuna ya da havuzdan taşınması ya da kaldırılması*

Bir kuyruk yöneticisini bağımsız yardımcı bellek havuzuna (IASP) taşımak için örnek bir program ve bir kuyruk yöneticisini IASP 'den kaldırmak için gereken komutlar.

Bu görev hakkında

Aşağıdaki örnekte şunu göz önünde bulundurun:

- [MANAGER NAME], kuyruk yöneticinizin adıdır.

- [IASP NAME], IASP ' inizin adıdır.
- [MANAGER LIBRARY], kuyruk yöneticisi kitaplığınızın adıdır.
- [MANAGER DIRECTORY], kuyruk yöneticisi dizininizin adıdır.

Yordam

1. Birincil düğümünüzü ve yedek düğümlerinizi tanımlayın.
2. Birincil düğümünüz üzerinde aşağıdaki yordamı gerçekleştirin:

- a) Kuyruk yöneticinizin sona erdiğinden emin olun.
- b) Komutu kullanarak IASP ' nin vary on olduğundan emin olun

```
VRYCFG CFGOBJ([IASP NAME]) CFGTYPE(*DEV) STATUS(*ON)
```

- c) IASP altında kuyruk yöneticileri dizinini yaratın.
Kök altında IASP ' nin adını taşıyan bir dizin vardır:

```
QSH CMD('mkdir -p /[IASP_NAME]/QIBM/UserData/mqm/qmgrs/')
```

- d) Aşağıdaki komutu kullanarak, yöneticinizin IFS nesnelerini IASP altında yarattığınız kuyruk yöneticileri dizinine taşıyın:

```
QSH CMD('mv /QIBM/UserData/mqm/qmgrs/[MANAGER NAME]  
/[IASP NAME]/QIBM/UserData/mqm/qmgrs/')
```

- e) Şu komutu kullanarak MGRLIB adlı geçici bir saklama dosyası yaratın:

```
CRTSAVF QGPL/MGRLIB
```

- f) Aşağıdaki komutu kullanarak kuyruk yöneticisi kitaplığınızı MGRLIB saklama dosyasına saklayın:

```
SAVLIB LIB([MANGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)
```

- g) Aşağıdaki komutu kullanarak kuyruk yöneticisi kitaplığını silin ve tüm sorgu iletilerini yoksayın:

```
DLTLIB [MANAGER LIBRARY]
```

- h) Aşağıdaki komutu kullanarak kuyruk yöneticisi kitaplığınızı IASP ' ye geri yükleyin:

```
RSTLIB SAVLIB([MANAGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)  
RSTASPDEV([IASP NAME])
```

- i) Aşağıdaki komutu kullanarak geçici saklama dosyasını silin:

```
DLTF FILE(QGPL/MGRLIB)
```

- j) Aşağıdaki komutu kullanarak, IASP altında kuyruk yöneticisi IFS nesnelere sembolik bir bağlantı yaratın:

```
ADDLNK OBJ('/[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')  
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

- k) Aşağıdaki komutu kullanarak IASP ' ye bağlanın:

```
SETASPGRP [IASP NAME]
```

- l) Şu komutu kullanarak kuyruk yöneticinizi başlatın:

```
STRMQM [MANAGER NAME]
```

3. Yedek düğümünüz ya da düğümlerinizi için aşağıdaki yordamı gerçekleştirin:

a) Aşağıdaki komutu kullanarak geçici bir kuyruk yöneticisi dizini yaratın:

```
QSH CMD('mkdir -p /[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

b) Aşağıdaki komutu kullanarak kuyruk yöneticisi geçici dizinine sembolik bir bağlantı yaratın:

```
ADDLNK OBJ('/[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')  
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

c) Aşağıdaki komutu kullanarak geçici dizini silin:

```
QSH CMD('rm -r /[IASP NAME]')
```

d) /QIBM/UserData/mqm/mqs.inidosisyasının sonuna aşağıdakileri ekleyin:

```
QueueManager:  
Name=[MANAGER NAME]  
Prefix=/QIBM/UserData/mqm  
Library=[MANAGER LIBRARY]  
Directory=[MANAGER DIRECTORY]
```

4. Bir kuyruk yöneticisini IASP ' den kaldırmak için aşağıdaki komutları verin:

- VRYCFG CFGOBJ ([IASP ADI]) CFGTYPE (*DEV) STATUS (*ON)
- SETASPGRP [IASP ADI]
- ENDMQM [YÖNETİCİ ADI]
- DLTMQM [YÖNETİCİ ADI]

IBM i **IBM i üzerinde ASP için ikizlenmiş günlük yapılandırması**

İkizlenmiş günlükler arasında zamanuyumlu eşleme kullanarak güçlü bir çok eşgörünümlü kuyruk yöneticisi yapılandırın.

İkizlenmiş kuyruk yöneticisi konfigürasyonu, temel ya da bağımsız yardımcı bellek havuzlarında (ASP) yaratılan günlükleri kullanır.

IBM sistemlerinde, kuyruk yöneticisi verileri günlüklere ve bir dosya sistemine yazılır. Günlükler, kuyruk yöneticisi verilerinin ana kopyasını içerir. Günlükler, zamanuyumlu ya da zamanuyumsuz günlük kopyalama kullanılarak sistemler arasında paylaşılır. Bir kuyruk yöneticisi yönetim ortamını yeniden başlatmak için yerel ve uzak günlüklerin birlikte kullanılması gerekir. Kuyruk yöneticisi yeniden başlatması, sunucudaki yerel ve uzak günlüklerin karışık kullanımından ve paylaşılan ağ dosya sistemindeki kuyruk yöneticisi verilerinden günlük kayıtlarını okur. Dosya sistemindeki veriler, kuyruk yöneticisinin yeniden başlatılmasını hızlandırır. Denetim noktaları, dosya sistemi ile günlükler arasındaki eşzamanlama noktalarını işaretleyerek dosya sisteminde saklanır. Denetim noktasından önce saklanan günlük kayıtları, tipik kuyruk yöneticisi yeniden başlatması için gerekli değildir. Ancak, dosya sistemindeki veriler güncel olmayabilir ve kuyruk yöneticisinin yeniden başlatılmasını tamamlamak için denetim noktasından sonraki günlük kayıtları kullanılır. Yönetim ortamına bağlı günlüklerdeki veriler, yeniden başlatmanın başarıyla tamamlanabilmesi için güncel tutulur.

Ancak, yedek sunucudaki uzak günlük zamanuyumsuz olarak kopyalanıyorsa ve hata eşzamanlanmadan önce ortaya çıktıysa, günlük kayıtları bile güncel olmayabilir. Uyumlulaştırılmamış bir uzak günlüğü kullanarak bir kuyruk yöneticisini yeniden başlatmaya karar vererseniz, yedek kuyruk yöneticisi etkin eşgörünüm başarısız olmadan önce silinen iletileri yeniden işleyebilir ya da etkin eşgörünüm başarısız olmadan önce alınan iletileri işlemeyebilir.

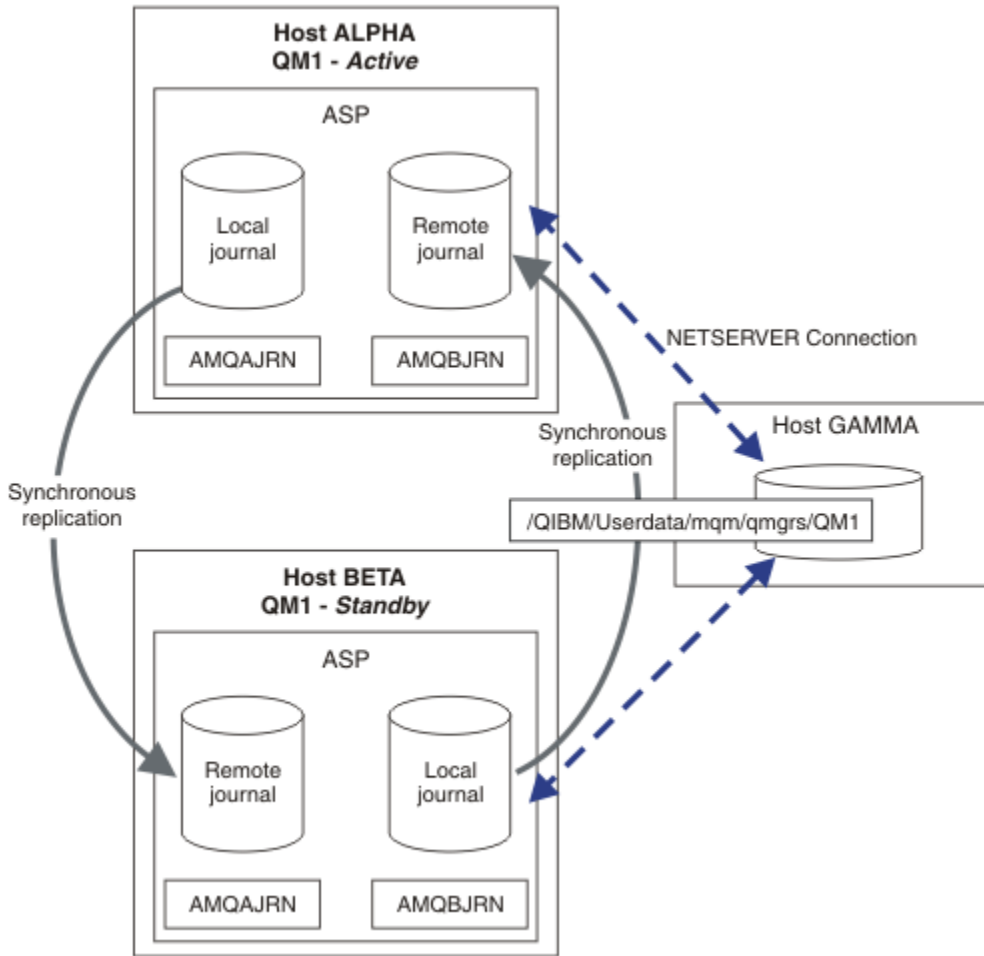
Başka bir olasılık da, dosya sisteminin en son denetim noktası kaydını içermesi ve yedekteki eşzamanlanmamış bir uzak günlüğün içermemesi olabilir. Bu durumda, kuyruk yöneticisi otomatik olarak yeniden başlatılmaz. Uzak günlüğün eşzamanlanmasını ya da yedek kuyruk yöneticisini dosya sisteminden başlatmayı bekleme seçeneğiniz vardır. Bu durumda, dosya sistemi kuyruk yöneticisi verilerinin uzak günlükten daha yeni bir denetim noktasını içerse de, etkin eşgörünüm başarısız olmadan önce işlenen tüm iletileri içermeyebilir. Günlüklerle zamanuyumlu olmayan bir soğuk yeniden başlatma işleminden sonra bazı iletiler yeniden işlenebilir ve bazıları işlenmeyebilir.

Çok eşgörünümlü bir kuyruk yöneticisiyle, hangi kuyruk yöneticisinin etkin ve hangisinin yedek veritabanı olduğunu denetlemek için dosya sistemi de kullanılır. Etkin yönetim ortamı, kuyruk yöneticisi verileri için bir kilit elde eder. Yedek veritabanı kilidi almak için bekler ve kilitlendiğinde etkin yönetim ortamı olur. Kilit, olağan şekilde sona ererse, etkin eşgörünüm tarafından serbest bırakılır. Dosya sistemi etkin yönetim ortamının başarısız olduğunu saptarsa ya da dosya sistemine erişemezse, kilit dosya sistemi tarafından serbest bırakılır. Dosya sistemi, arızanın saptanmasına ilişkin gereksinimleri karşılamalıdır; bkz. [Paylaşılan dosya sistemlerine ilişkin gereksinimler](#).

IBM i üzerindeki çok eşgörünümlü kuyruk yöneticilerinin mimarisi, sunucu ya da kuyruk yöneticisi hatasının ardından otomatik yeniden başlatma sağlar. Kuyruk yöneticisi verilerinin saklandığı dosya sistemi hatasının ardından kuyruk yöneticisi verilerinin geri yüklenmesini de destekler.

Şekil 24 sayfa 430'de ALPHA başarısız olursa, ikizlenmiş günlüğü kullanarak BETA' da QM1 ' yi el ile yeniden başlatabilirsiniz. Çok eşgörünümlü kuyruk yöneticisi yeteneği QM1'e eklendiğinde, ALPHA' daki etkin yönetim ortamı başarısız olursa, QM1 yedek yönetim ortamı BETA ' da otomatik olarak devam eder. QM1 , yalnızca QM1' un etkin örneği değil, başarısız olan sunucu ALPHA ise de otomatik olarak devam edebilir. BETA, etkin kuyruk yöneticisi yönetim ortamının anasistemi olduğunda, yedek yönetim ortamı ALPHA üzerinde başlatılabilir.

Şekil 24 sayfa 430 içinde, kuyruk yöneticisi verilerini saklamak için NetServer olanağını kullanarak bir kuyruk yöneticisinin iki eşgörünümü arasındaki günlükleri ikizleyen bir yapılandırma gösterilir. Örüntüyü, daha fazla günlük ve dolayısıyla daha fazla eşgörünüm içerecek şekilde genişletebilirsiniz. “IBM i üzerindeki kuyruk yöneticisi günlükleri” sayfa 410başlıklı konuda açıklanan günlük adlandırma kurallarını izleyin. Şu anda bir kuyruk yöneticisinin çalışan eşgörünümlerinin sayısı ikiyle sınırlıdır, biri etkin ve biri yedekte.



Şekil 24. Kuyruk yöneticisi günlüğünü ikizleme

ALPHA anasistemindeki QM1 için yerel günlük AMQAJRN (ya da daha çok tam olarak QMQM1/AMQAJRN) olarak adlandırılır ve BETA üzerinde günlük QMQM1/AMQBJRN olarak adlandırılır. Her yerel günlük, kuyruk yöneticisinin diğer tüm örneklerinde uzak günlüklere kopyalar. Kuyruk yöneticisinin konfigürasyonu iki eşgörünümle tanımlanmışsa, yerel günlük bir uzak günlüğe kopyalanır.

*SYNC ya da *ASYNC uzak günlük eşlemesi

IBM i günlükleri zamanuyumlu (*SYNC) kullanılarak ikizlenir ya da zamanuyumsuz (*ASYNC) günlük kaydı; bkz. [Uzak günlük yönetimi](#).

Şekil 24 sayfa 430 içindeki eşleme kipi, *ASYNC değil *SYNC. *ASYNC daha hızlıdır, ancak uzak günlük durumu *ASYNC PENDING olduğunda bir hata oluşursa, yerel ve uzak günlük tutarlı değildir. Uzak günlüğün yerel günlüğe yetişmesi gerekir. *SYNC seçeneğini belirlerseniz, yerel sistem, yazma işleminin tamamlanmasını gerektiren bir çağrıdan dönmeden önce uzak günlüğü bekler. Yerel ve uzak günlükler genellikle birbirleriyle tutarlı olmaya devam eder. Yalnızca *SYNC işlemi belirlenmiş bir süreden daha uzun sürerse¹, ve uzak günlük kaydı devre dışı bırakılır, günlüklerin eşzamanlaması sona erer. Günlük ileti kuyruğuna ve QSYSOPR ' ye bir hata kaydedilir. Kuyruk yöneticisi bu iletiyi algılar, kuyruk yöneticisi hata günlüğüne bir hata yazar ve kuyruk yöneticisi günlüğünün uzak eşlemesini devre dışı bırakır. Etkin kuyruk yöneticisi eşgörünümü, bu günlüğe uzaktan günlük kaydı yapılmadan devam eder. Uzak sunucu yeniden kullanılabilir olduğunda, zamanuyumlu uzak günlük eşlemesini el ile yeniden etkinleştirmeniz gerekir. Daha sonra günlükler yeniden eşitlenir.

Şekil 24 sayfa 430 içinde gösterilen *SYNC / *ASYNC konfigürasyonu ile ilgili bir sorun, BETA üzerinde yedek kuyruk yöneticisi eşgörünümünün denetimi nasıl ele aldığıdır. BETA üzerindeki kuyruk yöneticisi eşgörünümü ilk kalıcı iletisini yazar yazmaz, ALPHA üzerindeki uzak günlüğü güncellemeye çalışır. Eğer kontrolün ALPHA 'dan BETA' ya geçmesinin nedeni ALPHA 'ya olan başarısızlık ve ALPHA hala çalışmıyorsa, ALPHA 'ya uzaktan günlük kaydı başarısız olur. BETA, ALPHA 'ın yanıt vermesini bekler ve daha sonra uzak günlük kaydını devre dışı bırakır ve iletileri yalnızca yerel günlük kaydı ile işlemeye devam eder. BETA, ALPHA 'nin kapalı olduğunu tespit etmek için bir süre beklemek zorunda. Bu da bir süre hareketsizliğe neden olur.

Uzak günlük kaydını *SYNC ya da *ASYNC olarak ayarlama arasındaki seçim bir takas seçeneğidir. Çizelge 24 sayfa 431 , bir kuyruk yöneticisi çifti arasında *SYNC ve *ASYNC günlük kaydı kullanımı arasındaki değiş tokuşları özetler:

Çizelge 24. Uzak günlük kaydı seçenekleri			
Etkin	Yedek	*SYNC	*ASYNC
*SYNC		<ol style="list-style-type: none"> 1. Tutarlı yedek sisteme geçiş ve yedek sisteme geçiş 2. Yedek yönetim ortamı, yedeği devreye sokmadan hemen sonra devam etmez. 3. Uzak günlük kaydı her zaman kullanılabilir olmalıdır 4. Kuyruk yöneticisi başarımları, uzak günlük kaydına bağlıdır 	<ol style="list-style-type: none"> 1. Tutarlı yedek sisteme geçiş ve yedek sisteme geçiş 2. Yedekleme sunucusu kullanılabilir olduğunda uzak günlük kaydı *SYNC olarak değiştirilmelidir 3. Uzak günlük kaydı yeniden başlatıldıktan sonra kullanılabilir durumda kalmalıdır 4. Kuyruk yöneticisi başarımları, uzak günlük kaydına bağlıdır

¹ Belirtilen süre, IBM i 5 üzerinde 60 saniye ve IBM i 6.1 üzerinde 1-3600 saniye aralığındadır.

Çizelge 24. Uzak günlük kaydı seçenekleri (devamı var)

Etkin	Yedek	*SYNC	*ASYNC
		1. Mantıklı bir kombinasyon değil.	1. Hata durumunda yedek sisteme geçiş ya da geçiş işleminden sonra bazı iletiler kaybolabilir ya da çoğaltılabilir 2. Etkin yönetim ortamının gecikmeden devam edebilmesi için yedek yönetim ortamının her zaman kullanılabilir olması gerekmez. 3. Başarım, uzak günlük kaydına bağlı değildir

*SYNC / *SYNC

Etkin kuyruk yöneticisi eşgörünümü *SYNC günlük kaydını kullanır ve yedek kuyruk yöneticisi eşgörünümü başladığında hemen *SYNC günlük kaydını kullanmayı dener.

1. Uzak günlük, etkin kuyruk yöneticisinin yerel günlüğüyle işlemsel olarak tutarlıdır. Kuyruk yöneticisi yedek yönetim ortamına geçirilirse, hemen devam edebilir. Yedek yönetim ortamı olağan durumda iletileri kaybetmeden ya da yinelemeden devam eder. İletiler yalnızca son denetim noktasından bu yana uzak günlük kaydı başarısız olduğunda kaybolur ya da yinelenir ve daha önce etkin olan kuyruk yöneticisi yeniden başlatılamaz.
2. Kuyruk yöneticisi yedek yönetim ortamına geçemezse, hemen başlatılamayabilir. Yedek kuyruk yöneticisi yönetim ortamı *SYNC günlük kaydı ile etkinleştirilir. Yedek sisteme geçiş işleminin nedeni, yedek yönetim ortamını barındıran sunucuda uzaktan günlüğe kaydetmeyi engelleyebilir. Kuyruk yöneticisi, kalıcı iletiler işlenmeden önce sorunun saptanmasını bekler. Günlük ileti kuyruğuna ve QSYSOPR ' ye bir hata kaydedilir. Kuyruk yöneticisi bu iletiyi algılar, kuyruk yöneticisi hata günlüğüne bir hata yazar ve kuyruk yöneticisi günlüğünün uzak eşlemesini devre dışı bırakır. Etkin kuyruk yöneticisi eşgörünümü, bu günlüğe uzaktan günlük kaydı yapılmadan devam eder. Uzak sunucu yeniden kullanılabilir olduğunda, zamanuyumlu uzak günlük eşlemesini el ile yeniden etkinleştirmeniz gerekir. Daha sonra günlükler yeniden eşitlenir.
3. Uzak günlüğün kopyalandığı sunucu, uzak günlüğü korumak için her zaman kullanılabilir olmalıdır. Uzak günlük genellikle yedek kuyruk yöneticisini barındıran sunucuya kopyalanır. Sunucu kullanılamaz duruma gelebilir. Günlük ileti kuyruğuna ve QSYSOPR ' ye bir hata kaydedilir. Kuyruk yöneticisi bu iletiyi algılar, kuyruk yöneticisi hata günlüğüne bir hata yazar ve kuyruk yöneticisi günlüğünün uzak eşlemesini devre dışı bırakır. Etkin kuyruk yöneticisi eşgörünümü, bu günlüğe uzaktan günlük kaydı yapılmadan devam eder. Uzak sunucu yeniden kullanılabilir olduğunda, zamanuyumlu uzak günlük eşlemesini el ile yeniden etkinleştirmeniz gerekir. Daha sonra günlükler yeniden eşitlenir.
4. Uzak günlük kaydı, yerel günlük kaydından daha yavaştır ve sunucular geniş bir mesafeye ayrılmışsa önemli ölçüde daha yavaştır. Kuyruk yöneticisi, kuyruk yöneticisi başarımını azaltan uzak günlük kaydını beklemelidir.

Bir sunucu çifti arasındaki *SYNC / *SYNC konfigürasyonu, yedek sisteme geçiş işleminden sonra yedek eşgörünümün sürdürülmesinde bir gecikme avantajına sahiptir. *SYNC / *ASYNC konfigürasyonunda bu sorun yoktur.

*SYNC / *SYNC , uzak bir günlük kullanılabilir olduğu sürece, yedek sisteme geçiş ya da yedek sisteme geçiş işleminden sonra ileti kaybını garanti etmez. Hata durumunda yedek sisteme geçiş ya da yedek sisteme geçiş sonrasında ileti kaybı riskini azaltmak istiyorsanız, iki seçeneğiniz vardır. Uzak günlük etkin değilse, etkin yönetim ortamını durdurun ya da birden çok sunucuda uzak günlükler yaratın.

*SYNC / *ASYNC

Etkin kuyruk yöneticisi yönetim ortamı *SYNC günlük kaydını kullanır ve yedek kuyruk yöneticisi eşgörünümü başlatıldığında *ASYNC günlük kaydını kullanır. Yeni yedek yönetim ortamını barındıran

sunucu kullanılabilir olduktan kısa bir süre sonra, sistem işletmeninin etkin yönetim ortamındaki uzak günlüğü *SYNColarak değiştirmesi gerekir. İşletmen, uzak günlüğü *ASYNCC'den *SYNCC 'ye çevirdiğinde, uzak günlüğün durumu *ASYNCPENDise etkin eşgörünüm duraklatılır. Etkin kuyruk yöneticisi eşgörünümü, uzak günlüğe kalan günlük girişleri aktarıncaya kadar bekler. Uzak günlük yerel günlükle uyumlulaştırıldığında, yeni yedek veritabanı yeni etkin yönetim ortamıyla işlemsel olarak tutarlıdır. Çok eşgörünümlü kuyruk yöneticilerinin yönetimi açısından, *SYNCC / *ASYNCC yapılandırmasında IBM i sistem işletmeninin ek bir görevi vardır. İşleç, başarısız olan kuyruk yöneticisi eşgörünümünü yeniden başlatmanın yanı sıra uzak günlük kaydını *SYNCC olarak da değiştirmelidir.

1. Uzak günlük, etkin kuyruk yöneticisinin yerel günlüğüyle işlemsel olarak tutarlıdır. Etkin kuyruk yöneticisi yönetim ortamı değiştirilirse ya da yedek yönetim ortamına geçilemezse, yedek yönetim ortamı hemen devam edebilir. Yedek yönetim ortamı olağan durumda iletileri kaybetmeden ya da yinelemeden devam eder. İletiler yalnızca son denetim noktasından bu yana uzak günlük kaydı başarısız olduğunda kaybolur ya da yinelenir ve daha önce etkin olan kuyruk yöneticisi yeniden başlatılamaz.
2. Sistem işletmeni, etkin örneği barındıran sistem yeniden kullanılabilir olduktan kısa bir süre sonra *ASYNCC uzak günlüğünü *SYNCC olarak değiştirmelidir. İşletmen, uzak günlüğü *SYNCC olarak değiştirmeden önce uzak günlüğün yakalanmasını bekleyebilir. Diğer bir seçenek olarak, işletmen uzak yönetim ortamını hemen *SYNCC olarak değiştirebilir ve etkin yönetim ortamını yedek yönetim ortamı günlüğü yakalanıncaya kadar beklemeye zorlayabilir. Uzak günlük kaydı *SYNCC olarak ayarlandığında, yedek yönetim ortamı genellikle etkin yönetim ortamıyla işlemsel olarak tutarlıdır. İletiler yalnızca son denetim noktasından bu yana uzak günlük kaydı başarısız olduğunda kaybolur ya da yinelenir ve daha önce etkin olan kuyruk yöneticisi yeniden başlatılamaz.
3. Konfigürasyon bir yedek sisteme geçiş ya da yedek sisteme geçiş işlevinden geri yüklendiğinde, uzak günlüğün barındırıldığı sunucu her zaman kullanılabilir olmalıdır.

Yedek kuyruk yöneticisinin hata durumunda yedek sisteme geçiş işleminden sonra hızlı bir şekilde devam etmesini istiyorsanız *SYNCC / *ASYNCC seçeneğini belirleyin. Yeni etkin yönetim ortamında uzak günlük ayarını *SYNCC değerine el ile geri yüklemeniz gerekir. *SYNCC / *ASYNCC konfigürasyonu, çok eşgörünümlü bir kuyruk yöneticisi çiftini yönetme kalıbıyla eşleşir. Bir yönetim ortamı başarısız olduktan sonra, yedek yönetim ortamı yeniden başlatılmadan önce etkin yönetim ortamının başarısız olamayacağı bir süre vardır.

***ASYNCC / *ASYNCC**

Etkin ve yedek kuyruk yöneticilerini barındıran her iki sunucunun konfigürasyonu *ASYNCC uzak günlük kaydını kullanacak şekilde tanımlanmıştır.

1. Yedek sisteme geçiş ya da yedek sisteme geçiş gerçekleştiğinde, kuyruk yöneticisi yeni sunucudaki günlükle devam eder. Yedek sisteme geçiş ya da yedek sisteme geçiş gerçekleştiğinde günlük eşzamanlanmayabilir. Sonuç olarak iletiler kaybolabilir ya da çoğaltılabilir.
2. Yedek kuyruk yöneticisini barındıran sunucu kullanılamasa da etkin yönetim ortamı çalışır. Yerel günlük, kullanılabilir olduğunda yedek sunucuyla zamanuyumsuz olarak kopyalanır.
3. Yerel kuyruk yöneticisinin başarımı, uzak günlük kaydından etkilenmez.

Başarım temel gereksiniminizse, *ASYNCC / *ASYNCC seçeneğini belirleyin ve hata durumunda yedek sisteme geçiş ya da yedek sisteme geçiş işleminden sonra bazı iletileri kaybetmeye ya da yinelemeye hazırsınız.

***ASYNCC / *SYNCC**

Bu seçenek birleşimini kullanmak için bir neden yoktur.

Uzak günlükten kuyruk yöneticisi etkinleştirilmesi

Günlükler zamanuyumlu ya da zamanuyumsuz olarak kopyalanır. Uzak günlük etkin olmayabilir ya da yerel günlüğe yetişiyor olabilir. Uzak günlük, zamanuyumlu olarak kopyalanmış olsa bile, yakın zamanda etkinleştirilmiş olabileceği için arayı kapatıyor olabilir. Kuyruk yöneticisinin başlatma sırasında kullandığı uzak günlüğün durumuna uyguladığı kurallar şunlardır:

1. Bekleme kipindeki uzak günlükten yeniden kayıttan yürümesi gerekiyorsa ve günlük durumu *FAILED ya da *INACTPENDise, bekleme başlatma işlemi başarısız olur.

2. Yedek veritabanını etkinleştirme işlemi başladığında, yedek veritabanındaki uzak günlük durumu *ACTIVE ya da *INACTIVE olmalıdır. Durum *INACTIVE ise, tüm günlük verileri kopyalanmamışsa etkinleştirme başarısız olabilir.

Ağ dosya sistemindeki kuyruk yöneticisi verilerinin uzak günde olandan daha yeni bir denetim noktası kaydı varsa bu hata oluşur. Uzak günlük, denetim noktaları arasındaki varsayılan 30 dakikalık aralık içinde iyi bir şekilde etkinleştirildiği sürece, hata oluşma olasılığı düşüktür. Yedek kuyruk yöneticisi, dosya sisteminden daha yeni bir denetim noktası kaydını okursa, başlamaz.

Bir seçeneğiniz vardır: Etkin sunucudaki yerel günlükün geri yüklenmesini bekleyin ya da yedek kuyruk yöneticisini soğuk olarak başlatın. Soğuk başlatmayı seçerseniz, kuyruk yöneticisi günlük verisi olmadan başlar ve dosya sistemindeki kuyruk yöneticisi verilerinin tutarlılığına ve eksiksizliğine dayanır.

Not: Bir kuyruk yöneticisini soğuk bir şekilde başlatırsanız, son denetim noktasından sonra iletileri kaybetme ya da yinleme riskini çalıştırırsınız. İletiler hareketleri günlüğe yazıldı, ancak hareketlerin bazıları dosya sistemindeki kuyruk yöneticisi verilerine yazılmamış olabilir. Bir kuyruk yöneticisini soğuk başlattığınızda, yeni bir günlük başlatılır ve dosya sistemindeki kuyruk yöneticisi verilerine yazılmayan işlemler kaybolur.

3. Yedek kuyruk yöneticisi etkinleştirilmesi, yedekteki uzak günlük durumunun *ASYNCPEND ya da *SYNCPEND değerini *ASYN ya da *SYN olarak değişmesini bekler. İletiler, yürütme denetleyicisinin iş günlüğüne düzenli aralıklarla yazılır.

Not: Bu durumda etkinleştirme, etkinleştirilmekte olan yedek kuyruk yöneticisinin yerel uzak günlükünü bekliyor. Kuyruk yöneticisi, uzak günlük olmadan devam etmeden önce de bir süre bekler. Uzak günlüğüne (ya da günlüklerine) zamanuymulu olarak yazmaya çalışıldığında bekler ve günlük kullanılamaz.

4. Günlük durumu *FAILED ya da *INACTPEND olarak değişirse etkinleştirme durur.

Etkinleştirmede kullanılacak yerel ve uzak günlüklerin adları ve durumları, kuyruk yöneticisi hata günlüğüne yazılır.

IBM i IBM i üzerinde NetServer ve günlük ikizleme kullanılarak çok eşgörünümlü kuyruk yöneticisinin yaratılması

İki IBM i sunucusunda çalışacak çok eşgörünümlü bir kuyruk yöneticisi yaratın. Kuyruk yöneticisi verileri, NetServer kullanılarak üçüncü bir IBM i sunucusunda saklanır. Kuyruk yöneticisi günlüğü, uzak günlük kaydı kullanılarak iki sunucu arasında ikizlenir. **ADDMQMARN** komutu, uzak günlüklerin yaratılmasını kolaylaştırmak için kullanılır.

Başlamadan önce

1. Görev için üç IBM i sunucusu gerekir. Örnekteki ALPHA ve BETA olmak üzere iki örnek üzerine IBM MQ kurun. ürün en az IBM WebSphere MQ 7.0.1 Fix Pack 1 olmalıdır.
2. Üçüncü sunucu, NetServer tarafından ALPHA ve BETA ' ya bağlanan bir IBM i sunucusudur. Kuyruk yöneticisi verilerini paylaşmak için kullanılır. IBM MQ kuruluşu olması gerekmez. IBM MQ ' yi geçici bir adım olarak sunucuya kurmak, kuyruk yöneticisi izinlerini ve izinlerini ayarlamak için yararlıdır.
3. QMQM kullanıcı profilinin üç sunucuda da aynı parolaya sahip olduğundan emin olun.
4. IBM i NetServer ürününü kurun; bkz. [i5/OS NetServer](#).

Bu görev hakkında

Şekil 25 sayfa 437 içinde gösterilen yapılandırmayı oluşturmak için aşağıdaki adımları gerçekleştirin. Kuyruk yöneticisi verileri IBM i NetServer kullanılarak bağlanır.

- ALPHA ve BETA ' dan kuyruk yöneticisi verilerini depolamak için GAMMA üzerindeki izin paylaşımına bağlantılar oluşturun. Görev ayrıca gerekli izinleri, kullanıcı profillerini ve parolaları da ayarlar.

- Kuyruk yöneticisi eşgörünümlerini çalıştıracak IBM i sistemlerine RDBE (Relational Database Entries; İlişkisel Veri Tabanı Girişleri) ekleyin. RDBE girişleri, uzak günlük kaydı için kullanılan IBM i sistemlerine bağlanmak için kullanılır.
- ALPHA IBM i sunucusunda QM1 kuyruk yöneticisini yaratın.
- Diğer IBM i sunucusunda (BETA) QM1 için kuyruk yöneticisi denetim bilgilerini ekleyin.
- Her iki kuyruk yöneticisi yönetim ortamı için her iki IBM i sunucusunda uzak günlükler yaratın. Her kuyruk yöneticisi yerel günlüğe yazar. Yerel günlük uzak günlüğe kopyalanır. **ADDMQMJRN** komutu, günlüklerin ve bağlantıların eklenmesini kolaylaştırır.
- Yedek yönetim ortamına izin veren kuyruk yöneticisini başlatın.

Yordam

1. Görevi yapın, [“IBM i üzerinde NetServer kullanılarak kuyruk yöneticisi verileri için ağ paylaşımı yaratılması” sayfa 422.](#)

Sonuç olarak, ALPHA ve BETA 'nın GAMMA 'da /QIBM/UserData/mqm/qmgrs ' a işaret eden bir payı (/QNTC/GAMMA/WMQ) vardır. QMQM ve QMQMADM kullanıcı profilleri gerekli izinlere sahiptir ve QMQM üç sistemde de eşleşen parolalara sahiptir.

2. IBM i sistemlerine, anasistem kuyruk yöneticisi yönetim ortamlarına gidecek ilişkisel veri tabanı girişleri (RDBE) ekleyin.

- a) ALPHA ' da BETA ile bağlantı oluşturun.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

- b) BETA 'da ALPHA' ya bağlantı oluşturun.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. GAMMA ' da kuyruk yöneticisi verilerini kaydederek ALPHA üzerinde QM1 kuyruk yöneticisini oluşturun.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIRP(' /QNTC/GAMMA/WMQ ')
```

Yol, kuyruk yöneticisi verilerini yaratmak için NetServer ' ı kullanır.

4. ALPHA ' ya koş. Komut, BETA ' ya uzak bir günlük ekler.

```
ADDMQMJRN MQMNAME(QM1) RMTJRN RDB(BETA)
```

Etkin eşgörünümü ALPHA olduğunda, ALPHA üzerindeki yerel günlüğünde günlük girişleri yaratır. ALPHA 'daki yerel günlük BETA' daki uzak günlüğe kopyalanır.

5. IBM MQ yapılandırma verilerini ALPHA üzerinde incelemek için bu komutu kullanın.

Bilgiler sonraki adımda gereklidir.

Bu örnekte, aşağıdakiler için ALPHA üzerinde aşağıdaki yapılandırma yaratılır:

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

6. Komutu kullanarak BETA üzerinde QM1 kuyruk yöneticisi örneği oluşturun. BETA üzerinde kuyruk yöneticisi denetim bilgilerini değiştirmek için BETA üzerinde aşağıdaki komutu çalıştırın.

```
ADDMQMINF MQMNAME(QM1)
PREFIX(' /QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMQM1)
DATAPATH(' /QNTC/GAMMA/WMQ /QM1 ')
```

İpucu: Yapılandırma bilgilerini kopyalayıp yapıştırın. Kuyruk yöneticisi kısmı ALPHA ve BETA ' da aynıdır.

7. BETA üzerinde çalıştır. Komut, BETA 'da yerel bir günlük ve ALPHA' da uzak bir günlük ekler.

```
ADDMQMJRN MQMNAME(QM1) RMTJRNRDB(ALPHA)
```

Etkin eşgörünümü BETA olduğunda, BETA üzerindeki yerel günlüğünde günlük girişleri yaratır. BETA 'daki yerel günlük, ALPHA' daki uzak günlüğe kopyalanır.

Not: Diğer bir seçenek olarak, zamanuyumsuz günlük kaydı kullanarak BETA 'dan ALPHA' ya uzaktan günlük kaydı ayarlamak isteyebilirsiniz.

“7” sayfa 436. adımda komut yerine BETA 'dan ALPHA' ya zamanuyumsuz günlük kaydı ayarlamak için bu komutu kullanın.

```
ADDMQMJRN MQMNAME (QM1) RMTJRNRDB (ALPHA) RMTJRNDLV (*ASYNCR)
```

Hatanın kaynağı sunucu ya da ALPHA üzerindeki günlük kaydı ise, BETA, yeni günlük girişlerinin ALPHA ' ya kopyalanmasını beklemeden başlar.

ALPHA yeniden çevrimiçi olduğunda, komutu kullanarak eşleme kipini *SYNColarak değiştirin.

Günlüklerin zamanuyumlu, zamanuyumsuz olarak mı, yoksa her ikisinin karışımı olarak mı ikizleneceğine karar vermek için “IBM i üzerinde ASP için ikizlenmiş günlük yapılandırması” sayfa 429 içindeki bilgileri kullanın. Varsayılan değer, uzak günlükten gelen bir yanıt için 60 saniyelik bekleme süresi ile zamanuyumlu olarak eşlenmesidir.

8. ALPHA ve BETA üzerindeki günlüklerin etkinleştirildiğini ve uzak günlük kopyalama durumunun geçerli kılındığını doğrulayın.

a) -Alfa 'da.

```
WRKMQMJRN MQMNAME(QM1)
```

b) Beta 'da:

```
WRKMQMJRN MQMNAME(QM1)
```

9. ALPHA ve BETA ' da kuyruk yöneticisi eşgörünümünü başlatın.

a) ALPHA ' da ilk örneği başlatın ve onu etkin örnek haline getirin. Yedek yönetim ortamına geçiş etkinleştiriliyor.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) İkinci eşgörünümü BETA üzerinde başlatın ve yedek eşgörünüm haline getirin.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

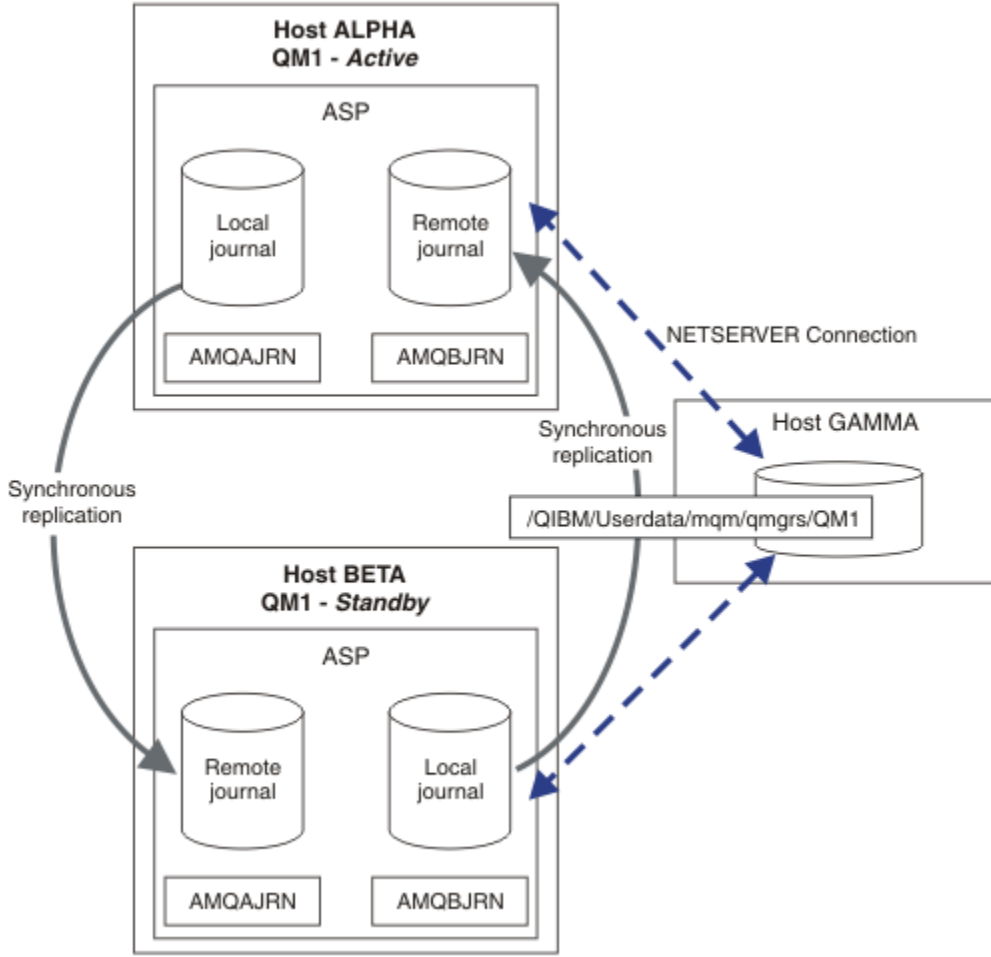
Sonuçlar

Kuyruk yöneticisi durumunu denetlemek için kullanılır:

1. ALPHA üzerindeki kuyruk yöneticisi eşgörünümünün durumu şu olmalıdır.

2. BETA ' daki kuyruk yöneticisi eşgörünümünün durumu şu olmalıdır.

Örnek



Şekil 25. İkizlenmiş günlük konfigürasyonu

Sonraki adım

• Etkin ve yedek yönetim ortamlarının otomatik olarak değiştiğini doğrulayın. Geçiş işlemini sınamak için örnek yüksek kullanılabilirlik örnek programlarını çalıştırabilirsiniz; bkz. [High availability sample programs](#). Örnek programlar ' C ' istemcileri. Bunları bir Windows ya da Unix altyapısından çalıştırabilirsiniz.

1. Yüksek kullanılabilirlikli örnek programları başlatın.
2. ALPHA ' da, kuyruk yöneticisini sona erdirin ve geçiş için istekte bulunun:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. BETA üzerindeki eşgörünümün etkin olup olmadığını denetleyin.
4. ALPHA ' da yeniden başlat

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

• Alternatif yüksek kullanılabilirlikli yapılandırmalara bakın:

1. Kuyruk yöneticisi verilerini bir Windows sunucusuna yerleştirmek için NetServer 'ı kullanın.
2. Kuyruk yöneticisi günlüğünü ikizlemek için uzak günlük kaydını kullanmak yerine, günlüğü bağımsız bir ASP 'de saklayın. Bağımsız ASP 'yi ALPHA' dan BETA 'ya aktarmak için IBM i kümeleme özelliğini kullanın.

IBM i NetServer ve IBM i üzerinde günlük ikizleme kullanılarak tek bir eşgörünüm kuyruk yöneticisinin çok eşgörünümlü bir kuyruk yöneticisine dönüştürülmesi

Tek bir eşgörünüm kuyruk yöneticisini çok eşgörünümlü bir kuyruk yöneticisine dönüştürün. Kuyruk yöneticisi verilerini NetServer ile bağlı bir ağ paylaşımına taşıyın. Uzak günlük kaydı kullanarak kuyruk yöneticisi günlüğünü ikinci bir IBM i sunucusuna ikizleyebilirsiniz.

Başlamadan önce

1. Görev için üç IBM i sunucusu gerekir. Örnekteki ALPHA sunucusunda var olan IBM MQ kuruluşu en az IBM WebSphere MQ 7.0.1 Fix Pack 1 olmalıdır. ALPHA, örnekte QM1 adlı bir kuyruk yöneticisini çalıştırıyor.
2. Örnekte BETA olan ikinci IBM i sunucusuna IBM MQ yazılımını kurun.
3. Üçüncü sunucu, NetServer tarafından ALPHA ve BETA 'ya bağlanan bir IBM i sunucusudur. Kuyruk yöneticisi verilerini paylaşmak için kullanılır. IBM MQ kuruluşu olması gerekmez. IBM MQ 'yi geçici bir adım olarak sunucuya kurmak, kuyruk yöneticisi dizinlerini ve izinlerini ayarlamak için yararlıdır.
4. QMQM kullanıcı profilinin üç sunucuda da aynı parolaya sahip olduğundan emin olun.
5. IBM i NetServer ürününü kurun; bkz. [i5/OS NetServer](#).

Bu görev hakkında

Tek bir yönetim ortamı kuyruk yöneticisini [Şekil 26 sayfa 442](#) içinde gösterilen çok eşgörünümlü kuyruk yöneticisine dönüştürmek için aşağıdaki adımları izleyin. Tek eşgörünüm kuyruk yöneticisi görevde silinir ve daha sonra, kuyruk yöneticisi verilerini NetServer ile bağlı ağ paylaşımında saklayarak yeniden yaratılır. Bu yordam, kuyruk yöneticisi dizinlerinin ve dosyalarının **CPY** komutunu kullanarak ağ paylaşımına taşınmasından daha güvenilirdir.

- ALPHA ve BETA 'dan kuyruk yöneticisi verilerini depolamak için GAMMA üzerindeki izin paylaşımına bağlantılar oluşturun. Görev ayrıca gerekli izinleri, kullanıcı profillerini ve parolaları da ayarlar.
- Kuyruk yöneticisi eşgörünümünü çalıştıracak IBM i sistemlerine RDBE (Relational Database Entries; İlişkisel Veri Tabanı Girişleri) ekleyin. RDBE girişleri, uzak günlük kaydı için kullanılan IBM i sistemlerine bağlanmak için kullanılır.
- Kuyruk yöneticisi günlüklerini ve tanımlamalarını saklayın, kuyruk yöneticisini durdurun ve silin.
- Kuyruk yöneticisi verilerini GAMMA üzerindeki ağ paylaşımında saklayarak kuyruk yöneticisini yeniden oluşturun.
- Kuyruk yöneticisinin ikinci eşgörünümünü diğer sunucuya ekleyin.
- Her iki kuyruk yöneticisi yönetim ortamı için her iki IBM i sunucusunda uzak günlükler yaratın. Her kuyruk yöneticisi yerel günlüğe yazar. Yerel günlük uzak günlüğe kopyalanır. **ADDMQMJRN** komutu, günlüklerin ve bağlantıların eklenmesini kolaylaştırır.
- Yedek yönetim ortamına izin veren kuyruk yöneticisini başlatın.

Not:

Görevin [“4” sayfa 439](#) . adımında, QM1 adlı tek eşgörünüm kuyruk yöneticisini silebilirsiniz. Kuyruk yöneticisi silindiğinde, kuyruklardaki tüm kalıcı iletiler silinir. Bu nedenle, kuyruk yöneticisini dönüştürmeden önce kuyruk yöneticisi tarafından saklanan tüm iletilerin işlenmesini tamamlayın. Tüm iletilerin işlenmesi mümkün değilse, [“4” sayfa 439](#) . adımdan önce kuyruk yöneticisi kitaplığını yedek. [“5” sayfa 439](#) . adımdan sonra kuyruk yöneticisi kitaplığını geri yükleyin.

Not:

Görevin “5” sayfa 439 . adımımda, QM1öğesini yeniden oluşturun. Kuyruk yöneticisi aynı ada sahip olsa da, farklı bir kuyruk yöneticisi tanıtıcısına sahiptir. Kuyruk yöneticisi kümelemesi, kuyruk yöneticisi tanıtıcısını kullanır. Bir kümede kuyruk yöneticisini silmek ve yeniden yaratmak için önce kümeden kuyruk yöneticisini kaldırmanız gerekir; bkz. Kümeden kuyruk yöneticisinin kaldırılması: Alternatif yöntem ya da Kümeden kuyruk yöneticisinin kaldırılması. Kuyruk yöneticisini yeniden yarattığınızda kümeye ekleyin. Önceki adla aynı olmasına rağmen, kümedeki diğer kuyruk yöneticileri için yeni bir kuyruk yöneticisi gibi görünüyor.

Yordam

1. Görevi yapın, “IBM i üzerinde NetServer kullanılarak kuyruk yöneticisi verileri için ağ paylaşımı yaratılması” sayfa 422.

Sonuç olarak, ALPHA ve BETA 'nın GAMMA' da /QIBM/UserData/mqm/qmgrs ' a işaret eden bir payı (/QNTC/GAMMA/WMQ) vardır. QMQM ve QMQMADM kullanıcı profilleri gerekli izinlere sahiptir ve QMQM üç sistemde de eşleşen parolalara sahiptir.

2. IBM i sistemlerine, anasistem kuyruk yöneticisi yönetim ortamlarına gidecek ilişkiyel veri tabanı girişleri (RDBE) ekleyin.

- a) ALPHA ' da BETA ile bağlantı oluşturun.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

- b) BETA 'da ALPHA' ya bağlantı oluşturun.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. Kuyruk yöneticisi nesnelere yeniden yaratan komut dosyalarını oluşturun.

```
QSAVEQMGR LCLQMGRNAM(QM1) FILENAME(' *CURLIB/QMQSC (QM1) ')  
OUTPUT(*REPLACE) MAKEAUTH(*YES) AUTHFN(' *CURLIB/QMAUT (QM1) ')
```

4. Kuyruk yöneticisini durdurun ve silin.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ENDCCTJOB(*YES) RCDMQMIMG(*YES) TIMEOUT(15)  
DLTMQM MQMNAME(QM1)
```

5. GAMMA ' da kuyruk yöneticisi verilerini kaydederek ALPHA üzerinde QM1 kuyruk yöneticisini oluşturun.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)  
MQMDIRP(' /QNTC/GAMMA/WMQ ')
```

Yol, kuyruk yöneticisi verilerini yaratmak için NetServer ' ı kullanır.

6. Kaydedilen tanımlamalardan QM1 için kuyruk yöneticisi nesnelere yeniden yaratın.

```
STRMQMQSC SRCMBR(QM1) SRCFILE(*CURLIB/QMQSC) MQMNAME(QM1)
```

7. Kaydedilen bilgilerden yetkileri uygulayın.

- a) Saklanan yetki programını derleyin.

```
CRTCLPGM PGM(*CURLIB/QM1) SRCFILE(*CURLIB/QMAUT)  
SRCMBR(QM1) REPLACE(*YES)
```

- b) Yetkileri uygulamak için programı çalıştırın.

```
CALL PGM(*CURLIB/QM1)
```

c) QM1 ile ilgili güvenlik bilgilerini yenileyin.

```
RFRMQMAUT MQMNAME(QM1)
```

8. ALPHA ' ya koş. Komut, BETA ' ya uzak bir günlük ekler.

```
ADDQMJRNRN MQMNAME(QM1) RMTJRNRDB(BETA)
```

Etkin eşgörünümü ALPHA olduğunda, ALPHA üzerindeki yerel günlüğünde günlük girişleri yaratır. ALPHA 'daki yerel günlük BETA' daki uzak günlüğe kopyalanır.

9. IBM MQ yapılanış verilerini ALPHA üzerinde incelemek için bu komutu kullanın.

Bilgiler sonraki adımda gereklidir.

Bu örnekte, aşağıdakiler için ALPHA üzerinde aşağıdaki yapılanış yaratılır:

```
Name=QM1  
Prefix=/QIBM/UserData/mqm  
Library=QMQM1  
Directory=QM1  
DataPath= /QNTC/GAMMA/WMQ /QM1
```

10. Komutu kullanarak BETA üzerinde QM1 kuyruk yöneticisi örneği oluşturun. BETA üzerinde kuyruk yöneticisi denetim bilgilerini değiştirmek için BETA üzerinde aşağıdaki komutu çalıştırın.

```
ADDQMINF MQMNAME(QM1)  
PREFIX('/QIBM/UserData/mqm')  
MQMDIR(QM1)  
MQMLIB(QMQM1)  
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

İpucu: Yapılandırma bilgilerini kopyalayıp yapıştırın. Kuyruk yöneticisi kısmı ALPHA ve BETA ' da aynıdır.

11. BETA üzerinde çalıştır. Komut, BETA 'da yerel bir günlük ve ALPHA' da uzak bir günlük ekler.

```
ADDQMJRNRN MQMNAME(QM1) RMTJRNRDB(ALPHA)
```

Etkin eşgörünümü BETA olduğunda, BETA üzerindeki yerel günlüğünde günlük girişleri yaratır. BETA 'daki yerel günlük, ALPHA' daki uzak günlüğe kopyalanır.

Not: Diğer bir seçenek olarak, zamanuyumsuz günlük kaydı kullanarak BETA 'dan ALPHA' ya uzaktan günlük kaydı ayarlamak isteyebilirsiniz.

“7” sayfa 436. adımda komut yerine BETA 'dan ALPHA' ya zamanuyumsuz günlük kaydı ayarlamak için bu komutu kullanın.

```
ADDQMJRNRN MQMNAME (QM1) RMTJRNRDB (ALPHA) RMTJRNDLV (*ASYNCR)
```

Hatanın kaynağı sunucu ya da ALPHA üzerindeki günlük kaydı ise, BETA, yeni günlük girişlerinin ALPHA ' ya kopyalanmasını beklemeden başlar.

ALPHA yeniden çevrimiçi olduğunda, komutu kullanarak eşleme kipini *SYNCR olarak değiştirin.

Günlüklerin zamanuyumlu, zamanuyumsuz olarak mı, yoksa her ikisinin karışımı olarak mı ikizleneceğine karar vermek için “IBM i üzerinde ASP için ikizlenmiş günlük yapılandırması” sayfa 429 içindeki bilgileri kullanın. Varsayılan değer, uzak günlükten gelen bir yanıt için 60 saniyelik bekleme süresi ile zamanuyumlu olarak eşlenmesidir.

12. ALPHA ve BETA üzerindeki günlüklerin etkinleştirildiğini ve uzak günlük kopyalama durumunun geçerli kılındığını doğrulayın.

a) -Alfa 'da.

```
WRKMQMJRN MQMNAME(QM1)
```

b) Beta 'da:

```
WRKMQMJRN MQMNAME(QM1)
```

13. ALPHA ve BETA ' da kuyruk yöneticisi eşgörünümlerini başlatın.

a) ALPHA ' da ilk örneği başlatın ve onu etkin örnek haline getirin. Yedek yönetim ortamına geçiş etkinleştiriliyor.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) İkinci eşgörünümü BETA üzerinde başlatın ve yedek eşgörünüm haline getirin.

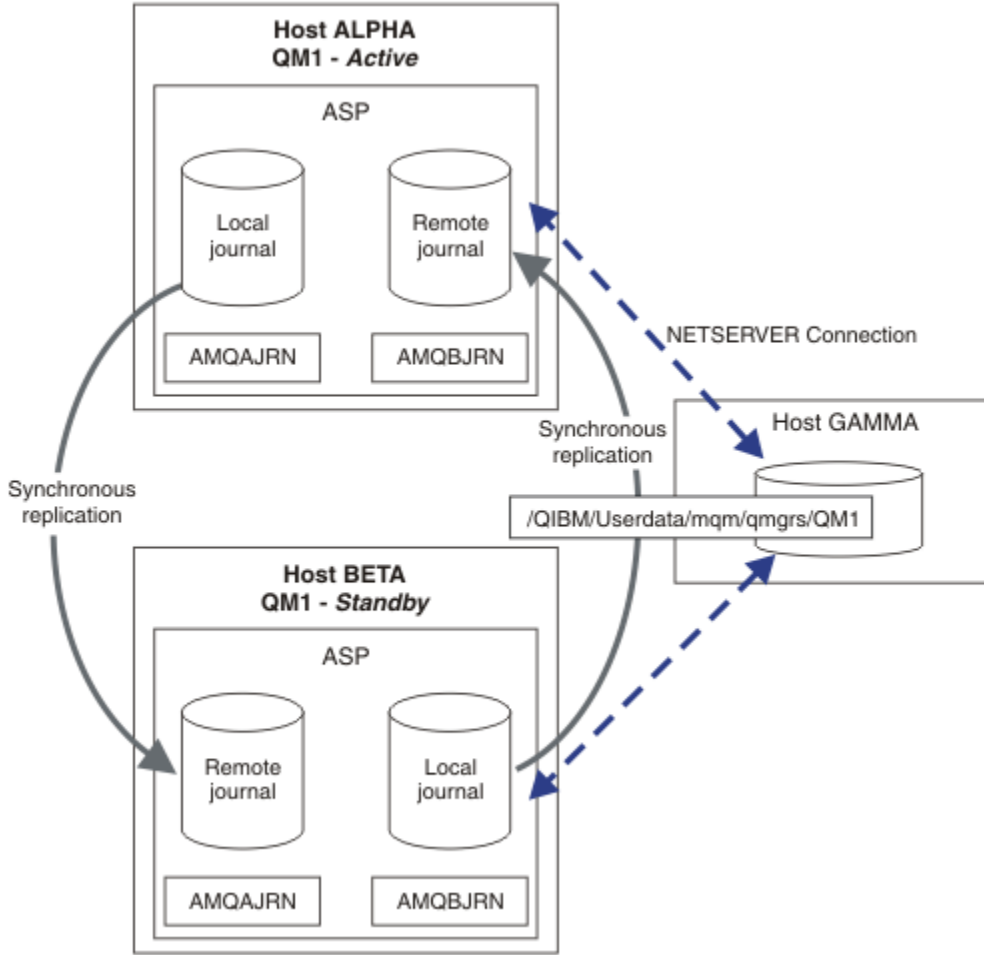
```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

Sonuçlar

Kuyruk yöneticisi durumunu denetlemek için kullanılır:

1. ALPHA üzerindeki kuyruk yöneticisi eşgörünümünün durumu şu olmalıdır.
2. BETA ' daki kuyruk yöneticisi eşgörünümünün durumu şu olmalıdır.

Örnek



Şekil 26. İkizlenmiş günlük konfigürasyonu

Sonraki adım

- Etkin ve yedek yönetim ortamlarının otomatik olarak değiştiğini doğrulayın. Geçiş işlemini sınamak için örnek yüksek kullanılabilirlik örnek programlarını çalıştırabilirsiniz; bkz. [High availability sample programs](#). Örnek programlar 'C' istemcileri. Bunları bir Windows ya da Unix altyapısından çalıştırabilirsiniz.

1. Yüksek kullanılabilirlikli örnek programları başlatın.
2. ALPHA ' da, kuyruk yöneticisini sona erdirin ve geçiş için istekte bulunun:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. BETA üzerindeki eşgörünümün etkin olup olmadığını denetleyin.
4. ALPHA ' da yeniden başlat

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Alternatif yüksek kullanılabilirlikli yapılandırmalara bakın:

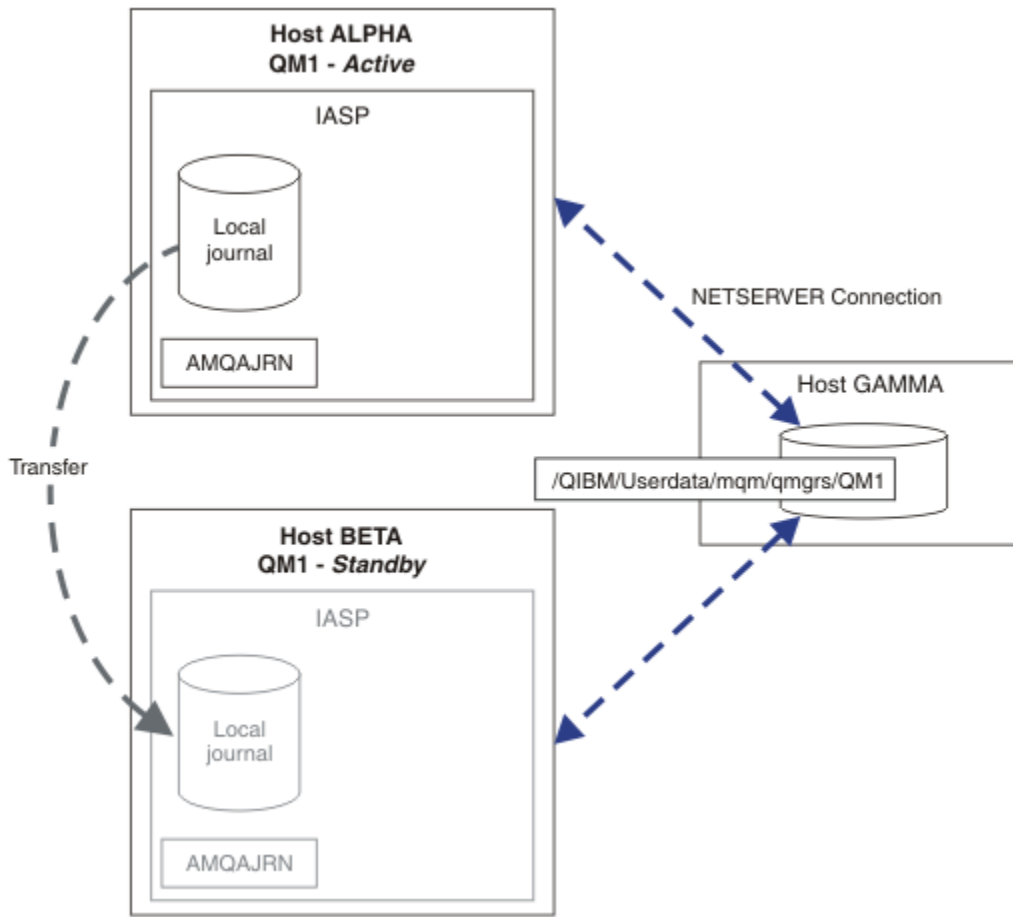
1. Kuyruk yöneticisi verilerini bir Windows sunucusuna yerleştirmek için NetServer ' ı kullanın.

2. Kuyruk yöneticisi günlüğünü ikizlemek için uzak günlük kaydını kullanmak yerine, günlüğü bağımsız bir ASP 'de saklayın. Bağımsız ASP 'yi ALPHA' dan BETA 'ya aktarmak için IBM i kümeleme özelliğini kullanın.

IBM i **IBM i üzerinde anahtarlamalı bağımsız ASP günlük yapılandırması**

Çok eşgörünümlü bir kuyruk yöneticisi konfigürasyonu yaratmak için bağımsız bir ASP günlüğünü eşlemeniz gerekmez. Bağımsız ASP 'yi etkin kuyruk yöneticisinden yedek kuyruk yöneticisine aktarmak için bir aracı otomatikleştirmeniz gerekir. Çok eşgörünümlü bir kuyruk yöneticisinin kullanılmasını gerektirmeyen, bağımsız bir ASP kullanılarak kullanılabilir alternatif yüksek kullanılabilirlik çözümleri vardır.

Bağımsız bir ASP kullanırken, kuyruk yöneticisi günlüğünü ikizlemeniz gerekmez. Küme yönetimini kurduysanız ve kuyruk yöneticisi yönetim ortamlarını barındıran sunucular aynı küme kaynağı grubundaysa, etkin yönetim ortamını çalıştıran anasistem başarısız olursa, kuyruk yöneticisi günlüğü etkin sunucunun kısa bir mesafesi içinde otomatik olarak başka bir sunucuya aktarılabilir. Ayrıca, planlı bir anahtarın parçası olarak günlüğü el ile aktarabilir ya da bağımsız ASP 'yi programlı olarak aktarmak için bir komut yordamı yazabilirsiniz.



Şekil 27. Bağımsız ASP kullanarak bir kuyruk yöneticisi günlüğünü aktarma

Çok eşgörünümlü kuyruk yöneticisi işlemi için, kuyruk yöneticisi verileri paylaşılan bir dosya sisteminde saklanmalıdır. Dosya sistemi çeşitli altyapılarda barındırılabilir. Çok eşgörünümlü kuyruk yöneticisi verilerini bir ASP ya da bağımsız ASP 'de saklayamazsınız.

Paylaşılan dosya sistemi yapılandırma iki rol gerçekleştirir: Aynı kuyruk yöneticisi verileri, kuyruk yöneticisinin tüm eşgörünümleri arasında paylaşılır. Dosya sisteminin, başlatıldıktan sonra kuyruk yöneticisinin yalnızca bir eşgörünümünün kuyruk yöneticisi verilerine erişmesini sağlayan güçlü bir kilitleme protokolü olmalıdır. Kuyruk yöneticisi başarısız olursa ya da dosya sunucusuyla iletişim kesilirse,

dosya sistemi etkin yönetim ortamı tarafından tutulan ve artık syste dosyasıyla iletişim kurmayan kuyruk yöneticisi verilerine ilişkin kilidi serbest bırakmalıdır. Yedek kuyruk yöneticisi yönetim ortamı, kuyruk yöneticisi verilerine okuma/yazma erişimi kazanabilir. Çok eşgörünümlü kuyruk yöneticileriyle doğru bir şekilde çalışmak için dosya sistemi iletişim kuralı bir kural kümesine uymalıdır; bkz. [“IBM i üzerinde yüksek düzeyde kullanılabilirlik çözümünün bileşenleri” sayfa 421.](#)

Kilitleme mekanizması, kuyruk yöneticisini başlatma komutunu diziselleştirir ve kuyruk yöneticisinin hangi eşgörünümlünün etkin olduğunu denetler. Bir kuyruk yöneticisi etkin duruma geldikten sonra, kuyruklarını sizin ya da HA kümesinin yedek sunucuya aktardığı yerel günlükten yeniden oluşturur. Aynı kuyruk yöneticisiyle yeniden bağlantı kurmayı bekleyen yeniden bağlanabilir istemciler yeniden bağlanır ve herhangi bir hareket geri çekilir. Kuyruk yöneticisi hizmetleri olarak başlatılmak üzere yapılandırılan uygulamalar başlatılır.

Küme kaynak yöneticisini yapılandırarak ya da bağımsız ASP 'yi el ile aktararak, bağımsız ASP' deki başarısız olan etkin kuyruk yöneticisi yönetim ortamındaki yerel günlüğün, yeni etkinleştirilen yedek kuyruk yöneticisi eşgörünümlünü barındıran sunucuya aktarıldığından emin olmanız gerekir. Bağımsız ASP 'lerin kullanılması, yedekleme ve olağanüstü durumdan kurtarma için bağımsız ASP' yi kullanmaya karar verirsiniz ve çok eşgörünümlü kuyruk yöneticisi yapılandırması için uzak günlük ikizlemeyi kullanmaya karar verirsiniz, uzak günlüklerin ve ikizlemenin yapılandırılmasını önlemez.

Bağımsız bir ASP kullanmayı seçtiyseniz, göz önünde bulundurabileceğiniz alternatif yüksek kullanılabilirlikli yapılandırmalar vardır. Bu çözümlerin arka planı [“Bağımsız ASP 'ler ve yüksek düzeyde kullanılabilirlik” sayfa 447](#) içinde açıklanmıştır.

1. Çok eşgörünümlü kuyruk yöneticilerini kullanmak yerine, tek bir eşgörünüm kuyruk yöneticisini tamamen bağımsız bir ASP ' ye kurun ve yapılandırın ve kuyruk yöneticisini başarısız yapmak için IBM i yüksek kullanılabilirlik hizmetlerini kullanın. Kuyruk yöneticisinin sunucudan bağımsız olarak başarısız olup olmadığını saptamak için çözümü bir kuyruk yöneticisi izleme programıyla genişletmeniz gerekir. Bu, *Supportpac MC41: Configuring IBM MQ for iSeries for High Availability* içinde sağlanan çözümün temelidir.
2. Bağımsız ASP 'yi yerel veriyolunda değiştirmek yerine bağımsız ASP' yi ikizlemek için bağımsız ASP ' leri ve siteler arası ikizlemeyi (XSM) kullanın. Bu, bağımsız ASP çözümünün coğrafi aralığını, günlük kayıtlarının uzun mesafeli olarak yazılması için geçen süreye kadar genişletir.

IBM i *IBM i üzerinde bağımsız bir ASP ve NetServer kullanarak çok eşgörünümlü kuyruk yöneticisi yaratılması*

İki IBM i sunucusunda çalışacak çok eşgörünümlü bir kuyruk yöneticisi yaratın. Kuyruk yöneticisi verileri, NetServer kullanılarak bir IBM i sunucusu saklanır. Kuyruk yöneticisi günlüğü bağımsız bir ASP ' de saklanır. Kuyruk yöneticisi günlüğünü içeren bağımsız ASP ' yi diğer IBM i sunucusuna aktarmak için IBM i kümelemesini ya da el ile yordamını kullanın.

Başlamadan önce

1. Görev için üç IBM i sunucusu gerekir. Örnekteki ALPHA ve BETA olmak üzere iki örnek üzerine IBM MQ kurun. ürün en az IBM WebSphere MQ 7.0.1 Fix Pack 1 olmalıdır.
2. Üçüncü sunucu, NetServer tarafından ALPHA ve BETA ' ya bağlanan bir IBM i sunucusudur. Kuyruk yöneticisi verilerini paylaşmak için kullanılır. IBM MQ kuruluşu olması gerekmez. IBM MQ ' yi geçici bir adım olarak sunucuya kurmak, kuyruk yöneticisi izinlerini ve izinlerini ayarlamak için yararlıdır.
3. QMQM kullanıcı profilinin üç sunucuda da aynı parolaya sahip olduğundan emin olun.
4. IBM i NetServer ürününü kurun; bkz. [i5/OS NetServer](#).
5. Bağımsız ASP ' yi başarısız olan kuyruk yöneticisinden devralmakta olan yedek veritabanına aktarmak için yordamlar oluşturun. *SupportPac MC41: Configuring IBM MQ for iSeries for High Availability* başlıklı konudaki tekniklerden bazılarını bağımsız ASP aktarım yordamlarınızın tasarlanmasında yardımcı olabilir.

Bu görev hakkında

Şekil 28 sayfa 446 içinde gösterilen yapılandırmayı oluşturmak için aşağıdaki adımları gerçekleştirin. Kuyruk yöneticisi verileri IBM i NetServer kullanılarak bağlanır.

- ALPHA ve BETA ' dan kuyruk yöneticisi verilerini depolamak için GAMMA üzerindeki dizin paylaşımına bağlantılar oluşturun. Görev ayrıca gerekli izinleri, kullanıcı profillerini ve parolaları da ayarlar.
- ALPHA IBM i sunucusunda QM1 kuyruk yöneticisini yaratın.
- Diğer IBM i sunucusunda (BETA) QM1 için kuyruk yöneticisi denetim bilgilerini ekleyin.
- Yedek yönetim ortamına izin veren kuyruk yöneticisini başlatın.

Yordam

1. Görevi yapın, “IBM i üzerinde NetServer kullanılarak kuyruk yöneticisi verileri için ağ paylaşımı yaratılması” sayfa 422.

Sonuç olarak, ALPHA ve BETA 'nın GAMMA' da /QIBM/UserData/mqm/qmqm ' a işaret eden bir payı (/QNTC/GAMMA/WMQ) vardır. QMQM ve QMQMADM kullanıcı profilleri gerekli izinlere sahiptir ve QMQM üç sistemde de eşleşen parolalara sahiptir.

2. GAMMA ' da kuyruk yöneticisi verilerini kaydederek ALPHA üzerinde QM1 kuyruk yöneticisini oluşturun.

```
CRTMQM MQMNAME(QM1) UDLSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIR(' /QNTC/GAMMA/WMQ ')
```

Yol, kuyruk yöneticisi verilerini yaratmak için NetServer ' ı kullanır.

3. IBM MQ yapısını verilerini ALPHA üzerinde incelemek için bu komutu kullanın.

Bilgiler sonraki adımda gereklidir.

Bu örnekte, aşağıdakiler için ALPHA üzerinde aşağıdaki yapılanış yaratılır:

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

4. Komutu kullanarak BETA üzerinde QM1 kuyruk yöneticisi örneği oluşturun. BETA üzerinde kuyruk yöneticisi denetim bilgilerini değiştirmek için BETA üzerinde aşağıdaki komutu çalıştırın.

```
ADDQMINF MQMNAME(QM1)
PREFIX(' /QIBM/UserData/mqm ')
MQMDIR(QM1)
MQMLIB(QMOM1)
DATAPATH(' /QNTC/GAMMA/WMQ /QM1 ')
```

İpucu: Yapılandırma bilgilerini kopyalayıp yapıştırın. Kuyruk yöneticisi kısmı ALPHA ve BETA ' da aynıdır.

5. ALPHA ve BETA ' da kuyruk yöneticisi eşgörünümlerini başlatın.

- a) ALPHA ' da ilk örneği başlatın ve onu etkin örnek haline getirin. Yedek yönetim ortamına geçiş etkinleştiriliyor.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- b) İkinci eşgörünümü BETA üzerinde başlatın ve yedek eşgörünüm haline getirin.

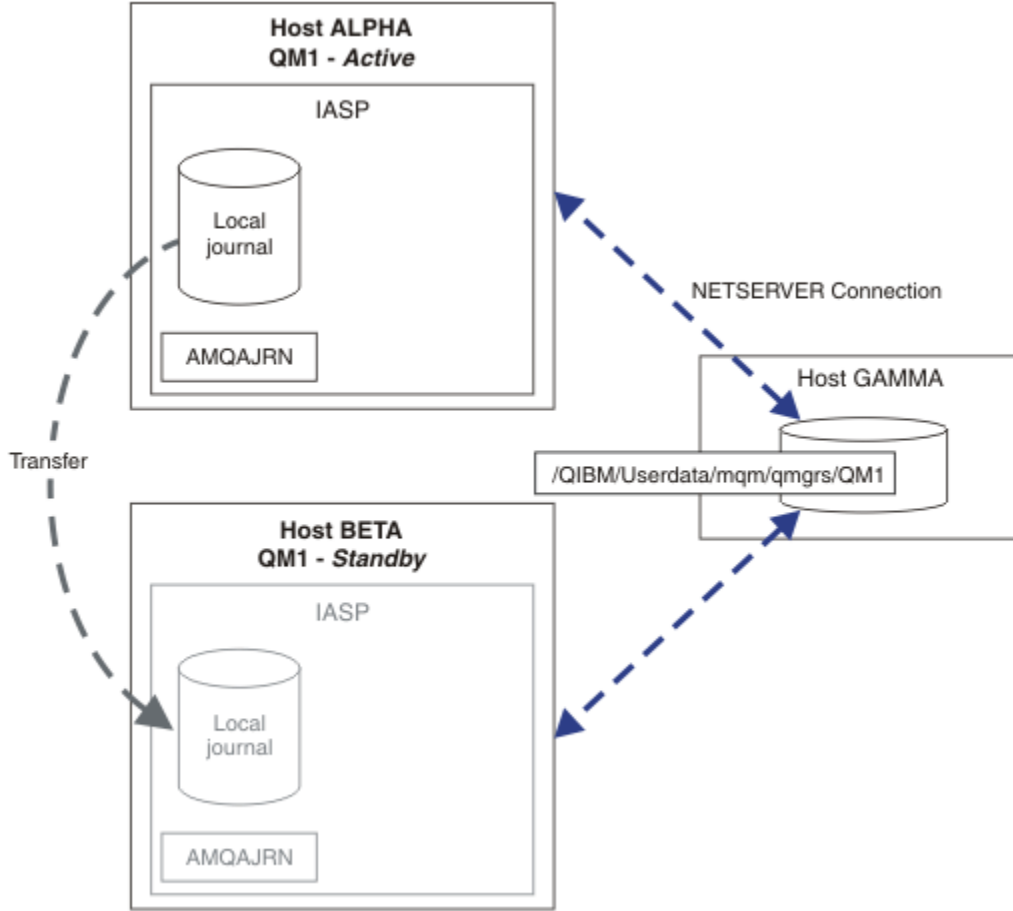
```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

Sonuçlar

Kuyruk yöneticisi durumunu denetlemek için kullanılır:

1. ALPHA üzerindeki kuyruk yöneticisi eşgörünümünün durumu şu olmalıdır.
2. BETA 'daki kuyruk yöneticisi eşgörünümünün durumu şu olmalıdır.

Örnek



Şekil 28. Bağımsız ASP kullanarak bir kuyruk yöneticisi günlüğünü aktarma

Sonraki adım

- Etkin ve yedek yönetim ortamlarının otomatik olarak değiştiğini doğrulayın. Geçiş işlemini sınamak için örnek yüksek kullanılabilirlik örnek programlarını çalıştırabilirsiniz; bkz. [High availability sample programs](#). Örnek programlar 'C' istemcileri. Bunları bir Windows ya da Unix altyapısından çalıştırabilirsiniz.

1. Yüksek kullanılabilirlikli örnek programları başlatın.
2. ALPHA 'da, kuyruk yöneticisini sona erdirin ve geçiş için istekte bulunun:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. BETA üzerindeki eşgörünümün etkin olup olmadığını denetleyin.
4. ALPHA 'da yeniden başlat

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Alternatif yüksek kullanılabilirlikli yapılandırmalara bakın:

1. Kuyruk yöneticisi verilerini bir IBM i sunucusuna yerleştirmek için NetServer ' ı kullanın.
2. Kuyruk yöneticisi günlüğünü yedek sunucuya aktarmak için bağımsız bir ASP kullanmak yerine, günlüğü yedek sunucuya ikizlemek için uzak günlük kaydını kullanın.

IBM i Bağımsız ASP ' ler ve yüksek düzeyde kullanılabilirlik

Bağımsız ASP 'ler, uygulamaların ve verilerin sunucular arasında taşınmasını sağlar. Bağımsız ASP 'lerin esnekliği, bunların bazı IBM i yüksek düzeyde kullanılabilirlik çözümlerinin temeli olduğu anlamına gelir. Kuyruk yöneticisi günlüğü için bir ASP mi, yoksa bağımsız ASP mi kullanacağınızı göz önünde bulundurarak, bağımsız ASP 'lere dayalı olarak diğer yüksek kullanılabilirlikli konfigürasyonları göz önünde bulundurmanız gerekir.

Yardımcı depolama havuzları (ASP), IBM i mimarisinin bir yapı taşıdır. Disk birimleri, tek bir ASP oluşturmak için birlikte gruplanır. Nesnelere farklı ASP 'lere yerleştirilerek, bir ASP' deki verileri başka bir ASP ' deki disk hatalarından etkilenmekten koruyabilirsiniz.

Her IBM i sunucusu, sistem ASP 'si olarak bilinen en az bir *temel* ASP 'ye sahiptir. ASP1olarak ve bazen *SYSBAS olarak bilinir. Aynı ad alanını paylaştığından, sistem ASP 'sinden ayırt edilemeyen en fazla 31 temel *kullanıcı* ASP 'nin konfigürasyonunu uygulamanın bakış açısından tanımlayabilirsiniz. Uygulamaları birçok diske dağıtmak için birden çok temel ASP kullanarak başarıyı artırabilir ve kurtarma süresini kısaltabilirsiniz. Birden çok temel ASP 'nin kullanılması, disk arızalarına karşı bir dereceye kadar yalıtım sağlayabilir, ancak genel olarak güvenilirliği artırmaz.

Bağımsız ASP 'ler özel bir ASP tipidir. Bunlara genellikle bağımsız disk havuzları denir. Bağımsız disk havuzları, IBM i yüksek kullanılabilirliğin temel bileşenidir. Kendilerini bağımsız disk depolama birimlerine bağlı oldukları geçerli sistemden bağımsız olarak kabul eden verileri ve uygulamaları saklayabilirsiniz. Değişimli kullanılabilir ya da değişimli kullanılamaz bağımsız ASP 'lerin konfigürasyonunu tanımlayabilirsiniz. Kullanılabilirlik açısından bakıldığında, genellikle, sunucudan sunucuya otomatik olarak aktarılabilen değişimli kullanılabilir bağımsız ASP 'lerle ilgilenir. Sonuç olarak, bağımsız ASP 'deki uygulamaları ve verileri sunucudan sunucuya taşıyabilirsiniz.

Temel kullanıcı ASP 'lerinden farklı olarak, bağımsız ASP'ler sistem ASP 'si ile aynı ad alanını paylaşmaz. Kullanıcı ASP 'leriyle çalışan uygulamalar, bağımsız bir ASP ile çalışmak için değişiklik gerektirir. Kullandığınız yazılımınızı ve üçüncü kişi yazılımınızı doğrulamanız gerekir, bağımsız bir ASP ortamında çalışır.

Bağımsız ASP farklı bir sunucuya eklendiğinde, bağımsız ASP 'nin ad alanı, sistem ASP 'sinin ad alanıyla birleştirilmeli. Bu sürece bağımsız ASP 'nin *değişken* adı verilir. Bağımsız bir ASP 'yi sunucuyu IPLing işlemi gerçekleştirilmeden kullanılabilir duruma alabilirsiniz. Bağımsız ASP 'lerin otomatik olarak bir sunucudan diğerine aktarılması için kümeleme desteği gerekir.

Bağımsız ASP 'lerle güvenilir çözümler oluşturma

Bir ASP 'ye günlük kaydı yapmak ve günlük işlemeyi kullanmak yerine bağımsız bir ASP'ye kaydetme, yedek kuyruk yöneticisine başarısız olan kuyruk yöneticisi eşgörünümünden yerel günlüğün bir kopyasını sağlamak için alternatif bir yöntem sağlar. Bağımsız ASP 'yi otomatik olarak başka bir sunucuya aktarmak için, kümeleme desteğini kurmuş ve yapılandırmış olmanız gerekir. Çok eşgörünümlü kuyruk yöneticilerini kullanarak, küme desteğine ve düşük düzeyli disk ikizlemesine dayalı olarak bağımsız ASP 'ler için çok sayıda yüksek kullanılabilirlikli çözüm vardır.

Aşağıdaki listede, bağımsız ASP 'lere dayalı güvenilir bir çözüm oluşturmak için gereken bileşenler açıklanmaktadır.

Günlük kaydı

Kuyruk yöneticileri ve diğer uygulamalar, sunucu arızası nedeniyle bellekteki veri kaybına karşı korumak üzere kalıcı verileri güvenli bir şekilde diske yazmak için yerel günlükleri kullanır. Bu bazen anlık tutarlılık olarak adlandırılır. Belirli bir süre içinde gerçekleşen birden çok güncellemenin tutarlılığını garanti etmez.

Kesinleştirme denetimi

Genel işlemleri kullanarak, günlüğe yazılan verilerin tutarlı olması için iletilere ve veritabanlarına ilişkin güncellemeleri koordine edebilirsiniz. İki aşamalı kesinleştirme protokolü kullanılarak belirli bir süre boyunca tutarlılık sağlar.

Anahtarlamalı disk

Anahtarlamalı diskler, HA kümesindeki aygıt kümesi kaynak grubu (CRG) tarafından yönetilir. CRG, planlanmamış bir kesinti durumunda bağımsız ASP ' leri otomatik olarak yeni bir sunucuya değiştirir. CRG ' ler coğrafi olarak yerel GÇ veriyolunun boyutlarıyla sınırlıdır.

Değişimli kullanılabilir bağımsız bir ASP ' de yerel günlüğünüzü yapılandırarak, günlüğü farklı bir sunucuya aktarabilir ve iletileri işlemeye devam edebilirsiniz. Bağımsız ASP başarısız olmadıkça, syncpoint denetimi olmadan ya da syncpoint denetimiyle kesinleştirilmeden kalıcı iletilerde yapılan hiçbir değişiklik kaybolmaz.

Değişimli kullanılabilir bağımsız ASP ' lerde hem günlük kaydı, hem de kesinleştirme denetimi kullanıyorsanız, veri tabanı günlüklerini ve kuyruk yöneticisi günlüklerini farklı bir sunucuya aktarabilir ve tutarlılık ya da kesinleştirilmiş hareket kaybı olmadan işlemleri sürdürmeye devam edebilirsiniz.

Siteler arası ikizleme (XSM)

XSM, birincil bağımsız ASP 'yi TCP/IP ağı üzerinden coğrafi olarak uzak ikincil bağımsız ASP' ye ikizler ve bir hata durumunda denetimi otomatik olarak aktarır. Zamanuyumlu ya da zamanuyumsuz bir ikizleme konfigürasyonu tanımlama seçeneğiniz vardır. Zamanuyumlu ikizleme, üretim sistemindeki yazma işlemleri tamamlanmadan önce veriler ikizlendiği, ancak ikincil bağımsız ASP ' nin güncel olduğu garanti edildiği için kuyruk yöneticisinin başarımını azaltır. Ancak zamanuyumsuz ikizleme kullanırsanız, ikincil bağımsız ASP ' nin güncel olduğunu garanti edemezsiniz. Zamanuyumsuz ikizleme, ikincil bağımsız ASP ' nin tutarlılığını korur.

Üç adet XSM teknolojisi vardır.

Coğrafi ikizleme

Coğrafi ikizleme, kümelemenin bir uzantısıdır ve bağımsız ASP ' leri geniş bir alanda değiştirmenizi sağlar. Hem zamanuyumlu hem de zamanuyumsuz kiplere sahiptir. Yalnızca zamanuyumlu kipte yüksek kullanılabilirliği garanti edebilirsiniz, ancak bağımsız ASP ' lerin ayrılması performansı çok fazla etkileyebilir. Yerel olarak yüksek kullanılabilirlik ve uzaktan olağanüstü durumdan kurtarma sağlamak için coğrafi ikizlemeyi anahtarlamalı diskle birleştirebilirsiniz.

Metro Mirroring

Metro Mirroring, yerel veriyoluna göre daha uzun mesafelerde hızlı yerel zamanuyumlu ikizleme sağlayan bir aygıt düzeyi hizmetidir. Kuyruk yöneticisinin yüksek kullanılabilirlikli olmasını sağlamak ve bağımsız ASP ' nin iki kopyasına sahip olarak kuyruk yöneticisi günlüğünün yüksek kullanılabilirlikli olmasını sağlamak için bunu çok eşgörünümlü bir kuyruk yöneticisiyle birleştirebilirsiniz.

Genel ikizleme

Genel ikizleme, zamanuyumsuz ikizleme sağlayan ve daha uzun mesafelerde yedekleme ve olağanüstü durumdan kurtarma için uygun olan, ancak yüksek kullanılabilirlik için normal bir seçim değildir, çünkü yalnızca zaman tutarlılığını para birimi yerine tutar.

Dikkate almanız gereken önemli karar noktaları şunlardır:

ASP mi, bağımsız ASP mi?

Çok eşgörünümlü kuyruk yöneticilerini kullanmak için bir IBM i HA kümesini çalıştırmamız gerekmez. Bağımsız ASP 'ler kullanıyorsanız ya da bağımsız ASP ' ler gerektiren diğer uygulamalar için kullanılabilirlik gereksinimleriniz varsa, bağımsız ASP ' leri seçebilirsiniz. Bağımsız ASP ' lerin çok eşgörünümlü kuyruk yöneticileriyle birleştirilmesi, kuyruk yöneticisi izlemesinin yerine kuyruk yöneticisi hatasını saptama aracı olarak kullanılmaya değebilir.

Uygun mu?

Kurtarma zamanı hedefi (RTO) nedir? Neredeyse kesintisiz davranışın görünmesini istiyorsanız, hangi çözüm en hızlı kurtarma zamanına sahiptir?

Günlük kullanılabilirliği?

Günlüğü tek bir hata noktası olarak nasıl ortadan kaldırıyorsunuz? RAID 1 aygıtlarını ya da daha iyisini kullanarak bir donanım çözümünü benimseyebilir ya da kopya günlüklerini ya da disk ikizlemesini kullanarak bir yazılım çözümünü birleştirebilir ya da kullanabilirsiniz.

Mesafe?

Etkin ve yedek kuyruk yöneticisi yönetim ortamlarının birbirinden ne kadar uzakta olduğu. Kullanıcılarınız, yaklaşık 250 metreden daha uzun mesafelerde eşzamanlı olarak eşlemenin performans düşüşünü tolere edebilir mi?

Yetenek mi?

Çözümün düzenli olarak sürdürülmesi ve uygulanmasında yer alan yönetim görevlerinin otomatikleştirilmesi için yapılacak işler vardır. Otomasyonu yapmak için gerekli beceriler, ASP 'lere ve bağımsız ASP' lere dayalı çözümler için farklıdır.

IBM i

IBM i üzerinde çok eşgörünümlü bir kuyruk yöneticisinin silinmesi

Çok eşgörünümlü bir kuyruk yöneticisini silmeden önce, uzak günlük kaydını durdurun ve kuyruk yöneticisi eşgörünümlerini kaldırın.

Başlamadan önce

1. Bu örnekte, QM1 kuyruk yöneticisinin iki eşgörünümlü ALPHA ve BETA sunucularında tanımlanmıştır. ALPHA aktif eşgörünüm, BETA ise bekleme. QM1 kuyruk yöneticisiyle ilişkili kuyruk yöneticisi verileri, IBM i sunucusu GAMMA ' da NetServerkullanılarak saklanır. Bkz. ["IBM i üzerinde NetServer ve günlük ikizleme kullanılarak çok eşgörünümlü kuyruk yöneticisinin yaratılması" sayfa 434.](#)
2. ALPHA ve BETA arasında bağlantı kurulmalıdır; böylece, tanımlanan uzak günlükler IBM MQtarafından silinebilir.
3. Sistem komutlarını kullanarak /QNTC dizin ve sunucu dizini dosya paylaşımına erişilebildiğini doğrulayın **EDTF** ya da **WRKLNK**

Bu görev hakkında

Çok eşgörünümlü bir kuyruk yöneticisini **DLTMQM** komutunu kullanarak sunucudan silmeden önce, **RMVMQMINF** komutunu kullanarak diğer sunuculardaki kuyruk yöneticisi yönetim ortamlarını kaldırın.

RMVMQMINF komutunu kullanarak bir kuyruk yöneticisi yönetim ortamını kaldırdığınızda, AMQöneki eklenmiş ve yönetim ortamıyla ilişkilendirilmiş yerel ve uzak günlükler silinir. Sunucu için yerel olan kuyruk yöneticisi yönetim ortamıyla ilgili yapılanış bilgileri de silinir.

Kuyruk yöneticisinin geri kalan yönetim ortamını bulunduran sunucuda **RMVMQMINF** komutunu çalıştırmayın. Bunu yapmak, **DLTMQM** ' nun düzgün çalışmasını önler.

DLTMQM komutunu kullanarak kuyruk yöneticisini silin. Kuyruk yöneticisi verileri ağ paylaşımında kaldırılır. AMQ öneki eklenmiş ve yönetim ortamıyla ilişkilendirilmiş yerel ve uzak günlükler silinir. **DLTMQM** , sunucu için yerel olan kuyruk yöneticisi yönetim ortamıyla ilgili yapılanış bilgilerini de siler.

Örnekte, yalnızca iki kuyruk yöneticisi yönetim ortamı vardır. IBM MQ , bir etkin kuyruk yöneticisi yönetim ortamı ve bir yedek yönetim ortamı olan, çalışmakta olan bir çok eşgörünümlü yapılanış destekler. Çalışan yapılandırılmalarda kullanmak üzere ek kuyruk yöneticisi eşgörünümleri yarattıysanız, geri kalan eşgörünümlü silmeden önce **RMVMQMINF** komutunu kullanarak bunları kaldırın.

Yordam

1. Kuyruk yöneticisi yönetim ortamları arasındaki uzak günlük kaydını etkinlik dışı kılmak için her sunucuda **CHGMQMJRN RMTJRNSTS (*INACTIVE)** komutunu çalıştırın.
 - a) -Alfa 'da.

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('BETA') RMTJRNSTS(*INACTIVE)
```

b) Beta 'da:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('ALPHA') RMTJRNSTS(*INACTIVE)
```

2. Her iki QM1 yönetim ortamını da durdurmak için etkin kuyruk yöneticisi yönetim ortamı olan ALPHA üzerinde **ENDMQM** komutunu çalıştırın.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) INSTANCE(*ALL) ENDCCTJOB(*YES)
```

3. Yönetim ortamına ilişkin kuyruk yöneticisi kaynaklarını ALPHA ve BETA 'dan kaldırmak için ALPHA 'da **RMVMQMINF** komutunu çalıştırın.

```
RMVMQMINF MQMNAME(QM1)
```

RMVMQMINF , QM1 kuyruk yöneticisi yapılış bilgilerini ALPHA ' dan kaldırır. Günlük adının öneki AMQise, QM1 ile ilişkili yerel günlüğü ALPHA ' dan siler. Günlük adının başına AMQ öneki eklendiyse ve bir uzak günlük yaratıldıysa, uzak günlük BETA ' dan da kaldırılır.

4. QM1 ögesini silmek için BETA üzerinde **DLTMQM** komutunu çalıştırın.

```
DLTMQM MQMNAME(QM1)
```

DLTMQM , GAMMA üzerindeki ağ paylaşımında kuyruk yöneticisi verilerini siler. QM1 kuyruk yöneticisi yapılış bilgilerini BETA ' dan kaldırır. Günlük adının öneki AMQise, QM1 ile ilişkili yerel günlüğü BETA ' dan siler. Günlük adının başına AMQ öneki eklendiyse ve bir uzak günlük yaratıldıysa, uzak günlük ALPHA ' dan da kaldırılır.

Sonuçlar

DLTMQM ve **RMVMQMINF** , **CRTMQM** ve **ADDMQJRN** tarafından yaratılan yerel ve uzak günlükleri siler. Bu komutlar günlük nesnelere de siler. Günlükler ve günlük nesnelere, AMQ ile başlayan adlara sahip olma adlandırma kuralını izlemelidir. **DLTMQM** ve **RMVMQMINF** , mqs . ini ' den kuyruk yöneticisi nesnelere, kuyruk yöneticisi verilerini ve kuyruk yöneticisi yapılış bilgilerini kaldırır.

Sonraki adım

Diğer bir yaklaşım, "1" sayfa 449 . adımda günlük kaydını devre dışı bıraktıktan sonra ve kuyruk yöneticisi yönetim ortamlarını sona erdirmeden önce aşağıdaki komutları vermektir. Ya da adlandırma kuralına uymadıysanız, günlükleri ve günlük nesnelere ad temelinde silmeniz gerekir.

1. -Alfa 'da.

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('BETA')
```

2. Beta 'da:

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('ALPHA')
```

Günlükleri sildikten sonra, diğer adımlarla devam edin.

IBM i üzerinde çok eşgörünümlü bir kuyruk yöneticisinin yedeklenmesi

Bu yordam, yerel sunucudaki kuyruk yöneticisi nesnelere ve ağ dosya sunucusundaki kuyruk yöneticisi verilerinin nasıl yedekleneceğini gösterir. Örneği, diğer kuyruk yöneticileri için verileri yedeklemek üzere uyarlayın.

Başlamadan önce

Bu örnekte, QM1 kuyruk yöneticisiyle ilişkili kuyruk yöneticisi verileri, NetServer kullanılarak GAMMA adlı IBM i sunucusunda saklanır. Bkz. “IBM i üzerinde NetServer ve günlük ikizleme kullanılarak çok eşgörünümlü kuyruk yöneticisinin yaratılması” sayfa 434. IBM MQ , ALPHA ve BETA sunucularına kurulur. Kuyruk yöneticisi QM1, ALPHA ve BETA ' da yapılandırılır.

Bu görev hakkında

IBM i , uzak bir dizinden veri saklanmasını desteklemez. Dosya sistemi sunucusunda yerel yedekleme yordamlarını kullanarak uzak bir dosya sisteminde kuyruk yöneticisi verilerini saklayın. Bu görevde, ağ dosya sistemi bir IBM i sunucusu olan GAMMA ' da bulunur. Kuyruk yöneticisi verileri, GAMMA üzerindeki bir saklama dosyasında yedeklenir.

Ağ dosya sistemi Windows ya da Linux üzerindeyse, kuyruk yöneticisi verilerini sıkıştırılmış bir dosyada saklayabilir ve saklayabilirsiniz. Tivoli Storage Manager gibi bir yedekleme sisteminiz varsa, kuyruk yöneticisi verilerini yedeklemek için bunu kullanın.

Yordam

1. QM1 ile ilişkilendirilmiş kuyruk yöneticisi kitaplığı için ALPHA ' da bir saklama dosyası yaratın.

Saklama dosyasını adlamak için kuyruk yöneticisi kitaplık adını kullanın.

```
CRTSAVF FILE(QGPL/QMQM1)
```

2. Kuyruk yöneticisi kitaplığını ALPHA ' daki saklama dosyasına kaydedin.

```
SAVLIB LIB(QMQM1) DEV(*SAVF) SAVF(QGPL/QMQM1)
```

3. GAMMA ' da kuyruk yöneticisi veri dizini için bir saklama dosyası oluşturun.

Saklama dosyasını adlamak için kuyruk yöneticisi adını kullanın.

```
CRTSAVF FILE(QGPL/QMDQM1)
```

4. GAMMA ' daki yerel dizinden kuyruk yöneticisi verilerinin kopyasını kaydedin.

```
SAV DEV('/QSYS.LIB/QGPL.LIB/QMDQM1.FILE') OBJ('/QIBM/Userdata/mqm/qmgrs/QM1')
```

IBM i Çok eşgörünümlü kuyruk yöneticilerini ayarlamak için komutlar

IBM MQ , günlük kopyalamanın konfigürasyonunu tanımlamak, yeni kuyruk yöneticisi eşgörünümleri eklemek ve kuyruk yöneticilerini bağımsız ASP kullanacak şekilde yapılandırmak için komutlar içerir.

Yerel ve uzak günlüklerin yaratılmasını ve yönetilmesini sağlayan günlük komutları,

ADDMQMJRN

Bu komutla, bir kuyruk yöneticisi yönetim ortamı için adlandırılmış yerel ve uzak günlükler yaratabilir ve eşlemenin zamanuyumlu ya da zamanuyumsuz olup olmadığını, zamanuyumlu zamanaşımının ne olduğunu ve uzak günlüğün hemen etkinleştirilip etkinleştirilmeyeceğini yapılandırabilirsiniz.

CHGMQMJRN

Komut, kopya günlüklerini etkileyen zamanaşımı, durum ve teslim parametrelerini değiştirir.

RMVMQMJRN

Adlandırılmış *uzak* günlükleri bir kuyruk yöneticisi yönetim ortamından kaldırır.

WRKMQMJRN

Yerel kuyruk yöneticisi yönetim ortamına ilişkin yerel ve uzak günlüklerin durumunu listeler.

mqs.ini dosyasını deęiřtiren ařaęıdaki komutları kullanarak ek kuyruk yöneticisi eřęörünümlelerini ekleyin ve yönetin.

ADDMQMINF (MQMINF)

Komut, farklı bir IBM i sunucusunda yeni bir kuyruk yöneticisi yönetim ortamı eklemek için DSPMQMINF komutuyla mqs.ini dosyasından açtığınız bilgileri kullanır.

RMVMQMINF

Bir kuyruk yöneticisi eřęörünümünü kaldırın. Varolan bir kuyruk yöneticisinin yönetim ortamını kaldırmak ya da farklı bir sunucudan silinmiř bir kuyruk yöneticisine iliřkin yapılanıř bilgilerini kaldırmak için bu komutu kullanın.

CRTMQM komutu, çok eřęörünümlü bir kuyruk yöneticisinin yapılandırılmasına yardımcı olacak üç parametreye sahiptir.

MQMDIRP (*DFT | *dizin-önek*)

Aęa baęlı depolamada kuyruk yöneticisi verileriyle eřlenen bir baęlama noktası seçmek için bu parametreyi kullanın.

ASP (*SYSTEM|*ASPDEV| *yardımcı bellek havuzu-no*)

Kuyruk yöneticisi günlükünü sisteme ya da temel kullanıcı ASP 'sine yerleřtirmek için *SYSTEMya da *yardımcı bellek havuzu-numarası* belirleyin. Kuyruk yöneticisi günlükünü baęımsız bir ASP ' ye yerleřtirmek için *ASPDEV seçeneęini belirleyin ve **ASPDEV** parametresini kullanarak bir aygıt adı belirleyin.

ASPDEV (*ASP|*aygıt-adi*)

Birincil ya da ikincil baęımsız ASP aygıtının *aygıt-adi* deęerini belirleyin. *ASP seçeneęinin belirlenmesi, **ASP** (*SYSTEM) deęerinin belirlenmesiyle aynı sonucu alır.

IBM i IBM i üzerinde başarıml ve disk hata durumunda yedek sisteme geçiř ile ilgili önemli noktalar

Performansı ve güvenilirlięi artırmak için farklı yardımcı depolama havuzları kullanın.

Uygulamalarınızda çok sayıda kalıcı ileti ya da büyük ileti kullanırsanız, bu iletiyi diske yazmak için harcanan süre, sistemin başarımlında önemli bir faktör haline gelir.

Bu olasılıkla başa çıkmak için yeterli disk etkinleřtirmesine sahip olduęunuzdan emin olun ya da kuyruk yöneticisi günlük nesnelerinin tutulduęu ayrı bir Yardımcı Bellek Havuzu (ASP) düşünün.

CRTMQMASP parametresini kullanarak kuyruk yöneticinizi yaratırken kuyruk yöneticisi kitaplıęınızın ve günlüklerinizin hangi ASP ' de saklanacaęını belirleyebilirsiniz. Varsayılan olarak, kuyruk yöneticisi kitaplıęı ve günlükleri ve IFS verileri sistem ASP 'sinde saklanır.

ASP 'ler, bir ya da daha çok belirli disk birimindeki nesnelerin yalıtılmasına izin verir. Bu, bir disk ortamı hatası nedeniyle veri kaybını da azaltabilir. Çoęu durumda, yalnızca etkilenen ASP ' deki disk birimlerinde saklanan veriler kaybolur.

Hata durumunda yedek sisteme geçiř sağlamak ve disk çekiřmesini azaltmak için, kuyruk yöneticisi kitaplıęını ve günlük verilerini ayrı kullanıcı ASP ' lerinde kök IFS dosya sisteminde saklamanız önerilir.

Daha fazla bilgi için IBM i belgelerinde [Backup and recovery](#) (Yedekleme ve kurtarma) bařlıklı konuya bakın.

IBM i IBM i üzerinde IBM MQ kitaplıklarını saklamak için SAVLIB Olanაının Kullanılması

IBM MQ kitaplıklarını saklamak için SAVLIB LIB(*ALLUSR) kullanamazsınız; bu kitaplıkların adları Q ile başlar.

Yalnızca *SAVF dışında bir saklama aygıtı kullanıyorsanız, tüm kuyruk yöneticisi kitaplıklarını saklamak için SAVLIB LIB(QM*) komutunu kullanabilirsiniz. DEV(*SAVF) için, sisteminizdeki her kuyruk yöneticisi kitaplıęı için bir SAVLIB komutu kullanmanız gerekir.

Bu bölümde, IBM MQ for IBM i ' i nasıl durduracağınızı (zarif bir şekilde sonlandıracağınızı) açıklanır.

IBM MQ for IBM i ' i susturmak için:

1. Herhangi bir nesneye erişemediğinizi doğrulayarak yeni bir etkileşimli IBM MQ for IBM i oturumunda oturum açın.
2. Aşağıdakilere sahip olduğunuzdan emin olun:
 - QMQM kitaplığı için *ALLOBJ yetkisi ya da nesne yönetimi yetkisi
 - ENDSBS komutunu kullanmak için yeterli yetki
3. Tüm kullanıcılara IBM MQ for IBM i ' in durdurulmasını önerin.
4. Daha sonra nasıl devam edeceğiniz, tek bir kuyruk yöneticisini kapatmak (susturmak) isteyip istemediğinize (diğerlerinin var olabileceği) bağlıdır (bkz. [“IBM MQ for IBM i için tek bir kuyruk yöneticisinin kapatılması” sayfa 453](#)) ya da tüm kuyruk yöneticileri (bkz. [“IBM MQ for IBM i için tüm kuyruk yöneticileri kapatılıyor” sayfa 455](#)).
5. Qshell 'de şu komutu girerek mqweb sunucusunu kapatın:

```
/QIBM/ProdData/mqm/bin/endmqweb
```

ENDMQM parametresi ENDCCTJOB (*YES)

ENDMQM parametresi ENDCCTJOB (*YES), IBM MQ for IBM i V6.0 sürümünde ve sonraki sürümlerinde önceki sürümlerle karşılaştırıldığında farklı çalışır.

Önceki sürümlerde, ENDCCTJOB (*YES) belirtimini belirttiğinizde, MQ sizin için uygulamalarınızı zorla sonlandırır.

IBM MQ for IBM i V6.0 ya da sonraki bir yayın düzeyinde ENDCCTJOB (*YES) değerini belirlediğinizde, uygulamalarınız sonlandırılmaz, bunun yerine kuyruk yöneticisiyle bağlantısı kesilir.

ENDCCTJOB (*YES) değerini belirlerseniz ve bir kuyruk yöneticisinin sona erdiğini saptamak için yazılmamış uygulamalarınız varsa, yeni bir MQI çağrısı yayınlandığında, çağrı MQRC_CONNECTION_BROKEN (2009) hatasıyla döndürülür.

ENDCCTJOB (*YES) yöntemine alternatif olarak, ENDCCTJOB (*NO) parametresini kullanın ve WRKMQM seçenek 22 'yi (İşlerle çalış) kullanarak, kuyruk yöneticisinin yeniden başlatılmasını önleyecek uygulama işlerini el ile sona erdirin.

Üç tip kapatma işlemi anlamak için bu bilgileri kullanın.

Aşağıdaki yordamlarda, örnek bir kuyruk yöneticisi adı olan QMgr1 ve örnek bir altsistem adı olan SUBXkullanılır. Gerekirse, bu adları kendi değerinize değiştirin.

Planlanan kapatma

IBM i üzerinde bir kuyruk yöneticisinin planlı olarak kapatılması

1. Kapatmadan önce şunu yürütün:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNTA(*YES)
```

2. Kuyruk yöneticisini kapatmak için aşağıdakileri yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

QMgr1 sona ermezse, kanal ya da uygulamalar meşgul olabilir.

3. QMgr1 ' i hemen kapatmanız gerekiyorsa, aşağıdakileri yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

Planlanmamış kapatma

1. Kuyruk yöneticisini kapatmak için aşağıdakileri yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

QMgr1 sona ermezse, kanal ya da uygulamalar meşgul olabilir.

2. QMgr1 ' i hemen kapatmanız gerekirse, aşağıdakileri yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

Olağandışı koşullar altında kapat

1. Kuyruk yöneticisini kapatmak için aşağıdakileri yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

QMgr1 sona ermezse, 3. adımdan devam edin ve aşağıdakileri yapın:

- QMgr1 kendi altsisteminde ya da
- QMgr1 ile aynı altsistemi paylaşan tüm kuyruk yöneticilerini sona erdirebilirsiniz. Bu tür tüm kuyruk yöneticileri için planlanmamış kapatma yordamını kullanın.

2. Altsistemi paylaşan tüm kuyruk yöneticileri için yordamdaki tüm adımları gerçekleştirdiğinizde (örneklerimizde SUBX) şunları yürütün:

```
ENDSBS SUBX *IMMED
```

Bu komut tamamlanamazsa, planlanmamış kapatma yordamını kullanarak tüm kuyruk yöneticilerini kapatın ve makinenizde bir IPL işlemi gerçekleştirin.

Uyarı: Makinenizde hemen bir IPL gerçekleştirmeye hazır değilseniz, ENDJOB ya da ENDSBSsonucunda sona ermeyen IBM MQ işleri için ENDJOBABN komutunu kullanmayın.

3. Aşağıdaki işlemleri gerçekleştirerek altsistemi başlatın:

```
STRSBS SUBX
```

4. Aşağıdaki komutu yürüterek kuyruk yöneticisini hemen kapatın:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(10)
```

5. Şu komutu yürüterek kuyruk yöneticisini yeniden başlatın:

```
STRMQM MQMNAME(QMgr1)
```

Bu başarısız olursa, ve siz:

- Bir IPL gerçekleştirerek makinenizi yeniden başlatmış veya
- Tek bir kuyruk yöneticisine sahip

IBM MQ paylaşılan belleğini şu komutu yürüterek topla:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

5. adımı yinelemeden önce.

Kuyruk yöneticisinin yeniden başlatılması birkaç saniyeden uzun sürerse, IBM MQ , iş günlüğüne başlatma işleminin ilerleyişini ayrıntılarıyla açıklayan durum iletileri ekler.

Kuyruk yöneticisini yeniden başlatmaya devam ederseniz, IBM desteğine başvurun. Yapacağınız başka bir işlem kuyruk yöneticisine zarar verebilir ve IBM MQ kurtarma işlemini gerçekleştiremez.

IBM i

IBM MQ for IBM i için tüm kuyruk yöneticileri kapatılıyor

Üç tip kapatma işlemini anlamak için bu bilgileri kullanın.

Yordamlar, tek bir kuyruk yöneticisiyle hemen hemen aynıdır, ancak mümkün olan yerlerde kuyruk yöneticisi adı yerine *ALL kullanılır ve tersi durumda, her kuyruk yöneticisi adını kullanarak yinelenen bir komut kullanılır. Yordamlar boyunca, örnek bir kuyruk yöneticisi adı olan QMgr1 ve örnek bir altsistem adı olan SUBXkullanılır. Bunları kendininkiyle değiştir.

Planlanan kapatma

1. Kapanmadan bir saat önce aşağıdakileri yürütün:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNTA(*YES)
```

Sona erdirmek istediğiniz her kuyruk yöneticisi için bunu yineleyin.

2. Kuyruk yöneticisini kapatmak için aşağıdakileri yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

Sona erdirmek istediğiniz her kuyruk yöneticisi için bunu yineleyin; ayrı komutlar koşut olarak çalışabilir.

Herhangi bir kuyruk yöneticisi makul bir süre (örneğin, 10 dakika) içinde sona ermezse, 3. adıma geçin.

3. Tüm kuyruk yöneticilerini hemen kapatmak için aşağıdakileri yürütün:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

Planlanmamış kapatma

1. Bir kuyruk yöneticisini kapatmak için aşağıdakileri yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Sona erdirmek istediğiniz her kuyruk yöneticisi için bunu yineleyin; ayrı komutlar koşut olarak çalışabilir.

Kuyruk yöneticileri sona ermezse, kanal ya da uygulamalar meşgul olabilir.

2. Kuyruk yöneticilerini hemen kapatmanız gerekirse, aşağıdakileri yürütün:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

Olağandışı koşullar altında kapat

1. Kuyruk yöneticilerini sona erdirmek için şunları yürütün:

```
ENDMQM MQMNAME(QMg1) OPTION(*IMMED)
```

Sona erdirmek istediğiniz her kuyruk yöneticisi için bunu yineleyin; ayrı komutlar koşturularak çalışabilir.

2. Aşağıdaki işlemleri gerçekleştirerek altsistemleri sona erdirin (örneklerimizde SUBX):

```
ENDSBS SUBX *IMMED
```

Sona erdirmek istediğiniz her altsistem için bu işlemi yineleyin; ayrı komutlar koşturularak çalışabilir.

Bu komut tamamlanamazsa, sisteminizde bir IPL işlemi gerçekleştirin.

Uyarı: Sisteminizde bir IPL gerçekleştirmeye hemen hazır değilseniz, ENDJOB ya da ENDSBSsonucunda sona ermeyen işler için ENDJOBABN komutunu kullanmayın.

3. Aşağıdaki işlemleri yürüterek altsistemleri başlatın:

```
STRSBS SUBX
```

Başlatmak istediğiniz her altsistem için bu işlemi yineleyin.

4. Aşağıdaki komutu yürüterek kuyruk yöneticilerini hemen kapatın:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

5. Şu komutu yürüterek kuyruk yöneticilerini yeniden başlatın:

```
STRMQM MQMNAME(QMg1)
```

Başlatmak istediğiniz her kuyruk yöneticisi için bunu yineleyin.

Herhangi bir kuyruk yöneticisi yeniden başlatması birkaç saniyeden uzun sürerse IBM MQ , başlatma işleminin ilerleyişini aralıklı olarak ayrıntılandıran durum iletileri görüntüler.

Herhangi bir kuyruk yöneticisini yeniden başlatmaya devam ederseniz, IBM destek ile iletişim kurun. Yapacağınız başka bir işlem kuyruk yöneticilerine zarar verebilir ve MQSeries ya da IBM MQ kurtarma işlemini gerçekleştiremez.

z/OS

YönetmeIBM MQ for z/OS

IBM MQ for z/OS , MQSC ve PCF komutları, ürünle birlikte sağlanan yardımcı programlar ve programlar kümesi ve yetkili uygulamalar tarafından denetlenebilir ve yönetilebilir.

IBM MQ for z/OS ' in nasıl yönetileceğini ve üstlenmeniz gereken farklı yönetim görevlerine ilişkin ayrıntılar için aşağıdaki bağlantılara bakın.

IBM MQ for z/OS ' i Linux kabuğunda çalışan IBM MQ Explorer kullanarak da yönetebilirsiniz. Daha fazla bilgi için bkz [“IBM MQ Explorer kullanarak yönetim” sayfa 138](#).

İlgili kavramlar

[IBM MQ for z/OS kavramlar](#)

İlgili görevler

[“YönetmeIBM MQ” sayfa 7](#)

IBM MQ kuyruk yöneticilerinizi ve ilişkili kaynaklarınızı denetlemek için, bu kaynakları etkinleştirmek ve yönetmek için kullanabileceğiniz görevler kümesinden tercih ettiğiniz yöntemi seçin.

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

z/OS üzerinde kuyruğu ni bkz olarak belirtir)

z/OS

z/OS üzerinde kuyruk yöneticisi komutları verilmesi

IBM MQ işletim ortamının çoğunu denetim komutlarını kullanarak denetleyebilirsiniz. IBM MQ for z/OS konsolundan, kullanıma hazırlama giriş veri kümelerinden, toplu iş yardımcı programından ya da yetkili uygulamalardan MQSC ve PCF komutları verebilirsiniz.

Bu görev hakkında

Kuyruk yöneticilerini doğrudan denetlemek için toplu kipte ya da etkileşimli kipte MQSC komutları kullanılır. Kuyruk yöneticilerini denetleyen uygulamalar yaratmanıza yardımcı olması için PCF komutlarını kullanırsınız. MQSC komutları, insan tarafından okunabilir metin biçimindedir; PCF komutları ise, uygulamaların metin dizgilerini ayrıştırmasına gerek kalmadan istek oluşturmasına ve yanıtları okumasına izin verir. MQSC komutları gibi, uygulamalar da PCF komutlarını komut giriş kuyruğuna ileti olarak göndererek verir.

Aşağıdaki konularda, IBM MQ for z/OS konsolundan, kullanıma hazırlama giriş veri kümelerinden, toplu iş yardımcı programından ya da yetkili uygulamalardan kuyruk yöneticisi komutlarını nasıl verdiğiniz açıklanmaktadır.

Tüm komutlar tüm kaynaklardan yayınlanmaz. Bkz. [“IBM MQ for z/OS üzerinde MQSC ve PCF komutlarını yayınlatabileceğiniz kaynaklar” sayfa 457.](#)

İlgili görevler

[z/OS üzerinde TSO ortamı için örnek uygulamaların hazırlanması](#)

İlgili bilgiler

[IBM MQ 'nin MQSC komutlarını kullanarak denetlenmesi](#)

z/OS

IBM MQ for z/OS üzerinde MQSC ve PCF komutlarını yayınlatabileceğiniz kaynaklar

IBM MQ for z/OS konsolundan, kullanıma hazırlama giriş veri kümelerinden, toplu iş yardımcı programından ya da yetkili uygulamalardan MQSC ve PCF komutları verebilirsiniz. Tüm komutlar tüm bu kaynaklardan verilemez.

Hangi MQSC ve PCF komutları her bir IBM MQ nesnesini denetleyebilir

Çizelge 1 ("Command summary for IBM MQ for z/OS"), IBM MQ for z/OS üzerinde her bir IBM MQ nesnesini değiştirmek, tanımlamak, silmek ve görüntülemek için kullanılacak MQSC ve PCF komutlarını eşler. Ayrıca bkz. [MQSC komutları başvurusu](#) ve [“IBM MQ Programlanabilir Komut Biçimlerini Kullanma” sayfa 26.](#)

Komutların verilebildiği kaynakların listesi

Uygun bir yetkili kullanıcıysanız, aşağıdaki kaynaklardan IBM MQ komutlarını verebilirsiniz:

- z/OS konsolu ya da eşdeğeri (SDSF/TSO gibi).

Ayrıca bkz. [“z/OS üzerindeki işlemleri ve denetim panolarını kullanma” sayfa 471.](#)

Not: z/OS konsolunu kullanırken, bir komutun başlangıcını /cpf eklemeniz gerekir; burada cpf , kuyruk yöneticisi altsisteminin komut önekidir.

- Kullanıma hazırlama giriş veri kümeleri CSQINP1, CSQINP2, CSQINPT ve CSQINPX.

Bkz. [“IBM MQ for z/OS için başlatma komutları” sayfa 468.](#)

- z/OS ana alma komutu yordamı, MGCRE (SVC 34).

- Sıralı veri kümesindeki komutların listesini işleyen CSQUTIL gibi IBM MQ toplu iş yardımcı programları.

Bkz. [“IBM MQ for z/OS yardımcı programlarının kullanılması” sayfa 479.](#)

- SYSTEM.COMMAND.INPUT kuyruğu.

Uygulama aşağıdakilerden herhangi biri olabilir:

- Toplu iş bölgesi programı
- CICS uygulaması
- IMS uygulaması
- Bir TSO uygulaması
- Başka bir IBM MQ sistemindeki bir uygulama programı ya da yardımcı program

Bkz. [“IBM MQ for z/OS yönetimi için programlar yazılıyor” sayfa 488 ve z/OS üzerinde TSO ortamı için örnek uygulamaların hazırlanması.](#)

Tüm komutlar tüm kaynaklardan verilemez

Komutlar, bu çıkarıtım -ilgili çıkarıtım her kara göre sınıflandırılır.? de? de.

1

CSQINP1

2

CSQINP2

C

z/OS konsolu

R

CSQUTIL, CSQINPT, CSQINPX ya da yetkili uygulamalar aracılığıyla komut sunucusu ve komut kuyruğu.

MQSC komutları başvurusundakikomut açıklamalarında, bu kaynaklar her komut tanımında 1, 2, C ve R karakterlerinin kullanımıyla tanıtılır. [Çizelge 2 \(" IBM MQ for z/OSiçin komut özeti"\)](#) içinde MQSC komutları ve bunların yayınlanabileceği kaynaklar özetlenir.

İlgili görevler

[z/OS üzerinde TSO ortamı için örnek uygulamaların hazırlanması](#)

İlgili bilgiler

[IBM MQ ' nin MQSC komutlarını kullanarak denetlenmesi](#)

► z/OS IBM MQ for z/OS için komut özeti

Ana MQSC ve PCF komutlarının ve IBM MQ for z/OSüzerinde MQSC komutlarını çalıştırabileceğiniz kaynakların özeti.

[Çizelge 25 sayfa 458](#) , her IBM MQ nesnesini değiştirmek, tanımlamak, silmek ve görüntülemek için IBM MQ for z/OS üzerinde hangi MQSC ve PCF komutlarının kullanılabileceğini eşler.

Çizelge 25. Nesne tipine göre ana MQSC ve PCF komutlarının özeti				
MQSC komutu	Çeviri	Tanımla	GÖRÜNTÜLE	SİL
PCF komutu	Değiştir	Oluştur/Kopyala	Sor	Sil
AUTHINFO	X	X	X	X
CFSTATUS			X	
CFSTRUCT	X	X	X	X
Kanal	X	X	X	X
DURUM			X	
ADı LISTESİ	X	X	X	X

Çizelge 25. Nesne tipine göre ana MQSC ve PCF komutlarının özeti (devamı var)

MQSC komutu	Çeviri	Tanımla	GÖRÜNTÜLE	SİL
Süreç	X	X	X	X
QDIĞER AD	M	M	M	M
QCLUSTER			M	
QLOCAL	M	M	M	M
QMGR	X		X	
QMODEL	M	M	M	M
QUZAK	M	M	M	M
kuyruk	P	P	X	P
QSTATUS (DURUM)			X	
STGCLASS	X	X	X	X

Çizelge simgelerine ilişkin tuş:

- M = Yalnızca MQSC
- Yalnızca P = PCF
- X = her ikisi

Diğer IBM MQ kaynaklarını yönetmenizi ve Çizelge 25 sayfa 458’inde özetlenenlere ek olarak diğer işlemleri gerçekleştirmenizi sağlayan birçok MQSC ve PCF komutu vardır.

Çizelge 26 sayfa 459 , her MQSC komutunu ve her komutun verilebildiği yeri gösterir.

- CSQINP1 kullanıma hazırlama girişi veri kümesi
- CSQINP2 kullanıma hazırlama girişi veri kümesi
- z/OS konsolu (ya da eşdeğeri)
- SYSTEM.COMMAND.INPUT kuyruğu ve komut sunucusu (uygulamalar, CSQUTIL ya da CSQINPX kullanıma hazırlama giriş veri kümesinden)

Çizelge 26. MQSC komutlarının çalıştırılacağı kaynaklar

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
AUTHINFO DEĞİŞTİR		X	X	X
BUFFPOOL DEĞİŞTİR		X	X	X
ALTER CFSTRUCT (CFSTRUCT)		X	X	X
KANAL DEĞİŞTİR		X	X	X
ADı LISTESİ		X	X	X
İŞLEMI DEĞİ		X	X	X
ALTER PSID (PSID)			X	X
QALIAS DEĞİŞTİR		X	X	X
ALTER QLOCAL (YEREL)		X	X	X

Çizelge 26. MQSC komutlarının çalıştırılacağı kaynaklar (devamı var)

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
ALTER QMGR		X	X	X
QMODEL DEĞİŞTİR		X	X	X
QUZAK DEĞİŞTİR		X	X	X
GüVENLİK DEĞ	X	X	X	X
ALTER SMDS (SMDS)		X	X	X
STGClass DEĞİŞTİR		X	X	X
ALTER SUB			X	X
KONUYU DEĞİŞTİR		X	X	X
 ALTER TRACE	X	X	X	X
Günlüğü ARŞIV	X	X	X	X
YEDEK CFSTRUCT			X	X
QLOCAL TEMİZLE		X	X	X
TOPICSTR TEMİZLE			X	X
AUTHINFO TANIMLAYIN		X	X	X
ARABELLEK HAVUZU TANIMLAYIN	X			
CFSTRUCT 'U TANIMLAYIN		X	X	X
KANAL TANIMLAYIN		X	X	X
Günlüğü TANIM			X	X
MAXSMSGS TANIMLAYIN		X	X	X
AD LISTESİ TANIMLAYIN		X	X	X
Süreç TANIMLE		X	X	X
PSID TANIMLAYIN	X		X	X
QALIAS TANIMLAYIN		X	X	X
QLOCAL TANIMLAYIN		X	X	X
QXX_ENCODE_CASE_ONE modeli TANIMLAYIN		X	X	X
QUZAK TANIMLAYIN		X	X	X
STGClass TANIMLAYIN		X	X	X
ALT öğeyi TAN			X	X
KONUYU TANIMLAYIN		X	X	X
AUTHINFO ÖĞESİNİ SİL		X	X	X
ARABELLEK HAVUZUNU SİL		X	X	X

Çizelge 26. MQSC komutlarının çalıştırılacağı kaynaklar (devamı var)

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
CFSTRUCT 'U SIL		X	X	X
BAĞLANTI SIL			X	X
ADı SIL		X	X	X
Süreci Sil		X	X	X
PSID ' YI SIL			X	X
QDIĞER ADı SIL		X	X	X
QLOCAL ÖĞESINI SIL		X	X	X
QMODEL SIL		X	X	X
QREMOTE ÖĞESINI SIL		X	X	X
STGCLASS DEĞERINI SIL		X	X	X
ALT ÖĞEYI SIL			X	X
KONUYU SIL		X	X	X
ARŞIV GÖRÜNTÜ	X	X	X	X
AUTHINFO BİLGİLERİNİ GÖRÜNTÜLE		X	X	X
CFSTATUS DURUMUNU GÖRÜNTÜLE			X	X
CFSTRUCT 'U GÖRÜNTÜLE		X	X	X
KANAL GÖRÜNTÜLE		X	X	X
CHINIT GÖRÜNTÜLE			X	X
CHLAUTH 'U GÖRÜNTÜLE		X	X	X
CHSTATUS DURUMUNU GÖRÜNTÜLE			X	X
CLUSQMGR ' YI GÖRÜNTÜLE			X	X
CMDSERV 'I GÖRÜNTÜLE	X	X	X	X
KONN GÖRÜNTÜLE		X	X	X
GRUBU GÖRÜNTÜLE		X	X	X
Günlüğünü	X	X	X	X
MAXSMSGS DEĞERİNİ GÖRÜNTÜLE		X	X	X
ADı LİSTESİ GÖRÜNTÜLE		X	X	X
SÜRECI GÖRÜNTÜ		X	X	X
PUBSUB 'I GÖRÜNTÜLE		X	X	X
QDIĞER ADINI GÖ		X	X	X

Çizelge 26. MQSC komutlarının çalıştırılacağı kaynaklar (devamı var)

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
QKÜMESİNİ GÖRÜNTÜLE		X	X	X
QLOCAL GÖRÜNTÜLE		X	X	X
QMGR ' YI GÖRÜNTÜLE		X	X	X
QMODEL MODELİNİ GÖRÜNTÜLE		X	X	X
QUZAK GÖRÜNTÜLE		X	X	X
QSTATUS DURUMUNU GÖRÜNTÜLE		X	X	X
KUYRUK GÖRÜNTÜLE		X	X	X
SBSTATUS DURUMUNU GÖRÜNTÜLE			X	X
GÜVENLİK GÖRÜNTÜLE			X	X
SMDS GÖRÜNTÜLE		X	X	X
SMDSCONN 'YI GÖRÜNTÜLE		X	X	X
STGClass GÖRÜNTÜLE		X	X	X
ALT öğEYI Gö			X	X
SİSTEMİ GÖRÜNTÜLE	X	X	X	X
TCLUSTER 'I GÖRÜNTÜLE		X	X	X
İş PARÇACIĞIN		X	X	X
KONU GÖRÜNTÜLE		X	X	X
TPSTATÜYÜ GÖRÜNTÜLE		X	X	X
İZLEMEYİ GÖRÜNTÜLE	X	X	X	X
EKRAN KULLANIMI		X	X	X
QLOCAL DEĞERİNİ TAŞIR		X	X	X
PING KANALI			X	X
BDS LERİ KURTAR			X	X
CFSTRUCT 'U KURTARIN			X	X
KÜMEYİ YENİLE			X	X
QMGR ' YI YENİLE		X	X	X
Güvenliği yenileme			X	X
CFSTRUCT 'U İLK DURUMUNA GETİR			X	X
KANALI İLK DURUMUNA GETİR			X	X
KÜMEYİ SIFIR			X	X
QMGR ' YI SIFIR		X	X	X

Çizelge 26. MQSC komutlarının çalıştırılacağı kaynaklar (devamı var)

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
QSTATS DEĞİŞTİRMESİ		X	X	X
SMDS ' I SIFIRLA			X	X
TPUANı SIFIR			X	X
KANAL ÇÖZÜMLE			X	X
ŞÜPHELI ÇÖZÜMLE		X	X	X
QMGR ' YI SÜRDÜR			X	X
RVERIFY GÜVENLİĞİ		X	X	X
ARŞİVI AYARLA	X	X	X	X
CHLAUTH AYARLA		X	X	X
GÜNLÜĞÜ AYARLA	X	X	X	X
SİSTEMİ AYARLA	X	X	X	X
BAŞLANGIÇ KANALI			X	X
CHINIT BAŞLAYIN		X	X	X
CMDSERV ' YI BAŞLAT	X	X	X	
DİNLEYİCİYİ BAŞLAYIN			X	X
QMGR ' YI BAŞLAT			X	
SMDSCONN ' U BAŞLAT		X	X	X
İZLEMİYİ BAŞLAT	X	X	X	X
BAĞLANTI DURDUR			X	X
CHINIT ' I BIRAKMAK			X	X
DURDUR CMDSERV	X	X	X	
DİNLEYİCİYİ DURDUR			X	X
QMGR ' YI DURDUR			X	X
SMDSCONN ' U DURDUR		X	X	X
İZLEMİYİ DURDUR	X	X	X	X
QMGR ' YI ASKIYA AL			X	X

MQSC komutlarında, her komut tanımı bu komutun çalıştırılabileceği kaynakları tanımlar.

z/OS üzerinde bir kuyruk yöneticisini başlatmak ve durdurmak için MQSC ' nin kullanılması

IBM MQ for z/OS üzerinde denetim komutlarının kullanılmasına giriş: IBM MQ kurulduktan sonra, bir kuyruk yöneticisini başlatmak ve durdurmak için MQSC komutlarını kullanın.

Başlamadan önce

IBM MQ kurulduktan sonra, bu, resmi bir z/OS altsistemi olarak tanımlanır. Bu ileti, z/OS' in ilk program yüklemesi (IPL) sırasında görüntülenir:

```
CSQ3110I +CSQ1 CSQ3UR00 - SUBSYSTEM ssnm INITIALIZATION COMPLETE
```

Burada *ssnm* , IBM MQ altsistem adıdır.

Bundan sonra, *sistem denetim komutları verme yetkisi olan z/OS konsoldan* bu altsisteme ilişkin kuyruk yöneticisini başlatabilirsiniz; Yani, bir z/OS SYS komut grubu. START komutunu yetkili konsoldan çalıştırmanız gerekir; komutu JES ya da TSO aracılığıyla yayınlayamazsınız.

Kuyruk paylaşım gruplarını kullanıyorsanız, kuyruk yöneticisini başlatmadan önce önce RRS 'yi ve sonra Db2' i başlatmanız gerekir.

Bu görev hakkında

Bir kuyruk yöneticisi olağan koşullarda durduğunda, son işlemi bir sonlandırma denetim noktası almaktır. Bu denetim noktası ve günlükler, kuyruk yöneticisine yeniden başlatması gereken bilgileri verir.

Aşağıdaki adımlar START ve STOP komutlarıyla ilgili bilgileri içerir ve olağandışı bir sonlandırma gerçekleştikten sonra başlatma hakkında kısa bir genel bakış içerir.

Yordam

1. Kuyruk yöneticisi başlatılması

Bir kuyruk yöneticisini START QMGR komutunu vererek başlatabilirsiniz. Ancak, gereken yetkiniz yoksa, START komutunu başarıyla kullanamazsınız. IBM MQ güvenliğiyle ilgili bilgi için bkz. [z/OS üzerinde güvenliğin ayarlanması](#) . Aşağıdaki kod START komutunun örneklerini göstermektedir. Bir MQSC komutuna bir komut öneki dizisi (CPF) ile önek eklemelisiniz.

```
+CSQ1 START QMGR
+CSQ1 START QMGR PARM(NEWLOG)
```

START QMGR komutunun sözdizimiyle ilgili bilgi için [START QMGR](#) başlıklı konuya bakın.

Kuyruk yöneticisini toplu iş olarak çalıştıramaz ya da z/OS komutunu START kullanarak başlatamazsınız. Bu yöntemler, IBM MQ için olağandışı biten bir adres alanı başlatabilir. CSQUTIL yardımcı programından ya da benzer bir kullanıcı uygulamasından kuyruk yöneticisi de başlatamazsınız.

Ancak, z/OS MGCRC (SVC 34) hizmetine START QMGR komutunu geçirerek, APF yetkili programdan bir kuyruk yöneticisi başlatabilirsiniz.

Kuyruk paylaşım gruplarını kullanıyorsanız, kuyruk yöneticisini başlattığınızda ilişkili Db2 sistemleri ve RRS etkin olmalıdır.

Başlatma seçenekleri

Bir kuyruk yöneticisini başlattığınızda, bir sistem parametresi modülü yüklenir. Sistem değiştirgesi modülünün adını aşağıdaki iki yoldan biriyle belirtebilirsiniz:

- Örneğin, /cpf START QMGR komutunun PARM değiştirgesiyle

```
/cpf START QMGR PARM(CSQ1ZPRM)
```

- Başlatma yordamındaki bir parametreyle, örneğin, JCL EXEC deyimini şu şekilde kodlayın:

```
//MQM EXEC PGM=CSQYASCP,PARM='ZPARAM(CSQ1ZPRM)'
```


Bir sistem parametresi modülü, kuyruk yöneticisi uyarlandığında belirtilen bilgileri sağlar.

IBM MQ 9.1.0'den, kuyruk yöneticisi kullanımının kaydedileceği ürünü belirtmek için **QMGRPROD** seçeneğini ve kullanılıyorsa AMS' nin eşdeğerini belirtmek için **AMSPROD** seçeneğini kullanabilirsiniz. İzin verilen değerlerin ayrıntıları için MQSC START QMGR komutuna bakın.

Örnek bir JCL EXEC deyimi:

```
//MQM EXEC PGM=CSQYASCP,PARM='QMGRPROD(MQ)'
```

Ölçülen kullanım ve ürün kaydıyla ilgili daha fazla bilgi için bkz. [z/OS MVS Product Management](#).

JCL yordamındaki bir ya da daha çok değiştirgeyi kuyruk yöneticisine değiştirmek için ENVPARAM seçeneğini de kullanabilirsiniz.

Örneğin, kuyruk yöneticisi başlatma yordamınızı, **DDname** CSQINP2 değişkenini olacak şekilde güncelleyebilirsiniz. Bu, başlatma yordamını değiştirmeden CSQINP2 **DDname** ' yi değiştirebileceğiniz anlamına gelir. Bu, değişiklikleri uygulamak, işlemler için geri alma ve kuyruk yöneticisi işlemleri sağlamak için yararlıdır.

CSQ1 kuyruk yöneticisine ilişkin başlatma yordamınızın şöyle olduğunu varsayın:

```
//CSQ1MSTR PROC INP2=NORM
//MQMESA EXEC PGM=CSQYASCP
//STEPLIB DD DISP=SHR,DSN=th1qua1.SCSQANLE
// DD DISP=SHR,DSN=th1qua1.SCSQAUTH
// DD DISP=SHR,DSN=db2qua1.SDSNLOAD
//BDS1 DD DISP=SHR,DSN=myqua1.BSDS01
//BDS2 DD DISP=SHR,DSN=myqua1.BSDS02
//CSQP0000 DD DISP=SHR,DSN=myqua1.PSID00
//CSQP0001 DD DISP=SHR,DSN=myqua1.PSID01
//CSQP0002 DD DISP=SHR,DSN=myqua1.PSID02
//CSQP0003 DD DISP=SHR,DSN=myqua1.PSID03
//CSQINP1 DD DISP=SHR,DSN=myqua1.CSQINP(CSQ1INP1)
//CSQINP2 DD DISP=SHR,DSN=myqua1.CSQINP(CSQ1&INP2.)
//CSQOUT1 DD SYSOUT=*
//CSQOUT2 DD SYSOUT=*
```

Kuyruk yöneticinizi aşağıdaki komutla başlatabilirsiniz:

```
+CSQ1 START QMGR
```

Bu durumda, kullanılan CSQINP2 , CSQ1NORMadlı bir üyedir.

Ancak, yeni bir program takımı yarattığınızda, kuyruk yöneticisi CSQ1' i bir sonraki başlatışınızda, CSQINP2 tanımlamaları CSQ1NEWüyesinden alınacağını varsayOK!.. ve bu Sonra de, Bunu yapmak için kuyruk yöneticisini şu komutla başlatabilirsiniz:

```
+CSQ1 START QMGR ENVPARAM('INP2=NEW')
```

ve CSQ1NEW , CSQ1NORMyerine kullanılır. Not: z/OS , sembolik parametrelere ilişkin KEYWORD=va1ue belirtilmelerini (INP2=NEWiçinde olduğu gibi) 255 karakterle sınırlar.

Olağandışı sonlandırmadan sonra başlatma

IBM MQ , yeniden başlatmanın olağan bir sona erdirme mi, yoksa olağandışı bir sonlandırma mı izlediğine otomatik olarak karar verir.

Olağandışı bittikten sonra bir kuyruk yöneticisinin başlatılması, STOP QMGR komutu verildikten sonra başlatılmasından farklıdır. STOP QMGR ' dan sonra sistem düzgün bir şekilde işini bitirir ve durmadan önce bir sonlandırma denetim noktası alır. Kuyruk yöneticisini yeniden başlattığınızda, sistem kapatma sırasında sistem durumunu saptamak için sistem denetim noktası ve kurtarma günlüğündeki bilgileri kullanır.

Ancak, kuyruk yöneticisi olağandışı sona ererse, işi bitirmeden ya da sonlandırma denetim noktası almadan sona erer. Bir olağandışı sonlandıktan sonra kuyruk yöneticisini yeniden başlattığınızda,

günlükteki bilgileri kullanarak sonlandırma sırasında durumu bilgisini yeniler ve çeşitli görevlerin durumunu size bildirir. Olağan durumda, yeniden başlatma işlemi tüm tutarsız durumları çözer. Ancak bazı durumlarda, tutarsızlıkları çözmek için belirli adımlar atmanız gerekir.

Başlatma işlemi için kullanıcı iletileri

Bir kuyruk yöneticisini başarıyla başlattığınızda, kuyruk yöneticisi bir dizi başlatma iletisi üretir.

2. Bir kuyruk yöneticisini durdurun.

Bir kuyruk yöneticisini durdurmadan önce, IBM MQ ile ilgili tüm WTOR (write için ya) gerektiren şebek'de görünür; örneğin, günlük istekleri alınmalı. Aşağıdaki komutların her biri, çalışmakta olan bir kuyruk yöneticisini sonlandırır.

```
+CSQ1 STOP QMGR
+CSQ1 STOP QMGR MODE(QUIESCE)
+CSQ1 STOP QMGR MODE(FORCE)
+CSQ1 STOP QMGR MODE(RESTART)
```

STOP QMGR komutu varsayılan olarak STOP QMGR MODE (quiesce) değerini alır.

QUIESCE kipinde IBM MQ , yeni bağlantı iş parçacıklarının yaratılmasına izin vermez, ancak var olan iş parçacıklarının devam etmesine izin verir; yalnızca tüm iş parçacıkları sona erdiğinde sona erer. Kuyruk yöneticisinin susturulması durumunda, uygulamalar bilgilendirilmek isteyebilir. Bu nedenle, mümkün olan yerlerde QUIESCE kipini kullanarak, bildirim isteğinde bulunan uygulamaların bağlantıyı kesme fırsatına 'e 'ı 'e 'ı sahip olmanızı sağlayın. Ayrıntılar için [Sonlandırma sırasında neler olur gerekir eder bunlar bunlar bunlar görmek için](#) konusuna bakın.

Kuyruk yöneticisi, STOP QMGR MODE (QUIESCE) komutuna yanıt olarak makul bir süre içinde sonlandırmazsa, bağlantı iş parçacıklarının olup olmadığını saptamak için DISPLAY CONN komutunu kullanın ve ilişkili uygulamaları sonlandırmak için gereken adımları gerçekleştirin. İş parçacığı yoksa, STOP QMGR MODE (FORCE) komutunu verin.

MVS Automatic Restart Manager (ARM) 'den STOP QMGR MODE (QUIESCE) ve STOP QMGR MODE (FORCE) komutları IBM MQ , ARM' nin kuyruk yöneticisini otomatik olarak yeniden başlatmasını önleyerek? ya (ma { KM }) STOP QMGR MODE (RESTART) komutu, IBM MQ 'un ARM' den kaydını kaldırma unda ... STOP QMGR MODE (FORCE) komutuyla aynı şekilde çalışır. Bu, kuyruk yöneticisinin hemen otomatik yeniden başlatma için uygun olduğu anlamına gelir.

IBM MQ altsistemi ARM ' ye kayıtlı değilse, STOP QMGR MODE (RESTART) komutu reddedilir ve z/OS konsoluna aşağıdaki ileti gönderilir:

```
CSQY205I ARM element arm-element is not registered
```

Bu ileti yayınlanmazsa, kuyruk yöneticisi otomatik olarak yeniden başlatılır. ARM hakkında daha fazla bilgi için bkz. [“z/OS Automatic Restart Manager \(ARM\) olanağının kullanılması”](#) sayfa 548.

Kuyruk yöneticisi adres alanını yalnızca STOP QMGR MODE (FORCE) kuyruk yöneticisini sonlandırmazsa iptal edin.

Bir kuyruk yöneticisi adres alanı iptal edilerek ya da STOP QMGR MODE (force) komutu kullanılarak durdurulursa, tutarlılık bağlı CICS ya da IMS sistemleriyle korunur. Kaynakların yeniden eşzamanlanması, bir kuyruk yöneticisi yeniden başlatıldığında ve CICS ya da IMS sistemiyle bağlantı kurulduğunda tamamlanır.

Not: Kuyruk yöneticinizi durdurduğunuzda, IEF352I iletisini bulabilirsiniz. z/OS , adres alanını kullanılamaz olarak işaretlemese bütünlük açısına neden olacağını saptarsa bu iletiyi yayınlar. Bu iletiyi yoksayabilirsiniz.

İletileri durdur

STOP QMGR komutunu verdikten sonra, CSQY009I ve CSQY002I iletilerini alın; örneğin:

```
CSQY009I +CSQ1 ' STOP QMGR' COMMAND ACCEPTED FROM  
USER(userid), STOP MODE(FORCE)  
CSQY002I +CSQ1 QUEUE MANAGER STOPPING
```

Burada `userid` , `STOP QMGR` komutunu veren kullanıcı kimliğidir ve `MODE` parametresi komutta belirtilene bağlıdır.

`STOP` komutu başarıyla tamamlandığında, z/OS konsolunda aşağıdaki iletiler görüntülenir:

```
CSQ9022I +CSQ1 CSQYASCP ' STOP QMGR' NORMAL COMPLETION  
CSQ3104I +CSQ1 CSQ3EC0X - TERMINATION COMPLETE
```

ARM kullanıyorsanız ve `MODE (RESTART)` belirlemediyseniz, aşağıdaki ileti de görüntülenir:

```
CSQY204I +CSQ1 ARM DEREGISTER for element arm-element type  
arm-element-type successful
```

Aşağıdaki ileti görüntüleninceye kadar kuyruk yöneticisini yeniden başlatamazsınız:

```
CSQ3100I +CSQ1 CSQ3EC0X - SUBSYSTEM ssnm READY FOR START COMMAND
```

z/OS konsolundan ya da eşdeğerinden komut verilmesi

IBM MQ MQSC ve PCF komutlarını bir z/OS konsolundan ya da eşdeğeri konsolundan yayınlatabilirsiniz. IBM MQ komutlarını, SDSF gibi z/OS komutlarını yayınlatabileceğiniz herhangi bir yerden ya da MGCRC makrosunu kullanarak bir program aracılığıyla da yayınlatabilirsiniz. z/OS konsolunu kullanırken, bir komutun başlangıcını /cpf eklersiniz.

Başlamadan önce

z/OS konsolu tarafından tüm komutlar yayınlanmaz. Komut tanımı konuları (MQSC komutlarının alt öğeleri) içinde, konsol tarafından verilebilecek her komut 'C' karakteriyle tanımlıdır. [Çizelge 2 \(" IBM MQ for z/OS için komut özeti"\)](#) içinde MQSC komutları ve bunların yayınlanabileceği kaynaklar özetlenir.

Bir IMS uçbiriminden IMS/SSR komut biçimini kullanarak IBM MQ komutları yayınlanamazsınız. Bu işlev IMS bağdaştırıcısı tarafından desteklenmez.

SDSF tarafından sağlanan giriş alanı, özellikle kanallara ilişkin komutlar olmak üzere bazı komutlar için yeterli uzun olmayabilir.

Konsola yazılan bir komutun sonucu olarak görüntülenebilecek veri miktarı üst sınırı 32 KB 'dir.

Bu görev hakkında

Uygun bir yetkili kullanıcıysanız, z/OS konsolundan ya da eşdeğeri (SDSF/TSO gibi) IBM MQ komutlarını verebilirsiniz.

z/OS konsolunu kullanırken, /cpf komutunu bir komutun başlangıca eklemeniz gerekir; burada cpf , kuyruk yöneticisi altsisteminin komut önekidir.

Aşağıdaki adımlar, PCF adlarını değil, MQSC komut adlarını kullanan komutlara ve özniteliklere gönderme yaparlar.

Yordam

- Komut öneki dizgilerini kullan

Her IBM MQ komutunun başına bir komut öneki dizgisi (CPF) eklenmelidir.

z/OSaltında birden çok IBM MQ altsistemi çalışabildiğinden, CPF komutu hangi IBM MQ altsisteminin işlediğini belirtmek için kullanılır.

Örneğin, CSQ1adlı altsisteme ilişkin kuyruk yöneticisini başlatmak için, burada CPF ' +CSQ1 ', işletmen konsolunda aşağıdaki komutu verin:

```
+CSQ1 START QMGR
```

Bu CPF, Komut öneki dizilimlerini tanımlama (CPF) başlıklı konuda açıklandığı gibi altsistem adı çizelgesinde (CSQ1altsistemi için) tanımlanmalıdır. Örneklerde, komut öneki olarak +CSQ1 dizgisi kullanılır.

- Komutları yayınlamak için z/OS konsolunu kullanın

z/OS konsolundan basit komutlar yazabilirsiniz; örneğin:

```
+CSQ1 DISPLAY QUEUE (TRANSMIT .QUEUE .PROD) TYPE (QLOCAL)
```

Ancak, karmaşık komutlar ya da sık sık kullandığınız komut kümeleri için, diğer komut verme yöntemleri daha iyidir.

- Komut yanıtlarını al

Komutlara doğrudan yanıt, komutu veren konsola gönderilir. IBM MQ , z/OSiçinde bulunan *Extended Console Support (EMCS)* işlevini destekler ve bu nedenle 4 baytlık tanıtıcıları olan konsollar kullanılabilir. Ayrıca, komut MGCRC makrosunu kullanan bir program tarafından yayınlandığında, START QMGR ve STOP QMGR dışındaki tüm komutlar Komut ve Yanıt Belirteçlerinin (CART) kullanılmasını destekler.

İlgili görevler

[“z/OS üzerindeki işlemleri ve denetim panolarını kullanma” sayfa 471](#)

Bu panoları, IBM MQ nesnelere tanımlamak, görüntülemek, değiştirmek ya da silmek için kullanabilirsiniz. Günlük yönetim ve nesnelere üzerinde küçük değişiklikler yapmak için panoları kullanın.

[z/OS üzerinde TSO ortamı için örnek uygulamaların hazırlanması](#)

IBM MQ for z/OS için başlatma komutları

Kuyruk yöneticisinin başlatılmasını denetlemek için kullanıma hazırlama komutları kullanılabilir.

Kullanıma hazırlama giriş veri kümelerindeki komutlar, kuyruk yöneticisi başlatıldığında IBM MQ kullanıma hazırlandığında işlenir. Kullanıma hazırlama giriş veri kümelerinden üç tip komut verilebilir:

- Başka bir yerde tanımlanamayan IBM MQ varlıklarını tanımlayan komutlar; örneğin, DEFINE BUFFPOOL.

Bu komutlar, CSQINP1DD adıyla tanıtılan veri kümesinde bulunmalıdır. Bunlar, başlatma işleminin yeniden başlatma aşamasından önce işlenir. Bunlar konsol, işlemler ve denetim panoları ya da bir uygulama programı aracılığıyla yayınlanamaz. Bu komutlara verilen yanıtlar, başlatılan görev yordamının CSQOUT1 deyiminde gönderme yapılan sıralı veri kümesine yazılır.

- Yeniden başlatmadan sonra kurtarılabilir IBM MQ nesnelere tanımlamak için kullanılan komutlar. Bu tanımlar, CSQINP2DD adıyla tanıtılan veri kümesinde belirtilmelidir. Bunlar sayfa kümesi sıfır içinde depolanır. CSQINP2 , kullanıma hazırlama işleminin yeniden başlatma aşamasından sonra işlenir. Bu komutlara verilen yanıtlar, başlatılan görev yordamının CSQOUT2 deyiminde gönderme yapılan sıralı veri kümesine yazılır.
- IBM MQ nesnelere işlemek için komutlar. Bu komutlar, CSQINP2DD adıyla tanıtılan veri kümesinde de belirtilmelidir. Örneğin, IBM MQ tarafından sağlanan örnek, altsistem için bir girilmeyen ileti kuyruğu belirtmek üzere bir ALTER QMGR komutu içerir. Bu komutlara verilen yanıt CSQOUT2 çıkış veri kümesine yazılır.

Not: IBM MQ nesnelere CSQINP2içinde tanımlanmışsa IBM MQ , kuyruk yöneticisinin her başlatılışında bunları yeniden tanımlamayı dener. Nesnelere önceden varsa, bunları tanımlama girişimi başarısız olur. Nesnelere CSQINP2' de tanımlanmanız gerekiyorsa, DEFINE komutlarının REPLACE değiştirgesini kullanarak bu sorunu önleyebilirsiniz; ancak, kuyruk yöneticisinin önceki çalışması sırasında yapılan değişiklikleri geçersiz kılar.

Örnek kullanıma hazırlama veri kümesi üyeleri IBM MQ for z/OS ile sağlanır. Bunlar [IBM MQ ile verilen örnek tanımları](#) içinde açıklanır.

Dağıtılmış kuyruğa alma için kullanıma hazırlama komutları

START CHINIT komutu için CSQINP2 kullanıma hazırlama veri kümesini de kullanabilirsiniz. Dağıtılmış kuyruğa alma ortamınızı (örneğin, dinleyicileri başlatma gibi) tanımlamak için bir dizi başka komuta gereksinim duyarsanız, IBM MQ , kanal başlatıcısı tarafından başlatılan görev yordamının bir parçası olarak işlenen CSQINPX adlı üçüncü bir kullanıma hazırlama giriş verileri kümesi sağlar.

Veri kümesinde bulunan MQSC komutları kanal başlatıcı kullanıma hazırlama sonunda yürütülür ve çıkış, CSQOUTX DD deyimiyle belirtilen veri kümesine yazılır. Örneğin, dinleyicileri başlatmak için CSQINPX kullanıma hazırlama veri kümesini kullanabilirsiniz.

IBM MQ for z/OS ile birlikte örnek bir kanal başlatıcı kullanıma hazırlama veri kümesi üyesi sağlanır. Bu, [IBM MQ ile verilen örnek tanımları](#) içinde açıklanır.

Yayınlama/abone olma için kullanıma hazırlama komutları

Yayınlama/abone olma ortamınızı tanımlamak için bir dizi komuta gereksinim duyarsanız (örneğin, abonelikleri tanımlarken) IBM MQ , CSQINPT adı verilen dördüncü bir kullanıma hazırlama giriş verileri kümesi sağlar.

Veri kümesinde bulunan MQSC komutları yayınlama/abone olma kullanıma hazırlama sonunda yürütülür ve çıkış, CSQOUTT DD deyimiyle belirtilen veri kümesine yazılır. Örneğin, abonelikleri tanımlamak için CSQINPT kullanıma hazırlama veri kümesini kullanabilirsiniz.

IBM MQ for z/OS ile birlikte örnek bir yayınlama/abone olma kullanıma hazırlama veri kümesi üyesi sağlanır. Bu, [IBM MQ ile verilen örnek tanımları](#) içinde açıklanır.

IBM MQ for z/OS üzerinde özel ve genel tanımlar

IBM MQ for z/OS üzerinde bir nesne tanımladığınızda, bu tanımlamayı diğer kuyruk yöneticileriyle (*genel* tanımlaması) paylaşmak isteyip istemediğinizi ya da nesne tanımlamasının yalnızca bir kuyruk yöneticisi tarafından mı (*özel* tanımlama) kullanılacağını seçebilirsiniz. Buna *atmanesnesi* denir.

Genel tanımlama

Kuyruk yöneticiniz bir kuyruk paylaşım grubuna aitse, oluşturacağınız nesne tanımlamalarını grubun diğer üyeleriyle paylaşmayı seçebilirsiniz. Bu, bir nesneyi bir kez tanımlamanız ve tüm sistem için gereken toplam tanımlama sayısını azaltmanız gerektiği anlamına gelir.

Genel nesne tanımlamaları bir *paylaşılan havuzda* (Db2 paylaşılan veritabanı) tutulur ve kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından kullanılabilir. Bu nesnelerin grup (GROUP) olarak yok edilme durumu vardır.

Özel tanımlama

Yalnızca bir kuyruk yöneticisinin gerektirdiği bir nesne tanımlaması yaratmak istiyorsanız ya da kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesi değilse, bir kuyruk paylaşım grubunun diğer üyeleriyle paylaşılmayan nesne tanımlamaları yaratabilirsiniz.

Özel nesne tanımlamaları, tanımlayan kuyruk yöneticisinin sıfır sayfa kümesinde tutulur. Bu nesneler QMGR ' ye sahip.

You can create private definitions for all types of IBM MQ objects except CF structures (that is, channels, namelists, process definitions, queues, queue managers, storage class definitions, and authentication information objects), and global definitions for all types of objects except queue managers.

IBM MQ , bir grup nesnesinin tanımlamasını, onu kullanan her kuyruk yöneticisinin sayfa kümesi sıfırına otomatik olarak kopyalar. İstiyorsanız, tanımlamanın kopyasını geçici olarak değiştirebilir ve IBM MQ gerekirse, havuz kopyasından sayfa kümesi kopyalarını yenilemenize olanak sağlar.

IBM MQ her zaman, başlatma sırasında havuz kopyasından sayfa kümesi kopyalarını yenilemeyi dener (kanal komutları için, kanal başlatıcısı yeniden başlatıldığında ya da grup nesnesi değiştirildiğinde bu yapılır).

Not: Tanımlamanın kopyası grubun tanımlamasından yenilenir; ancak, tanımlamanın kopyasını yarattıktan sonra grup tanımlaması değiştiyse.

Bu, kuyruk yöneticisi etkin değilken yapılan değişiklikler de içinde olmak üzere, sayfa kümesi kopyalarının havuzdaki sürümü yansıtmasını sağlar. Kopyalar, DEFINE REPLACE komutları oluşturularak yenilenir; bu nedenle, yenileme işleminin gerçekleştirilmediği durumlar vardır; örneğin:

- Kuyruğun bir kopyası açıksa, kuyruk kullanımını değiştiren bir yenileme başarısız olur.
- Bir kuyruğun kopyasında iletiler varsa, o kuyruğu silen bir yenileme başarısız olur.
- Bir kuyruğun kopyasının değiştirilmesi için ALTER ve FORCE gerekir.

Bu durumlarda, yenileme o kopya üzerinde gerçekleştirilmez, ancak diğer tüm kuyruk yöneticilerindeki kopyalar üzerinde gerçekleştirilir.

Kuyruk yöneticisi kapatılıp bağımsız olarak yeniden başlatılırsa, örneğin, kuyruktaki ilişkili iletiler yoksa, nesnelerin yerel kopyaları silinir.

Yalnızca yerel kuyruklar için geçerli olan üçüncü bir nesne atma işlemi vardır. Bu, paylaşılan kuyruklar yaratmanızı sağlar. Paylaşılan bir kuyruğa ilişkin tanımlama, paylaşılan havuzda tutulur ve kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından kullanılabilir. Ayrıca, paylaşılan bir kuyruktaki iletiler, kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından da kullanılabilir. Bu, Paylaşılan kuyruklar ve kuyruk paylaşım grupları içinde açıklanır. Paylaşılan kuyrukların nesne atma durumu SHARED (paylaşılan).

Aşağıdaki çizelge, bağımsız olarak başlatılan kuyruk yöneticilerine ilişkin nesne düzenleme seçeneklerinin etkisini ve bir kuyruk paylaşım grubunun üyesi olarak özetler.

Süre	Bağımsız kuyruk yöneticisi	Kuyruk paylaşım grubunun üyesi
QMGR	Sıfır sayfa kümesinde tutulan nesne tanımlaması.	Sıfır sayfa kümesinde tutulan nesne tanımlaması.
GRUP	izin verilmez.	Paylaşılan havuzda tutulan nesne tanımlaması. Gruptaki her kuyruk yöneticisinin sayfa kümesinde yerel kopya tutuldu.
PAYLAŞILAN	izin verilmez.	Kuyruk tanımlaması paylaşılan havuzda tutuldu. Gruptaki herhangi bir kuyruk yöneticisinin kullanabileceği iletiler.

Genel tanımlamaları işleme

Paylaşılan havuzda tutulan bir nesnenin tanımlamasını değiştirmek istiyorsanız, havuzdaki sürümü mü, yoksa sıfır sayfa kümesindeki yerel kopyayı mı değiştirmek istediğinizi belirtmeniz gerekir. Bunu yapmak için komutun bir parçası olarak nesne atma özelliğini kullanın.

Komutları z/OS üzerinde farklı kuyruk yöneticilerine yönlendirme

Komutun hangi kuyruk yöneticisinde çalışacağını denetlemek için *komut kapsamını* kullanabilirsiniz.

Bir komutu girildiği kuyruk yöneticisinde ya da kuyruk paylaşım grubundaki farklı bir kuyruk yöneticisinde yürütmeyi seçebilirsiniz. Belirli bir komutu, bir kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine paralel olarak çalıştırmayı da seçebilirsiniz. Bu, hem MQSC komutları, hem de PCF komutları için olanaklıdır.

Bu, *komut kapsamı* tarafından belirlenir. Komut kapsamı, nesnenin hangi sürümüyle çalışmak istediğinizi saptamak için nesne atımıyla birlikte kullanılır.

Örneğin, tanımlaması paylaşılan havuzda tutulan bir nesnenin bazı özniteliklerini değiştirmek isteyebilirsiniz.

- Sürümü yalnızca bir kuyruk yöneticisinde değiştirmek ve havuzdaki sürümde ya da diğer kuyruk yöneticileri tarafından kullanılanlar üzerinde değişiklik yapmamak isteyebilirsiniz.
- Paylaşılan havuzdaki sürümü gelecekteki kullanıcılar için değiştirmek, ancak var olan kopyaları değiştirmeden bırakmak isteyebilirsiniz.
- Paylaşılan havuzdaki sürümü değiştirmek isteyebilirsiniz, ancak yaptığınız değişikliklerin, nesnenin bir kopyasını sıfır sayfa kümesinde tutan kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine hemen yansıtılmasını da isteyebilirsiniz.

Komutun bu kuyruk yöneticisinde mi, başka bir kuyruk yöneticisinde mi, yoksa tüm kuyruk yöneticisinde mi yürütüldüğünü belirtmek için komut kapsamını kullanın. İşlemekte olduğunuz nesnenin paylaşılan havuzda (grup nesnesi) mı, yoksa sayfa kümesi sıfır (kuyruk yöneticisi nesnesi) üzerinde yerel bir kopya mı olduğunu belirtmek için nesne atma özelliğini kullanın.

Paylaşılan bir kuyrukla çalışmak için komut kapsamını ve nesne atma özelliğini belirtmeniz gerekmez; kuyruk paylaşım grubundaki her kuyruk yöneticisi paylaşılan kuyruğu tek bir kuyruk olarak işler.

z/OS z/OS üzerindeki işlemleri ve denetim panolarını kullanma

Bu panoları, IBM MQ nesnelere tanımlamak, görüntülemek, değiştirmek ya da silmek için kullanabilirsiniz. Günlük yönetim ve nesnelere üzerinde küçük değişiklikler yapmak için panoları kullanın.

Başlamadan önce

IBM MQ for z/OS işlem ve denetim panoları (CSQOREXX), sürüm 7 'den itibaren eklenen tüm yeni işlevleri ve değişiklikleri desteklemeyebilir. Örneğin, konu nesnelere ya da aboneliklerin doğrudan işlenmesine ilişkin pano yoktur. Yayınlama/abone olma tanımlarını ve diğer panolarda doğrudan kullanılmayan diğer sistem denetimlerini yönetmek için aşağıdaki desteklenen düzenekleri kullanın:

1. IBM MQ Gezgin
2. z/OS Konsol
3. Programlanabilir Komut Biçimi (PCF) iletileri
4. CSQUTIL ' in COMMAND işlevi
5. IBM MQ Console

CSQOREXX panolarındaki soysal **Command** işleminin, SMDS ile ilgili komutlar da içinde olmak üzere, geçerli bir MQSC komutu vermenize olanak verdiğini unutmayın. CSQUTIL ' in COMMAND işlevi tarafından verilen tüm komutları kullanabilirsiniz.

IBM MQ komutlarını doğrudan panolardaki komut satırından yayınlamazsınız.

İşlemleri ve denetim panolarını kullanmak için doğru güvenlik yetkisine sahip olmanız gerekir; bu, [Komut güvenliği ve komut kaynak güvenliği için kullanıcı kimliklerikismında açıklanmıştır](#).

CSQUTIL ya da CSQOREXX panolarını kullanarak kullanıcı kimliği ve parola sağlayamazsınız. Bunun yerine, kullanıcı kimliğiniz MQCONN ' daki BATCH profili için UPDATE yetkisine sahipse, **CHCKLOCL**(REQUIRED) ayarını atlayabilirsiniz. Ek bilgi için [CHCKLOCL ' in yerel olarak bağlı uygulamalarda kullanılması](#) başlıklı konuya bakın.

Birçok nesne ayarlıyor ya da değiştiriyorsanız, CSQUTIL yardımcı programının COMMAND işlevini kullanın. Bkz. ["IBM MQ for z/OS için CSQUTIL yardımcı programının kullanılması" sayfa 481](#).

Bu görev hakkında

İşlemler ve denetim panoları, kanal başlatıcısına (örneğin, bir kanal ya da TCP/IP dinleyicisi başlatmak için), kümelemeye ve güvenliğe ilişkin denetimleri destekler. Ayrıca, iş parçacıkları ve sayfa kümesi kullanımı hakkında bilgi görüntülemenizi de sağlar.

Paneller, sistem komutu giriş kuyruğu aracılığıyla bir kuyruk yöneticisine MQSC tipi IBM MQ komutları gönderilerek çalışır.

Örnek

Bir pano oturumu başlattığınızda görüntülenen pano:

```
IBM MQ for z/OS - Main Menu
Complete fields. Then press Enter.
Action . . . . . 1      0. List with filter  4. Manage
                          1. List or Display  5. Perform
                          2. Define like    6. Start
                          3. Alter          7. Stop
                          8. Command
Object type . . . . . CHANNEL +
Name . . . . . *
Disposition . . . . . A  Q=Qmgr, C=Copy, P=Private, G=Group,
                          S=Shared, A=All
Connect name . . . . . M01C - local queue manager or group
Target queue manager . . M01C
                          - connected or remote queue manager for command input
Action queue manager . . M01C - command scope in group
Response wait time . . . 30  5 - 999 seconds
(C) Copyright IBM Corporation 1993, 2024. All rights reserved.
Command ==>
F1=Help      F2=Split      F3=Exit      F4=Prompt      F9=SwapNext F10=Messages
F12=Cancel
```

Bu panoda aşağıdaki gibi işlemleri gerçekleştirebilirsiniz:

- İstedığınız yerel kuyruk yöneticisini seçin ve bu kuyruk yöneticisinde, uzak kuyruk yöneticisinde ya da yerel kuyruk yöneticisiyle aynı kuyruk paylaşım grubundaki başka bir kuyruk yöneticisinde verilen komutları isteyip istemediğinizi belirleyin. Değiştirmeniz gerekiyorsa, kuyruk yöneticisi adını üzerine yazın.
- **Eylem** alanına uygun sayıyı yazarak gerçekleştirmek istediğiniz işlemi seçin.
- Çalışmak istediğiniz nesne tipini belirleyin. Nesne tiplerinin ne olduğundan emin değilseniz, nesne tipleriyle ilgili yardım almak için F1 işlev tuşuna basın.
- Çalışmak istediğiniz nesne tipinin atılmasını belirtin.
- Belirlenen tipte nesnelerin listesini görüntüler. **Ad** alanına bir yıldız işareti (*) yazın ve işlem kuyruğu yöneticisinde önceden tanımlanmış olan nesnelerin listesini (belirtilen tipte) görüntülemek için **Enter** tuşuna basın. Daha sonra, sırayla çalışmak için bir ya da daha çok nesne seçebilirsiniz. Tüm işlemler listeden kullanılabilir.

Not: Görüntülenmekte olan nesneler listesiyle sonuçlanan seçimleri yapmanız ve daha sonra bu listeden çalışmanız önerilir. **Görüntüle** işlemini kullanın; bu işlem tüm nesne tipleri için geçerlidir.

z/OS İşlemler ve denetim panoları için çağırma ve kurallar

IBM MQ denetim ögesini denetleyebilir ve denetim komutlarını ISPF panolarından yayınlayabilirsiniz.

IBM MQ işlemlerine ve denetim panolarına erişme

IBM MQ için ISPF/PDF birincil seçenekler menüsü güncellendiye, o menüden IBM MQ işlemlerine ve denetim panolarına erişebilirsiniz. Menünün güncellenmesiyle ilgili ayrıntılar için [Görev 20: İşlemleri ve denetim panolarını ayarlamak](#) konusuna bakın.

IBM MQ işlemlerine ve denetim panolarına TSO komut işlemcisi panosundan (genellikle ISPF/PDF birincil seçenekler menüsünde seçenek 6) erişebilirsiniz. Bunu yapmak için çalıştırdığınız yürütücünün adı CSQOREXX 'tir. İki parametresi vardır; th1qua1 , kullanılacak IBM MQ kitaplıkları için üst düzey niteleyicidir ve lang1etter , kullanılacak ulusal dil kitaplıklarını tanımlayan harftir (örneğin, U.S). İngilizce). IBM MQ kitaplıkları ISPF kuruluşunuzda kalıcı olarak kuruluysa, parametreler atlanabilir. Diğer bir seçenek olarak, TSO komut satırından CSQOREXX komutunu kullanabilirsiniz.

Bu paneller, en az resmi eğitimi olan operatörler ve yöneticiler tarafından kullanılmak üzere tasarlanmıştır. Panoları çalışırken bu yönergeleri okuyun ve önerilen farklı görevleri deneyin.

Not: Panoları kullanırken, SYSTEM.CSQOREXX.* oluşturulmuştur.

İşlemler ve denetim panolarına ilişkin kurallar

IBM MQ karakter dizilimleri ve adlarına ilişkin genel kurallar hakkında [IBM MQ nesnelere adlandırma kuralları](#) başlıklı konuya bakın. Ancak, yalnızca işlemler ve denetim panoları için geçerli olan bazı kurallar vardır:

- Dizgileri (örneğin, açıklamaları) tek ya da çift tırnak içine almayın.
- Bir metin alanına kesme işareti ya da tırnak işareti eklerseniz, bunu yinelemeniz ya da bir çıkış karakteri eklemeniz gerekmez. Karakterler tam olarak sizin yazdığınız şekilde kaydedilir; örneğin:

```
This is Maria's queue
```

Pano işlemcisi bunları IBM MQ' e geçirmeniz için iki katına çıkartır. Ancak, bunu yapmak için verilerinizin kesilmesi gerekirse, kesilir.

- Çoğu alanda büyük ya da küçük harf karakterleri kullanabilirsiniz; bu karakterler Enter tuşuna bastığınızda büyük harflerle katlanır. Kural dışı durumlar şunlardır:
 - Depolama sınıfı adları ve bağlaşım olanağı yapısı adları. Bu adları A-Z arası büyük harfle başlamalı ve ardından A-Z arasındaki büyük harfler ya da sayısal karakterler kullanılmalıdır.
 - Çevrilmeyen belirli alanlar. Bunlar arasında aşağıdakiler yer alır:
 - Uygulama tanıtıcısı
 - Açıklama
 - Ortam verileri
 - Nesne adları (ancak küçük harfli bir nesne adı kullanırsanız, bunu z/OS konsolunda giremeyebilirsiniz)
 - Uzak sistem adı
 - Tetikleyici verileri
 - Kullanıcı verileri
- Adlarda, baştaki boşluklar ve baştaki alt çizgiler yoksayılr. Bu nedenle, boş ya da alt çizgiyle başlayan nesne adlarına sahip olamazsınız.
- Alt çizgiler, boş alanların kapsamını göstermek için kullanılır. Enter tuşuna bastığınızda, sondaki alt çizgiler boşluklarla değiştirilir.
- Birçok açıklama ve metin alanı, IBM MQ tarafından bağımsız olarak işlenmekte olan birden çok bölümde sunulur. Bu, sondaki boşlukların korunduğu ve metnin bitişik olmadığı anlamına gelir.

Boş alanlar

Bir IBM MQ nesnesi için **Tanımla** işlemini belirlediğinizde, tanım panosundaki her alan bir değer içerir. IBM MQ ' in değerleri nereden almasıyla ilgili bilgi için görüntü panoları için genel yardıma (genişletilmiş yardım) bakın. Boşluk içeren bir alanın üzerine yazarsanız ve boşluklara izin verilmezse, IBM MQ kuruluş varsayılan değerini alana koyar ya da gerekli değeri girmenizi ister.

Bir IBM MQ nesnesi için **Değiştirme** işlemini belirlediğinizde, değiştirme panosundaki her alan o alanın yürürlükteki değerini içerir. Boşluk içeren bir alanın üzerine yazarsanız ve bu alana boşluk girilmesine izin verilmezse, bu alanın değeri değişmez.

z/OS z/OS üzerindeki nesnelere ve işlemler

İşlemler ve denetim panoları size birçok farklı nesne tipi ve bunlar üzerinde gerçekleştirebileceğiniz bir dizi işlem sunar.

İşlemler ilk panoda listelenir ve nesnelere işlemenizi ve onlarla ilgili bilgileri görüntülemenizi sağlar. Bu nesnelere, bazı fazladan nesnelere birlikte tüm IBM MQ nesnelere içerir. Nesnelere aşağıdaki kategorilere ayrılır.

- Kuyruklar, işlemler, kimlik doğrulama bilgileri nesnelere, ad listesi, depolama sınıfları ve CF yapıları
- Kanallar
- Küme nesnelere
- Kuyruk yöneticisi ve güvenliği
- Bağlantılar
- Sistem

IBM MQ nesnelereyle yapılabilecek işlemlere ilişkin çapraz başvuru tablosu için İşlemler konusuna bakın.

Kuyruklar, işlemler, kimlik doğrulama bilgileri nesnelere, ad listesi, depolama sınıfları ve CF yapıları

Bunlar temel IBM MQ nesnelereidir. Her tipte birden çok olabilir. LIST ya da DISPLAY, LIST ve FILTER, DEFINE, DEFINE, MANAGE ve ALTER işlemlere kullanılarak bunlar listelenebilir, süzgeçle listelenebilir, tanımlanabilir ve silinebilir ve görüntülenebilecek ve değiştirilebilecek öznelere sahip olabilir. (Nesnelere MANAGE (YÖNET) işlemlere kullanılarak silinir.)

Bu kategori aşağıdaki nesnelere oluşur:

QLOCAL	Yerel kuyruk
QUZAK	Uzak kuyruk
QDIĞER AD	Bir kuyruğa dolaylı başvuru için diğere ad kuyruğu
QMODEL	Kuyrukları dinamik olarak tanımlamak için model kuyruğu
kuyruk	Herhangi bir kuyruk tipi
QSTATUS (DURUM)	Yerel kuyruğun durumu
Süreç	Tetikleme olayı oluştuğunda başlatılacak uygulamaya ilişkin bilgiler
AUTHINFO	Kimlik doğrulama bilgileri: LDAP sunucularını kullanarak CRL (Certificate Revocation List; Sertifika İptal Listesi) denetimini gerçekleştirmek için gerekli tanımlar
ADı LISTESI	Kuyruk ya da küme gibi adların listesi
STGCLASS	Depolama sınıfı
CFSTRUCT	bağlaşım olanağı (CF) yapısı
CFSTATUS	CF yapısının durumu

Kanallar

Kanallar, dağıtılmış kuyruğa alma için kullanılır. Her tipte birden çok tip olabilir ve bunlar, süzgeç kullanılarak listelenebilir, tanımlanabilir, silinebilir, görüntülenebilir ve değiştirilebilir. Ayrıca, START, STOP ve PERFORM işlemlere kullanarak kullanılabilir başka işlevler de vardır. PERFORM, kanal işlevlerini ilk durumuna getirme, ping ve resolve işlevleri sağlar.

Bu kategori aşağıdaki nesnelere oluşur:

Kanal	Herhangi bir kanal tipi
Gönderen	Gönderen kanalı
SERVER	Sunucu kanalı
Alıcı	Alıcı kanalı
İsteyen	İsteyen kanalı

CLUSRCVR	Küme-alıcı kanalı
KLUSSDR	Küme-gönderen kanalı
SVRCONN	Sunucu bağlantısı kanalı
CLNTCONN	İstemci-bağlantı kanalı
DURUM	Kanal bağlantısının durumu

Küme nesneleri

Küme nesneleri, bir kümeye ait olan kuyruklar ve kanallar için otomatik olarak yaratılır. Temel kuyruk ve kanal tanımlamaları başka bir kuyruk yöneticisinde olabilir. Her tipte birden çok olabilir ve adlar yinelenir. Bunlar listelenebilir, süzgeçle birlikte listelenebilir ve görüntülenebilir. PERFORM, START ve STOP, LIST işlemleriyle de kullanılabilir.

Bu kategori aşağıdaki nesnelere oluşur:

KLUSQ	Kümeye ait bir kuyruk için yaratılan küme kuyruğu
CLUSCHL.	Kümeye ait bir kanal için oluşturulan küme kanalı
CLUSQMgr	Küme kuyruğu yöneticisi, bir küme kanalı ile aynı, ancak kuyruk yöneticisi adıyla tanıtılır

Küme kanalları ve küme kuyruğu yöneticilerinin PERFORM, START ve STOP işlemleri vardır, ancak DISPLAY işlemi aracılığıyla dolaylı olarak gerçekleştirilir.

Kuyruk yöneticisi ve güvenlik

Kuyruk yöneticisi ve güvenlik nesnelерinin tek bir eşgörünümü vardır. Bunlar listelenebilir ve görüntülenebilecek ve değiştirilebilecek özniteliklere (LIST ya da DISPLAY ve ALTER işlemleri kullanılarak) ve PERFORM işlemi kullanılarak kullanılacak diğer işlemlere sahip olabilir.

Bu kategori aşağıdaki nesnelere oluşur:

Yönetici	Kuyruk yöneticisi: PERFORM işlemi, kümeyi askıya alma ve sürdürme işlemleri sağlar
GÜVENLİK	Güvenlik işlemleri: PERFORM işlemi, yenileme ve yeniden doğrulama işlemleri sağlar

Bağlantı

Bağlantılar, süzgeçle birlikte listelenebilir ve görüntülenebilir.

Bu kategori yalnızca CONNECT bağlantı nesnesinden oluşur.

Sistem

Diğer işlemlerden oluşan bir derlem. Bu kategori aşağıdaki nesnelere oluşur:

SYSTEM	Sistem işlemleri
CONTROL	SYSTEM için eşanlı

Kullanılabilir işlemler şunlardır:

LIST ya da DISPLAY	Kuyruk paylaşım grubunu, dağıtılmış kuyruğa alma, sayfa kümesi ya da veri kümesi kullanım bilgilerini görüntüler.
GERÇEKLEŞTİR	Kümelemeyi yenile ya da ilk durumuna getir
START	Kanal başlatıcıyı ya da dinleyicileri başlat
DUR	Kanal başlatıcıyı ya da dinleyicileri durdur

İşlemler

Her nesne tipi için gerçekleştirebileceğiniz işlemler aşağıdaki çizelgede gösterilir:

<i>Çizelge 27. IBM MQ nesneleri için geçerli işlemler ve denetim panosu işlemleri</i>								
Nesne	Değiştir	Benzerler ini tanımla	Yönet (1)	Liste ya da Görüntü	Süzgeç İçerenler Listesi	Gerçekleş tirme	Başlat	Durdur
AUTHINFO	X	X	X	X	X			
CFSTATUS				X				
CFSTRUCT	X	X	X	X	X			
Kanal	X	X	X	X	X	X	X	X
DURUM				X	X			
CLNTCONN	X	X	X	X	X			
CLUSCHL.				X	X	X (2)	X (2)	X (2)
KLUSQ				X	X			
CLUSQMGR				X	X	X (2)	X (2)	X (2)
CLUSRCVR	X	X	X	X	X	X	X	X
KLUSSDR	X	X	X	X	X	X	X	X
CONNECT				X	X			
CONTROL				X		X	X	X
Yönetici	X			X		X		
ADı LISTESİ	X	X	X	X	X			
Süreç	X	X	X	X	X			
QDIĞER AD	X	X	X	X	X			
QLOCAL	X	X	X	X	X			
QMODEL	X	X	X	X	X			
QUZAK	X	X	X	X	X			
QSTATUS (DURUM)				X	X			
kuyruk	X	X	X	X	X			
Alıcı	X	X	X	X	X	X	X	X
İsteyen	X	X	X	X	X	X	X	X
GÜVENLİK	X			X		X		
Gönderen	X	X	X	X	X	X	X	X
SERVER	X	X	X	X	X	X	X	X
SVRCONN	X	X	X	X	X		X	X
STGCLASS	X	X	X	X	X			
SYSTEM				X		X	X	X

Not:

1. Delete ve diğer işlevleri sağlar
2. Listele ya da Görüntüle işleminin kullanılması

z/OS üzerinde nesne işlemleri

Çalışmanız gereken nesnenin *atma* işlemini belirtebilirsiniz. Yok etme, **tanım** nesnesinin nerede tutulduğunu ve nesnenin nasıl davrandığını belirtir.

Yok etme yalnızca aşağıdaki nesne tiplerinden herhangi biriyle çalışıyorsanız önemlidir:

- Kuyruklar
- kanallar
- süreçler
- ad listesi
- depolama sınıfları
- kimlik doğrulama bilgileri nesneleri

Başka nesne tipleriyle çalışıyorsanız, yok etme dikkate edilmez.

İzin verilen değerler şunlardır:

Q

QMGR. Nesne tanımlamaları kuyruk yöneticisinin sayfa kümesinde bulunur ve yalnızca kuyruk yöneticisi tarafından erişilebilir.

C

ANLAŞILDI. Nesne tanımlamaları kuyruk yöneticisinin sayfa kümesinde bulunur ve yalnızca kuyruk yöneticisi tarafından erişilebilir. Bunlar, atma grubu olarak tanımlanan nesnelerin yerel kopyalarıdır.

P

ÖZEL. Nesne tanımlamaları kuyruk yöneticisinin sayfa kümesinde bulunur ve yalnızca kuyruk yöneticisi tarafından erişilebilir. Nesnelere, QMGR ya da COPY yok etme özelliğine sahip olarak tanımlandı.

G

GRUP. Nesne tanımlamaları paylaşılan havuzda yer alır ve kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından erişilebilir.

S

PAYLAŞILAN Bu yok etme yalnızca yerel kuyruklar için geçerlidir. Kuyruk tanımlamaları paylaşılan havuzdadır ve kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından erişilebilir.

A

-HEPSI. İşlem kuyruğu yöneticisi hedef kuyruk yöneticisi ya da * ise, **tüm** işlemlerine ilişkin nesnelere içerilir; tersi durumda, yalnızca QMGR ve COPY işlemlerine ilişkin nesnelere içerilir. Bu varsayılandır.

z/OS üzerinde ISPF denetim panosunu kullanarak bir kuyruk yöneticisi, varsayılan değer ve düzey seçilmesi

Kuyruk yöneticilerinizi denetlemek için ISPF ' de CSQOREXX exec 'i kullanabilirsiniz.

İlk panoyu görüntülerken herhangi bir kuyruk yöneticisine bağlı değilsiniz. Ancak, Enter tuşuna bastığınızda, kuyruk yöneticisine ya da **Bağlantı adı** alanında adı belirtilen kuyruk paylaşım grubundaki bir kuyruk yöneticisine bağlanırsınız. Bu alanı boş bırakabilirsiniz; bu, toplu iş uygulamaları için varsayılan kuyruk yöneticisini kullanmakta olduğunuz anlamına gelir. Bu, CSQBDEFV ' de tanımlanır (bununla ilgili bilgi için [Görev 19: Toplu İş, TSO ve RRS bağdaştırıcılarının ayarlanması](#) konusuna bakın).

İstedığınız işlemlerin gerçekleştirileceği kuyruk yöneticisini belirtmek için **Hedef kuyruk yöneticisi** alanını kullanın. Bu alanı boş bırakırsanız, varsayılan değer olarak **Bağlantı adı** alanında belirtilen kuyruk yöneticisi kullanılır. Bağlandığınız hedef kuyruk yöneticisi olmayan bir hedef kuyruk yöneticisi belirtebilirsiniz. Bu durumda, olağan durumda bir kuyruk yöneticisi diğer adı tanımlaması sağlayan uzak

kuyruk yöneticisi nesnesinin adını belirtirsiniz (bu ad, komut giriş kuyruğunu açarken *ObjectQMgrAdı* olarak kullanılır). Bunu yapmak için, uzak kuyruk yöneticisine erişmek üzere uygun kuyruklara ve kanallara sahip olmanız gerekir.

İşlem kuyruğu yöneticisi alanı, **Hedef kuyruk yöneticisi** alanında belirtilen kuyruk yöneticisiyle aynı kuyruk paylaşım grubunda bulunan bir kuyruk yöneticisini, istekte bulunacağınız işlemlerin gerçekleştirileceği kuyruk yöneticisi olarak belirtmenizi sağlar. Bu alanda * belirlerseniz, istediğiniz işlemler kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine gerçekleştirilir. Bu alanı boş bırakırsanız, varsayılan değer olarak **Hedef kuyruk yöneticisi** alanında belirtilen değer kullanılır. **İşlem kuyruğu yöneticisi** alanı, MQSC komutları içinde açıklanan CMDSCOPE komut değiştiricisinin kullanılmasına karşılık gelir.

Kuyruk yöneticisi varsayılanları

Herhangi bir kuyruk yöneticisi alanını boş bırakırsanız ya da bir kuyruk paylaşım grubuna bağlanmayı seçerseniz, **Enter** tuşuna bastığınızda ikincil bir pencere açılır. Bu pencere, kullanacağınız kuyruk yöneticilerinin adlarını doğrular. Devam etmek için **Enter** tuşuna basın. Bazı istekler yaptıktan sonra ilk panoya döndüğünüzde, gerçek adlarla doldurulmuş alanlar bulursunuz.

Kuyruk yöneticisi düzeyleri

İşlem kuyruğu yöneticisi IBM MQ 8.0.0ya da üstünde değilse, bazı alanlar görüntülenmez ve bazı değerler girilemiyor. Birkaç nesne ve eyleme izin verilmez. Böyle durumlarda, devam etmek istediğinizi onaylamanızı isteyen ikincil bir pencere açılır.

z/OS üzerinde ISPF denetim panolarıyla işlev tuşlarının ve komut satırının kullanılması

Panoları kullanmak için, işlev tuşlarını kullanmanız ya da ISPF denetim panosu komut alanına eşdeğer komutları girmeniz gerekir.

- İşlev tuşları
 - İşlemlerinizi işleme
 - “IBM MQ kullanıcı iletilerinin görüntülenmesi” sayfa 479
 - İşlemlerinizi iptal etme
 - Yardım alma
- Komut satırının kullanılması

İşlev tuşları

İşlev tuşlarının IBM MQ için özel ayarları vardır. (Bu, işlev tuşları için ISPF varsayılan değerlerini kullanamayacağınız anlamına gelir; KEYLIST OFF ISPF komutunu daha önce herhangi bir yerde kullandıysanız, herhangi bir işlem ve denetim panosunun komut alanına KEYLIST ON yazmanız ve ardından IBM MQ ayarlarını etkinleştirmek için Enter tuşuna basmanız gerekir.)

Bu işlev tuşu ayarları, “z/OS üzerindeki işlemleri ve denetim panolarını kullanma” sayfa 471 içinde gösterildiği gibi panolarda görüntülenebilir. Ayarlar gösterilmezse, herhangi bir işlemin ve denetim panosunun komut alanına PFSHOW yazın ve **Enter** tuşuna basın. Ayarların görüntüsünü kaldırmak için PFSHOW OFF komutunu kullanın.

İşlemler ve denetim panolarındaki işlev tuşu ayarları CUA standartlarına uygundur. Anahtar ayarını olağan ISPF yordamlarıyla (**KEYLIST** yardımcı programı gibi) değiştirebilmeniz de, değiştirmeniz önerilmez.

Not: **PFSHOW** ve **KEYLIST** komutlarının kullanılması, sahip olduğunuz diğer mantıksal ISPF ekranlarını etkiler ve bu ekranların ayarları, işlemlerden ve denetim panolardan ayrıldığınızda kalır.

İşlemlerinizi işleme

Bir panoda istenen işlemi gerçekleştirmek için **Enter** tuşuna basın. Panodaki bilgiler, işlemek üzere kuyruk yöneticisine gönderilir.

Panolarda **Enter** tuşuna her bastığınızda IBM MQ bir ya da daha çok işletmen iletisi oluşturur. İşlem başarılı olduysa, CSQ9022I onay iletisini alırsınız, tersi durumda bazı hata iletileri alırsınız.

IBM MQ kullanıcı iletilerinin görüntülenmesi

IBM MQ kullanıcı iletilerini görmek için herhangi bir panoda F10 işlev tuşuna basın.

İşlemlerinizi iptal etme

İlk panoda hem F3 , hem de F12 işlem denetim panolarından çıkıp ISPF' ye geri dönün. Kuyruk yöneticisine bilgi gönderilmez.

Başka bir panoda, geçerli panelden çıkmak için F3 ya da F12 tuşuna basarak **son Enter tuşuna bastıktan sonra yazdığınız verileri yoksayabilirsiniz**. Kuyruk yöneticisine yine bilgi gönderilmez.

- F3 , sizi doğrudan ilk panoya geri götürür.
- F12 , sizi önceki panoya geri götürür.

Yardım alma

Her panoda ilişkili birlikte olmak "olmak" için yardım panoları vardır. Yardım panoları ISPF iletişim kurallarını kullanır:

- Görevle ilgili genel yardımı (genişletilmiş yardım) görmek için herhangi bir panoda F1 işlev tuşuna basın.
- Belirli bir alana ilişkin yardımı görmek için, imleç herhangi bir alanda olduğu şekilde F1 işlev tuşuna basın.
- Genel yardım almak için herhangi bir alan yardım panosunda F5 işlev tuşuna basın.
- Temel panoya (F1 işlev tuşuna bastığınız panoya) dönmek için F3 işlev tuşuna basın.
- İşlev tuşlarıyla ilgili yardım almak için herhangi bir yardım panosundan F6 işlev tuşuna basın.

Yardım bilgileri ikinci ya da sonraki sayfalara devam ederse, panonun sağ üst kısmında bir **Diğer** göstergesi görüntülenir. Yardım sayfalarında dolaşmak için bu işlev tuşlarını kullanın:

- F11 (varsa) sonraki yardım sayfasına gitmek için.
- Önceki yardım sayfasına (varsa) dönmek için F10 tuşuna basın.

Komut satırının kullanılması

İşlev tuşlarından kullanılabildikleri için, işlemler ve denetim panoları tarafından kullanılan komutları vermek için komut satırını kullanmanız gerekmez. Komut satırı, olağan ISPF komutlarını (**PFSHOW** gibi) girmenizi sağlamak için sağlanır.

ISPF komutu PANELID ON , yürürlükteki CSQOREXX panosunun adını görüntüler.

Komut satırı, başlangıçta panoların alt kısmındaki varsayılan konumda, sahip olduğunuz ISPF ayarlarından bağımsız olarak görüntülenir. Komut satırının konumunu değiştirmek için herhangi bir işlem ve denetim panosundan SETTINGS ISPF komutunu kullanabilirsiniz. Ayarlar, işlemler ve denetim panoları ile sonraki oturumlarda hatırlanır.

z/OS

IBM MQ for z/OS yardımcı programlarının kullanılması

IBM MQ for z/OS , sistem denetimiyle ilgili yardım almak için kullanabileceğiniz bir dizi yardımcı program sağlar.

IBM MQ for z/OS , aşağıdakiler de içinde olmak üzere çeşitli yönetim görevlerini gerçekleştirmenize yardımcı olacak bir dizi yardımcı program sağlar:

- İleti güvenliği ilkelerini yönetin.

- Yedekleme, geri yükleme ve yeniden düzenleme görevlerini gerçekleştirin.
- Komutları ve nesne tanımlamalarını yayınlayın.
- Veri dönüştürme çıktıları oluşturun.
- Önyükleme veri kümesini değiştirin.
- Günlüklerle ilgili bilgileri listeleyin.
- Günlükleri yazdırın.
- Db2 çizelgelerini ve diğer Db2 yardımcı programlarını ayarlayın.
- İletileri, gönderilteki gönder belki risydı.

İleti güvenliği ilkesi yardımcı programı

İleti güvenliği ilkesi yardımcı programı (CSQ0UTIL), ileti güvenliği ilkelerini yönetmek için bağımsız bir yardımcı program olarak çalışır. Ek bilgi için [İleti güvenliği ilkesi yardımcı programı \(CSQ0UTIL\)](#) konusuna bakın.

CSQUTIL yardımcı programı

Bu, yedekleme, geri yükleme ve yeniden düzenleme görevlerinde size yardımcı olmak için sağlanan bir yardımcı programdır. Bkz. [“IBM MQ for z/OS için CSQUTIL yardımcı programının kullanılması”](#) sayfa 481.

Veri dönüştürme çıkış yardımcı programı

IBM MQ for z/OS veri dönüştürme çıkış yardımcı programı (**CSQUCVX**) Veri dönüştürme çıkışı yordamlarını yaratmak için bağımsız bir yardımcı program olarak çalışır.

Değişiklik günlüğü stok yardımcı programı

IBM MQ for z/OS değişiklik günlüğü döküm yardımcı programı (**CSQJU003**) önyükleme veri kümesini (BSDS) değiştirmek için bağımsız bir yardımcı program olarak çalışır. Yardımcı programı kullanarak aşağıdaki işlevleri gerçekleştirebilirsiniz:

- Etkin ya da arşiv günlüğü veri kümeleri ekleyin ya da silin.
- Arşiv günlükleri için parola sağlayın.

Günlük eşleme yazdırma yardımcı programı

IBM MQ for z/OS yazdırma günlüğü eşleme yardımcı programı (**CSQJU004**) Aşağıdaki bilgileri listelemek için bağımsız bir yardımcı program olarak çalışır:

- Tüm etkin ve arşiv günlük veri kümelerinin her iki kopyası için günlük veri kümesi adı ve günlük RBA ilişkilendirmesi. İkili günlük kaydı etkin değilse, veri kümelerinin yalnızca bir kopyası vardır.
- Yeni günlük verileri için etkin günlük veri kümeleri.
- Önyükleme veri kümesindeki (BSDS) denetim noktası kayıtları kuyruğunun içeriği.
- Arşiv günlüğü komut geçmiş kaydının içeriği.
- Sistem ve yardımcı zaman damgaları.

Günlük yazdırma yardımcı programı

Günlük yazdırma yardımcı programı (**CSQ1LOGP**) bağımsız bir yardımcı program olarak çalıştırılır. Yardımcı programı aşağıdaki belirtilerek çalıştırabilirsiniz:

- Bir önyükleme veri kümesi (BSDS)
- Etkin günlükler (BSDS olmadan)

- Günlükleri arşivle (BSDS olmadan)

Kuyruk paylaşım grubu yardımcı programı

Kuyruk paylaşım grubu yardımcı programı (**CSQ5PQSG**) Db2 çizelgelerini ayarlamak ve kuyruk paylaşım grupları için gereken diğer Db2 görevlerini gerçekleştirmek için bağımsız bir yardımcı program olarak çalışır.

Etkin günlük preformat yardımcı programı

Etkin günlük önbiçimli yardımcı programı (**CSQJUFMT**) Etkin günlük veri kümelerini, bir kuyruk yöneticisi tarafından kullanılmadan önce biçimlendirir. Etkin günlük veri kümeleri yardımcı program tarafından önceden biçimlendirildiyse, kuyruk yöneticisinin etkin günlükler üzerinden ilk geçişte günlük yazma performansı iyileştirilir.

Teslim mektubu kuyruk işleyicisi yardımcı programı

Teslim edilen ileti kuyruğu işleyicisi yardımcı programı (**CSQUDLQH**) bağımsız bir yardımcı program olarak çalışır. Bu ileti, ileti kuyruğundaki iletileri denetler ve bunları yardımcı programa sağladığınız bir dizi kurala göre işler.

Kuyruk yükleme ve boşa alma yardımcı programı

Kuyruk yükleme ve boşa alma yardımcı programı, bir kuyruğun ya da iletilerinin içeriğini bir dosyaya kopyalar ya da taşır. Yardımcı program başlangıçta IBM MQ Supportpac MO03' te **QLOAD** yardımcı programı olarak gönderildi. IBM MQ 8.0 ' den, uyumluluk için QLOAD diğer adıyla SCSQLOAD kitaplığında **CSQDMSG** yürütülebilir modülü olarak ürünle tümleştirilir. Örnek JCL, SCSQPROC ' de CSQ4QLOD üyesi olarak sağlanır.

Multiplatforms için eşdeğer yardımcı program **dmpmqmsg** olarak adlandırılır. z/OS ile ilgili farklar da içinde olmak üzere kullanılabilir seçeneklerin ayrıntıları için bkz. **dmpmqmsg** (kuyruk yükleme ve boşa alma).

İletileri İletilerin z/OS üzerinde bir veri kümesinden bir kuyruğa geri yüklenmesi (LOAD) ve z/OS üzerinde bir veri kümesinden iletileri kuyruğa geri yükleme (SLOAD) içinde açıklandığı gibi yeniden yükleyebilirsiniz.

z/OS IBM MQ for z/OS için CSQUTIL yardımcı programının kullanılması

CSQUTIL yardımcı programı, yedekleme, geri yükleme ve yeniden düzenleme görevlerini gerçekleştirmenize ve komutları ve nesne tanımlamalarını işlemenize yardımcı olmak için IBM MQ for z/OS ile birlikte sağlanır.

Bu görev hakkında

Aşağıdaki işlevleri çağırarak için bu yardımcı programı kullanın. Örneğin, CSQUTIL yardımcı programının COMMAND işlevini kullanarak sıralı bir veri kümesinden komut verebilirsiniz. Bu işlev, komutları ileti olarak *sistem komutu giriş kuyruğuna* aktarır ve SYSPRINT ' deki özgün komutlarla birlikte yazdırılan yanıtı bekler.

CSQUTIL yardımcı programına ilişkin ek bilgi için bkz. [IBM MQ yardımcı programı \(CSQUTIL\)](#).

Yordam

- [Komut](#)

MQSC komutları yayınlamak, nesne tanımlamalarını kaydetmek ve istemci-kanal tanımlama dosyaları yapmak için bu işlevi kullanın.

- [kopyala](#)

Adlandırılmış bir IBM MQ for z/OS ileti kuyruğunun ya da adlandırılmış bir sayfa kümesinin tüm kuyruklarının içeriğini okumak ve bunları sıralı bir dosyaya koymak ve özgün kuyruğu korumak için bu işlevi kullanın.

- KOPYALAMA

Tüm sayfa kümelerini daha büyük sayfa kümelerine kopyalamak için bu işlevi kullanın.

- EMPTY

Bu işlev, adlandırılmış bir IBM MQ for z/OS ileti kuyruğunun içeriğini ya da adlandırılmış bir sayfa kümesinin tüm kuyruklarının içeriğini, kuyrukların tanımlarını tutarak silmek için kullanılır.

- Biçim

IBM MQ for z/OS sayfa kümelerini biçimlendirmek için bu işlevi kullanın.

- İletilerin z/OS üzerinde bir veri kümesinden bir kuyruğa geri yüklenmesi (LOAD)

Bu işlev, adlandırılmış bir IBM MQ for z/OS ileti kuyruğunun içeriğini ya da adlandırılmış bir sayfa kümesinin tüm kuyruklarının içeriğini COPY işlevi tarafından yaratılan sıralı bir dosyadan geri yüklemek için kullanılır.

- SAYFABİLGİSİ

Bir ya da daha çok sayfa kümesinden sayfa kümesi bilgilerini almak için bu işlevi kullanın.

- YENİDEN SAYFA

Tüm sayfa kümelerini diğer sayfa kümesi veri kümelerine kopyalamak ve kopyadaki günlük bilgilerini sıfırlamak için bu işlevi kullanın.

- SKOPI

Kuyruk yöneticisi çevrimdışıyken bir kuyruğun içeriğini bir veri kümesine kopyalamak için bu işlevi kullanın.

- SDEFS

Kuyruk yöneticisi çevrimdışıyken nesnelere için bir tanımlama komutları kümesi üretmek için bu işlevi kullanın.

- YÜKLEME

Daha önceki bir COPY ya da SKOPI işleminin hedef veri kümesinden iletileri geri yüklemek için bu işlevi kullanın. SLOAD tek bir kuyruğu işler.

- Diğerine Geç

Küme gönderen kanallarıyla ilişkili iletim kuyruğunu değiştirmek ya da sorgulamak için bu işlevi kullanın.

z/OS

z/OS üzerinde Command Facility olanağının kullanılması

Kuyruk yöneticisine geçirecek MQSC komutlarını girmek ya da değiştirmek için düzenleyiciyi kullanın.

Komut Olanakı 'nı başlatmak için, CSQOPRIA birincil panosunda **8 Komut** seçeneğini belirleyin.

Size sıralı bir dosyanın düzenleme oturumu sunulur: *prefix.CSQUTIL.COMMANDS*; bkz. [IBM MQ komutlarını verme](#).

Komut öneki dizgisiyle (CPF) komutlara önek eklemeniz gerekmez.

Sonraki satırlarda MQSC komutlarına devam etmek için, yürürlükteki satırı + ya da -devam karakterleriyle sonlandırın. Diğer bir seçenek olarak, uzun MQSC komutları ya da komut içindeki uzun öznitelik değerlerinin değerlerini sağlamak için satır düzenleme kipini kullanın.

satır düzenleme

Satır düzenlemeyi kullanmak için, imleci düzenleme panosunda uygun satıra taşıyın ve kaydırılabilir bir panoda tek bir satır görüntülemek için **F4** seçeneğini kullanın. Tek bir satır en çok 32 760 bayt veri olabilir.

Satır düzenlemeyi bırakmak için:

- **F3 Çık** seçeneği, hat ve çıkışlarda yapılan değişiklikleri saklar
- **F12 cancel** (F12 cancel), satırda yapılan düzenleme panosu atılması değişikliklerine geri döner.

Düzenleme oturumunda yapılan değişiklikleri atmak için, **F12 İptal** düğmesini kullanarak dosyanın içeriğini değiştirmeden düzenleme oturumunu sonlandırın. Komutlar yürütülmez.

Komutlar yürütülüyor

MQSC komutlarını girmeyi tamamladığınızda, dosyanın içeriğini saklamak için **F3 çık** ile düzenleme oturumunu sonlandırın ve komutları kuyruk yöneticisine geçirmek için CSQUTIL ' i çağırın. Komut işleme çıkışı *prefix.CSQUTIL.OUTPUT*çıkışı. Yanıtları görüntüleyebilmemiz için bu dosyada otomatik olarak bir düzenleme oturumu açılır. Bu oturumdan çıkmak ve ana menüye dönmek için **F3 exit** düğmesine basın.

z/OS

z/OS üzerinde IBM MQ nesneleriyle çalışma

Bu belgede açıklanan görevlerin çoğu IBM MQ nesnelерinin işlenmesini içerir. Nesne tipleri, kuyruk yöneticileri, kuyruklar, süreç tanımlamaları, ad bilgileri, kanallar, istemci bağlantısı kanalları, dinleyiciler, hizmetler ve kimlik doğrulama bilgileri nesnelерidir.

- [Basit kuyruk nesnelерinin tanımlanması](#)
- [Diğer nesne tiplerini tanımlama](#)
- [Nesne tanımlamalarıyla çalışılması](#)
- [Ad listesi ile çalışma](#)

Basit kuyruk nesnelерinin tanımlanması

Yeni bir nesne tanımlamak için, var olan bir tanımlamayı temel olarak kullanın. Bunu üç şekilde yapabilirsiniz:

- İlk panoda belirlenen seçeneklerin sonucu olarak görüntülenen bir listenin üyesi olan bir nesneyi seçerek. Daha sonra işlem tipi 2 'yi girin (**Böyle tanımla**) Seçilen nesnenin yanındaki işlem alanında. Yeni nesneniz, yok etme dışında, seçilen nesnenin özniteliklerini içerir. Daha sonra, yeni nesnenizdeki öznitelikleri istediğiniz gibi değiştirebilirsiniz.
- İlk panoda **Benzer tanımla** işlem tipini seçin, **Nesne tipi** alanında tanımladığınız nesne tipini girin ve **Ad** alanına var olan belirli bir nesnenin adını girin. Yeni nesneniz, atma dışında, **Ad** alanında adlandırdığınız nesneyle aynı özniteliklere sahiptir. Daha sonra, yeni nesne tanımlamanızdaki öznitelikleri istediğiniz gibi değiştirebilirsiniz.
- **Benzer tanımla** işlem tipini seçerek, bir nesne tipi belirtin ve **Ad** alanını boş bırakın. Daha sonra yeni nesnenizi tanımlayabilirsiniz ve bu nesne, kuruluşunuz için tanımlanmış varsayılan özniteliklere sahiptir. Daha sonra, yeni nesne tanımlamanızdaki öznitelikleri istediğiniz gibi değiştirebilirsiniz.

Not: İlk panoda tanımlamakta olduğunuz nesnenin adını girmezsiniz, ancak size sunulan **Tanımla** panosunda girmeniz gerekir.

Aşağıdaki örnek, şablon olarak var olan bir kuyruğu kullanarak yerel bir kuyruğun nasıl tanımlanacağını göstermektedir.

Yerel kuyruk tanımlanması

İşlemler ve denetim panolarından bir yerel kuyruk nesnesi tanımlamak için, yeni tanımlamanıza temel olarak varolan bir kuyruk tanımlamasını kullanın. Tamamlanması gereken birkaç pano var. Tüm panoları tamamladığınızda ve özniteliklerin doğru olduğundan emin olduğunuzda, tanımınızı kuyruk yöneticisine göndermek için Enter tuşuna basın; daha sonra, gerçek kuyruk yaratılır.

İlk panoda ya da ilk panoda belirlenen seçeneklerin sonucu olarak görüntülenen listedeki bir nesne girişine karşı **Like** işlemini kullanın.

Örneğin, ilk panodan başlayarak şu alanlara veri girin:

Eylem	2 (Beğenmeyi tanımla)
Nesne tipi	QLOCAL
NAME	QUEUE.YOU.LIKE. Bu, yeni kuyruğunuza ilişkin öznelikler sağlayan kuyruğun adıdır.

Define a Local Queue (Yerel Kuyruk Tanımla) panosunu görüntülemek için Enter tuşuna basın. Yeni kuyruk için ad girmeniz için kuyruk adı alanı boş bırakılır. Tanımlama, bu yeni tanımlamayı temel almakta olduğunuz kuyruğun tanımlamasıdır. Bu alanı, yeni kuyruk için kendi tanımlamanızla birlikte yazın.

Diğer alanlardaki değerler, atma dışında, bu yeni kuyruğu temel almakta olduğunuz kuyruğun değerleridir. Bu alanları istediğiniz gibi yazabilirsiniz. Örneğin, uygun yetkili uygulamalar bu kuyruğa ileti yerleştirebiliyorsa, **Put enabled** (geçerli kılındı) alanına Y yazın (zaten Y değilse).

Alan yardımı almak için imleci bir alana taşıyın ve F1işlev tuşuna basın. Alan yardımı, her öznelik için kullanılacak değerlerle ilgili bilgi sağlar.

İlk panoyu tamamladığınızda, ikinci panoyu görüntülemek için F8 işlev tuşuna basın.

İpuçları:

1. Bu aşamada Enter tuşuna basmayın; tersi durumda, geri kalan alanları doldurmadan önce kuyruk yaratılır. (Önceden Enter tuşuna basarsanız, endişelenmeyin; tanımınızı daha sonra değiştirebilirsiniz.)
2. F3 ya da F12işlev tuşlarına basmayın, yoksa yazdığınız veriler kaybolur.

Tetikleyici tanımlaması, olay denetimi ve geriletme raporlama panoları da içinde olmak üzere diğer panoları görmek ve tamamlamak için F8 işlev tuşuna defalarca basın.

Yerel kuyruk tanımlamanız tamamlandığında

Tanımlamanız tamamlandığında, bilgileri işlenmek üzere kuyruk yöneticisine göndermek için Enter tuşuna basın. Kuyruk yöneticisi, belirttiğiniz tanımlamaya göre kuyruğu yaratır. Kuyruğun yaratılmasını istemiyorsanız, çıkmak ve tanımlamayı iptal etmek için F3 işlev tuşuna basın.

Diğer nesne tiplerini tanımlama

Diğer nesne tiplerini tanımlamak için, Yerel kuyruk tanımlama konusunda açıklandığı gibi, yeni tanımlamanıza temel olarak varolan bir tanımlamayı kullanın.

İlk panoda ya da ilk panoda belirlenen seçeneklerin sonucu olarak görüntülenen listedeki bir nesne girişine karşı **Like** işlemi kullanın.

Örneğin, ilk panodan başlayarak şu alanlara veri girin:

Eylem	2 (Beğenmeyi tanımla)
Nesne tipi	QALIAS, NAMELIST, PROCESS, CHANNEL VE DİĞER KAYNAK NESNELERİ.
NAME	Boş bırakın ya da aynı tipte var olan bir nesnenin adını girin.

İlgili DEFINE panolarını görüntülemek için Enter tuşuna basın. Alanları gerektiği gibi doldurun ve bilgileri kuyruk yöneticisine göndermek için yeniden Enter tuşuna basın.

Yerel bir kuyruk tanımlamak gibi, başka bir nesne tipi tanımlamak için genellikle birkaç panelin tamamlanması gerekir. Bir ad listesi tanımlamak için “Namelistlerle çalışma” sayfa 485 içinde açıklandığı gibi ek iş gerekir.

Nesne tanımlamalarıyla çalışılması

Bir nesne tanımlandığında, nesneyi değiştirmek, görüntülemek ya da yönetmek için **İşlem** alanında bir işlem belirtebilirsiniz.

Her bir durumda, aşağıdakilerden birini yapabilirsiniz:

- İlk panoda belirlenen seçeneklerin sonucu olarak görüntülenen listeden çalışmak istediğiniz nesneyi seçin. Örneğin, nesneleri görüntülemek için **İşlem** alanına 1 , **Nesne tipi** alanında Queue ve **Ad** alanında * girdikten sonra, sistemde tanımlı tüm kuyrukların bir listesi görüntülenir. Daha sonra, bu listeden çalışmanız gereken kuyruğu seçebilirsiniz.
- **Nesne tipi** ve **Ad** alanlarını doldurarak, çalıştığınız nesneyi belirttiğiniz başlangıç panosundan başlayın.

Nesne tanımlamasını değiştirme

Bir nesne tanımlamasını değiştirmek için işlem 3 'ü belirtin ve ALTER panolarını görmek için Enter tuşuna basın. Bu paneller DEFINE panellerine çok benzer. İsteddiğiniz değerleri değiştirebilirsiniz. Değişiklikleriniz tamamlandığında, bilgileri kuyruk yöneticisine göndermek için Enter tuşuna basın.

Nesne tanımının görüntülenmesi

Bir nesnenin ayrıntılarını değiştirmeden görmek istiyorsanız, işlem 1 'i belirtin ve DISPLAY panolarını görmek için Enter tuşuna basın. Bu panolar, alanlardan hiçbirini değiştiremeyeceğiniz dışında, DEFINE panolarına benzer. Başka bir nesnenin ayrıntılarını görüntülemek için nesne adını değiştirin.

Nesne silinmesi

Bir nesneyi silmek için, işlem 4 'ü (Yönet) belirtin ve **Sil** işlemi, sonuçtaki menüde gösterilen işlemlerden biridir. **Sil** işlemi seçin.

İsteğinizi onaylamanız isteniyor. F3 ya da F12işlev tuşuna basarsanız, istek iptal edilir. Enter tuşuna basarsanız, istek onaylanır ve kuyruk yöneticisine iletilir. Belirlediğiniz nesne daha sonra silinir.

Not: Kanal başlatıcısı başlatılmadıkça, kanal nesnesinin çoğu tipini silemezsiniz.

Namelistlerle çalışma

Ad listesi ile çalışırken, diğer nesnelere için olduğu gibi devam edin.

DEFINE BENZERİ ya da ALTER işlemlerinde, listeye ad eklemek ya da listedeki adları değiştirmek için F11 işlev tuşuna basın. Bu, ISPF düzenleyicisiyle çalışmayı içerir ve tüm olağan ISPF düzenleme komutları kullanılabilir. Ad alanına her bir adı ayrı bir satıra girin.

ISPF düzenleyicisini bu şekilde kullandığınızda, işlev tuşu ayarları olağan ISPF ayarlarıdır ve diğer işlemler ve denetim panoları tarafından kullanılanlar **değildir** .

Listede küçük harfli adlar belirtmeniz gerekiyorsa, düzenleyici panosu komut satırında CAPS (OFF) belirtin. Bunu yaptığınızda, CAPS (ON) belirtilinceye kadar gelecekte düzenlediğiniz tüm namelistler küçük harfli olur.

Ad listesi düzenlemeyi tamamladığınızda, ISPF düzenleme oturumunu sona erdirmek için F3 işlev tuşuna basın. Ardından değişiklikleri kuyruk yöneticisine göndermek için Enter tuşuna basın.

Dikkat: Bu aşamada Enter tuşuna değil, bunun yerine F3 işlev tuşuna basarsanız, yazdığınız güncellemeleri kaybedersiniz.

► z/OŞ

Sistemin birden çok küme iletim kuyruğu kullanılarak uygulanması

Kanalın tek bir kümede ya da çakışan bir kümede kullanılması fark etmez. Kanal seçilip başlatıldığında, kanal, tanımlamalara bağlı olarak iletim kuyruğunu seçer.

Yordam

- DEFCLXQ seçeneğini kullanıyorsanız, bkz. "[Kuyrukların ve anahtarların otomatik tanımının kullanılması](#)" sayfa 486.
- Aşamalı bir yaklaşım kullanıyorsanız, bkz. "[Aşamalı bir yaklaşım kullanarak küme gönderen kanallarınızı değiştirme](#)" sayfa 486.

Kuyrukların ve anahtarların otomatik tanımının kullanılması

DEFCLXQ seçeneğini kullanmayı planlıyorsanız bu seçeneği kullanın. Her kanal ve her yeni kanal için bir kuyruk oluşturulacak.

Yordam

1. SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE ve gerekiyorsa öznelikleri değiştirin.

Bu kuyruk, SCSQPROC (csq4insx) üyesinde tanımlanır.

2. SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE model kuyruğu.
3. Bu model kuyruğu ve SYSTEM.CLUSTER.TRANSMIT.* * kuyruklar.

z/OS için kanal başlatıcı tarafından başlatılan görev kullanıcı kimliğinin gerekleri:

- CLASS (MQADMIN) için erişimi denetle

```
ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channelname
```

- CLASS (MQQUEUE) için güncelleme erişimi

```
ssid.SYSTEM.CLUSTER.TRANSMIT.channelname
```

Aşamalı bir yaklaşım kullanarak küme gönderen kanallarınızı değiştirme

Aşamalı bir yaklaşım kullanmayı planlıyorsanız bu seçeneği kullanın. Bu süreç, işletmenizin gereksinimlerine uyacak şekilde çeşitli zamanlarda yeni küme gönderen kanallarına geçmenize olanak sağlar.

Başlamadan önce

- İş uygulamalarınızı ve hangi kanalların kullanıldığını belirleyin.
- Kullandığınız kuyruklar için, içinde buldukları kümeleri görüntüleyin.
- Bağlantı adlarını, uzak kuyruk yöneticilerinin adlarını ve kanalın desteklediği kümeleri gösteren kanalları görüntüleyin.

Bu görev hakkında

- Bir iletim kuyruğu yaratır. z/OS ' da kuyruk için hangi sayfa kümesini kullandığınızı göz önünde bulundurmak isteyebilirsiniz.
- Kuyruk için güvenlik ilkesi ayarlayın.
- Kuyruk izlemesini bu kuyruk adını içerecek şekilde değiştirin.
- Bu iletim kuyruğunu hangi kanalların kullanacağına karar verin. Kanallar benzer bir ada sahip olmalıdır, bu nedenle genel karakterler ' * ' CCLCHNAME içinde kanalı tanıtır.
- Yeni işlevi kullanmaya hazır olduğunuzda, iletim kuyruğunu, bu iletim kuyruğunu kullanacak kanalların adını belirtecek şekilde değiştirin. Örneğin, CLUSTER1.TOPARISya da CLUSTER1.* ya da *.TOPARIS
- Kanalları başlat

Yordam

1. Kümede tanımlı küme gönderen kanallarını görüntülemek için DIS CLUSQMGR(xxxx) XMITQ komutunu kullanın; burada xxxx uzak kuyruk yöneticisinin adıdır.
2. İletim kuyruğuna ilişkin güvenlik tanımını ayarlayın ve kuyruğa kanal başlatıcısına erişim yetkisi verin.

3. Kullanılacak iletim kuyruğunu tanımlayın ve USAGE (XMITQ) değerini belirleyin INDXTYPE (CORRELID) SHARE ve CLCHNAME (değer)

Kanal başlatıcı tarafından başlatılan görev kullanıcı kimliği için aşağıdaki erişim gereklidir:

```
alter class(MQADMIN) ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channel  
update class(MQQUEUE ssid.SYSTEM.CLUSTER.TRANSMIT.channel
```

Ve SWITCH komutunu kullanan kullanıcı kimliği için aşağıdaki erişim gerekir:

```
alter cl(MQADMIN) ssid.QUEUE.queueName
```

4. Kanalları durdurun ve yeniden başlatın.

Kanal değişikliği, kanal bir MQSC komutu kullanmaya başladığında ya da CSQUTIL kullandığınızda oluşur. Hangi kanalların yeniden başlatılması gerektiğini CSQUTIL SWITCH CHANNEL (*) STATUS ile belirleyebilirsiniz.

Kanal başlatıldığında sorun yaşıyorsanız, kanalı durdurun, sorunları çözün ve kanalı yeniden başlatın.

CLCHNAME özneteliğini gerek duyduğunuz sıklıkta değiştirebileceğinizi unutmayın.

Kanal başlatıldığında CLCHNAME değeri kullanılır; bu nedenle, kanal başlatıldığı andan itibaren tanımlamaları kullanmaya devam ederken CLCHNAME tanımını değiştirebilirsiniz. Kanal, yeniden başlatıldığında yeni tanımı kullanır.

z/OS ' da bir iletim kuyruğunda yapılan değişikliğin geri alınması bir)

Sonuçlar beklediğiniz gibi değilse, bir değişikliği geri almak için bir süreciniz olması gerekir.

Ne yanlış gidebilir ki?

Yeni iletim kuyruğu beklediğiniz değilse:

1. CLCHNAME ' in beklediğiniz gibi olup olmadığını denetleyin
2. Geçiş işleminin tamamının olup olmadığını denetlemek için iş günlüğünü gözden geçirin. Değilse, daha sonra kanalın yeni iletim kuyruğunu bekleyin ve denetleyin.

Birden çok küme iletim kuyruğu kullanıyorsanız, iletim kuyrukları tanımlarını açık olarak tasarlamamız ve karmaşık çakışma girebilirsiniz mi daha lar bu olarak? Bu şekilde, sorun varsa, özgün kuyruklara ve yapılandırmaya geri dönebildiğinizden emin olabilirsiniz.

Farklı bir iletim kuyruğunu kullanmaya taşıma sırasında sorunlarla karşılaşırsanız, değişikliğe devam etmeden önce sorunları çözmeniz gerekir.

Yeni bir değişiklik isteği yapılmadan önce var olan bir değişiklik isteği tamamlanmalıdır. Örneğin, siz:

1. En çok bir derinlik içeren yeni bir iletim kuyruğu tanımlayın ve gönderilmeyi bekleyen 10 ileti var.
2. İletim kuyruğunu, CLCHNAME parametresinde kanal adını belirtecek şekilde değiştirin.
3. Kanalı durdurun ve yeniden başlatın. İletileri taşıma girişimi başarısız olur ve sorunları bildirir.
4. İletim kuyruğundaki CLCHNAME parametresini boş olacak şekilde değiştirin.
5. Kanalı durdurun ve yeniden başlatın. Kanal, özgün isteği tamamlamaya devam eder ve kanal yeni iletim kuyruğunu kullanmaya devam eder.
6. İletilerin taşınmasının başarıyla tamamlanması için sorunların çözülmesi ve kanalı yeniden başlatılması gerekiyor.

Kanal yeniden başlatıldığında, tüm gün mine (değişiklik o seni) alır; CLCHNAME değerini boşluk olarak belirlediyseniz, kanal belirlenen iletim kuyruğunu kullanmaz.

Bu örnekte, iletim kuyruğundaki CLCHNAME ögesinin boşluk olarak değiştirilmesi, kanalın SYSTEM.CLUSTER.TRANSMIT kuyruğu. Örneğin, bir soysal ad ya da kuyruk yöneticisi özniteliği DEFCLXQ kanal olarak ayarlanabilir, bu nedenle kanal SYSTEM.CLUSTER.TRANSMIT KUYRUĞUNU İLETİN.

z/OS IBM MQ for z/OS yönetimi için programlar yazılıyor

Bir kuyruk yöneticisini denetlemek için kendi uygulama programlarınızı yazabilirsiniz. Kendi denetim programlarınızı yazmaya ilişkin gereksinimleri anlamak için bu konuyu kullanın.

Genel Programlama Arabirimi Bilgilerinin Başlatılması

Bu konu kümesi, bir IBM MQ uygulama programından IBM MQ komutlarını vermenize olanak sağlayacak ipuçları ve kılavuzları içerir.

Not: Bu konuda, MQI çağrıları C dili gösterimi kullanılarak açıklanır. COBOL, PL/I ve çevirici dillerindeki çağrıların tipik çağrıları için bkz. [İşlev çağrıları](#).

İşlerin nasıl yürüdüğünü anlamak

Anahatlarda, bir uygulama programından komut verme yordamı aşağıdaki gibidir:

1. IBM MQ komutunu, *istek iletisi* adı verilen bir IBM MQ iletisi tipinde oluşturun. Komut MQSC ya da PCF biçiminde olabilir.
2. Gönder (MQPUT komutunu kullanın) bu ileti, sistem komutu giriş kuyruğu adı verilen özel bir kuyruğa gönderilir. IBM MQ komut işlemcisi komutu çalıştırır.
3. Al (MQGET komutunu kullanın) yanıt kuyruğunda *yanıt iletileri* olarak komutun sonuçları. Bu iletiler, komutunuzun başarılı olup olmadığını ve başarılı olup olmadığını, sonuçların ne olduğunu belirlemek için gereksinim duyduğunuz kullanıcı iletilerini içerir.

Daha sonra, sonuçları işlemek uygulama programınıza kalmış olur.

Bu konu kümesi şunları içerir:

z/OS Denetim programları için kuyrukların hazırlanması

Denetim programları, sistem komutu girişi ve alma yanıtları için bir dizi önceden tanımlanmış kuyruk gerektirir.

Bu kısım, MQSC biçimindeki komutlar için geçerlidir. PCF ' deki eşdeğeri için bkz. [“IBM MQ Programlanabilir Komut Biçimlerini Kullanma” sayfa 26](#).

MQPUT ya da MQGET çağrılarını yayınlamadan önce, kullanacağınız kuyrukları tanımlamalısınız ve açmalısınız.

Sistem komutu giriş kuyruğunun tanımlanması

Sistem komutu giriş kuyruğu, SYSTEM.COMMAND.INPUTGIRIŞI. Sağlanan CSQINP2 kullanıma hazırlama veri kümesi thlqual.SCSQPROC(CSQ4INSG), sistem komutu giriş kuyruğu için varsayılan bir tanımlama içerir. Diğer platformlardaki IBM MQ ile uyumluluk için, bu kuyruğun diğer adı olan SYSTEM.ADMIN.COMMAND.QUEUE da sağlanır. Ek bilgi için bkz. [IBM MQ ile verilen örnek tanımlar](#).

Yanıt kuyruğunun tanımlanması

IBM MQ komut işlemcisinden yanıt iletileri almak için bir yanıt kuyruğu tanımlamanız gerekir. Bu, yanıt iletilerinin üzerine konmasına izin veren öznitelikleri olan herhangi bir kuyruk olabilir. Ancak, olağan işlem için şu öznitelikleri belirtin:

- KULLANIM (NORMAL)
- NOTRIGGER (uygulamanız tetikleme kullanmıyorsa)

Komutlar için kalıcı iletiler kullanmaktan kaçınınız, ancak bunu yapmayı seçerseniz, yanıt kuyruğunun geçici bir dinamik kuyruk olmaması gerekir.

Sağlanan CSQINP2 kullanıma hazırlama veri kümesi thlqual.SCSQPROC(CSQ4INSG), SYSTEM.COMMAND.REPLY.MODEL. Dinamik bir yanıt kuyruğu yaratmak için bu modeli kullanabilirsiniz.

Not: Komut işlemcisi tarafından oluşturulan yanıtlar en çok 15 000 bayt uzunluğunda olabilir.

Yanıt kuyruğu olarak kalıcı bir dinamik kuyruk kullanırsanız, uygulamanız kuyruğu silme girişiminde bulunmadan önce tüm PUT ve GET işlemlerinin tamamlanmasına izin vermelidir; tersi durumda, MQR2055 (MQR_Q_NOT_EMPTY) döndürülebilir. Bu durumda, birkaç saniye sonra kuyruk silme işlemini yeniden deneyin.

Sistem komutu giriş kuyruğunun açılması

Sistem komutu giriş kuyruğunu açabilmeniz için önce uygulama programınızın kuyruk yöneticimize bağlı olması gerekir. Bunu yapmak için MQI çağrısı MQCONN ya da MQCONNX ' i kullanın.

Daha sonra, sistem komutu giriş kuyruğunu açmak için MQI çağrısı MQOPEN ' i kullanın. Bu aramayı kullanmak için:

1. **Options** değiştirgesini MQOO_OUTPUT olarak ayarlayın
2. MQOD nesnesi tanımlayıcı alanlarını aşağıdaki gibi ayarlayın:

ObjectType

MQOT_Q (nesne bir kuyruktur)

ObjectName

SYSTEM.COMMAND.INPUT

ObjectQMgrName

İstek iletilerinizi yerel kuyruk yöneticimize göndermek istiyorsanız, bu alanı boş bırakın. Bu, komutlarınızın yerel olarak işlendiği anlamına gelir.

IBM MQ komutlarınızın uzak bir kuyruk yöneticisinde işlenmesini istiyorsanız, adını buraya yazın. Ayrıca, [Dağıtılmış kuyruğa alma ve kümeler](#) konusunda açıklandığı gibi doğru kuyruklara ve bağlantılara sahip olmanız gerekir.

Yanıt kuyruğunun açılması

Yanıtları bir IBM MQ komutundan almak için bir yanıt kuyruğu açmanız gerekir. Bunu yapmanın bir yolu, model kuyruğunun (bir MQOPEN çağrısında SYSTEM.COMMAND.REPLY.MODEL), yanıt kuyruğu olarak kalıcı bir dinamik kuyruk yaratılmasıdır. Bu aramayı kullanmak için:

1. **Options** değiştirgesini MQOO_INPUT_SHARED olarak ayarlayın
2. MQOD nesnesi tanımlayıcı alanlarını aşağıdaki gibi ayarlayın:

ObjectType

MQOT_Q (nesne bir kuyruktur)

ObjectName

Yanıt kuyruğunun adı. Belirttiğiniz kuyruk adı bir model kuyruğu nesnesinin adıyla, kuyruk yöneticisi dinamik bir kuyruk yaratır.

ObjectQMgrName

Yerel kuyruk yöneticinizden yanıt almak için bu alanı boş bırakın.

DynamicQName

Yaratılacak dinamik kuyruğun adını belirleyin.

Komut sunucusunun kullanılması

Komut sunucusu, komut işlemcisi bileşeniyle çalışan bir IBM MQ bileşenidir. İletileri yorumlayan, yönetim isteklerini çalıştıran ve yanıtları yönetim uygulamanıza geri gönderen komut sunucusuna biçimlendirilmiş iletiler gönderebilirsiniz.

Komut sunucusu, sistem komutu giriş kuyruğundan istek iletilerini okur, bunları doğrular ve geçerli iletileri komut işlemcisine komut olarak iletir. Komut işlemcisi, komutları işler ve yanıtları, belirlediğiniz yanıt kuyruğuna yanıt iletileri olarak yerleştirir. İlk yanıt iletileri CSQN205I kullanıcı iletilerini içerir. Ek bilgi için bkz. [“Komut sunucusundan yanıt iletilerinin yorumlanması” sayfa 493](#) . Komut sunucusu, kanal başlatıcı ve kuyruk paylaşım grubu komutlarını da, nereden verilse, işler.

Komutlarınızı işleyen kuyruk yöneticisinin tanımlanması

Bir denetim programından çıkardığınız komutları işleyen kuyruk yöneticisi, iletinin yerleştirildiği sistem komutu giriş kuyruğunun iyesi olan kuyruk yöneticisidir.

Komut sunucusunun başlatılması

Olağan durumda, kuyruk yöneticisi başlatıldığında komut sunucusu otomatik olarak başlatılır. START QMGR komutundan CSQ9022I 'START QMGR' NORMAL TAMAMLANMA iletileri döndürülür döndürülmez kullanıma sunulur. Sistem sona erdirme aşamasında bağlı tüm görevlerin bağlantısı kesildiğinde komut sunucusu durdurulur.

START CMDSERV ve STOP CMDSERV komutlarını kullanarak komut sunucusunu kendiniz denetleyebilirsiniz. IBM MQ yeniden başlatıldığında komut sunucusunun otomatik olarak başlatılmasını önlemek için, CSQINP1 ya da CSQINP2 kullanıma hazırlama veri kümelerinize bir STOP CMDSERV komutu ekleyebilirsiniz. Ancak, herhangi bir kanal başlatıcısı ya da kuyruk paylaşım grubu komutlarının işlenmesini önlediği için bu önerilmez.

STOP CMDSERV komutu, yürürlükteki iletiyi işlemeyi bitirir bitirmez ya da işlenmekte olan bir ileti yoksa, komut sunucusunu durdurur.

Komut sunucusu programdaki bir STOP CMDSERV komutuyla durdurulduysa, programdan başka komut işlenemez. Komut sunucusunu yeniden başlatmak için, z/OS konsolundan bir START CMDSERV komutu vermeniz gerekir.

Kuyruk yöneticisi çalışırken komut sunucusunu durdurur ve yeniden başlatırsanız, komut sunucusu yeniden başlatıldığında sistem komutu giriş kuyruğunda bulunan tüm iletiler işlenir. Ancak, komut sunucusu durdurulduktan sonra kuyruk yöneticisini durdurup yeniden başlattığınızda, yalnızca sistem komutu giriş kuyruğundaki kalıcı iletiler komut sunucusu yeniden başlatıldığında işlenir. Sistem komutu giriş kuyruğundaki tüm kalıcı olmayan iletiler kaybolur.

Komut sunucusuna komut gönderilmesi

Her komut için, komutu içeren bir ileti oluşturup sistem komutu giriş kuyruğuna koyarsınız.

IBM MQ komutlarını içeren bir ileti oluşturma

Gerekli komutları içeren istek iletileri oluşturarak bir uygulama programında IBM MQ komutlarını birleştirebilirsiniz. Bu tür komutların her biri için:

1. Komutu gösteren bir karakter dizisi içeren bir arabellek yaratın.
2. Çağrıya ilişkin **buffer** değiştirgesinde arabellek adını belirterek bir MQPUT çağrısı yayınlayın.

Bunu C' de yapmanın en basit yolu 'char' kullanarak bir arabellek tanımlamaktır. Örneğin:

```
char message_buffer[ ] = "ALTER QLOCAL(SALES) PUT(ENABLED)";
```

Bir komut oluşturduğunuzda, boş sonlandırılmış bir karakter dizisi kullanın. Bu şekilde tanımlanan bir komutun başında bir komut öneki dizisi (CPF) belirtmeyin. Bu, komut dosyalarını başka bir kuyruk yöneticisinde çalıştırmak istiyorsanız değiştirmeniz gerekmediği anlamına gelir. Ancak, yanıt kuyruğuna konan yanıt iletilerine bir CPF eklendiğini göz önünde bulundurmanız gerekir.

Komut sunucusu, tırnak işareti içinde değilse, tüm küçük harfleri büyük harfe katlar.

Komutlar en çok 32 762 karakter uzunluğunda olabilir.

İletilerin sistem komutu giriş kuyruğuna konması

Sistem komutu giriş kuyruğuna komut içeren istek iletilerini koymak için MQPUT çağrısı kullanın. Bu çağrıda, önceden açtığınız yanıt kuyruğunun adını belirtirsiniz.

MQPUT çağrısı kullanmak için:

1. Bu MQPUT deęiřtirgelerini ayarlayın:

Hconn

MQCONN ya da MQCONNX çağrısıyla döndürölen baęlantı tanıtıcısı.

Hobj

Sistem komutu giriş kuyruęu için MQOPEN çağrısıyla döndürölen nesne tanıtıcısı.

BufferLength

Biçimlenmiř komutun uzunluęu.

Buffer

Komutu içeren arabelleęin adı.

2. Bu MQMD alanlarını ayarla:

MsgType

MQMT_REQUEST

Format

MQFMT_STRING ya da MQFMT_NONE

Kuyruk yöneticisiyle aynı kod sayfasını kullanmıyorsanız, *CodedCharSetId* deęerini uygun olarak ayarlayın ve MQFMT_STRING deęerini ayarlayın; böylece komut sunucusu iletiyi dönüřtürebilir. MQFMT_ADMIN deęerini ayarlamayın; bu, komutunuzun PCF olarak yorumlanmasına neden olur.

ReplyToQ

Yanıt kuyruęunuzun adı.

ReplyToQMGr

Yanıtların yerel kuyruk yöneticinize gönderilmesini istiyorsanız, bu alanı boş bırakın. IBM MQ komutlarınızın uzak bir kuyruk yöneticisine gönderilmesini istiyorsanız, adını buraya yazın. Ayrıca, Daęıtılmıř kuyruęa alma ve kümeler konusunda açıklandığı gibi doęru kuyruklara ve baęlantılara sahip olmanız gerekir.

3. Dięer MQMD alanlarını gerektięi gibi ayarlayın. Normalde komutlar için kalıcı olmayan iletiler kullanmalısınız.
4. *PutMsgOpts* seçeneklerini gerektięi gibi ayarlayın.

MQPMO_SYNCPOINT belirtirseniz (varsayılan), bir eřitleme noktası çağrısıyla MQPUT çağrısı izlemelisiniz.

MQPUT1 ve sistem komutu giriş kuyruęunun kullanılması

Sistem komutu giriş kuyruęuna tek bir ileti koymak istiyorsanız, **MQPUT1** çağrısı kullanabilirsiniz.

Bu çağrı, bir **MQOPEN**' in iřlevlerini birleřtirir, ardından bir iletinin **MQPUT** , ardından bir **MQCLOSE** ve ardından bir çağrı gelir. Bu çağrısı kullanırsanız, parametreleri uygun řekilde deęiřtirin. Ayrıntılar için MQPUT1 çağrısı kullanılarak kuyruęa bir ileti yerleřtirilmesi bařlıklı konuya bakın.

Komutlarınıza verilen yanıtlar alınıyor

Komut sunucusu, aldıęı her istek ileti için yanıt kuyruęuna bir yanıt gönderir. Her yönetim uygulaması yanıt iletilerini almalı ve iřlemelidir.

Komut iřlemcisi komutlarınızı iřlerken, yanıt iletileri MQPUT çağrısında belirtilen yanıt kuyruęuna yerleřtirilir. Komut sunucusu, yanıt iletilerini aldıęı komut iletiyle aynı kalıcılıkla gönderir.

Yanıt bekleniyor.

İstek iletinizden bir yanıt almak için MQGET çağrısına bakın. Bir istek iletisi birden çok yanıt iletisi üretebilir. Ayrıntılar için bkz. [“Komut sunucusundan yanıt iletilerinin yorumlanması” sayfa 493.](#)

Bir MQGET çağrısına ilişkin bir yanıt iletisinin oluşturulmasını bekleyeceği zaman aralığını belirtebilirsiniz. Yanıt almazsanız, [“Yanıt almazsanız” sayfa 494](#) konusunun başındaki denetim listesini kullanın.

MQGET çağrısı kullanmak için:

1. Şu parametreleri ayarlayın:

Hconn

MQCONN ya da MQCONNX çağrısıyla döndürülen bağlantı tanıtıcısı.

Hobj

Yanıt kuyruğu için MQOPEN çağrısıyla döndürülen nesne tanıtıcısı.

Buffer

Yanıtın gönderileceği alanın adı.

BufferLength

Yanıtı alacak arabelleğin uzunluğu. Bu değer en az 80 byte olmalıdır.

2. Yalnızca verdiğiniz komuttan yanıt aldığınızdan emin olmak için uygun *MsgId* ve *CorrelId* alanlarını belirtmeniz gerekir. Bunlar, MQPUT çağrısında belirttiğiniz rapor seçeneklerine (MQMD_REPORT) bağlıdır:

MQRO_NONE

İkili sıfır, '00 ... 00' (24 boş değer).

MQRO_NEW_MSG_ID

İkili sıfır, '00 ... 00' (24 boş değer).

Bu seçeneklerden hiçbiri belirtilmediyse, varsayılan değer budur.

MQRO_PASS_MSG_ID

MQPUT ürünündeki *MsgId* .

MQRO_NONE

MQPUT çağrısından *MsgId* .

MQRO_COPY_MSG_ID_TO_CORREL_ID

MQPUT çağrısından *MsgId* .

Bu seçeneklerden hiçbiri belirtilmediyse, varsayılan değer budur.

MQRO_PASS_CORREL_ID

MQPUT çağrısından *CorrelId* .

Rapor seçenekleriyle ilgili daha fazla ayrıntı için bkz. [Rapor seçenekleri ve ileti işaretleri.](#)

3. Aşağıdaki *GetMsgOpts* alanlarını ayarlayın:

Options

MQGMO_WAIT

Kuyruk yöneticisiyle aynı kod sayfasını kullanmıyorsanız, MQGMO_CONVERT ayarını yapın ve *CodedCharSetId* değerini MQMD ' de uygun olarak ayarlayın.

WaitInterval

Yerel kuyruk yöneticisinden gelen yanıtlar için 5 saniye deneyin. Milisaniye cinsinden kodlanmış, bu 5.000 olur. Uzak kuyruk yöneticisinden gelen yanıtlar ve kanal denetimi ve durum komutları için 30 saniye deneyin. Milisaniye olarak kodlanınca, bu 30.000 olur.

Atılan iletiler

Komut sunucusu bir istek iletisinin geçerli olmadığını saparsa, bu iletiyi atar ve adı belirtilen yanıt kuyruğuna CSQN205I iletisini yazar. Yanıt kuyruğu yoksa, CSQN205I iletisi gitmeyen ileti kuyruğuna yerleştirilir. Bu iletideki dönüş kodu, özgün istek iletisinin neden geçerli olmadığını gösterir:

00D5020F	MQMT_REQUEST tipinde değil.
00D50210	Uzunluğu sıfır.
00D50212	32 762 bayttan uzun.
00D50211	İçinde boşluklar var.
00D5483E	Dönüştürülmesi gerekiyordu, ancak <i>Format</i> MQFMT_STRING değildi.
Diğer	Bkz. Komut sunucusu kodları

Komut sunucusu yanıt iletisi tanımlayıcısı

Herhangi bir yanıt iletisi için aşağıdaki MQMD ileti tanımlayıcısı alanları ayarlanır:

<i>MsgType</i>	MQMT_REPLY
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQENC_NATIVE
<i>Priority</i>	Yayınladığınız iletideki MQMD ' ye gelince.
<i>Persistence</i>	Yayınladığınız iletideki MQMD ' ye gelince.
<i>CorrelId</i>	MQPUT rapor seçeneklerine bağlıdır.
<i>ReplyToQ</i>	Yok.

Komut sunucusu, MQPMO yapısının *Options* alanını MQPMO_NO_SYNCPOINT olarak ayarlar. Bu, yanıtları sonraki eşitleme noktasında bir grup olarak değil, oluşturuldukları şekilde alabileceğiniz anlamına gelir.

Komut sunucusundan yanıt iletilerinin yorumlanması

IBM MQ tarafından doğru olarak işlenen her istek iletisi en az iki yanıt iletisi üretir. Her yanıt iletisi tek bir IBM MQ kullanıcı iletisi içerir.

Yanıtın uzunluğu, verilen komuta bağlıdır. Alabileceğiniz en uzun yanıt bir **DISPLAY NAMELIST** komutundan alınmaktadır ve bu, en çok 15 000 bayt uzunluğunda olabilir.

İlk kullanıcı iletisi (CSQN205I) her zaman şunları içerir:

- Yanıtların geri kalanını almak için bir döngüde sayaç olarak kullanabileceğiniz yanıtların sayısı (ondalık olarak). Sayım bu ilk iletiyi içerir.
- Komut ön işlemcisinin dönüş kodu.
- Komut işlemcisinin neden kodu olan bir neden kodu.

Bu ileti bir CPF içermiyor.

Örneğin:

```
CSQN205I    COUNT=    4, RETURN=0000000C, REASON=00000008
```

COUNT alanı 8 bayt uzunluğudur ve sağa yaslıdır. Her zaman 18. konumdan başlar, yani COUNT=' den hemen sonra. RETURN alanı, onaltılı karakter olarak 8 bayt uzunluğundadır ve 35 konumunda RETURN=' den hemen sonradır. REASON alanı, onaltılı karakter olarak 8 bayt uzunluğundadır ve 52 konumunda REASON=' den hemen sonradır.

RETURN= deęeri 00000000 ve REASON= deęeri 00000004 ise, yanıt iletileri kümesi eksik olur. CSQN205I iletileriyle belirtilen yanıtları aldıktan sonra, başka bir yanıt kümesi beklemek için başka bir MQGET çağırısı yayınlayın. Sonraki yanıt kümesindeki ilk ileti yeniden CSQN205I olur; bu ileti, kaç yanıt olduğunu ve daha fazla yanıt olup olmadığını gösterir.

Tek tek iletilere ilişkin daha fazla ayrıntı için [IBM MQ for z/OS iletileri, tamamlama ve neden kodları belgelerine](#) bakın.

İngilizce olmayan bir dil özellięi kullanıyorsanız, yanıtların metni ve düzeni burada gösterilenlerden farklıdır. Ancak, CSQN205I iletilerindeki sayı ve dönüş kodlarının boyutu ve konumu aynıdır.

z/OS Yanıt almazsanız

Komut sunucusundan istekte bulunmak için yanıt almazsanız, yapabileceğiniz bir dizi adım vardır.

İstek iletinize yanıt almazsanız, aşağıdaki denetim listesiyle çalışın:

- Komut sunucusu çalışıyor mu?
- *WaitInterval* yeterince uzun mu?
- Sistem komutu giriş ve yanıt kuyrukları doğru tanımlandı mı?
- Bu kuyruklara yapılan MQOPEN çağrıları başarılı oldu mu?
- MQPUT ve MQGET çağrıları için hem sistem komutu giriş, hem de yanıt kuyrukları etkin mi?
- Kuyruklarınızın MAXDEPTH ve MAXMSGL özniteliklerini artırmaya dikkat ettin mi?
- *CorrelId* ve *MsgId* alanlarını doğru kullanıyor musunuz?
- Kuyruk yöneticisi hala çalışıyor mu?
- Komut doğru bir şekilde inşa edildi mi?
- Tüm uzak bağlantılarınız doğru şekilde tanımlandı ve çalışıyor mu?
- MQPUT çağrıları doğru tanımlandı mı?
- Yanıt kuyruęu, kalıcı dinamik kuyruk yerine geçici bir dinamik kuyruk olarak tanımlandı mı? (İstek iletili kalıcıysa, yanıt için kalıcı bir dinamik kuyruk kullanmanız gerekir.)

Komut sunucusu yanıtları oluşturduğunda, ancak bunları belirlediğiniz yanıt kuyruęuna yazmadığında, bunları ileti kuyruęuna yazar.

z/OS Komutları MGCRE kullanarak aktarma

Uygun yetkiyle bir uygulama programı, bir z/OS hizmet yordamını kullanarak birden çok kuyruk yöneticisiyle istekte bulunabilir.

Doęru yetkiye sahipseniz, MGCRE (SVC 34) z/OS hizmeti tarafından programınızdan birden çok kuyruk yöneticilerine IBM MQ komutlarını geçirebilirsiniz. Ek bilgi için [z/OS MVS Programming: Authorized Assembler Services Guide](#) belgesine bakın.

CPF deęeri, komutun yöneltildięi kuyruk yöneticisini tanıtır. CPF ' ler hakkında bilgi için [Komut güvenlięi ve komut kaynaęı güvenlięi için kullanıcı kimlikleri ve "z/OS üzerinde kuyruk yöneticisi komutları verilmesi" sayfa 457](#) başlıklı konuya bakın.

MGCRE kullanıyorsanız, komuta doğrudan yanıt almak için bir Komut ve Yanıt Simgesi (CART) kullanabilirsiniz.

z/OS Komut örnekleri ve yanıtları

Komut sunucusuna ve komut sunucusundan gelen yanıtlara örnek olarak bu konuyu kullanın.

Aşaęıda, IBM MQ iletilerinde oluşturulabilecek bazı komut örnekleri ve yanıtlar olan kullanıcı iletileri yer almaktadır. Aksi belirtilmedikçe, yanıtın her satırı ayrı bir iletidir.

- [DEFINE komutundan gelen iletiler](#)
- [DELETE komutundan gelen iletiler](#)

- DISPLAY komutlarından gelen iletiler
- CMDSCOPE içeren komutlardan gelen iletiler
- CMDSCOPE ile komut oluşturan komutlardan gelen iletiler

DEFINE komutundan gelen iletiler

Komut:

```
DEFINE QLOCAL(Q1)
```

şu iletileri üretir:

```
CSQN205I    COUNT=    2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQMMSGP ' DEFINE QLOCAL' NORMAL COMPLETION
```

Bu yanıt iletileri olağan tamamlanınca üretilir.

DELETE komutundan gelen iletiler

Komut:

```
DELETE QLOCAL(Q2)
```

şu iletileri üretir:

```
CSQN205I    COUNT=    4, RETURN=0000000C, REASON=00000008
CSQM125I +CSQ1 CSQMUQLC QLOCAL (Q2) QSGDISP(QMGR) WAS NOT FOUND
CSQM090E +CSQ1 CSQMUQLC FAILURE REASON CODE X'00D44002'
CSQ9023E +CSQ1 CSQMUQLC ' DELETE QLOCAL' ABNORMAL COMPLETION
```

Bu iletiler, Q2 adlı yerel bir kuyruğun var olmadığını gösterir.

DISPLAY komutlarından gelen iletiler

Aşağıdaki örneklerde, bazı DISPLAY komutlarındaki yanıtlar gösterilir.

Gitmeyen iletiler kuyruğunun adını bulma

Bir kuyruk yöneticisine ilişkin gitmeyen iletiler kuyruğunun adını öğrenmek istiyorsanız, bu komutu bir uygulama programından verin:

```
DISPLAY QMGR DEADQ
```

Gerekli adı alabileceğiniz aşağıdaki üç kullanıcı iletisi döndürülür:

```
CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000
CSQM409I +CSQ1 QMNAME(CSQ1) DEADQ(SYSTEM.DEAD.QUEUE      )
CSQ9022I +CSQ1 CSQMDRTS ' DISPLAY QMGR' NORMAL COMPLETION
```

DISPLAY QUEUE komutundan gelen iletiler

Aşağıdaki örneklerde, bir komutun sonuçlarının o komutta belirtilen özniteliklere nasıl bağlı olduğu gösterilmiştir.

Örnek 1

Aşağıdaki komutu kullanarak bir yerel kuyruk tanımlarsınız:

```
DEFINE QLOCAL(Q1) DESCR('A sample queue') GET(ENABLED) SHARE
```

Bir uygulama programından aşağıdaki komutu çalıştırırsanız:

```
DISPLAY QUEUE(Q1) SHARE GET DESCR
```

şu üç kullanıcı ileti döndürülür:

```
CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(Q1) TYPE(
QLOCAL ) QSGDISP(QMGR )
DESCR(A sample queue
) SHARE GET(ENABLED )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION
```

Not: İkinci ileti (CSQM401I) burada dört satırı kaplayan şekilde gösterilir.

Örnek 2

İki kuyruğunun A harfiyle başlayan adları vardır:

- A1 , PUT özneliği DISABLED olarak ayarlanmış yerel bir kuyruktur.
- A2 , PUT özneliği ENABLED olarak ayarlanmış bir uzak kuyruktur.

Bir uygulama programından aşağıdaki komutu çalıştırırsanız:

```
DISPLAY QUEUE(A*) PUT
```

şu dört kullanıcı ileti döndürülür:

```
CSQN205I    COUNT=    4, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(A1) TYPE(
QLOCAL ) QSGDISP(QMGR )
PUT(DISABLED )
CSQM406I +CSQ1 QUEUE(A2) TYPE(
QREMOTE ) PUT(ENABLED )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION
```

Not: İkinci ve üçüncü iletiler (CSQM401I ve CSQM406I) burada üç ve iki satırı kaplar.

DISPLAY NAMELIST komutundan gelen iletiler

Aşağıdaki komutu kullanarak bir ad listesi tanımlarsınız:


```
DEFINE NAMELIST(N1) NAMES(Q1,SAMPLE_QUEUE)
```

Bir uygulama programından aşağıdaki komutu çalıştırırsanız:

```
DISPLAY NAMELIST(N1) NAMES NAMCOUNT
```

Aşağıdaki üç kullanıcı iletisi döndürülür:

```
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000000
CSQM407I +CSQ1 NAMELIST(N1 ) QS
GDISP(QMGR ) NAMCOUNT( 2) NAMES(Q1
,SAMPLE_QUEUE )
CSQ9022I +CSQ1 CSQMDMSG ' DISPLAY NAMELIST' NORMAL COMPLETION
```

Not: İkinci ileti (CSQM407I) burada üç satırı kaplıyor.

CMDScope içeren komutlardan gelen ileteler

Aşağıdaki örnekler, CMDSCOPE özniteliğiyle girilen komutlardan gelen yanıtları göstermektedir.

ALTER PROCESS komutundan gelen ileteler

Komut:

```
ALT PRO(V4) CMDSCOPE(*)
```

aşağıdaki ileteleri üretir:

```
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'ALT PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ26
CSQM125I !MQ26 CSQMMSGP PROCESS(V4) QSGDISP(QMGR) WAS NOT FOUND
CSQM090E !MQ26 CSQMMSGP FAILURE REASON CODE X'00D44002'
CSQ9023E !MQ26 CSQMMSGP ' ALT PRO' ABNORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP ' ALT PRO' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000008
CSQN123E !MQ25 'ALT PRO' command for CMDSCOPE(*) abnormal completion
```

Bu ileteler, komutun MQ25 kuyruk yöneticisine girildiğini ve iki kuyruk yöneticisine (MQ25 ve MQ26) gönderildiğini belirtir. Komut MQ25 'te başarılı oldu, ancak süreç tanımlaması MQ26' da yok, bu nedenle komut o kuyruk yöneticisinde başarısız oldu.

DISPLAY PROCESS komutundan gelen ileteler

Komut:

```
DIS PRO(V*) CMDSCOPE(*)
```

aşağıdaki ileteleri üretir:

```

CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ26
CSQM408I !MQ26 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ26 PROCESS(V3) QSGDISP(QMGR)
CSQ9022I !MQ26 CSQMDRTS ' DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 7, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ25
CSQM408I !MQ25 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ25 PROCESS(V2) QSGDISP(GROUP)
CSQM408I !MQ25 PROCESS(V3) QSGDISP(QMGR)
CSQM408I !MQ25 PROCESS(V4) QSGDISP(QMGR)
CSQ9022I !MQ25 CSQMDRTS ' DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS PRO' command for CMDSCOPE(*) normal completion

```

Bu iletiler, komutun MQ25 kuyruk yöneticisine girildiğini ve iki kuyruk yöneticisine (MQ25 ve MQ26) gönderildiğini belirtir. Her kuyruk yöneticisindeki tüm işlemlere ilişkin bilgiler V harfiyle başlayan adlarla görüntülenir.

DISPLAY CHSTATUS komutundaki iletiler

Komut:

```
DIS CHS(VT) CMDSCOPE(*)
```

aşağıdaki iletileri üretir:

```

CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS CHS' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ25
CSQM422I !MQ25 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ25 CSQXDRTS ' DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ26
CSQM422I !MQ26 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ26 CSQXDRTS ' DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS CHS' command for CMDSCOPE(*) normal completion

```

Bu iletiler, komutun MQ25 kuyruk yöneticisine girildiğini ve iki kuyruk yöneticisine (MQ25 ve MQ26) gönderildiğini belirtir. Her kuyruk yöneticisinde kanal durumuyla ilgili bilgiler görüntülenir.

STOP CHANNEL komutundan gelen iletiler

Komut:

```
STOP CHL(VT) CMDSCOPE(*)
```

şu iletileri üretir:

```
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'STOP CHL' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQM134I !MQ25 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
SQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQM134I !MQ26 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQ9022I !MQ26 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQ9022I !MQ25 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'STOP CHL' command for CMDSCOPE(*) normal completion
```

Bu iletiler, komutun MQ25 kuyruk yöneticisine girildiğini ve iki kuyruk yöneticisine (MQ25 ve MQ26) gönderildiğini belirtir. Kanal VT her kuyruk yöneticisinde durduruldu.

CMDScope ile komut oluşturan komutlardan gelen iletiler

Komut:

```
DEF PRO(V2) QSGDISP(GROUP)
```

şu iletileri üretir:

```
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQM122I !MQ25 CSQMMSGP ' DEF PRO' COMPLETED FOR QSGDISP(GROUP)
CSQN138I !MQ25 'DEFINE PRO' command generated for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ26
CSQ9022I !MQ26 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DEFINE PRO' command for CMDSCOPE(*) normal completion
```

Bu iletiler, komutun MQ25 kuyruk yöneticisine girildiğini belirtir. Nesne paylaşılan havuzda yaratıldığında, başka bir komut oluşturuldu ve kuyruk paylaşım grubundaki tüm etkin kuyruk yöneticilerine gönderildi (MQ25 ve MQ26).

z/OS

z/OS üzerinde IBM MQ kaynaklarının yönetilmesi

IBM MQ for z/OS tarafından kullanılan kaynakların (örneğin, günlük dosyalarının, veri kümelerinin, sayfa kümelerinin, arabellek havuzlarının ve bağlaşım olanağı yapılarının yönetilmesi) nasıl yönetileceğini öğrenmek için bu konudaki bağlantıları kullanın.

IBM MQ for z/OS kullanırken tamamlamanız gereken farklı yönetim görevlerine ilişkin ayrıntılar için aşağıdaki bağlantıları kullanın:

- [“Günlüklerin yönetilmesi” sayfa 500](#)
- [“Önyükleme veri kümesinin \(BSDS\) yönetilmesi” sayfa 509](#)
- [“Sayfa kümelerini yönetme” sayfa 517](#)
- [“Sayfa kümelerinin yedeklanması ve kurtarılması” sayfa 524](#)
- [“CSQUTIL kullanılarak kuyrukların yedeklenmesi ve geri yüklenmesi” sayfa 528](#)

- “Arabellek havuzlarının yönetilmesi” sayfa 528
- “z/OS üzerinde kuyruk paylaşım gruplarının ve paylaşılan kuyrukların yönetilmesi” sayfa 529

İlgili kavramlar

IBM MQ for z/OS kavramlar

“YönetmeIBM MQ for z/OS” sayfa 456

IBM MQ for z/OS , MQSC ve PCF komutları, ürünle birlikte sağlanan yardımcı programlar ve programlar kümesi ve yetkili uygulamalar tarafından denetlenebilir ve yönetilebilir.

“IBM MQ for z/OS üzerinde MQSC ve PCF komutlarını yayınlayabileceğiniz kaynaklar” sayfa 457

IBM MQ for z/OS konsolundan, kullanıma hazırlama giriş veri kümelerinden, toplu iş yardımcı programından ya da yetkili uygulamalardan MQSC ve PCF komutları verebilirsiniz. Tüm komutlar tüm bu kaynaklardan verilemez.

“z/OS üzerinde kurtarma ve yeniden başlatma” sayfa 539

IBM MQ tarafından kullanılan kurtarma ve yeniden başlatma mekanizmalarını anlamak için bu konuyu kullanın.

İlgili görevler

z/OS üzerinde IBM MQ ortamınızın planlanması

z/OS üzerinde kuyruk yöneticilerini yapılandırma

IBM MQ yardımcı programlar da cı larda bakın unun larda unun unun z/OS

İlgili başvurular

“IBM MQ for z/OS yardımcı programlarının kullanılması” sayfa 479

IBM MQ for z/OS , sistem denetimiyle ilgili yardım almak için kullanabileceğiniz bir dizi yardımcı program sağlar.

[Programlanabilir komut biçimleri başvurusu](#)

z/OS Günlüklerin yönetilmesi

Günlük arşivleme işlemi, günlük kaydı sıkıştırması, günlük kaydı kurtarma ve yazdırma günlük kayıtları da içinde olmak üzere IBM MQ günlük dosyalarınızı nasıl yöneteceğinizi anlamak için bu konuyu kullanın.

Bu konuda, IBM MQ günlüklerinin yönetilmesiyle ilgili görevler açıklanmaktadır. Aşağıdaki kısımları içerir:

z/OS Günlükleri ARCHIVE LOG komutuyla arşivleme

Yetkili bir işletmen, **ARCHIVE LOG** komutunu kullanarak gerektiğinde geçerli IBM MQ etkin günlük veri kümelerini arşivleyebilir.

ARCHIVE LOG komutunu verdiğinizde, IBM MQ yürürlükteki etkin günlük veri kümelerini keser, ardından zamanuyumsuz bir boşaltma işlemi çalıştırır ve BSDS ' yi boşaltma işleminin bir kaydıyla günceller.

ARCHIVE LOG komutunun bir **MODE (QUIESCE)** seçeneği vardır. Bu seçenikle, IBM MQ işleri ve kullanıcıları bir kesinleştirme noktasından sonra susturulur ve sonuç tutarlılık noktası boşaltılmadan önce yürürlükteki etkin günlükte yakalanır.

Müşteri yeri dışında kurtarma için bir yedekleme stratejisi planlarken **MODE (QUIESCE)** seçeneğini kullanmayı düşünün. Kurtarma sırasında arşiv günlüğü en güncel yedek sayfa kümesi kopyasıyla kullanıldığında veri tutarsızlıklarının sayısını en aza indiren, sistem çapında bir tutarlılık noktası yaratır. Örneğin:

```
ARCHIVE LOG MODE(QUIESCE)
```

ARCHIVE LOG komutunu bir **TIME** değiştirgesi belirtmeden çalıştırırsanız, susturma süresi varsayılan olarak CSQ6ARVP makrosunun **QUIESCE** değiştirgesinin değerine ayarlanır. **ARCHIVE LOG MODE (QUIESCE) ' in** tamamlanması için gereken süre, belirtilen süreden azsa, komut başarıyla

tamamlanır; tersi durumda, süre dolduğunda komut başarısız olur. **TIME** seçeneğini kullanarak zaman dönemini belirttik olarak belirtebilirsiniz; örneğin:

```
ARCHIVE LOG MODE(QUIESCE) TIME(60)
```

Bu komut, **ARCHIVE LOG** işlemi gerçekleşmeden önce 60 saniyeye kadar olan bir susturma süresini belirtir.

Dikkat: Zaman önemli olduğunda **TIME** seçeneğinin kullanılması, IBM MQ kaynaklarını kullanan tüm işler ve kullanıcılar için IBM MQ kullanılabilirliğini önemli ölçüde kesintiye uğratabilir.

Varsayılan olarak, komut, komutu sunacağınız zamandan itibaren zamanuyumsuz olarak işlenir. (Komutun diğer IBM MQ komutlarıyla zamanuyumlu olarak işlenmesi için **QUIESCE** ile birlikte **WAIT (YES)** seçeneğini kullanın, ancak z/OS konsolunun tüm **QUIESCE** dönemi için IBM MQ komut girişinden kilitlendiğini unutmayın.)

Susturma süresi boyunca:

- Kuyruk yöneticisindeki işlerin ve kullanıcıların kesinleştirme işlemlerinden geçmelerine izin verilir, ancak kesinleştirmeden sonra herhangi bir IBM MQ kaynağını güncellemeye çalışıyorlarsa askıya alınır.
- İşler ya da askıya alınmış kullanıcılar tarafından tutulan kilitleri bekliyor olabileceğinden, yalnızca verileri okuyan işler ve kullanıcılar etkilenebilir.
- Yeni görevler başlatılabilir, ancak verileri güncelleyemezler.

DISPLAY LOG komutunun çıkışı, bir susturma işleminin yürürlükte olduğunu belirtmek için CSQV400I iletisini kullanır.

Örneğin, IBM MQ 9.1.4' dan:

```
CSQJ322I +CSQ1 DISPLAY LOG report ...
Parameter  Initial value      SET value
-----
INBUFF      60
OUTBUFF     400
MAXRTU      2
MAXARCH     2
TWOACTV     YES
TWOARCH     YES
TWOBSDS     YES
OFFLOAD     YES
MAXCNOFF    0
WRTHRS      20
DEALLCT     0
COMPLLOG    NONE
ZHYWRITE    NO
End of LOG report
CSQJ370I +CSQ1 LOG status report ...
Copy %Full zHyperWrite Encrypted DSName
  1    68 NO          NO      VICY.CSQ1.LOGCOPY1.DS01
  2    68 NO          NO      VICY.CSQ1.LOGCOPY2.DS01
Restarted at 2019-08-15 09:49:30 using RBA=000000000891B000
Latest RBA=000000000891CCF8
Offload task is AVAILABLE
Full logs to offload - 0 of 4
CSQV400I +CSQ1 ARCHIVE LOG QUIESCE CURRENTLY ACTIVE
CSQ9022I +CSQ1 CSQJC001 ' DISPLAY LOG' NORMAL COMPLETION
```

Örneğin, IBM MQ 9.1.2öncesinde Long Term Support ve Continuous Delivery için:

```
CSQJ322I +CSQ1 DISPLAY LOG report ...
Parameter  Initial value      SET value
-----
INBUFF      60
OUTBUFF     400
MAXRTU      2
MAXARCH     2
TWOACTV     YES
TWOARCH     YES
TWOBSDS     YES
```

```
OFFLOAD      YES
MAXCNOFF    0
WRTHRS      20
DEALLCT     0
COMPLG      NONE
ZHYWRITE    NO                YES
End of LOG report
CSQJ370I +CSQ1 LOG status report ...
Copy %Full PPRC DSName
  1      68 NO VICY.CSQ1.LOGCOPY1.DS01
  2      68 NO VICY.CSQ1.LOGCOPY2.DS01
Restarted at 2014-04-15 09:49:30 using RBA=000000000891B000
Latest RBA=000000000891CCF8
Offload task is AVAILABLE
Full logs to offload - 0 of 4
CSQV400I +CSQ1 ARCHIVE LOG QUIESCE CURRENTLY ACTIVE
CSQ9022I +CSQ1 CSQJC001 ' DISPLAY LOG' NORMAL COMPLETION
```

Tüm güncellemeler susturulduğunda, BSDS ' deki susturma geçmişi kaydı, etkin günlük veri kümelerinin kesildiği tarih ve saat ile ve geçerli etkin günlük veri kümelerinde son yazılan RBA ile güncellenir. IBM MQ , geçerli etkin günlük veri kümelerini keser, sonraki kullanılabilir etkin günlük veri kümelerine geçer ve boşaltma işleminin başladığını belirten CSQJ311I iletisini yayınlar.

Susturma süresi sona ermeden güncellemeler susturulamazsa, IBM MQ CSQJ317I iletisini yayınlar ve **ARCHIVE LOG** işlemi sona erer. Yürürlükteki etkin günlük veri kümeleri kesilmez ya da sonraki kullanılabilir günlük veri kümelerine geçilmez ve boşaltma işlemi başlatılır.

Susturma başarılı olsun ya da olmasın, askıya alınan tüm kullanıcılar ve işler sürdürülür ve IBM MQ , susturmanın sona erdirildiğini ve güncelleme etkinliğinin devam ettiğini belirten CSQJ312I iletisini yayınlar.

Geçerli etkin günlük kullanılabilir son etkin günlük veri kümesi olduğunda **ARCHIVE LOG** yayınlanırsa, komut işlenmez ve IBM MQ aşağıdaki iletiyi yayınlar:

```
CSQJ319I -csect-name CURRENT ACTIVE LOG DATA SET IS THE LAST (CSQJ319I YÜRÜRLÜKTEKİ ETKİN GÜNLÜK VERİ KÜMESİ SON)
KULLANILABİLİR GÜNLÜK VERİ KÜMESİ. ARŞİVLEME GÜNLÜĞÜ
SONA ERDİRİLECEK
```

Başka bir **ARCHIVE LOG** komutu devam ederken **ARCHIVE LOG** komutu verilirse, yeni komut işlenmez ve IBM MQ şu iletiyi yayınlar:

```
CSQJ318I -ARŞİV GÜNLÜK KOMUTU ZATEN DEVAM EDİYOR
```

Arşivleme sırasında verilen iletilerle ilgili bilgi için [IBM MQ for z/OS ile ilgili iletiler başlıklı konuya](#) bakın.

Bir hatadan sonra günlük arşivi işlemini yeniden başlatma

Günlük arşivleme işlemi sırasında bir sorun varsa (örneğin, ayırma ya da manyetik bant bağlamasıyla ilgili bir sorun), etkin günlüğün arşivlenmesi askıya alınabilir. Arşiv işlemini iptal edebilir ve aşağıdaki komutu kullanarak yeniden başlatabilirsiniz:

```
ARCHIVE LOG CANCEL OFFLOAD
```

Bu komut, devam etmekte olan boşaltma işlemlerini iptal eder ve arşiv işlemini yeniden başlatır. Arşivlenmemiş en eski günlük veri kümesiyle başlar ve boşaltılması gereken tüm etkin günlük veri kümelerinden devam eder. Askıya alınan günlük arşivi işlemleri yeniden başlatılır.

Bu komutu, yürürlükteki günlük arşivi görevinin artık çalışmadığından eminseniz ya da başarısız olan önceki bir girişimi yeniden başlatmak istiyorsanız kullanın. Bunun nedeni, komutun boşaltma görevinin olağandışı bir şekilde sonlandırılmasına neden olması ve bu da bir dökümle sonuçlanabilir.

Arşivlemeyi ve günlüğe kaydetmeyi denetleme

CSQ6LOGP, CSQ6ARVP ve CSQ6SYSP makrolarını kullanarak sıkıştırma, yazdırma, arşivleme, kurtarma ve günlüğe kaydetme işlemlerini denetleyebilirsiniz. Özel nesnelere yapılan değişikliklerin yalnızca IBM MQ günlüklerine kaydedildiğini unutmayın. Grup nesnelere (paylaşılan gelen kanallar gibi) üzerindeki değişiklikler de günlüğe kaydedilir; çünkü tanımlar grubun etrafına yayılır ve yerel olarak tutulur.

Arşivlemenin ve günlüğe kaydetmenin birçok yönü, kuyruk yöneticisi uyarlandığında sistem değiştirgesi modülünün CSQ6LOGP, CSQ6ARVP ve CSQ6SYSP makroları kullanılarak ayarlanan değiştirgelerle denetlenir. Bu makroların ayrıntıları için bkz. [Sistem parametre modülünüzü izle](#) .

Bir kuyruk yöneticisi IBM MQ MQSC SET LOG, SET SYSTEM ve SET ARCHIVE komutlarını kullanarak çalışırken bu parametrelerin bazıları değiştirilebilir. Bunlar [Çizelge 28 sayfa 503](#) içinde gösterilir:

SET komutu	Parametreler
LOG	WRTHRSR, MAXARCH, DEALLCT, MAXRTU, COMPLOG
ARŞIV	Tümü
SYSTEM	YÜKLEME

MQSC DISPLAY LOG, DISPLAY ARCHIVE ve DISPLAY SYSTEM komutlarını kullanarak tüm parametrelerin ayarlarını görüntüleyebilirsiniz. Bu komutlar, arşivleme ve günlüğe kaydetmeyle ilgili durum bilgilerini de gösterir.

Günlük sıkıştırmanın denetlenmesi

Şunlardan birini kullanarak günlük kayıtlarının sıkıştırılmasını etkinleştirebilir ve devre dışı bırakabilirsiniz

- MQSC ' de SET ve DISPLAY LOG komutları; bkz. [MQSC komutları](#)
- PCF arabirimi çağırılıyor. Bakınız [“IBM MQ Programlanabilir Komut Biçimlerine Giriş” sayfa 25](#)
- Sistem değiştirgesi modülünde CSQ6LOGP makrosunun kullanılması; bkz. [CSQ6LOGP](#)

Günlük kayıtlarının yazdırılması

CSQ1LOGP yardımcı programını kullanarak günlük kayıtlarını alabilir ve yazdırabilirsiniz. Yönergeler için bkz. [Günlük yazdırma yardımcı programı](#).

Günlüklerin kurtarılması

Normalde, özellikle ikili günlük kaydı kullanıyorsanız, IBM MQ günlüklerini yedeklemeniz ve geri yüklemeniz gerekmez. Ancak, bir günlükteki G/Ç hatası gibi nadir durumlarda günlükleri kurtarmanız gerekebilir. Veri kümesini silmek ve yeniden tanımlamak için Erişim Yöntemi Hizmetleri 'ni kullanın ve ardından buna karşılık gelen ikili oturumu kopyalayın.

Arşiv günlüğü veri kümeleri atılıyor

Arşiv günlüğü veri kümelerinizi atabilir ve günlükleri otomatik olarak ya da el ile atmayı seçebilirsiniz.

Bir sayfa kümesi kaybolursa, iş birimi kurtarma, sayfa kümesi ortam kurtarma ya da CF yapısı kaybolursa CF yapısı ortam kurtarma işlemlerini gerçekleştirebilmek için yeterli günlük verisi tutmanız gerekir. Kurtarma için gerekli olabilecek arşiv günlüğü veri kümelerini atmayın; bu arşiv günlüğü veri kümelerini atmazsanız, gerekli kurtarma işlemlerini gerçekleştiremeyebilirsiniz.

Arşiv günlüğü veri kümelerinizin atılabileceğini doğruladıysanız, bunu aşağıdaki yollardan biriyle yapabilirsiniz:

- [Otomatik arşiv günlüğü veri kümesi silme](#)
- [Arşiv günlüğü veri kümelerinin el ile silinmesi](#)

Otomatik arşiv günlüğü veri kümesi silme

Arşiv günlüğü veri kümelerini otomatik olarak silmek için bir DASD ya da manyetik bant yönetim sistemi kullanabilirsiniz. IBM MQ arşiv günlüğü veri kümelerine ilişkin alıkoyma süresi, CSQ6ARVP kuruluş makrosunda ARCRETN alıkoyma süresi alanı tarafından belirtilir (ek bilgi için [CSQ6ARVP olanağının kullanılması](#) kısmına bakın).

Alıkoyma süresi için varsayılan değer, arşiv günlüklerinin 9999 gün boyunca saklanacağını belirtir (üst sınır).

Önemli: Saklama süresini değiştirebilirsiniz, ancak planladığınız yedekleme döngüsü sayısını karşılayabildiğinizden emin olmanız gerekir.

IBM MQ , arşiv günlüğü veri kümeleri yaratıldığında JCL değiştirgesi RETPD için değer olarak alıkoyma süresi değerini kullanır.

MVS/DFP depolama yönetimi altsistemi (SMS) tarafından ayarlanan alıkoyma süresi bu IBM MQ parametresi tarafından geçersiz kılınabilir. Genellikle, alıkoyma süresi IBM MQ ya da SMS ile belirtilen daha küçük değere ayarlanır. Depolama yöneticisi ve IBM MQ yöneticisi, IBM MQ için uygun bir alıkoyma süresi değerini kabul etmelidir.

Not: IBM MQ , BSDS ' den arşiv günlüğü veri kümeleriyle ilgili bilgileri silmek için otomatikleştirilmiş bir yöntemle sahip değildir; bazı manyetik bant yönetim sistemleri, saklama sürelerinin dış el ile geçersiz kılınmasını sağlar. Bu nedenle, bir arşiv günlüğü veri kümesiyle ilgili bilgiler, veri kümesi saklama süresi sona erdikten ve veri kümesi manyetik bant yönetim sistemi tarafından çizildikten çok sonra da BSDS ' de olabilir. Tersine, arşiv günlüğü veri kümesi sayısı üst sınırı aşılmış olabilir ve BSDS ' deki veriler, veri kümesi süre bitim tarihine ulaşmadan önce atılmış olabilir.

Arşiv günlüğü veri kümeleri otomatik olarak silinirse, işlemin BSDS ' deki arşiv günlükleri listesini güncellemediğini unutmayın. BSDS ' yi "[BSDS ' nin Değiştirilmesi](#)" sayfa 511 içinde açıklandığı gibi değişiklik günlüğü stok yardımcı programıyla güncelleyebilirsiniz. Güncelleme gerekli değil. Eski arşiv günlüklerinin kaydedilmesi BSDS ' de yer kaybına, ancak başka bir zarar vermez.

Arşiv günlüğü veri kümelerini el ile silme

Tüm günlük kayıtlarını, CSQI024I ve CSQI025I iletilerinde tanımlanan en düşük RBA ' ya kadar saklamalısınız. Bu RBA, [Yöntem 1: Tam yedeklemek](#) kullanılarak bir kurtarma noktası oluştururken yayınladığınız DISPLAY USAGE komutu kullanılarak alınır.

Herhangi bir günlüğü atmadan önce [Paylaşılan olmayan kaynaklar için kurtarma noktası yaratılması](#) başlıklı belgeyi okuyun.

Arşiv günlüğü veri kümelerini bul ve at

Kurtarma için gereken günlük RBA alt sınırını belirledikten sonra, aşağıdaki yordamı gerçekleştirerek yalnızca daha önceki günlük kayıtlarını içeren arşiv günlüğü veri kümelerini bulabilirsiniz:

1. BSDS içeriğini yazdırmak için günlük eşlemi yazdırma yardımcı programını kullanın. Çıkişın bir örneği için bkz. [Yazdırma günlüğü eşlemi yardımcı programı](#).
2. Çıkişın ARCHIVE LOG COPY n DATA SETS başlıklı bölümlerini bulun. İkili günlük kaydı kullanıyorsanız, iki bölüm vardır. STARTRBA ve ENDRBA etiketli sütunlar, her birimde bulunan RNA ' ların aralığını gösterir. CSQI024I ve CSQI025I iletileriyle bulduğunuz RBA alt sınırını içeren birimleri bulun. Bunlar saklamanız gereken en eski hacimlerdir. Çift günlük kaydı kullanıyorsanız, bu tür iki birim vardır.

Uygun bir aralığa sahip birim yoksa, aşağıdaki durumlardan biri geçerlidir:

- RBA alt sınırı henüz arşivlenmedi ve tüm arşiv günlüğü birimlerini atabilirsiniz.
- Birim sayısı, CSQ6LOGP makrosunun MAXARCH parametresi tarafından izin verilen sayıyı aştığında BSDS ' deki arşiv günlüğü birimlerinin listesi. BSDS bir arşiv günlüğü birimini kaydetmezse, o birim kurtarma için kullanılamaz. Bu nedenle, BSDS ' ye var olan birimlerle ilgili bilgi eklemeyi düşünün. Yönergeler için, bkz. "[Arşiv günlüklerine ilişkin değişiklikler](#)" sayfa 513.

MAXARCH değerini artırmayı da düşünün. Ek bilgi için [CSQ6LOGP' nin Kullanılması](#) başlıklı konuya bakın.

3. Saklamak istediğiniz en eski birimin STARTRBA değerinden küçük bir ENDRBA değerine sahip arşiv günlüğü veri kümesini ya da birimi silin. İkili günlük kaydı kullanıyorsanız, bu iki kopyayı da silin.

BSDS girişleri etrafı sardığı için, BSDS arşiv günlüğü bölümündeki ilk birkaç giriş sondaki girdilerden daha yeni olabilir. Tarih ve saat birleşimine bakın ve yaşlarını karşılaştırın. LOGRBA alt sınırını içeren arşiv günlüğü girişinden önce tüm girişleri atabileceğinizi varsaymayın.

Veri kümelerini silin. Arşivler kasetteyse, kasetleri silin. DASD ' delerse, her veri kümesini silmek için bir z/OS yardımcı programını çalıştırın. Daha sonra, BSDS ' nin yalnızca var olan arşiv birimlerini listelemesini istiyorsanız, atılan birimlere ilişkin girişleri silmek için değişiklik günlüğü döküm yardımcı programını (CSQJU003) kullanın. Bir örnek için bkz. [“Arşiv günlüklerine ilişkin değişiklikler”](#) sayfa 513 .

z/OS **Günlük çekmenin etkisi**

Uzun süreli işlemler, günlük veri kümelerine yayılan iş günlüğü kayıtlarının birimine neden olabilir. IBM MQ , günlük shunting yöntemini kullanarak bu senaryoyu işler; günlük kayıtlarını, tutulan günlük verilerinin miktarını eniyilemek için taşıyan bir teknik ve kuyruk yöneticisi yeniden başlatma süresi.

Bir iş birimi uzun olarak değerlendirildiğinde, her günlük kaydının bir gösterimi günlüğün daha aşağısına yazılır. Bu, *log shunting*(günlük shunting) olarak bilinir. [Günlük dosyaları](#) kısmında daha ayrıntılı olarak açıklanmıştır.

Kuyruk yöneticisi, iş biriminin bütünlüğünü sağlamak için bir hatadan sonra asıl kayıtlar yerine bu parçalı günlük kayıtlarını kullanır. Bunun iki yararı vardır:

- iş birimi koordinasyonu için saklanması gereken günlük verisi miktarı azaltıldı
- kuyruk yöneticisi yeniden başlatma sırasında daha az günlük verisi geçilmelidir, böylece kuyruk yöneticisi daha hızlı yeniden başlatılır

Shunted günlük kayıtları, ortam kurtarma işlemleri için yeterli bilgi içermiyor.

Günlükte tutulan veriler iki ayrı amaç için kullanılır; ortam kurtarma ve iş birimi koordinasyonu. Bir CF yapısını ya da sayfa kümesini etkileyen bir ortam hatası oluşursa, kuyruk yöneticisi, önceki bir kopyayı geri yükleyerek ve bunu günlükte bulunan verileri kullanarak güncelleyerek ortamı hata noktasına kadar kurtarabilir. Bir iş biriminde gerçekleştirilen kalıcı etkinlik günlüğe kaydedilir; böylece, bir hata oluşması durumunda, geri çekilebilir ya da kilitler değiştirilen kaynaklarda kurtarılabilir. Kuyruk yöneticisi kurtarmasını etkinleştirmek için alıkoymanız gereken günlük verisi miktarı bu iki öğeden etkilenir.

Ortam kurtarma için, en azından en son ortam kopyasından ortam kurtarma gerçekleştirebilmek ve geri çekilmek için yeterli günlük verilerini saklamanız gerekir. (Siteniz, eski yedeklerden kurtarma yeteneğini önleyebilir.) İş birimi bütünlüğü için, en eski uçuş ya da belirsiz iş birimlerinize ilişkin günlük verilerini saklamanız gerekir.

Kuyruk yöneticisi, sistemi yönetmeniz için her günlük arşivinde eski iş birimlerini algılar ve bunları CSQJ160 ve CSQJ161 iletilerinde raporlar. Bir iç görev, bu eski iş birimlerine ilişkin iş birimi günlüğü bilgilerini okur ve günlükteki geçerli konuma daha kısa bir biçimde yeniden yazar. CSQR026 iletili, bunun ne zaman olduğunu gösterir. MQSC komutu DISPLAY USAGE TYPE (DATASET), günlük verilerinin alınmasını yönetmenize de yardımcı olabilir. Komut, aşağıdaki üç kurtarma bilgisini bildirir:

1. İş birimi kurtarma işlemi için günlüğün ne kadarının saklanması gerektiğini belirtir.
2. Sayfa kümelerinin ortam kurtarması için günlüğün ne kadarının saklanması gerektiğini belirler.
3. Kuyruk paylaşım grubundaki bir kuyruk yöneticisi için, CF yapılarının ortam kurtarması için günlüğün ne kadarının saklanması gerekir.

Bu bilgi parçalarının her biri için, gerekli en eski günlük verilerini bir veri kümesiyle eşleme girişiminde bulunuldu. Yeni iş birimleri başladıkça ve durdukça, (1) günlüğün daha yeni bir konumuna taşınması beklenecektir. Hareket etmezse, uzun süren UOW iletileri sizi bir sorun olduğu konusunda uyarıyor. (2), kuyruk yöneticisi şimdi kapatılıp yeniden başlatıldıysa, sayfa kümesi ortam kurtarması ile ilgilidir. Sayfa kümelerinizi en son ne zaman yedeklediğiniz ya da bir sayfa kümesi hatası olduğunda hangi

yedeği kullanmanız gerekeceği hakkında bilgi sahibi değildir. Genellikle, arabellek havuzlarında tutulan değişiklikler sayfa kümelerine yazılırken, denetim noktası işlemesi sırasında günlükte daha yeni bir konuma taşınır. (3) içinde, kuyruk yöneticisi, bu kuyruk yöneticisinde ya da kuyruk paylaşım grubundaki diğer kuyruk yöneticilerine alınan CF yapısı yedeklerini bilir. Ancak CF yapısı kurtarma işlemi, son yedeklemeden bu yana CF yapısıyla etkileşimde bulunan kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinden alınan günlük verilerinin birleştirilmesini gerektirir. Bu, günlük verilerinin, zaman damgasına dayalı bir günlük kaydı sıra numarasıyla (ya da LRSN) tanıtıldığı ve kuyruk paylaşım grubundaki farklı kuyruk yöneticilerinden farklı olacak bir RBA yerine, kuyruk paylaşım grubunun tamamında geçerli olduğu anlamına gelir. Olağan durumda, BACKUP CFSTRUCT komutları kuyruk paylaşım grubundaki bu ya da diğer kuyruk yöneticileri üzerinde gerçekleştirildiğinden, günlükte daha yeni bir konuma geçer.

z/OS Kuyruk yöneticisinin günlüğünün ilk durumuna getirilmesi

Kuyruk yöneticisinin günlüğünün nasıl sıfırlanacağını anlamak için bu konuyu kullanın.

Kuyruk yöneticisi günlüğü RBA 'nın günlük RBA aralığının sonundan 0 'a kadar kaymasına izin vermemelisiniz; bu bir kuyruk yöneticisi kesintisine neden olur ve tüm kalıcı veriler kurtarılamaz duruma gelir. Günlük RBA 'nın sonu FFFFFFFF (6 baytlık RNA' lar kullanıyorsa) ya da FFFFFFFFFFFFFFFF (8 baytlık RNA ' lar kullanıyorsa) değeridir.

Kuyruk yöneticisi, kullanılan günlük aralığının önemli olduğunu ve planlanmamış bir kesintiyi önlemek için işlem yapmayı planladığınızı belirtmek üzere [CSQI045I](#), [CSQI046E](#), [CSQI047E](#), [CSQJ031D](#) ve [CSQJ032E](#) iletileri yayınlar.

RBA değeri FFF800000000 (6 baytlık günlük RNA 'ları kullanımdaysa) ya da FFFFFFFC00000000 (8 baytlık günlük RNA' ları kullanıyorsa) değerine ulaştığında kuyruk yöneticisi [00D10257](#) neden koduyla sona erer.

6 baytlık günlük RNA 'ları kullanımdaysa, kuyruk yöneticisini, [“Daha büyük günlük Görelî Bayt Adresi Uygulanıyor”](#) sayfa 508 başlıklı konuda açıklanan işlemi izleyerek kuyruk yöneticisinin günlüğünü sıfırlamak yerine 8 baytlık günlük RNA' larını kullanacak şekilde dönüştürmeyi düşünün. Bir kuyruk yöneticisinin 8 baytlık günlük RNA ' larını kullanacak şekilde dönüştürülmesi, günlüğün ilk durumuna getirilmesinden daha kısa bir kesinti gerektirir ve günlüğü ilk durumuna getirmeden önce geçen süreyi artırır.

Kuyruk yöneticisi kullanıma hazırlanırken yayınlanan [CSQJ034I](#) iletileri, kuyruk yöneticisine ilişkin günlük RBA aralığının yapılandırılmış olarak sona erdiğini gösterir ve 6 baytlık ya da 8 baytlık günlük RNA ' larının kullanılıp kullanılmadığını belirlemek için kullanılabilir.

Kuyruk yöneticisinin günlüğünü ilk durumuna getirmek için izlenecek yordam aşağıdaki gibidir:

1. Çözümlememiş iş birimlerini çözün. CSQR005I iletilerinde kuyruk yöneticisi başlatılırken, çözülmemiş iş birimi sayısı INDOUBT sayısı olarak görüntülenir. Her denetim noktasında ve kuyruk yöneticisi sona erdirildiğinde, kuyruk yöneticisi otomatik olarak komutu verir

Çözümlememiş iş birimlerine ilişkin bilgi sağlamak için **DISPLAY CONN(*) TYPE(CONN) ALL WHERE(UOWSTATE EQ UNRESOLVED)** .

Kurtarma birimlerinin çözümlenmesine ilişkin bilgi için [Nasıl belirsiz kurtarma birimlerinin çözümlendiğini](#) başlıklı konuya bakın. Nihai çözüm, belirsiz kurtarma birimlerini el ile çözmek için **RESOLVE INDOUBT MQSC** komutunu kullanmaktır.

2. Kuyruk yöneticisini temiz bir şekilde kapatın.

Bu komutların her ikisi de değiştirilen sayfaları arabellek havuzlarından sayfa kümelerine boşaltmak için **STOP QMGR** ya da **STOP QMGR MODE(FORCE)** kullanabilirsiniz.

3. Kuyruk yöneticisi bir kuyruk paylaşım grubunun parçasıysa, kuyruk paylaşım grubundaki tüm yapılar için diğer kuyruk yöneticilerine CFSTRUCT yedekleri alın. Bu, en son yedeklerin bu kuyruk yöneticisinin günlüğünde olmamasını ve CFSTRUCT kurtarma işlemi için bu kuyruk yöneticisinin günlüğünün gerekli olmamasını sağlar.
4. CSQJU003 kullanarak yeni günlükler ve BSDS tanımlayın (değişiklik günlüğü dökümü yardımcı programının kullanılmasına ilişkin ek bilgi için [Değişiklik günlüğü dökümü yardımcı programı](#) konusuna bakın).

5. Bu kuyruk yöneticisine ilişkin tüm sayfa kümelerinde **CSQUTIL RESETPAGE** komutunu çalıştırın (bu işlevin kullanılmasıyla ilgili ek bilgi için [Sayfanın kopyalanması ve günlüğün ilk durumuna getirilmesi](#) konusuna bakın). Sayfa kümesi RNA ' ları bağımsız olarak sıfırlanabildiği için, bu adım için geçen süreyi azaltmak üzere birden çok eşzamanlı iş (örneğin, sayfa başına bir iş) sunulabileceği unutulmamalıdır.
6. Kuyruk yöneticisini yeniden başlat

Uyarı iletileri

IBM MQ günlüğün sonunun yaklaştığını algıladığında, konsol iletilerini aşağıdaki sırada yayınlar ve bu, günlüğün ilk durumuna getirilmesinin planlanması gerektiğini gösterir. Bu bölümde, iletiler 6 baytlık günlük RBA değerlerini gösterir. IBM MQ 8 baytlık günlük RBA kipinde ancak farklı değerlerle çalışırken aynı konsol iletileri yayınlanır; 8 baytlık günlük RBA eşikleri için bkz. [“Uyarı eşikleri” sayfa 508](#) .

1. IBM MQ günlüğün sonunun yakın gelecekte yaklaştığını algırsa (yaklaşık %94 tam) IBM MQ , konsol iletilerini CSQI045I, aşağıdaki örnekte olduğu gibi gönderir:

```
CSQI045I -CSQ7 CSQILCUR Log RBA has reached 0000F00000000000.  
Plan a log reset
```

2. IBM MQ , günlüğün sonu yaklaştığında (yaklaşık %97 dolu) aşağıdaki CSQI046E hata konsolu iletilerini yayınlar. Bu, IBM MQ yöneticisine yakında işlem yapması gerektiğini bildirir.

```
CSQI046E -CSQ7 CSQILCUR Log RBA has reached 0000F80000000000.  
Perform a log reset
```

3. CSQI046E iletilerini yayınladıktan sonra, IBM MQ sonraki günlük anahtarında UYARI sözcüğüyle birlikte aşağıdaki CSQJ032E konsol iletilerini yayınlar:

```
CSQJ032E -CSQ7 CSQJW307 WARNING - APPROACHING END OF  
THE LOG RBA RANGE OF 0000FFFFFFFFFFFF. CURRENT LOG RBA IS 0000F80000022000.
```

4. CSQI046E ve CSQJ032E konsol iletilerini yayınladıktan sonra, IBM MQ hemen IBM MQ yöneticisi müdahalesi gerektirmeyen bir hata iletilerini daha yayınlar. IBM MQ konsol iletilerini yayınlar CSQI047E (günlük yaklaşık %99 dolu olduğunda):

```
CSQI047E -CSQ7 CSQILCUR Log RBA has reached 0000FF0000000000.  
Stop queue manager and reset logs
```

5. Günlük RBA FF8000000000değerine ulaştığında, IBM MQ durumun aciliyetini artırır ve CRITICAL sözcüğüyle CSQJ032E konsol iletilerini yayınlar:

```
CSQJ032E -CSQ7 CSQJW009 CRITICAL - APPROACHING END OF THE LOG RBA RANGE OF 0000FFFFFFFFFFFF.  
CURRENT LOG RBA IS 0000FFF7FFFFDFFF.
```

6. Günlük RBA üst sınırına ulaşıldığında kuyruk yöneticisi başlatılırsa, aşağıdaki CSQJ031D konsol iletilerini yayınlanır. Bu aşama, IBM MQ yöneticisinin girişini gerektirir:

```
CSQJ031D -CSQ7 CSQYSTR THE LOG RBA RANGE MUST BE RESET.  
REPLY 'Y' TO CONTINUE STARTUP OR 'N' TO SHUTDOWN
```

7. IBM MQ başlatma işlemi, CSQJ031Diletilerine yanıt verinceye kadar askıda kalır.

Bu iletilerinin amacı, IBM MQ yöneticisine günlükleri sıfırlamak için bir sistem kesintisi planlaması için zaman vermektir. İdeal bir yapılandırmada, büyük olasılıkla bir kuyruk paylaşım grubunda (QSG) iş yükünü paylaşan en az iki kuyruk yöneticisi vardır. Biri bakım için kapalı olduğunda diğeri iş almaya devam edebilir.

RBA sona yaklaştıkça IBM MQ ' in sorun yaşadığı konsol iletilerinin önem derecesi daha da büyür. İdeal olarak IBM MQ yöneticiniz, ilk konsol iletilerini görüntülediğinde günlük RBA ' yı sıfırlamayı planlamalıdır.

Uyarı ve hata konsolu iletilerini yoksayırsa, günlük RBA FFF800000000değerine ulaştığında IBM MQ neden koduyla sonlandırılır 5C6-00D10257 ; bu noktada IBM MQ kullanılabilir aralığın kuyruk yöneticisinin devam edemeyecek kadar küçük olduğunu belirler. Bu noktaya ulaşıldığında, tek seçenek bir kesinti almak ve günlüğü sıfırlamak ya da günlük RBA boyutunu genişletmektir.

Not: Günlüğün sonuna ulaşıldığında, devam eden herhangi bir iş birimi (UOW) çözümlenemez; bunlar günlük sıfırlama işlemi sırasında kaybolur. Kuyruk yöneticisini başlatmak ve herhangi bir UOW 'yi çözümlmek için yeterli RBA aralığı bırakılmalıdır. IBM MQ , günlüğün sonunun yaklaştığını bildirmek için konsol iletilerini birkaç kez yayınladığından, bir günlük sıfırlaması planlanmalıdır.

Uçuştaki UOW 'ları kaybetmemek için tercih edilen seçenek, günlük RBA' yı 8 bayt kullanacak şekilde genişletmektir. Bu, uzun bir süre için günlük RBA sıfırlaması gerekmediği anlamına gelir.

Uyarı eşikleri

Aşağıdaki çizelge, günlük RBA uzunluğuna dayalı olarak eşikleri listeler.

Konsol iletileri	6 baytlık günlük RBA	8 baytlık günlük RBA
CSQI045I	0000F00000000000	FFFF800000000000
CSQI046E	0000F80000000000	FFFFC00000000000
CSQI047E	0000FF8000000000	FFFFFC0000000000
CSQJ032E	0000FF8000000000 0000FF8000000000	FFFFFC0000000000 FFFFFC0000000000
CSQJ031D	0000FF8000000000	FFFFFC0000000000

Notlar:

- CSQJ032E iletileri için, ilk sayı UYARI metnine, ikinci sayı konsol iletilerindeki CRITICAL metnine uygulanır.
- CSQJ031D iletileri yalnızca IBM MQ kullanıma hazırlama sırasında yayınlanır.

İlgili kavramlar


“Daha büyük günlük Görelî Bayt Adresi Uygulanıyor” sayfa 508

IBM MQ for z/OS 8.0' den önce IBM MQ for z/OS , günlük içindeki verilerin konumunu tanımlamak için 6 baytlık bir günlük RBA kullandı. IBM MQ for z/OS 8.0' den günlük RBA 8 bayt uzunluğunda olabilir ve günlüğü ilk durumuna getirmeden önce geçen süreyi artırır.


Daha büyük günlük Görelî Bayt Adresi Uygulanıyor

IBM MQ for z/OS 8.0' den önce IBM MQ for z/OS , günlük içindeki verilerin konumunu tanımlamak için 6 baytlık bir günlük RBA kullandı. IBM MQ for z/OS 8.0' den günlük RBA 8 bayt uzunluğunda olabilir ve günlüğü ilk durumuna getirmeden önce geçen süreyi artırır.



Uyarı:  Kuyruk yöneticileriniz IBM MQ 9.3.0'den önce yaratıldıysa, IBM MQ 9.3.0 ' da yaratılan kuyruk yöneticileri ve daha sonra bu özellik etkinleştirildiyse, bu özelliği etkinleştirmek için aşağıdaki yordamı gerçekleştirmeniz gerekir.

8 baytlık günlük RBA ' yı etkinleştirmeyi planlarken dikkat edilecek noktalar için [Adreslenebilir günlük aralığı üst sınırını artırmayı planlama](#) başlıklı konuya bakın.

 Tek bir IBM MQ for z/OS kuyruk yöneticisinde 8 baytlık günlük RBA ' yı etkinleştirmek için bu yönergeleri gösterildiği sırayla gerçekleştirin. Kuyruk paylaşım grubundaki kuyruk yöneticileri için, sırayla her kuyruk yöneticisindeki adımları gerçekleştirin.

- Geçerli BSDS ' ye benzer özniteliklere sahip yeni BSDS veri kümeleri ayırın. CSQ4BSDS örneğini uyarlayabilir ve ilgisiz herhangi bir deyimli silebilir ya da varolan JCL ' yi kullanabilir, ancak BSDS adını ++HLQ++ .NEW . BSDS01 gibi bir şekilde değiştirebilirsiniz.

Notlar:

- Yeni BSDS 'yi ayırmak için işi göndermeden önce yeni BSDS' nin özniteliklerini denetleyin. Değişebilecek tek öznitelik BSDS ' nin boyutudur.

b. Yeni BSDS yürürlükteki BSDS ' den daha fazla veri içerdiğinden, yeni veri kümelerinin yeterli kullanılabilir alanla ayrıldığından emin olmanız gerekir. thlqual.SCSQPROC(CSQ4BSDS) içindeki örnek JCL, yeni bir BSDS tanımlanırken önerilen değerleri içerir.

2. Kuyruk yöneticisini temiz bir şekilde kapatın.

3. Var olan BSDS ' yi yeni BSDS veri kümelerine dönüştürmek için [BSDS dönüştürme yardımcı programını \(CSQJUCNV\)](#) çalıştırın. Bu genellikle birkaç saniye sürer.

Var olan BSDS ' niz bu işlem sırasında değiştirilmez ve başarısız bir dönüştürme durumunda kuyruk yöneticisinin kullanıma hazırlanması için bunu kullanabilirsiniz.

4. Yürürlükteki BSDS 'yi eski BSDS olacak şekilde ve yeni BSDS' yi yürürlükteki BSDS olacak şekilde yeniden adlandırın; böylece, kuyruk yöneticisini yeniden başlattığınızda yeni veri kümeleri kullanılır. DFSMS Access Method Services ALTER komutunu kullanabilirsiniz; örneğin:

```
ALTER '++HLQ++.BSDS01' NEWNAME('++HLQ++.OLD.BSDS01')
ALTER '++HLQ++.NEW.BSDS01' NEWNAME('++HLQ++.BSDS01')
```

VSAM kümesinin hem veri hem de izin bölümlerini yeniden adlandırmak için komutlar verdiğinizden emin olun.

5. Kuyruk yöneticisini yeniden başlatın. 6 baytlık günlük RBA ' yi kullanırken olduğu gibi aynı süre içinde başlamalıdır.

Dönüştürülen BSDS ' ye erişilememesi nedeniyle kuyruk yöneticisi başarıyla yeniden başlatılamazsa, hatanın nedenini saptamaya çalışın, sorunu çözün ve işlemi yeniden deneyin. Gerekirse, yardım almak için IBM destek merkezimize başvurun.

Gerekirse, değişiklik şu noktada şu şekilde geri çekilebilir:

- Geçerli BSDS, yeni BSDS olacak şekilde yeniden adlandırılıyor.
- Eski BSDS, yürürlükteki BSDS olacak şekilde yeniden adlandırılıyor.
- Kuyruk yöneticisi yeniden başlatılıyor.

6. Kuyruk yöneticisi dönüştürülen BSDS ile başarıyla yeniden başlatıldıktan sonra, eski BSDS ' yi kullanarak kuyruk yöneticisini başlatmayı denemeyin.

7. Kuyruk yöneticisine ilişkin günlük RBA ' nın sonunun yapılandırıldığı şekilde belirtilmesi için kuyruk yöneticisi kullanıma hazırlanırken CSQJ034I iletisi yayınlanır. Görüntülenen günlük RBA aralığının sonunun FFFFFFFFFFFFFFFF olduğunu doğrulayın. Bu, 8 baytlık günlük RBA ' nın kullanımda olduğunu gösterir.

İlgili görevler

[Adreslenebilir günlük aralığı üst sınırını artırmayı planlama](#)

İlgili başvurular

[Daha büyük günlük görelî byte adresi](#)

[BSDS dönüştürme yardımcı programı \(CSQJUCNV\)](#)

Önyükleme veri kümesinin (BSDS) yönetilmesi

Önyükleme veri kümesi (BSDS), günlük veri kümelerine ve günlük kayıtlarına başvurmak için kullanılır. BSDS ' yi nasıl inceleyebileceğinizi, değiştirebileceğinizi ve kurtarabileceğinizi anlamak için bu konuyu kullanın.

Daha fazla bilgi için bkz. [Önyükleme veri kümesi](#).

Bu konuda, önyükleme veri kümesinin yönetilmesiyle ilgili görevler açıklanmaktadır. Aşağıdaki kısımları içerir:

- [“BSDS ' nin ne içerdiğini bulma” sayfa 510](#)
- [“BSDS ' nin Değiştirilmesi” sayfa 511](#)

- “BSDS ' nin kurtarılması” sayfa 515

z/OS **BSDS ' nin ne içerdiğini bulma**

BSDS içeriğini incelemek için günlük eşlemi yazdırma yardımcı programını (CSQJU004) kullanabilirsiniz.

Günlük eşlemi yazdırma yardımcı programı (CSQJU004), BSDS ' de saklanan bilgileri listeleyen bir toplu iş yardımcı programıdır. Bunun çalıştırılmasıyla ilgili yönergeler için bkz. [Yazdırma günlüğü eşlemi yardımcı programı](#).

BSDS aşağıdakileri içerir:

- [Zaman damgaları](#)
- [Etkin günlük veri kümesi durumu](#)

BSDS ' deki zaman damgaları

Yazdırma günlüğü eşlemi yardımcı programının çıkışı, BSDS ' de saklanan çeşitli sistem olaylarının tarih ve saatini kaydetmek için kullanılan zaman damgalarını gösterir.

Raporun üstbilgi bölümünde aşağıdaki zaman damgaları yer alır:

SISTEM DAMGASI

BSDS ' nin en son güncellendiği tarihi ve saati gösterir. BSDS zaman damgası aşağıdaki zamanlarda güncellenebilir:

- Kuyruk yöneticisi başlar.
- Günlük yazma etkinlikleri sırasında yazma eşiğine ulaşılır. Belirttiğiniz çıkış arabelleklerinin sayısına ve sistem etkinlik hızına bağlı olarak, BSDS saniyede birkaç kez güncellenebilir ya da birkaç saniye, dakika ya da saat güncellenmeyebilir. Yazma eşiğine ilişkin ayrıntılar için, [CSQ6LOGP](#) içinde [CSQ6LOGP](#) makrosunun [WRTHRS](#) değiştirgesine bakın.
- IBM MQ , bir hata nedeniyle olağan ikili BSDS kipinden tek bir BSDS kipine geçer. Bu durum, bir BSDS kaydını alma, ekleme, güncelleme ya da silme isteği başarısız olduğunda ortaya çıkabilir. Bu hata oluştuğunda IBM MQ , geri kalan BSDS ' deki zaman damgasını, devre dışı bırakılan BSDS ile bir zaman damgası uyumsuzluğu için zorlayacak şekilde günceller.

YARDIMCI PROGRAM

BSDS içeriğinin değişiklik günlüğü stok yardımcı programı tarafından değiştirildiği tarih ve saat (CSQJU003).

Raporun etkin ve arşiv günlüğü veri kümeleri bölümüne aşağıdaki zaman damgaları eklenir:

Etkin günlük tarihi

Etkin günlük girişinin BSDS 'de yaratıldığı, yani CSQJU003 NEWLOG' un gerçekleştirildiği tarih.

Etkin günlük süresi

Etkin günlük girişinin BSDS 'de yaratıldığı, yani CSQJU003 NEWLOG' un gerçekleştirildiği saat.

Arşiv günlüğü tarihi

Arşiv günlüğü girişinin BSDS ' de yaratıldığı tarih; yani, CSQJU003 NEWLOG işlemi yapıldığında ya da arşivin kendisi yapıldığında.

Arşiv günlüğü saati

Arşiv günlüğü girişinin BSDS ' de yaratıldığı zaman; yani, CSQJU003 NEWLOG işlemi yapıldığında ya da arşivin kendisi yapıldığında.

Etkin günlük veri kümesi durumu

BSDS, etkin günlük veri kümesinin durumunu aşağıdakilerden biri olarak kaydeder:

YENİ

Veri kümesi tanımlandı, ancak IBM MQ tarafından hiçbir zaman kullanılmadı ya da günlük, veri kümesi ilk kullanılmadan önce bir noktaya kesildi. Her iki durumda da, RBA değerlerini başlatan ve sona erdiren veri kümesi sıfırlanır.

Yeniden Kullanılabilir

Veri kümesi tanımlandı, ancak IBM MQ tarafından hiç kullanılmadı ya da veri kümesi boşaltıldı. Yazdırma günlüğü işlemi çıkışında, son REUSABLE veri kümesine ilişkin başlangıç RBA değeri, son arşiv günlüğü veri kümesinin başlangıç RBA değerine eşittir.

KULLANILAMAZ

Veri kümesi, boşaltılmamış kayıtlar içeriyor.

Durduruldu

Boşaltma işlemcisi bir kaydı okurken bir hatayla karşılaştı ve bu kayıt etkin günlüğün diğer kopyasından alınamadı.

kesildi

Aşağıdakilerden birini yapın:

- Bir G/Ç hatası oluştu ve IBM MQ bu veri kümesine yazmayı durdurdu. Etkin günlük veri kümesi, başlangıç RBA ' dan başlayarak ve kesilen etkin günlük veri kümesindeki son geçerli kayıt kesimine kadar devam ederek boşaltılır. Son geçerli kayıt kesiminin RBA değeri, etkin günlük veri kümesinin bitiş RBA değerinden düşük. Günlük kaydı, bir sonraki kullanılabilir etkin günlük veri kümesine geçer ve kesintisiz olarak devam eder.

veya

- Etkin günlüğü kısaltan bir ARCHIVE LOG işlevi çağrıldı.

Durum, yazdırma günlüğü işlemi yardımcı programının çıkışında görüntülenir.

z/OS **BSDS ' nin Değiştirilmesi**

IBM MQ bunu otomatik olarak yaptığından, BSDS ' yi günlük kaydı olayları kayıtlarıyla güncel tutmak için özel adımlar atmanız gerekmez.

Ancak, aşağıdakilerden birini yaparsanız BSDS ' yi değiştirmek isteyebilirsiniz:

- Daha fazla etkin günlük veri kümesi ekleyin.
- Etkin günlük veri kümelerini yeni ayrılmış veri kümelerine (örneğin, daha büyük etkin günlük ayrımları sağlarken) kopyalayın.
- Günlük veri kümelerini diğer aygıtlara taşıyın.
- Hasarlı bir BSDS ' yi kurtarın.
- Eski arşiv günlüğü veri kümelerini atın.

Değişiklik günlüğü stok yardımcı programını (CSQJU003) çalıştırarak BSDS ' yi değiştirebilirsiniz. Bu yardımcı programı yalnızca kuyruk yöneticisi etkin değilse çalıştırın ya da tutarsız sonuçlar alabilirsiniz. Yardımcı programın işlemi, SYSIN veri kümesindeki deyimler tarafından denetlenir. Bu bölümde birkaç örnek gösterilmektedir. Eksiksiz yönergeler için [Değişiklik günlüğü stok yardımcı programı](#) başlıklı konuya bakın.

Etkin günlük veri kümesini, IBM MQ kuyruk yöneticisi başlatıldığında etkin günlük veri kümelerini dışlayıcı (DISP = OLD) olarak ayırdığı için, kuyruk yöneticisi etkin olmadığında kopyalayabilirsiniz.

z/OS **Etkin günlüklere ilişkin değişiklikler**

BSDS ' yi kullanarak etkin günlükleri nasıl değiştirebileceğinizi anlamak için bu konuyu kullanın.

Değişiklik günlüğü yardımcı programını kullanarak etkin günlükler için BSDS 'ye giriş ekleyebilir, var olanları silebilir ve BSDS' deki girişleri kaydedebilirsiniz. Yalnızca örnekler burada gösterilir; gösterilen veri kümesi adlarını kullanmak istediğiniz adlarla değiştirin. Yardımcı programa ilişkin daha fazla ayrıntı için bkz. [Değişiklik günlüğü stok yardımcı programı](#).

Daha fazla bilgi için bu bölümlere bakın:

- [BSDS ' ye kayıt girişleri eklenmesi](#)
- [BSDS ' den etkin günlük veri kümesine ilişkin bilgilerin silinmesi](#)
- [BSDS ' de günlük verileri kümesine ilişkin bilgilerin kaydedilmesi](#)

- Etkin günlüğün büyüklüğünün artırılması
- CSQJUFMT kullanımı

BSDS ' ye kayıt girişleri eklenmesi

Etkin bir günlük "durduruldu" olarak işaretlendiyse, günlüğe kaydetme için yeniden kullanılmaz; ancak, okuma için kullanılmaya devam eder. Yeni etkin günlük veri kümelerini tanımlamak için erişim yöntemi hizmetlerini kullanın, daha sonra BSDS ' ye yeni veri kümelerini kaydetmek için değişiklik günlüğü stok yardımcı programını kullanın. Örneğin, şunu kullanın:

```
NEWLOG DSNAMES=MQM111.LOGCOPY1.DS10,COPY1
NEWLOG DSNAMES=MQM111.LOGCOPY2.DS10,COPY2
```

Eski bir etkin günlük veri kümesinin içeriğini yeniyeye kopyalarsanız, RBA aralığını ve NEWLOG işlemindeki başlangıç ve bitiş zaman damgalarını da verebilirsiniz.

Etkin günlük veri kümesine ilişkin bilgilerin BSDS ' den silinmesi

Etkin bir günlük veri kümesine ilişkin bilgileri BSDS ' den silmek için aşağıdakileri kullanabilirsiniz:

```
DELETE DSNAMES=MQM111.LOGCOPY1.DS99
DELETE DSNAMES=MQM111.LOGCOPY2.DS99
```

BSDS ' deki günlük verileri kümesine ilişkin bilgilerin kaydedilmesi

BSDS ' de var olan bir etkin günlük veri kümesine ilişkin bilgileri kaydetmek için aşağıdakileri kullanın:

```
NEWLOG DSNAMES=MQM111.LOGCOPY1.DS10,COPY2,STARTIME=19930212205198,
ENDTIME=19930412205200,STARTRBA=6400,ENDRBA=94FF
```

Bu tip bilgileri içeren bir kaydı BSDS ' ye aşağıdaki nedenlerden ötürü eklemeniz gerekebilir:

- Veri kümesine ilişkin giriş silindi, ancak yeniden gerekiyor.
- Bir etkin günlük veri kümesinin içeriğini başka bir veri kümesine kopyalayabilirsiniz.
- BSDS ' yi bir yedek kopyadan kurtarıyorsunuz.

Etkin günlüğün boyutunu artırma

Bu süreci gerçekleştirmenin iki yöntemi vardır.

1. Kuyruk yöneticisi etkin olduğunda:

- a. JCL kullanarak yeni daha büyük günlük veri kümeleri tanımlayın.
- b. Yeni günlük veri kümelerini MQSC DEFINE LOG komutunu kullanarak etkin kuyruk yöneticisine ekleyin.
- c. Yeni daha büyük bir günlük olacak şekilde, yürürlükteki etkin günlüğü taşımak için MQSC ARCHIVE LOG komutunu kullanın.
- d. Daha küçük etkin günlük veri kümesinin arşivinin tamamlanmasını bekleyin.
- e. Eski küçük etkin günlükleri kaldırmak için CSQJU003 yardımcı programını kullanarak kuyruk yöneticisini kapatın.
- f. Kuyruk yöneticisini yeniden başlatın.

2. Kuyruk yöneticisi etkin olmadığına:

- a. Kuyruk yöneticisini durdurun. IBM MQ , etkin olduğunda tüm etkin günlük veri kümelerini özel kullanımı için ayırdığından bu adım gereklidir.
- b. Etkin günlük veri kümelerinizi yeniden adlandırmak için NEWNAME seçeneğiyle Access Method Services ALTER deyimini kullanın.
- c. Daha büyük etkin günlük veri kümeleri tanımlamak için Erişim Yöntemi Hizmetleri DEFINE seçeneğini kullanın.

Eski veri kümesi adlarını yeniden kullanarak, BSDS'lerde yeni adlar oluşturmak için değişiklik günlüğü stok yardımcı programını çalıştırmanız gerekmez. Eski veri kümesi adları ve doğru RBA aralıkları BSDS'lerde zaten var.

- d. Eski (yeniden adlandırılmış) veri kümelerini uygun yeni veri kümelerine kopyalamak için Erişim Yöntemi Hizmetleri REPRO 'yu kullanın.

Not: Bu adım uzun sürebilir, bu nedenle kuruluşunuz bu süre boyunca işlem dışı olabilir.

- e. Kuyruk yöneticisini başlatın.

Tüm günlük veri kümeleriniz aynı büyüklükteyse, sisteminiz operasyonel olarak daha tutarlı ve verimli olacaktır. Günlük veri kümeleri aynı boyutta değilse, sisteminizin günlüklerini izlemek daha zordur ve böylece alan boşa gider.

CSQJUFMT kullanımı

Etkin bir günlüğün büyüklüğünü artırırken CSQJUFMT biçimini çalıştırmayın.

CSQJUFMT komutunu çalıştırırsanız (kuyruk yöneticisi yeni etkin günlüğe ilk kez yazdığı anda başarımlı avantajı sağlamak için), ileti alırsınız:

```
IEC070I 203-204,XS95GTLX,REPRO02,OUTPUT,B857,SPMG02, 358
IEC070I MG.W.MG4E.LOGCOPY1.DS02,MG.W.MG4E.LOGCOPY1.DS02.DATA,
IDC3302I ACTION ERROR ON MG.W.MG4E.LOGCOPY1.DS02
IDC3351I ** VSAM I/O RETURN CODE IS 28 - RPLFDBWD = X'2908001C'
IDC31467I MAXIMUM ERROR LIMIT REACHED.

IDC0005I NUMBER OF RECORDS PROCESSED WAS 0
```

Ayrıca, Erişim Yöntemi Hizmetleri REPRO 'yu kullanıyorsanız, yeni bir boş günlük tanımladığınızdan emin olun.

Eski (yeniden adlandırılan) veri kümesini ilgili yeni veri kümesine kopyalamak için REPRO kullanırsanız, varsayılan değer NOREPLACE olur.

Bu, REPRO 'nun önceden belirlenmiş veri kümesinde bulunan bir kaydı değiştirmede anlamına gelir. Veri kümesinde biçimlendirme yapıldığında, RBA değeri sıfırlanır. Net sonuç, biçimlendirmeden sonra boş olmayan bir veri kümesidir.

Arşiv günlüklerine ilişkin değişiklikler

Arşiv günlüklerinin nasıl değiştirileceğini anlamak için bu konuyu kullanın.

Arşiv günlükleri için BSDS 'deki girdilere ekleme yapabilir, bu girdilerin parolasını silebilir ve değiştirebilirsiniz. Yalnızca örnekler burada gösterilir; gösterilen veri kümesi adlarını kullanmak istediğiniz adlarla değiştirin. Yardımcı programa ilişkin daha fazla ayrıntı için bkz. [Değişiklik günlüğü stok yardımcı programı](#).

- [Arşiv günlüğü eklenmesi](#)
- [Arşiv günlüğünün silinmesi](#)
- [Arşiv günlüğünün parolasının değiştirilmesi](#)

Arşiv günlüğü eklenmesi

Bir nesnenin kurtarılması, var olan bir arşiv günlüğü veri kümesinin okunmasına bağlı olduğunda, IBM MQ 'in bulabilmesi için BSDS' nin o veri kümesiyle ilgili bilgiler içermesi gerekir. BSDS 'de var olan bir arşiv günlüğü veri kümesine ilişkin bilgileri kaydetmek için aşağıdakileri kullanın:

```
NEWLOG DSN=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04,  
UNIT=TAPE,STARTRBA=3A190000,ENDRBA=3A1F0FFF,CATALOG=NO
```

Arşiv günlüğünün silinmesi

Bir ya da daha fazla birimdeki arşiv günlüğü veri kümesinin tamamını silmek için aşağıdakileri kullanın:

```
DELETE DSN=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04
```

Arşiv günlüğünün parolasının değiştirilmesi

Var olan bir arşiv günlüğü veri kümesinin parolasını değiştirirseniz, BSDS ' deki bilgileri de değiştirmeniz gerekir.

1. Günlük eşleme yazdırma yardımcı programını kullanarak BSDS ' yi listeleyin.
2. CSQJU003 yardımcı programının DELETE işlevini kullanarak, değiştirilen parolayla arşiv günlüğü veri kümesine ilişkin girişi silin ([Değişiklik günlüğü dökümü yardımcı programı](#) konusuna bakın).
3. Veri kümesini yeni bir arşiv günlüğü veri kümesi olarak adlandırın. CSQJU003 yardımcı programının NEWLOG işlevini kullanın (bkz. [Değişiklik günlüğü dökümü yardımcı programı](#)) ve yeni parolayı, başlangıç ve bitiş RNA ' larını ve birim seri numaralarını verin (günlük eşleme yazdırma yardımcı programı çıkışında bulunur, bkz. [Yazdırma günlüğü eşleme yardımcı programı](#)).


Yeni arşiv günlüğü veri kümelerinin parolasını değiştirmek için şunu kullanın:

```
ARCHIVE PASSWORD= password
```

Yeni arşiv günlüğü veri kümelerine parola yerleştirmeyi durdurmak için aşağıdakileri kullanın:

```
ARCHIVE NOPASSWD
```

Not: ARCHIVE yardımcı program işlevini yalnızca dış güvenlik yöneticiniz yoksa kullanın.

 **Günlükler ve BSDS için üst düzey niteleyicinin (HLQ) değiştirilmesi**

Üst düzey niteleyiciyi (HLQ) değiştirmek için gereken yordamı anlamak için bu konuyu kullanın.

Başlamadan önce

Günlüklerden ya da veri kümelerinden herhangi birini yeni veri kümelerine kopyalamadan önce kuyruk yöneticisini olağan şekilde sona erdirmeniz gerekir. Bunun nedeni, verilerin tutarlı olmasını ve yeniden başlatma sırasında kurtarma gerekmesini sağlamaktır.

Bu görev hakkında

Bu görev, günlüklere ve BSDS 'ye ilişkin HLQ' nun nasıl değiştirileceğine ilişkin bilgi sağlar. Bunu yapmak için aşağıdaki adımları izleyin:

Yordam

1. Günlük veri kümesi bilgilerini kaydetmek için CSQJU004 günlük yazdırma yardımcı programını çalıştırın. Bu bilgiler daha sonra gereklidir.
2. Aşağıdakilerden birini yapabilirsiniz:
 - a) Yeniden adlandırılacak günlük ve BSDS veri kümelerinde yeniden adlandırma ile DSS yedekleme ve geri yükleme işlemini çalıştırın ya da
 - b) HLQ veri kümelerini oluşturmak ve eski veri kümelerinden verileri kopyalamak için AMS DEFINE ve REPRO kullanın.
3. MSTR ve CHIN yordamlarını yeni veri kümelerini gösterecek şekilde değiştirin.
4. CSQJU003komutunu kullanarak BSDS ' nin yeni kopyasındaki eski günlük bilgilerini silin.
5. CSQJU003NEWLOG işlevini kullanarak yeni günlük veri kümelerini yeni BSDS ' ye tanımlayın. HLQ dışında her günlüğe ilişkin tüm bilgileri aynı tutun.
6. Yeni BSDS, eski BSDS ' deki eski günlükler için kaydedilen aynı bilgileri yansıtmalıdır. Değişen tek şey HLQ olmalı.

Sonraki adım

Kuyruk yöneticisini başlatmadan önce, eski ve yeni BSDS 'ye ilişkin CSQJU004 çıkışını karşılaştırarak, bunların TAM olarak aynı (HLQS' ler dışında) göründüklerinden emin olun.

Not: Bu işlemler gerçekleştirilirken dikkatli olunmalıdır. Yanlış işlemler kurtarılamayan durumlara neden olabilir. PRINT LOG MAP UTILITY çıkışını denetleyin ve kurtarma ya da yeniden başlatma için gereken tüm bilgilerin içerildiğinden emin olun.

BSDS ' nin kurtarılması

IBM MQ ikili BSDS kipinde çalışıyorsa ve bir BSDS zarar görürse, IBM MQ ' i tek BSDS kipine zorlayarak IBM MQ sorunsuz çalışmaya devam eder (bir sonraki yeniden başlatmaya kadar).

Ortamı ikili BSDS kipine döndürmek için:

1. Zarar görmüş BSDS ' yi yeniden adlandırmak ya da silmek ve zarar görmüş BSDS ile aynı adı taşıyan yeni bir BSDS tanımlamak için Erişim Yöntemi Hizmetleri 'ni kullanın. Örnek denetim deyimleri, thlqual.SCSQPROCiçindeki CSQ4BREC işinde bulunabilir.
2. Yeni ayrılan veri kümesinde geçerli BSDS ' nin bir kopyasını oluşturmak ve ikili BSDS kipini yeniden oluşturmak için IBM MQ komutunu RECOVER BSDS komutunu verin.

IBM MQ tek BSDS kipinde çalışıyorsa ve BSDS zarar görmüşse ya da IBM MQ ikili BSDS kipinde çalışıyorsa ve her iki BSDS zarar görmüşse, kuyruk yöneticisi durur ve BSDS veri kümeleri onarıncaya kadar yeniden başlatılmaz. Bu durumda:

1. En son arşiv günlüğü veri kümesiyle ilişkilendirilmiş BSDS ' yi bulun. En son arşiv günlüğünün veri kümesi adı, CSQJ003Iiletisinin son oluşumunda iş günlüğünde görünür; bu, boşaltma işleminin başarıyla tamamlandığını gösterir. Bu yordamın geri kalanına hazırlık olarak, bu iletide belirtilen tüm başarılı arşivlerin kaydını tutmak iyi bir uygulamadır.
 - Arşiv günlükleri DASD üzerindeyse, BSDS kullanılabilir herhangi bir DASD üzerinde ayrılır. BSDS adı, karşılık gelen arşiv günlüğü veri kümesi adı gibidir; aşağıdaki örnekte olduğu gibi, A 'dan B' ye son niteleyicinin yalnızca ilk harfini değiştirin:

Arşiv günlüğü adı

CSQ.ARCHLOG1. **A** 0000001

BSDS kopyası adı

CSQ.ARCHLOG1. **B** 0000001

- Arşiv günlükleri manyetik banttaysa, BSDS ilk arşiv günlüğü biriminin ilk veri takısıdır. BSDS, sonraki birimlerde yinelenmez.
2. En son arşiv günlüğü veri kümesinin BSDS 'nin kopyası yoksa (örneğin, boşaltılırken bir hata oluştuğu için), daha önceki boşaltma işlemlerinden BSDS' nin önceki bir kopyasını bulun.

3. Erişim Yöntemi Hizmetleri ALTER komutunu NEWNAME seçeneğiyle kullanarak *hasarlı* BSDS'lerini yeniden adlandırın. Zarar görmüş bir BSDS ' yi silmek istiyorsanız, Access Method Services DELETE komutunu kullanın. Zarar görmüş her BSDS için, yeni bir BSDS ' yi değiştirme veri kümesi olarak tanımlamak üzere Erişim Yöntemi Hizmetleri 'ni kullanın. thlqual.SCSQPROC içindeki CSQ4BREC işi, yeni bir BSDS tanımlamak için Erişim Yöntemi Hizmetleri denetim deyimlerini içerir.
4. BSDS ' yi arşiv günlüğünden “3” sayfa 516. adımda tanımladığınız yeni BSDS'lerden birine kopyalamak için Erişim Yöntemi Hizmetleri REPRO komutunu kullanın. İkinci yedek BSDS ' ye herhangi bir veri kopyalamayın, bunu “5” sayfa 517. adımda yaparsınız.

- a. Yeni BSDS ' nin içeriğini yazdırın.

Yeni BSDS ' nin içeriğini yazdırmak için günlük eşlemi yazdırma yardımcı programını (CSQJU004) kullanın. Bu, kurtarma işine devam etmeden önce yeni BSDS ' nin içeriğini gözden geçirmenizi sağlar.

- b. Yeni BSDS ' deki arşiv günlüğü veri kümesi dökümünü güncelleyin.

Yazdırma günlüğü eşlemi yardımcı programının çıkışını inceleyin ve yeni BSDS 'nin BSDS' nin kopyalandığı arşiv günlüğünün kaydını içermediğini doğrulayın. Yeni BSDS eski bir kopyaysa, dökümü daha yakın zamanda oluşturulan tüm arşiv günlüğü veri kümelerini içermeyebilir. Arşiv günlüğü veri kümelerinin BSDS dökümü, geçerli altsistem dökümünü yansıtacak şekilde güncellenmelidir.

BSDS 'nin kopyalandığı arşiv günlüğüne ilişkin bir kayıt ekleyerek, yeni BSDS' yi güncellemek için değişiklik günlüğü dökümü yardımcı programını (CSQJU003) kullanın. Arşiv günlüğü veri kümesi parola korumalı ise, NEWLOG işlevinin PASSWORD seçeneğini kullanın. Ayrıca, arşiv günlüğü veri kümesi kataloğa alındıysa, NEWLOG işlevinin CATALOG seçeneğinin doğru olarak CATALOG=YES olarak ayarlandığından emin olun. BSDS kopyasından daha sonra yaratılan diğer arşiv günlüğü veri kümelerini eklemek için NEWLOG deyimini kullanın.

- c. Yeni BSDS ' deki parolaları güncelleyin.

BSDS, arşiv günlüğü veri kümeleri ve etkin günlük veri kümeleri için parolalar içerir. Yeni BSDS ' deki parolaların kuruluşunuz tarafından kullanılan geçerli parolaları yansıttığından emin olmak için, değişiklik günlüğü dökümü ARCHIVE yardımcı programı işlevini PASSWORD seçeneğiyle kullanın.

- d. Yeni BSDS ' de etkin günlük veri kümesi dökümünü güncelleyin.

Olağan dışı durumlarda, kuruluşunuz BSDS kopyalandıktan sonra etkin günlük veri kümelerini eklemiş, silmiş ya da yeniden adlandırmış olabilir. Bu durumda, yeni BSDS, kuruluşunuzda kullanılmakta olan etkin günlük veri kümelerinin gerçek sayısını ya da adlarını yansıtmaz.

Yeni BSDS günlük dökümünden etkin bir günlük veri kümesini silmeniz gerekirse, değişiklik günlüğü stok yardımcı programı DELETE işlevini kullanın.

Yeni BSDS günlük dökümüne etkin bir günlük veri kümesi eklemeniz gerekiyorsa, değişiklik günlüğü döküm yardımcı programı NEWLOG işlevini kullanın. RBA aralığının NEWLOG işlevinde doğru belirtildiğinden emin olun. Etkin günlük veri kümesi parola korumalı ise, PASSWORD seçeneğini kullanın.

Yeni BSDS günlük dökümünde etkin bir günlük veri kümesini yeniden adlandırmanız gerekirse, değişiklik günlüğü döküm yardımcı programı DELETE işlevini ve ardından NEWLOG işlevini kullanın. RBA aralığının NEWLOG işlevinde doğru belirtildiğinden emin olun. Etkin günlük veri kümesi parola korumalı ise, PASSWORD seçeneğini kullanın.

- e. Yeni BSDS ' deki etkin günlük RBA aralıklarını güncelleyin.

Daha sonra, kuyruk yöneticisi yeniden başlatıldığında, BSDS 'de listelenen etkin günlük veri kümelerinin RNA' larını gerçek etkin günlük veri kümelerinde bulunan RNA ' larla karşılaştırır. RNA ' lar kabul etmezse, kuyruk yöneticisi yeniden başlatılmaz. BSDS ' nin eski bir kopyası kullanıldığında sorun büyütülür. Bu sorunu çözmek için, günlük dökümünü değiştirme yardımcı programını (CSQJU003) kullanarak gerçek etkin günlük veri kümelerindeki RNA 'ları kullanarak BSDS' de bulunan RNA ' ları ayarlayın. Bunu şu şekilde yaparsınız:

- Etkin günlük veri kümesinin özet raporunu yazdırmak için günlük kayıtlarını yazdırma yardımcı programını (CSQ1LOGP) kullanma. Bu, başlangıç ve bitiş RNA ' larını gösterir.
- Tüm etkin günlük veri kümelerinin RNA ' ları bilindiğinde, gerçek RBA aralıklarının az önce yazdığınız RBA aralıklarıyla karşılaştırılması.

RBA aralıkları tüm etkin günlük veri kümeleri için eşitse, ek iş olmadan sonraki kurtarma adımına geçebilirsiniz.

RBA aralıkları eşit değilse, BSDS ' deki değerleri gerçek değerleri yansıtacak şekilde ayarlayın. RBA aralığının ayarlanması gereken her etkin günlük veri kümesi için, yeni BSDS ' deki dökümden etkin günlük veri kümesini silmek için değişiklik günlüğü stok yardımcı programı DELETE işlevini kullanın. Daha sonra, etkin günlük verileri kümesini BSDS olarak yeniden tanımlamak için NEWLOG işlevini kullanın. Etkin günlük veri kümeleri parola korumalı ise, NEWLOG işlevinin PASSWORD seçeneğini kullanın.

- f. Etkin günlüğün her bir kopyası için yalnızca iki etkin günlük veri kümesi belirtilirse, IBM MQ kuyruk yöneticisi yeniden başlatılırken zorluk çekebilir. Etkin günlük veri kümelerinden biri dolu olduğunda ve boşaltılmadığında, ikinci etkin günlük veri kümesi dolmaya yaklaştığında sorun ortaya çıkabilir. Bu durumda, etkin günlüğün her bir kopyası için yeni bir etkin günlük veri kümesi ekleyin ve yeni BSDS günlük dökümünde her bir yeni etkin günlük veri kümesini tanımlayın.

Etkin günlüğün her bir kopyası için yeni bir etkin günlük veri kümesi tanımlamak için Erişim Yöntemi Hizmetleri DEFINE komutunu kullanın ve yeni BSDS ' deki yeni etkin günlük veri kümelerini tanımlamak için değişiklik günlüğü döküm yardımcı programı NEWLOG işlevini kullanın. NEWLOG deyiminde RBA aralıklarını belirtmenize gerek yoktur. Ancak, etkin günlük veri kümeleri parola ile korunuyorsa, NEWLOG işlevinin PASSWORD seçeneğini kullanın. Bu görevi gerçekleştirmeye ilişkin örnek denetim deyimleri, thlqual.SCSQPROÇ içindeki CSQ4LREC içinde bulunabilir.

5. Güncellenen BSDS ' yi ikinci yeni BSDS veri kümesine kopyalayın. BSDSler artık aynı.

Bu noktada, ikinci değiştirme BSDS ' nin içeriğini yazdırmak için günlük eşlemi yazdırma yardımcı programını (CSQJU004) kullanın.

6. Geçerli etkin günlük veri kümenizi kaybederseniz ne yapacağınız hakkında bilgi için [Etkin günlük sorunları](#) başlıklı konuya bakın.

7. Yeni oluşturulan BSDS ' yi kullanarak kuyruk yöneticisini yeniden başlatın. IBM MQ , geçerli RBA ' yı ve hangi etkin günlüklerin arşivlenmesi gerektiğini belirler.

Sayfa kümelerini yönetme

Bir kuyruk yöneticisiyle ilişkili sayfa kümelerinin nasıl yönetileceğini anlamak için bu konuyu kullanın.

Bu konuda, bir kuyruk yöneticisiyle ilişkili sayfa kümelerinin nasıl ekleneceği, kopyalanacağı ve genel olarak yönetileceği açıklanmaktadır. Aşağıdaki kısımları içerir:

- [“Sayfa kümeleri için üst düzey niteleyicinin \(HLQ\) değiştirilmesi” sayfa 518](#)
- [“Kuyruk yöneticisine sayfa kümesi eklenmesi” sayfa 518](#)
- [“Sayfa kümelerinden biri dolduğunda ne yapmanız gerekir?” sayfa 518](#)
- [“Sayfa kümelerindeki yükleri dengeleme” sayfa 519](#)
- [Sayfa kümesinin büyüklüğünü artırma](#)
- [“Sayfa kümesinin küçültülmesi” sayfa 523](#)
- [“Bir sayfa kümesinin yeniden tanıtılması” sayfa 523](#)
- [“Sayfa kümelerinin yedeklanması ve kurtarılması” sayfa 524](#)
- [“Sayfa kümelerinin silinmesi” sayfa 527](#)
- [“CSQUTIL kullanılarak kuyrukların yedeklenmesi ve geri yüklenmesi” sayfa 528](#)

Sayfa kümeleri, depolama sınıfları, arabellekler ve arabellek havuzlarına ilişkin açıklamalar ve geçerli olan bazı başarımlar için [Sayfa kümeleri](#) başlıklı konuya bakın.

Sayfa kümeleri için üst düzey niteleyicinin (HLQ) değiştirilmesi

Bu görev, sayfa kümeleri için HLQ ' nun nasıl değiştirileceğine ilişkin bilgi verir. Bu görevi gerçekleştirmek için aşağıdakileri yapın:

1. Yeni HLQ sayfa kümelerini tanımlayın.
2. Boyut ayırma eski sayfa kümeleriyle aynıysa, REPRO kullanarak var olan sayfa kümesini boş yeni HLQ sayfa kümelerine kopyalayın.
3. Sayfa kümelerinin büyüklüğünü artırıyorsanız, hedef sayfaları biçimlendirmek için CSQUTIL 'in FORMAT işlevini ve kaynak sayfa kümesindeki tüm iletileri hedef sayfa kümesine kopyalamak için CSQUTIL' in COPYPAGE işlevini kullanın.
Daha fazla bilgi için bkz. [Sayfa kümelerini biçimlendirme \(FORMAT\)](#) ve [Sayfa kümesini genişletme \(COPYPAGE\)](#).
4. Kuyruk yöneticisi yordamındaki CSQP00xx DD deyimini, yeni HLQ sayfa kümelerini gösterecek şekilde değiştirin.

Kuyruk yöneticisini yeniden başlatın ve sayfa kümelerindeki değişiklikleri doğrulayın.

Kuyruk yöneticisine sayfa kümesi eklenmesi

Bu tanımlama, çalışmakta olan bir kuyruk yöneticiniz olduğunu varsayar. Örneğin, kuyruk yöneticinizin yeni kuyrukları kullanan yeni uygulamalarla başa çıkması gerekiyorsa, bir sayfa kümesi eklemeniz gerekebilir.

Yeni bir sayfa kümesi eklemek için aşağıdaki yordamı kullanın:

1. Yeni sayfa kümesini tanımlayın ve biçimlendirin. Örnek JCL ' yi thlqual.SCSQPROC(CSQ4PAGE) içinde temel olarak kullanabilirsiniz. Daha fazla bilgi için bkz. [Sayfa kümelerini biçimlendirme \(FORMAT\)](#).
Amaçladığınız bu değilse, kullanılmakta olan sayfa kümelerini biçimlendirmemeye dikkat edin. Bu durumda, FORMAT yardımcı program işlevinin FORCE seçeneğini kullanın.
2. Sayfa kümesini bir arabellek havuzuyla ilişkilendirmek için DEFINE PSID komutunu DSN seçeneğiyle kullanın.
3. DEFINE STGCLASS komutlarını vererek, sayfa kümeniz için uygun depolama sınıfı tanımlamalarını ekleyin.
4. İsteğe bağlı olarak, kuyruk yöneticinizin nasıl yapılandırıldığını belgelemek için:
 - a. Yeni sayfa kümesini, kuyruk yöneticiniz için başlatılan görev yordamına ekleyin.
 - b. CSQINP1 kullanıma hazırlama veri kümenize yeni sayfa kümesi için bir tanımlama ekleyin.
 - c. CSQ4INYP kullanıma hazırlama veri kümesi üyenize yeni depolama sınıfı için bir tanımlama ekleyin.

DEFINE PSID ve DEFINE STGCLASS komutlarına ilişkin ayrıntılar için [DEFINE PSID](#) ve [DEFINE STGCLASS](#) başlıklı konuya bakın.

Sayfa kümelerinden biri dolduğunda ne yapmanız gerekir?

IBM MQ DISPLAY USAGE komutunu kullanarak sayfa kümelerinin kullanımı hakkında bilgi alabilirsiniz. Örneğin, komut:

```
DISPLAY USAGE PSID(03)
```

Sayfa takımı 03 'ün yürürlükteki durumunu görüntüler. Bu, bu sayfa kümesinin kaç tane boş sayfası olduğunu gösterir.

Sayfa kümeleriniz için ikincil kapsamlar tanımladıysanız, bunlar her dolduruldukları zaman dinamik olarak genişletilir. Sonuç olarak, tüm ikincil kapsamlar kullanılabilir ya da başka disk yeri yoktur. Bu durumda, bir uygulama MQRC_STORAGE_MEDIUM_FULL dönüş kodunu alır.

Bir uygulama MQI çağrısından MQRC_STORAGE_MEDIUM_FULL dönüş kodunu alırsa, bu, sayfa kümesinde yeterli yer kalmadığını gösterir. Sorun devam ederse ya da yeniden oluşma olasılığı varsa, sorunu çözmek için bir şey yapmanız gerekir.

Bu soruna çeşitli yollarla yaklaşabilirsiniz:

- Kuyrukları bir sayfa kümesinden diğerine taşıyarak sayfa kümeleri arasındaki yükü dengeleyebilirsiniz.
- Sayfa kümesini genişletin. Yönergeler için bkz. [“Sayfa kümesinin boyutunu artırma” sayfa 521](#) .
- Sayfa kümesini, 4 GB 'den 64 GB' ye kadar genişletilecek şekilde yeniden tanımlayın. Yönergeler için [4 GB ' den büyük bir sayfa kümesi tanımlama](#) başlıklı konuya bakın.

Sayfa kümelerindeki yükleri dengeleme

Sayfa kümelerinde yük dengeleme, bir ya da daha çok kuyrukla ilişkili iletilerin bir sayfa kümesinden başka bir sayfa kümesine taşınması, daha az kullanılan sayfa kümesine taşınması anlamına gelir. Sayfa kümesini genişletmek pratik değilse bu tekniği kullanın.

Hangi kuyrukların sayfa kümesi kullandığını tanımlamak için uygun IBM MQ komutlarını kullanın. Örneğin, hangi kuyrukların sayfa kümesi 02 ile eşlendiğini öğrenmek için, önce şu komutu kullanarak sayfa kümesi 02 ile hangi depolama sınıflarının eşlendiğini bulun:

```
DISPLAY STGCLASS(*) PSID(02)
```

Daha sonra hangi kuyrukların hangi depolama sınıfını kullandığını öğrenmek için aşağıdaki komutu kullanın:

```
DISPLAY QUEUE(*) TYPE(QLOCAL) STGCLASS
```

Paylaşılmayan bir kuyruğun taşınması

Kuyrukları ve iletilerini bir sayfa kümesinden diğerine taşımak için MQSC TAŞMA QLOCAL komutunu kullanın ([TAŞMA QLOCAL](#) başlıklı konuda açıklanmaktadır). Yeni bir sayfa kümesine taşımak istediğiniz kuyruğu ya da kuyrukları tanımladığınızda, bu kuyrukların her biri için aşağıdaki yordamı izleyin:

1. Taşımak istediğiniz kuyruğun hiçbir uygulama tarafından kullanılmadığını (yani, DISPLAY QSTATUS komutundaki IPPROCS ve OPPROCS değerlerinin sıfır olduğunu) ve kesinleştirilmemiş iletileri olmadığını (DISPLAY QSTATUS komutundaki UNCOM değeri NO) doğrulayın.

Not: Bu durumun devam etmesini sağlamanın tek yolu, kuyruğun güvenlik yetkisini geçici olarak değiştirmektir. Ek bilgi için [Kuyruk güvenliği için tanımlar](#) konusuna bakın.

Bunu yapamazsanız, uygulamalar PUT (DISABLED) ayarı gibi önlem adımlarına rağmen kuyruğu kullanmaya başlarsa, bu yordamdaki sonraki aşamalar başarısız olabilir. Ancak, bu yordam iletileri hiçbir zaman kaybetmez.

2. MQPUT ' leri geçersiz kılmak için kuyruk tanımlamasını değiştirerek, uygulamaların taşınmakta olan kuyruğa ileti koymasını önleyin. Kuyruk tanımlamasını PUT (DISABLED) olarak değiştirin.
3. Şu komutu kullanarak, taşınmakta olan kuyrukla aynı özniteliklere sahip geçici bir kuyruk tanımlayın:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
```

Not: Bu geçici kuyruk önceki bir çalıştırmadan önceden varsa, tanımlamadan önce silin.

4. Aşağıdaki komutu kullanarak iletileri geçici kuyruğa taşıyın:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Taşımakta olduğunuz kuyruğu silmek için şu komutu kullanın:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Gerekli sayfa kümesiyle eşlenen yeni bir depolama sınıfı tanımlayın; örneğin:

```
DEFINE STGCLASS(NEW) PSID(nn)
```

Yeni depolama sınıfı tanımlamasını, CSQINP2 veri kümelerine sonraki kuyruk yöneticisi yeniden başlatması için hazır olarak ekleyin.

7. Depolama sınıfı özneliğini değiştirerek, taşımakta olduğunuz kuyruğu yeniden tanımlayın:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW)
```

Kuyruk yeniden tanımlandığında, [“3” sayfa 519](#). adımda yaratılan geçici kuyruğa dayalıdır.

8. Aşağıdaki komutu kullanarak iletileri yeni kuyruğa geri taşıyın:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

9. [“3” sayfa 519](#) . adımda yaratılan kuyruk artık gerekli değil. Silmek için aşağıdaki komutu kullanın:

```
DELETE QL(TEMP_QUEUE)
```

10. Taşınmakta olan kuyruk CSQINP2 veri kümelerinde tanımlandıysa, CSQINP2 veri kümelerinde uygun DEFINE QLOCAL komutunun STGCLASS özneliğini değiştirin. REPLACE anahtar sözcüğünü ekleyerek, varolan kuyruk tanımlamasının yerine yenisini koyun.

[Şekil 29 sayfa 521](#) , bir yük dengeleme işinden alınan bir çıkarmayı gösterir.


```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
// DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_To_Move) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_To_Move)
DEFINE STGCLASS(NEW) PSID(2)
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_To_Move)
DELETE QL(TEMP_QUEUE)
/*

```

Şekil 29. Bir sayfa kümesine ilişkin yük dengeleme işinden çıkar

Sayfa kümesinin boyutunu artırma

Başlangıçta 4 GB 'den büyük bir sayfa kümesi ayırabilirsiniz, bkz. [4 GB' den büyük bir sayfa kümesi tanımlama](#)

Bir sayfa kümesi, EXPAND (SYSTEM) ya da EXPAND (USER) belirtilerek doldukça otomatik olarak genişletilecek şekilde tanımlanabilir. Sayfa kümeniz EXPAND (NONE) ile tanımlandıysa, iki yoldan biriyle genişletebilirsiniz:

- Tanımlamasını otomatik genişletmeye izin verecek şekilde değiştirin. Bkz. [Otomatik genişletmeye izin vermek için sayfa kümesinin değiştirilmesi](#)
- Yeni, daha büyük bir sayfa kümesi oluşturun ve iletileri eski sayfa kümesinden yeni sayfaya kopyalayın. Bkz. [İletilerin yeni, daha büyük bir sayfa kümesine taşınması](#)

4 GB ' den büyük bir sayfa kümesi tanımlama

IBM MQ , veri kümesi VSAM için 'genişletilmiş adreslenirlik' ile tanımlandıysa, 64 GB ' ye kadar ayarlanmış bir sayfa kullanabilir. Genişletilmiş adreslenebilirlik, bir SMS veri sınıfı tarafından verilen bir özneliktir.

Not: Sayfa kümeleri ve etkin günlük veri kümeleri, genişletilmiş adres birimlerinin (EAV) genişletilmiş adresleme alanı (EAS) bölümünde bulunmaya hak kazanır ve z/OS V1.12' den bir arşiv günlüğü veri kümesi de EAS içinde yer alabilir.

Aşağıdaki örnek JCL 'de gösterilen örnekte,' EXTENDED 'yönetim sınıfı' Extended addressability 'ile SMS' ye tanımlanmıştır. Var olan sayfa kümeniz genişletilmiş adreslenirliğe sahip olarak tanımlanmamışsa, genişletilmiş adreslenirlik biçimi veri kümesine geçmek için aşağıdaki yöntemi kullanın.

1. Kuyruk yöneticisini durdurun.
2. Var olan sayfa kümesini yeniden adlandırmak için Erişim Yöntemi Hizmetleri 'ni kullanın.
3. Var olan sayfa kümesiyle aynı boyutta, ancak DATACLAS (EXTENDED) ile bir hedef sayfa kümesi tanımlayın.

Not: Genişletilmiş biçimli veri kümeleri SMS tarafından yönetilmelidir. Bunlar, VSAM veri kümeleri için genişletilmiş biçim isteme mekanizmalarıdır:

- DSNTYPE değeri EXT olan bir veri sınıfı ve gerekli ya da tercih edilen değeri belirtmek için alt parametre R ya da P kullanılarak.

- DD deyiminde DSNTYPE=RADIO (geniřletilmiř biçim gereklidir) ya da DSNTYPE=EXTPREF (geniřletilmiř biçim tercih edilir) kodlaması.
- DD deyimindeki LIKE= parametresinin, var olan bir geniřletilmiř biçim veri kümesine gönderme yapmak üzere kodlanması.

Daha fazla bilgi için bakınız: [Restrictions on Defining Extended-Format Data Sets](#).

4. Kaynak sayfa kümesindeki tüm iletileri hedef sayfa kümesine kopyalamak için CSQUTIL ' in COPYPAGE işlevini kullanın. Daha fazla ayrıntı için bkz. [Sayfa kümesini geniřletme \(COPYPAGE\)](#) .
5. Kuyruk yöneticisini yeniden başlatın.
6. Sayfa kümesini, sistem geniřletmesini kullanacak şekilde deęiřtirin ve yürürlükteki ayırmanın ötesine büyümeye devam etmesini saęlayın.

Ařaęıdaki JCL, örnek Eriřim Yöntemi Hizmetleri komutlarını gösterir:

```
//S1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALTER 'VICY.CSQ1.PAGE01' -
NEWNAME('VICY.CSQ1.PAGE01.OLD')
ALTER 'VICY.CSQ1.PAGE01.DATA' -
NEWNAME('VICY.CSQ1.PAGE01.DATA.OLD')
DEFINE CLUSTER (NAME('VICY.CSQ1.PAGE01') -
MODEL('VICY.CSQ1.PAGE01.OLD') -
DATACLAS(EXTENDED))
/*
```

Sayfa kümesinin otomatik geniřletmeye izin verecek şekilde deęiřtirilmesi

ALTER PSID komutunu EXPAND (USER) ya da EXPAND (SYSTEM) seçenekleriyle birlikte kullanın. Sayfa kümelerinin geniřletilmesiyle ilgili genel bilgi için [ALTER PSID](#) ve [Sayfa kümesinin geniřletilmesi \(COPYPAGE\)](#) bařlıklı konuya bakın.

İletilerin yeni, daha büyük bir sayfa kümesine tařınması

Bu teknik, kuyruk yöneticisinin durdurulmasını ve yeniden bařlatılmasını içerir. Bu, yeniden bařlatma sırasında paylařılan kuyruklarda olmayan kalıcı olmayan iletileri siler. Silinmesini istemediđiniz kalıcı olmayan iletileriniz varsa, bunun yerine yük dengelemeyi kullanın. Daha fazla ayrıntı için bkz. [“Sayfa kümelerindeki yükleri dengeleme” sayfa 519](#). Bu tanımda, geniřletmek istediđiniz sayfa kümesine *kaynak* sayfa kümesi denir; yeni, daha büyük sayfa kümesine *hedef* sayfa kümesi denir.

Ařaęıdaki adımları izleyin:

1. Kuyruk yöneticisini durdurun.
2. Hedef sayfa kümesini, daha büyük bir ikincil kapsam deęeriyle kaynak sayfa kümesinden daha büyük olduđundan emin olarak tanımlayın.
3. Hedef sayfa kümesini biçimlendirmek için CSQUTIL ' in FORMAT işlevini kullanın. Daha fazla ayrıntı için bkz. [Sayfa kümelerini biçimlendirme \(FORMAT\)](#) .
4. Kaynak sayfa kümesindeki tüm iletileri hedef sayfa kümesine kopyalamak için CSQUTIL ' in COPYPAGE işlevini kullanın. Daha fazla ayrıntı için bkz. [Sayfa kümesini geniřletme \(COPYPAGE\)](#) .
5. Ařaęıdakilerden birini yaparak, hedef sayfa kümesini kullanarak kuyruk yöneticisini yeniden başlatın:
 - Kuyruk yöneticisi tarafından bařlatılan görev yordamını, hedef sayfa kümesine gönderme yapacak şekilde deęiřtirin.
 - Kaynak sayfa kümesini silmek için Eriřim Yöntemi Hizmetleri 'ni kullanın ve hedef sayfa kümesini, kaynak sayfa kümesiyle aynı adı vererek yeniden adlandırın.

Dikkat:

Herhangi bir IBM MQ sayfa kümesini silmeden önce, gerekli yedek kopyaları oluřturmuř olduđunuzdan emin olun.

Sayfa kümesinin küçültülmesi

IBM MQ yöneticisi dışındaki tüm kullanıcıların kuyruk yöneticisini kullanmasını önleyin. Örneğin; erişim güvenliği ayarlarını değiştirerek.

Çoğunlukla boş olan büyük bir sayfa kümeniz varsa (DISPLAY USAGE komutunda gösterildiği gibi), boyutunu azaltmak isteyebilirsiniz. Bunu yapma yordamı, CSQUTIL ' in COPY, FORMAT ve LOAD işlevlerinin kullanılmasını içerir (bkz. [IBM MQ yardımcı program](#)). Bu sayfa kümesinin boyutunu küçültmek pratik olmadığından, bu yordam sıfır (0) sayfa kümesi için çalışmaz; bunu yapmanın tek yolu kuyruk yöneticisini yeniden başlatmaktır (bkz. [“Kuyruk yöneticisini yeniden başlatma” sayfa 546](#)). Bu yordamın önkoşulu, tüm UOW ' lerin tamamlanmasını ve sayfa kümelerinin tutarlı olmasını sağlamak için tüm kullanıcıları sistemden kaldırmayı denemektir.

1. Kuyruk yöneticisini durdurmak için QUIESCE ya da FORCE özniteliğiyle STOP QMGR komutunu kullanın.
2. Büyük sayfa kümesindeki tüm ileti verilerini kopyalamak ve bunları sıralı bir veri kümesine saklamak için, CSQUTIL ' in SKOPI işlevini PSID seçeneğiyle çalıştırın.
3. Büyük sayfa kümesini değiştirmek için yeni bir daha küçük sayfa kümesi veri kümesi tanımlayın.
4. [“3” sayfa 523](#). adımda yarattığınız sayfa kümesinde CSQUTIL ' in FORMAT TYPE (NEW) işlevini çalıştırın.
5. [“3” sayfa 523](#). adımda yaratılan sayfa kümesini kullanarak kuyruk yöneticisini yeniden başlatın.
6. [“2” sayfa 523](#). adım sırasında saklanan tüm iletileri geri yüklemek için CSQUTIL ' in LOAD işlevini çalıştırın.
7. Tüm kullanıcıların kuyruk yöneticisine erişmesine izin verin.
8. Eski büyük sayfa kümesini silin.

Bir sayfa kümesinin yeniden tanıtılması

Bazı senaryolarda, eski bir sayfa kümesini yeniden kuyruk yöneticisine çevrimiçi duruma getirebilmek yararlıdır. Belirli bir işlem yapılmazsa, eski sayfa kümesi çevrimiçi duruma getirildiğinde kuyruk yöneticisi, sayfa ayarında ve denetim noktası kayıtlarında saklanan sayfa kümesi kurtarma RBA ' nın eski olduğunu tanır ve bu nedenle, sayfayı güncel duruma getirmek için ayarlanan sayfanın ortam kurtarmasını otomatik olarak başlatır.

Bu tür bir ortam kurtarma işlemi yalnızca kuyruk yöneticisi yeniden başlatıldığında gerçekleştirilebilir ve özellikle manyetik bantta tutulan arşiv günlüklerinin okunması gerekiyorsa çok uzun sürebilir. Ancak, normalde bu durumda, sayfa kümesi müdahale süresi için çevrimdışı olmuştur ve bu nedenle günlük, sayfa kümesi kurtarmasıyla ilgili herhangi bir bilgi içermez.

Aşağıdaki üç seçenek kullanılabilir:

Tam ortam kurtarma işleminin gerçekleştirilmesine izin verin.

1. Kuyruk yöneticisini durdurun.
2. Hem kuyruk yöneticisi için başlatılan görev yordamında hem de CSQINP1 kullanıma hazırlama veri kümesinde sayfa kümesi için tanımlamaların bulunduğunu doğrulayın.
3. Kuyruk yöneticisini yeniden başlatın.

Sayfa kümesindeki iletilerin yok edilmesine izin verin.

Bu seçenek, bir sayfa kümesinin uzun bir süre (örneğin, birkaç ay) çevrimdışı olduğu ve artık farklı bir amaç için yeniden kullanılmasına karar verildiği durumlarda kullanışlıdır.

1. Sayfa kümesini, TYPE (NEW) seçeneğiyle CSQUTIL ' in FORMAT işlevini kullanarak biçimlendirin.
2. Sayfa kümesine ilişkin tanımlamaları hem kuyruk yöneticisi için başlatılan görev yordamına, hem de CSQINP1 kullanıma hazırlama veri kümesine ekleyin.
3. Kuyruk yöneticisini yeniden başlatın.

Biçimleme için TYPE (NEW) seçeneğinin kullanılması, sayfa kümesinin yürürlükteki içeriğini temizler ve kuyruk yöneticisine, sayfa kümesiyle ilgili denetim noktasındaki geçmiş bilgileri yoksaymasını bildirir.

Ortam kurtarma işlemini önleyerek sayfa kümesini çevrimiçi duruma getirin.

Bu tekniği yalnızca, kuyruk yöneticisinin temiz bir şekilde kapatılmasından bu yana sayfa kümesinin çevrimdışı olduğundan eminseniz kullanın. Bu seçenek, genellikle kuyruk yöneticisi başlatılırken yedeklemenin çalıştırılması gibi işletim sorunları nedeniyle, sayfa kümesinin kısa bir süre çevrimdışı kaldığı durumlarda uygundur.

1. Sayfa kümesini, TYPE (REPLACE) seçeneğiyle CSQUTIL ' in FORMAT işlevini kullanarak biçimlendirin.
2. Sayfa kümesini, DSN seçeneğiyle DEFINE PSID komutunu kullanarak kuyruk yöneticisine dinamik olarak ekleyin ya da kuyruk yöneticisi yeniden başlatıldığında eklenmesine izin verin.

Biçimlendirmek için TYPE (REPLACE) seçeneğinin kullanılması, sayfa kümesinin kuyruk yöneticisi tarafından düzgün bir şekilde kapatılıp kapatılmadığını denetler ve ortam kurtarma işleminin gerçekleştirilmemesi için bunu işaretler. Sayfa kümesinin içeriğinde başka değişiklik yapılmaz.

Sayfa kümelerinin yedeklanması ve kurtarılması

Yedekleme ve kurtarma için farklı mekanizmalar vardır. Bu mekanizmaları anlamak için bu konuyu kullanın.

Bu bölümde aşağıdaki konular açıklanmaktadır:

- [“Paylaşılmayan kaynaklar için kurtarma noktası yaratılması” sayfa 524](#)
- [“Sayfa kümelerini yedekleme” sayfa 525](#)
- [“Sayfa kümeleri kurtarılıyor” sayfa 526](#)
- [Sayfa kümelerinin silinmesi](#)

Paylaşılan kaynaklar için bir kurtarma noktası oluşturma hakkında bilgi için bkz. [“Paylaşılan kuyrukların kurtarılması” sayfa 533](#).

Paylaşılmayan kaynaklar için kurtarma noktası yaratılması

IBM MQ , aşağıdaki durumlarla birlikte nesnelere ve paylaşılmayan kalıcı iletileri geçerli durumlarına kurtarabilir:

1. Daha önceki bir noktadan sayfa kümelerinin kopyaları var.
2. Bu noktadan itibaren kurtarma gerçekleştirmek için tüm IBM MQ günlükleri kullanılabilir.

Bunlar, paylaşılmayan kaynaklar için bir kurtarma noktasını gösterir.

Hem nesnelere hem de iletiler sayfa kümelerinde tutulur. Aynı sayfa kümesinde farklı kuyruklardan birden çok nesne ve ileti bulunabilir. Kurtarma amacıyla, nesnelere ve iletiler yalıtımda yedeklenemez; bu nedenle, verilerin doğru şekilde kurtarılmasını sağlamak için bir sayfa kümesinin bir bütün olarak yedeklenmesi gerekir.

IBM MQ kurtarma günlüğü, nesnelere yapılan tüm kalıcı iletilerin ve değişikliklerin kaydını içerir. IBM MQ başarısız olursa (örneğin, bir sayfa kümesindeki bir G/Ç hatası nedeniyle), yedek kopyayı geri yükleyip kuyruk yöneticisini yeniden başlatarak sayfa kümesini kurtarabilirsiniz. IBM MQ , günlük değişikliklerini yedek kopyanın noktasından sayfa kümesine uygular.

Kurtarma noktası yaratmanın iki yolu vardır:

Tam Olarak Yedekle

Tüm güncellemeleri sayfa kümelerinde zorlayan kuyruk yöneticisini durdurun.

Bu, yalnızca yedeklenen sayfa kümesi veri kümelerini ve o noktadan itibaren günlükleri kullanarak kurtarma noktasından yeniden başlatmanızı sağlar.

Yaklaşık yedekleme

Kuyruk yöneticisini durdurmadan sayfa kümelerinin *bulanık* yedek kopyalarını alın.

Bu yöntemi kullanırsanız ve ilişkili günlükleriniz daha sonra zarar görürse ya da kaybolursa, kurtarmak için yaklaşık sayfa kümesi yedek kopyalarını kullanamazsınız. Bunun nedeni, yaklaşık sayfa kümesi yedek kopyalarının kuyruk yöneticisinin durumunun tutarsız bir görünümünü içermesi ve kullanılabilir günlüklere bağımlı olmasıdır. Günlükler kullanılmıyorsa, altsistem etkin değilken alınan son yedek sayfa kümesi kopyaları kümesine dönmeniz gerekir ([Yöntem 1](#)) ve o zamandan kalma veri kaybını kabul edin.

Yöntem 1: Tam yedekleme

Bu yöntem, kuyruk yöneticisinin kapatılmasını içerir. Bu, sayfa kümelerinin tutarlı durumda olması için sayfa kümelerindeki tüm güncelleştirmeleri zorlar.

1. Kuyruk yöneticisini kullanan tüm IBM MQ uygulamalarını durdurun (önce tamamlamalarını sağlayın). Bu, örneğin, erişim güvenliği ya da kuyruk ayarları değiştirilerek yapılabilir.
2. Tüm etkinlik tamamlandığında, belirsiz kurtarma birimlerini görüntüleyin ve çözünüz. ([DISPLAY CONN](#) ve [RESOLVE INDOUBT](#) içinde açıklandığı gibi, [DISPLAY CONN](#) ve [RESOLVE INDOUBT](#) komutlarını kullanın.)

Bu, sayfa kümelerini tutarlı bir duruma getirir; bunu yapmazsanız, sayfa kümeleriniz tutarlı olmayabilir ve etkili bir şekilde yaklaşık bir yedekleme işlemi yapıyorsunuz.

3. En son günlük verilerinin günlük veri kümelerine yazıldığından emin olmak için [ARCHIVE LOG](#) komutunu verin.
4. [STOP QMGR MODE \(QUIESCE\)](#) komutunu verin. [CSQI024I](#) ya da [CSQI025I](#) iletilerine en düşük RBA değerini kaydedin (ek bilgi için bkz. [CSQI024I](#) ve [CSQI025I](#)). Günlük veri kümelerini, RBA değeriyle gösterilen günlük veri kümesinden başlayarak geçerli günlük veri kümesine kadar tutmanız gerekir.
5. Tüm kuyruk yöneticisi sayfa kümelerinin yedek kopyalarını alın (bkz. [“Sayfa kümelerini yedekleme” sayfa 525](#)).

Yöntem 2: Bulanık yedekleme

Bu yöntem, kuyruk yöneticisinin sona erdirilmesini içermez. Bu nedenle, güncellemeler yedekleme işlemi sırasında sanal saklama alanı arabelleklerinde olabilir. Bu, sayfa kümelerinin tutarlı durumda olmadığı ve yalnızca günlüklerle kurtarma için kullanılabileceği anlamına gelir.

1. [DISPLAY USAGE TYPE \(ALL\)](#) komutunu verin ve RBA değerini [CSQI024I](#) ya da [CSQI025I](#) iletilerine kaydedin (ek bilgi için bkz. [CSQI024I](#) ve [CSQI025I](#)).
2. Sayfa kümelerinin yedek kopyalarını alın (bkz. [“Sayfa kümelerini yedekleme” sayfa 525](#)).
3. En son günlük verilerinin günlük veri kümelerine yazıldığından emin olmak için [ARCHIVE LOG](#) komutunu verin. Kurtarma noktasından yeniden başlatmak için, RBA değerinin gösterdiği günlük verileri kümesinden başlayarak günlük verileri kümelerini geçerli günlük verileri kümesine kadar tutmanız gerekir.

Sayfa kümelerini yedekleme

Bir sayfa kümesini kurtarmak için, IBM MQ günlüğün ne kadar geriye gideceğini bilmeli. IBM MQ , her sayfa kümesinin ilk sayfasında *kurtarma günlüğü sıra numarası* (LSN) adı verilen bir günlük RBA numarası tutar. Bu sayı, IBM MQ 'un sayfa kümesini kurtarabileceği günlükteki başlangıç RBA' sıdır. Bir sayfa kümesini yedeklerken bu sayı da kopyalanır.

Kopya daha sonra sayfa kümesini kurtarmak için kullanılırsa, IBM MQ 'in bu RBA değerinden geçerli RBA' ya tüm günlük kayıtlarına erişimi olmalıdır. Bu, saklamak istediğiniz sayfa kümesinin en eski yedek kopyasından IBM MQ ' i kurtarmak için yeterli günlük kaydını tutmanız gerektiği anlamına gelir.

Sayfa kümelerini kopyalamak için [ADRDSSU COPY](#) işlevini kullanın.

Daha fazla bilgi için [COPY DATASET Command Sözdizimi for Logical Data Set](#) belgelerine bakın.

Örneğin:

```
//STEP2 EXEC PGM=ADRDSSU,REGION=6M
```

```
//SYSPRINT DD SYSOUT=H
//SYSIN DD *
COPY -
DATASET(INCLUDE(SCENDATA.MQPA.PAGESET.*)) -
RENAMEU(SCENDATA.MQPA.PAGESET.** , SCENDATA.MQPA.BACKUP1.** ) -
SPHERE -
REPUNC -
FASTREPLICATION(PREF )-
CANCELERROR -
TOL(ENQF)
/*
//
```

Kuyruk yöneticisi çalışırken sayfa kümesini kopyalarsanız, önce sayfa kümesinin sıfır sayfasını kopyalayan bir kopyalama yardımcı programı kullanmanız gerekir. Bunu yapmazsanız, sayfa kümenizdeki verileri bozabilirsiniz.

Bir sayfa kümesini dinamik olarak genişletme işlemi kesintiye uğrarsa (örneğin, kaybedilmekte olan sisteme güç vermek gibi), bir sayfa kümesinin yedeğini almak için ADRDSSU 'yu kullanmaya devam edebilirsiniz.

Bir Erişim Yöntemi Hizmetleri IDCAMS LISTCAT ENT('page set data set name') ALLOCgerçekleştirirseniz, HI-ALLOC-RBA ögesinin HI-USED-RBAdeğerinden yüksek olduğunu görürsünüz.

Bu sayfa kümesinin bir sonraki dolmasında, mümkünse, yeniden uzatılır ve yüksek kullanılan RBA ile en yüksek ayrılan RBA arasındaki sayfalar, başka bir yeni kapsamlı birlikte kullanılır.

Nesne tanımlamalarınızın yedeklenmesi

Nesne tanımlamalarınızın kopyalarını da yedeklemelisiniz. Bunu yapmak için, CSQUTIL COMMAND işlevinin MAKEDEF özelliğini kullanın ([IBM MQ \(COMMAND\)](#) komutlarının verilmesi konusunda açıklanmıştır).

Kuyruk yöneticinizin yedek kopyasını her aldığınızda nesne tanımlamalarınızı yedekleyin ve en güncel sürümü tutun.

Sayfa kümeleri kurtarılıyor

Kuyruk yöneticisi bir hata nedeniyle sona erdirildiyse, olağan durumda yeniden başlatma sırasında tüm kurtarma işlemleri gerçekleştirilirken kuyruk yöneticisi yeniden başlatılabilir. Ancak, sayfa kümeleriniz ya da günlük veri kümeleriniz kullanılmıyorsa, bu tür bir kurtarma mümkün değildir. Artık kurtarabileceğiniz kapsam, sayfa kümelerinin ve günlük veri kümelerinin yedek kopyalarının kullanılabilirliğine bağlıdır.

Kurtarma noktasından yeniden başlatmak için aşağıdaki özelliklere sahip olmanız gerekir:

- Kurtarılacak sayfa kümesinin yedek kopyası.
- “Yöntem 2: Bulanık yedekleme” sayfa 525’inde açıklanan "bulanık" yedekleme işlemi, kaydedilen RBA değerini içeren günlük veri kümesini, ARCHIVE LOG komutuyla yapılan günlük veri kümesini ve bunlar arasındaki tüm günlük veri kümelerini kullandıysanız.
- Tam yedekleme kullandıysanız, ancak ARCHIVE LOG komutuyla yapılan günlük veri kümelerine sahip değilseniz, CSQUTIL yardımcı programının FORMAT TYPE (REPLACE) işlevini tüm sayfa kümeleriniz için çalıştırmanız **gerekmez** .

Bir sayfayı yürürlükteki durumuna ayarlamak için, ARCHIVE LOG komutundan bu yana tüm günlük veri kümelerine ve kayıtlarına sahip olmanız gerekir.

Bir sayfa kümesini kurtarmak için iki yöntem vardır. İki yöntemden birini kullanmak için kuyruk yöneticisinin durdurulması gerekir.

Basit kurtarma

Bu daha basit bir yöntemdir ve çoğu kurtarma durumu için uygundur.

1. Yedekten geri yüklemek istediğiniz sayfa kümesini silin.

2. Sayfa kümenizi yedek kopyadan kurtarmak için ADRDSSU COPY işlevini kullanın.

Diğer bir seçenek olarak, yedek kopyanızı özgün adla yeniden adlandırabilir ya da kuyruk yöneticisi yordamınızdaki CSQP00xx DD deyimini yedek sayfa kümenizi gösterecek şekilde değiştirebilirsiniz. Ancak, sayfa kümesini kaybeder ya da bozursanız, artık geri yüklenecek bir yedek kopyanızı alamazsınız.

3. Kuyruk yöneticisini yeniden başlatın.

4. Kuyruk yöneticisi başarıyla yeniden başlatıldığında, uygulamalarınızı yeniden başlatabilirsiniz.

5. Geri yüklenen sayfa için olağan yedekleme yordamlarınızı yeniden uygulayın.

Gelişmiş kurtarma

Bu yöntem, kurtarılması gereken büyük bir sayfa kümeniz varsa ya da son yedek kopyasının alınmasından bu yana sayfa kümesinde çok fazla etkinlik varsa, başarımlarını sağlar. Ancak, basit yöntemden daha fazla el ile müdahale gerektirir; bu, hata riskini ve kurtarma işlemini gerçekleştirmek için geçen süreyi artırabilir.

1. Yedekten geri yüklemek istediğiniz sayfa kümesini silin ve yeniden tanımlayın.

2. Sayfa kümesinin yedek kopyasını yeni sayfa kümesine kopyalamak için ADRDSSU ' yu kullanın. Yeni sayfa kümenizi, dinamik olarak genişletilebilmesi için ikincil bir kapsam değeriyle tanımlayın.

Diğer bir seçenek olarak, yedek kopyanızı özgün adla yeniden adlandırabilir ya da kuyruk yöneticisi yordamınızdaki CSQP00xx DD deyimini yedek sayfa kümenizi gösterecek şekilde değiştirebilirsiniz. Ancak, sayfa kümesini kaybeder ya da bozursanız, artık geri yüklenecek bir yedek kopyanızı alamazsınız.

3. Kuyruk yöneticinize ilişkin CSQINP1 tanımlamalarını değiştirerek, kurtarılan sayfa kümesiyle ilişkilendirilmiş arabellek havuzunun olabildiğince büyük olmasını sağlayabilirsiniz. Arabellek havuzunu büyütmek için, değiştirilen sayfaların tümünü arabellek havuzunda bulundurabilir ve G/Ç miktarını sayfa kümesine azaltabilirsiniz.

4. Kuyruk yöneticisini yeniden başlatın.

5. Kuyruk yöneticisi başarıyla yeniden başlatıldığında, durdurun (susturarak) ve o sayfa kümesine ilişkin olağan arabellek havuzu tanımlamasını kullanarak yeniden başlatın. Bu ikinci yeniden başlatma işlemi başarıyla tamamlandıktan sonra, uygulamalarınızı yeniden başlatabilirsiniz.

6. Geri yüklenen sayfa için olağan yedekleme yordamlarınızı yeniden uygulayın.

Kuyruk yöneticisi yeniden başlatıldığında ne olur

Kuyruk yöneticisi yeniden başlatıldığında, sayfa kümesine ilişkin yeniden başlatma noktasından başlayarak, günlükte kayıtlı olan sayfa kümesinde yapılan tüm değişiklikleri uygular. IBM MQ , bu şekilde birden çok sayfa kümesini kurtarabilir. Sayfa kümesi, gerekiyorsa, ortam kurtarma sırasında dinamik olarak genişletilir.

Yeniden başlatma sırasında IBM MQ , aşağıdan en düşük değeri olarak başlatılacağı günlük RBA ' yı belirler:

- Her sayfa kümesine ilişkin denetim noktası günlüğü kaydından kurtarma LSN 'si.
- Her sayfa kümesindeki sıfır numaralı sayfadan kurtarma LSN 'si.
- Yedekleme işlemi gerçekleştirildiği sırada sistemdeki en eski tamamlanmamış kurtarma biriminin RBA ' sı.

Tüm nesne tanımlamaları sayfa kümesi sıfır değerinde saklanır. İletiler herhangi bir sayfa kümesinde saklanabilir.

Not: Sayfa kümesi sıfır yoksa, kuyruk yöneticisi yeniden başlatılamaz.

Sayfa kümelerinin silinmesi

Bir sayfa kümesini DELETE PSID komutunu kullanarak silebilirsiniz; bu komutun ayrıntıları için [DELETE PSID](#) başlıklı konuya bakın.

Hala herhangi bir depolama sınıfı tarafından başvuru alan bir sayfa kümesini silemezsiniz. Bir sayfa kümesine gönderme yapan depolama sınıflarını bulmak için DISPLAY STGCLASS 'ı kullanın.

Veri kümesi IBM MQ içinden serbest bırakıldı, ancak silinmedi. İleride kullanılmak üzere kullanılabilir olmaya devam eder ya da z/OS olanakları kullanılarak silinebilir.

Sayfa kümesini, kuyruk yöneticiniz için başlatılan görev yordamından kaldırın.

Sayfa kümesinin tanımlamasını CSQINP1 kullanıma hazırlama veri kümenizden kaldırın.

CSQUTIL kullanılarak kuyrukların yedeklenmesi ve geri yüklenmesi

CSQUTIL kullanarak yedekleme ve geri yükleme hakkında daha fazla bilgi için bu konuyu başvuru olarak kullanın.

Kuyrukları yedeklemek ve geri yüklemek için CSQUTIL yardımcı programı işlevlerini kullanabilirsiniz. Bir kuyruğu yedeklemek için, kuyruktaki iletileri bir veri kümesine kopyalamak üzere COPY ya da SKOPI işlevini kullanın. Kuyruğu geri yüklemek için tamamlayıcı işlevi LOAD ya da SLOAD kullanın. Daha fazla bilgi için bkz. [IBM MQ yardımcı programı](#).

Arabellek havuzlarının yönetilmesi

Arabellek havuzlarınızı değiştirmek ya da silmek istiyorsanız bu konuyu kullanın.

Bu konuda, arabellek havuzlarının nasıl değiştirileceği ve silineceği açıklanmaktadır. Aşağıdaki kısımları içerir:

- [“Arabellek havuzundaki arabellek sayısının değiştirilmesi” sayfa 528](#)
- [“Arabellek havuzunun silinmesi” sayfa 529](#)

Arabellek havuzları, kuyruk yöneticisi kullanıma hazırlanırken, CSQINP1 kullanıma hazırlama giriş veri kümesinden verilen DEFINE BUFFPOOL komutları kullanılarak tanımlanır. Kuyruk yöneticisi çalışırken, bu konuda ayrıntılı olarak açıklanan süreçler kullanılarak, bunların öznitelikleri iş gereksinimlerine yanıt olarak değiştirilebilir. Kuyruk yöneticisi, denetim noktası günlüğü kayıtlarına yürürlükteki arabellek havuzu özniteliklerini kaydeder. Bunlar, CSQINP1 içindeki arabellek havuzu tanımlaması REPLACE öznitelikli içeriyorsa, sonraki kuyruk yöneticisi yeniden başlatıldığında otomatik olarak geri yüklenir.

Yürürlükteki arabellek özniteliklerini görüntülemek için DISPLAY USAGE komutunu kullanın.

DEFINE PSID komutunu DSN seçeneğiyle kullanarak, arabellek havuzlarını devingen olarak da tanımlayabilirsiniz.

Arabellek havuzlarını devingen olarak değiştirirseniz, bunların tanımlamalarını CSQINP1 kullanıma hazırlama veri kümesinde de güncellemeniz gerekir.

Sayfa kümelerine, depolama sınıflarına, arabelleklere ve arabellek havuzlarına ilişkin açıklamalar ve geçerli başarımlar konularından bazıları için bkz. [z/OS üzerinde planlama](#).

Not: Arabellek havuzları önemli bir saklama alanı kullanır. Bir arabellek havuzunun büyüklüğünü artırdığınızda ya da yeni bir arabellek havuzu tanımladığınızda, yeterli saklama alanı olduğunu doğrulayın. Daha fazla bilgi için bkz. [Adres alanı depolama alanı](#).

Arabellek havuzundaki arabellek sayısının değiştirilmesi

Bir arabellek havuzu çok küçükse, koşul konsolda CSQP020E iletilisiyle sonuçlanabilir; aşağıdaki gibi ALTER BUFFPOOL komutunu kullanarak bu havuza daha fazla arabellek ayırabilirsiniz:

1. Günlükteki CSQY220I iletilerine bakarak yeni arabellekler için ne kadar yer olduğunu belirleyin. Kullanılabilir alan MB cinsinden raporlanır. Arabelleğin büyüklüğü 4 KB olduğu için, her MB kullanılabilir alan 256 arabellek ayırmanızı sağlar. Diğer görevler için bazı görevler gerektiğinden, arabelleklere tüm boş alanı ayırmayın.

Arabellek havuzu sabit 4 KB sayfa kullanıyorsa, yani PAGECLAS özneliği FIXED4KBise, LPAR ' da yeterli gerçek saklama alanı bulunduğundan emin olun.

2. Bildirilen boş alan yetersizse, komutu kullanarak başka bir arabellek havuzundaki bazı arabellekleri serbest bırakın

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

Burada *buf-pool-id* , yer kazanmak istediğiniz arabellek havuzudur ve *integer* , bu arabellek havuzuna ayrılacak yeni arabellek sayısıdır; bu arabellek havuzu için ayrılan özgün arabellek sayısından daha küçük olmalıdır.

3. Komutu kullanarak genişletmek istediğiniz arabellek havuzuna arabellek ekleyin

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

Burada *buf-pool-id* , genişletilecek arabellek havuzudur ve *integer* , bu arabellek havuzuna ayrılacak yeni arabellek sayısıdır; bu arabellek havuzu için ayrılan özgün arabellek sayısından büyük olmalıdır.

Arabellek havuzunun silinmesi

Bir arabellek havuzu artık herhangi bir sayfa kümesi tarafından kullanılmadığı zaman, bu havuza ayrılan sanal saklama alanını serbest bırakmak için havuzu silin.

Bir arabellek havuzunu DELETE BUFFPOOL komutunu kullanarak silebilirsiniz. Herhangi bir sayfa kümesi bu arabellek havuzunu kullanıyorsa komut başarısız olur.

Sayfa kümelerinin nasıl silineceğine ilişkin bilgi için bkz. [“Sayfa kümelerinin silinmesi” sayfa 527](#) .

z/OS üzerinde kuyruk paylaşım gruplarının ve paylaşılan kuyrukların yönetilmesi

IBM MQ farklı tiplerde paylaşılan kaynak kullanabilir; örneğin, kuyruk paylaşım grupları, paylaşılan kuyruklar ve bağlaşım olanağı. Bu paylaşılan kaynakları yönetmek için gereken yordamları gözden geçirmek için bu konuyu kullanın.

Bu bölüm, aşağıdaki konularla ilgili bilgi içerir:

- [“Kuyruk paylaşım gruplarını yönetme” sayfa 529](#)
- [“Paylaşılan kuyrukların yönetilmesi” sayfa 532](#)
- [“Grup nesnelere yönetme” sayfa 537](#)
- [“Bağlaşım olanağının yönetilmesi” sayfa 537](#)

Kuyruk paylaşım gruplarını yönetme

Bir kuyruk paylaşım grubuna (QSG) kuyruk yöneticisi ekleyebilir ya da kaldırabilir ve ilişkili Db2 çizelgelerini yönetebilirsiniz.

Bu konuda aşağıdaki görevlerle ilgili kısımlar bulunur:

- [“Kuyruk Paylaşım Grubunun Ayarlanması” sayfa 530](#)
- [“Kuyruk paylaşım grubuna kuyruk yöneticisi eklenmesi” sayfa 530](#)
- [“Kuyruk yöneticisini kuyruk paylaşım grubundan kaldırma” sayfa 531](#)
- [“Db2 çizelgelerinden bir kuyruk paylaşım grubunun kaldırılması” sayfa 532](#)
- [“Db2 tanımlamalarının tutarlılığının doğrulanması” sayfa 532](#)

Kuyruk Paylaşım Grubunun Ayarlanması

Her kuyruk paylaşım grubunun adı en çok dört karakterdir. Ad ağızınızda benzersiz olmalı ve herhangi bir kuyruk yöneticisi adından farklı olmalıdır.

Bir kuyruk paylaşım grubu oluşturmak için aşağıdaki adımları izleyin:

1. Bu, Db2 veri paylaşım grubunu kullanan ilk kuyruk paylaşım grubuysa, Db2 ortamı ayarlayın.
2. Bağlaşım olanağını ayarlayın.
3. Kuyruk paylaşım grubunu Db2 tablolarına ekleyin. Kuyruk paylaşım grubu yardımcı programının ADD QSG işlevini kullanın (CSQ5PQSG). Bu program Kuyruk paylaşım grubu yardımcı programı kısmında açıklanmıştır. thlqual.SCSQPROC(CSQ45AQS) içinde bir örnek sağlanır.
4. “Kuyruk paylaşım grubuna kuyruk yöneticisi eklenmesi” sayfa 530 içindeki adımları izleyerek kuyruk paylaşım grubuna bir kuyruk yöneticisi ekleme
5. “Bağlaşım olanağı yapısı eklenmesi” sayfa 538 içindeki adımları izleyerek uygulama yapılarını IBM MQ olarak tanımlayın.
6. Gerekirse, paylaşılmayan kuyrukları paylaşılan kuyruklara geçirin.
7. Kullanılabilirlik için, kuyruk paylaşım grubu içinde ve dışında paylaşılan kanallar oluşturun.
 - Kuyruk paylaşım grubuna yönelik bağlantılar için:
 - İş yükünü QSG ' deki kullanılabilir kuyruk yöneticileri arasında dağıtmak için bir VIPA yuvası ya da donanım yönlendiricisi ayarlayın.
 - Kanal tanımının QSG ' deki tüm kuyruk yöneticilerinin kullanımına sunulduğundan emin olmak için QSGDISP (GROUP) ile bir alıcı kanal tanımlayın.
 - QSG ' ye MCA kanal bağlantıları için her kuyruk yöneticisinde INDISP (GROUP) ile bir dinleyici başlatın. QSG ' ye istemci bağlantıları INDISP (QMGR) ile başlayan bir dinleyiciye bağlanmaya devam etmelidir.
 - Uygulamaları, belirli bir kuyruk yöneticisi adı yerine QSG adını kullanarak bağlanacak şekilde değiştirin.
 - Uygulamaların QSG 'deki herhangi bir kuyruk yöneticisine bağlanmasına izin vermek için QSG' deki tüm kuyruk yöneticilerindeki kanal kimlik doğrulama kurallarının aynı olduğundan emin olun.
 - Kuyruk paylaşım grubu dışı bağlantılar için:
 - Paylaşılan bir iletim kuyruğu tanımlayın.
 - Giden kanalı QSGDISP (GROUP) ve DEFCDISP (SHARED) ile tanımlayın.

Var olan bir kanalı paylaşılan bir kanala dönüştürürseniz, kanal tarafından kullanılan eşitleme kuyruğu değiştiği için kanalı başlatmadan önce RESET CHANNEL komutunu vermeniz gerekebilir.

Kuyruk paylaşım grubuna kuyruk yöneticisi eklenmesi

Kuyruk yöneticisi, var olan bir kuyruk paylaşım grubuna eklenebilir.

Aşağıdakilere dikkat edin:

- Kuyruk yöneticilerini ekleyebilmeniz için kuyruk paylaşım grubunun var olması gerekir.
- Kuyruk yöneticisi, tek bir kuyruk paylaşım grubunun üyesi olabilir.

Bir kuyruk paylaşım grubuna kuyruk yöneticisi eklemek için aşağıdaki adımları izleyin:

1. Kuyruk yöneticisi ve kanal başlatıcısı kullanıcı kimliklerine uygun erişim vermek için kuyruk paylaşım grubu için ESM güvenlik denetimlerini uygula içindeki görevleri gerçekleştirin.
2. Kuyruk paylaşım grubunda, verileri SMDS ' ye boşaltmak üzere yapılandırılmış CF yapıları varsa, SMDS ortamını ayarla içindeki görevleri gerçekleştirin.
3. Kuyruk yöneticisini durdurun.

4. Kuyruk paylaşım grubu yardımcı programının ADD QMGR işlevini kullanın (CSQ5PQSG). Bu program kuyruk paylaşım grubu yardımcı programında açıklanmaktadır. thlqual.SCSQPROC(CSQ45AQM) içinde bir örnek verilmiştir.
5. Kuyruk paylaşım grubu verileri eklemek için sistem parametre modülünüzü değiştirin :
 - a. CSQ6SYSP ' yi değiştirerek QSGDATA parametresini belirleyin. Ek bilgi için bkz. using CSQ6SYSP .
 - b. Sistem parametre modülünü derleyip bağlayın. Yükleme modülü için farklı bir ad kullanmak isteyebilirsiniz.
 - c. Başlatma işlecinizi yeni modülü kullanacak şekilde değiştirin.
6. Gerekli CF yapılarını ve SYSTEM kuyruklarını tanımlayan thlqual.SCSQPROC(CSQ4INSS) adlı örnek üyeyi kopyalayın ve uyarlayın. Uyarlanmış üyeyi, kuyruk yöneticisi başlatma JCL 'sinde CSQINP2 DD ' ye ekleyin.
7. Kuyruk paylaşım grubu sistem parametresi modülünü kullanarak kuyruk yöneticinizi yeniden başlatın.
8. İsteğe bağlı olarak, kuyruk yöneticisi adı yerine, kuyruk paylaşım grubu adının önek olarak eklendiği güvenlik tanımlarına geçin.
9. QSG 'ye bağlantılar için paylaşılan kanallar kullanılıyorsa, uygulamaların QSG' deki herhangi bir kuyruk yöneticisine bağlanmasına izin vermek için QSG ' deki diğer kuyruk yöneticilerine ikizleme yapan kanal doğrulama kuralları oluşturun.
10. İsteğe bağlı olarak, QSG 'deki kuyruk yöneticisine bağlı uygulamaların QSG' deki diğer kuyruk yöneticilerinin barındırdığı kuyruklara ileti koymasına izin vermek için aşağıdakilerden birini yapın:
 - ALTER QMGR IGQ (ENABLED) komutunu vererek grup içi kuyruğa alma komutunu açın.
 - İletim kuyruklarını ve kanallarını QSG ' deki diğer kuyruk yöneticilerine tanımlayın. Hedef kuyruk yöneticileriyle aynı adı taşıyan iletim kuyruklarının tanımlanması, uzak kuyruklar ve kuyruk yöneticisi diğer adlarını tanımlama gereksinimini ortadan kaldırır.

Not: Daha önceki IBM MQ sürümlerini çalıştıran kuyruk yöneticilerini içeren var olan bir kuyruk paylaşım grubuna bir kuyruk yöneticisi eklemek için, öncelikle gruptaki en yüksek IBM MQ sürümü için birlikte bulunma PTF 'sini gruptaki önceki her sürüm kuyruk yöneticisine uygulamanız gerekir.

Kuyruk yöneticisini kuyruk paylaşım grubundan kaldırma

Kuyruk yöneticisinin günlükleri başka bir işlem için gerekli değilse ve kuyruk yöneticisinin sahip olduğu tüm SMDS boşsa, kuyruk yöneticisini kuyruk paylaşım grubundan kaldırabilirsiniz.

Ek bilgi için Paylaşılan ileti veri kümelerinin silinmesi ve DELETE CFSTRUCT başlıklı konuya bakın.

Günlükler aşağıdakiler içeriyorsa gereklidir:

- Kuyruk paylaşım grubu tarafından kullanılan bağlaşımla (CF) uygulama yapılarından birinin en son yedeklemesi
- İleride gerçekleştirilecek bir geri yükleme işleminin gerektirdiği veriler; yani, kuyruk yöneticisi son yedekleme dışlama aralığı değeriyle açıklanan zamandan bu yana kurtarılabılır bir yapı kullanmıştır.

Bu noktalardan biri ya da her ikisi geçerse ya da kuyruk yöneticisinin sahip olduğu bir SMDS ileti içeriyorsa, kuyruk yöneticisi kaldırılamaz. İleride gerçekleştirilecek bir geri yükleme işleminde hangi kuyruk yöneticilerinin günlüklerinin gerekli olduğunu saptamak için, TYPE (BACKUP) seçeneğiyle MQSC DISPLAY CFSTATUS komutunu kullanın (bu komutun ayrıntıları için bkz. DISPLAY CFSTATUS).

Bir kuyruk yöneticisini bir kuyruk paylaşım grubundan kaldırmak için aşağıdaki adımları kullanın:

1. Kuyruk yöneticisine bağlı olan ve iletileri paylaşılan kuyruklara koyan uygulamaları durdurun.
2. Bu kuyruk yöneticisiyle ilgili belirsiz iş birimlerini çözün.
3. DISPLAY USAGE TYPE (SMDS) komutunu vererek, kuyruk yöneticisine ait herhangi bir SMDS ' de ileti olup olmadığını saptayın.

4. Herhangi bir uygulama yapısı için boşaltılmış iletiler varsa, bu iletiler kuyruktan alınincaya kadar bekleyin. DISPLAY USAGE TYPE (SMDS) tarafından bildirilen boşaltılan iletilerin sayısı, devam etmeden önce sıfır olmalıdır.
5. STOP QMGR MODE (QUIESCE) komutunu kullanarak kuyruk yöneticisini düzgün bir şekilde kapatın.
6. Sonraki adımda BACKUP CFSTRUCT komutunda belirteceğiniz EXCLINT parametresinin değerine en az eşdeğer bir aralık bekleyin.
7. Başka bir kuyruk yöneticisinde, MQSC BACKUP CFSTRUCT komutunu kullanarak ve önceki adımda gerektiği gibi bir EXCLINT değeri belirterek, kurtarılabilir her CF yapısı için bir CF yapısı yedeklemesi çalıştırın.
8. DISPLAY CFSTATUS (*) TYPE (BACKUP) komutunun çıkışını inceleyerek, kuyruk yöneticisinin günlüklerinin hiçbir CF yapısını geri yüklemek için gerekli olmadığını doğrulayın.
9. Kuyruk yöneticisini kuyruk paylaşım grubundan kaldırmak için CSQ5PQSG yardımcı programının REMOVE QMGR işlevini kullanın. Bu program Kuyruk paylaşım grubu yardımcı programı kısmında açıklanmıştır. thlqual.SCSQPROC(CSQ45RQM) içinde bir örnek sağlanmıştır.
10. Kuyruk yöneticisini yeniden başlatmadan önce, QSGDATA sistem parametresini varsayılan değerine sıfırlayın ve sistem parametre modülünü yeniden yaratın. Sistem değişikliklerinizin uyarılanmasına ilişkin bilgi için CSQ6SYSP' nin Kullanılması başlıklı konuya bakın.

Bir kuyruk paylaşım grubundaki son kuyruk yöneticisini kaldırırken, REMOVE yerine FORCE seçeneğini kullanmanız gerektiğini unutmayın. Bu işlem, kuyruk yöneticisini kuyruk paylaşım grubundan kaldırır, ancak kurtarma için gereken kuyruk yöneticisi günlüklerinin tutarlılık denetimlerini gerçekleştirmez. Bu işlemi yalnızca kuyruk paylaşım grubunu siliyorsanız gerçekleştirmeniz gerekir.

Db2 çizelgelerinden bir kuyruk paylaşım grubunun kaldırılması

Db2 çizelgelerinden bir kuyruk paylaşım grubunu kaldırmak için, kuyruk paylaşım grubu yardımcı programının (CSQ5PQSG) REMOVE QSG işlevini kullanın. Bu program Kuyruk paylaşım grubu yardımcı programı kısmında açıklanmıştır. thlqual.SCSQPROC(CSQ45RQS) içinde bir örnek verilmiştir.

Bir kuyruk paylaşım grubunu yalnızca, kuyruk paylaşım grubundan ("Kuyruk yöneticisini kuyruk paylaşım grubundan kaldırma" sayfa 531 içinde açıklandığı gibi) tüm kuyruk yöneticilerini kaldırdıktan sonra ortak Db2 veri paylaşım grubu tablolarından kaldırabilirsiniz.

Kuyruk paylaşım grubu denetim çizelgesinden kuyruk paylaşım grubu kaydı silindiğinde, bu kuyruk paylaşım grubuyla ilgili tüm nesnelere ve denetim bilgileri diğer IBM MQ Db2 çizelgelerinden silinir. Bu, paylaşılan kuyruk ve grup nesnesi bilgilerini içerir.

Db2 tanımlamalarının tutarlılığının doğrulanması

Bir kuyruk paylaşım grubu içindeki paylaşılan kuyruklara ilişkin sorunlar, Db2 nesne tanımlamalarının herhangi bir nedenle tutarsız olması durumunda ortaya çıkabilir.

Kuyruk yöneticilerine, CF yapılarına ve paylaşılan kuyruklara ilişkin Db2 nesne tanımlamalarının tutarlılığını doğrulamak için, kuyruk paylaşım grubu yardımcı programının (CSQ5PQSG) VERIFY QSG işlevini kullanın. Bu program Kuyruk paylaşım grubu yardımcı programı kısmında açıklanmıştır.

Paylaşılan kuyrukların yönetilmesi

Paylaşılan kuyrukların nasıl kurtarılacağını, taşınacağını ve geçirileceğini anlamak için bu konuyu kullanın.

Bu bölümde aşağıdaki görevler açıklanır:

- "Paylaşılan kuyrukların kurtarılması" sayfa 533
- "Paylaşılan kuyrukların taşınması" sayfa 533
- "Paylaşılmayan kuyrukların paylaşılan kuyruklara geçirilmesi" sayfa 536
- Db2 Bağlantısının Askıya Alınması

Paylaşılan kuyrukların kurtarılması

IBM MQ , varsa, paylaşılan kuyruklardaki kalıcı iletileri kurtarabilir:

- İletileri içeren CF yapılarının yedekleri gerçekleştirildi.
- Yedeklerin alındığı noktadan kurtarma gerçekleştirmek için, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine ilişkin tüm günlükler kullanılabilir.
- Db2 kullanılabilir ve yapı yedekleme çizelgesi en son CF yapısı yedeğinden daha yeni.

Paylaşılan kuyruktaki iletiler bir bağlaşım olanağı (CF) yapısında saklanır. Kalıcı iletiler paylaşılan kuyruklara konabilir ve paylaşılmayan kuyruklardaki kalıcı iletiler gibi bunlar da kuyruk yöneticisi günlüğüne kopyalanır. MQSC [BACKUP CFSTRUCT](#) ve [RECOVER CFSTRUCT](#) komutları, olası olmayan bir bağlaşım olanağı arızası durumunda bir CF yapısının kurtarılmasına olanak sağlamak için sağlanır. Bu gibi durumlarda, etkilenen yapıda saklanan kalıcı olmayan iletiler kaybolur, ancak kalıcı iletiler kurtarılabilir. Yapı kurtarıncaya kadar, yapıyı kullanan diğer uygulama etkinlikleri önlenir.

Kurtarmayı etkinleştirmek için, bağlaşım olanağı listesi yapılarınızı sık sık MQSC [BACKUP CFSTRUCT](#) komutunu kullanarak yedeklemelisiniz. CF yapısındaki iletiler, yedeklemeyi yapan kuyruk yöneticisinin etkin günlük verileri kümesine yazılır. Yedeklemenin kaydını şu adrese yazar: Db2: Yedeklenmekte olan CF yapısının adı, yedeklemeyi yapan kuyruk yöneticisinin adı, bu yedeklemeye ilişkin RBA aralığı ve yedekleme süresi. CF liste yapılarını, paylaşılan kuyrukları etkin olarak kullanmıyor olsanız da yedekleyin; örneğin, ileride kullanmak üzere bir kuyruk paylaşım grubu oluşturduysanız.

Kurtarma işlemini gerçekleştirebilen kuyruk yöneticisine bir MQSC [RECOVER CFSTRUCT](#) komutu vererek bir CF yapısını kurtarabilirsiniz; kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisini kullanabilirsiniz. Kurtarılacak tek bir CF yapısı belirtebilir ya da aynı anda birden çok CF yapısını kurtarabilirsiniz.

Daha önce de belirtildiği gibi, CF liste yapılarınızı sık sık yedeklemek önemlidir; tersi durumda, bir CF yapısının kurtarılması uzun sürebilir. Ayrıca, kurtarma işlemi iptal edilemez.

Paylaşılan bir kuyruğun tanımlaması bir Db2 veritabanında tutulur ve bu nedenle, gerekirse standart Db2 veritabanı yordamları kullanılarak kurtarılabilir. Ek bilgi için [Paylaşılan kuyruklar ve kuyruk paylaşım grupları](#) konusuna bakın.

Paylaşılan kuyrukların taşınması

Bu bölümde, paylaşılan bir kuyruğu bir bağlaşım olanağı yapısından diğerine taşıyarak yük dengeleme işleminin nasıl gerçekleştirileceği açıklanmaktadır. Ayrıca, paylaşılmayan bir kuyruğun paylaşılan bir kuyruğa nasıl taşınacağını ve paylaşılan bir kuyruğun paylaşılmayan bir kuyruğa nasıl taşınacağını da açıklar.

Bir kuyruğu taşıdığınızda, yordamın bir parçası olarak geçici bir kuyruk tanımlamanız gerekir. Bunun nedeni, her kuyruğun benzersiz bir adı olması gerektiğinden, kuyruklar farklı kuyruk yer imlerine sahip olsa bile, aynı ada sahip iki kuyruğunuz olamaz. IBM MQ , aynı ada sahip iki kuyruğu (adım [“2” sayfa 534](#)) tolere eder, ancak kuyrukları kullanamazsınız.

- Kuyruğun bir bağlaşım olanağı yapısından diğerine taşınması
- Paylaşılmayan bir kuyruğun paylaşılan bir kuyruğa taşınması
- Paylaşılan bir kuyruğun paylaşılmayan bir kuyruğa taşınması

Kuyruğun bir bağlaşım olanağı yapısından diğerine taşınması

Kuyrukları ve iletilerini bir CF yapısından diğerine taşımak için MQSC [TAŞINIR QLOCAL](#) komutunu kullanın. Yeni bir CF yapısına taşımak istediğiniz kuyruğu ya da kuyrukları tanımladığınızda, her bir kuyruğu taşımak için aşağıdaki yordamı kullanın:

1. Taşımak istediğiniz kuyruğun herhangi bir uygulama tarafından kullanılmadığını, yani IPPROCS ve OPPROCS kuyruk özniteliklerinin kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine sıfır olduğunu doğrulayın.

2. MQPUT ' leri geçersiz kılmak için kuyruk tanımlamasını değiştirerek, uygulamaların taşınmakta olan kuyruğa ileti koymasını önleyin. Kuyruk tanımlamasını PUT (DISABLED) olarak değiştirin.
3. Aşağıdaki komutu kullanarak, taşınmakta olan kuyrukla aynı özniteliklere sahip geçici bir kuyruk tanımlayın:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
```

Not: Bu geçici kuyruk önceki bir çalıştırmadan varsa, tanımlamadan önce silin.

4. Aşağıdaki komutu kullanarak iletileri geçici kuyruğa taşıyın:

```
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
```

5. Taşınmakta olduğunuz kuyruğu silmek için şu komutu kullanın:

```
DELETE QLOCAL(Queue_TO_MOVE)
```

6. Aşağıdaki komutu kullanarak CFSTRUCT özniteliğini değiştirerek, taşınmakta olan kuyruğu yeniden tanımlayın:

```
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
```

Kuyruk yeniden tanımlandığında, “3” sayfa 534. adımda yaratılan geçici kuyruğa dayalıdır.

7. Aşağıdaki komutu kullanarak iletileri yeni kuyruğa geri taşıyın:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_TO_MOVE)
```

8. “3” sayfa 534 . adımda yaratılan kuyruk artık gerekli değil. Silmek için aşağıdaki komutu kullanın:

```
DELETE QL(TEMP_QUEUE)
```

9. Taşınmakta olan kuyruk CSQINP2 veri kümelerinde tanımlandıysa, CSQINP2 veri kümelerinde uygun DEFINE QLOCAL komutunun CFSTRUCT özniteliğini değiştirin. REPLACE anahtar sözcüğünü ekleyerek, varolan kuyruk tanımlamasının yerine yenisini koyun.

Şekil 30 sayfa 535 içinde, bir kuyruğu bir CF yapısından diğerine taşımaya ilişkin örnek bir iş gösterilmektedir.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqua1.SCSQANLE,DISP=SHR
//      DD DSN=thlqua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*

```

Şekil 30. Bir kuyruğu bir CF yapısından diğerine taşımak için örnek iş

Paylaşılmayan bir kuyruğun paylaşılan bir kuyruğa taşınması

Paylaşılmayan bir kuyruğu paylaşılan bir kuyruğa taşıma yordamı, bir kuyruğu bir CF yapısından diğerine taşıma yordamına benzer (bkz. [“Kuyruğun bir bağlaşım olanağı yapısından diğerine taşınması”](#) sayfa 533). Şekil 31 sayfa 535 , bunu yapmak için örnek bir iş verir.

Not: Paylaşılan kuyruklardaki iletilerin ileti büyüklüğü, ileti kalıcılığı ve kuyruk dizini tipi üst sınırlamalarına tabi olduğunu unutmayın; bu nedenle, paylaşılmayan bazı kuyrukları paylaşılan bir kuyruğa taşıyamayabilirsiniz.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqua1.SCSQANLE,DISP=SHR
//      DD DSN=thlqua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*

```

Şekil 31. Paylaşılmayan bir kuyruğu paylaşılan bir kuyruğa taşımak için örnek iş

Paylaşılan bir kuyruğun paylaşılmayan bir kuyruğa taşınması

Paylaşılan bir kuyruğu paylaşılmayan bir kuyruğa taşıma yordamı, bir kuyruğu bir CF yapısından diğerine taşıma yordamına benzer (bkz. [“Kuyruğun bir bağlaşım olanağı yapısından diğerine taşınması”](#) sayfa 533).

Şekil 32 sayfa 536 , bunu yapmak için örnek bir iş verir.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqua1.SCSQANLE,DISP=SHR
//      DD DSN=thlqua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) STGCLASS(NEW) QSGDISP(QMGR)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*

```

Şekil 32. Paylaşılan bir kuyruğu paylaşılmayan bir kuyruğa taşımak için örnek iş

Paylaşılmayan kuyrukların paylaşılan kuyruklara geçirilmesi

Paylaşılmayan kuyrukların paylaşılan kuyruklara geçirilmesinin iki aşaması vardır:

- Kuyruk paylaşım grubundaki ilk (ya da tek) kuyruk yöneticisini geçirme
- Kuyruk paylaşım grubundaki diğer kuyruk yöneticilerini geçirme

Kuyruk paylaşım grubundaki ilk (ya da tek) kuyruk yöneticisini geçirme

Şekil 31 sayfa 535 içinde, paylaşılmayan bir kuyruğu paylaşılan bir kuyruğa taşımak için örnek bir iş gösterilmektedir. Bunu, yeni düzeye geçirilmesi gereken her kuyruk için yapın.

Not:

1. Paylaşılan kuyruklardaki iletiler, ileti büyüklüğü, ileti kalıcılığı ve kuyruk dizini tipi üst sınırlamalarına tabidir; bu nedenle, paylaşılmayan bazı kuyrukları paylaşılan bir kuyruğa taşıyamayabilirsiniz.
2. Paylaşılan kuyruklar için doğru dizin tipini kullanmalısınız. Bir iletim kuyruğunu paylaşılan kuyruk olacak şekilde geçirirseniz, dizin tipi MSGID olmalıdır.

Kuyruk boşsa ya da üzerindeki iletileri saklamanız gerekmezse, kuyruğun geçirilmesi daha kolaydır. Şekil 33 sayfa 536 içinde, bu koşullarda kullanılacak örnek bir iş gösterilmektedir.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqua1.SCSQANLE,DISP=SHR
//      DD DSN=thlqua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
DELETE QL(TEMP_QUEUE)
/*

```

Şekil 33. İleti olmadan paylaşılmayan bir kuyruğu paylaşılan kuyruğa taşımak için örnek iş

Kuyruk paylaşım grubundaki diğer kuyruk yöneticilerini geçirme

1. Varolan bir paylaşılan kuyrukla aynı adı taşımayan her kuyruk için, kuyruğu Şekil 31 sayfa 535 ya da Şekil 33 sayfa 536 içinde açıklandığı gibi taşıyın.
2. Varolan bir paylaşılan kuyrukla aynı adı taşıyan kuyruklar için, iletileri Şekil 34 sayfa 537 içinde gösterilen komutları kullanarak paylaşılan kuyruğa taşıyın.

```
MOVE QLOCAL(Queue_To_Move) QSGDISP(QMGR) TOQLOCAL(Queue_To_Move)
DELETE QLOCAL(Queue_To_Move) QSGDISP(QMGR)
```

Şekil 34. İletilerin paylaşılmayan bir kuyruktan var olan bir paylaşılan kuyruğa taşınması

Db2 bağlantısı askıya alınıyor

Kuyruk yöneticinizi durdurmadan Db2 çizelgelerine ya da paketine bakım ya da hizmet uygulamak istiyorsanız, veri paylaşım grubundaki (DSG) kuyruk yöneticilerinin Db2 bağlantısını geçici olarak kesmeniz gerekir.

Bunu yapmak için:

1. MQSC komutunu kullanın: SUSPEND QMGR FACILITY (Db2).
2. Bağları yap.
3. RESUME QMGR FACILITY (Db2) MQSC komutunu kullanarak Db2 ile yeniden bağlantı kurun

Bu komutların kullanımıyla ilgili sınırlamalar olduğunu unutmayın.



Uyarı: Db2 bağlantısı askıya alındığında, aşağıdaki işlemler kullanılamaz. Bu nedenle, işletmenizin en az meşgul olduğu bir zamanda bu işi yapmanız gerekir.

- Denetim için Paylaşılan kuyruk nesnelere erişim (tanımlama, silme, değiştirme)
- Paylaşılan kanallar başlatılıyor
- İletilerin Db2 içinde saklanması
- CFSTRUCT ' yi yedekleme ya da kurtarma

z/OS Grup nesneleri yönetme

Grup nesneleriyle nasıl çalışacağınızı anlamak için bu konuyu kullanın.

IBM MQ , bir grup nesnesinin tanımlamasını, onu kullanan her kuyruk yöneticisinin sayfa kümesi sıfırına otomatik olarak kopyalar. Tanımlamanın kopyasını geçici olarak değiştirebilir ve IBM MQ , havuz kopyasından sayfa kümesi kopyalarını yenilemenizi sağlar. IBM MQ , başlangıçta havuz kopyasından sayfa kümesi kopyalarını her zaman yenilemeyi dener (kanal nesnelere için, kanal başlatıcı yeniden başlatıldığında yapılır). Bu, kuyruk yöneticisi etkin değilken yapılan değişiklikler de içinde olmak üzere, sayfa kümesi kopyalarının havuzdaki sürümü yansıtmasını sağlar.

Yenileme işleminin gerçekleştirilmediği durumlar vardır; örneğin:

- Kuyruğun bir kopyası açıksa, kuyruğun kullanımını değiştirecek bir yenileme başarısız olur.
- Kuyruğun bir kopyasında iletiler varsa, o kuyruğu silecek bir yenileme başarısız olur.

Bu durumlarda, yenileme o kopya üzerinde gerçekleştirilmez, ancak diğer tüm kuyruk yöneticilerindeki kopyalar üzerinde gerçekleştirilir. Bir grup nesnesi ekledikten, değiştirdikten ya da sildikten sonra ve kuyruk yöneticisini ya da kanal başlatıcısını yeniden başlattıktan sonra kopyalama nesnelere ilgili sorunları denetleyin ve düzeltin.

z/OS Bağlaşım olanağının yönetilmesi

Bağlaşım olanağı (CF) yapılarının nasıl ekleneceğini ya da kaldırılacağını anlamak için bu konuyu kullanın.

Bu bölümde aşağıdaki görevler açıklanır:

- “Bağlaşım olanağı yapısı eklenmesi” sayfa 538
- “Bağlaşım olanağı yapısının kaldırılması” sayfa 538

Bağlaşım olanağı yapısı eklenmesi

Bir bağlaşım olanağı yapısı eklemek için aşağıdaki yordamı kullanın:

1. CFRM ilkesi veri kümesinde CF yapısını tanımlayın. [Bağlaşım olanağının ayarlanması](#) bölümünde, bağlaşım olanağı yapılarının adlandırılmasına ilişkin kurallar ve CFRM ilke veri kümesinde yapıların nasıl tanımlanacağı açıklanmaktadır.
2. İleti verilerini SMDS ' ye boşaltmak için yapıyı yapılandırmak istiyorsanız, veri kümelerini ayırın ve önceden biçimlendirin. Ayrıntılar için bkz. [Paylaşılan ileti veri kümesi oluşturma](#) .
3. [DEFINE CFSTRUCT](#) komutunu kullanarak yapıyı IBM MQ olarak tanımlayın.

Bağlaşım olanağı yapısının kaldırılması

Bir bağlaşım olanağı yapısını kaldırmak için aşağıdaki yordamı kullanın:

1. Silmek istediğiniz bağlaşım olanağı yapısını kullanarak tüm kuyrukların listesini almak için aşağıdaki komutu kullanın:

```
DISPLAY QUEUE(*) QSGDISP(SHARED) CFSTRUCT(structure-name)
```

2. Yapıyı kullanan tüm kuyrukları silin.
3. [DELETE CFSTRUCT](#) komutunu kullanarak CF yapısını IBM MQ içinden silin.
4. Yapı ileti verilerini SMDS 'ye yükleyecek şekilde yapılandırıldıysa, SMDS' yi silin.
5. CFRM ilke veri kümenizden yapı tanımını kaldırın ve IXCMIAPU yardımcı programını çalıştırın. (Bu, bağlaşım olanağını ayarlayan özelleştirme görevinin tersidir; açıklamalar için [Eşleştirme olanağının ayarlanması](#) kısmına bakın.)

Ayarlama bağlaşım tesis listesi izleme

Bağlaşım olanağı listesi izlemesini anlamak için bu konuyu kullanın

Bağlaşım olanağı (CF) liste izlemesi, IBM MQ paylaşılan kuyruklarını içeren liste yapılarının durumunu izlemek için kullanılır. Paylaşılan bir kuyruğa bir ileti eklendiğinde ve kuyruğun derinlik geçişleri sıfırdan sıfıra geçtiğinde, CF kuyruk paylaşım grubundaki tüm kuyruk yöneticilerini bilgilendirir. Kuyruk yöneticilerine bildirildiğinde, TRIGGER (FIRST) kullanan tetikleyici izleme programlarını ya da get-wait işlemleri gerçekleştiren uygulamaları bildirmek de dahil olmak üzere bir dizi işlem gerçekleştirebilir.

Varsayılan olarak CF, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerini aynı anda bilgilendirir. Bazı yapılandırmalarda bu, aşağıdakiler gibi sorunlara neden olabilir:

- İletilerin büyük bir yüzdesinin kuyruk paylaşım grubundaki belirli bir kuyruk yöneticisine, genellikle en hızlı LPAR 'da çalışan ya da CF' ye en yakın kuyruk yöneticisine gittiği çarpık iş yükü dağılımı ya da
- Çok sayıda başarısız alma, boşa harcanan CPU süresine neden olur.

z/OS V2R3 , **KEYRNOTIFYDELAY** adlı yeni bir bağlaşım olanağı kaynak yöneticisi (CFRM) özneliği sunar. Bu öznelik, paylaşılan kuyruklar (yönetim yapısı değil, uygulama yapıları) içeren ve belirli iş yükleri için iş yükü sapması ve boş MQGET çağrılarının ya da boş MQGET çağrılarının etkilerini en aza indirebilen liste yapılarıyla birlikte kullanılabilir.

KEYRNOTIFYDELAY yalnızca, CFLEVEL 22 ya da üstünde çalışan bir CF ' deki yapılarda ayarlanabilir.

Değeri, 0 ile 1.000.000 mikrosaniye arasında bir ile yedi ondalık basamak arasında olmalıdır. Sıfır dışında bir değere ayarlanırsa ve bir kuyruk geçişlerinin derinliği sıfırdan sıfır olmayan bir değere ayarlanırsa, CF kuyruk paylaşım grubundan tek bir kuyruk yöneticisi seçer ve kuyruk yöneticisini gruptaki diğer tüm kuyruk yöneticilerinden önce bildirir.

Kuyruk yöneticisi çevrimsel sıralı olarak seçilir. Seçilen kuyruk yöneticisi iletiyi **KEYRNOTIFYDELAY** tarafından açıklanan zaman aralığı içinde işlemezse, kuyruk paylaşım grubundaki diğer tüm kuyruk yöneticilerine de bildirim gönderilir.

KEYRNOTIFYDELAY ile ilgili daha fazla bilgi için: [Keyrange Monitoring Notification Delay\(Anahtar Aralığı İzleme Bildirimi Gecikmesine İlişkin Açıklamalar\)](#) başlıklı konuya bakın.

LISTNOTIFYDELAY ve **SUBNOTIFYDELAY**adlı benzer iki CFRM özneliği olduğunu unutmayın. Bunların hiçbirisi IBM MQ iş yükü üzerinde ölçülebilir bir etkiye sahip değildir.

z/OS

z/OS üzerinde kurtarma ve yeniden başlatma

IBM MQtarafından kullanılan kurtarma ve yeniden başlatma mekanizmalarını anlamak için bu konuyu kullanın.

z/OS

yeniden başlatmaIBM MQ

Bir kuyruk yöneticisi sona erdikten sonra, kuyruk yöneticisinin nasıl sonlandığına bağlı olarak farklı yeniden başlatma yordamları gerekir. Kullanabileceğiniz farklı yeniden başlatma yordamlarını anlamak için bu konuyu kullanın.

Bu konu, aşağıdaki durumlarda kuyruk yöneticinizi nasıl yeniden başlatacağınıza ilişkin bilgileri içerir:

- [“Olağan kapanmadan sonra yeniden başlatma” sayfa 539](#)
- [“Olağandışı bir sonlandırmadan sonra yeniden başlatma” sayfa 539](#)
- [“Sayfa kümelerinizi kaybetdiyseniz yeniden başlatma” sayfa 540](#)
- [“Günlük veri kümelerinizi kaybetdiyseniz yeniden başlatma” sayfa 540](#)
- [CF yapılarınızı kaybetdiyseniz yeniden başlatma](#)

Olağan kapanmadan sonra yeniden başlatma

Kuyruk yöneticisi STOP QMGR komutuyla durdurulduysa, sistem düzgün bir şekilde çalışmasını bitirir ve durmadan önce bir sonlandırma denetim noktası alır. Kuyruk yöneticisini yeniden başlattığınızda, sistem kapatma sırasında sistem durumunu belirlemek için sistem denetim noktası ve kurtarma günlüğündeki bilgileri kullanır.

Kuyruk yöneticisini yeniden başlatmak için, [“z/OS üzerinde bir kuyruk yöneticisini başlatmak ve durdurmak için MQSC 'nin kullanılması” sayfa 463](#)konusunda açıklandığı gibi START QMGR komutunu verin.

Olağandışı bir sonlandırmadan sonra yeniden başlatma

IBM MQ , yeniden başlatmanın olağan bir sona erdirmeye mi, yoksa olağandışı bir sonlandırma mı olduğunu otomatik olarak saptar.

Kuyruk yöneticisinin olağandışı bittikten sonra başlatılması, STOP QMGR komutu verildikten sonra başlatılmasından farklıdır. Kuyruk yöneticisi olağandışı bir şekilde sona ererse, işini bitirmeden ya da bir sonlandırma denetim noktası almadan sona erer.

Kuyruk yöneticisini yeniden başlatmak için, [“z/OS üzerinde bir kuyruk yöneticisini başlatmak ve durdurmak için MQSC 'nin kullanılması” sayfa 463](#)konusunda açıklandığı gibi START QMGR komutunu verin. Olağandışı bir sonlandırmadan sonra bir kuyruk yöneticisini yeniden başlattığınızda, günlükteki bilgileri kullanarak sonlandırma sırasındaki durumunu yeniler ve çeşitli görevlerin durumunu size bildirir.

Olağan durumda, yeniden başlatma işlemi tüm tutarsız durumları çözer. Ancak bazı durumlarda, tutarsızlıkları çözmek için belirli adımlar atmanız gerekir. Bu, "İş birimlerinin el ile kurtarılması" sayfa 553'te açıklanmıştır.

Sayfa kümelerinizi kaybetdiyseniz yeniden başlatma

Sayfa kümelerinizi kaybetdiyseniz, kuyruk yöneticisini yeniden başlatmadan önce bunları yedek kopyalarınızdan geri yüklemeniz gerekir. Bu, "Sayfa kümelerinin yedeklanması ve kurtarılması" sayfa 524'te açıklanmıştır.

Ortam kurtarma için gereken süre nedeniyle, kuyruk yöneticisinin bu koşullar altında yeniden başlatılması uzun sürebilir.

Günlük veri kümelerinizi kaybetdiyseniz yeniden başlatma

Bir kuyruk yöneticisini durdurduktan sonra (STOP QMGR komutu kullanılarak), günlüğün her iki kopyası da kaybolur ya da zarar görürse, tutarlı bir sayfa kümesi kümeniz varsa (Yöntem 1: Tam yedekleme kullanılarak üretilir) kuyruk yöneticisini yeniden başlatabilirsiniz.

Bu yordamı izleyin:

1. Kuyruk yöneticinizde var olan her sayfa kümesine karşılık gelecek yeni sayfa kümeleri tanımlayın. Sayfa kümesi tanımlamasıyla ilgili bilgi için Görev 15: Sayfa kümelerinizi tanımlama başlıklı konuya bakın.
Her yeni sayfa kümesinin ilgili kaynak sayfa kümesinden büyük olduğundan emin olun.
2. Hedef sayfa kümesini biçimlendirmek için CSQUTIL ' in FORMAT işlevini kullanın. Daha fazla ayrıntı için bkz. Sayfa kümelerini biçimlendirme .
3. Var olan sayfa kümelerini kopyalamak ya da yerinde ilk durumuna getirmek için CSQUTIL 'in RESETPAGE işlevini kullanın ve her sayfada günlük RBA' yı ilk durumuna getirin. Bu işleve ilişkin ek bilgi için Sayfa kümesinin kopyalanması ve günlüğün ilk durumuna getirilmesi başlıklı konuya bakın.
4. Kuyruk yöneticisi günlük veri kümelerinizi ve BSDS ' yi CSQJU003 kullanarak yeniden tanımlayın (Değişiklik günlüğü döküm yardımcı programı konusuna bakın).
5. Yeni sayfa kümelerini kullanarak kuyruk yöneticisini yeniden başlatın. Bunu yapmak için aşağıdakilerden birini yapın:
 - Kuyruk yöneticisi tarafından başlatılan görev yordamını, yeni sayfa kümelerine gönderme yapacak şekilde değiştirin. Ek bilgi için Görev 6: IBM MQ kuyruk yöneticisi için yordamlar yaratılması başlıklı konuya bakın.
 - Eski sayfa kümelerini silmek için Erişim Yöntemi Hizmetleri 'ni kullanın ve sonra yeni sayfa kümelerini yeniden adlandırarak eski sayfa kümeleriyle aynı adları verin.

Dikkat: Herhangi bir IBM MQ sayfa kümesini silmeden önce, gerekli yedek kopyaları oluşturduğunuzdan emin olun.

Kuyruk yöneticisi bir kuyruk paylaşım grubunun üyesiye, GROUP ve SHARED nesne tanımlamaları olağan durumda kayıp ya da zarar görmüş günlüklerden etkilenmez. Ancak, kayıp ya da hasarlı günlüklerin kapsadığı bir iş biriminde paylaşılan kuyruk iletileri varsa, bu tür kesinleştirilmemiş iletiler üzerindeki etki öngörülemez.

Not: Günlükler zarar görürse ve kuyruk yöneticisi bir kuyruk paylaşım grubunun üyesiye, paylaşılan kalıcı iletileri kurtarma yeteneği kaybolabilir. RECOVER (YES) özneliğine sahip tüm CF yapıları için, kuyruk paylaşım grubundaki başka bir etkin kuyruk yöneticisinde BACKUP CFSTRUCT komutunu hemen verin.

CF yapılarınızı kaybetdiyseniz yeniden başlatma

Kuyruk yöneticisi sonlanmadığı için CF yapılarınızı kaybederseniz yeniden başlatmanız gerekmez.

z/OS z/OS üzerinde alternatif site kurtarma

Tek bir kuyruk yöneticisini ya da bir kuyruk paylaşım grubunu kurtarabilir ya da disk ikizlemeyi düşünebilirsiniz.

Daha fazla ayrıntı için aşağıdaki bölümlere bakın:

- [Alternatif bir sitede tek bir kuyruk yöneticisinin kurtarılması](#)
- [Kuyruk paylaşım grubunun kurtarılması](#).
 - [CF yapısı ortam kurtarma](#)
 - [Ana sitede kuyruk paylaşım grubunun yedeklenmesi](#)
 - [Diğer sitede bir kuyruk paylaşım grubunun kurtarılması](#)
- [Disk İkizlemenin Kullanılması](#)

Alternatif bir yerde tek bir kuyruk yöneticisinin kurtarılması

IBM MQ bilgi işlem merkezinin toplam kaybı oluşursa, kurtarma yerindeki başka bir kuyruk yöneticisinde ya da kuyruk paylaşım grubunda kurtarma yapabilirsiniz. (Bir kuyruk paylaşım grubuna ilişkin diğer site kurtarma yordamı için bkz. "[Diğer sitede bir kuyruk paylaşım grubunun kurtarılması](#)" sayfa 544 .)

Kurtarma yerindeki başka bir kuyruk yöneticisinde kurtarma yapmak için, sayfa kümelerini ve günlükleri düzenli olarak yedeklemeniz gerekir. Tüm veri kurtarma işlemlerinde olduğu gibi, olağanüstü durum kurtarma hedefleri, mümkün olduğunca az veri, iş yükü işleme (güncelleme) ve zaman kaybıdır.

Kurtarma yerinde:

- Kurtarma kuyruğu yöneticileri, kayıp kuyruk yöneticileriyle aynı adlara **sahip olmalıdır** .
- Her kurtarma kuyruğu yöneticisinde kullanılan sistem parametresi modülü (örneğin, CSQZPARM), ilgili kayıp kuyruk yöneticisiyle aynı parametreleri içermelidir.

Bunu yaptığınızda, aşağıdaki yordamda açıklandığı gibi tüm kuyruk yöneticilerinizi yeniden oluşturun. Bu, tek bir kuyruk yöneticisine ilişkin kurtarma yerinde olağanüstü durumdan kurtarma gerçekleştirmek için kullanılabilir. Tüm kullanılabilir öğeler şunlardır:

- Arşiv günlüklerinin ve BSDSlerin kopyaları, birincil yerde olağan bir şekilde çalıştırılarak yaratılır (etkin günlükler, birincil yerdeki kuyruk yöneticisiyle birlikte kaybolur).
- Birincil sitedeki kuyruk yöneticisinden, kullanılabilir en son arşiv günlüğü kopyalarından aynı yaşta ya da daha eski olan sayfa kümelerinin kopyaları.

Etkin ve arşiv günlükleri için ikili günlük kaydı kullanabilirsiniz; bu durumda, BSDS güncellemelerini her iki kopyaya da uygulamanız gerekir:

1. Yeni sayfa kümesi veri kümeleri tanımlayın ve bunları birincil siteden sayfa kümelerinin kopyalarında bulunan verilerle yükleyin.
2. Yeni etkin günlük veri kümeleri tanımlayın.
3. Yeni bir BSDS veri kümesi tanımlayın ve en son arşivlenen BSDS 'yi içine kopyalamak için Erişim Yöntemi Hizmetleri REPRO' yu kullanın.
4. Bu en son BSDS ' den gelen bilgileri yazdırmak için CSQJU004 günlük eşlemi yazdırma yardımcı programını kullanın. Bu BSDS arşivlendiğinde, en son arşivlenmiş günlük etkin günlük olarak kesilmiş olur ve arşivlenmiş bir günlük olarak görüntülenmez. Bu günlüğün STARTRBA ve ENDRBA kayıtlarını kaydedin.
5. "[4](#)" sayfa 541. Adımda kaydedilen STARTRBA ve ENDRBA ' yı kullanarak, geri yüklediğiniz BSDS' de bu en son arşiv günlüğü veri kümesini kaydetmek için değişiklik günlüğü döküm yardımcı programını (CSQJU003) kullanın.
6. BSDS ' den tüm etkin günlük bilgilerini kaldırmak için CSQJU003 DELETE seçeneğini kullanın.
7. BSDS ' ye etkin günlük eklemek için CSQJU003 NEWLOG seçeneğini kullanın, STARTRBA ya da ENDRBA belirtmeyin.

8. BSDS ' ye yeniden başlatma denetim kaydı eklemek için CSQJU003 ögesini kullanın. CRESTART CREATE , ENDRBA=highrba değerini belirtin; burada highrba , kullanılabilir en son arşiv günlüğünün yüksek RBA değeri (Adım "4" sayfa 541 içinde bulunur) artı 1 'dir.

BSDS şimdi tüm etkin günlüklerin boş olduğunu, tüm arşivlenmiş günlüklerinizin kullanılabilir olduğunu ve günlüklerinizin sonundan başka denetim noktası olmadığını açıklar.

9. START QMGR komutuyla kuyruk yöneticisini yeniden başlatın. Başlatma sırasında, aşağıdaki gibi bir işletmen yanıt iletisi yayınlanır:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.  
REPLY Y TO CONTINUE, N TO CANCEL
```

Kuyruk yöneticisini başlatmak için Y yazın. Kuyruk yöneticisi başlatılır ve CRESTART deyiminde belirtilen ENDRBA ' ya kadar olan verileri kurtarır.

CSQJU003 ve CSQJU004' un kullanılmasıyla ilgili bilgi için [IBM MQ yardımcı programları- z/OS reference belgesine](#) bakın.

Aşağıdaki örnekte, 6, 7 ve 8. adımlara ilişkin CSQJU003 için örnek giriş deyimleri gösterilmektedir:

```
* Step 6  
DELETE DSNAME=MQM2.LOGCOPY1.DS01  
DELETE DSNAME=MQM2.LOGCOPY1.DS02  
DELETE DSNAME=MQM2.LOGCOPY1.DS03  
DELETE DSNAME=MQM2.LOGCOPY1.DS04  
DELETE DSNAME=MQM2.LOGCOPY2.DS01  
DELETE DSNAME=MQM2.LOGCOPY2.DS02  
DELETE DSNAME=MQM2.LOGCOPY2.DS03  
DELETE DSNAME=MQM2.LOGCOPY2.DS04  
  
* Step 7  
NEWLOG DSNAME=MQM2.LOGCOPY1.DS01,COPY1  
NEWLOG DSNAME=MQM2.LOGCOPY1.DS02,COPY1  
NEWLOG DSNAME=MQM2.LOGCOPY1.DS03,COPY1  
NEWLOG DSNAME=MQM2.LOGCOPY1.DS04,COPY1  
NEWLOG DSNAME=MQM2.LOGCOPY2.DS01,COPY2  
NEWLOG DSNAME=MQM2.LOGCOPY2.DS02,COPY2  
NEWLOG DSNAME=MQM2.LOGCOPY2.DS03,COPY2  
NEWLOG DSNAME=MQM2.LOGCOPY2.DS04,COPY2  
  
* Step 8  
CRESTART CREATE,ENDRBA=063000
```

Kurtarma yerinde kanal başlatıcıyı yeniden başlatmak için göz önünde bulundurmanız gereken şeyler, kanal başlatıcıyı farklı bir z/OS görüntüsünde yeniden başlatmak için ARM kullanılırken karşılaşılan durumlar gibidir. Ek bilgi için bkz. "IBM MQ ağında ARM ' nin kullanılması" sayfa 550 . Kurtarma stratejiniz, IBM MQ ürün kitaplıklarının ve IBM MQ (örneğin, CICS) kullanan uygulama programlama ortamlarının kurtarılmasını da kapsamaludur.

Değişiklik günlüğü döküm yardımcı programının diğer işlevleri (CSQJU003) olağanüstü durumdan kurtarma senaryolarında da kullanılabilir. HIGHRBA işlevi, önyükleme veri kümesi içindeki en yüksek RBA yazılan ve en yüksek RBA boşaltılan değerlerinin güncellenmesine olanak sağlar. CHECKPT işlevi, yeni denetim noktası kuyruğu kayıtlarının eklenmesini ya da BSDS ' de varolan denetim noktası kuyruğu kayıtlarının silinmesini sağlar.

Dikkat: Bu işlevler IBM MQ verilerinizin bütünlüğünü etkileyebilir. Bunları yalnızca IBM hizmet personelinin rehberliğinde olağanüstü durumdan kurtarma senaryolarında kullanın.

Hızlı kopyalama teknikleri

Kuyruk yöneticisi dondurulurken tüm sayfa kümelerinin ve günlüklerin kopyaları yapılırsa, kopyalar, diğer bir sitede kuyruk yöneticisini yeniden başlatmak için kullanılacak tutarlı bir küme olur. Bunlar genellikle, çok az ortam kurtarma işlemi gerçekleştirildiğinden kuyruk yöneticisinin çok daha hızlı yeniden başlatılmasını sağlar.

Kuyruk yöneticisini dondurmak için SUSPEND QMGR LOG komutunu kullanın. Bu komut, arabellek havuzlarını sayfa kümelerine temizler, bir denetim noktası alır ve diğer günlük yazma etkinliklerini durdurur. Günlük yazma etkinliği askıya alındıktan sonra, bir RESUME QMGR LOG komutu verilinceye kadar kuyruk yöneticisi etkin bir şekilde dondurulur. Kuyruk yöneticisi dondurulurken, sayfa kümeleri ve günlükler kopyalanabilir.

Sayfa kümelerini ve günlükleri hızla kopyalamak için FLASHCOPY ya da SNAPSHOT gibi araçları kullanarak kuyruk yöneticisinin dondurulduğu süre en aza indirgenebilir.

Ancak, bir kuyruk paylaşım grubu içinde, SUSPEND QMGR LOG komutu bu kadar iyi bir çözüm olmayabilir. Etkili olması için, günlüklerin kopyalarının tümünün kurtarma için aynı zaman noktasını içermesi gerekir; bu, SUSPEND QMGR LOG komutunun aynı anda kuyruk paylaşım grubu içindeki tüm kuyruk yöneticilerine verilmesi ve bu nedenle tüm kuyruk paylaşım grubunun bir süre dondurulması gerektiği anlamına gelir.

Kuyruk paylaşım grubunun kurtarılması

Birincil site olağanüstü durumunda, ana sitedeki yedek veri kümelerini kullanarak uzak bir yerde bir kuyruk paylaşım grubunu yeniden başlatabilirsiniz. Bir kuyruk paylaşım grubunu kurtarmak için, kurtarma işlemini kuyruk paylaşım grubundaki tüm kuyruk yöneticileri arasında koordine etmek ve öncelikle Db2 olmak üzere diğer kaynaklarla koordine etmek gerekir. Bu bölümde, bu görevler ayrıntılı olarak açıklanır.

- [CF yapısı ortam kurtarma](#)
- [Ana sitede kuyruk paylaşım grubunun yedeklenmesi](#)
- [Diğer sitede bir kuyruk paylaşım grubunun kurtarılması](#)

CF yapısı ortam kurtarma

Kalıcı iletileri paylaşılan bir kuyrukta tutmak için kullanılan bir CF yapısının ortam kurtarması, günlüğe kaydedilen güncellemelerin uygulanmasıyla iletilebilecek bir ortam yedeğine sahip olmasına dayanır. MQSC BACKUP CFSTRUCT komutunu kullanarak CF yapılarınızın yedeklerini düzenli aralıklarla alın. Paylaşılan kuyruklara (MQGET ve MQPUT) ilişkin tüm güncellemeler, güncellenmenin gerçekleştirildiği kuyruk yöneticisinin günlüğüne yazılır. Bir CF yapısının ortam kurtarmasını gerçekleştirmek için, o CF yapısını kullanan tüm kuyruk yöneticilerinin günlüklerinden o yedeğe günlüğe kaydedilen güncellemeleri uygulamanız gerekir. MQSC RECOVER CFSTRUCT komutunu kullandığınızda, IBM MQ ilgili kuyruk yöneticilerindeki günlükleri otomatik olarak birleştirir ve güncellemeleri en son yedeklemeye uygular.

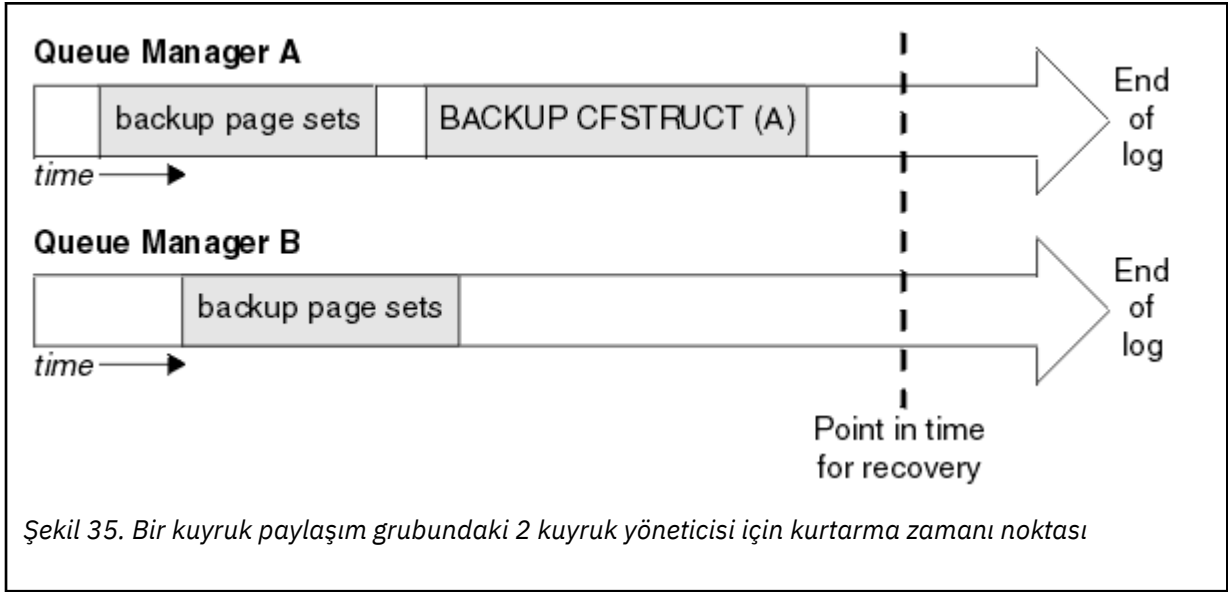
CF yapısı yedeklemesi, BACKUP CFSTRUCT komutunu işleyen kuyruk yöneticisinin günlüğüne yazılır; bu nedenle, toplanacak ve diğer yere taşınacak ek veri kümesi yoktur.

Ana sitede kuyruk paylaşım grubunun yedeklenmesi

Ana sitede, düzenli olarak tutarlı bir yedekleme kümesi oluşturmanız gerekir; bu, bir olağanüstü durum durumunda, kuyruk paylaşım grubunu alternatif bir yerde yeniden oluşturmak için kullanılabilir. Tek bir kuyruk yöneticisi için kurtarma, genellikle uzak yerde bulunan günlüklerin sonu olan isteğe bağlı bir zaman noktasına kadar olabilir. Ancak, kalıcı iletiler paylaşılan bir kuyrukta saklandığı durumlarda, kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisi kuyrukta güncelleme (MQPUT ya da MQGET) gerçekleştirmiş olabileceğinden, paylaşılan kuyrukları kurtarmak için kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinin günlükleri birleştirilmelidir.

Bir kuyruk paylaşım grubunu kurtarmak için, tüm kuyruk yöneticilerinin günlük verilerinin günlük aralığı içinde olan bir zaman noktası oluşturmanız gerekir. Ancak, günlükten yalnızca **iletilebileceğiniz** için, bu zaman noktasının BACKUP CFSTRUCT komutu verildikten sonra ve sayfa kümesi yedeklemeleri gerçekleştirildikten sonra olması gerekir. (Genellikle, kurtarma zamanındaki nokta, bir iş gününün ya da haftanın sonuna karşılık gelebilir.)

Aşağıdaki çizge, bir kuyruk paylaşım grubundaki iki kuyruk yöneticisine ilişkin zaman çizelgesini göstermektedir. Her kuyruk yöneticisi için, sayfa kümelerinin yaklaşık yedekleri alınır (bkz. [Yöntem 2: Bulanık yedekleme](#)). A kuyruk yöneticisinde, BACKUP CFSTRUCT komutu verilir. Daha sonra, etkin günlüğü kesmek ve diğer siteye taşınabilen kuyruk yöneticisinden çevrimdışı ortama kopyalamak için her kuyruk yöneticisinde bir ARCHIVE LOG komutu verilir. Günlüğün sonu, ARCHIVE LOG komutunun verildiği saati tanımlar ve bu nedenle, genellikle diğer sitede bulunan günlük verilerinin kapsamını işaretler. Kurtarma için zaman noktası, herhangi bir sayfa kümesi ya da CF yapısı yedeklerinin sonu ile alternatif sitede bulunan en eski günlük sonu arasında olmalıdır.



IBM MQ , CF yapısı yedeklemeleriyle ilişkili bilgileri Db2içindeki bir çizelgeye kaydeder. Gereksinimlerinize bağlı olarak, IBM MQ 'in kurtarılması için zamanı Db2ile koordine etmek isteyebilirsiniz ya da IBM MQ CSQ.ADMIN_B_STRBACKUP çizelgesi.

Bir kurtarma işlemine hazırlanmak için:

1. Kuyruk paylaşım grubundaki her kuyruk yöneticisi için sayfa kümesi yedekleri oluşturun.
2. RECOVER (YES) özniteliğiyle her CF yapısı için bir BACKUP CFSTRUCT komutu verin. Bu komutları tek bir kuyruk yöneticisinden ya da iş yükünü dengelemek için kuyruk paylaşım grubu içindeki farklı kuyruk yöneticilerinden verebilirsiniz.
3. Tüm yedeklemeler tamamlandıktan sonra, etkin günlüğü değiştirmek ve kuyruk paylaşım grubundaki her kuyruk yöneticisinin günlüklerinin ve BSDS'lerinin kopyalarını yaratmak için ARCHIVE LOG komutunu verin.
4. Sayfa kümesi yedeklerini, arşivlenmiş günlükleri, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinin arşivlenmiş BSDS 'lerini ve seçtiğiniz Db2 yedekleme bilgilerini, müşteri yeri dışında iletir.

Diğer sitede bir kuyruk paylaşım grubunun kurtarılması

Kuyruk paylaşım grubunu kurtarmadan önce ortamı hazırlamanız gerekir:

1. Kuyruk paylaşım grubunu kurduğunuzda, uygulama girişimleri için bağlantı olanağınızda eski bilgiler varsa, önce bunu temizlemeniz gerekir:

Not: Bağlaşım tesisinde eski bilgilere sahip değilseniz, bu adımı atlayabilirsiniz.

- a. Bu kuyruk paylaşım grubuna ilişkin CF yapılarını görüntülemek için aşağıdaki z/OS komutunu girin:

```
D XCF,STRUCTURE,STRNAME= qsgname
```


- b. Kuyruk paylaşım grubu adıyla başlayan tüm yapılar için, z/OS komutunu SETXCF FORCE CONNECTION komutunu kullanarak bağlantıyı bu yapılardan uzak tutabilirsiniz:

```
SETXCF FORCE,CONNECTION,STRNAME= strname,CONNAME=ALL
```

- c. Her yapı için aşağıdaki komutu kullanarak tüm CF yapılarını silin:

```
SETXCF FORCE,STRUCTURE,STRNAME= strname
```

2. Db2 sistemlerini ve veri paylaşım gruplarını geri yükleyin.
3. CSQ.ADMIN_B_STRBACKUP çizelgesi, ana yerde alınan en son yapı yedekleriyle ilgili bilgileri içerir.

Not: STRBACKUP çizelgesinin en son yapı yedekleme bilgilerini içermiş olması önemlidir. Eski yapı yedekleme bilgileri, son bir DISPLAY USAGE TYPE (DATASET) komutuyla verilen bilgilerin sonucu olarak atıldığınız veri kümelerini gerektirebilir; bu, kurtarılan CF yapınızın doğru bilgiler içermediği anlamına gelir.

4. Kuyruk paylaşım grubundaki her kuyruk yöneticisi için CSQ5PQSG yardımcı programının ADD QMGR komutunu çalıştırın. Bu işlem, her kuyruk yöneticisi için XCF grup girişini geri yükler.

Yardımcı programı bu senaryoda çalıştırdığınızda aşağıdaki iletiler olağandır:

```
CSQU566I Unable to get attributes for admin structure, CF not found  
or not allocated  
CSQU546E Unable to add QMGR queue_manager_name entry,  
already exists in DB2 table CSQ.ADMIN_B_QMGR  
CSQU148I CSQ5PQSG Utility completed, return code=4
```

Kuyruk paylaşım grubundaki kuyruk yöneticilerini kurtarmak için:

1. Yeni sayfa kümesi veri kümeleri tanımlayın ve bunları birincil siteden sayfa kümelerinin kopyalarında bulunan verilerle yükleyin.
2. Yeni etkin günlük veri kümeleri tanımlayın.
3. Yeni bir BSDS veri kümesi tanımlayın ve en son arşivlenen BSDS 'yi içine kopyalamak için Erişim Yöntemi Hizmetleri REPRO' yu kullanın.
4. Bu en son BSDS ' den gelen bilgileri yazdırmak için CSQJU004 günlük eşleme yazdırma yardımcı programını kullanın. Bu BSDS arşivlendiğinde, en son arşivlenmiş günlük etkin günlük olarak kesilmiş olur ve arşivlenmiş bir günlük olarak görüntülenmez. Bu günlüğün STARTRBA, STARTLRSN, ENDRBA ve ENDLRSN değerlerini kaydedin.
5. “4” sayfa 545. adımda kaydedilen değerleri kullanarak, geri yüklediğiniz en son arşiv günlüğü veri kümesini BSDS ' ye kaydetmek için değişiklik günlüğü döküm yardımcı programını (CSQJU003) kullanın.
6. BSDS ' den tüm etkin günlük bilgilerini kaldırmak için CSQJU003 DELETE seçeneğini kullanın.
7. BSDS ' ye etkin günlük eklemek için CSQJU003 NEWLOG seçeneğini kullanın, STARTRBA ya da ENDRBA belirtmeyin.
8. Kuyruk paylaşım grubu için *recoverylrsn* değerini hesaplayın. *recoverylrsn* , kuyruk paylaşım grubundaki tüm kuyruk yöneticilerindeki en düşük ENDLRSN'lerdir (Adım “4” sayfa 545 içinde kaydedildiği gibi), eksi 1. Örneğin, kuyruk paylaşım grubunda iki kuyruk yöneticisi varsa ve bunlardan birine ilişkin ENDLRSN B713 3C72 22C5 ise, diğeri B713 3D45 2123 ise, *recoverylrsn* B713 3C72 22C4' dür.
9. BSDS ' ye yeniden başlatma denetim kaydı eklemek için CSQJU003 ögesini kullanın. Belirtin:

```
CRESTART CREATE,ENDLRSN= recoverylrsn
```

Burada *recoverylrsn* , “8” sayfa 545. Adımda kaydettiğiniz değerdir.

BSDS şimdi tüm etkin günlüklerin boş olduğunu, tüm arşivlenmiş günlüklerinizin kullanılabilir olduğunu ve günlüklerinizin sonundan başka denetim noktası olmadığını açıklar.

CRESTART kaydını, kuyruk paylaşım grubu içindeki her kuyruk yöneticisi için BSDS ' ye eklemelisiniz.

10. START QMGR komutuyla kuyruk paylaşım grubundaki her kuyruk yöneticisini yeniden başlatın. Başlatma sırasında, aşağıdaki gibi bir işletmen yanıt iletisi yayınlanır:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.  
REPLY Y TO CONTINUE, N TO CANCEL
```

Kuyruk yöneticisini başlatmak için Y yanıtını verin. Kuyruk yöneticisi başlatılır ve CRESTART deyiminde belirtilen ENDRBA ' ya kadar olan verileri kurtarır.

Başlatılan ilk IBM MQ kuyruk yöneticisi, kendi kuyruk paylaşım grubunun diğer üyeleri için denetim yapısı bölümlerini yeniden oluşturabilir ve artık bu aşamada kuyruk paylaşım grubundaki her kuyruk yöneticisinin yeniden başlatılması gerekmez.

11. Tüm kuyruk yöneticilerine ilişkin denetim yapısı verileri yeniden oluşturulduğunda, her CF uygulama yapısı için bir RECOVER CFSTRUCT komutu verin.

Tek bir kuyruk yöneticisindeki tüm yapılar için RECOVER CFSTRUCT komutunu çalıştırırsanız, günlük birleştirme işlemi yalnızca bir kez gerçekleştirilir; bu nedenle, her bir kuyruk yöneticisinin günlük birleştirme adımını gerçekleştirilmesi gereken her CF yapısı için farklı bir kuyruk yöneticisinde komutu vermektten daha hızlıdır.

Bir kuyruk paylaşım grubunda koşullu yeniden başlatma işlemi kullanıldığında, IBM MQ kuyruk yöneticileri, eşdüzey denetimci yeniden oluşturma işlemi gerçekleştirirken, BSDS 'nin kendi CRESTART LRSN ile aynı CRESTART LRSN' yi içerdiğini denetleyin. Bu, yeniden oluşturulan yönetici yapısının bütünlüğünü sağlamak için tasarlanmıştır. Bu nedenle, grubun herhangi bir üyesinin bir sonraki koşulsuz yeniden başlatılmasından önce kendi CRESTART bilgilerini işleyebilmeleri için QSG ' deki diğer eşleri yeniden başlatmak önemlidir.

Disk İkizlemenin Kullanılması

Birçok kuruluş artık alternatif bir yerde veri kümelerinin zamanuyumlu kopyalarını oluşturmak için IBM Metro Mirror (önceki adıyla PPRC) gibi disk ikizleme teknolojilerini kullanır. Bu gibi durumlarda, IBM MQ sayfa kümeleri ve alternatif sitedeki günlükler ana sitedekilerle etkili bir şekilde aynı olduğundan, ayrıntılı adımların çoğu gereksiz hale gelir. Bu tür teknolojilerin kullanıldığı durumlarda, diğer bir sitede bir kuyruk paylaşım grubunu yeniden başlatma adımları şu şekilde özetlenebilir:

- Diğer sitedeki IBM MQ CF yapılarını temizleyin. (Bunlar genellikle önceki herhangi bir olağanüstü durum kurtarma egzersizinden kalan bilgileri içerir).
- Db2 sistemlerini ve IBM MQ kuyruk paylaşım grubu tarafından kullanılan veritabanındaki tüm çizelgeleri geri yükleyin.
- Kuyruk yöneticilerini yeniden başlatın. IBM WebSphere MQ 7.0.1' den önce, kuyruk yöneticisi yeniden başlatılırken her kuyruk yöneticisi denetim yapısının kendi bölümünü kurtardığı için kuyruk paylaşım grubunda tanımlanan her kuyruk yöneticisinin yeniden başlatılması gerekir. Her kuyruk yöneticisi yeniden başlatıldıktan sonra, ana LPAR ' larında olmayanlar yeniden kapatılabilir. İlk IBM MQ kuyruk yöneticisi, kendi kuyruk paylaşım grubunun diğer üyeleri için denetim yapısı bölümlerini yeniden oluşturmayı başlattı ve artık kuyruk paylaşım grubundaki her kuyruk yöneticisinin yeniden başlatılması gerekmez.
- Yönetici yapısı yeniden oluşturulduktan sonra uygulama yapılarını kurtarın.

IBM MQ 9.1.2ve daha sonra, Metro Mirror kullanılarak ikizlenen etkin günlüklere yazılırken zHyperkullanımını destekler. zHyperWrite, Metro Mirror kullanımının performans üzerindeki etkisini azaltmaya yardımcı olabilir; daha fazla bilgi için bkz. [Metro Mirror özelliğinin IBM MQ ile kullanılması](#).

Kuyruk yöneticisini yeniden başlatma

Kuyruk yöneticisi olağandışı sonlandıysa, yeniden başlatamayabilirsiniz. Bunun nedeni, sayfa kümelerinizin ya da günlüklerinizin kaybolması, kesilmesi ya da bozulması olabilir. Bu durumda, kuyruk yöneticisini yeniden kullanıma hazırlamanız gerekebilir (soğuk başlatma işlemi gerçekleştirin).

Dikkat

Yalnızca kuyruk yöneticisini başka bir şekilde yeniden başlatamazsanız soğuk başlatma işlemi gerçekleştirin. Soğuk başlatma işlemi, kuyruk yöneticinizi ve nesne tanımlamalarınızı kurtarmanızı sağlar; ileti verilerinizi **kurtaramazsınız** . Bunu yapmadan önce, bu konuda açıklanan diğer yeniden başlatma senaryolarının sizin için çalışmadığını denetleyin.

Yeniden başlattığınızda, tüm IBM MQ nesnelerinizi tanımlanır ve kullanılabilir, ancak ileti verisi yoktur.

Not: Bir kümenin parçasıyken kuyruk yöneticisini yeniden kullanıma hazırlamayın. Önce kümeden kuyruk yöneticisini kaldırmanız (kümedeki diğer kuyruk yöneticilerindeki RESET CLUSTER komutlarını kullanarak), daha sonra kümeyi yeniden kullanıma hazırlamanız ve son olarak yeni bir kuyruk yöneticisi olarak kümeye yeniden tanıtmamız gerekir.

Bunun nedeni, yeniden başlatma sırasında kuyruk yöneticisi tanıtıcısının (QMID) değiştirilmesidir; bu nedenle, eski kuyruk yöneticisi tanıtıcısına sahip küme nesneleri kümeden kaldırılmalıdır.

Daha fazla bilgi için aşağıdaki bölümlere bakın:

- [Kuyruk paylaşım grubunda olmayan bir kuyruk yöneticisinin yeniden kullanıma hazırlanması](#)
- [Kuyruk paylaşım grubundaki kuyruk yöneticilerinin yeniden başlatılması](#)

Kuyruk paylaşım grubunda olmayan bir kuyruk yöneticisini yeniden başlatma

Bir kuyruk yöneticisini yeniden kullanıma hazırlamak için aşağıdaki yordamı izleyin:

1. Kuyruk yöneticisini yeniden başlattığınızda kullanılacak nesne tanımlaması deyimlerini hazırlayın. Bunu yapmak için aşağıdakilerden birini yapın:
 - Sayfa kümesi sıfır kullanılabiliriyorsa, CSQUTIL SDEFS işlevini kullanın ([IBM MQ tanımlama komutlarının bir listesinin üretilmesi](#) konusuna bakın). Tüm nesne tiplerine (kimlik doğrulama bilgileri nesnelere, CF yapıları, kanallar, ad listesi, işlemler, kuyruklar ve depolama sınıfları) ilişkin tanımlamaları almanız gerekir.
 - Sayfa kümesi sıfır yoksa, nesne tanımlamalarınızı son yedeklediğiniz zamandaki tanımlamaları kullanın.
2. Kuyruk yöneticisi veri kümelerinizi yeniden tanımlayın ("[1](#)" sayfa 547 adımını tamamlamadan bunu yapmayın).
Daha fazla bilgi için bkz. [Önyükleme ve günlük veri kümelerini oluşturma ve sayfa kümelerinizi tanımlama](#) .
3. Yeni tanımlanan ve kullanıma hazırlanan günlük veri kümelerini, BSDS ve sayfa kümelerini kullanarak kuyruk yöneticisini yeniden başlatın. "[1](#)" sayfa 547 . adımda yarattığınız nesne tanımlaması giriş deyimlerini, CSQINP2 kullanıma hazırlama giriş veri kümesinde giriş olarak kullanın.

Kuyruk paylaşım grubundaki kuyruk yöneticilerini yeniden başlatma

Bir kuyruk paylaşım grubunda, kuyruk yöneticisinin yeniden başlatılması daha karmaşıktır. Sayfa kümesi ya da günlük sorunları nedeniyle bir ya da daha çok kuyruk yöneticisinin yeniden başlatılması gerekebilir, ancak Db2 ya da bağlaşım olanağıyla ilgili sorunlar da olabilir. Bu nedenle, bazı alternatifler vardır:

Soğuk başlangıç

Kuyruk paylaşım grubunun tamamının yeniden başlatılması, tüm bağlaşım olanağı yapılarının zorlanması, kuyruk paylaşım grubuna ilişkin tüm nesne tanımlamalarının Db2'iden temizlenmesi, günlüklerin ve BSDS ' nin silinmesi ya da yeniden tanımlanması ve kuyruk paylaşım grubundaki tüm kuyruk yöneticileri için sayfa kümelerinin biçimlendirilmesini içerir.

Paylaşılan tanımlar alıkonda

Günlükleri ve BSDS ' yi silin ya da yeniden tanımlayın, kuyruk paylaşım grubundaki tüm kuyruk yöneticileri için sayfa kümelerini biçimlendirin ve tüm bağlaşım olanağı yapılarını zorlayın. Yeniden başlatma işlemi, tüm iletiler silinecek. Kuyruk yöneticileri, Db2 veritabanında var olan GROUP

nesnelere karşılık gelen COPY nesnelere yeniden yaratır. Paylaşılan kuyruklar var olmaya devam eder ve kullanılabilir.

Tek kuyruk yöneticisi yeniden başlatıldı

Tek kuyruk yöneticisi için günlükleri ve BSDS ' yi ve biçim sayfa kümelerini silin ya da yeniden tanımlayın (bu, tüm özel nesnelere ve iletilerini siler). Yeniden başlatmada, kuyruk yöneticisi Db2 veritabanında var olan GROUP nesnelere karşılık gelen COPY nesnelere yeniden yaratır. Paylaşılan kuyruklar, üzerindeki iletiler gibi var olmaya devam eder ve kullanılabilir.

Bir kuyruk paylaşım grubunun zaman noktası kurtarma işlemi

Bu, alternatif yer olağanüstü durumdan kurtarma senaryosudur.

Paylaşılan nesnelere, Db2 kurtarma işlemi tarafından ulaşılan zaman noktasına kadar kurtarılır (A Db2 sistemi başarısız olur başlıklı konuda açıklanmıştır). Her kuyruk yöneticisi, alternatif sitede bulunan yedek kopyalardan bir zaman noktasına kadar kurtarılabilir.

Kalıcı iletiler kuyruk paylaşım gruplarında kullanılabilir ve MQSC RECOVER CFSTRUCT komutu kullanılarak kurtarılabilir. Bu komutun hata zamanını kurtardığını unutmayın. Ancak, kalıcı olmayan paylaşılan kuyruk iletilerinin kurtarılması yoktur; bunlar, CSQUTIL yardımcı programının COPY işlevini kullanarak bağımsız olarak yedek kopya oluşturmadığınız sürece kaybolur.

Farklı kuyruk yöneticilerindeki (gerçekte kurtarılmakta olan nesnelere) yerel nesnelere arasında bir bağımlılık olmadığı için, her kuyruk yöneticisinin aynı anda geri yüklenmesi gerekmez ve yeniden başlatma sırasında Db2 ile kuyruk yöneticisi yeniden eşzamanlaması, kuyruk yöneticisi temelinde bir kuyruk yöneticisi tarafından gerektiği şekilde COPY nesnelere yeniden yaratır ya da siler.

z/OS z/OS Automatic Restart Manager (ARM) olanağının kullanılması

Kuyruk yöneticilerinizi otomatik olarak yeniden başlatmak için ARM ' yi nasıl kullanabileceğinizi anlamak için bu konuyu kullanın.

Bu bölüm, aşağıdaki konularla ilgili bilgi içerir:

- [“ARM nedir?” sayfa 548](#)
- [“ARM ilkeleri” sayfa 549](#)
- [“IBM MQ ağında ARM ' nin kullanılması” sayfa 550](#)

ARM nedir?

z/OS Automatic Restart Manager (ARM), kuyruk yöneticilerinizin kullanılabilirliğini artırabilen bir z/OS kurtarma işlevidir. Bir iş ya da görev başarısız olduğunda ya da çalıştığı sistem başarısız olduğunda, ARM işi ya da görevi işletmen müdahalesi olmadan yeniden başlatabilir.

Bir kuyruk yöneticisi ya da kanal başlatıcısı başarısız olursa, ARM bunu aynı z/OS görüntüsünde yeniden başlatır. z/OSve bu nedenle bir grup ilgili altsistem ve uygulama başarısız olduysa, ARM, sistem şebekesi içindeki başka bir z/OS görüntüsünde, tüm arızalı sistemleri önceden tanımlanmış bir sırayla otomatik olarak yeniden başlatabilir. Buna *sistemler arası yeniden başlatmadendir*.

ARM tarafından kanal başlatıcısı yalnızca istisnai durumlarda yeniden başlatın. Kuyruk yöneticisi ARM tarafından yeniden başlatıldıysa, kanal başlatıcısını CSQINP2 kullanıma hazırlama veri kümesinden yeniden başlatın (bkz. [“IBM MQ ağında ARM ' nin kullanılması” sayfa 550](#)).

z/OS hatası durumunda sysplex içindeki farklı bir z/OS görüntüsünde bir kuyruk yöneticisini yeniden başlatmak için ARM ' yi kullanabilirsiniz. IBM MQ ARM ' nin farklı bir z/OS görüntüsünde yeniden başlatılmasına ilişkin ağ etkileri [“IBM MQ ağında ARM ' nin kullanılması” sayfa 550](#) içinde açıklanmıştır.

Otomatik yeniden başlatmayı etkinleştirmek için:

- Bir ARM çift veri kümesi ayarlayın.
- z/OS ' in bir *ARM ilkesinde* gerçekleştirilmesini istediğiniz otomatik yeniden başlatma işlemlerini tanımlayın.

- ARM ilkesini başlatın.

Ayrıca, IBM MQ başlatma sırasında ARM ' ye kaydolmalıdır (bu otomatik olarak gerçekleşir).

Not: Farklı z/OS görüntülerinde kuyruk yöneticilerini otomatik olarak yeniden başlatmak istiyorsanız, her kuyruk yöneticisini, ilgili kuyruk yöneticisinin yeniden başlatılabileceği her z/OS görüntüsünde bir altsistem olarak, sysplex çapında benzersiz dört karakterli altsistem adıyla tanımlamanız gerekir.

ARM çift veri kümeleri

ARM için gerekli olan çift veri kümelerini tanımladığınızdan ve ARM desteği istediğiniz herhangi bir kuyruk yöneticisini başlatmadan önce bunların çevrimiçi ve etkin olduğundan emin olun. Kuyruk yöneticisi başlatılırken çift veri kümesi kullanılmıyorsa IBM MQ otomatik ARM kaydı başarısız olur. Bu durumda IBM MQ , çift veri kümesinin olmamasının ARM desteği istemediğiniz anlamına geldiğini varsayar ve kullanıma hazırlama devam eder.

ARM çift veri kümeleriyle ilgili bilgi için [z/OS MVS Setting up a Sysplex](#) başlıklı konuya bakın.

z/OS ARM ilkeleri

Automatic Restart Manager ilkeleri, bir kuyruk yöneticisinin yeniden başlatılmasını denetleyebilen ARM işlevlerini denetleyen kullanıcı tanımlı kurallardır.

ARM işlevleri, kullanıcı tanımlı *ARM ilkesi* tarafından denetlenir. ARM tarafından yeniden başlatılacak bir kuyruk yöneticisi örneğini çalıştıran her z/OS görüntüsü, etkin bir ARM ilkesiyle bir ARM çift veri kümesine bağlanmalıdır.

IBM , varsayılan bir ARM ilkesi sağlar. z/OS ile birlikte verilen [denetim verileri yardımcı programı](#) (IXCMIAPU) kullanarak yeni ilkeler tanımlayabilir ya da ilke varsayılanlarını geçersiz kılabilirsiniz. z/OS *MVS Setting up a Sysplex* , bu yardımcı programı açıklar ve ARM ilkesinin nasıl tanımlanacağına ilişkin tüm ayrıntıları içerir.

Şekil 36 sayfa 549 içinde bir ARM ilkesi örneği gösterilmektedir. Bu örnek ilke, kuyruk yöneticisi başarısız olursa ya da tüm sistem başarısız olursa, sistem birleşimi içindeki kuyruk yöneticisini yeniden başlatır.

```
//IXCMIAPU EXEC PGM=IXCMIAPU,REGION=2M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(ARM)
DEFINE POLICY NAME(ARMPOL1) REPLACE(YES)
RESTART_GROUP(DEFAULT)
ELEMENT(*)
RESTART_ATTEMPTS(0) /* Jobs not to be restarted by ARM */
RESTART_GROUP(GROUP1)
ELEMENT(SYSMQMGRMQ*) /* These jobs to be restarted by ARM */
/*
```

Şekil 36. Örnek ARM ilkesi

Daha fazla bilgi için bakınız:

- [ARM ilkesi tanımlanması](#)
- [ARM ilkesinin etkinleştirilmesi](#)
- [ARM ' ye Kaydolma](#)

ARM ilkesi tanımlanması

ARM ilkenizi aşağıdaki gibi ayarlayın:

- Kuyruk yöneticisi yönetim ortamına bağlanan CICS ya da IMS altsistemlerini de içeren her kuyruk yöneticisi yönetim ortamı için RESTART_GROUPS tanımlayın. Bir altsistem adlandırma kuralı

kullanırsanız, '?' ve öge adlarınızda '*' genel arama karakterlerini kullanarak RESTART_GROUPS tanımlayabilir.

- Kanal başlatıcılarınız için TERMTYPE (ELEMTERM) belirtip, bunların yalnızca kanal başlatıcısı başarısız olursa ve z/OS görüntüsü başarısız olmazsa yeniden başlatılacağını belirtin.
- Kuyruk yöneticilerinizden birinin başarısız olması ya da z/OS görüntüsünün başarısız olması durumunda yeniden başlatılacağını belirtmek için TERMTYPE (ALLTERM) değerini belirtin.
- Hem kuyruk yöneticileri hem de kanal başlatıcıları için RESTART_METHOD (BOTH, PERSIST) belirleyin. Bu, ARM 'ye son başlatma sırasında sakladığı JCL' yi (sistem simgelerinin çözülmesinden sonra) kullanarak yeniden başlatmasını söyler. ARM 'a, tek tek ögenin başarısız olup olmadığına ya da z/OS görüntüsünün başarısız olup olmadığına bakılmaksızın bunu yapması gerektiğini bildirir.
- Diğer tüm ARM ilkesi seçenekleri için varsayılan değerleri kabul edin.

ARM ilkesinin etkinleştirilmesi

Otomatik yeniden başlatma yönetimi ilkenizi başlatmak için aşağıdaki z/OS komutunu verin:

```
SETXCF START,POLICY,TYPE=ARM,POLNAME= mypol
```

İlke başlatıldığında, ARM çift veri kümesine bağlı tüm sistemler aynı etkin ilkeyi kullanır.

Otomatik yeniden başlatmayı geçersiz kılmak için SETXCF STOP komutunu kullanın.

ARM ' ye kaydolma

IBM MQ , kuyruk yöneticisi başlatılırken (ARM kullanılabilirliğine bağlı olarak) otomatik olarak *ARM ögesi* olarak kaydeder. Talep edilmedikçe, kapatma aşamasında kayıttan çıkar.

Başlatma sırasında, kuyruk yöneticisi ARM ' nin kullanılabilir olup olmadığını belirler. Bu durumda, IBM MQ SYMQMGR *ssid* adını kullanarak kayıt olur; burada *ssid* dört karakterlik kuyruk yöneticisi adıdır ve SYMQMGR öge tipidir.

STOP QMGR MODE (QUIESCE) ve STOP QMGR MODE (FORCE) komutları, kuyruk yöneticisinin ARM 'den kaydını alır (başlangıçta ARM' ye kayıtlıysa). Bu, ARM ' nin bu kuyruk yöneticisini yeniden başlatmasını önler. STOP QMGR MODE (RESTART) komutu, kuyruk yöneticisinin ARM ' den kaydını kaldırıyor, bu nedenle hemen otomatik yeniden başlatma için uygun.

Her kanal başlatıcı adresi alanı ARM ' nin kullanılıp kullanılmayacağını ve kullanılacaksa SYMQCH *ssid* öge adına kaydedilip kaydedilmeyeceğini belirler; burada *ssid* , kuyruk yöneticisi adıdır ve SYMQCH öge tipidir.

Kanal başlatıcısı, olağan şekilde durduğunda ARM ' den her zaman kayıttan çıkar ve yalnızca olağandışı bir durumda kayıtlı kalır. Kuyruk yöneticisi başarısız olursa, kanal başlatıcısının kaydı her zaman kaldırılır.

z/OS IBM MQ ağında ARM ' nin kullanılması

Kuyruk yöneticinizi, kanal başlatıcıları ve ilişkili dinleyiciler kuyruk yöneticisi yeniden başlatıldığında otomatik olarak başlatılacak şekilde ayarlayabilirsiniz.

Hem LU 6.2 hem de TCP/IP iletişim protokolleri için aynı z/OS görüntüsünde kuyruk yöneticisinin tam otomatik olarak yeniden başlatılmasını sağlamak için:

- CSQINPX veri kümesine uygun START LISTENER komutunu ekleyerek, dinleyicilerinizi otomatik olarak başlatın.
- CSQINP2 veri kümesine uygun START CHINIT komutunu ekleyerek kanal başlatıcınızı otomatik olarak başlatın.

Bir kuyruk yöneticisini TCP/IP ya da LU6.2 ile yeniden başlatmak için bakınız:

- [“TCP/IP ile farklı bir z/OS görüntüsünde yeniden başlatma” sayfa 551](#)
- [“LU 6.2 ile farklı bir z/OS görüntüsünde yeniden başlatma” sayfa 552](#)

CSQINP2 ve CSQINPX veri kümeleriyle ilgili bilgi için [Görev 13: Kullanıma hazırlama giriş veri kümelerinin uyarlanması](#) başlıklı konuya bakın.

TCP/IP ile farklı bir z/OS görüntüsünde yeniden başlatma

İletişim protokolünüz olarak TCP/IP kullanıyorsanız ve sanal IP adreslerini kullanıyorsanız, bunları diğer z/OS görüntülerinde kurtarılacak şekilde yapılandırabilir ve bu şekilde bu kuyruk yöneticisine bağlanan kanalların herhangi bir değişiklik yapmadan yeniden bağlanmasına izin verebilirsiniz. Ters durumda, bir kuyruk yöneticisini farklı bir z/OS görüntüsüne taşıdıktan sonra, yalnızca kümeleri kullanıyorsanız ya da WLM dinamik Etki Alanı Adı Sistemi (DNS) mantıksal grup adını kullanarak bir kuyruk paylaşım grubuna bağlanıyorsanız, bir TCP/IP adresini yeniden ayarlayabilirsiniz.

- [Kümeleme kullanılırken](#)
- [Bir kuyruk paylaşım grubuna bağlanırken](#)

Kümeleme kullanılırken

z/OS ARM, aynı sistem şebekesinde farklı bir z/OS görüntüsünde kuyruk yöneticisini yeniden başlatarak bir sistem hatasına yanıt verir; bu sistem, özgün z/OS görüntüsüne göre farklı bir TCP/IP adresine sahiptir. Aşağıda, ARM yeniden başlatma işlemi tarafından farklı bir z/OS görüntüsüne taşıdıktan sonra bir kuyruk yöneticisinin TCP/IP adresini yeniden atamak için IBM MQ kümelerini nasıl kullanabileceğiniz açıklanmaktadır.

Bir istemci kuyruk yöneticisi kuyruk yöneticisi hatasını saptadığında (kanal hatası olarak), küme iletim kuyruğundaki uygun iletileri, hedef küme kuyruğunun farklı bir eşgörünümünü barındıran farklı bir sunucu kuyruk yöneticisine yeniden yerleştirerek yanıt verir. Ancak, sunucu kuyruk yöneticisi toplu iş sonu işlemi sırasında başarısız olduğu için, benzerlik kısıtları ya da şüpheli iletiler ile özgün sunucuya bağlı iletileri yeniden ayıramaz. Bu iletileri işlemek için aşağıdakileri yapın:

1. Her z/OS kuyruk yöneticisine farklı bir küme alıcı kanal adı ve farklı bir TCP/IP kapısı ayırın. İki sistemin z/OS görüntüsünde tek bir TCP/IP yığını paylaşabilmesi için her kuyruk yöneticisinin farklı bir kapiya gereksinimi vardır. Bunlardan biri, başlangıçta bu z/OS görüntüsünde çalışan kuyruk yöneticisidir, diğeri ise bir sistem hatasının ardından ARM 'nin z/OS görüntüsünde yeniden başlatacağı kuyruk yöneticisidir. ARM 'nin herhangi bir z/OS görüntüsünde herhangi bir kuyruk yöneticisini yeniden başlatabilmesi için her bir z/OS görüntüsünde her bir kapiyı yapılandırın.
2. Kanal başlatıcı başlatılırken gönderme yapılması gereken her kuyruk yöneticisi ve z/OS görüntü birleşimi için farklı bir kanal başlatıcı komutu giriş kütüğü (CSQINPX) yaratın.

Her CSQINPX kütüğü, o kuyruk yöneticisine özgü bir START LISTENER PORT (kapı) komutu ve o kuyruk yöneticisine ve z/OS görüntü birleşimine özgü bir küme alıcı kanalı için ALTER CHANNEL komutu içermelidir. ALTER CHANNEL komutunun bağlantı adını, yeniden başlatıldığı z/OS görüntüsünün TCP/IP adına ayarlaması gerekir. Bağlantı adının bir parçası olarak yeniden başlatılan kuyruk yöneticisine özgü kapı numarasını içermelidir.

Her kuyruk yöneticisinin başlatma JCL 'si, bu CSQINPX dosyası için sabit bir veri kümesi adına sahip olabilir ve her z/OS görüntüsü, paylaşılmayan bir DASD birimindeki her CSQINPX dosyasının farklı bir sürümüne sahip olmalıdır.

ARM yeniden başlatma işlemi gerçekleşirse, IBM MQ değiştirilen kanal tanımlamasını küme havuzuna yayınlar; bu da, sunucu kuyruk yöneticisiyle ilgilendiğini belirten tüm istemci kuyruk yöneticilerine yayınlar.

İstemci kuyruk yöneticisi, sunucu kuyruk yöneticisi hatasını kanal hatası olarak değerlendirir ve başarısız olan kanalı yeniden başlatmayı dener. İstemci kuyruk yöneticisi yeni sunucu bağlantı adını öğrendiğinde, kanal yeniden başlatma işlemi istemci kuyruk yöneticisini yeniden başlatılan sunucu kuyruk yöneticisine bağlar. İstemci kuyruk yöneticisi daha sonra iletilerini yeniden eşzamanlayabilir,

istemci kuyruk yöneticisinin iletim kuyruğundaki şüpheli iletileri çözebilir ve olağan işleme devam edebilir.

Bir kuyruk paylaşım grubuna bağlanırken

Bir kuyruk paylaşım grubuna TCP/IP dinamik Etki Alanı Ad Sistemi (DNS) mantıksal grup adı üzerinden bağlanırken, kanal tanımınızdaki bağlantı adı, fiziksel makinenin anasistem adı ya da IP adresi değil, kuyruk paylaşım grubunuzun mantıksal grup adını belirler. Bu kanal başlatıldığında, dinamik DNS ' ye bağlanır ve kuyruk paylaşım grubundaki kuyruk yöneticilerinden birine bağlanır. Bu işlem, IBM MQ for z/OS için kuyruk paylaşım grupları kullanılarak iletişim kurulması başlıklı konuda açıklanmıştır.

Olası olmayan bir görüntü arızası durumunda aşağıdakilerden biri ortaya çıkar:

- Hatalı görüntüdeki kuyruk yöneticileri, sistem şebekenizde çalışan dinamik DNS ' den kaydını kaldırıyor. Kanal, bağlantı hatasına REDENYOR durumuna girerek yanıt verir ve daha sonra, sistem şebekesinde çalışan dinamik DNS ' ye bağlanır. Dinamik DNS, geri kalan görüntülerde çalışmaya devam eden kuyruk paylaşım grubunun geri kalan üyelerinden birine gelen isteği ayırır.
- Kuyruk paylaşım grubundaki başka bir kuyruk yöneticisi etkin değilse ve ARM, kuyruk yöneticisini ve kanal başlatıcısını farklı bir görüntüde yeniden başlatırsa, grup dinleyicisi bu yeni görüntüden dinamik DNS ' ye kayıt olur. Bu, mantıksal grup adının (kanalın bağlantı adı alanından) dinamik DNS ' ye bağlandığı ve daha sonra aynı kuyruk yöneticisine bağlı olduğu ve şimdi farklı bir görüntüde çalıştığı anlamına gelir. Kanal tanımında değişiklik yapılması gerekmedi.

Bu tür bir kurtarma işleminin gerçekleştirilmesi için aşağıdaki noktaların belirtilmesi gerekir:

- z/OS işletim sistemi üzerinde dinamik DNS, sysplex içindeki z/OS görüntülerinden birinde çalışır. Bu görüntü başarısız olursa, dinamik DNS ' nin konfigürasyonu, sistem şebekesinde etkin olan ikincil bir ad sunucusunun birincil ad sunucusuna alternatif olarak çalışacağı şekilde tanımlanmalıdır. Birincil ve ikincil dinamik DNS sunucularına ilişkin bilgileri OS/390 SecureWay CS IP Configuration adlı elkitabında bulabilirsiniz.
- TCP/IP grup dinleyicisi, bu z/OS görüntüsünde kullanılmayacak belirli bir IP adresinde başlatılmış olabilir. Bu durumda, dinleyicinin yeni görüntüdeki farklı bir IP adresinde başlatılması gerekebilir. Sanal IP adresleri kullanıyorsanız, bunları diğer z/OS görüntülerinde kurtarılacak şekilde yapılandırabilirsiniz; böylece, START LISTENER komutunda herhangi bir değişiklik yapılması gerekmez.

LU 6.2 ile farklı bir z/OS görüntüsünde yeniden başlatma

Yalnızca LU 6.2 iletişim protokollerini kullanıyorsanız, sysplex içindeki farklı bir z/OS görüntüsünde bir kuyruk yöneticisi otomatik olarak yeniden başlatıldıktan sonra ağ bağlantısını etkinleştirmek için aşağıdaki yordamı gerçekleştirin:

- Sysplex içindeki her kuyruk yöneticisini benzersiz bir altsistem adıyla tanımlayın.
 - Sysplex içindeki her kanal başlatıcısı benzersiz bir LUNAME ile tanımlayın. Bu, hem kuyruk yöneticisi özniteliklerinde, hem de START LISTENER komutunda belirtilir.
- Not:** LUNAME, APPC yan çizelgesindeki bir girişi adlandırıyor ve bu da bunu gerçek LUNAME ile eşler.
- Sysplex içindeki her z/OS görüntüsü tarafından başvuru paylaşılan bir APPC yan çizelgesi ayarlayın. Bu, her kanal başlatıcısının LUNAME ' si için bir giriş içermelidir. Bununla ilgili bilgi için z/OS MVS Planning: APPC/MVS Management belgesine bakın.
 - Sistem şebekesi içindeki her kanal başlatıcı için bir APPCPM xx SYS1.PARMLIB üyesini, ilgili kanal başlatıcısı için APPC yan çizelge girişini etkinleştirmek üzere bir LUADD içerecek şekilde ayarlayın. Bu üyeler her z/OS resmi tarafından paylaşılmalıdır. Uygun SYS1.PARMLIB üyesi, aşağıdaki metinde açıklandığı gibi, farklı bir z/OS görüntüsünde kuyruk yöneticisi (ve kanal başlatıcısı) ARM yeniden başlatılırken otomatik olarak yayınlanan z/OS SET APPC= xxkomutuyla etkinleştirilir.
 - Her kanal başlatıcısı için bu SYS1.PARMLIB üyesinin xx sonekini belirtmek üzere LU62ARM kuyruk yöneticisi öznitelikliğini kullanın. Bu, kanal başlatıcısının LUNAME ' yi etkinleştirmek için gereken z/OS SET APPC= xx komutunu vermesine neden olur.

ARM ilkenizi, kanal başlatıcısını yalnızca z/OS görüntüsü açıkken başarısız olursa yeniden başlatacak şekilde tanımlayın; XCFAS adres alanıyla ilişkili kullanıcı kimliğinin IBM MQ komutunu START CHINIT komutunu verme yetkisi olmalıdır. z/OS görüntüsü de başarısız olursa kanal başlatıcısı otomatik olarak yeniden başlatmayın; bunun yerine, kanal başlatıcısını ve dinleyicilerini başlatmak için CSQINP2 ve CSQINPX veri kümelerindeki komutları kullanın.

z/OS İş birimlerinin el ile kurtarılması

Bir kuyruk paylaşım grubundaki iş birimlerini CICS, IMS, RRS ya da diğer kuyruk yöneticilerini el ile kurtarabilirsiniz. Kuyruk yöneticisine yönelik her bağlantıyla ilişkili iş birimlerinin durumunu görüntülemek için kuyruk yöneticisi komutlarını kullanabilirsiniz.

Bu konu, aşağıdaki konularla ilgili bilgi içerir:

- [“Bağlantıların ve iş parçacıklarının görüntülenmesi” sayfa 553](#)
- [“CICS kurtarma birimlerinin el ile kurtarılması” sayfa 553](#)
- [“IMS kurtarma birimlerinin el ile kurtarılması” sayfa 557](#)
- [“RRS kurtarma birimlerinin el ile kurtarılması” sayfa 559](#)
- [“Kuyruk paylaşım grubundaki başka bir kuyruk yöneticisinde kurtarma birimlerinin kurtarılması” sayfa 559](#)

Bağlantıların ve iş parçacıklarının görüntülenmesi

Kuyruk yöneticilerine ve ilişkili iş birimlerine yönelik bağlantılar hakkında bilgi almak için [DISPLAY CONN](#) komutunu kullanabilirsiniz. Şu anda neler olduğunu görmek için etkin iş birimlerini görüntüleyebilir ya da kuyruk yöneticisinin kapanmasına izin vermek için nelerin sonlandırılması gerektiğini görebilir ve çözümlenmemiş iş birimlerini görüntüleyebilirsiniz.

Etkin iş birimleri

Yalnızca etkin iş birimlerini görüntülemek için şunu kullanın:

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ ACTIVE)
```

Çözümlememiş iş birimleri

Çözülmemiş bir iş birimi ("belirsiz iş parçacığı" olarak da bilinir), iki aşamalı kesinleştirme işleminin ikinci geçişinde bulunan bir birimdir. Kaynaklar, IBM MQ içinde onun adına tutulur. Çözümlememiş iş birimlerini görüntülemek için şunu kullanın:

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

Çözümlememiş iş birimlerinin durumunu çözmek için dış müdahale gerekir. Bu, yalnızca kurtarma koordinatörünü (CICS, IMS ya da RRS) başlatmayı içerebilir ya da aşağıdaki bölümlerde açıklandığı gibi daha fazlasını içerebilir.

z/OS CICS kurtarma birimlerinin el ile kurtarılması

CICS bağdaştırıcısı yeniden başlatıldığında ne olacağını anlamak için bu konuyu kullanın ve daha sonra, ortaya çıkan çözümlenmemiş kurtarma birimleriyle nasıl başa çıkılacağını açıkla.

CICS bağdaştırıcısı yeniden başlatıldığında ne olur?

Bir bağlantı kesildiğinde, bağdaştırıcının *yeniden bağlanma işlemi* sırasında *yeniden başlatma aşamasından* geçmesi gerekir. Yeniden başlatma aşaması kaynakları yeniden eşitler. CICS ve IBM MQ arasındaki yeniden eşzamanlama, belirsiz iş birimlerinin tanımlanmasını ve çözülmesini sağlar.

Yeniden eşzamanlama aşağıdaki nedenlerle gerçekleştirilebilir:

- Dağıtılmış kuyruğa alma bileşeninden belirtik bir istek
- IBM MQ ile bağlantı kurulduğunda örtük istek

Yeniden eşzamanlamanın nedeni IBM MQ bağlantısıysa, olayların sırası şöyledir:

1. Bağlantı işlemi, IBM MQ' den belirsiz iş birimi (UOW) tanıtıcılarının listesini alır.
2. UOW tanıtıcıları konsolda CSQC313I iletilerinde görüntülenir.
3. UOW tanıtıcıları CICS' e iletilir.
4. CICS , her belirsiz UOW tanıtıcısı için bir yeniden eşzamanlama görevi (CRSY) başlatır.
5. Her belirsiz UOW için görevin sonucu konsolda görüntülenir.

Bağlanma işlemi sırasında görüntülenen iletileri denetlemeniz gerekir:

CSQC313I

Bir UOW ' nin şüphe içinde olduğunu gösterir.

CSQC400I

UOW ' yi tanımlar ve bunu aşağıdaki iletilerden biri izler:

- CSQC402I ya da CSQC403I UOW ' nin başarıyla çözüldüğünü (kesinleştirildiğini ya da geriletildiğini) gösterir.
- CSQC404E, CSQC405E, CSQC406E ya da CSQC407E UOW ' nin çözülmediğini gösterir.

CSQC409I

Tüm UOW ' lerin başarıyla çözümlendiğini gösterir.

CSQC408I

Tüm UOW ' lerin başarıyla çözümlenmediğini gösterir.

CSQC314I

Bir * ile vurgulanan UOW tanıtıcılarının otomatik olarak çözümlenmediğini uyarır. Bu kullanıcı arabirimleri, yeniden başlatıldığında dağıtılmış kuyruğa alma bileşeni tarafından belirtik olarak çözülmelidir.

Şekil 37 sayfa 555 içinde, z/OS konsolunda görüntülenen bir yeniden başlatma iletileri kümesi örneği gösterilmektedir.

```

CSQ9022I +CSQ1 CSQYASCP ' START QMGR' NORMAL COMPLETION
+CSQC323I VICIC1 CSQCCON CONNECT received from TERMID=PB62 TRANID=CKCN
+CSQC303I VICIC1 CSQCCON CSQCSERV loaded. Entry point is 850E8918
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F0E2178D25 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F055B2AC25 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6EFFD60D425 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F07AB56D22 is in doubt
+CSQC307I VICIC1 CSQCCON Successful connection to subsystem VC2
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAD18) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAA10) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BA708) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAE88) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAB80) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA878) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA570) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA268) connect
successful
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F0E2178D25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F055B2AC25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F07AB56D22
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6EFFD60D425
+CSQC409I VICIC1 CSQCTRUE Resynchronization completed successfully

```

Şekil 37. Örnek yeniden başlatma iletileri

CSQC313I iletilerinin toplam sayısı, CSQC402I artı CSQC403I iletilerinin toplam sayısına eşit olmalıdır. Toplamlar eşit değilse, bağlantı işleminin çözemediği UOW ' lar vardır. Çözümlemeyen UOW ' lar CICS (örneğin, soğuk başlatma) ya da IBM MQ ile ilgili ya da kuyruğa alma ile ilgili sorunlardan kaynaklanır. Bu sorunlar düzeltilindiğinde, bağlantıyı kestikten sonra yeniden bağlanarak başka bir yeniden eşzamanlama başlatabilirsiniz.

Diğer bir seçenek olarak, RESOLVE INDOUBT komutunu ve CSQC400I iletilerinde gösterilen UOW tanıtıcısını kullanarak, bekleyen her UOW ' yi kendiniz çözebilirsiniz. Daha sonra, CICS içindeki *kurtarma tanımlayıcıları birimini* temizlemek için bir bağlantı kesme ve bağlanma başlatmanız gerekir. UOW ' leri el ile çözmek için UOW ' nin doğru sonucunu bilmeniz gerekir.

Çözülmemiş UOW ' larla ilişkili tüm iletiler IBM MQ tarafından kilitlenir ve hiçbir Toplu İş, TSO ya da CICS görevi bunlara erişemez.

CICS başarısız olursa ve acil durumda yeniden başlatma gerekirse, CICS sisteminin GENERIC APPLID ' i *değiştirmeyin* . Bunu yapar ve IBM MQ ' a yeniden bağlanırsanız, IBM MQ ile veri bütünlüğü garanti edilemez. Bunun nedeni, IBM MQ ' in yeni CICS yönetim ortamını farklı bir CICS olarak işlemesi olabilir (APPLID farklı olduğu için). Bundan sonra, belirsiz bir çözüm yanlı CICS günlüğüne dayalı olur.

CICS kurtarma birimlerinin el ile çözümlenmesi

Bağdaştırıcı olağandışı bitirse, CICS ve IBM MQ oluşturma işlemi, olağandışı sona neden olan altsisteme bağlı olarak, dinamik olarak ya da yeniden başlatma sırasında listeler.

Not: İş birimlerini göstermek için DFH\$INDB örnek programını kullanırsanız, IBM MQ UOW ' larını her zaman doğru şekilde göstermediğini bulabilirsiniz.

CICS IBM MQ ile bağlantı kurduğunda, çözümlenmemiş bir ya da daha fazla kurtarma birimi olabilir.

Konsola aşağıdaki iletilerden biri gönderilir:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E
- CSQC408I

Bu iletilerin anlamı hakkında ayrıntılı bilgi için [CICS bağdaştırıcısı ve Köprü iletileri](#) iletilerine bakın.

CICS , bağlantı başlatılırken çözülmeyen kurtarma birimlerinin ayrıntılarını saklar. Bir girdi, IBM MQ tarafından sunulan listede artık görüntülenmediğinde temizlenir.

CICS ' in çözemediğini kurtarma birimleri, IBM MQ komutları kullanılarak el ile çözülmelidir. Bu manuel yordam, yalnızca işletim hataları ya da yazılım sorunlarının otomatik çözümlenmeyi engellediği durumlarda gerekli olduğu için, kuruluş içinde nadiren kullanılır. *Belirsiz çözüm sırasında bulunan tutarsızlıklar araştırılmalıdır.*

Kurtarma birimlerini çözmek için:

1. Aşağıdaki komutu kullanarak IBM MQ adresinden kurtarma birimlerinin bir listesini edinin:

```
+CSQ1 DISPLAY CONN( * ) WHERE(UOWSTATE EQ UNRESOLVED)
```

Aşağıdaki iletiyi alırsınız:

```
CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN(BC85772CBE3E0001)
EXTCONN(C3E2D8C3C7D9F0F940404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2005-02-04)
UOWLOGTI(10.17.44)
UOWSTDA(2005-02-04)
UOWSTTI(10.17.44)
UOWSTATE(UNRESOLVED)
NID(IYRCSQ1 .BC8571519B60222D)
EXTURID(BC8571519B60222D)
QMURID(0000002BDA50)
URTYPE(CICS)
USERID(MQTEST)
APPLTAG(IYRCSQ1)
ASID(0000)
APPLTYPE(CICS)
TRANSID(GP02)
TASKNO(0000096)
END CONN DETAILS
```

CICS bağlantıları için NID, eşitleme noktası günlük girişleri yazılırken CICS applid 'den ve CICS tarafından sağlanan benzersiz bir numaradan oluşur. Bu benzersiz sayı, hem CICS sistem günlüğüne, hem de IBM MQ syncpoint işleme zamanında günlüğe yazılan kayıtlarda saklanır. Bu değere CICS içinde *kurtarma simgesi* olarak atıfta bulunulur.

2. Belirli bir kurtarma birimiyle ilgili girişler için CICS günlüğünü tarayın.

Kurtarma simgesi alanının (JCSRMTKN) ağ tanıtıcısından elde edilen değere eşit olduğu, görevle ilgili kuruluşa ilişkin bir PREPARE kaydı arayın. Ağ tanıtıcısı, DISPLAY CONN komut çıkışında IBM MQ tarafından sağlanır.

Kurtarma birimlerine ilişkin CICS günlüğündeki PREPARE kaydı CICS görev numarasını sağlar. Bu CICS görevine ilişkin günlükteki diğer tüm girdiler bu sayı kullanılarak bulunabilir.

Günlüğü tararken CICS günlük yazdırma yardımcı programı DFHJUP ' u kullanabilirsiniz. Bu programı kullanmaya ilişkin ayrıntılar için bkz. *CICS Operations and Utilities Guide*.

3. Belirli bir kurtarma birimiyle ilgili NID ' ye sahip kayıtlar için IBM MQ günlüğünü tarayın. Daha sonra, bu kurtarma birimine ilişkin günlük kayıtlarının geri kalanını almak için bu kayıttaki URID değerini kullanın.

IBM MQ günlüğünü tararken, IBM MQ başlatma iletisi CSQJ001I 'nin bu oturum için başlangıç RBA' sını sağladığını unutmayın.

Günlük kayıtlarını yazdırma programı (CSQ1LOGP) bu amaçla kullanılabilir.

4. Gerekliyse, IBM MQ içinde belirsiz bir çözüm yapın.

IBM MQ , IBM MQ RESOLVE INDOUBT komutunu kullanarak bir kurtarma birimine ilişkin kurtarma işlemini gerçekleştirmek üzere yönlendirilebilir.

Belirli bir *bağlantı-adı* ile ilişkili tüm iş parçacıklarını kurtarmak için NID (*) seçeneğini kullanın.

Komut, iş parçacığının kesinleştirildiğini ya da geriletildiğini gösteren aşağıdaki iletilerden birini üretir:

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id ABORT SCHEDULED
```

Belirsiz bir çözüm gerçekleştirirken CICS ve bağdaştırıcı, yalnızca IBM MQ kaynakları etkilendiği için kurtarma birimlerini kesinleştirmek ya da geri almak için IBM MQ komutlarını bilmez. Ancak CICS , IBM MQ tarafından çözülemeyen belirsiz iş parçacıklarına ilişkin ayrıntıları saklar. Bu bilgiler, gösterilen liste boş olduğunda ya da liste CICS ' in ayrıntılarına sahip olduğu bir kurtarma birimini içermediğinde temizlenir.

z/OS IMS kurtarma birimlerinin el ile kurtarılması

IMS bağdaştırıcısı yeniden başlatıldığında ne olacağını anlamak için bu konuyu kullanın ve daha sonra, ortaya çıkan çözümlenmemiş kurtarma birimleriyle nasıl başa çıkılacağını açıkla.

IMS bağdaştırıcısı yeniden başlatıldığında ne olur?

IBM MQ bağlantısı yeniden başlatıldığında, bir kuyruk yöneticisi yeniden başlatıldıktan ya da bir IMS / START SUBSYS komutu uygulandıktan sonra, IMS aşağıdaki yeniden eşzamanlama işlemini başlatır:

1. IMS , kesinleştirme ya da geriletme çözüm parametresiyle, IBM MQ IMS bağdaştırıcısına her defasında bir tane olmak üzere inandığı iş birimi tanıtıcılarının listesini gösterir.
2. IMS bağdaştırıcısı çözüm isteğini IBM MQ 'e iletir ve sonucu IMS' e geri bildirir.
3. Tüm IMS çözüm isteklerini işledikten sonra IMS bağdaştırıcısı, IBM MQ 'un IMS sistemi tarafından başlatılan tüm UOW' ların listesinden IBM MQ alınır. Bunlar, CSQQ008Iiletisindeki IMS ana uçbirimine raporlanır.

Not: Bir UOW şüpheliyken, ilişkili IBM MQ iletileri IBM MQ tarafından kilitlenir ve herhangi bir uygulama tarafından kullanılamaz.

IMS kurtarma birimlerinin el ile çözümlenmesi

IMS IBM MQ ile bağlantı kurduğunda, IBM MQ çözümlenmemiş bir ya da daha fazla belirsiz kurtarma birimine sahip olabilir.

IBM MQ , IMS ' in çözemediği belirsiz kurtarma birimlerine sahipse, IMS ana uçbiriminde aşağıdaki ileti yayınlanır:

```
CSQQ008I nn units of recovery are still in doubt in queue manager qmgr-name
```

Bu ileti yayınlandıysa, IMS soğuk başlatılmış ya da tamamlanmamış bir günlük manyetik bantı ile başlatılmış olabilir. Bu ileti, bir yazılım hatası ya da başka bir altsistem hatası nedeniyle IBM MQ ya da IMS olağandışı sonlandıysa da yayınlanabilir.

CSQQ008I iletisi alındıktan sonra:

- Bağlantı etkin kalır.
- IMS uygulamaları IBM MQ kaynaklarına erişmeye devam edebilir.
- Bazı IBM MQ kaynakları kilitli kalır.

Belirsiz iş parçacığı çözülmezse, IMS ileti kuyrukları birikmeye başlayabilir. IMS kuyrukları sığasını doldurursa, IMS sona erer. Bu olası zorluğun farkında olmanız ve belirsiz kurtarma birimleri tam olarak çözülünceye kadar IMS ' i izlemeniz gerekir.

Kurtarma yordamı

IMS iş birimlerini kurtarmak için aşağıdaki yordamı kullanın:

1. IMS günlüğünü /SWI OLDS kullanarak kapatmaya zorlayın ve IMS günlüğünü arşivleyin. Önceki IMS günlük manyetik bandındaki kayıtları yazdırmak için DFSERA10yardımcı programını kullanın. X ' 3730 ' günlük kayıtları, bir phase-2 kesinleştirme isteği ve tipi X ' 38 ' günlük kayıtlarının bir durdurma isteğini gösterdiğini belirtir. Her bağımlı bölgedeki son işlem için istenen işlemi kaydedin.
2. DL/I toplu işini, kesinleştirme noktasına ulaşmamış olan her PSB ' yi geri almak için çalıştırın. İşlemler hala işlenmekte olduğu için işlem biraz zaman alabilir. Ayrıca, işlemin geri kalanını ve ileti kuyruklarının geri kalanını etkileyebilecek sayıda kaydı kilitleyebilir.
3. Aşağıdaki komutu kullanarak IBM MQ ' den gelen belirsiz kurtarma birimlerinin bir listesini üretin:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

Aşağıdaki iletiyi alırsınız:

```
CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3E2C5C3F240404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2005-02-15)
UOWLOGTI(16.39.43)
UOWSTDA(2005-02-15)
UOWSTTI(16.39.43)
UOWSTATE(UNRESOLVED)
NID(IM8F .BC45A794D3810344)
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0000)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)
```

IMS için NID, IMS bağlantı adından ve IMStarafından sağlanan benzersiz bir numaradan oluşur. Bu değere IMS içinde *kurtarma simgesi* olarak atıfta bulunulur. Daha fazla bilgi için [IMS belgelerine](#) bakın.

4. DISPLAY THREAD iletilerinde görüntülenen NID 'leri (IMSID artı onaltılı OASN) DFSERA10 çıkışında gösterilen OASN' lerle (4 byte onlu) karşılaştırın. Kesinleştirmeye ya da geri çekilmeye karar verin.
5. IBM MQ içinde [RESOLVE INDOUBT](#) komutuyla aşağıdaki gibi belirsiz bir çözüm gerçekleştirin:

```
RESOLVE INDOUBT( connection-name )
ACTION(COMMIT|BACKOUT)
NID( network-id )
```

bağlantı-adi ile ilişkili tüm iş parçacıklarını kurtarmak için NID (*) seçeneğini kullanın. Komut, iş parçacığının kesinleştirildiğini ya da geriletildiğini belirtmek için aşağıdaki iletilerden biriyle sonuçlanır:

```
CSQV414I  THREAD network-id COMMIT SCHEDULED
CSQV415I  THREAD network-id BACKOUT SCHEDULED
```

Belirsiz bir çözüm gerçekleştirirken, IMS ve bağdaştırıcı, yalnızca IBM MQ kaynakları etkilendiği için belirsiz kurtarma birimlerini kesinleştirmek ya da geri almak için IBM MQ komutlarını bilmez.

z/OS RRS kurtarma birimlerinin el ile kurtarılması

Belirsiz RRS kurtarma birimlerinin olup olmadığını ve bu kurtarma birimlerinin el ile nasıl çözüleceğini belirlemek için bu konuyu kullanın.

RRS IBM MQ' e bağlandığında, IBM MQ çözümlenmemiş bir ya da daha fazla belirsiz kurtarma birimine sahip olabilir. IBM MQ , RRS ' nin çözemediği belirsiz kurtarma birimlerine sahipse, z/OS konsolunda aşağıdaki iletilerden biri yayınlanır:

- CSQ3011I
- CSQ3013I
- CSQ3014I
- CSQ3016I

Hem IBM MQ hem de RRS, belirsiz kurtarma birimlerine ilişkin bilgileri ve bunları el ile çözme tekniklerini görüntülemek için araçlar sağlar.

IBM MQ içinde, belirsiz IBM MQ iş parçacıklarına ilişkin bilgileri görüntülemek için DISPLAY CONN komutunu kullanın. Komutun çıkışı, RRS eşgüdümçü olarak kullanılan IBM MQ iş parçacıklarına ilişkin RRS kurtarma tanıtıcılarını içerir. Bu, kurtarma biriminin sonucunu belirlemek için kullanılabilir.

IBM MQ belirsiz iş parçacığını el ile çözmek için RESOLVE INDOUBT komutunu kullanın. Bu komut, doğru kararın ne olduğunu belirledikten sonra kurtarma birimini kesinleştirmek ya da geri almak için kullanılabilir.

z/OS Kuyruk paylaşım grubundaki başka bir kuyruk yöneticisinde kurtarma birimlerinin kurtarılması

Bir kuyruk paylaşım grubundaki diğer kuyruk yöneticilerine ilişkin kurtarma birimlerini tanımlamak ve el ile kurtarmak için bu konuyu kullanın.

Bir kuyruk paylaşım grubunun üyesi olan bir kuyruk yöneticisi başarısız olursa ve yeniden başlatılamazsa, gruptaki diğer kuyruk yöneticileri eşdüze kurtarma gerçekleştirebilir ve bu gruptan devralabilir. Ancak, kurtarma biriminin son atılması yalnızca başarısız olan kuyruk yöneticisi tarafından bilindiğinden, kuyruk yöneticisinde eşdüze kurtarma tarafından çözülemeyecek belirsiz kurtarma birimleri olabilir. Bu kurtarma birimleri, kuyruk yöneticisi sonunda yeniden başlatıldığında çözülür, ancak o zamana kadar şüphe içinde kalır.

Bu, belirli kaynakların (örneğin, iletiler) kilitlenerek gruptaki diğer kuyruk yöneticileri tarafından kullanılamaz kılınabileceği anlamına gelir. Bu durumda, etkin olmayan kuyruk yöneticisinde bu iş birimlerini görüntülemek için DISPLAY THREAD komutunu kullanabilirsiniz. İletileri gruptaki diğer kuyruk yöneticilerinin kullanımına sunmak için bu kurtarma birimlerini el ile çözmek istiyorsanız, RESOLVE INDOUBT komutunu kullanabilirsiniz.

Belirsiz kurtarma birimlerini görüntülemek için DISPLAY THREAD komutunu verdiğinizde, etkin olmayan kuyruk yöneticisinin adını belirtmek için QMNAME anahtar sözcüğünü kullanabilirsiniz. Örneğin, aşağıdaki komutu çalıştırırsanız:

```
+CSQ1 DISPLAY THREAD(*) TYPE(INDOUBT) QMNAME(QM01)
```

Aşağıdaki iletileri alırsınız:

```
CSQV436I +CSQ1 INDOUBT THREADS FOR QM01 -
NAME  THREAD-XREF  URID NID
USER1  0000000000000000000000000000 CSQ:0001.0
USER2  0000000000000000000000000000 CSQ:0002.0
DISPLAY THREAD REPORT COMPLETE
```

Belirtilen kuyruk yöneticisi etkinse, IBM MQ belirsiz iş parçacıklarına ilişkin bilgi döndürmez, ancak aşağıdaki iletiyi verir:

```
CSQV435I CANNOT USE QMNAME KEYWORD, QM01 IS ACTIVE
```

Belirsiz iş parçacıklarını el ile çözmek için IBM MQ komutunu RESOLVE INDOUBT komutunu kullanın. Komutta etkin olmayan kuyruk yöneticisinin adını belirtmek için QMNAME anahtar sözcüğünü kullanın.

Bu komut, kurtarma birimini kesinleştirmek ya da geri almak için kullanılabilir. Komut, yalnızca kurtarma biriminin paylaşılan kısmını çözer; yerel iletiler etkilenmez ve kuyruk yöneticisi yeniden başlatılıncaya ya da CICS, IMSya da RRS toplu işine yeniden bağlanıncaya kadar kilitli kalır.

z/OS IBM MQ ve IMS

IBM MQ , IMS, IBM MQ - IMS bağdaştırıcısı ve IBM MQ - IMS köprüsü ile arabirim oluşturan iki bileşen sağlar. Bu bileşenlere genellikle IMS bağdaştırıcısı ve IMS köprüsü denir.

z/OS IMS bağdaştırıcısının çalıştırılması

IBM MQ ' i IMS sistemlerine bağlayan IMS bağdaştırıcısının nasıl çalıştırılacağını anlamak için bu konuyu kullanın.

Not: IMS bağdaştırıcısı herhangi bir işlem ve denetim panosu içermiyor.

Bu konu aşağıdaki kısımları içerir:

- [“IMS bağlantılarını denetleme” sayfa 560](#)
- [“IMS denetim bölgesinden bağlanma” sayfa 561](#)
- [“Belirsiz kurtarma birimlerinin görüntülenmesi” sayfa 563](#)
- [“IMS bağımlı bölge bağlantılarını denetleme” sayfa 565](#)
- [“IMS bağlantısı kesiliyor” sayfa 567](#)
- [“IMS tetikleyici izleme programının denetlenmesi” sayfa 567](#)

z/OS IMS bağlantılarını denetleme

IBM MQbağlantısını denetleyen ve izleyen IMS işletmen komutlarını anlamak için bu konuyu kullanın.

IMS , IBM MQbağlantısını denetlemek ve izlemek için aşağıdaki işletmen komutlarını sağlar:

/SUBSYS DEĞİŞTİR

Belirsiz bir kurtarma birimini IMSiçinden siler.

/OASN ALT SINIRLARINI GÖRÜNTÜLE

Bekleyen kurtarma öğelerini görüntüler.

/ALT SINIR GÖRÜNTÜLE

Bağlantı durumunu ve iş parçacığı etkinliğini görüntüler.

/START SUBSYS

IMS denetim bölgesini bir kuyruk yöneticisine bağlar.

/STOP SUBSYS

Bir kuyruk yöneticisiyle IMS bağlantısını keser.

/İZLE

IMS izlemesini denetler.

Bu komutlarla ilgili daha fazla bilgi için kullandığınız IMS düzeyine ilişkin *IMS/ESA Operator's Reference* adlı elkitabına bakın.

IMS komut yanıtları, komutun verildiği uçbirime gönderilir. IMS komutlarını verme yetkisi IMS güvenliğine dayalıdır.

z/OS IMS denetim bölgesinden bağlanma

IMS 'den IBM MQ' e bağlanmak için kullanılabilecek mekanizmaları anlamak için bu konuyu kullanın.

IMS , denetim bölgesinden IMSkullanan her kuyruk yöneticisine bir bağlantı oluşturur. IMS aşağıdaki yollardan biriyle bağlantı kurmak için etkinleştirilmelidir:

- Aşağıdakilerden biri sırasında otomatik olarak:
 - Soğuk başlatma.
 - IMS kapatıldığında IBM MQ bağlantısı etkinse, IMSsıcak başlatma.
- IMS komutuna yanıt olarak:

```
/START SUBSYS sysid
```

Burada *sysid* , kuyruk yöneticisi adıdır.

Kuyruk yöneticisinin etkin olup olmadığına bakılmaksızın komut verilebilir.

Kuyruk yöneticisine ilk IBM MQ API çağrısı yapıncaya kadar bağlantı kurulmaz. Bu zamana kadar, IMS komutu /DIS SUBSYS 'NOT CONN' olarak durumu gösterir.

IMS ' u ve kuyruk yöneticisini başlatma sıranız önemli değil.

Kuyruk yöneticisi STOP QMGR komutu, IMS komutu ya da olağandışı bir sonla durdurulursa, IMS kuyruk yöneticisiyle bağlantı otomatik olarak yeniden etkinleştirilemez. Bu nedenle, IMS komutunu /START SUBSYS kullanarak bağlantı kurmanız gerekir.

Kuyruk yöneticisi konsolu günlüğünde buna benzer bir IMS komutu görüldüyse:

```
MODIFY IMS*,SS*
```

IMS ana günlüğünü denetleyin ve IBM MQ ' in IMS Adapter MODIFY komutlarını yayınlamak için RACF yetkisine sahip olduğundan emin olun.

Bağdaştırıcının kullanıma hazırlanması ve kuyruk yöneticisine bağlanması

Bağdaştırıcı, IMS dış Altsistem Bağlantı Olanakları kullanılarak IMS denetim ve bağımlı bölgelerine yüklenen bir modül kümesidir.

Bu yordam bağdaştırıcıyı başlatır ve kuyruk yöneticisine bağlanır:

1. IMSiçinden altsistem üyesini (SSM) okuyun.-PROCLIB. Seçilen SSM bir IMS EXEC parametresi. Üyede, IMS ' un bağlanabileceği her kuyruk yöneticisi için bir giriş vardır. Her bir girdi, IBM MQ bağdaştırıcısına ilişkin denetim bilgilerini içerir.
2. IMS bağdaştırıcısını yükleyin.

Not: IMS , SSM üyesinde tanımlanan her IBM MQ eşgörünümü için bağdaştırıcı modüllerinin bir kopyasını yükler.

3. IBM MQ için dış altsistem görevini ekleyin.

4. Bağdaştırıcıyı, bağlantı adı olarak CTL EXEC parametresi (IMSID) ile çalıştırın.

İşlem, bağlantının kullanıma hazırlama işleminin bir parçası ya da IMS komutunun bir sonucu/START SUBSYS ile aynıdır.

IMS bağlantı kurmaya çalışırken kuyruk yöneticisi etkinse aşağıdaki iletiler gönderilir:

• z/OS konsoluna:

```
DFS3613I ESS TCB INITIALIZATION COMPLETE
```

• IMS ana uçbirimine:

```
CSQQ000I IMS/TM imsid connected to queue manager ssnm
```

IMS bağlantıyı kurmaya çalıştığı anda ve *kuyruk yöneticisi etkin olmadığı anda*, IMS ana uçbirimine bir uygulama MQI çağrısı yaptığı anda aşağıdaki iletiler gönderilir:

```
CSQQ001I IMS/TM imsid not connected to queue manager ssnm.  
Notify message accepted  
DFS3607I MQM1 SUBSYSTEM ID EXIT FAILURE, FC = 0286, RC = 08,  
JOBNAME = IMSEMPR1
```

IMS bağlantısı başlatıldığında ya da sistem başlatıldığında DFS3607I iletileri alırsanız, bu, kuyruk yöneticisinin kullanılmadığını gösterir. Çok sayıda iletinin oluşturulmasını önlemek için aşağıdakilerden birini yapmanız gerekir:

1. İlgili kuyruk yöneticisini başlatın.

2. IMS komutunu verin:

```
/STOP SUBSYS
```

Böylece IMS kuyruk yöneticisine bağlanmayı beklemez.

İkparçası ya: iklerinde yine (ikisinde de), DFS3607I iletisi ve ilişkili CSQQ001I iletisi, bölgede her zamanlanan bir iş için ve kuyruk yöneticisine her bağlantı isteğinde bulunulduğunda bir uygulama tarafından yayınlanır.

İş parçacığı eki

Bir MPP ya da IFP bölgesinde, IMS , ilk uygulama programı o bölgeye zamanlandığında, o uygulama programı bir IBM MQ çağrısı yapmasa da iş parçacığı bağlantısı kurar. Bir BMP bölgesinde, uygulama ilk IBM MQ çağrısını (MQCONN ya da MQCONNX) yaptığı anda iş parçacığı bağlantısı kurulur. Bu iş parçacığı, bölgenin süresi boyunca ya da bağlantı durduruluncaya kadar korunur.

İletiyi yönlendirilen ve ileti odaklı olmayan bölgeler için, iş parçacığıyla ilişkili kurtarma iş parçacığı çapraz başvuru tanıtıcısı *Thread-xref*:

```
PSTid + PSBname
```

Burada:

PSTid

Bölüm belirtimi çizelgesi bölge tanıtıcısı

PSBname

Program belirtimi öbek adı

Bağlantı tanıtıcılarını IBM MQ komutlarında benzersiz tanıtıcılar olarak kullanabilirsiniz; bu durumda IBM MQ , bu tanıtıcıları otomatik olarak oluşturduğu herhangi bir işletmen iletime ekler.

z/OS Belirsiz kurtarma birimlerinin görüntülenmesi

Kurtarma birimlerini belirsiz bir şekilde görüntüleyebilir ve kurtarma girişiminde bulunabilirsiniz.

Bu konudaki belirsiz kurtarma birimlerini listelemek ve kurtarmak için kullanılan işletim adımları yalnızca nispeten basit durumlar içindir. IMS' e bağlıyken kuyruk yöneticisi olağandışı bir şekilde sona ererse, IBM MQ farkında olmadan IMS işi kesinleştirebilir ya da geri çekebilir. Kuyruk yöneticisi yeniden başlatıldığında, bu iş *şüpheli* olarak adlandırılır. İşin durumu hakkında bir karar verilmelidir.

Belirsiz kurtarma birimlerinin listesini görüntülemek için şu komutu verin:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

IBM MQ , aşağıdaki gibi bir iletiyle yanıt verir:

```
CSQM201I +CSQ1 CSQMDRTC DIS CONN DETAILS
CONN(BC0F6125F5A30001)
EXTCONN(C3E2D8C3C3E2D8F140404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-11-02)
UOWLOGTI(12.27.58)
UOWSTDA(2004-11-02)
UOWSTTI(12.27.58)
UOWSTATE(UNRESOLVED)
NID(CSQ1CHIN.BC0F5F1C86FC0766)
EXTURID(00000000000001F000000007472616E5F6964547565204E6F762020...)
QMURID(0000000026232)
URTYPE(XA)
USERID( )
APPLTAG(CSQ1CHIN)
ASID(0000)
APPLTYPE(CHINIT)
CHANNEL( )
CONNNAME( )
END CONN DETAILS
```

Bu iletideki özniteliklerin açıklaması için [DISPLAY CONN](#) komutunun açıklamasına bakın.

Belirsiz kurtarma birimlerinin kurtarılması

Belirsiz kurtarma birimlerini kurtarmak için şu komutu verin:

```
+CSQ1 RESOLVE INDOUBT( connection-name ) ACTION(COMMIT|BACKOUT)
NID( net-node.number )
```

Burada:

connection-name

IMS sistem tanıtıcısı.

İŞLEM

Bu kurtarma biriminin kesinleştirileceğini (COMMIT) ya da geri çekileceğini (BACKOUT) belirtir.

net-node.number

İlişkili net-node.number.

RESOLVE INDOUBT komutunu yayınladığınızda, aşağıdaki iletilerden biri görüntülenir:

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id BACKOUT SCHEDULED
```

Artık kurtarma girişleri çözümleniyor

Belirli zamanlarda IMS , artık kurtarma girişlerinin (RRE) bir listesini oluşturur. RRE 'ler, IBM MQ ' in şüphe içinde olabileceği kurtarma birimleridir. Çeşitli durumlarda ortaya çıkar:

- Kuyruk yöneticisi etkin değilse, IMS , kuyruk yöneticisi etkin oluncaya kadar çözülemeyecek RRE ' lere sahiptir. Bu RRE ' ler sorun değil.
- Kuyruk yöneticisi etkinse ve IMS'e bağlıysa ve IMS , IBM MQ ' in kesinleştirdiği işi geri çevirirse, IMS bağdaştırıcısı CSQQ010Eiletisini yayınlar. İki sistemdeki verilerin tutarlı olması gerekiyorsa, bir sorun vardır. Bu sorunu çözüme hakkında bilgi için bkz. [“IMS kurtarma birimlerinin el ile kurtarılması” sayfa 557.](#)
- Kuyruk yöneticisi etkinse ve IMS'e bağlıysa, bu sorunla ilgili herhangi bir ileti sizi bilgilendirmemiş olsa da, RRE' ler olabilir. IMS ile IBM MQ bağlantısı kurulduktan sonra, bir sorun olup olmadığını öğrenmek için aşağıdaki IMS komutunu verebilirsiniz:

```
/DISPLAY OASN SUBSYS sysid
```

RRE ' yi temizlemek için aşağıdaki IMS komutlarından birini verin:

```
/CHANGE SUBSYS sysid RESET
/CHANGE SUBSYS sysid RESET OASN nnnn
```

Burada *nnnn* , +CSQ1 DISPLAY komutunuza yanıt olarak listelenen kaynak uygulama sıra numarasıdır. Bu, program eşgörünümünün zamanlama numarasıdır ve son IMS soğuk başlatmasından bu yana program başlatma sırasındaki yerini verir. IMS , aynı zamanlama numarasına sahip iki belirsiz kurtarma birimine sahip olamaz.

Bu komutlar IMS durumunu sıfırlar; Bu, IBM MQ ile herhangi bir iletişimle sonuçlanmaz.

IMS bağımlı bölge bağlantılarını denetleme

IMS ile IBM MQarasındaki bağlantıları denetleyebilir, izleyebilir ve gerektiğinde sonlandırabilirsiniz.

IMS bağımlı bölge bağlantılarının denetlenmesi aşağıdaki etkinlikleri içerir:

- [Bağımlı bölgelerden bağlanma](#)
- [Bölge hatası seçenekleri](#)
- [Bağlantılarda etkinliği izleme](#)
- [Bağımlı bölgelerle bağlantıyı kesme](#)

Bağımlı bölgelerden bağlanma

Denetim bölgesinde kullanılan IMS bağdaştırıcısı da bağımlı bölgelere yüklenir. Her bir bağımlı bölgeden IBM MQile bağlantı kurulur. Bu bağlantı, IBM MQ ve IMS işinin bağlılığını koordine etmek için kullanılır. Bağlantıyı başlatmak ve kurmak için IMS aşağıdakileri yapar:

1. Altsistem üyesini (SSM) IMSiçinden okur.-PROCLIB.

Bağımlı bölge EXEC parametresinde bir altsistem üyesi belirlenebilir. Belirtilmezse, denetim bölgesi SSM kullanılır. Bölge IBM MQ' a hiçbir zaman bağlanamayacaksa, bağdaştırıcının yüklenmesini önlemek için giriş olmayan bir üye belirtin.

2. IBM MQ bağdaştırıcısını yükler.

Toplu ileti programı için, uygulama ilk ileti sistemi komutunu verinceye kadar yükleme işlemi gerçekleştirilmez. Bu sırada IMS , bağlantıyı kurmaya çalışır.

İleti işleme programı bölgesi ya da IMS hızlı yol bölgesi için, bölge kullanıma hazırlandığında bu girişim yapılır.

Bölge hata seçenekleri

Kuyruk yöneticisi etkin değilse ya da ilk ileti sistemi komutu uygulama programlarından gönderildiğinde kaynaklar kullanılmıyorsa, yapılan işlem SSM girişinde belirtilen hata seçeneğine bağlıdır. Seçenekler şunlardır:

R

Uygulamaya uygun dönüş kodu gönderilir.

Q

Uygulama, U3051olağandışı bitiş koduyla sona erer. Giriş iletisi yeniden kuyruğa alındı.

A

Uygulama U3047olağandışı bitiş koduyla sona eriyor. Giriş iletisi atılır.

Bağlantılarda etkinliği izleme

Bir uygulama ilk başarılı IBM MQ isteğini yaptığında bağımlı bölgeden bir iş parçacığı oluşturulur. IBM MQ' den aşağıdaki komutu girerek, bağlantıları ve bunları kullanmakta olan uygulamalarla ilgili bilgileri görüntüleyebilirsiniz:

```
+CSQ1 DISPLAY CONN(*) ALL
```

Komut, aşağıdaki gibi bir ileti üretir:

```
CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3C3E2D8F140404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-12-15)
UOWLOGTI(16.39.43)
UOWSTDA(2004-12-15)
UOWSTTI(16.39.43)
UOWSTATE(ACTIVE)
NID( )
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0049)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)
```

Denetim bölgesi için *thread-xref*, CONTROL özel değeridir. Bağımlı bölgeler için, PSBname ile bitleştirilmiş PSTid 'dir. *auth-id*, iş kartındaki kullanıcı alanı ya da z/OS başlatılan yordamlar çözümgesindeki kimlik.

Görüntülenen listeye ilişkin açıklamalar için, [IBM MQ for z/OS iletileri](#), tamamlama ve neden kodları belgelerinde CSQV402I iletilisinin tanımına bakın.

IMS, IBM MQ bağlantısını izlemek için bir görüntü komutu sağlar. Her bağımlı bölge bağlantısında hangi programın etkin olduğunu, LTERM kullanıcı adını ve denetim bölgesi bağlantı durumunu gösterir. Komut:

```
/DISPLAY SUBSYS name
```

IMS ile IBM MQ arasındaki bağlantının durumu aşağıdakilerden biri olarak gösterilir:

```
CONNECTED
NOT CONNECTED
CONNECT IN PROGRESS
STOPPED
STOP IN PROGRESS
INVALID SUBSYSTEM NAME= name
SUBSYSTEM name NOT DEFINED BUT RECOVERY OUTSTANDING
```

Her bağımlı bölgedeki iş parçacığı durumu aşağıdakilerden biridir:

```
CONN
CONN, ACTIVE (includes LTERM of user)
```

Bağımlı bölgelerle bağlantıyı kesme

IMSSSM üyesindeki değerleri değiştirmek için. PROCLIB, bağımlı bir bölgenin bağlantısını kesersiniz. Bunu yapmak için aşağıdakileri yapmanız gerekir:

1. IMS komutunu verin:

```
/STOP REGION
```

2. SSM üyesini güncelleyin.

3. IMS komutunu verin:

```
/START REGION
```

IMS bağlantısı kesiliyor

IMS ya da kuyruk yöneticisi sona erdiğinde bağlantı sona erdirilir. Diğer bir seçenek olarak, IMS ana uçbirim işleci bağlantıyı belirttik olarak bozabilir.

IMS ile IBM MQarasındaki bağlantıyı sonlandırmak için aşağıdaki IMS komutunu kullanın:

```
/STOP SUBSYS sysid
```

Komut, genellikle ana uçbirim işlecini (MTO) yayınlayan uçbirime aşağıdaki iletiyi gönderir:

```
DFS058I STOP COMMAND IN PROGRESS
```

IMS komutu:

```
/START SUBSYS sysid
```

yeniden bağlantı kurmak için gereklidir.

Not: Bir IMS tetikleyici izleme programı çalışıyorsa, IMS komut /STOP SUBSYS tamamlanmaz.

IMS tetikleyici izleme programının denetlenmesi

IMS tetikleyici izleme programını durdurmak ve başlatmak için CSQQTRMN hareketini kullanabilirsiniz.

IMS tetikleyici izleme programı (CSQQTRMN hareketi), IMS tetikleyici izleme programının ayarlanması başlıklı konuda açıklanmıştır.

IMS tetikleyici izleme programını denetlemek için aşağıdaki konuya bakın:

- [CSQQTRMN ' nin Başlatılması](#)
- [CSQQTRMN Durduruluyor](#)

CSQQTRMN başlatılıyor

1. İzlemek istediğiniz her başlatma kuyruğu için CSQQTRMN programını çalıştıran toplu iş odaklı bir BMP başlatın.
2. Toplu iş JCL ' yi değiştirerek, aşağıdaki bilgileri içeren bir veri kümesini gösteren CSQQUT1 DDadını ekleyin:

```
QMGRNAME=q_manager_name    Comment: queue manager name
INITQUEUENAME=init_q_name   Comment: initiation queue name
LTERM=lterm                 Comment: LTERM to remove error messages
CONSOLEMESSAGES=YES        Comment: Send error messages to console
```

Burada:

q_manager_name	Kuyruk yöneticisinin adı (bu değer boşsa, CSQQDEFV ' de varsayılan değer kullanılır)
init_q_name	İzlenecek başlatma kuyruğunun adı
lterm	Hata iletilerinin hedefi için IMS LTERM adı (bu boşsa, varsayılan değer MASTER olur).
CONSOLEMESSAGES= YES	Aday IMS LTERM ' ye gönderilen iletilerin de z/OS konsoluna gönderilmesini ister. Bu parametre atlanırsa ya da yanlış yazılırsa, varsayılan değer konsola ileti gönderilmesidir.

3. CSQQUT1 girişinin işlenmesine ilişkin yazılı bir rapor istiyorsanız, CSQQUT2 DD adını ekleyin.

Not:

1. CSQQUT1 veri kümesi LRECL=80 ile tanımlanır. Veri kümesinden diğer DCB bilgileri alınır. CSQQUT2 veri kümesine ilişkin DCB RECFM=VBA ve LRECL=125' dir.
2. Her kayda yalnızca bir anahtar sözcük koyabilirsiniz. Anahtar sözcük değeri, anahtar sözcüğü izleyen ilk boşlukla sınırlanmıştır; bu, açıklamaları ekleyebileceğiniz anlamına gelir. 1. kolondaki yıldız işareti, giriş kaydının tamamının bir açıklama olduğu anlamına gelir.
3. QMGRNAME ya da LTERM anahtar sözcüklerinden birini yanlış yazdıysanız, CSQQTRMN o anahtar sözcük için varsayılan değeri kullanır.
4. Tetikleyici izleme programı BMP işini sunmadan önce altsistemin IMS (/START SUBSYS komutuyla) içinde başlatıldığından emin olun. Başlatılmazsa, tetikleyici izleme işiniz U3042olağandışı bitiş koduyla sona erer.

CSQQTRMN Durduruluyor

CSQQTRMN başlatıldıktan sonra, aşağıdaki olaylardan biri nedeniyle IBM MQ ile IMS arasındaki bağlantı kesilinceye kadar çalışır:

- kuyruk yöneticisi sona eriyor
- IMS Son

ya da bir z/OS STOP **jobname** komutu girilir.

z/OS IMS köprüsünün denetlenmesi

IMS köprüsünü denetlemek için kullanabileceğiniz IMS komutlarını anlamak için bu konuyu kullanın.

IBM MQ-IMS köprüsünü denetlemek için IBM MQ komutu yoktur. Ancak, iletilerin IMS ' e gönderilmesini aşağıdaki şekillerde durdurabilirsiniz:

- Paylaşılmayan kuyruklar için, tüm köprü kuyrukları için ALTER QLOCAL (xxx) GET (DISABLED) komutunu kullanarak.
- Kümelenmiş kuyruklar için, SUSPEND QMGR CLUSTER (xxx) komutunu kullanarak. Bu, yalnızca başka bir kuyruk yöneticisi kümelenmiş köprü kuyruğunu da barındırdığında geçerlidir.
- Kümelenmiş kuyruklar için, SUSPEND QMGR FACILITY (IMSBRIDGE) komutunu kullanarak. IMS'e başka ileti gönderilmez, ancak bekleyen işlemlere ilişkin yanıtlar IMS' den alınır.

IMS ' e yeniden ileti göndermeye başlamak için RESUME QMGR FACILITY (IMSBRIDGE) komutunu verin. Köprünün askıya alınıp alınmadığını görüntülemek için MQSC DISPLAY SYSTEM komutunu da kullanabilirsiniz.

Bu komutların ayrıntıları için [MQSC komutları](#) konusuna bakın.

Daha fazla bilgi için bkz:

- [“IMS köprüsünün başlatılması ve durdurulması” sayfa 569](#)
- [“IMS bağlantılarını denetleme” sayfa 569](#)
- [Köprü kuyruklarının denetlenmesi](#)
- [“IMS köprüsünün yeniden eşitlenmesi” sayfa 570](#)
- [tpipe adlarıyla çalışılması](#)
- [İletilerin IMS içinden silinmesi](#)
- [Veri bağlantılarının silinmesi](#)
- [“IMS İşlem Süre Bitimi” sayfa 572](#)

IMS köprüsünün başlatılması ve durdurulması

OTMA ' yı başlatarak IBM MQ köprüsünü başlatın. IMS komutunu kullanın:

```
/START OTMA
```

ya da IMS sistem parametrelerinde OTMA=YES belirterek otomatik olarak başlatın. OTMA önceden başlatıldıysa, kuyruk yöneticisi başlatma işlemi tamamlandığında köprü otomatik olarak başlatılır. OTMA başlatıldığında bir IBM MQ olay iletisi üretilir.

IMS komutunu kullanın:

```
/STOP OTMA
```

OTMA iletişimini durdurmak için. Bu komut yayınlandığında, bir IBM MQ olay iletisi üretilir.

IMS bağlantılarını denetleme

IMS , IBM MQbağlantısını denetlemek ve izlemek için bu işletmen komutlarını sağlar:

/DEQUEUE TMEMBER *tmember* TPIPE *tpipe*

İletileri Tpipe ' dan kaldırır. Tüm iletileri kaldırmak için PURGE ya da yalnızca ilk iletiyi kaldırmak için PURGE1 değerini belirtin.

/OTAMı GÖRÜNTÜLE

OTMA sunucusu, istemcileri ve istemci durumuna ilişkin özet bilgileri görüntüler.

/DISPLAY TMEMBER *ad*

Bir OTMA istemcisine ilişkin bilgileri görüntüler.

/DISPLAY TRACE TMEMBER *ad*

İzlenmekte olan bilgilere ilişkin bilgileri görüntüler.

/SECURE OTMA

Güvenlik seçeneklerini belirler.

/BAŞLANGIÇ OTAMı

OTMA aracılığıyla iletişimi etkinleştirir.

/START TMEMBER *tmember* TPIPE *tpipe*

Tpipe adını başlatır.

/STOP OTMA

OTMA üzerinden iletişimi durdurur.

/STOP TMEMBER *tmember* TPIPE *tpipe*

Tpipe adını durdurur.

/İZLE

IMS izlemesini denetler.

Bu komutlarla ilgili daha fazla bilgi için kullandığınız IMS düzeyine ilişkin *IMS/ESA Operators Reference* adlı elkitabına bakın.

IMS komut yanıtları, komutun verildiği uçbirime gönderilir. IMS komutlarını verme yetkisi IMS güvenliğine dayalıdır.

Köprü kuyruklarının denetlenmesi

Köprü üzerinden *tmember* XCF üye adıyla kuyruk yöneticisiyle iletişimi durdurmak için aşağıdaki IMS komutunu verin:

```
/STOP TMEMBER tmember TPIPE ALL
```

İletişimi sürdürmek için aşağıdaki IMS komutunu verin:

```
/START TMEMBER tmember TPIPE ALL
```

Bir kuyruğa ilişkin Tpipes, MQ DISPLAY QUEUE komutu kullanılarak görüntülenebilir.

Kuyruk yöneticisiyle tek bir Tpipe üzerinde iletişimi durdurmak için aşağıdaki IMS komutunu verin:

```
/STOP TMEMBER tmember TPIPE tpipe
```

Her etkin köprü kuyruğu için bir ya da iki Tpipe yaratılır; bu nedenle, bu komutun verilmesi IBM MQ kuyruğuyla iletişimi durdurur. İletişimi sürdürmek için aşağıdaki IMS komutunu kullanın:

```
/START TMEMBER tmember TPIPE tpipe
```

Diğer bir seçenek olarak, IBM MQ kuyruğunun özniteliklerini, kuyruğun engellenmesi için değiştirebilirsiniz.

IMS köprüsünün yeniden eşitlenmesi

IMS köprüsü, kuyruk yöneticisi, IMSya da OTMA yeniden başlatıldığında otomatik olarak yeniden başlatılır.

IMS köprüsü tarafından üstlenilen ilk görev, IMSile yeniden eşitlenmektir. Bu, eşitlenmiş her Tpipe ' da IBM MQ ve IMS sıra numaralarının denetlenmesini içerir. Bir IBM MQ - IMS köprü kuyruğundan, kesinleştirme kipi sıfır (kesinleştirme sonrası gönder) kullanılarak IMS ' e kalıcı iletiler gönderildiğinde eşitlenmiş bir Tpipe kullanılır.

Köprü IMSile yeniden eşitlenemezse, CSQ2023E iletisinde IMS algılama kodu döndürülür ve OTMA bağlantısı durdurulur. Köprü tek bir IMS Tpipe ile yeniden eşzamanlayamazsa, CSQ2025E iletisinde IMS

algılama kodu döndürülür ve Tpipe durdurulur. Bir Tpipe soğuk başlatıldıysa, kurtarılabilir sıra numaraları otomatik olarak 1 'e sıfırlanır.

Köprü, bir Tpipe ile yeniden eşzamanlama sırasında eşleşmeyen sıra numaraları saptaysa, CSQ2020E iletisi yayınlanır. IMS Tpipe ile yeniden eşzamanlamayı başlatmak için IBM MQ RESET TPIPE komutunu kullanın. XCF grubu ve üye adını ve Tpipe adını belirtmeniz gerekir; bu bilgiler ileti tarafından sağlanır.

Aşağıdakileri de belirtebilirsiniz:

- IBM MQ tarafından gönderilen iletiler için Tpipe içinde ayarlanacak ve iş ortağının alma sıra numarası olarak ayarlanacak yeni bir kurtarılabilir sıra numarası. Bunu belirtmezseniz, iş ortağının alma sıra numarası geçerli IBM MQ gönderme sıra numarasına ayarlanır.
- IBM MQ tarafından alınan iletiler için Tpipe içinde ayarlanacak ve iş ortağının gönderme sıra numarası olarak ayarlanacak yeni bir kurtarılabilir sıra numarası. Bunu belirtmezseniz, iş ortağının gönderme sıra numarası geçerli IBM MQ alma sıra numarasına ayarlanır.

Tpipe ile ilişkili çözülmemiş bir kurtarma birimi varsa, bu ileti de bildirilir. Kurtarma biriminin kesinleştirilmesini mi, yoksa geri çekilmesini mi istediğinizi belirtmek için IBM MQ RESET TPIPE komutunu kullanın. Kurtarma birimini kesinleştirirseniz, ileti kümesi IMS' e gönderilmiştir ve köprü kuyruğundan silinir. Kurtarma birimini dışarı çıkardıysanız, iletiler daha sonra IMS' e gönderilmek üzere köprü kuyruğuna döndürülür.

Kesinleştirme kipi 1 (gönderme ve kesinleştirme) Tpipe 'lar uyumlulaştırılmaz.

Kesinleştirme kipi 1 hareketleriyle ilgili dikkat edilecek noktalar

IMS içinde, kesinleştirme kipi 1 (CM1) hareketleri, çıkış yanıtlarını eşitleme noktasından önce gönderir.

CM1 hareketi yanıtı gönderemeyebilir; örneğin:

- Yanıtın gönderileceği Tpipe durdurulur
- OTMA durduruldu
- OTMA istemcisi (kuyruk yöneticisi) gitti
- Yanıtın gönderileceği kuyruk ve gönderilmeyen iletiler kuyruğu kullanılmıyor

Bu nedenlerle, IMS uygulamasının iletiyi sözde olarak göndermesi U0119 koduyla sona erer. IMS hareketi ve programı bu durumda durdurulmaz.

Bu nedenler genellikle iletilerin IMS'e gönderilmesini ve IMS' den gelen yanıtların gönderilmesini önler. Aşağıdaki durumlarda U0119 olağandışı sonu oluşabilir:

- İleti IMS içindeyken Tpipe, OTMA ya da kuyruk yöneticisi durdurulur
- IMS gelen iletiye farklı bir Tpipe ile yanıt gönderir ve Tpipe durdurulur
- IMS farklı bir OTMA istemcisine yanıt verdi ve bu istemci kullanılmıyor.

U0119 olağandışı sona erdiğinde, hem IMS 'e gelen ileti, hem de IBM MQ ' e gönderilen yanıt iletileri kaybolur. CMO işleminin çıkışı bu nedenlerden birinden ötürü teslim edilemiyorsa, IMS içindeki Tpipe üzerinde kuyruğa alınır.

Tpipe adlarıyla çalışma

IBM MQ - IMS köprüsünü denetlemek için kullanılan komutların çoğu *tpipe* adını gerektirir. Tpipe adının diğer ayrıntılarını nasıl bulabileceğinizi anlamak için bu konuyu kullanın.

IBM MQ - IMS köprüsünü denetleyen komutların çoğu için *tpipe* adları gerekir. Tpipe adlarını DISPLAY QUEUE komutundan alabilir ve aşağıdaki noktaları not edebilirsiniz:

- bir yerel kuyruk tanımlandığında *tpipe* adları atanır
- Yerel kuyruğa iki *tpipe* adı verilir; biri eşitleme için, diğeri eşitlenmemiş için.
- *tpipe* adları, IMS ile IBM MQ arasında o yerel kuyruğa özgü bir iletişim gerçekleşinceye kadar IMS tarafından bilinmez.

- Bir tpipe 'ın IBM MQ - IMS köprüsü tarafından kullanılabilmesi için, ilişkili kuyruğu, doğru XCF grubu ve üye adı alanlarının tamamlandığı bir Depolama Sınıfına atanmalıdır.

IMS içinden iletileri silme

Tmember/Tpipe durdurulursa, IMS köprüsü aracılığıyla IBM MQ için yönlendirilecek bir ileti silinebilir. XCF üye adı *tmember* olan kuyruk yöneticisine ilişkin bir iletiyi silmek için aşağıdaki IMS komutunu verin:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE1
```

Tpipe 'daki tüm iletileri silmek için aşağıdaki IMS komutunu girin:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE
```

Veri bağlantıları siliniyor

IMS tpipes ögesini kendiniz silemezsiniz. Bunlar IMS tarafından aşağıdaki zamanlarda silinir:

- Eşitlenmiş tpipes, IMS soğuk başlatıldığında silinir.
- IMS yeniden başlatıldığında eşitlenmemiş tpipes silinir.

IMS İşlem Süre Bitimi

Süre bitimi bir işlemle ilişkilendirilir; herhangi bir IBM MQ iletiyle ilişkilendirilmiş bir süre bitimi olabilir. Süre bitim aralığı, MQMD.Expiry alanı kullanılarak uygulamadan IBM MQ' e iletilir. Süre, bir iletinin süresi dolmadan önceki süredir ve saniyenin onda biri olarak ifade edilir. Bir iletinin MQGET işlemini gerçekleştirme girişimi, süresi dolduktan sonra iletinin kuyruktan kaldırılmasına ve süre bitimi işleminin gerçekleştirilmesine neden olur. IBM MQ ağındaki kuyruk yöneticileri arasında bir ileti akışı gerçekleştiğinde süre bitimi azalır. IMS köprüsünden OTMA 'ya bir IMS ileti iletildiğinde, kalan ileti süre bitimi işlemi sona erme zamanı olarak OTMA' ya geçirilir.

Bir hareketin süre bitimi belirtilirse, OTMA giriş hareketlerini IMS içinde üç farklı yerde süre bitimine uğratar:

- XCF ' den giriş ileti alınıyor
- giriş ileti kuyruğa alma süresi
- uygulama GU süresi

GU zamanından sonra süre bitimi gerçekleştirilmez.

EXPTIME işlemi aşağıdaki işlemleri gerçekleştirerek sağlanabilir:

- IMS hareket tanımlaması
- IMS OTMA ileti üstbilgisi
- IMS DFSINSX0 kullanıcı çıkışı
- IMS CREATE ya da UPDATE TRAN komutları

IMS , bir hareketi 0243 ile olağandışı sonlandırarak ve bir ileti yayınlayarak işlemin süresinin dolduğunu gösterir. Yayınlanan ileti, paylaşılmayan kuyruk ortamında DFS555I ya da paylaşılan kuyruk ortamında DFS2224I ' dir.

Advanced Message Security adres alanı, z/OS MODIFY komutunu kullanarak komutları kabul eder.

Yordam

- z/OS üzerinde Advanced Message Security dosyasını değiştirin.

Advanced Message Security (AMS) adres alanı için komut girmek üzere z/OS MODIFY komutunu kullanın.

Örneğin:

```
F qmgrAMSM, cmd
```

Burada *qmgr* , başlatılan görev adının önekidir.

Aşağıdaki çizelge, kabul edilen MODIFY komutlarını açıklar:

Çizelge 29. Advanced Message Security adres alanı MODIFY komutları		
Komut	Seçenek	Açıklama
GÖRÜNTÜLE		Sürüm bilgilerini görüntüle
YENİLE	ANAHTARLIK ilke TÜMÜ	Anahtarlık sertifikalarını, güvenlik ilkelerini ya da her ikisini yenileyin.
SMFAUDIT	BAŞARILI ARIZA TÜMÜ	AMS iletileri başarıyla koruduğunda ya da korumasını kaldırdığında, AMS iletileri koruyamadığında ya da korumasını kaldırdığında ya da her ikisi birden SMF denetiminin gerekip gerekmediğini belirleyin.
SMFTYPE	0-255	AMS iletileri koruduğunda ya da korumasını kaldırdığında oluşturulacak SMF kayıt tipini ayarlayın. SMF denetimini geçersiz kılmak için 0 kayıt tipini belirtin.

Not: Bir seçenek belirtmek için virgülle ayrılmalıdır. Örneğin:

```
F qmgrAMSM, REFRESH KEYRING
F qmgrAMSM, SMFAUDIT ALL
F qmgrAMSM, SMFTYPE 180
```

- z/OS üzerinde Advanced Message Security ürününü yenileyin.

REFRESH komutu çalıştırılarak etkili olan değişiklikler, **REFRESH** komutu tamamlandıktan sonra MQOPEN komutu veren uygulamalar için geçerlidir. Kuyruğu açık olan var olan uygulamalar, uygulama kuyruğu açıldığında seçenekleri kullanmaya devam eder. Yeni değerleri kullanmak için uygulamanın kuyruğu kapatması ve yeniden açması gerekir.

- z/OS üzerinde AMS komutunu başlatın ve durdurun.

Advanced Message Security adres alanını başlatmak ya da durdurmak için bir komut girmenize gerek yoktur. AMS adres alanı, AMS CSQ6SYSP' nin **SPLCAP** parametresiyle etkinleştirildiyse kuyruk yöneticisi başlatıldığında otomatik olarak başlatılır ve kuyruk yöneticisi durdurulduğunda durdurulur.

YönetmeIBM MQ Internet Pass-Thru

Bu bölümde, IBM MQ Internet Pass-Thru (MQIPT) ürününün nasıl yönetileceğini açıklanmaktadır.

mqipt.conf yapılandırma dosyasında [Configuring IBM MQ Internet Pass-Thru](#) başlıklı konuda açıklandığı gibi değişiklik yaparak MQIPT dosyasını yapılandırın. MQIPT'ı yeniden başlatmadan yapılandırma değişikliklerini yürürlüğe sokmak üzere MQIPT ' un yenilenmesi de içinde olmak üzere MQIPTkomutunu yönetmek için **mqiptAdmin** komutunu kullanın. **mqiptAdmin** komutunu kullanarak MQIPT komutunu yönetme hakkında bilgi için bkz. [“MQIPT komutunu komut satırını kullanarak yönetme” sayfa 576.](#)

MQIPT başlatılıyor ve durduruluyor

MQIPT komutunu komut satırından başlatabilir ya da sistem başlatıldığında otomatik olarak başlatabilirsiniz. MQIPT komutunu **mqiptAdmin** komutunu kullanarak durdurabilirsiniz.

MQIPT komutunu komut satırından başlatma

MQIPT bir kuruluş dizinine kurulur; örneğin:

- **Windows** C:\MQIPT on Windows systems, yürütülebilir komut dosyaları C:\MQIPT\bin
- **Linux** **AIX** /opt/mqipt on AIX and Linux systems, yürütülebilir komut dosyaları /opt/mqipt/bin

MQIPT ayrıca, mqipt.conf yapılandırma dosyasını ve çalışırken MQIPT tarafından çıkışa yazılan dosyaları içeren bir ana dizini de kullanır. MQIPT ana dizininin aşağıdaki alt dizinleri, MQIPT ilk kez çağırıldığında otomatik olarak oluşturulur:

- Herhangi bir First Failure Support Technology (FFST) ve izleme dosyasının yazıldığı bir errors dizini
- Bağlantı günlüğünün saklandığı bir logs dizini

MQIPT ' in çalıştırıldığı kullanıcı kimliğinin bu dizinleri yaratma izni olmalıdır ya da diğer bir seçenek olarak, dizinler önceden var olmalıdır ve kullanıcı kimliğinin bu dizinlerde dosya yaratma, okuma ve yazma izni olmalıdır. Ayrıca, bir Java security manager ilkesi kullanıyorsanız, güvenlik ilkesinin bu dizinler için gerekli izinleri vermesi gerekir. Security Manager ilke ayarlarıyla ilgili daha fazla bilgi için bkz. [Java security manager.](#)

Kuruluş dizinini ana dizin olarak kullanabilirsiniz. Bu dizini kullanırsanız, MQIPT ' un çalıştırıldığı kullanıcı kimliğinin uygun izinlere sahip olduğundan ve herhangi bir Security Manager ilkesinin doğru yapılandırıldığından emin olmanız gerekir.

MQIPT' i başlatmak için, MQIPT kuruluş dizininin bin dizininde bulunan **mqipt** komutunu kullanın. Örneğin, aşağıdaki komut, ana dizin olarak C:\mqiptHome dizinini kullanan bir MQIPT örneğini başlatır:

```
mqipt C:\mqiptHome
```

mqipt komutuyla ilgili daha fazla bilgi için bkz. [mqipt \(start MQIPT\).](#)

Başlatılmakta olan MQIPT yönetim ortamına verilecek bir ad belirtmek için **mqipt** komutunu kullanabilirsiniz. MQIPT yönetim ortamının adı, komut kapısı kullanması gerekmeden **mqiptAdmin** komutuyla MQIPT yerel yönetim ortamlarını denetlemek için kullanılır. Bu değiştirge belirtilmezse, MQIPT yönetim ortamının adı olarak MQIPT ana dizininin adı kullanılır.

Konsol iletileri MQIPTdurumunu gösterir. Bir hata oluşursa, bkz. [Sorun Giderme IBM MQ Internet Pass-Thru.](#) Aşağıdaki iletiler, MQIPT başarıyla başlatıldığında çıkışın bir örneğidir:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is C:\mqiptHome
MQCPI021 Password checking has been enabled on the command port
MQCPI144 MQ Advanced capabilities not enabled
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1414 is starting and will forward messages to :
MQCPI034 ...examplehost(1414)
MQCPI035 ...using MQ protocols
MQCPI057 ...trace level 5 enabled
MQCPI078 Route 1414 ready for connection requests
```

MQIPT otomatik olarak başlatılıyor

MQIPT ürününü, sistem başlatıldığında otomatik olarak başlayan bir sistem hizmeti olarak kurabilirsiniz. MQIPT hizmetini kurmak ve kaldırmak için **mciptService** komutunu kullanın.

- **Windows** Windows sistemlerinde **mciptService** komutu, MQIPT ürününü Windows hizmeti olarak kurar.
- **Linux** **AIX** AIX and Linux sistemlerinde **mciptService** komutu, sistem önyüklendiğinde başlayan bir System V init hizmeti olarak MQIPT ürününü kurar. System V init özelliğini desteklemeyen Linux sistemlerinde, hizmet olarak MQIPT ' yi yönetmek için `systemd` gibi başka bir yöntem kullanın.

MQIPT hizmeti başlatıldığında, tüm etkin MQIPT rotaları başlar. Hizmet durdurulduğunda, tüm rotalar anında kapanmaya maruz bırakılıyor.

Sistemde birden çok MQIPT kurulumu olsa bile, bir sisteme yalnızca bir MQIPT hizmeti kurabilirsiniz.

mciptService komutuyla ilgili daha fazla bilgi için bkz. [mciptService \(MQIPT hizmetini yönetme\)](#).

durdurmaMQIPT

mciptAdmin komutunu **-stop** parametresiyle kullanarak MQIPT ' i durdurabilirsiniz.

Örneğin, aşağıdaki komut, yerel olarak **mciptAdmin** komutuyla aynı kullanıcı kimliği altında çalışan `mcipt1` adlı bir MQIPT yönetim ortamını durdurur:

```
mciptAdmin -stop -n ipt1
```

mciptAdmin komutu, aşağıdaki yöntemlerden birini kullanarak yönetmek için etkin MQIPT yönetim ortamına bağlanır:

- komut kapısını kullanmadan yerel bir MQIPT yönetim ortamına bağlanarak.
- bir komut kapısıyla ağ bağlantısı kurarak.

mciptAdmin komutunun bir komut kapısına komut gönderilerek MQIPT komutunu durdurmak için kullanılabilmesi için **RemoteShutDown** özelliği `true` olarak ayarlanarak uzaktan kapatma etkinleştirilmelidir.

MQIPT komutunu **mciptAdmin** komutunu kullanarak yönetme hakkında daha fazla bilgi için bkz. [“MQIPT komutunu komut satırını kullanarak yönetme” sayfa 576](#).

Parola şifreleme anahtarının belirtilmesi

IBM MQ 9.1.5'den MQIPT yapılandırması, varsayılan anahtar dışında bir şifreleme anahtarı kullanılarak şifrelenen parolalar içeriyorsa, MQIPT ' un başlatılırken okuyabileceği bir dosyada parola şifreleme anahtarını sağlamanız gerekir.

Parola şifreleme anahtarı dosyası

MQIPT tarafından depolanmak ve kullanılmak üzere şifrelenen parolalar, sağladığınız bir şifreleme anahtarı kullanılarak şifrelenebilir. Bir şifreleme anahtarı sağlamazsanız, varsayılan şifreleme anahtarı kullanılır. Bir parola şifreleme anahtarı belirtmeniz gerekmez, ancak bunu yapmak daha güvenlidir. Kendi şifreleme anahtarınızı belirtmezseniz, varsayılan şifreleme anahtarı kullanılır.

Bir parola şifreleme anahtarı sağlarsanız, bu anahtarın parolaları ve MQIPT' u şifrelemek için kullanılan **mciptPW** komutuyla erişilebilen bir dosyada saklanması gerekir. Dosyanın içeriğiyle ilgili tek kısıtlama, dosyanın en az bir karakter ve yalnızca bir metin satırı içermesi gerektirir.

Not: Yetkisiz kullanıcıların şifreleme anahtarını okumasını önlemek için parola şifreleme anahtarı dosyasında uygun dosya izinlerinin ayarlandığından emin olmanız gerekir. Yalnızca **mciptPW** komutunu çalıştıran kullanıcı ve MQIPT ' un çalıştırıldığı kullanıcı, parola şifreleme anahtarını okumak için yetkiye gereksinim duyar.

MQIPTörneğine ilişkin tüm depolanan parolaları şifrelemek ve parolaların şifresini çözmek için aynı parola şifreleme anahtarı kullanılır. Bu nedenle, her MQIPT kuruluşu için yalnızca tek bir parola şifreleme anahtarı dosyası gerekir.

Bir MQIPT kuruluşuna ilişkin parola şifreleme anahtarı değiştirilirse, tüm şifrelenmiş parolaların yeni şifreleme anahtarı kullanılarak yeniden şifrelenmesi gerekir.

BaşlatılıyorMQIPT

Parola şifreleme anahtarı dosyasının varsayılan adı `MQIPT_HOME_DIR/mqipt_cred.key` dir; burada `MQIPT_HOME_DIR` , `mqipt.conf` yapılandırma dosyasının saklandığı dizindir. MQIPT ' u otomatik olarak başlatılan bir hizmet olarak çalıştırmayı planlıyorsanız, parola şifreleme anahtarı dosyasını varsayılan adla oluşturmanız gerekir.

Parola şifreleme anahtarı dosyası, varsayılan addan farklı bir adla oluşturulursa, dosyanın adı, dosya başlatıldığında MQIPT ' e sağlanmalıdır. Parola şifreleme anahtarı dosyasının adı, tercih sırasına göre aşağıdaki yöntemlerden herhangi biri kullanılarak belirtilebilir:

1. MQIPT' i başlatmak için kullanılan `mqipt` komutundaki `-sf` parametresi.
2. `MQS_MQIPTCRED_KEYFILE` ortam değişkeni.
3. `com.ibm.mq.ipd_cred.keyfile` Java özelliği.

Parola şifreleme anahtarı dosyası adı sağlanmazsa, dosya varsa, varsayılan dosya adı kullanılır. Varsayılan parola şifreleme anahtarı dosyası yoksa, varsayılan parola şifreleme anahtarı kullanılır.

MQIPT komutunu komut satırını kullanarak yönetme

MQIPTkomutunu yönetmek için komut satırındaki `mqiptAdmin` komutunu kullanabilirsiniz.

Aşağıdaki yönetim işlevlerini gerçekleştirmek için `mqiptAdmin` komutunu kullanabilirsiniz:

- MQIPT' in etkin yerel örneklerini listeleyin.
- Yapılandırma dosyasında değişiklik yaptıktan sonra bir MQIPT örneğini yenileyin.
- Bir MQIPTörneğini durdurun.

`mqiptAdmin` komutu, MQIPT kuruluş dizininin bin alt dizininde bulunur.

`mqiptAdmin` komutu, aşağıdaki yöntemlerden birini kullanarak yönetmek için etkin MQIPT yönetim ortamına bağlanır:

- bir komut kapısıyla ağ bağlantısı kurarak.
- komut kapısını kullanmadan yerel bir MQIPT yönetim ortamına bağlanarak.

`mqiptAdmin` komutu önceki MQIPTsürümleriyle uyumludur, ancak `mqiptAdmin` komutunun sürümünden daha yüksek bir sürüm olan MQIPT sürümlerini yönetmek için bu komutu kullanamazsınız. Farklı MQIPTsürümlerini içeren bir ortamda, `mqiptAdmin` komutunun en son sürümünü kullanmanız gerekir.

`mqiptAdmin` komutunun sözdizimiyle ilgili daha fazla bilgi için bkz. [mqiptAdmin \(denetleme MQIPT\)](#).

Komut kapısı olmayan yerel yönetim

IBM MQ 9.2.0içinden, MQIPT yerel yönetim ortamları bir komut kapısı kullanılmadan yönetilebilir. Yerel denetim, MQIPT komutunu yalnızca denetlemek istediğiniz MQIPT yönetim ortamıyla aynı sistemde çalıştırıldığında `mqiptAdmin` komutunu kullanarak denetlemenizi sağlar.

`mqiptAdmin` ' in komut kapısını kullanmadan yerel bir MQIPT yönetim ortamını denetleme yetkisine sahip olması için, MQIPT yönetim ortamının aynı sistemde ve `mqiptAdmin` ile aynı kullanıcı kimliği altında çalışıyor olması gerekir. Alternatif olarak, AIX and Linuxüzerinde `mqiptAdmin` rootolarak çalıştırılabilir.

Yerel yönetim varsayılan olarak etkindir. Yerel yönetimi devre dışı bırakmak için `LocalAdmin` yapılandırma özelliğini kullanın. `LocalAdmin` özelliği hakkında daha fazla bilgi için bkz. [LocalAdmin](#).

Yerel MQIPTyönetim ortamlarını denetlemek için her yönetim ortamına bir ad vermeniz gerekir. Bir MQIPT yönetim ortamına ad atamak için, MQIPT komutunu **mcipt** komutuyla başlatırken **-n** değiştirgesini kullanın. MQIPTbaşlatılırken bir ad belirtmezseniz, MQIPT yönetim ortamının adı olarak ana dizinin adı kullanılır. Örneğin, aşağıdaki komut MQIPT komutunu başlatır ve yönetim ortamına ipt1 adını atar:

```
mcipt /opt/mcipt1 -n ipt1
```

Yönetim ortamının adı olduktan sonra, **mciptAdmin** komutunda **-n** değiştirgesiyle adı belirterek yönetim ortamını denetleyebilirsiniz. Örneğin, aşağıdaki komut, yerel MQIPT yönetim ortamını ipt1adıyla durdurur:

```
mciptAdmin -stop -n ipt1
```

-list parametresiyle **mciptAdmin** komutunu kullanarak, **mciptAdmin** komutunun bir komut kapısı kullanmadan yönetme yetkisine sahip olduğu tüm yerel etkin MQIPT yönetim ortamlarını listeleyebilirsiniz. Örneğin, aşağıdaki komut, **mciptAdmin** komutunu başlatan kullanıcının yönetme yetkisine sahip olduğu tüm MQIPT yerel etkin yönetim ortamlarını listeler:

```
mciptAdmin -list
```

Komut kapısı kullanan yönetim

IBM MQ 9.2.0'den güvenli olmayan bir komut kapısı ve TLS ile güvenli kılınan bir komut kapısı ile MQIPT ' yi yapılandırabilirsiniz. Bu komut kapılarını, yönetmek istediğiniz MQIPT yönetim ortamıyla aynı sistemde ya da uzak bir sistemde bulunan herhangi bir kullanıcı olarak MQIPT ' i denetlemek için kullanabilirsiniz.

MQIPT ' in önceki sürümleri yalnızca güvenli olmayan komut kapısına verilen yönetim komutlarını kabul etti.

Not: Güvenli olmayan komut kapısına yapılan bağlantılar şifrelenmez, bu nedenle ağ üzerinden MQIPT erişim parolası da dahil olmak üzere güvenli olmayan komut kapısına gönderilen veriler ağ üzerindeki diğer kullanıcılar tarafından görülebilir.

MQIPT ' in **mciptAdmin** komutu tarafından verilen komutlar için bir komut kapısında dinlemesi için, **mcipt.conf** yapılandırma dosyasının genel bölümündeki **CommandPort** ya da **SSLCommandPort** özellikleri için bir değer belirtilmelidir.

MQIPT komut kapılarından birini etkinleştirmeden önce [Diğer güvenlikle ilgili önemli noktaları gözden geçirin](#). Komut kapıları tarafından alınan komutlar için kimlik doğrulamasını etkinleştirmeyi göz önünde bulundurun. Komut kapısı kimlik doğrulamasıyla ilgili daha fazla bilgi için bkz. [“Komut kapısı kimlik doğrulaması” sayfa 581](#).

Bir MQIPT yönetim ortamını komut kapısı kullanarak denetlemek için, MQIPT ' un çalıştığı anasistemin ağ adresini ve komut kapısı numarasını **mciptAdmin** komutunun parametreleri olarak belirtin. Örneğin, **mcipt.example.com** üzerinde çalışan MQIPT yönetim ortamını yenilemek için ve güvenli olmayan komut kapısı 1890 kapısında dinleyecek şekilde yapılandırıldıysa, aşağıdaki komutu verin:

```
mciptAdmin -refresh -r mcipt.example.com:1890
```

Anasistem adını ve kapı numarasını belirtmezseniz, **mciptAdmin** , localhost, 1881 numaralı kapıya bağlanmayı dener.

TLS komut kapısını kullanarak MQIPT yönetimi hakkında daha fazla bilgi için bkz. [“TLS komut kapısını kullanarak MQIPT ' yi yönetme” sayfa 577](#).

TLS komut kapısını kullanarak MQIPT ' yi yönetme

IBM MQ 9.2.0' den MQIPT , **mciptAdmin** komutu tarafından verilen yönetim komutlarını dinlemek için bir TLS komut kapısını kullanacak şekilde yapılandırılabilir. TLS komut kapısının kullanılması, **mciptAdmin** ile MQIPTarasındaki ağda MQIPT erişim parolası gibi hassas verileri korur. TLS komut kapısını yapılandırmak ve TLS komut kapısını kullanarak MQIPT ' i yönetmek için bu yordamı kullanın.

Bu görev hakkında

TLS komut kapısı, PKCS #12 anahtar halkasında ya da PKCS #11 Şifreleme Simgesi Arayüzünü destekleyen şifreleme donanımında saklanan bir sunucu sertifikasıyla yapılandırılmalıdır. Komut kapısı sunucu sertifikası, TLS anlaşması sırasında **mqiptAdmin** komutuna gönderilir. Bu görev, güvenilir bir sertifika kuruluşundan (CA) yeni bir sunucu sertifikası istediğinizi ve sertifikanın bir dosyada size iade edildiğini varsayar. **mqiptAdmin** komutu, sunucu sertifikasını imzalayan CA sertifikasını kullanarak komut kapısı sertifikasını doğrular. CA sertifikası, **mqiptAdmin** komutuyla erişilebilen bir PKCS #12 anahtar halkasında saklanmalıdır.

İstemci sertifikası kimlik doğrulaması TLS komut kapısı tarafından desteklenmiyor. Bir komut kapısına verilen yönetim komutlarına ilişkin kimlik doğrulamasını etkinleştirmek için bkz. "[Komut kapısı kimlik doğrulaması](#)" sayfa 581.

Bu yordam, **mqiptKeycmd** (iKeyman) komut satırı arabirimini (CLI) kullanarak TLS komut kapısını kullanmak için gerekli olan anahtar halkalarının ve sayısal sertifikaların nasıl yönetileceğini açıklar. CLI ' yi **mqiptKeycmd** komutunu kullanarak kullanabilirsiniz. Anahtar halkalarını ve dijital sertifikaları yönetmek için kullanılabilecek diğer komutlar hakkında daha fazla bilgi için bkz. [mqiptKeyman](#) ve [mqiptKeycmd](#).

Yordam

1. MQIPTeşgörünümü için TLS komut kapısını yapılandırmak üzere bu adımları izleyin.

a) TLS komut kapısı tarafından kullanılacak bir PKCS #12 anahtarlık dosyası oluşturun. Anahtar halkası, TLS komut kapısı sunucu sertifikasını saklamak için kullanılır.

CLI kullanarak bir anahtarlık dosyası oluşturmak için aşağıdaki komutu girin:

```
mqiptKeycmd -keydb -create -db filename -pw password -type pkcs12
```

Burada *kütükadı* , yaratılacak anahtarlık kütüğünün adı ve *parola* , anahtarlık parolasıdır.

b) CA imzalı TLS komut kapısı sunucu sertifikası için bir sertifika isteği oluşturun.

iKeyman CLI kullanılarak bir sertifika isteği oluşturmak için aşağıdaki komutu girin:

```
mqiptKeycmd -certreq -create -db filename -pw password  
-label label -size key_size -sig_alg algorithm  
-dn distinguished_name -file certreq_filename -type pkcs12
```

Burada:

-db kütükadı

Anahtarlık dosyası adını belirtir.

-pw parola

Anahtarlık parolasını belirtir.

-label etiket

Sertifika etiketini belirtir.

-size anahtar_büyüklüğü

Anahtar boyutunu belirtir.

-sig_alg algoritma

Girdinin anahtar çiftinin oluşturulması için kullanılan asimetrik imza algoritmasını belirtir.

-dn ayırt edici ad_adi

Çift tırnak içine alınmış X.500 ayırt edici adını belirtir.

-file certreq_dosyaadi

Sertifika isteğine ilişkin dosya adını belirtir.

c) "[1.b](#)" sayfa 578 . adımda yaratılan sertifika isteği dosyasını, imzalanacak CA ' ya gönderin.

d) CA size imzalı sertifikayı gönderdikten sonra, imzalı sertifikayı anahtarlık dosyasına alın.

İmzalı sertifikayı CLI kullanarak anahtar halkasına almak için aşağıdaki komutu girin:

```
mciptKeycmd -cert -receive -file cert_filename -db filename
-pw password -type pkcs12
```

Burada *cert_filename* sertifikayı içeren dosyanın adıdır, *filename* anahtarlık dosyasının adıdır ve *password* anahtarlık parolasıdır.

- e) **mciptPW** komutunu kullanarak anahtarlık parolasını şifreleyin.

Aşağıdaki komutu girin:

```
mciptPW -sf encryption_key_file
```

Burada *encryption_key_file* , MQIPT kuruluşunuz için parola şifreleme anahtarını içeren dosyanın adıdır. MQIPT kuruluşunuz varsayılan parola şifreleme anahtarını kullanıyorsa, **-sf** parametresini belirtmenize gerek yoktur. İstendiğinde şifrelenecek anahtarlık parolasını yazın.

mciptPW komutuyla ilgili daha fazla bilgi için [Anahtarlık parolasının şifrelenmesibaşlıklı konuya](#) bakın.

- f) TLS komut kapısını yapılandırmak için *mcipt.conf* yapılandırma dosyasını düzenleyin ve aşağıdaki özellikleri belirtin:

- i) **SSLCommandPort** özelliğinin değerini TLS komut kapısı numarasına ayarlayın.
- ii) **SSLCommandPortKeyRing** özelliğinin değerini, "1.a" sayfa 578. adımda oluşturulan anahtarlık dosyasının adına ayarlayın.
- iii) **SSLCommandPortKeyRingPW** değerini, "1.e" sayfa 579. adımda **mciptPW** komutuyla dizgi çıkışına ayarlayın.
- iv) **SSLCommandPortSiteLabel** özelliğinin değerini, "1.b" sayfa 578. adımda sertifika isteği oluşturulurken belirtilen TLS komut kapısı sertifikasının etiket adına ayarlayın.
- v) TLS komut kapısına gelen bağlantıları belirli bir ağ arabiriminden gelen bağlantılarla sınırlamak istiyorsanız, **SSLCommandPortListenerAddress** özelliğinin değerini, MQIPT ' un çalıştığı sistemdeki ağ arabirimlerinden birine ait bir ağ adresine ayarlayın. Örneğin, TLS komut kapısına gelen bağlantıları yalnızca yerel makineden gelenlerle sınırlamak için **SSLCommandPortListenerAddress** özelliğinin değerini localhost olarak ayarlayın.

- g) TLS komut kapısını etkinleştirmek için MQIPT komutunu başlatın ya da yenileyin.

MQIPT , yürürlükte olan TLS komut kapısı yapılandırmasını görüntülemek için aşağıdaki gibi konsol iletileri yayınlar:

```
MQCPI155 Listening for control commands on port 1882 on local address * using TLS
MQCPI139 .....secure socket protocols <NULL>
MQCPI031 .....cipher suites <NULL>
MQCPI032 .....key ring file c:\\iptHome\\ssl\\commandport.p12
MQCPI072 .....and certificate label mciptadmin
```

2. **mciptAdmin** komutunun MQIPT'ı yönetmek için kullanıldığı sistemde, **mciptAdmin** ' nin TLS komut kapısına bağlanmasını sağlamak için bu adımları izleyin.

- a) **mciptAdmin** komutu tarafından güvenilirlik deposu olarak kullanılacak bir PKCS #12 anahtar halkası yaratın.

CLI kullanarak bir anahtarlık dosyası oluşturmak için aşağıdaki komutu girin:

```
mciptKeycmd -keydb -create -db filename -pw password -type pkcs12
```

Burada *kütükadı* , yaratılacak anahtarlık kütüğünün adı ve *parola* , anahtarlık parolasıdır.

- b) TLS komut kapısı sertifikasını imzalayan CA sertifikasını, "2.a" sayfa 579. adımda oluşturulan anahtar halkasına aktarın.

CA sertifikasını iKeyman CLI kullanarak içe aktarmak için şu komutu girin:

```
mciptKeycmd -cert -add -db filename -pw password -type pkcs12
-label certlabel -file cert_filename
```

Burada:

Dosya adı

Anahtarlık dosyası adını belirtir

parola

Anahtarlık parolasını belirtir

sertifika etiketi

CA sertifikasına verilecek etiketi belirtir

cert_dosyaadı

CA sertifikasını içeren dosyanın adını belirtir

- c) **mqiPTPW** komutunu kullanarak anahtarlık parolasını şifreleyin.

Aşağıdaki komutu girin:

```
mqiPTPW -sf encryption_key_file
```

Burada *encryption_key_file* , parola şifreleme anahtarını içeren dosyanın adıdır. Parola şifreleme anahtarı dosyası, MQIPT yapılandırmasında parolaları şifrelemek için kullanılan farklı olabilir. **-sf** parametresiyle bir şifreleme anahtarı dosyası belirtmezseniz, varsayılan parola şifreleme anahtarı kullanılır. İstendiğinde şifrelenecek anahtarlık parolasını yazın.

mqiPTPW komutuyla ilgili daha fazla bilgi için [Anahtarlık parolasının şifrelenmesibaşlıklı konuya](#) bakın.

- d) **mqiPTAdmin** komutu tarafından kullanılacak bir özellikler dosyası yaratın ve aşağıdaki özellikleri belirtin:

```
SSLClientCAKeyRing=key_ring_file_name  
SSLClientCAKeyRingPW=key_ring_password  
PasswordProtectionKeyFile=encryption_key_file
```

Burada:

key_ring_dosyası_adi

“2.a” sayfa 579. adımda oluşturulan anahtarlık dosyasının adıdır.

anahtar_ring_parolası

“2.c” sayfa 580. adımda **mqiPTPW** komutu tarafından şifrelenmiş parola çıkımıdır.

şifreleme_anahtar_dosyası

parola şifreleme anahtarını içeren dosyanın adıdır. **PasswordProtectionKeyFile** özelliğini yalnızca “2.c” sayfa 580. adımda anahtarlık parolasını şifrelemek için bir şifreleme anahtarı dosyası kullanıldıysa belirtmeniz gerekir.

- e) Issue the **mqiPTAdmin** command to administer MQIPT, specifying the **-s** parameter to indicate that a TLS connection is required, and the **-p** parameter to specify the name of the properties file that was created in step “2.d” sayfa 580.

Örneğin, TLS komut kapısına bir yenileme komutu göndererek MQIPT örneğini yenilemek için aşağıdaki komutu girin:

```
mqiPTAdmin -refresh -r hostname:port -s -p properties_file
```

mqiPTAdmin komutu, MQIPT bağlantısının TLS ile korunduğunu onaylamak için aşağıdaki gibi bir ileti yayınlar:

```
MQCAI109 The connection to MQIPT is secured with TLSv1.2.
```

Sonraki adım

TLS komut kapısı tarafından alınan komutlar için kimlik doğrulamayı etkinleştirmek üzere “Komut kapısı kimlik doğrulaması” sayfa 581’indeki adımları izleyin.

Komut kapısı kimlik doğrulaması

MQIPT , bir parola kullanılarak güvenli olmayan komut kapısı ve TLS komut kapısı tarafından alınan komutları doğrulamak üzere yapılandırılabilir. Komut kapısı kimlik doğrulamasını etkinleştirmek için bu yordamı kullanın.

Bu görev hakkında

Komut, komut kapısı kimlik doğrulaması etkinleştirilmiş bir MQIPT yönetim ortamının komut kapısına bağlandığında, **mciptAdmin** komutu kullanıcılardan parola girmelerini ister. MQIPT , **mciptAdmin** komutunda girilen parolayı, MQIPT yapılandırmasında belirtilen erişim parolasıyla karşılaştırılarak doğrular.

Komut kapısı kimlik doğrulaması için ayarladığınız özellikler hem TLS komut kapısı hem de güvenli olmayan komut kapısı için geçerlidir.

Yordam

1. MQIPT erişim parolasını **mciptPW** komutunu kullanarak şifreleyin.

Aşağıdaki komutu girin:

```
mciptPW -sf encryption_key_file
```

Burada *encryption_key_file* , MQIPT kuruluşunuz için parola şifreleme anahtarını içeren dosyanın adıdır. MQIPT kuruluşunuz varsayılan parola şifreleme anahtarını kullanıyorsa, **-sf** parametresini belirtmenize gerek yoktur. İstendiğinde şifrelenecek erişim parolasını yazın.

MQIPT yapılandırmasında parolaları şifreleme hakkında daha fazla bilgi için bkz. [Saklanmış parolaları şifreleme](#).

2. *mcipt.conf* yapılandırma dosyasını düzenleyin ve aşağıdaki özellikleri belirtin:

```
AccessPW=encrypted_password  
RemoteCommandAuthentication=auth_setting
```

Burada:

şifrelendi_parola

"1" sayfa 581. adımda **mciptPW** komutu tarafından şifrelenmiş parola çıkımıdır.

yetki_ayarı

kimlik doğrulama gereksinimidir. Bu özellik aşağıdaki değerlerden birine ayarlanırsa, komut kapısı kimlik doğrulaması etkinleştirilir:

isteğe bağlı

Parola gerekli değildir, ancak parola sağlanırsa geçerli olmalıdır. Bu seçenek geçiş sırasında yararlı olabilir; örneğin.

gerekli

Bir komut kapısı tarafından alınan her komutla birlikte geçerli bir parola sağlanmalıdır.

Bu özelliklerle ilgili daha fazla bilgi için bkz. [MQIPT genel özellikler](#).

3. Değişiklikleri yürürlüğe sokmak için MQIPT ' i başlatın ya da yenileyin.

MQIPT , komut kapısı kimlik doğrulamasının etkinleştirilip etkinleştirilmediğini belirten bir ileti gönderir. Örneğin, MQIPT , **mciptAdmin** komutu her çalıştırıldığında geçerli bir parola girilmesini gerektirecek şekilde yapılandırıldıysa, aşağıdaki ileti yayınlanır:

```
MQCPI021 Password checking has been enabled on the command port
```

Yedeklerin yapılması

Düzenli yedekleme yordamlarınızın bir parçası olarak yedeklemeniz gereken birçok MQIPT dosyası vardır.

Aşağıdaki dosyaları düzenli olarak yedekleyin:

- Yapılandırma dosyası, mqipt.conf
- mqipt.conf içinde aşağıdaki özellikler tarafından belirtilen SSL/TLS anahtarlık dosyaları:
 - **SSLClientKeyRing**
 - **SSLClientCAKeyRing**
 - **SSLServerKeyRing**
 - **SSLServerCAKeyRing**
 - **SSLCommandPortKeyRing**
- mqipt.conf içinde aşağıdaki özellikler tarafından belirtilen SSL/TLS anahtar halkası parola dosyaları:
 - **SSLClientKeyRingPW**
 - **SSLClientCAKeyRingPW**
 - **SSLServerKeyRingPW**
 - **SSLServerCAKeyRingPW**
- MQIPT yapılandırması, varsayılan anahtar dışında bir şifreleme anahtarıyla şifrelenen parolalar içeriyorsa, parola şifreleme anahtarı dosyası.
- Bu özellik ayarlandıysa, **SecurityManagerPolicy** tarafından belirtilen ilke dosyası.
- mqipt.conf içinde aşağıdaki özellikler tarafından belirtilen güvenlik çıkış dosyaları ve sertifika çıkış dosyaları:
 - **SecurityExitName**
 - **SSLExitName**
- Denetim amacıyla gerekliyse, MQIPT ana dizininin log alt dizinindeki bağlantı günlüğü dosyaları.

Performans ayarı

Bir iş parçacığı havuzu ve boşta durma zamanasını belirtiminin birleşimini kullanarak her MQIPT rotasının görece performansını ayarlayabilirsiniz.

Bağlantı iş parçacıkları

Her MQIPT rotasına, gelen iletişim isteklerini işleyen koşut zamanlı olarak çalışan iş parçacıklarından oluşan bir çalışma havuzu atanır. Kullanıma hazırlama sırasında, bir iş parçacığı havuzu yaratılır (rotanın MinConnectionThreads özneliğinde belirtilen büyüklükte) ve ilk gelen isteği işlemek için bir iş parçacığı atanır. Bu istek geldiğinde, bir sonraki gelen istek için hazır başka bir iş parçacığı atanır. Tüm iş parçacıkları iş için atandığında, yeni bir iş parçacığı yaratılır, çalışma havuzuna eklenir ve iş için atanır.

Bu şekilde, havuz, iş parçacığı sayısı üst sınırına (**MaxConnectionThreads** içinde belirtilen) ulaşıncaya kadar büyür. Bir etkileşim sona erdiğinde ya da belirtilen boşta durma zamanasını süresi geçtiğinde iş parçacıkları havuza geri bırakılır. Çalışan iş parçacığı sayısı üst sınırına ulaşıldığında, sonraki gelen istek, bir iş parçacığı çalışma havuzuna geri gönderilinceye kadar bekler.

Kullanılabilir iş parçacığı sayısını artırarak isteklerin bekleme süresini kısaltabilirsiniz. Ancak, bu artışı kullanılabilir sistem kaynaklarıyla dengelemeniz gerekir.

Boşta durma zamanasını

Varsayılan olarak, boşta durma nedeniyle çalışan iş parçacıkları sonlandırılmaz. Bir iş parçacığı bir etkileşimde atandığında, olağan bir şekilde kapatılıncaya, rota devre dışı bırakılana ya da MQIPT kapatılıncaya kadar bu etkileşimde atanmış olarak kalır. İsteğe bağlı olarak, **IdleTimeout** özelliğinde bir boşta durma zamanasını aralığı (dakika) belirtebilirsiniz; böylece, belirtilen süre boyunca etkin olmayan iş parçacıkları geri dönüştürülebilir. İş parçacıkları, çalışma havuzuna geri yerleştirilerek kullanılmak üzere geri dönüştürülebilir.

IBM MQ etkinliđi kesintili ise, iř paracıklarının srekli olarak geri dnřtrlmemesi iin sađlıklı iřletim bildirim aralıđını MQIPT zamanařımından daha kk bir deđere ayarlayın.

Özel notlar

Bu belge, ABD'de kullanıma sunulan ürünler ve hizmetler için hazırlanmıştır.

IBM, bu belgede sözü edilen ürün, hizmet ya da özellikleri diğer ülkelerde kullanıma sunmayabilir. Bulduğunuz yerde kullanıma sunulan ürün ve hizmetleri yerel IBM müşteri temsilcisinden ya da çözüm ortağınızdan öğrenebilirsiniz. Bir IBM ürün, program ya da hizmetine gönderme yapılması, açık ya da örtük olarak yalnızca o IBM ürünü, programı ya da hizmetinin kullanılabilirliğini göstermez. Aynı işlevi gören ve IBM'in fikri mülkiyet haklarına zarar vermeyen herhangi bir ürün, program ya da hizmet de kullanılabilir. Ancak, IBM dışı ürün, program ya da hizmetlerle gerçekleştirilen işlemlerin değerlendirilmesi ve doğrulanması kullanıcının sorumluluğundadır.

IBM'in, bu belgedeki konularla ilgili patentleri ya da patent başvuruları olabilir. Bu belgenin size verilmiş olması, patentlerin izinsiz kullanım hakkının da verildiği anlamına gelmez. Lisansla ilgili sorularınızı aşağıdaki adrese yazabilirsiniz:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Çift byte (DBCS) bilgilerle ilgili lisans soruları için, ülkenizdeki IBM'in Fikri Haklar (Intellectual Property) bölümüyle bağlantı kurun ya da sorularınızı aşağıda adrese yazın:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japonya

İzleyen paragraf, bu tür kayıt ve koşulların, yasalarıyla bağdaşmadığı ülkeler ya da bölgeler için geçerli değildir: IBM BU YAYINI, "OLDUĞU GİBİ", HİÇBİR KONUDA AÇIK YA DA ÖRTÜK GARANTİ VERMEKSİZİN SAĞLAMAKTADIR; TİCARİ KULLANIMA UYGUNLUK AÇISINDAN HER TÜRLÜ GARANTİ VE BELİRLİ BİR AMACA UYGUNLUK İDDİASI AÇIKÇA REDDEDİLİR. Bazı ülkeler bazı işlemlerde garantinin açık ya da örtük olarak reddedilmesine izin vermez; dolayısıyla, bu bildirim sizin için geçerli olmayabilir.

Bu yayın teknik yanlışlar ya da yazım hataları içerebilir. Buradaki bilgiler üzerinde düzenli olarak değişiklik yapılmaktadır; söz konusu değişiklikler sonraki basımlara yansıtılacaktır. IBM, önceden bildirimde bulunmaksızın, bu yayında açıklanan ürünler ve/ya da programlar üzerinde iyileştirmeler ve/ya da değişiklikler yapabilir.

Bu belgede IBM dışı Web sitelerine yapılan göndermeler kullanıcıya kolaylık sağlamak içindir ve bu Web sitelerinin onaylanması anlamına gelmez. Bu Web sitelerinin içerdiği malzeme, bu IBM ürününe ilişkin malzemenin bir parçası değildir ve bu tür Web sitelerinin kullanılmasının sorumluluğu size aittir.

IBM'e bilgi ilettiğinizde, IBM bu bilgileri size karşı hiçbir yükümlülük almaksızın uygun gördüğü yöntemlerle kullanabilir ya da dağıtabilir.

(i) Bağımsız olarak yaratılan programlarla, bu program da içinde olmak üzere diğer programlar arasında bilgi değiş tokuşuna ve (ii) değiş tokuş edilen bilginin karşılıklı kullanımına olanak sağlamak amacıyla bu program hakkında bilgi sahibi olmak isteyen lisans sahipleri şu adrese yazabilirler:

IBM Corporation
Yazılım Birlikte Çalışabilirlik Koordinatörü, Bölüm 49XA
3605 Karayolu 52 N
Rochester, MN 55901
U.S.A.

Bu tür bilgiler, ilgili kayıt ve koşullar altında ve bazı durumlarda bedelli olarak edinilebilir.

Bu belgede açıklanan lisanslı program ve bu programla birlikte kullanılacak tüm lisanslı malzeme, IBM tarafından IBM Müşteri Sözleşmesi, IBM Uluslararası Program Lisans Sözleşmesi ya da taraflar arasında yapılan herhangi bir eşdeğer sözleşmenin koşulları kapsamında sağlanır.

Burada belirtilen performans verileri denetimli bir ortamda elde edilmiştir. Bu nedenle, başka işletim ortamlarında çok farklı sonuçlar alınabilir. Bazı ölçümler geliştirilme düzeyindeki sistemlerde yapılmıştır ve bu ölçümlerin genel kullanıma sunulan sistemlerde de aynı olacağı garanti edilemez. Ayrıca, bazı sonuçlar öngörü yöntemiyle elde edilmiş olabilir. Dolayısıyla, gerçek sonuçlar farklı olabilir. Bu belgenin kullanıcıları, kendi ortamları için geçerli verileri kendileri doğrulamalıdır.

IBM dışı ürünlerle ilgili bilgiler, bu ürünleri sağlayan firmalardan, bu firmaların yayın ve belgelerinden ve genel kullanıma açık diğer kaynaklardan alınmıştır. IBM bu ürünleri sinamamıştır ve IBM dışı ürünlerle ilgili performans doğruluğu, uyumluluk gibi iddiaları doğrulayamaz. IBM dışı ürünlerin yeteneklerine ilişkin sorular, bu ürünleri sağlayan firmalara yöneltilmelidir.

IBM'in gelecekteki yönelim ve kararlarına ilişkin tüm bildirimler değişebilir ve herhangi bir duyuruda bulunulmadan bunlardan vazgeçilebilir; bu yönelim ve kararlar yalnızca amaç ve hedefleri gösterir.

Bu belge, günlük iş ortamında kullanılan veri ve raporlara ilişkin örnekler içerir. Örneklerin olabildiğince açıklayıcı olması amacıyla kişi, şirket, marka ve ürün adları belirtilmiş olabilir. Bu adların tümü gerçek dışıdır ve gerçek iş ortamında kullanılan ad ve adreslerle olabilecek herhangi bir benzerlik tümüyle rastlantıdır.

YAYIN HAKKI LİSANSI:

Bu belge, çeşitli işletim platformlarında programlama tekniklerini gösteren, kaynak dilde yazılmış örnek uygulama programları içerir. Bu örnek programları, IBM'e herhangi bir ödemede bulunmadan, örnek programların yazıldığı işletim altyapısına ilişkin uygulama programlama arabirimiyle uyumlu uygulama programlarının geliştirilmesi, kullanılması, pazarlanması ya da dağıtılması amacıyla herhangi bir biçimde kopyalayabilir, değiştirebilir ve dağıtabilirsiniz. Bu örnekler her koşul altında tüm ayrıntılarıyla sinanmamıştır. Dolayısıyla, IBM bu programların güvenilirliği, bakım yapılabilirliği ya da işlevleri konusunda açık ya da örtük güvence veremez.

Bu bilgileri elektronik kopya olarak görüntülediyseniz, fotoğraflar ve renkli resimler görünmeyebilir.

Programlama arabirimi bilgileri

Sağlandıysa, programlama arabirimi bilgileri, bu programla birlikte kullanılmak üzere uygulama yazılımı oluşturmanıza yardımcı olmak amacıyla hazırlanmıştır.

Bu kitapta, müşterinin WebSphere MQ hizmetlerini elde etmek üzere program yazmasına olanak sağlayan amaçlanan programlama arabirimlerine ilişkin bilgiler yer alır.

Ancak, bu bilgiler tanılama, değiştirme ve ayarlama bilgilerini de içerebilir. Tanılama, değiştirme ve ayarlama bilgileri, uygulama yazılımlarınızda hata ayıklamanıza yardımcı olur.

Önemli: Bu tanılama, değiştirme ve ayarlama bilgilerini bir programlama arabirimi olarak kullanmayın; bu bilgiler değişebilir.

Ticari Markalar

IBM, IBM logosu, ibm.com, IBM Corporation 'ın dünya çapında birçok farklı hukuk düzeninde kayıtlı bulunan ticari markalarıdır. IBM ticari markalarının güncel bir listesine Web üzerinde "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml (Telif hakkı ve ticari marka bilgileri) başlıklı konudan ulaşılabilir. Diğer ürün ve hizmet adları IBM'in veya diğer şirketlerin ticari markaları olabilir.

Microsoft ve Windows, Microsoft Corporation firmasının ABD'de ve/ya da diğer ülkelerdeki markalarıdır.

UNIX, The Open Group şirketinin ABD ve diğer ülkelerdeki tescilli ticari markasıdır.

Linux, Linus Torvalds'ın ABD ve/ya da diğer ülkelerdeki tescilli ticari markasıdır.

Bu ürün, Eclipse Project (<https://www.eclipse.org/>) tarafından geliştirilen yazılımları içerir.

Java ve Java tabanlı tüm markalar ve logolar, Oracle firmasının ve/ya da iřtiraklerinin markaları ya da tescilli markalarıdır.



Parça numarası:

(1P) P/N: