

9.2

IBM MQ ' nun denetlenmesi

IBM

Not

Bu bilgileri ve desteklediđi ürünü kullanmadan önce, "[Özel notlar](#)" sayfa 533içindeki bilgileri okuyun.

This edition applies to version 9 release 2 of IBM® MQ and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

İçindekiler

Yönetme.....	7
IBM MQ kuyruk yöneticilerini ve ilişkili kaynakları yönetmenin yolları.....	8
Denetim komutlarını kullanarak IBM MQ uygulamasını yönetme.....	10
MQSC komutlarını kullanan yönetim.....	11
MQSC komutlarının etkileşimli olarak çalıştırılması.....	13
MQSC komutlarının metin dosyalarından çalıştırılması.....	17
z/OSüzerinde MQSC komutları yayınlayabileceğiniz kaynaklar.....	20
Başlatma sırasında bir MQSC komut dosyasından otomatik yapılandırma.....	21
Genel değerler ve özel anlamları olan karakterler.....	22
PCF komutlarını kullanarak IBM MQ yönetimini otomatikleştirme.....	23
IBM MQ Programlanabilir Komut Biçimlerine Giriş.....	23
PCF 'lerin kullanımını kolaylaştırmak için MQAI' nin kullanılması.....	35
REST APIkomutunu kullanarak yönetim.....	70
administrative REST APIile çalışmaya başlama.....	71
REST APIkomutunu kullanarak uzaktan yönetim.....	76
REST API zaman damgaları.....	80
REST API Hata işleme.....	80
REST API Keşif.....	82
REST API ulusal dil desteği.....	84
REST API sürümler.....	85
Web konsolu kullanan yönetim.....	87
Web konsoluyla çalışmaya başlama.....	87
Yeni Web Konsoluna Hızlı Gezin.....	89
Konsol tipleri arasında geçiş yapma.....	109
IBM MQ Explorerkomutunu kullanarak yönetim.....	110
IBM MQ Explorerile yapabildiğiniz.....	111
IBM MQ Explorer' in ayarlanması.....	112
IBM MQ Taskbar uygulamasını kullanma (yalnızcaWindows).....	118
IBM MQ uyarı izleyicisi uygulaması (yalnızca Windows).....	118
Yerel IBM MQ nesneleriyle çalışma.....	119
Kuyruk yöneticileriyle çalışılması.....	119
MQI kanalları durduruluyor.....	129
Yerel kuyruklarla çalışma.....	129
Uzak Kuyruklarla Çalışılması.....	138
Diğer ad kuyruklarıyla çalışılması.....	141
Model kuyruklarıyla çalışma.....	142
Ölü-harfli kuyruklarla çalışma.....	143
Yönetimle ilgili konularla çalışma.....	161
Aboneliklerle çalışma.....	164
Hizmetlerle çalışma.....	168
Tetikleme için nesnelere yönetme.....	175
İki sistem arasında dmpmqmsg yardımcı programının kullanılması.....	177
Uzak IBM MQ nesneleriyle çalışma.....	180
Uzak denetim için kuyruk yöneticilerinin yapılandırılması.....	181
Uzak denetim için komut sunucusunun yönetilmesi.....	185
Uzak kuyruk yöneticisinde MQSC komutları verilmesi.....	186
Kodlanmış karakter kümeleri arasında veri dönüştürme.....	188
YönetmeManaged File Transfer.....	192
MFT aracısının başlatılması.....	193
MFT araçlarını listeleme.....	198
MFT aracısını durdurma.....	198
Yeni bir dosya aktarımı başlatılıyor.....	199

Zamanlanmış bir dosya aktarma işlemi yaratılması.....	202
Bekleyen dosya aktarımlarıyla çalışma.....	203
Dosya aktarımlarının tetiklenmesi.....	204
Devam etmekte olan dosya aktarımlarını izleme.....	205
Aktarma Günlüğü 'nde dosya aktarımlarının durumunu görüntüleme.....	208
MFT kaynaklarının izlenmesi.....	210
Dosya aktarma şablonlarıyla çalışılması.....	239
Ktklerden iletilere veri aktarma.....	242
İletilerden dosyalara veri aktarılması.....	250
Protokol köprüsü.....	255
Connect:Direct köprüsü.....	276
Working with MFT from IBM Integration Bus.....	290
MFT kurtarma ve yeniden başlatma.....	290
Durdurulan aktarımların kurtarılması için bir zamanaşımı ayarlanması.....	291
YönetmeMQ Telemetry.....	296
Linux ve AIXüzerinde telemetri için kuyruk yöneticisi yapılandırılması.....	297
Windowsüzerinde telemetri kuyruk yöneticisi yapılandırılıyor.....	299
MQTT istemcilerine ileti göndermek için dağıtım kuyruğa alma yapılandırılması.....	300
MQTT istemci tanıtıcısı, yetkilendirme ve kimlik doğrulaması.....	303
TLS ' yi kullanarak telemetri kanalı kimlik doğrulaması.....	309
telemetri kanallarında yayın gizliliği.....	310
MQTT Java istemcilerinin ve telemetri kanallarının TLS yapılandırması.....	311
Telemetri kanalı JAAS yapılandırması.....	316
AMQP istemcisinin yönetilmesi.....	318
Viewing IBM MQ objects in use by AMQP clients.....	318
AMQP istemci tanıtıcısı, yetki kimliği ve kimlik doğrulaması.....	319
Kanallarda yayın gizliliği.....	321
AMQP istemcilerinin TLS ile yapılandırılması.....	322
AMQP istemcilerinin kuyruk yöneticisinden çıkarılması.....	323
Çoklu yayını yönetme.....	323
Çok noktaya yayın ile çalışmaya başlama.....	323
IBM MQ Multicast konu topolojisi.....	324
Çok hedefli iletilerin boyutunu denetleme.....	325
Multicast ileti sistemi için veri dönüştürmenin etkinleştirilmesi.....	327
Çok hedefli uygulama izleme.....	328
Çok hedefli ileti güvenilirliği.....	328
Gelişmiş çok noktaya gönderim görevleri.....	329
YönetmeIBM MQ for IBM i.....	332
CL komutlarını kullanarak IBM MQ for IBM i ' in yönetilmesi.....	332
IBM MQ for IBM i yönetimine ilişkin diğer yöntemler.....	346
Work management for IBM i.....	351
Availability, backup, recovery, and restart on IBM i.....	358
IBM MQ for IBM i susturulması.....	401
YönetmeIBM MQ for z/OS.....	404
IBM MQ for z/OS komutlarına komut verilmesi.....	405
IBM MQ for z/OS yardımcı programları.....	413
Çalışırken IBM MQ for z/OS.....	416
Writing programs to administer IBM MQ for z/OS.....	436
z/OS üzerindeki IBM MQ kaynaklarını yönetme.....	448
Recovery and restart on z/OS.....	486
IBM MQ ve IMS.....	508
Operating Advanced Message Security on z/OS.....	520
YönetmeIBM MQ Internet Pass-Thru.....	521
MQIPT başlatılıyor ve durduruluyor.....	522
Komut satırını kullanarak MQIPT uygulamasını yönetme.....	524
Yedeklemeler yapma.....	529
Performans ayarlaması.....	530

Özel notlar.....	533
Programlama arabirimi bilgileri.....	534
Ticari Markalar.....	534

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

Bu görev hakkında

IBM MQ nesnelere yerel olarak ya da uzaktan yönetebilirsiniz:

Yerel denetim

Yerel denetim, yerel sisteminizde tanımladığınız kuyruk yöneticilerindeki denetim görevlerini gerçekleştirmenizi sağlar. Örneğin, TCP/IP uçbirim öykünme programı **telnet** aracılığıyla diğer sistemlere erişebilir ve yönetim yönetimini orada gerçekleştirebilirsiniz. IBM MQ' ta, hiçbir kanal dahil olmadığından, bu iletişim işletim sistemi tarafından yönetildiği için bunu yerel yönetim olarak değerlendirebilirsiniz.

Daha fazla bilgi için bkz [“Yerel IBM MQ nesnelere çalışma” sayfa 119.](#)

Uzaktan yönetim


IBM MQ , uzaktan yönetim aracılığıyla tek bir iletişim noktasından yönetmeyi destekler. Uzak denetim, yerel sisteminizden, başka bir sistemde işlenen komutları vermenizi ve IBM MQ Explorer için de geçerli olduğunu sağlar. Örneğin, uzak bir kuyruk yöneticilerindeki bir kuyruk tanımlamasını değiştirmek için bir uzak komut yayınlatabilirsiniz. Bu sistemde oturum açmanıza gerek yoktur; ancak, tanımlanmış uygun kanallara sahip olmanız. Hedef sistemdeki kuyruk yöneticisi ve komut sunucusu çalışır durumda olmalıdır.

Bazı komutlar bu şekilde yayınlanamaz; özellikle, kuyruk yöneticilerini yaratma ya da başlatma ve komut sunucuları başlatılıyor. Bu tip bir görevi gerçekleştirmek için, uzak sistemde oturum açmanız ve komutları buradan yayınlamanız ya da sizin için komutları yayınlayabilecek bir işlem yapmanız gerekir. Bu kısıtlama, IBM MQ Explorer için de geçerlidir.

Daha fazla bilgi için bkz [“Uzak IBM MQ nesnelere çalışma” sayfa 180.](#)


Kuyruk yöneticilerinizi ve ilgili kaynaklarını IBM MQ' ta yaratmak ve denetlemek için kullanabileceğiniz çeşitli yöntemler vardır. Bu yöntemler, komut satırı arabirimlerini, grafik kullanıcı arabirimlerini ve bir denetim API 'sini içerir.

There are different sets of commands that you can use to administer IBM MQ depending on your platform:

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

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

-   [“The IBM MQ Explorer” sayfa 9](#)
- [“The IBM MQ Console” sayfa 10](#)
-  [“Microsoft Küme Hizmeti \(MSCS\)” sayfa 10](#)

 [IBM MQ for z/OS üzerindeki denetim arabirimleri ve seçeneklerle ilgili bilgi için bkz. “YönetmeIBM MQ for z/OS” sayfa 404.](#)

PCF komutlarını kullanarak hem yerel hem de uzak kuyruk yöneticilerine ilişkin bazı yönetim ve izleme görevlerini otomatikleştirebilirsiniz. Bu komutlar, bazı altyapılarda IBM MQ Administration Interface (MQAI) kullanılarak da basitleştirilebilir. Yönetim görevlerini otomatikleştirme hakkında daha fazla bilgi için bkz. [“PCF komutlarını kullanarak IBM MQ yönetimini otomatikleştirme” sayfa 23.](#)

İlgili kavramlar

[IBM MQ teknik genel bakış](#)

İlgili görevler

[Planlama](#)

[Yapılandırılıyor](#)

İlgili başvurular

[Komut kümeleri karşılaştırması](#)

IBM MQ kuyruk yöneticilerini ve ilişkili kaynakları yönetmenin yolları

IBM MQ kuyruk yöneticilerinin ve ilişkili kaynakların denetlenmesine ilişkin çeşitli seçenekler vardır.

IBM MQ denetim komutları

ALW

Denetim komutlarını kullanarak kuyruk yöneticilerindeki yönetim görevlerini kendileri gerçekleştirebilirsiniz.

IBM MQ for AIX, Linux®, and Windows sistemleri, sistem komut satırına gönderdiğiniz *denetim komutlarını* sağlar.

Denetim komutları, Kuyruk yöneticilerinin çoklu platformlar yaratılması ve yönetilmesi konusunda açıklanmaktadır. Denetim komutlarına ilişkin komut başvurusu için bkz. [IBM MQ Control komutları](#).

IBM MQ Script (MQSC) komutları

Kuyruk yöneticisinin kendisi de içinde olmak üzere kuyruk yöneticisi nesnelerini, kuyrukları, süreç tanımlamalarını, ad listelerini, kanalları, istemci bağlantı kanallarını, dinleyicileri, hizmetleri ve kimlik doğrulama bilgileri nesnelerini yönetmek için MQSC komutlarını kullanın.

You issue MQSC commands to a queue manager by using the **runmqsc** command. Bunu etkileşimli olarak yapabilir, komutları bir klavyeden çıkartabilir ya da standart giriş aygıtını (stdin) bir ASCII metin dosyasından komut dizisi çalıştırmak için yeniden yönlendirebilirsiniz. Her iki durumda da, komutların biçimi aynı olur.

Komutta ayarlanan işaretlere bağlı olarak **runmqsc** komutunu üç kipte çalıştırabilirsiniz:

- *Doğrulama kipi*; burada MQSC komutları yerel bir kuyruk yöneticisinde doğrulanır, ancak çalıştırılmaz.
- *Direct mode*(Doğrudan kip); burada MQSC komutları yerel bir kuyruk yöneticisinde çalıştırılır.
- *Dolaylı kip*; burada MQSC komutları uzak bir kuyruk yöneticisinde çalıştırılır.

MQSC commands are available on all platforms, IBM i ve z/OS dahil . MQSC komutları [Karşılaştırma komut kümelerinin karşılaştırılması](#) içinde özetlenir.

ALW

AIX, Linux, and Windows işletim sisteminde, MQSC ' yi sistem komut satırında verilen tek komutlar olarak kullanabilirsiniz. Daha karmaşık ya da birden çok komut yayınlamak için, MQSC komut satırından çalıştırdığınız bir dosyaya oluşturulabilir. MQSC komutları uzak kuyruk yöneticiye gönderilebilir. Tüm ayrıntılar için bkz. [“MQSC komutlarının metin dosyalarından çalıştırılması” sayfa 17.](#)

IBM i

IBM i sunucusundaki komutları vermek için, bir Komut Dosyası dosyasında komut listesi yaratın ve STRMQMQSC komutunu kullanarak dosyayı çalıştırın.

Notlar: IBM i

1. QTEMP kitaplığının kullanımı sınırlı olduğu için, QTEMP kitaplığını STRMQMQSC ' ye giriş kitaplığı olarak kullanmayın. Komuta giriş dosyası olarak başka bir kitaplık kullanmanız gerekir.

2. IBM üzerinde, bir komut dosyasından yayınlanan komutlara verilen MQSC yanıtları, kuyruğa yollanmış bir dosyada döndürülür.

MQSC komutlarını kullanmaya ilişkin daha fazla bilgi için bkz. [“MQSC komutlarını kullanan yönetim” sayfa 11.](#)

Programlanabilir Komut Biçimleri (PCF ' ler)

Programlanabilir Komut Biçimleri (PCF 'ler), bir program ile ağ içindeki herhangi bir kuyruk yöneticisi (PCF 'leri destekleyen) arasında değiş tokuş edilebilen komut ve yanıt iletilerini tanımlar. You can use PCF commands in a systems management application program for administration of IBM MQ objects: authentication information objects, channels, channel listeners, namelists, process definitions, queue managers, queues, services, and storage classes. Uygulama, yerel kuyruk yöneticisini kullanarak komut ve yanıt bilgilerini herhangi bir kuyruk yöneticisiyle, yerel ya da uzak herhangi bir kuyruk yöneticisiyle iletişim kurmak için ağdaki tek bir noktadan çalıştırılabilir.

PCF 'ler hakkında daha fazla bilgi için bkz. [“IBM MQ Programlanabilir Komut Biçimlerine Giriş” sayfa 23.](#)

Komutlara ve yanıtlara ilişkin PCF 'lerin ve yapıların tanımı için [Programlanabilir komut biçimleri](#) başvurusubaşlıklı konuya bakın.

The administrative REST API

administrative REST API , IBM MQ' u denetlemek için kullanabileceğiniz bir RESTful arabirimi sağlar. administrative REST API'ı kullandığınızda, IBM MQ nesnesini temsil eden bir URL' de bir HTTP yöntemi çağırıyorsunuz. Örneğin, aşağıdaki URL 'deki HTTP yöntemini kullanarak IBM MQ kuruluşlarına ilişkin bilgi isteyebilirsiniz:

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

You can use the administrative REST API with the HTTP/REST implementation of a programming language, or by using tools such as cURL, or a REST client browser add-on.

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

IBM i Denetim Dili (CL)



This language can be used to issue administration commands to IBM MQ for IBM i. Komutlar komut satırında ya da bir CL programı yazılarak yayınlanabilir. Bu komutlar, PCF komutlarına benzer işlevler gerçekleştirir, ancak biçim farklı olur. CL komutları yalnızca sunucular için tasarlanmıştır ve CL yanıtları insan tarafından okunabilir, PCF komutları ise platforma bağımsızdır ve program kullanımı için hem komut, hem de yanıt biçimleri tasarlanır.

IBM i Denetim Dili (CL) ile ilgili tüm ayrıntılar için [IBM MQ for IBM i CL komutları](#) konusuna bakın.

The IBM MQ Explorer



IBM MQ Explorer komutunu kullanarak aşağıdaki işlemleri gerçekleştirebilirsiniz:

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

On Windows and Linux systems, you can start IBM MQ Explorer by using the system menu, the MQExplorer executable file, or the **strmqcfig** command.

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

Daha fazla bilgi için bkz [“IBM MQ Explorerkomutunu kullanarak yönetim” sayfa 110.](#)

You can use IBM MQ Explorer to administer remote queue managers on other platforms including z/OS.

IBM MQ Explorer can be installed either as part of the product installation (see [IBM MQürünün kurulması ve kaldırılması](#)), or from the stand-alone IBM MQ Explorer download available from Fix Central (see [Installing and uninstalling IBM MQ Explorer as a stand-alone application on Linux and Windows](#)).

The IBM MQ Console

You can use the IBM MQ Console to administer IBM MQ from a web browser.

Daha fazla bilgi için bkz [“Web konsolu kullanan yönetim” sayfa 87.](#)

Microsoft Küme Hizmeti (MSCS)

Windows

Microsoft Cluster Service (MSCS), sunucuları bir *küme*ine kurmanızı, daha yüksek veri ve uygulama kullanılabilirliği sağlayarak ve sistemi yönetmeyi kolaylaştırır. MSCS, sunucu ya da uygulama hatalarını otomatik olarak algılayabilir ve bu hataları kurtarabilir.

MSCS algısındaki kümelerin IBM MQ kümeleriyle karıştırılmaması önemlidir. Bu ayrım aşağıdaki gibidir:

IBM MQ Kümeler

Bunlar, bir ya da daha çok bilgisayarda iki ya da daha çok kuyruk yöneticisi gruplarıdır; otomatik ara bağlantı sağlar ve yük dengeleme ve yedeklilik için kuyrukların bunlar arasında paylaşılmasına olanak tanır.

MSCS kümeleri

Bunlar, birbirine bağlı ve yapılandırılmış olan bilgisayar gruplarıdır. Böyle bir şekilde, bir hata oluşursa, MSCS bir *yedek sisteme geçiş*ilemini gerçekleştirir, uygulamaların durum verilerini bozuk bilgisayardan kümedeki başka bir bilgisayara aktarır ve bu işlem burada yeniden başlatılabilir.

[Microsoft Cluster Service \(MSCS\)](#) ' nin desteklenmesi, IBM MQ for Windows sisteminizin nasıl MSCS kullanacak biçimde yapılandırılmasına ilişkin ayrıntılı bilgi sağlar.

ALW Denetim komutlarını kullanarak IBM MQ uygulamasını yönetme

Denetim komutları, AIX, Linux, and Windowsüzerinde bir dizi IBM MQ denetim görevi gerçekleştirmenin bir yolunu sağlar.

Denetim komutlarını vermek istiyorsanız, kullanıcı kimliğinizin çoğu denetim komutu için mqm grubunun bir üyesi olması gerekir. Bu konuda daha fazla bilgi için bkz. [AIX, Linux, and Windowsüzerinde IBM MQ yönetimi yetkisi](#). Buna ek olarak, ortama özgü bilgileri not edin. Platformun veya platformların, işletiminizin kullandığı platformlar için.

Kuyruk yöneticisiyle çalışan denetim komutlarını kullanırken, çalışmakta olduğunuz kuyruk yöneticisiyle ilişkilendirilmiş kuruluştan bu komutu kullanmanız gerekir.

CHCKLOCL (REQUIRD) ile bağlantı kimlik doğrulamasını kullanmak üzere yapılandırılmış bir kuyruk yöneticisinde çalışan denetim komutları kullanılırken, bağlantı kurulamaması da gözlemlenir.

- Denetim komutu buna izin veriyorsa, bir kullanıcı kimliği ve parola belirtin.
- Varsa, denetim komutlarının MQSC eşdeğerlerini kullanın.

- Kuyruk yöneticisini -ns seçeneğini kullanarak başlatın; bağlantı kuramayan denetim komutları çalıştırılmalıdır.

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

Using control commands on Windows systems

Windows

IBM MQ for Windows' ta, bir komut isteminde denetim komutlarını giriniz.

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

Örneğin, komutta:

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

- Komut adı büyük harfli ya da küçük harfle ya da iki harfin bir karışımında girilebilir. Bunların tümü geçerlidir: `crtmqm`, `CRTMQM` ve `CRTmqm`.
- İşaret, -u, -U, /uya da /Uolarak girilebilir.
- `SYSTEM.DEAD.LETTER.QUEUE` ve `jupiter.queue.manager`, tam olarak gösterildiği şekilde girilmelidir.

Using control commands on AIX and Linux systems

Linux

AIX

IBM MQ for AIX or Linux sistemlerinde, denetim komutlarını bir kabuk penceresinde girmenizi sağlar.

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

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

- Komut adı `crtmqm` olmalı, `CRTMQM` olmalıdır.
- İşaret, -U değil, -u olmalıdır.
- Ölü-harflik kuyruğu `SYSTEM.DEAD.LETTER.QUEUE` olarak adlandırılır.
- Bağımsız değişken, `JUPITER.queue.manager`' den farklı olan `jupiter.queue.manager` olarak belirtilir.

Komutları, örneklerde gördüğünüz gibi, tam olarak yazmak için dikkatli olun.

İlgili başvurular

[IBM MQ Control komutları başvurusu](#)

MQSC komutlarını kullanan yönetim

Kuyruk yöneticisi nesnelerini, kuyrukları, süreç tanımlamalarını, kanalları, istemci bağlantı kanallarını, dinleyicileri, hizmetleri, ad listelerini, kümeleri ve kimlik doğrulama bilgileri nesnelerini de içeren kuyruk yöneticisi nesnelerini yönetmek için MQSC komutlarını kullanabilirsiniz. MQSC komutları tüm altyapılarda kullanılabilir.

Bu görev hakkında

MQSC komutlarını verme şekliniz platformunuza bağlıdır:

- **ALW** AIX, Linux, and Windows' ta, **runmqsc** komutunu kullanarak bir kuyruk yöneticisine MQSC komutları yayınlayın. **runmqsc** komutunu aşağıdaki şekillerde çalıştırabilirsiniz:

- Etkileşimli olarak, bir klavyeden komut yayınlayın. Bkz. [“MQSC komutlarının etkileşimli olarak çalıştırılması” sayfa 13.](#)
- ASCII metin kütüğünden. Bkz. [“MQSC komutlarının metin dosyalarından çalıştırılması” sayfa 17.](#)
- Uzak kuyruk yöneticisinde. Bkz. [“Uzak IBM MQ nesnelileriyle çalışma” sayfa 180.](#)
- **z/OS** z/OS üzerinde, komutta bağlı olarak bir dizi kaynaktan MQSC komutları yayınlanabilir. Daha fazla bilgi için bkz [“z/OS üzerinde MQSC komutları yayınlayabileceğiniz kaynaklar” sayfa 20.](#)

MQSC komutları [MQSC komutları](#) kısmında ayrıntılı olarak açıklanmıştır.

Yordam

- Her komut bir birincil değiştirgeyle (fil) başlar ve bu, onu izleyen ikincil bir parametre (bir isim) izler. Bu, en çok komutta yer alan bir nesnenin (parantez içinde) adını ya da soysal adını (parantez içinde) izler. Bunun ardından, parametreler genellikle herhangi bir sırada olabilir; bir parametrenin karşılık gelen bir değeri varsa, değer, ilgili olduğu parametrenin hemen ardından gerçekleşmelidir.

Not: **z/OS** z/OS üzerinde, ikincil parametre ikinci olmak zorunda değildir.

- Anahtar sözcükler, parantezler ve değerler, herhangi bir sayıda boşluk ve virgül ile ayrılabilir. Sözdizimi şemalarında gösterilen bir virgül her zaman bir ya da daha çok boşlukla değiştirilebilir. There must be at least one blank immediately preceding each parameter (after the primary parameter) 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 boşluk karakterlere önem vermektedir.

- Ek virgüller, boşluklara izin verilen ya da boşlukmuş gibi işlem gören herhangi bir yerde görünebilir (elbette, tırnak işaretleri içinde bulunan dizeler içinde bulunmuyorsa).
 - Yinelenen parametrelere izin verilmez. REPLACE NOREPLACE gibi "NO" sürümüne sahip bir değiştirgenin yinelenmesine de izin verilmez.
 - Boşlukları, küçük harf karakterlerini ya da özel karakterleri içeren dizgiler tek tırnak işareti içine alınmalıdır; ancak, aşağıdaki koşullardan biri doğru değilse:
 - Özel karakterler, aşağıdaki karakterlerden bir ya da daha fazladır:
 - Nokta (.)
 - Eğik Çizgi (/)
 - Alt çizgi (_)
 - Yüzde işareti (%)
 - **z/OS** Komut, IBM MQ for z/OS işlemleri ve denetim panolarından verilir.
 - Dizilim, yıldız işaretiyle biten genel bir değerdir. (IBM i üzerinde bunlar tek tırnak işareti içine alınmış olmalıdır)
 - Dizgi tek bir yıldız işaretidir; örneğin, TRACE (*) (IBM i üzerinde bunların tek tırnak işareti içine alınması gerekir)
 - Dizgi, iki nokta üst üste (CLASS) içeren bir aralık belirtimidir (CLASS (01:03))
- Dizginin kendisinde tek tırnak işareti varsa, tek tırnak işareti iki tek tırnak işareti ile gösterilir. Tırnak işaretleri içinde bulunmayan küçük harfli karakterler büyük harfe katlanır.

- **Multi** Multiplatforms üzerinde, herhangi bir karakter içermeyen (yani, arasında boşluk içermeyen iki tek tırnak işareti) tek tırnak işareti içine alınmış boş bir alan olarak yorumlanır, yani, (') ile aynı şekilde yorumlanan bir boşluk olur. Boş alan içermeyen iki tek tırnak işareti sıfır uzunluklu bir dizgi olarak

yorumlandığında, kullanılmakta olan öznitelik aşağıdaki özniteliklerden biriye, bu durum için kural dışı durum.

- TOPICSTR
- SUB
- USERDATA
- SELECTOR

• z/OS

z/OS'da, tek tırnak işareti içine alınmış boş bir alan istiyorsanız, bu alanı (' ') (' ') olarak girmeniz gerekir. Karakter içermeyen bir dizgi (") girilirken () aynı olur.

- SELECTOR, alt kullanıcı verileri gibi, MQCHARV tiplerini temel alan dizgi özniteliklerindeki sondaki boşluklar, anlamlı olarak işlenir. Bu, 'abc ' ' ' un 'abc ' ile eşit olmadığı anlamına gelir.
- Arasında önemli bir bilgi olmayan bir kapama ayracı ve ardından bir açma ayracı, özellikle dikkat edilenler dışında geçerli değildir. Örneğin, aşağıdaki dizgi geçerli değildir:

```
NAME ( )
```

- Anahtar sözcükler büyük ve küçük harfe duyarlı değildir: AltER, alter, ve ALTER, tüm kabul edilebilir. Tırnak işaretleri içinde yer almayan her şey büyük harfe katlanır.
- Eşanlamlılar bazı parametreler için tanımlanır. Örneğin, DEF her zaman DEFE eşanlamlısıdır, bu nedenle DEF QLOCAL geçerlidir. Eşanlamlılar, yalnızca minimum dizgiler değil; DEFI, DEFE için geçerli bir eşanlamlı değil.

Not: DELETE değiştirgesinin eşanlamlısı yok. Bu, DEF eşanlamlısı olan DEF ' yi kullanırken nesnelerin yanlışlıkla silinmesini önlemektedir.

- MQSC komutları, bazı özel karakterleri belirli anlamlara sahip olacak şekilde kullanır. Bu özel karakterlerle ve bunların nasıl kullanılacağı hakkında daha fazla bilgi için bkz. [“Genel değerler ve özel anlamları olan karakterler” sayfa 22.](#)

İlgili görevler

[MQSC komutlarıyla ilgili sorunların çözülmesi](#)

İlgili başvurular

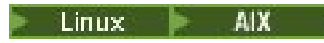
[runmqsc \(MQSC komutlarını çalıştırır\)](#)

MQSC komutlarının etkileşimli olarak çalıştırılması

Bir komut penceresi ya da kabuk kullanarak MQSC komutlarını etkileşimli olarak kullanabilirsiniz.

Başlamadan önce

runmqsc komutunu çalıştırdığınızda görüntülenen bilgi istemini ayarlayabilirsiniz. Bkz. [“MQSC komut istemi ayarlanıyor” sayfa 15.](#)

 UNIX and Linux altyapılarında MQSC komutlarını etkileşimli olarak çalıştırdığınızda, **runmqsc** komut satırı komut çağırma, komut tamamlama ve Emacs komut tuşlarını destekler. Bkz. [“AIX ve Linux üzerinde runmqsc komutu” sayfa 16.](#)

Bu görev hakkında

Yordam

1. MQSC komutlarını etkileşimli olarak kullanmak için 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.

- 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.

- Aşağıdaki komutu girerek MQSC komutlarıyla çalışmayı durdurun:

```
end
```

Diğer bir seçenek olarak, işletim sisteminiz için 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 standart çıkış aygıtına gönderilir. Komutu doğru girmediyseniz, doğru sözdizimini bulmak için komuta ilişkin başvuru bilgilerine bakın. Bkz. [MQSC komutları](#).

İlgili görevler

[“MQSC komutlarının metin dosyalarından çalıştırılması” sayfa 17](#)

MQSC komutlarının etkileşimli olarak çalıştırılması hızlı sınamalar için uygundur; ancak, çok uzun komutlarınız varsa ya da belirli bir komut sırasını arka arkaya kullanıyorsanız, `stdin` ' i bir metin dosyasından yeniden yönlendirebilirsiniz. Çıkışı bir dosyaya yeniden yönlendirebilirsiniz.

İlgili başvurular

[mqsc](#)

MQSC komut istemi ayarlanıyor





MQPROMPT ortam değişkenini kullanarak, MQSC komut istemini istediğiniz bir bilgi istemine ayarlayabilirsiniz.

Yordam

- **MQPROMPT** ortam değişkenini seçtiğiniz bilgi istemine ayarlayın.
runmqsc konsolu etkileşimli olarak çalıştırıldığında ve giriş bir dosyadan ya da standart giriş aygıtından (stdin) **runmqsc** ' e yeniden yönlendirildiğinde bilgi istemi eklenir.

Komut istemine düz metin ekleyebilir ve ortam değişkenlerini, +VARNAME+ gösterimini IBM MQ hizmet nesnesi tanımlamalarıyla aynı şekilde kullanarak da ekleyebilirsiniz. Daha fazla bilgi için bkz “[Hizmet tanımlarında değiştirilebilir eklemeler](#)” sayfa 172.

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

Değiştirilebilir ekleme	Açıklama
MQ_ANASISTEM_ADI	Sistemin anasistem adı
MQ_FILE_SEP	Platforma özgü dosya ayırıcısı: –  AIX and Linux sistemlerinde MQ_FILE_SEP '/' dir. –  Windows sistemlerinde MQ_FILE_SEP ' in yeri şudur: \
MQ_PATH_SEP	Platforma özgü yol ayırıcısı: –  AIX and Linux sistemlerinde MQ_PATH_SEP: :. –  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 kuruluşu ve anasistemiyle ilişkilidir.
- MQPROMPT, ekler genişletildiğinde en çok 256 karakterle sınırlıdır. Bu değer üzerindeki MQPROMPT genişletmeleri, tüm MQPROMPT dizgisinin genişletmeler olmadan kesilmesiyle sonuçlanır.

Örneğin, bilgi istemini MQSColarak ayarlamak için aşağıdaki komutlardan birini girin:

- 

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

- 

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

Örnek

Aşağıdaki örnek, AIX sisteminde **MQPROMPT** değişkeninin ayarlanmasını göstermektedir. Bilgi istemi, ilişkili sistem ortam değişkeninden alınan bir kullanıcı adını, kuyruk yöneticisi adını ve MQ değiştirilebilir eklerinden alınan IBM MQ anasistem adını 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 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

AIX ve Linux üzerinde runmqsc komutu

AIX ve Linux üzerindeki **runmqsc** komut satırı, komut geri çağırma, komut tamamlama ve Emacs komut tuşlarını destekler.

Aşağıdaki komut satırı düzenleyicisi işlevleri kullanılabilir:

- 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** komut satırı düzenleyici işlevlerine sahip değildir ve **runmqsc** komut satırı başlatıldığında bir ileti görüntülenir. Kurulacak lanetler kitaplığının adı UNIX platformuna bağlıdır:

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

AIX üzerinde ncurses ya da curses ürününü kurma

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 tanımlarının uyarlanması

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 tuşlarının devre dışı bırakılması

Bir ortam değişkeni ayarlayarak komutu geri çağırma, komut tamamlama ve Emacs komut tuşlarını geçersiz kılabilirsiniz. `MQ_OVERRIDE_LIBEDIT_LOAD` ortam değişkenini TRUE olarak ayarlayın.

`runmqsc` aşağıdaki bilgi iletisini gösterdiğinde bu ortam değişkeni geçici çözüm olarak kullanılabilir:

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

MQSC komutlarının metin dosyalarından çalıştırılması

MQSC komutlarının etkileşimli olarak çalıştırılması hızlı sınamalar için uygundur; ancak, çok uzun komutlarınız varsa ya da belirli bir komut sırasını arka arkaya kullanıyorsanız, `stdın` 'i bir metin dosyasından yeniden yönlendirebilirsiniz. Çıkışı bir dosyaya yeniden yönlendirebilirsiniz.

Bu görev hakkında

`runmqsc` komutuna ilişkin giriş, `stdın` olarak da adlandırılan *standart giriş aygıtı*ndan alınır. `stdın`, sisteme girişin alındığı aygıttır. Bu genellikle klavyedir, ancak girişin bir dize kapıdan ya da disk dosyasından geleceğini belirtebilirsiniz.

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

Aşağıdaki komutu kullanırken MQSC komutlarını bir komut dosyasında oluşturmak isteyebilirsiniz:

- ▶ **z/OS** CSQINP1, CSQINP2 ve CSQINPX kullanıma hazırlama veri kümeleri ya da z/OS üzerinde CSQUTIL toplu iş yardımcı programı.
- ▶ **IBM i** IBM üzerinde **STRMQM** komutu.
- ▶ **ALW** AIX, Linux, and Windows üzerinde **runmqsc** komutu.

MQPROMPT ortam değişkenini kullanarak, MQSC komut istemini istediğiniz bir bilgi istemine ayarlayabilirsiniz. Daha fazla bilgi için bkz "[MQSC komut istemi ayarlanıyor](#)" sayfa 15.

Yordam

1. Çalıştırmak istediğiniz MQSC komutlarını içeren bir metin dosyası yaratın.

- 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 eksi işaretiyle devam edemez. Komutun tamamı tek bir Escape komutunda bulunmalıdır. PCF komutlarına ilişkin bilgi için bkz. ["IBM MQ Programlanabilir Komut Biçimlerine Giriş" sayfa 23.](#)
- Multiplatformsve z/OS sistemlerinde,CSQUTIL toplu iş yardımcı programı tarafından 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 stdingeçirilerek yapılabilir.
- **Windows** Windowsüzerinde, 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.
- MQSC komutu sözdizimi hakkında bilgi için bkz. [MQSC komutları.](#)
- Metin dosyanızı yaratmanıza yardımcı olması için örnek MQSC komut dosyalarını kullanabilirsiniz:

amqscos0.tst

Örnek programlar tarafından kullanılan nesnelere tanımları.

amqscic0.tst

CICS hareketlerine ilişkin kuyrukların tanımları.

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

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.

2. Komutları çalıştırmadan yerel bir kuyruk yöneticisinde komut sözdiziminin doğru olduğunu doğrulayın. **runmqsc** komutundaki **-v** işaretini kullanın.

V 9.2.0

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

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

- 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 QmgrName < myprog.in
```

Döndürülen rapor, [Şekil 2 sayfa 20](#) içinde gösterilen rapora benzer.

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

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

V 9.2.0

- 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 19](#) , `myprog.in` MQSC komut kütüğünden bir alma işlemini gösterir ve [Şekil 2 sayfa 20](#) , `results.out` kütüğündeki çıkışın ilgili çıkarmasını gösterir.

Örnek

MQSC komutları, insan tarafından okunabilir biçimde, yani ASCII metinle yazılır. Aşağıdaki örnek, **DEFINE QLOCAL** MQSC komutunu gösteren bir MQSC komut dosyasından 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

İlgili görevler

“MQSC komut istemi ayarlanıyor” sayfa 15

MQPROMPT ortam değişkenini kullanarak, MQSC komut istemini istediğiniz bir bilgi istemine ayarlayabilirsiniz.

“MQSC komutlarının etkileşimli olarak çalıştırılması” sayfa 13

Bir komut penceresi ya da kabuk kullanarak MQSC komutlarını etkileşimli olarak kullanabilirsiniz.

İlgili başvurular

[mqsc](#)

z/OS

z/OS üzerinde MQSC komutları yayınlatabileceğiniz kaynaklar

MQSC komutları, komutta bağlı olarak çeşitli kaynaklardan yayınlanabilir.

Komutlar aşağıdaki kaynaklardan yayınlanabilir:

- z/OS konsolu ya da eşdeğeri
- Kullanıma hazırlama giriş verileri kümeleri CSQINP1, CSQINP2, CSQINPT ve CSQINPX
- CSQUTIL toplu iş yardımcı programı
- Uygun şekilde yetkilendirilmiş uygulamalar, komutları SYSTEM.COMMAND.INPUT kuyruğu

z/OS

Daha fazla ayrıntı için bkz. “IBM MQ for z/OS komutlarına komut verilmesi” sayfa 405,

Ancak, tüm bu kaynaklardan tüm komutlar verilemez. Komutlar şu şekilde yayınlanabildiğine göre sınıflandırılabilir:

1

CSQINP1

2

CSQINP2

C

z/OS konsolu

R

Komut sunucusu ve komut kuyruğu, CSQUTIL, CSQINPT, CSQINPX ya da uygulamalar aracılığıyla

MQSC komutlarındaki komut tanımları içinde, bu kaynaklar her komut tanımlamasında 1, 2, C ve R karakterleri kullanılmasıyla tanımlanır.

Başlatma sırasında bir MQSC komut dosyasından otomatik yapılandırma

IBM MQ 9.2.0' tan, kuyruk yöneticinizi, her kuyruk yöneticisi başlatması sırasında bir MQSC komut dosyasının içeriğini ya da MQSC komut dosyalarını otomatik olarak uygulamak için kuyruk yöneticinizi yapılandırabilirsiniz.

Bu işlevselliği, değiştirilebilecek bir yapılandırmaya sahip olmak için kullanabilir ve bir sonraki kuyruk yöneticisi yeniden başlatıldığında otomatik olarak yeniden oynatmanızı sağlar. Örnek olarak, komut dosyası ya da komut dosyaları takılı bir sürücüde bulunuyorsa, en son sürümün her kuyruk yöneticisine uygulandığında en son sürümün uygulandığı merkezi bir konfigürasyona sahip olmak mümkündür.

Bunun yararlı olacağı belirli bir senaryo, tek bir kümenin, tek bir yapılandırma kümesi içererek, kümedeki tüm kuyruk yöneticilerinde aynı tanımlamaları içerdiğinden emin olmak için kullanılır. Bunun bir örneği için bkz. [Yeni bir tek tip küme oluşturma](#).

Başlamadan önce

Aşağıdakileri kullanabilirsiniz:

1. Tek bir komut dosyası oluşturun ve MQSC komutlarını kullanarak bir metin dosyası oluşturun.
2. Bir MQSC komut dosyası kümesi:
 - Yapılandırmaların var olacağı bir dizini tanımlamak ve
 - Bu dizinde, her biri .mqscuzantısına sahip dosyalar oluşturun; örneğin, queues .mqsc.

Bu komut dosyasının her kuyruk yöneticisi başlangıcındaki yeniden uygulanmış olması göz önüne alındığında, komutların yeniden yürütülebilmesi önemlidir. Örneğin, bir **DEFINE** komutu **REPLACE** dizgisini içermelidir; tersi durumda, nesne zaten var olduğu için, komut ikinci kuyruk yöneticisi başlangıcındaki bir hata olarak görünür.

Bir MQSC komut dosyasında, başında * öneki olan her satırın bir açıklama olarak kabul edildiğine dikkat edin.

MQSC komut kütüklerinin otomatik olarak yapılandırılabilmesini sağlama

Yeni bir kuyruk yöneticisi yapılandırmak için, **-ic** işaretini **crtmqm** komutunu kullanarak ve belirli bir dosyayı ya da dizini gösterebilirsiniz. The supplied value is stored in the qm . ini file under the AutoConfig stanza, as attribute **MQSCConfig**.

Geçerli bir dosya ya da dizine işaret ederek, AutoConfig stanza özniteliği **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?

During queue manager startup, the configuration identified by the AutoConfig stanza attribute **MQSCConfig** is passed through **runmqsc** validation, to ensure valid syntax, and then stored in the queue manager data tree into the **autocfg** directory as a single file **cached.mqsc**.

Bir dizinden birden çok dosya işlendiğinde, bunlar alfabetik sırayla işlenir ve bir MQSC sonu ya da çıkış komutu içeriyorsa, o dosyanın içeriğinin geri kalanı atlanır.

Kuyruk yöneticisinin ilk başlangıcı sırasında, kütüğü ya da dizini okuyamama ya da MQSC sözdizimine sahip bir kütük geçerli olmayan bir kütüğü, kuyruk yöneticisinin hem konsola, hem de kuyruk yöneticisi hata günlüğüne uygun bir hata iletilmesiyle başlamasını önler.

Sonraki yeniden başlatmalarda, dosya ya da dizin okunamazsa ya da geçersiz MQSC sözdizimi içeriyorsa, önceden önbelleğe alınmış dosya kullanılır ve kuyruk yöneticisinin hata günlüğüne yazılan bir ileti bu dosyayı vurgular.

V 9.2.2 At the point the contents of the cached .mqsc are applied to the queue manager, when all of the MQSC commands have been applied, the queue manager is enabled for applications to connect. Uygulanmakta olan yapılandırmanın **runmqsc** günlüğü, kuyruk yöneticisinin Errors dizininde, autocfgmqsc .LOGadlı bir dosya olarak saklanır.

Ayrıca, başarıyla tamamlanmadığı herhangi bir MQSC komutu kuyruk yöneticisi hata günlüğüne kaydedilir ve komutun neden başarısız olduğunu saptamış olur.

Genel değerler ve özel anlamları olan karakterler


Bazı karakterler, örneğin, ters eğik çizgi (\) ve çift tırnak işareti (") MQSC komutlarıyla kullanıldığında karakterlerin özel anlamları vardır. Bazı özel karakterler, parametrelerle genel değerlere sahip olabilir, ancak doğru bir şekilde belirtilmelidir.

Precede backslash (\) and double quote (") characters with a \, that is, enter \\ or \" if you want \ or " in your text.

Bir değiştirgenin genel bir değeri varsa, bu değer bir yıldız işareti (*) ile sona erdirilir; örneğin, ABC*. Sosyal bir değer, birlikte başlayan tüm değerler anlamına gelir; bu nedenle ABC*, ABC ile başlayan tüm değerleri belirtir. Değerde tırnak işaretleri gerektiren karakterler kullanılırsa, yıldız işareti tırnak işaretlerinin içine yerleştirilmelidir. Böylece 'abc*'. Yıldız işareti, değerdeki son ya da tek karakter olmalıdır.

Sosyal değerlerde soru işareti (?) ve iki nokta üst üste (:) kullanılamaz.

Bir alanda bu özel karakterlerden herhangi birini kullanmanız gerektiğinde (örneğin, bir açıklama parçası olarak), tüm dizilimi tek tırnak içine almalısınız.

Karakter	Tanım
	Ayrırcı olarak boşluk kullanılır. Kesme imi (') içine alınmış dizgiler dışında, birden çok boşluk tek bir boşlukla eşdeğerdir. Bu dizgi özniteliklerinde, MQCHARV tiplerine dayalı sondaki boşluklar önemli olarak işlenir.
,	Virgül ayrırcı olarak kullanılır. Birden çok virgül, tek tırnak (') içine alınmış dizgiler dışında, tek bir virgülle eşdeğerdir.
'	Kesme işareti, bir dizginin başlangıcını ya da sonunu belirtir. IBM MQ , tırnak içine alınmış olan tüm karakterleri tam olarak girdikleri şekilde bırakır. Dizginin uzunluğu hesaplanırken, içeren kesme imi içerilmez.
"	Bir dizginin içindeki tek tırnak işaretleri, dizginin uzunluğunu hesaplarken IBM MQ tarafından bir karakter olarak ele alınır ve dizgi sonlandırılmaz.
=	 z/OS üzerinde, eşittir işareti, virgülle ya da boşluktan sona eren bir parametre değerinin başlangıcını belirtir.
(Açma ayrırcı, bir parametre değerinin ya da değer listesinin başlangıcını belirtir.
)	Kapama ayrırcı, bir değiştirge değerinin ya da değer listesinin sonunu belirtir.
:	İki nokta üst üste işareti de kapsayıcı bir aralığı belirtir. Örneğin (1: 5) anlamı (1,2,3,4, 5) anlamına gelir. Bu gösterim yalnızca TRACE komutlarında kullanılabilir.
*	Yıldız işareti her şeyi ifade eder. Örneğin, DISPLAY TRACE (*) , tüm izleri görüntülemek ve DISPLAY QUEUE (PAY*) anlamına gelen, PAY ile başlayan adlara sahip tüm kuyrukların görüntülenmesi anlamına gelir.

PCF komutlarını kullanarak IBM MQ yönetimini otomatikleştirme

Bazı yönetim ve izleme görevlerini otomatikleştirmek için kuruluşunuzda yararlı olabilecek bir karar verebileceğinize karar verebilirsiniz. Programlanabilir komut biçimi (PCF) komutlarını kullanarak, hem yerel hem de uzak kuyruk yöneticilerine ilişkin yönetim görevlerini otomatikleştirebilirsiniz. Bu bölümde, IBM MQ nesnelere yönetme deneyiminiz olduğu varsayılır.

PCF komutları

IBM MQ programmable command format (PCF) commands can be used to program administration tasks into an administration program. Bu şekilde, bir programdan, kuyruk yöneticisi nesnelere (kuyruklar, süreç tanımlamaları, ad listeleri, kanallar, istemci bağlantı kanalları, dinleyiciler, hizmetler ve kimlik doğrulama bilgileri nesnelere) yönlendirebilir ve hatta kuyruk yöneticilerini de yönlendirebilirsiniz.

PCF komutları, MQSC komutları tarafından sağlanan işlevlerin aynı aralığını kapsamaya devam eder. Bir programı, tek bir düğümden ağdaki herhangi bir kuyruk yöneticisine PCF komutları yayınlayabileceğiniz şekilde yazabilirsiniz. Bu şekilde, yönetim görevlerini merkezileştirebilir ve otomatikleştirebilirsiniz.

Her PCF komutu, bir IBM MQ iletilerinin uygulama verileri bölümüne gömülü bir veri yapısıdır. Her komut, başka bir iletiyle aynı şekilde, MQPUT MQI işlevini kullanarak hedef kuyruk yöneticisine gönderilir. Komut sunucusu, iletiyi alan kuyruk yöneticisinde çalıştırıyor; komut sunucusu bu iletiyi bir komut iletileri olarak yorumluyor ve komutu çalıştırır. Yanıtları almak için, uygulama bir MQGET çağrısını yayınlar ve yanıt verileri başka bir veri yapısında döndürülür. Uygulama daha sonra yanıtı işleyebilir ve uygun şekilde davranabilir.

Not: MQSC komutlarının tersine, PCF komutları ve yanıtları okuyabileceğiniz bir metin biçiminde değildir.

Kısaca, PCF komut iletileri yaratmak için gerekli olan bazı şeyler şunlardır:

İleti tanımlayıcısı

Bu, standart bir IBM MQ ileti tanımlayıcısının içinde yer alan bir açıklamadır:

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

Uygulama Verileri

PCF üstbilgisini de içeren PCF iletilerini içerir. Bu ileti aşağıdaki gibi:

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

PCF veri yapılarının tam açıklaması ve bunların nasıl gerçekleştirileceği için bkz. [“IBM MQ Programlanabilir Komut Biçimlerine Giriş” sayfa 23.](#)

PCF nesnesi öznitelikleri

PCF ' deki nesne öznitelikleri, MQSC komutları için olduğu gibi sekiz karakterle sınırlı değildir. Bunlar, bu kılavuzda italik kılavuzda gösterilirler. Örneğin, RQMNAME değerinin PCF eşdeğeri *RemoteQMGrName* olur.

Çıkış PCF ' leri

Çıkış PCF ' leri, ileti metninde MQSC komutları içeren PCF komutlarıdır. Bir uzak kuyruk yöneticisine komut göndermek için PCF ' leri kullanabilirsiniz. Çıkış PCF ' leri hakkında ek bilgi için [Escape\(Esc\)](#) konusuna bakın.

IBM MQ Programlanabilir Komut Biçimlerine Giriş

Programlanabilir Komut Biçimleri (PCF ' ler), bir program ile ağ içindeki herhangi bir kuyruk yöneticisi (PCF ' leri destekleyen) arasında değiş tokuş edilebilen komut ve yanıt iletilerini tanımlar. PCF ' ler kuyruk yöneticisi yönetimini ve diğer ağ yönetimini basitleştirir. Bunlar, dağıtık ağların karmaşık yönetimi sorununu çözmek için kullanılabilir. Özellikle de, ağların büyüklüğü ve karmaşıklığı gibi büyüdükçe.

Programlanabilir Komut Biçimleri aşağıdaki tarafından desteklenmektedir:

-  IBM MQ for AIX
-  IBM MQ for IBM i
-  IBM MQ - Linux
-  IBM MQ for Windows
-  IBM MQ for z/OS

Sorun PCF komutlarının çözmesi

Dağıtılmış ağların yönetimi karmaşık hale gelebilir. Yönetim sorunları, ağların büyüklüğü ve karmaşıklığı arttıkça gelişmeye devam ediyor.

İleti alışverişi ve kuyruğa alma ile ilgili denetim örnekleri şunlardır:

- Kaynak yönetimi.
Örneğin, kuyruk yaratma ve silme.
- Performans izleme.
Örneğin, kuyruk derinliği üst sınırı ya da ileti hızı üst sınırı.
- Kontrol.
Örneğin, kuyruk derinliği üst sınırı, ileti uzunluğu üst sınırı gibi kuyruk parametrelerinin ayarlanması ve kuyrukların etkinleştirilmesi ve geçersiz kılınması.
- İleti yönlendirmesi.
Bir ağ üzerinden alternatif rotaların tanımı.

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

PCF ' ler nedir?

PCF 'ler, bir program ile ağ içindeki herhangi bir kuyruk yöneticisi (PCF ' leri destekleyen) arasında değiş tokuş edilebilen komut ve yanıt iletilerini tanımlar. You can use PCF commands in a systems management application program for administration of IBM MQ objects: authentication information objects, channels, channel listeners, namelists, process definitions, queue managers, queues, services, and storage classes. Uygulama, yerel kuyruk yöneticisini kullanarak komut ve yanıt bilgilerini herhangi bir kuyruk yöneticisiyle, yerel ya da uzak herhangi bir kuyruk yöneticisiyle iletişim kurmak için ağdaki tek bir noktadan çalıştırılabilir.

Her kuyruk yöneticisinin standart kuyruk adı olan bir denetim kuyruğu vardır ve uygulamanız PCF komut iletilerini o kuyruğa gönderebilir. Her kuyruk yöneticisinin, denetim kuyruğundan 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, belirtilen yanıt kuyruğunuzu kullanarak uygulamanızı geri döndürülebilirler. PCF komutları ve yanıt iletileri, olağan Message Queue Interface (MQI) kullanılarak gönderilir ve alınır.

Parametreleri de içinde olmak üzere, kullanılabilir PCF komutlarının bir listesi için [Programlanır Komut Biçimlerinin Tanımlamaları](#) konusuna bakın.

IBM MQ Programlanır Komut Biçimlerinin Kullanılması

You can use PCFs in a systems management program for IBM MQ remote administration.

Bu bölümde aşağıdakiler yer alır:

- [“PCF komut iletileri” sayfa 25](#)
- [“IBM MQ' da PCF yanıtları” sayfa 27](#)

- **z/OS** “Genişletilmiş yanıtlar” sayfa 29
- [IBM MQ nesnelerinin adlandırılmasına ilişkin kurallar](#)
- [“IBM MQ' ta PCF komutları için yetki denetimi” sayfa 30](#)

PCF komut iletileri

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

Her komut ve parametreleri, bir PCF üstbilgisini içeren ayrı bir komut iletilisi olarak ve ardından bir dizi parametre yapısıyla gönderilir; PCF üstbilgisinin ayrıntıları için bkz. MQCFH-PCF üstbilgisine bir parametre yapısı örneęi için bakınız: MQCFST-PCF dizgi deęiřtirgesi. PCF üstbilgisi, komutu ve aynı iletide izleyen parametre yapılarının sayısını tanımlar. Her parametre yapısı komutta bir parametre saęlar.

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

Multi Multiplatforms' ta, PCF komutlarının gönderileceęi kuyruk her zaman SYSTEM.ADMIN.COMMAND.QUEUE.

z/OS z/OS'ta, komutlar SYSTEM.COMMAND.INPUT, ancak SYSTEM.ADMIN.COMMAND.QUEUE , bu iş için bir dięer ad olabilir. Bu kuyruęa hizmet veren komut sunucusu, yanıtları, komut iletisinin ileti tanımlayıcısında *ReplyToQ* ve *ReplyToQMgr* alanları tarafından tanımlanan kuyruęa gönderir.

PCF komut iletileri nasıl yayınlanacak?

PCF komutunu ve yanıt iletilerini kuyruklarından ve kuyruklarından almak ve almak için, olaęan Message Queue Interface (MQI) çağrılarını, MQPUT, MQGET vb. ' yi kullanın.

Not:

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

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

Bir PCF komutuna iliřkin ileti tanımlayıcısı

IBM MQ ileti tanımlayıcısı, [MQMD-İleti tanımlayıcısı](#) içinde tam olarak belgelenir.

Bir PCF komut iletilisi, ileti tanımlayıcısında ařaęıdaki alanları içerir:

Rapor

Herhangi bir geęerli deęer, gerektięi gibi.

MsgType

Yanıt gerektiren bir iletiyi belirtmek için bu alanın MQMT_REQUEST olması gerekir.

Son kullanma tarihi

Herhangi bir geęerli deęer, gerektięi gibi.

Geribildirim

MQFB_NONE deęerine ayarla

Multi **Kodlama**

Bir IBM MQ for Multiplatforms sistemine gönderiyorsanız, bu alanı ileti verileri için kullanılan kodlamaya ayarlayın. Gerekirse dönüřtürme geęerleştirilir.

Multi **CodedCharSetId**

Bir IBM MQ for Multiplatforms sistemine gönderiyorsanız, bu alanı ileti verileri için kullanılan kodlanmış karakter takımı tanıtıcısı olarak ayarlayın. Gerekirse dönüřtürme geęerleştirilir.

Biçim

MQFMT_ADMIN değerine ayarlayın.

Öncelik

Herhangi bir geçerli değer, gerektiği gibi.

Kalıcılık

Herhangi bir geçerli değer, gerektiği gibi.

MsgId

Gönderme uygulaması herhangi bir değer belirtebilir ya da kuyruk yöneticisinin benzersiz bir ileti tanıtıcısı oluşturmasını istemek için MQMI_NONE belirtilebilir.

CorrelId

Gönderme uygulaması herhangi bir değer belirtebilir ya da ilinti tanıtıcısı olmadığını belirtmek için MQCI_NONE belirtilebilir.

ReplyToQ

Yanıtı almak için kuyruğun adı.

ReplyToQMgr

Yanıta ilişkin kuyruk yöneticisinin adı (ya da boşluk).

İleti bağlamı alanları

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

Bir 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 değerine ayarla

OriginalLength

MQOL_UNDEFINED değerine ayarla

Kullanıcı Verilerinin Gönderilmesi

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

PCF iletilerinin belirlenen bir kuyruğa gönderilmesi ve alınması**PCF iletileri belirtilen bir kuyruğa gönderiliyor**

Belirlenen bir kuyruğa ileti göndermek için, mqPutBag çağrısı belirtilen torbayı bir PCF iletisine dönüştürür ve iletiyi belirtilen kuyruğa gönderir. Çantanın içeriği aramadan sonra değişmeden kalır.

Bu aramaya giriş olarak, aşağıdaki bilgileri sağlamanız gerekir:

- Bir MQI bağlantı tanıtıcısı.
- İletinin yerleştirileceği kuyruk için bir nesne tanıtıcısı.
- Bir ileti tanımlayıcısı. İleti tanımlayıcısına ilişkin daha fazla bilgi için bakınız: [MQMD-Message descriptor](#).
- MQPMO yapısını kullanarak İleti Seçenekleri 'ni yerleştirin. MQPMO yapısı hakkında daha fazla bilgi için bakınız: [MQPMO-Sput-message options](#).
- Bir iletiye dönüştürülecek çantanın tanıtıcısı.

Not: Çanta bir denetim iletisi içeriyorsa ve çantaya değer eklemek için mqAddInquiry çağrısı kullanıldıysa, MQIASY_COMMAND veri ögesinin değeri, MQAI tarafından tanınan bir SORGUL KOMUTU olmalıdır.

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

Belirlenen kuyruktan PCF iletileri alınıyor

Belirlenen kuyruktan ileti almak için, mqGetBag çağrısı, belirlenen bir kuyruktan bir PCF iletisi alır ve ileti verilerini bir veri torbasına dönüştürür.

Bu aramaya giriş olarak, aşağıdaki bilgileri sağlamanız gerekir:

- Bir MQI bağlantı tanıtıcısı.
- İletinin 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 (bu, MQGMO_SYNCPOINT seçeneği ile) ve iletinin biçimi desteklenmeyen bir biçimde varsa, iş birimi yedeklenebilir. Daha sonra, ileti kuyruktan yeniden yürürlüğe alınır ve mqGetBag çağrısı yerine MQGET çağrısını kullanarak alınabilir. İleti tanımlayıcısına ilişkin ek bilgi için [MQGMO-Get-message options](#) başlıklı konuya bakın.

- MQGMO yapısını kullanarak İleti Seçenekleri 'ni alın. MQGMO yapısı hakkında daha fazla bilgi için bakınız: [MQMD-Message Descriptor](#).
- Dönüştürülen iletiyi içermek için çantanın tanıtıcısı.

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

IBM MQ' da PCF yanıtları

Her komuta yanıt olarak, komut sunucusu bir ya da daha fazla yanıt iletisi oluşturur. Bir yanıt iletisinin, bir komut iletiyle benzer bir biçimi vardır.

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

Komut iletisinin PCF üstbilgi tipi MQCFT_COMMAND ise, yalnızca standart yanıtlar oluşturulur. Bu komutlar, z/OS dışındaki tüm platformlarda desteklenir. Older applications do not support PCF on z/OS ; the IBM MQ Windows Explorer is one such application (however, the IBM WebSphere MQ 6.0 or later IBM MQ Explorer does support PCF on z/OS).

Komut iletisinin PCF üstbilgisi tipi MQCFT_COMMAND_XR ise, genişletilmiş ya da standart yanıtlar oluşturulur. Bu komutlar, z/OS ve diğer bazı platformlarda desteklenir. z/OS üzerinde yayınlanan komutlar yalnızca genişletilmiş yanıtlar oluşturur. Diğer platformlarda, herhangi bir yanıt türü oluşturulabilir.

Tek bir komut soysal bir nesne adı belirtiyorsa, eşleşen her nesne için kendi iletisinde ayrı bir yanıt döndürülür. Yanıt oluşturma işlemi için, soysal adı taşıyan tek bir komut birden çok bağımsız komut olarak kabul edilir (MQCFC_SON ya da MQCFC_NOT_SON denetim alanı dışında). Ters durumda, bir komut iletisi bir yanıt iletisi oluşturur.

Bazı PCF yanıtları, istenmemiş olsa bile bir yapı döndürebilir. Bu yapı, yanıtın tanımlamasında gösterilir ([Programların Komut Biçimlerinin Tanımlamaları](#)) *her zaman geri döndürülür*. Bu nedenle, bu yanıtlar için, verilerin hangi nesneye uygulanacağı saptanması için yanıtta nesnelerin adının belirlenmesi gerekir.

Bir yanıt için ileti tanımlayıcısı

Yanıt iletisi, ileti tanımlayıcısında aşağıdaki alanları içerir:

MsgType

Bu alan MQMT_REPLY ' dir.

MsgId

Bu alan kuyruk yöneticisi tarafından oluşturulur.

CorrelId

Bu alan, komut iletisine ilişkin rapor seçeneklerine göre oluşturulur.

Biçim

Bu alan, MQFMT_ADMIN 'dir.

Kodlama

MQENC_NATIVE değerine ayarlayın.

CodedCharSetId

MQCCSI_Q_MGR olarak ayarlayın.

Kalıcılık

Komut iletisinde olduğu gibi.

Öncelik

Komut iletisinde olduğu gibi.

Yanıt MQPMO_PASS_IDENTITY_CONTEXT ile oluşturuldu.

Standart yanıtlar

MQCFT_COMMAND üstbilgi tipine sahip komut iletileri oluşturulur, standart yanıtlar oluşturulur. Bu komutlar, z/OS dışındaki tüm platformlarda desteklenir.

Üç tip standart yanıt vardır:

- Tamam yanıtı
- Hata yanıtı
- Veri yanıtı

Tamam yanıtı

Bu yanıt, bir MQCC_OK ya da MQCC_UYARI *CompCode* alanı ile bir komut biçimi üstbilgisiyle başlayan bir iletten oluşur.

MQCC_OK için, *Reason* MQRC_NONE olur.

MQCC_UYARI için *Reason* , uyarının niteliyi tanımlar. Bu durumda, komut biçimi üstbilgisi, bu neden koduna uygun bir ya da daha çok uyarı parametresi yapısı izlenebilir.

Her iki durumda da, bir sorgu komutuna ilişkin ek parametre yapıları aşağıdaki kısımlarda açıklandığı gibi izleyebilir.

Hata yanıtı

Komutta bir hata varsa, bir ya da daha çok hata yanıtı iletisi gönderilir (olağan durumda yalnızca tek bir yanıt iletisi olacak bir komut için birden çok ileti gönderilebilir). Bu hata yanıtı iletileri, uygun şekilde MQCFC_SON ya da MQCFC_NOT_SON değerine sahiptir.

Bu tür her ileti, *CompCode* değeri MQCC_FAILED ve belirli bir hatayı tanımlayan bir *Reason* alanı ile yanıt biçimi üstbilgisiyle başlar. Genel olarak, her ileti farklı bir hatayı açıklar. Ayrıca, her ileti, üstbilgiyi izleyen sıfır ya da bir (hiçbir zaman birden fazla) hata parametre yapılarına sahip olur. Bu parametre yapısı, varsa, bir MQCFIN yapısı ve aşağıdakilerden birini içeren bir *Parameter* alanı ile birlikte:

- MQIACF_PARAMETER_ID

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

- MQIACF_ERROR_ID

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

- MQIACF_SELECTOR

Bu değer, komutla gönderilen bir liste yapısı (MQCFIL) yinelenen bir seçici içerirse ya da geçerli olmayan bir seçici içeriyorsa oluşur. Komut biçimi üstbilgisindeki *Reason* alanı hatayı tanımlar ve MQCFIN yapısındaki *Value* alanı, hatanın içindeki MQCFIL yapısındaki parametre değeridir.

- MQIACF_ERROR_OFFSET

Bu değer, Ping Kanalı komutunda bir veri karşılaştırma hatası olduğunda ortaya çıkar. Yapıdaki *Value* alanı, Ping Kanalı karşılaştırma hatasının görelî konutdur.

- MQIA_CODED_CHAR_SET_ID

Bu değer, gelen PCF komut iletisinin ileti tanımlayıcısındaki kodlanmış karakter takımı tanıtıcısı, hedef kuyruk yöneticisininiyle eşleşmiyorsa oluşur. Yapıdaki *Value* alanı, kuyruk yöneticisinin kodlanmış karakter takımı tanımlayıcısıdır.

Son (ya da yalnızca) hata yanıtı iletisi bir özet yanıtıdır; *CompCode* alanı MQCC_FAILED ve bir *Reason* alanı MQRCCF_COMMAND_FAILED (başarısız). Bu iletinin, üstbilgiyi izleyen bir parametre yapısı yoktur.

Veri yanıtı

Bu yanıt, bir sorgu komutuna (daha önce açıklandığı gibi) bir Tamam yanıtı (anlatıldığı gibi) oluşur. Tamam yanıtı, istenen verileri içeren ek yapılarla ve Programlanabilir Komut Biçimlerinin Tanımlamaları içinde açıklandığı şekilde bulunur.

Uygulamalar, belirli bir sırayla döndürülmekte olan bu ek parametre yapılarına bağlı kalmamalıdır.

Genişletilmiş yanıtlar

z/OS üzerinde yayınlanan komutlar genişletilmiş yanıtlar oluşturur.

Üç tip genişletilmiş yanıt vardır:

- İleti yanıtı, MQCFT_XR_MSG tipi ile
- MQCFT_XR_ITEM tipi ile öge yanıtı
- MQCFT_XR_SUMMARY tipli özet yanıt

Her komut bir ya da daha çok yanıt kümesi üretebilir. Her yanıt kümesi, PCF üstbilgisinin *MsgSeqNumber* alanında 1 'den başlayarak sırayla numaralandırılmış bir ya da daha fazla iletiden oluşur. Her bir kümedeki son (ya da yalnızca) yanıtın *Control* alanının MQCFC_SON değeri var. Kümedeki diğer tüm yanıtlar için bu değer MQCFC_NOT_SON ' dir.

Herhangi bir yanıt, *Parameter* alanının MQBACF_RESPONSE_SET değerine ayarladığı isteğe bağlı MQCFBS yapılarından birini ya da daha fazlasını içerebilir. Değer, yanıt kümesi tanıtıcısı olarak ayarlanır. Tanıtıcılar benzersizdir ve yanıtı içeren yanıtların kümesini tanımlar. Her yanıt kümesi için, bunu tanımlayan bir MQCFBS yapısı vardır.

Genişletilmiş yanıtların en az iki parametre yapısı vardır:

- *Parameter* alanı MQBACF_RESPEONSE_ID değerine ayarlanmış bir MQCFBS yapısı. Bu alandaki değer, yanıtın ait olduğu yanıtların kümesinin tanıtıcısıdır. İlk setteki tanıtıcı isteğe bağlı. Sonraki kümelerde, tanıtıcı bir MQBACF_RESPONSE_SET yapısıyla önceden bildirilidir.
- *Parameter* alanı MQCACF_RESPONSE_Q_MGR_NAME olarak ayarlanmış bir MQCFST yapısı, yanıt kümesinin geldiği kuyruk yöneticisinin adı olan değer.

Birçok yanıtta ek parametre yapıları vardır ve bu yapılar aşağıdaki bölümlerde anlatılır.

MQCFC_SON bulunana kadar, yanıtların alınması dışında bir küme içinde kaç yanıt olduğunu önceden belirleyemezsiniz. Herhangi bir küme olarak kaç yanıt kümesi olduğunu önceden belirleyebilirsiniz de, ek kümelerin oluşturulduğunu belirtmek için MQBACF_RESPONSE_SET yapılarını içerebilecek şekilde ilerleyebilirsiniz.

Sorgulama komutlarına ek yanıtlar

Sorgu komutları olağan durumda, belirtilen arama ölçütleriyle eşleşen her öge için bir öge yanıtı (MQCFT_XR_ITEM tipinde) oluşturur. Öge yanıtında, üstbilgide MQCC_OK değeri olan bir *CompCode* alanı ve MQRC_NONE değeri olan bir *Reason* alanı var. Ayrıca, ögeyi ve istenen öznelikleri tanımlayan diğer değiştirge yapılarını da içerir (Programlanabilir Komut Biçimlerinin Tanımlamaları içinde açıklandığı gibi).

Bir öge hatalıysa, üstbilgideki *CompCode* alanında MQCC_FAILED değeri var ve *Reason* alanı, belirli bir hatayı tanımlar. Ögenin tanınması için ek parametre yapıları bulunur.

Belirli Sorgulama komutları, öge yanıtlarına ek olarak genel (ad-özel değil) ileti yanıtlarını döndürebilir. Bu yanıtlar, MQCFT_XR_MSG tipindeki yanıtların bilgi ya da hata olup olmadığını bildirir.

Sorgula komutu başarılı olursa, isteğe bağlı olarak bir özet yanıt (MQCFT_XR_SUMMARY tipi), MQCC_OK için *CompCode* değeri ve MQRC_NONE ile ilgili bir *Reason* alan değeri olabilir.

Sorgula komutu başarısız olursa, öge yanıtları döndürülebilir ve isteğe bağlı olarak *CompCode* değeri MQCC_FAILED ve bir *Reason* alan değeri olan MQRCCF_COMMAND_FAILED olan bir özet yanıtı (MQCFT_XR_SUMMARY tipi) olabilir.

Sorgudan başka komutlara verilen genişletilmiş yanıtlar

Başarılı komutlar, üstbilgideki *CompCode* alanının MQCC_OK değerine sahip olduğu ve *Reason* alanının MQRC_NONE değerine sahip olduğu ileti yanıtlarını oluşturur. Her zaman en az bir ileti vardır; bilgi verici (MQCFT_XR_MSG) ya da özet (MQCFT_XR_SUMMARY) olabilir. İsteğe bağlı olarak ek bilgi (MQCFT_XR_MSG tipi) iletileri de olabilir. Her bilgi iletisi, komutla ilgili ek bilgi içeren ek değiştirge yapılarını içerebilir; oluşabilecek yapılara ilişkin tek tek komut tanımlamalarına bakın.

Üstbilgideki *CompCode* alanının MQCC_FAILED değerine sahip olduğu ve *Reason* alanının belirli bir hatayı tanımlayıp, hata iletisi yanıtlarını (MQCFT_XR_MSG tipi) oluşturamayan komutlar. Her ileti, hatayla ilgili bilgi içeren ek değiştirge yapılarını içerebilir: oluşabilecek yapılara ilişkin tek tek hata açıklamalarına bakın. Bilgi iletisi yanıtı oluşturulabilir. There might, optionally, be a summary response (MQCFT_XR_SUMMARY), with a *CompCode* value of MQCC_FAILED, and a *Reason* field value of MQRCCF_COMMAND_FAILED.

CommandScope komutunu kullanarak komutlara verilen genişletilmiş yanıtlar

If a command uses the **CommandScope** parameter, or causes a command using the **CommandScope** parameter to be generated, there is an initial response set from the queue manager where the command was received. Daha sonra, komutun yönlendirildiği her bir kuyruk yöneticisi için ayrı bir küme ya da küme oluşturulur (birden çok bağımsız komut verildiyse gibi). Son olarak, alma kuyruk yöneticisinden, genel bir özet yanıtı içeren bir yanıt kümesi var (MQCFT_XR_SUMMARY yazın). MQCACF_RESPONSE_Q_MGR_NAME parametre yapısı, her bir kümeyi oluşturan kuyruk yöneticisini tanımlar.


İlk yanıt kümesi, aşağıdaki ek parametre yapılarına sahiptir:

- MQIACF_COMMAND_INFO (MQCFIN). Bu yapıdaki olası değerler MQCMDI_CMDSCOPE_ACCEPTED ya da MQCMDI_CMDSCOPE_GENERATED ' dir.
- MQIACF_CMDSCOPE_Q_MGR_COUNT (MQCFIN). Bu yapı, komutun gönderileceği kuyruk yöneticisi sayısını gösterir.

IBM MQ' ta PCF komutları için yetki denetimi

Bir PCF komutu işlendiğinde, gerekli IBM MQ nesne yetkisi denetimleri için komut iletisinde ileti tanımlayıcısından *UserIdentifier* değeri kullanılır. Yetki denetimi, bu konuda açıklandığı gibi her altyapıda farklı bir biçimde uygulanır.

Denetimler, komutun işlenmekte olduğu sistemde gerçekleştirilir; bu nedenle, bu kullanıcı kimliği hedef sistemde var olmalıdır ve komutu işlemek için gerekli yetkilerine sahip olmalıdır. İleti uzak bir sistemden geldiyse, hedef sistemde varolan kimliği elde etmenin bir yolu, hem yerel hem de uzak sistemlerde eşleşen bir kullanıcı kimliğine sahip olmaktır.

Not:  z/OS' ta yetki denetimi hakkında bilgi için bkz. [Görev 1: z/OS sistem parametrelerini belirleme.](#)

IBM MQ for IBM i

IBM i

Herhangi bir PCF komutunu işlemek için, kullanıcı kimliğinin hedef sistemdeki IBM MQ nesnesi için *dsp* yetkisine sahip olması gerekir.

In addition, IBM MQ object authority checks are performed for certain PCF commands, as shown in [Çizelge 2 sayfa 32.](#)

Çoğu durumda bu denetimler, yerel bir sistemde yayınlanan eşdeğer IBM MQ CL komutlarıyla gerçekleştirilen denetimlerle aynı denetimlerdir. IBM MQ yetkilerinden IBM i sistem yetkililerine eşleme ve IBM MQ CL komutlarına ilişkin yetki gereklilikleri hakkında daha fazla bilgi için [IBM i üzerinde güvenliğin ayarlanması](#) başlıklı konuya bakın. Çıktılarla ilgili güvenlik ayrıntıları, [Güvenlik çıkışı kullanarak bağlantı düzeyinde güvenlik](#) belgelerinde verilmiştir.

Aşağıdaki komutların herhangi birini işlemek için kullanıcı kimliğinin QMQMADM grup tanıtımının bir üyesi olması gerekir:

- Ping Kanalı
- Kanalı Değiştir
- Kanalı Kopyala
- Kanal Yarat
- Kanalı Sil
- Kanalı Sıfırla
- Kanalı Çözümle
- Başlangıç Kanalı
- Kanalı Durdur
- Kanal Başlatıcısını Başlat
- Kanal Dinleyicisi Başlat

IBM MQ for UNIX, Linux, and Windows

ALW

Herhangi bir PCF komutunun işlenmesi için, hedef sistemdeki kuyruk yöneticisi nesnesi için kullanıcı kimliği *dsp* yetkisine sahip olmalıdır. In addition, IBM MQ object authority checks are performed for certain PCF commands, as shown in [Çizelge 2 sayfa 32.](#)

Aşağıdaki komutların herhangi birini işlemek için the user ID must belong to group *mqm*.

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

- Kanalı Değiştir
- Kanalı Kopyala
- Kanal Yarat
- Kanalı Sil
- Ping Kanalı
- Kanalı Sıfırla
- Başlangıç Kanalı
- Kanalı Durdur
- Kanal Başlatıcısını Başlat

- Kanal Dinleyicisi Başlat
- Kanalı Çözümle
- Kümeyi Sıfırla
- Kümeyi Yenile
- Kuyruk Yöneticisini Askıya Al
- Kuyruk Yöneticisini Sürdür

Çoklu Platformlar için IBM MQ Nesnesi Yetkilileri

Multi

Çizelge 2. Nesne yetkileri		
Komut	IBM MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
Kimlik Doğrulama Bilgilerini Değiştir	dsp ve chg	yok
Kanalı Değiştir	dsp ve chg	yok
Kanal Dinleyiciyi Değiştir	dsp ve chg	yok
İstemci Bağlantı Kanalını Değiştir	dsp ve chg	yok
Ad listesini değiştir	dsp ve chg	yok
Süreci Değiştir	dsp ve chg	yok
Kuyruğu Değiştir	dsp ve chg	yok
Kuyruk Yöneticisini Değiştir	chg bkz. Not 3 ve Not 5	yok
Hizmeti Değiştir	dsp ve chg	yok
Kuyruğu Temizle	klr	yok
Kimlik Doğrulama Bilgilerini Kopyala	dsp	crt
Kimlik Doğrulama Bilgilerini Kopyala (Değiştir) bkz. Not 1	from: dsp to: chg	crt
Kanalı Kopyala	dsp	crt
Kanal Kopyala (Değiştir) bkz. Not 1	from: dsp to: chg	crt
Kanal Dinleyicisi Kopyala	dsp	crt
Channel Listener 'ı Kopyala (Değiştir) bkz. Not 1	from: dsp to: chg	crt
İstemci Bağlantı Kanalını Kopyala	dsp	crt
İstemci Bağlantı Kanalını Kopyala (Değiştir) bkz. Not 1	from: dsp to: chg	crt
Ad listesini kopyala	dsp	crt
Ad Listesi Kopyala (Değiştir) bkz. Not 1	from: dsp to: dsp ve chg	crt
İşlemi Kopyala	dsp	crt

Çizelge 2. Nesne yetkileri (devamı var)

Komut	IBM MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
Kopyalama İşlemi (Değiştir) bkz. Not 1	from: dsp to: chg	crt
Kuyruğu Kopyala	dsp	crt
Kopyalama Kuyruğu (Değiştir) bkz. Not 1	from: dsp to: dsp ve chg	crt
Kimlik Doğrulama Bilgileri Oluştur	(sistem varsayılan kimlik doğrulama bilgileri) dsp	crt
Kimlik Doğrulama Bilgileri Oluştur (Değiştir) bkz. Not 1	(sistem varsayılan kimlik doğrulama bilgileri) dsp to: chg	crt
Kanal Yarat	(sistem varsayılan kanalı) dsp	crt
Kanal Yarat (Değiştir) bkz. Not 1	(sistem varsayılan kanalı) dsp to: chg	crt
Kanal Dinleyicisi Yarat	(sistem varsayılan dinleyicisi) dsp	crt
Kanal Dinleyici Yarat (Değiştir) bkz. Not 1	(sistem varsayılan dinleyicisi) dsp to: chg	crt
İstemci Bağlantı Kanalı Oluştur	(sistem varsayılan kanalı) dsp	crt
İstemci Bağlantı Kanalı Oluştur (Değiştir) bkz. Not 1	(sistem varsayılan kanalı) dsp to: chg	crt
Ad Listesi Yarat	(sistem varsayılan ad listesi) dsp	crt
Ad Listesi Yarat (Değiştir) bkz. Not 1	(sistem varsayılan ad listesi) dsp to: dsp ve chg	crt
Süreç Yarat	(sistem varsayılan işlemi) dsp	crt
Süreç Yarat (Değiştir) bkz. Not 1	(sistem varsayılan süreci) dsp to: chg	crt
Kuyruk Yarat	(sistem varsayılan kuyruğu) dsp	crt
Kuyruk Yarat (Değiştir) bkz. Not 1	(sistem varsayılan kuyruğu) dsp to: dsp ve chg	crt
Hizmet oluştur	(sistem varsayılan kuyruğu) dsp	crt
Hizmet Oluştur (Değiştir) bkz. Not 1	(sistem varsayılan kuyruğu) dsp to: chg	crt
Kimlik Doğrulama Bilgilerini Sil	dsp ve dlt	yok
Yetki Kaydını Sil	(kuyruk yöneticisi nesnesi) chg bkz. Not 4	bkz. Not 4
Kanalı Sil	dsp ve dlt	yok
Kanal Dinleyiciyi Sil	dsp ve dlt	yok
İstemci Bağlantı Kanalını Sil	dsp ve dlt	yok
Ad listesini sil	dsp ve dlt	yok
Süreci Sil	dsp ve dlt	yok
Kuyruğu sil	dsp ve dlt	yok
Hizmeti Sil	dsp ve dlt	yok

Çizelge 2. Nesne yetkileri (devamı var)

Komut	IBM MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
Kimlik Doğrulama Bilgilerini Sorgula	dsp	yok
Yetki Kayıtlarını Sorgula	<i>bkz. Not 4</i>	<i>bkz. Not 4</i>
Kanal Sorgula	dsp	yok
Kanal İletişimcisi ile	dsp	yok
Sorgu Kanalı Durumu (ChannelType MQCHT_CLSSDR için)	inq	yok
İstemci Bağlantı Kanalını Sorgula	dsp	yok
Sorgu Adı Listesi	dsp	yok
Süreç Sorgula	dsp	yok
Sorgu Kuyruğu	dsp	yok
Sorgu Kuyruğu Yöneticisi	<i>not 3 'e bakın</i>	yok
Sorgu Kuyruğu Durumu	dsp	yok
Hizmeti Sorgula	dsp	yok
Ping Kanalı	Ctrl	yok
Ping Kuyruğu Yöneticisi	<i>not 3 'e bakın</i>	yok
Kuyruk Yöneticisini Yenile	(kuyruk yöneticisi nesnesi) chg	yok
Güvenliği Yenile (SecurityType MQSECTYPE_SSL için)	(kuyruk yöneticisi nesnesi) chg	yok
Kanalı Sıfırla	ctrlx	yok
Kuyruk Yöneticisini Sıfırla	(kuyruk yöneticisi nesnesi) chg	yok
Kuyruk İstatistiklerini Sıfırla	dsp ve chg	yok
Kanalı Çözümle	ctrlx	yok
Yetki Kaydını Ayarla	<i>(kuyruk yöneticisi nesnesi) chg bkz. Not 4</i>	<i>bkz. Not 4</i>
Başlangıç Kanalı	Ctrl	yok
Kanalı Durdur	Ctrl	yok
Bağlantıyı Durdur	(kuyruk yöneticisi nesnesi) chg	yok
Dinleyici Başlat	Ctrl	yok
Dinleyiciyi Durdur	Ctrl	yok
Hizmeti Başlat	Ctrl	yok
Hizmeti Durdur	Ctrl	yok
Esc	<i>bkz. Not 2</i>	<i>bkz. Not 2</i>

Notlar:

1. Değiştirilecek nesne varsa, bu komut geçerli olur; tersi durumda, yetki denetimi Create ya da Copy withod Replace gibi olur.
2. Gereken yetki, çıkış metninin tanımladığı MQSC komutu tarafından belirlenir ve önceki komutlardan birine eşdeğerdir.
3. Herhangi bir PCF komutunu işleyebilmek için, kullanıcı kimliğinin hedef sistemdeki kuyruk yöneticisi nesnesine ilişkin dsp yetkisi olmalıdır.
4. Bu PCF komutu, komut sunucusu -a değiştirilmesiyle başlatılmadıkça yetkilendirilir. Varsayılan olarak, komut sunucusu kuyruk yöneticisi başlatıldığında ve -a değiştirilmesi olmadan başlar. Ek bilgi için [Programların komut biçimleri başvurusubaşlıklı konuya](#) bakın.
5. Kuyruk yöneticisi için bir kullanıcı kimliği chg yetkisi verilmesi, tüm gruplar ve kullanıcılar için yetki kayıtlarını belirleme yeteneği verir. Bu yetkiyi sıradan kullanıcılara ya da uygulamalara atamayın.

IBM MQ , güvenlik denetleyebilmesi için kendi kullanıcı çıkış programlarınızı sağlayabilmeniz için bazı kanal güvenliği çıkış noktaları da sağlar. Ek bilgi için [Kanal görüntülenmesibaşlıklı konuya](#) bakın.

Multi

PCF 'lerin kullanımını kolaylaştırmak için MQAI' nin kullanılması

The IBM MQ Administration Interface (MQAI) is a programming interface to IBM MQ that is available on AIX, IBM i, Linux, ve Windows. Nesnelere (ya da değiştiricileri) Programların Komut Biçimleri (PCF ' ler) kullanmaktan daha kolay bir biçimde işlemek için, veri torbalarını kullanarak bir IBM MQ kuyruk yöneticisinde yönetim görevlerini gerçekleştirir.

The MQAI performs administration tasks on a queue manager through the use of *veri torbaları*. Veri torbaları, nesnelere özelliklerini (ya da parametrelerini), PCF ' leri kullanmaktan daha kolay bir şekilde işlemeye olanak sağlar.

MQAI ' yi kullanmanın yararları aşağıdaki gibidir:

PCF iletilerinin kullanımını basitleştirin

MQAI, IBM MQ' i yönetmek için daha kolay bir yöntemdir. MQAI olanağını kullanırsanız, kendi PCF iletilerinizi yazmanıza gerek yoktur. Bu, karmaşık veri yapılarıyla ilişkili sorunların önlenmesini sağlar.

MQI çağrılarını kullanarak yazılmış programlardaki parametreleri geçirmek için, PCF iletileri komutun ve dizinin ya da tamsayı verilerinin ayrıntılarının içerilmeli. Bu yapılandırmayı el ile oluşturmak için, programınıza her yapı için birkaç deyim eklemeniz ve bellek alanı ayırmanız gerekir. Bu görev uzun ve zahmetli olabilir.

MQAI geçiş değiştiricileri kullanarak yazılan programlar uygun veri torbasına yazılır ve her yapı için tek bir deyim gereklidir. MQAI veri torbalarının kullanımı, dizileri işleme ve saklama alanı ayırma gereksinimini kaldırır ve PCF ' nin ayrıntılarından bir miktar yalıtım sağlar.

Hata koşullarını daha kolay ele al

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

Uygulamalar arasında veri alışverişi

Uygulama verileri, PCF biçiminde gönderilir ve MQAI tarafından paketlenmez ve paketlenmez. İletilerinizin tamsayılar ve karakter dizilerinden oluşuyorsa, PCF verileri için IBM MQ yerleşik veri dönüştürmesinden yararlanmak için MQAI ' yi kullanabilirsiniz. Bu, veri dönüştürme çıktılarına yazma gereksinimini önler.

Veri çantasını yarattıktan ve doldurduktan sonra, mqExecute çağrısını kullanarak bir kuyruk yöneticisinin komut sunucusuna bir denetim komutu iletileri gönderebilirsiniz. Bu çağrı herhangi bir yanıt iletileri için bekler. mqExecute çağrısı, komut sunucusuyla değiş tokuş işlemi gerçekleştirir ve yanıtları bir *yanıt çantası*na döndürür.

MQAI ' nin kullanılmasına ilişkin örnekler

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

- [amqsaicq.c](#): Yerel bir kuyruk oluşturun.

- [amqsaiem.c](#): Basit bir olay izleyicisini kullanarak ekranda olayları görüntüler.
- [amqsailq.c](#): Tüm yerel kuyrukların ve bunların yürürlükteki derinliklerinin bir listesini yazdırır.
- [amqsaicl.c](#): Tüm kanalların ve türlerinin listesini yazdırır.

MQAI uygulamanızı oluşturma

Uygulamanızı MQAI kullanarak oluşturmak için, IBM MQ ile aynı kitaplıklara bağlanmanızı sağlar. IBM MQ uygulamalarınızın nasıl oluşturulacağı hakkında bilgi için bkz. [yordamsal uygulama oluşturulması](#).

MQAI kullanarak IBM MQ ' ı yapılandırmaya ilişkin ipuçları ve öneriler

MQAI, komut sunucusuyla doğrudan ilgilenmek yerine, denetim komutlarını komut sunucusuna göndermek için PCF iletilerini kullanır. Tips for configuring IBM MQ using the MQAI can be found in [“Hints and tips for using MQAI to configure IBM MQ” sayfa 36](#).

İlgili başvurular

[IBM MQ Administration Interface başvurusu](#)

Multi

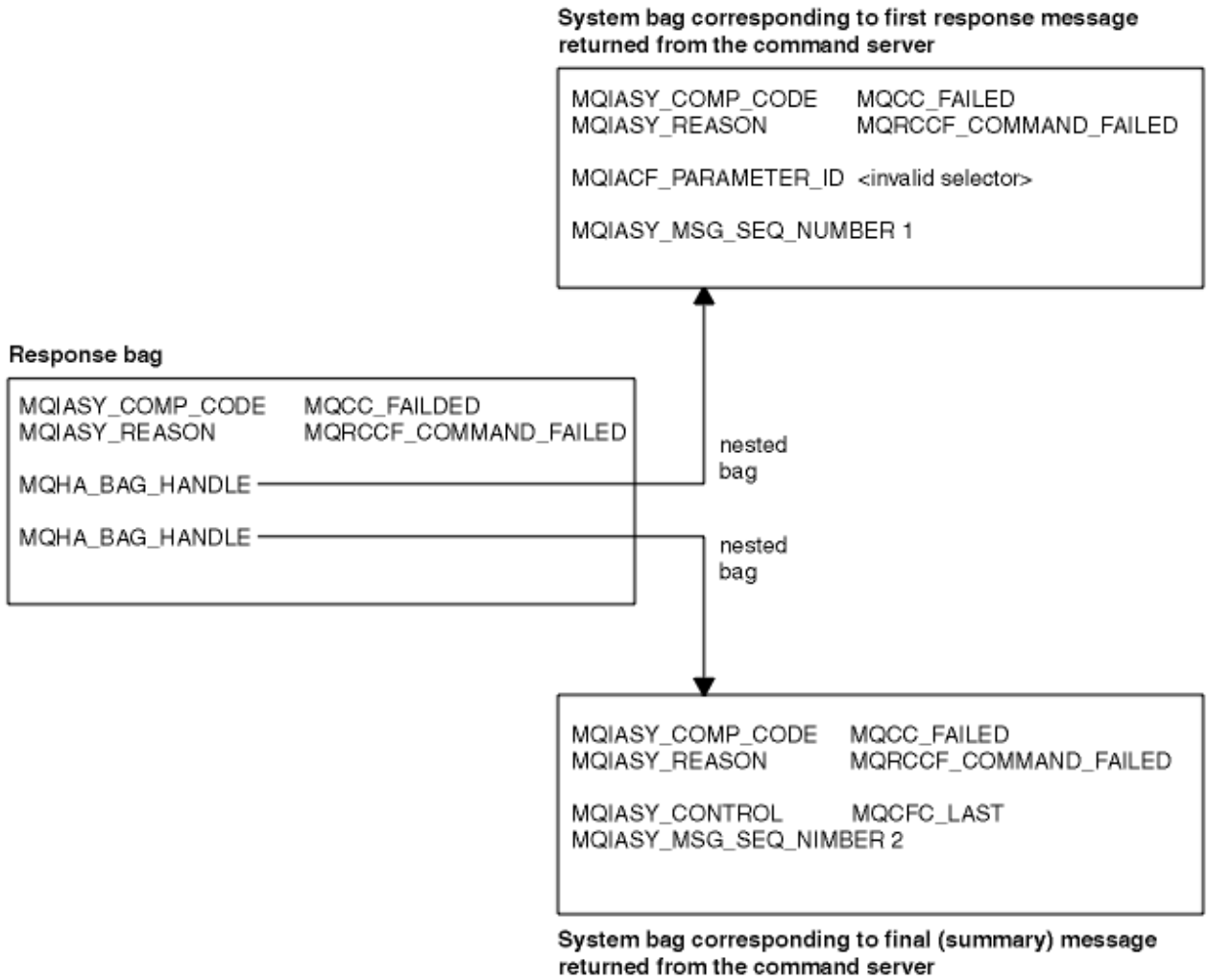
Hints and tips for using MQAI to configure IBM MQ

IBM MQ Denetim Arabirimi (MQAI), komut sunucusunun kendisiyle doğrudan ilgilenmek yerine, denetim komutlarını komut sunucusuna göndermek için PCF iletilerini kullanır. Burada, MQAI olanağını kullanarak IBM MQ ' un yapılandırılmasına ilişkin bazı ipuçları bulunur.

- IBM MQ içindeki karakter dizgileri, sabit uzunluğa kadar doldurulmuyorlar. C kullanarak, boş karakterlerle biten dizgiler, olağan durumda IBM MQ programlama arabirimlerine giriş değiştiricileri olarak sağlanabilir.
- Bir dizgi özniteliğinin değerini temizlemek için, özniteliği boş bir dizgi yerine tek bir boşluğa ayarlayın.
- Değiştirmek istediğiniz öznitelikleri önceden göz önünde bulundurun ve yalnızca bu öznitelikleri sorgulayın.
- Bazı öznitelikler değiştirilemez; örneğin, bir kuyruk adı ya da kanal tipi. Yalnızca değiştirilebilecek öznitelikleri değiştirmeyi denediğinizden emin olun. Belirli bir PCF değişiklik nesnesine ilişkin gerekli ve isteğe bağlı parametrelerin listesine bakın. Bkz. [Programlanabilir Komut Biçimlerinin Tanımlamaları](#).
- Bir MQAI çağrısı başarısız olursa, hataya ilişkin bazı ayrıntılar yanıt torbasına döndürülür. Daha sonra, MQHA_BAG_HANDLE seçicisi tarafından erişilebilen içiçe yerleştirilmiş bir çantada daha ayrıntılı bilgi bulunabilir. Örneğin, bir mqExecute çağrısı MQRCCF_COMMAND_FAILED neden koduyla başarısız olursa, yanıt torbasında bu bilgi döndürülür. Bu neden kodunun olası bir nedeni, belirtilen bir seçicinin komut iletilişi için geçerli olmadığından ve bu bilgi ayrıntısının, bir torba tanıtıcısı tarafından erişilebilen içiçe yerleştirilmiş bir torbada bulunması olabilir.

MQExecute ile ilgili daha fazla bilgi için bkz. [“Sending administration commands to the qm command server using the mqExecute call” sayfa 69](#)

Aşağıdaki çizge bu senaryoyu göstermektedir:



Multi Gelişmiş MQAI konuları

Dizin oluşturma, veri dönüştürme ve ileti tanımlayıcısının kullanımı hakkında bilgi

Dizin oluşturma

Dizinler, araya ekleme sırasını korumak için var olan veri öğelerini bir çantadan çıkarırken ya da çıkarırken kullanılır.

Veri dönüştürme

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

İleti tanımlayıcısının kullanımı

MQAI, veri torbası yaratıldığında bir başlangıç değeri olarak ayarlanmış bir ileti tanımlayıcısı oluşturur.

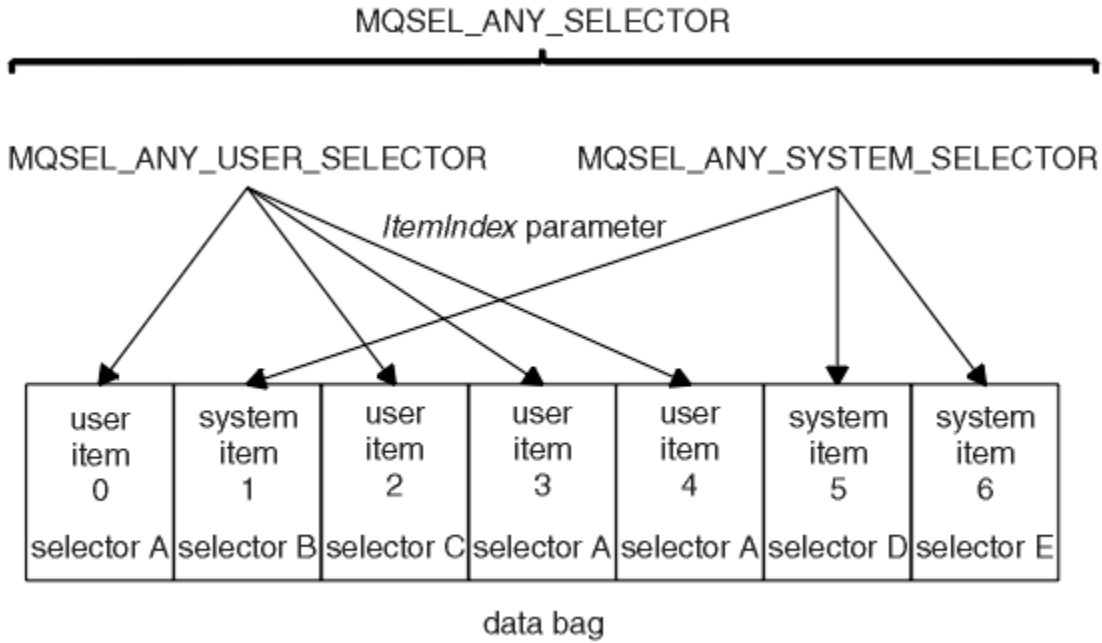
Multi MQAI içinde dizin oluşturma

Dizinler, var olan veri öğelerini bir çantadan değiştirirken ya da bir çantadan çıkarırken kullanılır. Üç tip dizin oluşturma vardır; bu da veri öğelerinin kolayca alınmasını sağlar.

Bir çantadaki bir veri öğesi içindeki her bir seçici ve değerin ilişkili üç dizin numarası vardır:

- Aynı seçiciye sahip diğer öğelerle göreli dizin.
- Öğenin ait olduğu seçici (kullanıcı ya da sistem) kategorisine göre dizin.
- Çantadaki (kullanıcı ve sistem) tüm veri öğelerle göreli dizin.

Bu, kullanıcı seçicileri, sistem seçicileri ya da her ikisi tarafından Şekil 3 sayfa 38' ta gösterildiği gibi dizinlemeye olanak sağlar.



Şekil 3. Dizin oluşturma

Şekil 3 sayfa 38' ta, kullanıcı ögesi 3 (seçici A) aşağıdaki izin çiftleri tarafından başvurulabilir:

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

Dizin, C içindeki bir dizi gibi sıfır tabanlı; eğer geçiş sayısı yoksa, dizin sıfır ile 'n-1' arasında değişir, boşluk olmaz.

Dizinler, var olan veri öğelerini bir çantadan değiştirirken ya da bir çantadan çıkarırken kullanılır. Bu şekilde kullanıldığında, ekleme sırası korunur, ancak diğer veri öğelerine ilişkin dizinler etkilenebilir. Buna örnek olarak bkz. “Bir çantadaki bilgilerin değiştirilmesi” sayfa 66 ve “Veri öğelerinin silinmesi” sayfa 68.

Üç tip izin oluşturma, veri öğelerinin kolayca alınmasına olanak sağlar. Örneğin, bir çantada belirli bir seçicinin üç eşgörünümü varsa, mqCountItems çağrısı o seçiciye ilişkin eşgörünümlerin sayısını sayabilir ve mqInquire * çağrılarını yalnızca bu değerleri sorgulamak için hem seçiciyi hem de dizini belirtebilir. Bu, kanallardaki çıkışların bazıları gibi bir değer listesi alabilen öznelikler için kullanışlıdır.

Multi MQAI ' de veri dönüştürme işlemi

Bir MQAI veri torbasında bulunan dizgiler, çeşitli kodlanmış karakter kümelerinde olabilir. Bu dizgiler, mqSetInteger çağrısı kullanılarak dönüştürülebilmektedir.

PCF iletileri gibi, bir MQAI veri torbasında bulunan dizgiler çeşitli kodlanmış karakter kümelerinde bulunabilir. Genellikle, bir PCF iletilerinde bulunan tüm dizgiler aynı kodlanmış karakter setinde bulunur; yani, kuyruk yöneticisiyle aynı ayardır.

Bir veri paketindeki her dizgi ögesi iki değer içerir; dizginin kendisi ve CCSID ' dir. Torbaya eklenen dizgi, mqAddDizgisi ya da mqSetDizgisi çağrısının **Buffer** değiştirgesinden alınır. CCSID değeri, bir MQIASY_CODED_CHAR_SET_ID seçicisini içeren sistem ögesinden alınır. Bu, çanta CCSID 'si olarak bilinir ve mqSetTamsayı çağrısını kullanarak değiştirilebilir.

Veri torbasında bulunan bir dizginin değerini sorguladığınızda, CCSID, çağrıdan gelen bir çıkış parametresidir.

Çizelge 3 sayfa 39 , veri torbalarını iletilere dönüştürürken uygulanan kuralları gösterir (tam tersi de geçerlidir):

Çizelge 3. CCSID işleme

MQAI çağırısı	CCSID	Aramak için giriş	Çağrılacak çıktı
mqBagToBuffer	Çanta CCSID 'si (<u>1</u>)	Yoksayıldı	Değiştirilmedi
mqBagToBuffer	Çantadaki Dizgi CCSID 'leri	Kullanılan	Değiştirilmedi
mqBagToBuffer	Arabellekteki CCSID CCSID 'leri	Burada geçerli değil	Paketteki CCSID 'lerden kopyalandı
mqBufferToBag	Çanta CCSID 'si (<u>1</u>)	Yoksayıldı	Değiştirilmedi
mqBufferToBag	Arabellekteki CCSID CCSID 'leri	Kullanılan	Değiştirilmedi
mqBufferToBagmqBufferToBag	Çantadaki Dizgi CCSID 'leri	Burada geçerli değil	Arabellek içindeki dizgi CCSID 'lerinden kopyalandı
mqPutBag	MQMD CCSID	Kullanılan	Değiştirilmemiş (<u>2</u>)
mqPutBag	Çanta CCSID 'si (<u>1</u>)	Yoksayıldı	Değiştirilmedi
mqPutBag	Çantadaki Dizgi CCSID 'leri	Kullanılan	Değiştirilmedi
mqPutBag	İletideki CCSID CCSID 'leri gönderildi	Burada geçerli değil	Paketteki CCSID 'lerden kopyalandı
mqGetBag	MQMD CCSID	İletin veri dönüştürmesi için kullanılır	Döndürülen veri CCSID değerine ayarla (<u>3</u>)
mqGetBag	Çanta CCSID 'si (<u>1</u>)	Yoksayıldı	Değiştirilmedi
mqGetBag	İletideki CCSID CCSID 'leri	Kullanılan	Değiştirilmedi
mqGetBag	Çantadaki Dizgi CCSID 'leri	Burada geçerli değil	İletideki CCSID CCSID 'lerinden kopyalandı
mqExecute	İstek-çanta CCSID 'si	İstek iletinin MQMD 'si için kullanılır (<u>4</u>)	Değiştirilmedi
mqExecute	Yanıt-çanta CCSID 'si	Yanıt iletinin veri dönüştürme için kullanılır (<u>4</u>)	Döndürülen veri CCSID değerine ayarla (<u>3</u>)
mqExecute	İstek paketindeki CCSID CCSID 'leri	İstek ileti için kullanılır	Değiştirilmedi
mqExecute	Yanıt paketindeki CCSID CCSID 'leri	Burada geçerli değil	Yanıt iletinde dizgi CCSID 'lerinden kopyalandı

Notlar:

1. Çanta CCSID 'si, selector MQIASY_CODED_CHAR_SET_ID olan sistem ögesidir.
2. MQCCSI_Q_MGR, gerçek kuyruk yöneticisi CCSID 'si olarak değiştirildi.
3. Veri dönüştürme istenirse, döndürülen veri CCSID değeri çıkış değeriyle aynı olur. Veri dönüştürme istenmiyorsa, döndürülen veri CCSID değeri, ileti değeriyle aynıdır. Veri dönüştürme istenirse, ancak başarısız olursa hiçbir ileti döndürülmediğini unutmayın.
4. CCSID ' nin CCSID değeri MQCC_DEFAULT ise, kuyruk yöneticisinin CCSID değeri kullanılır.

İlgili kavramlar

“Kodlanmış karakter kümeleri arasında veri dönüştürme” sayfa 188

IBM MQ tanımlı biçimlerde (yerleşik biçimler olarak da bilinir) ileti verileri, her iki karakter kümesinin tek bir dil ya da benzer dillerdeki bir grup ile ilişkisi olması koşuluyla, bir kodlanmış karakter takımından kuyruk yöneticisi tarafından dönüştürülebilmektedir.

“ccsid_part2.tbl dosyası” sayfa 190

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 'daki ileti tanımlayıcısının kullanılması

MQAI tarafından oluşturulan ileti tanımlayıcısı, veri torbası yaratıldığında bir başlangıç değerine ayarlanır.

PCF komut tipi, seçici MQIASY_TYPE ile sistem öğesinden alınır. Veri çantanızı oluşturduğunuzda, bu öğenin başlangıç değeri, oluşturduğunuz torba tipine bağlı olarak ayarlanır:

Çizelge 4. PCF komut tipi	
Çanta tipi	MQIASY_TYPE öğesinin ilk değeri
MQCBO_ADMIN_BAG	MQCFT_COMMAND
MQCBO_KOMUT_BAG	MQCFT_COMMAND
MQCBO_*	MQCFT_USER

When the MQAI generates a message descriptor, the values used in the **Format** and **MsgType** parameters depend on the value of the system item with selector MQIASY_TYPE as shown in [Çizelge 4 sayfa 40](#).

Çizelge 5. MQMD 'nin biçimi ve MsgType değıştirgeleri		
PCF komut tipi	Biçim	MsgTipi
MQCFT_COMMAND	MQFMT_ADMIN	MQMT_REQUEST
MQCFT_REPORT	MQFMT_ADMIN	MQMT_REPORT
MQCFT_RESAYET	MQFMT_ADMIN	MQMT_REPLY
MQCFT_TRACE_ROUTE	MQFMT_ADMIN	MQMT_DATAGRAM
MQCFT_EVENT	MQFMT_OLAY	MQMT_DATAGRAM
MQCFT_*	MQFMT_PCF	MQMT_DATAGRAM

Çizelge 5 sayfa 40 , bir denetim çantası ya da komut çantası oluşturursanız, ileti tanımlayıcısının *Format* 'inin MQFMT_ADMIN olduğunu ve *MsgType* ' in MQMT_REQUEST olduğunu gösterir. Bu, bir yanıt beklenirken komut sunucusuna bir PCF istek iletisi gönderilmesine uygundur.

İleti tanımlayıcısındaki diğler parametreler, Çizelge 6 sayfa 40içinde gösterilen değerleri alır.

Çizelge 6. İleti tanımlayıcı değeri	
Parametre	Değer
<i>StrucId</i>	MQMD_STRUC_ID
<i>Version</i>	MQMD_VERSION_1
<i>Report</i>	MQRO_NONE
<i>MsgType</i>	bkz. Çizelge 5 sayfa 40
<i>Expiry</i>	30 saniye (not “1” sayfa 41)
<i>Feedback</i>	MQFB_YOK

Çizelge 6. İleti tanımlayıcı değerleri (devamı var)

Parametre	Değer
Encoding	MQENC_NATIVE
CodedCharSetId	Torba CCSID değerine bağlıdır (not “2” sayfa 41)
Format	bkz. Çizelge 5 sayfa 40
Priority	MQPRI_PRIORITY_AS_Q_DEF
Persistence	MQPER_NOT_PERSISTENT
MsgId	MQMI_NONE
CorrelId	MQCI_NONE
BackoutCount	0
ReplyToQ	bkz. not “3” sayfa 41
ReplyToQMgr	boş

Notlar:

1. This value can be overridden on the mqExecute call by using the **OptionsBag** parameter. Bu konuda bilgi almak için bkz. [mqExecute](#).
2. Bkz. “MQAI ' de veri dönüştürme işlemi” sayfa 38.
3. MQMT_REQUEST tipinde iletiler için, kullanıcı tarafından belirtilen yanıt kuyruğunun ya da MQAI tarafından oluşturulan geçici dinamik kuyruğun adı. Aksi takdirde boştur.

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_SELECTOR, &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 hdl*/
             &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 nesneleri 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(MQCB0_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 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(&Conn, &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

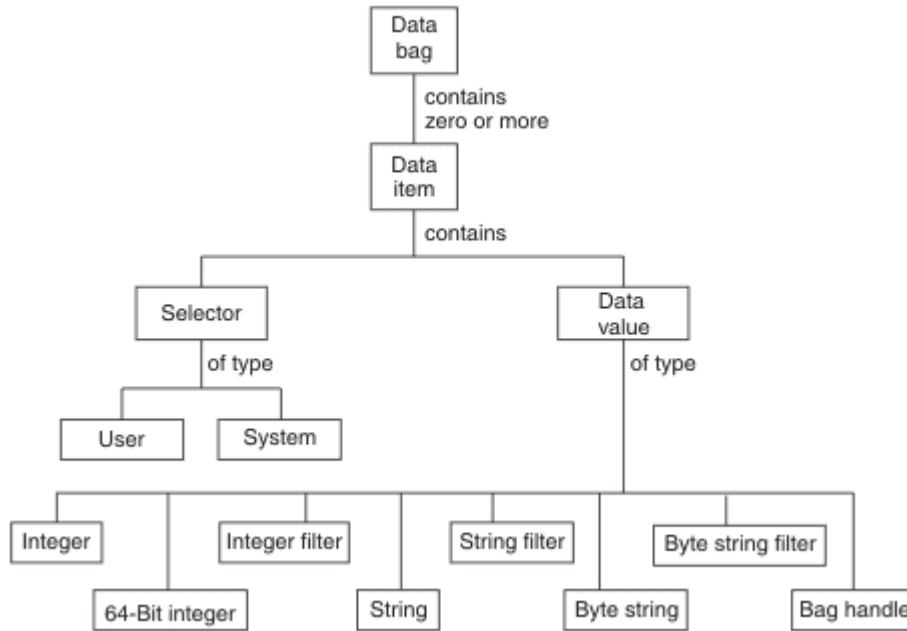
Veri torbası, IBM MQ Denetim Arabirimi 'ni (MQAI) kullanarak nesnelerin özelliklerini ya da değıştirgelerini işlemek için kullanılan bir araçtır.

Veri Çantaları

- Veri torbası sıfır ya da daha fazla veri öğesi içerir. Bu veri öğeleri, torbaya yerleştirildikleri için çantada sipariş edilir. Bu, *ekleme sırası* olarak adlandırılır. Each data item contains a *seçici* that identifies the data item and a *değer* of that data item that can be either an integer, a 64-bit integer, an integer filter, a string, a string filter, a byte string, a byte string filter, or a handle of another bag. Veri öğeleri, “MQAI içinde kullanılacak veri öğesi tipleri” sayfa 63’indeki ayrıntılarda açıklanır.

İki tip seçici vardır; *kullanıcı seçicileri* ve *sistem seçicileri*. Bunlar MQAI Selectors’inde anlatılır. Seçiciler genellikle benzersizdir, ancak aynı seçici için birden çok değere sahip olmak mümkündür. Bu durumda, bir *dizin*, gerekli olan seçiciye ilişkin belirli bir oluşumu belirtir. Dizinler “MQAI içinde dizin oluşturma” sayfa 37’inde açıklanmıştır.

Bu kavramların bir sıradüzeni Şekil 1’de gösterilir.



Şekil 4. MQAI kavramlarının sıradüzeni

Sıradüzen, önceki paragrafta açıklanmıştır.

Veri torbası tipleri

Gerçekleştirmek istediğiniz göreve bağlı olarak yaratmak istediğiniz veri torbasının 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 çanta. Yönetim çantası, "[Veri torbalarının yaratılması ve silinmesi](#)" sayfa 62 içinde açıkladığı gibi, belirli seçenekleri otomatik olarak belirtir.

komut çantası

IBM MQ nesnelerini yönetmeye ilişkin komutlar için de bir çanta oluşturulur. Ancak, denetim çantasının tersine, bu seçenekler kullanılabilir olsa da, komut çantası otomatik olarak belirli seçenekleri belirtmez. Seçeneklerle ilgili daha fazla bilgi için bkz. "[Veri torbalarının yaratılması ve silinmesi](#)" sayfa 62.

grup çantası

Gruplanmış veri öğeleri kümesini tutmak için kullanılan bir çanta. Grup çantaları, IBM MQ nesnelerini yönetmek için kullanılamaz.

Buna ek olarak, komut sunucusundan yanıt iletileri döndürüldüğünde ve bir kullanıcının çıkış torbasına yerleştirildiğinde, MQAI tarafından **sistem çantası** yaratılır. Kullanıcı tarafından bir sistem çantası değiştirilemez.

Veri çantalarını kullanarak veri torbalarını kullanmanın farklı yolları bu konuda yer alıyor:

Veri Torbalarının Kullanılması

Veri torbalarını kullanmanın farklı yolları aşağıdaki listede gösterilmiştir:

- You can create and delete data bags "[Veri torbalarının yaratılması ve silinmesi](#)" sayfa 62.
- You can send data between applications using data bags "[MQAI kullanılarak veri torbalarının alınması ve alınması](#)" sayfa 63.
- Veri çantalarına "[Paketlere MQAI ile veri öğeleri eklenmesi](#)" sayfa 64 veri öğeleri ekleyebilirsiniz.
- "[Bir çantaya sorgu komutu eklenmesi](#)" sayfa 65 veri torbası içine bir sorgu komutu ekleyebilirsiniz.
- You can inquire within data bags "[Veri torbaları araştırılıyor](#)" sayfa 65.
- Veri öğelerini bir veri torbası "[Veri öğelerini sayma](#)" sayfa 68 içinde sayabilirsiniz.
- Bir veri torbası "[Bir çantadaki bilgilerin değiştirilmesi](#)" sayfa 66 içindeki bilgileri değiştirebilirsiniz.
- Bir veri torbasını temizleyebilirsiniz "[mqClearBag çağrısını kullanarak bir torbayı temizleme](#)" sayfa 67.
- "[mqTruncateBag çağrısını kullanarak bir torbayı kesme](#)" sayfa 67 veri torbasını kesebilirsiniz.
- Torbaları ve arabellekleri ("[Çantalar ve arabellekler dönüştürülüyor](#)" sayfa 67) dönüştürebilirsiniz.

Multi Veri torbalarının yaratılması ve silinmesi

Veri torbaları oluşturma

MQAI olanağını kullanmak için, önce mqCreateBag çağrısını kullanarak bir veri torbası yaratın. Bu aramaya giriş olarak, çantanın oluşturulmasını denetlemek için bir ya da daha fazla seçenek sağlandınız.

MQCreateBag çağrısının **Options** parametresi, bir kullanıcı çantası, bir komut çantası, bir grup çantası ya da bir yönetim çantasının yaratılıp yaratılmayacağını seçmenize olanak sağlar.

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

- Bir çantada aynı seçiciye bitişik iki ya da daha çok bitişik yineleme olduğunda liste formunu kullanın.

- Parametrelerin doğru sırada olduğundan emin olmak için, veri öğelerini bir PCF iletisine eklendikçe yeniden sıralayın. Veri öğeleri hakkında daha fazla bilgi için bkz. [“MQAI içinde kullanılabilir veri öğesi tipleri” sayfa 63.](#)
- Çantaya eklediğiniz öğeler için kullanıcı seçicilerinin değerlerini denetleyin.

Yönetim çantaları otomatik olarak bu seçenekleri belirtir.

Bir veri torbası, tanıtıcısı ile tanımlanır. Çanta tanıtıcısı mqCreateBag 'dan döndürülür ve veri torbasını kullanan diğer tüm çağrılarda sağlanmalıdır.

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

Veri torbalarının silinmesi

Kullanıcı tarafından oluşturulan veri torbaları, mqDeleteBag çağrısı kullanılarak da silinmelidir. Örneğin, kullanıcı kodunda bir çanta oluşturulduysa, bu paket kullanıcı kodunda da silinmelidir.

Sistem çantaları, MQAI tarafından otomatik olarak yaratılır ve silinir. Bu konuda daha fazla bilgi için bkz. [“Sending administration commands to the qm command server using the mqExecute call” sayfa 69.](#)

Kullanıcı kodu bir sistem torbasını silemiyor.

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

Multi MQAI kullanılarak veri torbalarının alınması ve alınması

Data can also be sent between applications by putting and getting data bags using the mqPutBag and mqGetBag calls. Bu, IBM MQ Denetim Arabirimi 'nin (MQAI) uygulama yerine arabelleği işleyebilmesini sağlar.

mqPutBag çağrısı, belirtilen çantanın içindekileri bir PCF iletisine dönüştürür ve iletiyi belirtilen kuyruğa gönderir ve mqGetBag çağrısı, iletiyi belirtilen kuyruktan kaldırır ve veri torbasına geri çevirir. Therefore, the mqPutBag call is the equivalent of the mqBagToBuffer call followed by MQPUT, and the mqGetBag is the equivalent of the MQGET call followed by mqBufferToBag.

Belirli bir kuyrukta PCF iletilerinin gönderilmesine ve alınmasına ilişkin ek bilgi için [“PCF iletilerinin belirlenen bir kuyrukta gönderilmesi ve alınması” sayfa 26](#) başlıklı konuya bakın.

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

Multi MQAI içinde kullanılabilir veri öğesi tipleri

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

Bu kullanıcı öğeleri, denetlenmekte olan nesnelere öznelikleri gibi kullanıcı verilerini içerir. Oluşturulan iletiler üzerinde daha fazla denetim için sistem öğeleri kullanılmalıdır: örneğin, ileti üstbilgilerinin oluşturulması. Sistem öğeleri hakkında daha fazla bilgi için bkz. [“Sistem öğeleri ve MQAI” sayfa 64.](#)

Veri Öğesi Tipleri

Bir veri torbası oluşturduğunuzda, veri torbasını tamsayı ya da karakter dizilimi olarak yerleştirebilirsiniz. Üç tip öğe hakkında bilgi edinebilirsiniz.

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

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

- Dizgi süzgeci
- Bayt dizilimi
- Bayt dizilimi süzgeci
- Çanta tutamacı

Veri Öğelerini Kullanma

Bu, veri öğelerini kullanmanın aşağıdaki yollarıdır:

- “Veri öğelerini sayma” sayfa 68.
- “Veri öğelerinin silinmesi” sayfa 68.
- “Paketlere MQAI ile veri öğeleri eklenmesi” sayfa 64.
- “Veri öğelerini süzme ve sorgulama” sayfa 65.

Multi

Sistem öğeleri ve MQAI

Sistem öğeleri, aşağıdakiler için IBM MQ Denetim 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 dizgisiyle ilgili karakter kümesi tanıtıcısını işler.

Tüm veri öğeleri gibi, sistem öğeleri bir seçiciye ve bir değere sahip olur. Bu seçiciler ve bunların ne için oldukları hakkında bilgi için bkz. [MQAI Selectors](#).

Sistem öğeleri benzersizdir. Bir ya da daha çok sistem öğesi bir sistem seçiciyle tanımlanabilir. Her sistem seçicisinin tek bir örneği vardır.

Çoğu sistem öğesi değiştirilebilir (bkz. “Bir çantadaki bilgilerin değiştirilmesi” sayfa 66), ancak çanta-yaratma seçenekleri kullanıcı tarafından değiştirilemez. Sistem öğelerini silemezsiniz. (Bkz. “Veri öğelerinin silinmesi” sayfa 68.)

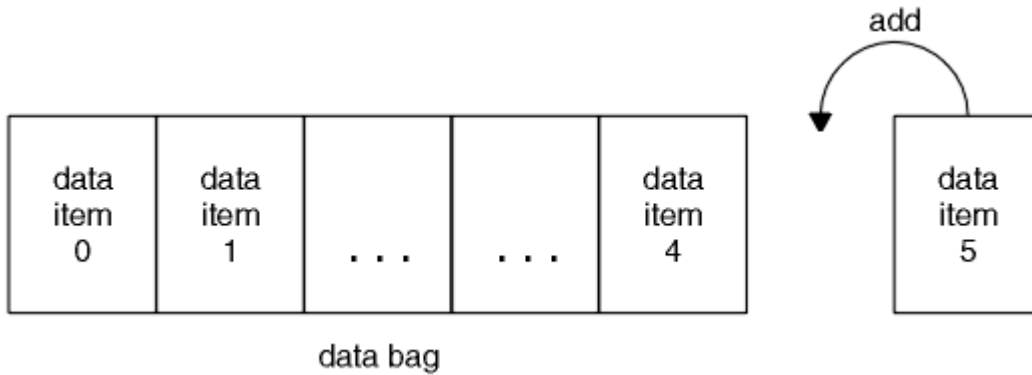
Multi

Paketlere MQAI ile veri öğeleri eklenmesi

When a data bag is created using the IBM MQ Administration Interface (MQAI), you can populate it with data items. Bu veri öğeleri, kullanıcı ya da sistem öğeleri olabilir.

Veri öğeleri hakkında daha fazla bilgi için bkz. “MQAI içinde kullanılacak veri öğesi tipleri” sayfa 63.

MQAI, tam sayı öğeleri, 64 bit tamsayı öğeleri, tamsayı süzgeç öğeleri, karakter dizilimi öğeleri, dizgi süzgeci, bayt dizgi öğeleri ve byte dizgi süzgeç öğelerini paketlere eklemenizi sağlar ve bu, [Şekil 5 sayfa 64](#) içinde gösterilir. Öğeler bir seçiciyle tanımlanır. Genellikle bir seçici yalnızca tek bir öğeyi tanımlar, ancak her zaman bu durumda değildir. Belirtilen seçiciye sahip bir veri öğesi torbada zaten mevcutsa, bu seçicinin bir ek eşgörünümü, çantanın sonuna eklenir.



Şekil 5. Veri öğeleri eklenmesi

mqAdd* çağrılarını kullanarak bir torbaya veri öğeleri ekleme:

- To add integer items, use the `mqAddInteger` call as described in [mqAddTamsayı](#)
- 64 bit tamsayı öğeleri eklemek için, [mqAddInteger64](#) içinde açıklandığı gibi `mqAddInteger64` çağrısını kullanın.
- Tamsayı süzgeci öğeleri eklemek için, [mqAddIntegerFilter](#) içinde açıklandığı gibi `mqAddIntegerFilter` çağrısını kullanın.
- To add character-string items, use the `mqAddString` call as described in [mqAddDizesi](#)
- To add string filter items, use the `mqAddStringFilter` call as described in [mqAddStringFilter](#)
- Bayt dizgi öğeleri eklemek için, [mqAddByteString](#) içinde açıklandığı gibi `mqAddByteString` çağrısını kullanın.
- To add byte string filter items, use the `mqAddByteStringFilter` call as described in [mqAddByteStringSüzgeci](#)

Bir torbaya veri öğeleri eklenmesine ilişkin daha fazla bilgi için bkz. "[Sistem öğeleri ve MQAI](#)" sayfa 64.

Multi

Bir çantaya sorgu komutu eklenmesi

The `mqAddInquiry` call is used to add an inquiry command to a bag. Arama, yönetim amaçlı olarak kullanılmak üzere özellikle yönetim paketleriyle kullanılabilir. It lets you specify the selectors of attributes on which you want to inquire from IBM MQ.

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

Multi

Veri öğelerini süzme ve sorgulama

MQAI olanağını kullanırken, IBM MQ nesnelere ilişkin özelliklerini sorgulamak için, programınıza döndürülen verileri iki şekilde denetleyebilirsiniz.

- `mqAddInteger` ve `mqAddString` çağrıları kullanılarak döndürülen verileri **süzgeç** ' e yapabilirsiniz. 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, sizin için sorun işlediğiniz nesnenin (bu durumda, bir kuyruktaki) özellikleriyle eşleşmesi gerektiğini belirtir.

Süzülebilecek diğer özellikler, "IBM MQ Programlanabilir Komut Biçimlerine Giriş" sayfa 23 içinde bulunabilecek PCF sorgulamaya * komutlarına karşılık gelir. Örneğin, bir kanalın özelliklerini sorgulamak için, bu ürün belgelerindeki Sorgulama Kanalı komutuna bakın. "Gerekli parametreler" ve Sorgulama Kanalı komutunun "İsteğe bağlı parametreleri", süzgeç uygulamak için kullanabileceğiniz seçicileri tanımlar.

- `mqAddSorgu` çağrısını kullanarak bir nesnenin **sorgu** belirli özelliklerine sahip olabilirsiniz. Bu, ilgilendiğiniz seçiciyi belirtir. Seçiciyi belirtmezseniz, nesnenin tüm özellikleri döndürülür.

Aşağıda, bir kuyruğun özelliklerini süzmek ve sorgulamak için bir örnek vardır:

```
/* 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 araştırılıyor

Aşağıdaki bilgileri araştırabilirsiniz:

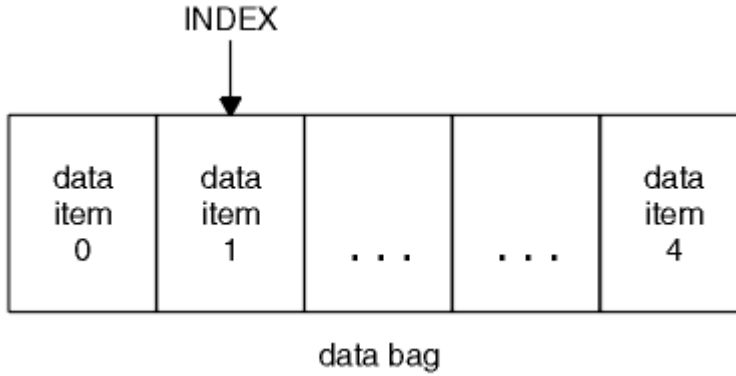
- The value of an integer item using the mqInquireInteger call. Bkz. [mqInquireInteger](#).
- mqInquireInteger64 çağrısını kullanan 64 bitlik bir tamsayı ögesinin değeri. Bkz. [mqInquireInteger64](#).
- mqInquireIntegerFilter çağrısını kullanan bir tamsayı süzgeci ögesinin değeri. Bkz. [mqInquireIntegerFilter](#).
- mqInquireDizgi çağrısını kullanan bir karakter dizilimi ögesinin değeri. Bkz. [mqInquireDizgi](#).
- mqInquireStringFilter çağrısını kullanan bir dizgi süzgeci ögesinin değeri. Bkz. [mqInquireStringFilter](#).
- mqInquireByteString çağrısını kullanan bir byte dizgi ögesinin değeri. Bkz. [mqInquireByteString](#).
- mqInquireByteStringFilter çağrısını kullanan bir byte dizgi süzgeci ögesinin değeri. Bkz. [mqInquireByteStringFilter](#).
- mqInquireBag çağrısını kullanarak bir torba tutamacının değeri. Bkz. [mqInquireBag](#).

mqInquireItemInfo çağrısını kullanarak, belirli bir ögenin tip (tamsayı, 64-bit tamsayı, tamsayı süzgeci, karakter dizgisi, dizgi süzgeci, bayt dizilimi, bayt dizgi süzgeci ya da çanta tanıtıcısı) ile ilgili olarak da bilgi edinebilirsiniz. Bkz. [mqInquireItemInfo](#).

Multi Bir çantadaki bilgilerin değiştirilmesi

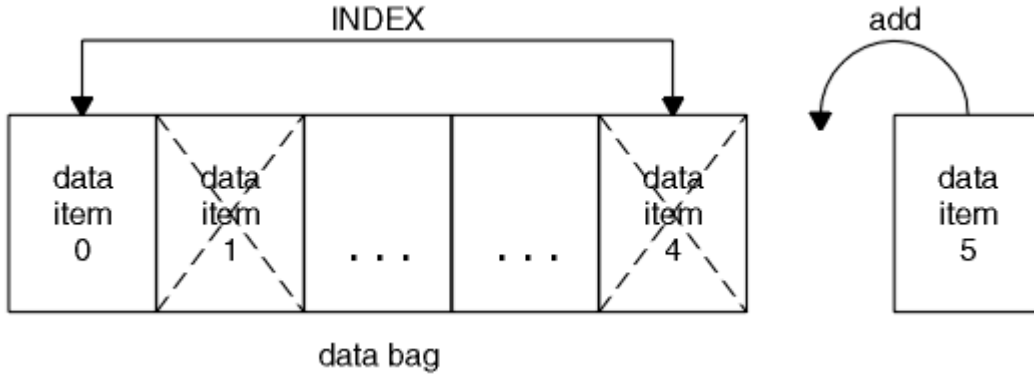
MQAI, bir çanta içindeki bilgileri mqSet * çağrılarını kullanarak değiştirmenize olanak sağlar. Şunları yapabilirsiniz:

1. Bir çantadaki veri öğelerini değiştirin. Dizin, değiştirilecek ögenin oluşumunun belirlenerek bir parametrenin tek tek bir eşgörünümünün değiştirilmesine olanak sağlar (bkz. [Şekil 6 sayfa 66](#)).



Şekil 6. Tek bir veri ögesinin değiştirilmesi

2. Belirtilen seçicinin var olan tüm oluşumlarını silin ve çantanın sonuna yeni bir geçiş ekleyin. (Bkz. [Şekil 7 sayfa 66](#).) Özel bir dizin değeri, bir değiştirgenin **Tümü** yönetim ortamlarının değiştirilmesine olanak sağlar.



Şekil 7. Tüm veri öğelerini değiştirme

Not: Dizin, çantanın içinde ekleme sırasını korur, ancak diğer veri öğelerinin dizinlerini etkileyebilir.

mqSetInteger çağrısı, bir çanta içindeki tamsayı öğelerini değiştirmenize olanak sağlar. mqSetInteger64 çağrısı, 64 bitlik tamsayı öğelerini değiştirmenize olanak sağlar. mqSetIntegerFilter çağrısı, tamsayı

süzgeci öğelerini değiştirmenize olanak sağlar. `mqSetString` çağrısı, karakter dizilimi öğelerini değiştirmenize olanak sağlar. `mqSetStringFilter` çağrısı, dizgi süzgeci öğelerini değiştirmenize olanak sağlar. `mqSetByteString` çağrısı byte dizgi öğelerini değiştirmenize olanak sağlar. `mqSetByteStringFilter` çağrısı, byte dizgisi süzgeç öğelerini değiştirmenize olanak sağlar. Diğer bir seçenek olarak, bu çağrıları, belirtilen seçicinin varolan tüm oluşumlarını silmek ve çantanın sonuna yeni bir geçiş eklemek için kullanabilirsiniz. Veri öğesi bir kullanıcı öğesi ya da bir sistem öğesi olabilir.

Bu aramaların tam açıklaması için bkz.

- [mqSetInteger](#)
- [mqSetInteger64](#)
- [mqSetIntegerFilter](#)
- [mqSetString](#)
- [mqSetStringFilter](#)
- [mqSetByteString](#)
- [mqSetByteStringFilter](#)

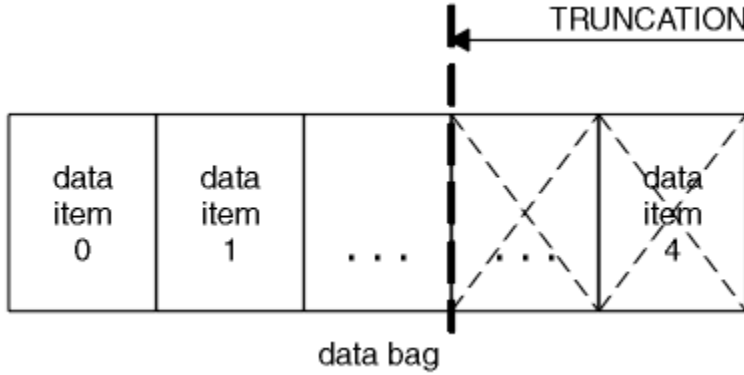
Multi `mqClearBag` çağrısını kullanarak bir torbayı temizleme

`mqClearBag` çağrısı, bir kullanıcı çantasından tüm kullanıcı öğelerini kaldırır ve sistem öğelerini ilk değerlerine döndürür. Çantanın içinde bulunan sistem çantaları da silinir.

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

Multi `mqTruncateBag` çağrısını kullanarak bir torbayı kesme

`mqTruncateBag` çağrısı, bir kullanıcı torbasındaki kullanıcı öğelerinin sayısını, en son eklenen öğeyle başlayarak, çantanın sonundaki öğeleri silerek azaltıyor. Örneğin, birden fazla ileti oluşturmak için aynı üstbilgi bilgileri kullanıldığında bu kullanılabilir.

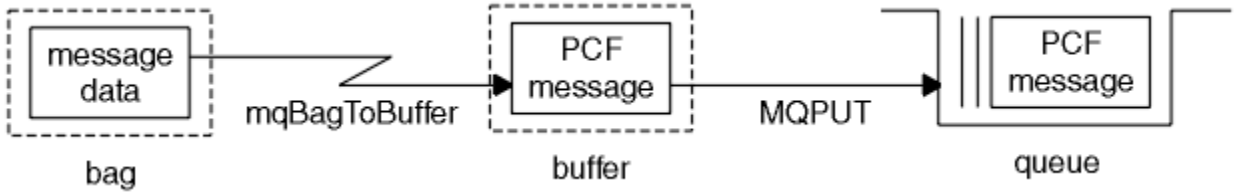


Şekil 8. Çantanın kesilmesi

`mqTruncateBag` çağrısının tam tanımı için [mqTruncateBag](#) başlıklı konuya bakın.

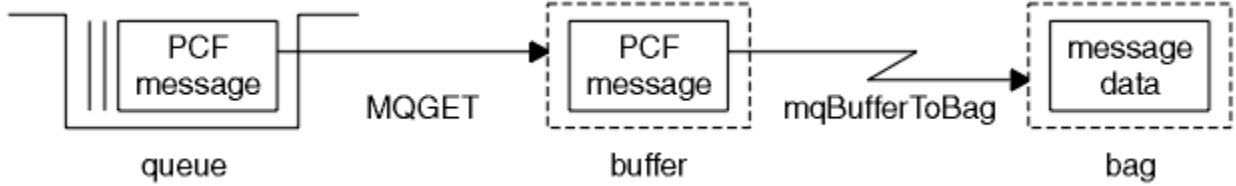
Multi Çantalar ve arabellekler dönüştürülüyor

Uygulamalar arasında veri göndermek için, ilk olarak ileti verileri bir torbaya yerleştirilir. Daha sonra, paketteki veriler `mqBagToBuffer` çağrısını kullanarak bir PCF iletisine dönüştürülür. PCF iletisi, `MQPUT` çağrısını kullanarak gerekli kuyruğa gönderilir. Bu, Şekil 9 sayfa 68’inde gösterilir. `mqBagToBuffer` çağrısının tam açıklaması için bkz. [mqBagToBuffer](#).



Şekil 9. Çantaların PCF iletilerine dönüştürülmesi

Verileri almak için, ileti MQGET çağrısını kullanarak bir arabelleğe alınır. Arabelleğindeki veriler, mqBufferToBag çağrısı kullanılarak bir torbaya dönüştürülmüş ve arabelleğin geçerli bir PCF iletisi içermesini sağlar. Bu, Şekil Şekil 10 sayfa 68’inde gösterilir. mqBufferToBag çağrısının tam tanımı için mqBufferToBag başlıklı konuya bakın.



Şekil 10. PCF iletileri torba forma dönüştürülüyor

Multi Veri öğelerini sayma

mqCountItems çağrısı, bir veri torbasında saklanan kullanıcı öğelerinin, sistem öğelerinin ya da her ikisinin sayısını sayar ve bu sayıyı döndürür. Örneğin, mqCountItems (Bag, 7, . . .), çantadaki öğelerin sayısını 7 ayırıcı ile döndürür. Öğeleri tek tek seçiciyle, kullanıcı seçicileri temelinde, sistem seçicileri temelinde ya da tüm seçiciler tarafından sayabilirler.

Not: Bu çağrı, çantadaki benzersiz seçicilerin sayısını değil, veri öğelerinin sayısını sayar. Bir seçici birden çok kez gerçekleşebilir, bu nedenle çantanın içinde veri öğelerinde daha az sayıda benzersiz seçici olabilir.

MqCountItems çağrısının tam açıklaması için bkz. [mqCountItems](#).

Multi Veri öğelerinin silinmesi

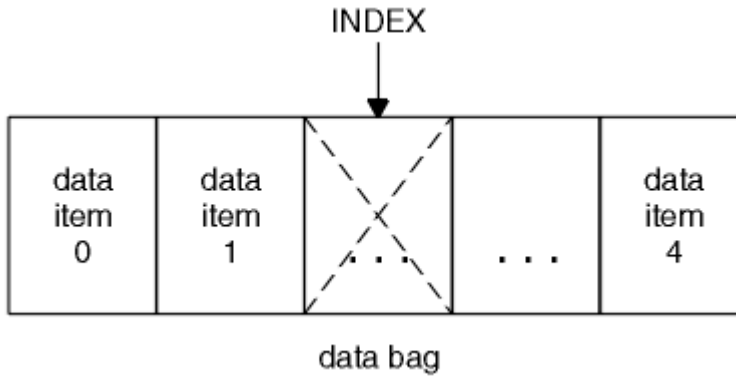
Çantalardaki öğeleri çeşitli şekillerde silebilirsiniz. Şunları yapabilirsiniz:

- Bir çantadan bir ya da daha fazla kullanıcı öğesini kaldırın. Ayrıntılı bilgi için bkz. “[MqDeleteItem](#) çağrısını kullanarak bir torbadan veri öğeleri silme” sayfa 68.
- Bir çantadan tüm kullanıcı öğelerini silin, yani, bir çanta temizliden. Ayrıntılı bilgi için bkz. “[mqClearBag](#) çağrısını kullanarak bir torbayı temizleme” sayfa 67.
- Bir çantanın sonundaki kullanıcı öğelerini silin, bu, bir poşet kısaltır. Ayrıntılı bilgi için bkz. “[mqTruncateBag](#) çağrısını kullanarak bir torbayı kesme” sayfa 67.

Multi MqDeleteItem çağrısını kullanarak bir torbadan veri öğeleri silme

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

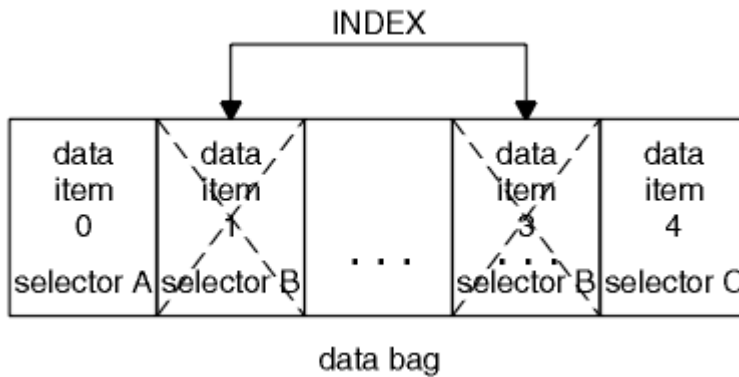
1. Belirtilen seçiciye ilişkin tek bir oluşum. (Bkz. [Şekil 11 sayfa 69](#).)



Şekil 11. Tek bir veri öğesinin silinmesi

ya da

2. Belirtilen seçicinin tüm oluşumları. (Bkz. Şekil 12 sayfa 69.)



Şekil 12. Tüm veri öğeleri siliniyor

Not: Dizin, çantanın içinde ekleme sırasını korur, ancak diğer veri öğelerinin dizinlerini etkileyebilir. Örneğin, mqDeleteItem çağrısı silinen öğede kalan boşluğu doldurmak için yeniden düzenlendiğinden, silinen öğeyi izleyen veri öğelerinin dizin değerlerini korumaz.

MqDeleteItem çağrısının tam açıklaması için bkz. [mqDeleteItem](#).

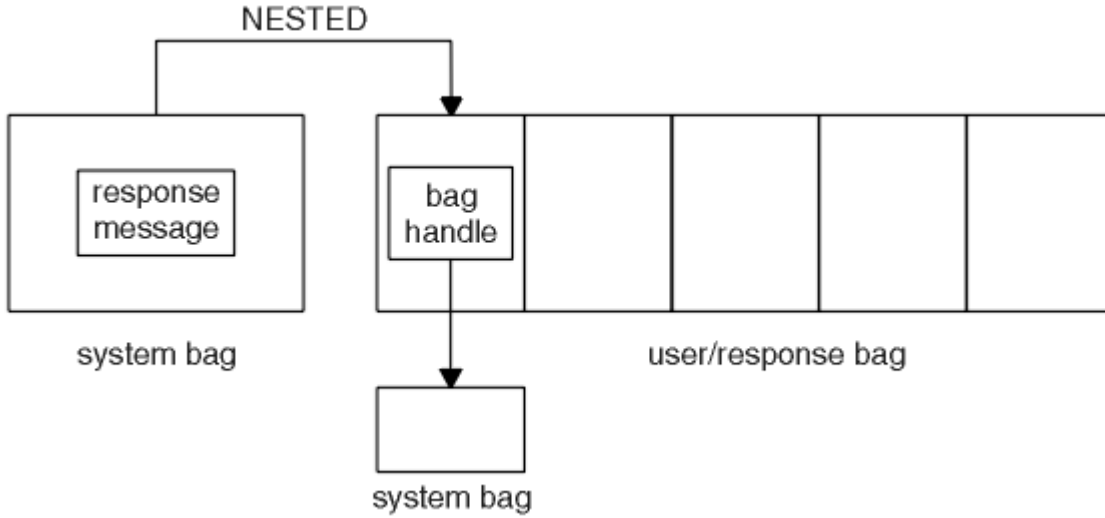
Multi Sending administration commands to the qm command server using the mqExecute call

Bir veri torbası yaratılıp doldurulduğunda, mqExecute çağrısını kullanarak kuyruk yöneticisinin komut sunucusuna bir yönetim komutu iletisi gönderilebilir. Bu, komut sunucusu ile değişimi işler ve bir çantadaki yanıtları döndürür.

Veri çantanızı yarattıktan ve doldurduktan sonra, bir kuyruk yöneticisinin komut sunucusuna bir denetim komutu iletisi gönderebilirsiniz. Bunu yapmanın en kolay yolu mqExecute çağrısını kullanmandır. mqExecute çağrısı, kalıcı olmayan bir ileti olarak bir denetim komutu iletisi gönderir ve herhangi bir yanıt bekler. Yanıt torbasında yanıtlar döndürülür. Bu bilgiler, birkaç IBM MQ nesnelere ya da bir dizi PCF hata yanıtı iletisine ilişkin özniteliklerle ilgili bilgileri içerebilir. Bu nedenle, yanıt çantası yalnızca bir dönüş kodu içerebilir ya da iç içe geçmiş değerini içerebilir.

Yanıt iletileri, sistem tarafından oluşturulan sistem çantalarına yerleştirilir. Örneğin, nesne adlarıyla ilgili sorgular için, bu nesne adlarını tutmak için bir sistem çantası oluşturulur ve çanta, kullanıcı çantasına eklenir. Bu çantalara verilen tutamaçlar, yanıt torbasına eklenir ve iç içe yerleşimli torbaya, MQHA_BAG_HANDLE seçici tarafından erişilebilir. Sistem çantası, yanıt paketi silininceye kadar silinmezse, depolama alanında kalır.

Şekil 13 sayfa 70 içinde iç içe yerleştirme kavramı gösterilir.



Şekil 13. İç içe yerleştirme

mqExecute çağrısına giriş olarak, aşağıdaki bilgileri sağlamanız gerekir:

- Bir MQI bağlantı tanıtıcısı.
- Yürütülecek komut. Bu, MQCMD_ * değerlerinden biri olmalıdır.
Not: Bu değer MQAI tarafından tanınmadıysa, değer kabul edilir. Ancak, çantaya değer eklemek için mqAddSorgu çağrısı kullanıldıysa, bu parametre, MQAI tarafından tanınan bir SORGUL KOMUTU olmalıdır. Yani, parametrenin MQCMD_INQUIRE_ * biçiminde olması gerekir.
- İsteğe bağlı olarak, aramanın işlenmesini denetleyen seçeneklerin bulunduğu bir torba tanıtıcısı. Bu, MQAI ' in her yanıt iletisi için beklemesi gereken süreyi (milisaniye olarak) belirtebileceğiniz bir yerdir.
- Çıkışı verilecek denetim komutunun ayrıntılarını içeren denetim torbasının tanıtıcısı.
- Yanıt iletilerini alan yanıt torbasının tanıtıcısı.

Aşağıdaki tutamaçlar isteğe bağlıdır:

- Denetim komutunun yerleştirileceği kuyruğun nesne tanıtıcısı.
Herhangi bir nesne tanıtıcısı belirlenmezse, denetim komutu SYSTEM.ADMIN.COMMAND.QUEUE . Bu varsayılandır.
- Yanıt iletilerinin yerleştirileceği kuyruğun bir nesne tanıtıcısı.
Yanıt iletilerini, MQAI tarafından otomatik olarak yaratılmış bir dinamik kuyruğa yerleştirmeyi seçebilirsiniz. Yaratılan kuyruk yalnızca çağrı süresi için var ve mqExecute çağrısından çıkışta MQAI tarafından silinmiş.

Örnekler için mqExecute çağrısının kullanımları için [Örnek kodu](#) başlıklı konuya bakın.

REST API komutunu kullanarak yönetim

Kuyruk yöneticileri ve kuyruklar ve Managed File Transfer araçları ve aktarımları gibi IBM MQ nesnelerini yönetmek için administrative REST API ' i kullanabilirsiniz. Bilgi, JSON biçiminde administrative REST API ' den gönderilir ve bu biçimden alınır. These RESTful APIs can help you to embed IBM MQ administration into popular DevOps and automation tooling.

Başlamadan önce

Kullanılabilir REST kaynaklarıyla ilgili başvuru bilgileri için bkz. [administrative REST API başvurusu](#).

Yordam

- [“administrative REST API ile çalışmaya başlama” sayfa 71](#)
- [“administrative REST API komutunu kullanma” sayfa 74](#)
- [“REST API komutunu kullanarak uzaktan yönetim” sayfa 76](#)
- [“REST API zaman damgaları” sayfa 80](#)
- [“REST API Hata işleme” sayfa 80](#)
- [“REST API Keşif” sayfa 82](#)
- [“REST API ulusal dil desteği” sayfa 84](#)

administrative REST API ile çalışmaya başlama

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

Başlamadan önce

administrative REST API komutunu kullanmaya başlamanız için, bu görevdeki örneklerin aşağıdaki gereksinimleri vardır:

- Bu örnekler, REST isteklerinin sistemdeki kuyruk yöneticilerine ilişkin bilgileri görüntülemesini ve kuyruk yaratılmasını, kuyruğun güncellenmesi, görüntülenmesi ve silinmesi için cURL 'yi kullanır. Bu nedenle, bu görevi tamamlamak için sisteminizde kurulu olan cURL 'ye gereksinim duyarsınız.
- Bu görevi tamamlamak için, **dspmweb** komutunu kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

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

IBM i IBM üzerinde, komutların QSHELL içinde çalıştırılması gerekir.

Yordam

1. mweb sunucusu administrative REST API, MFT, messaging REST API ya da IBM MQ Console için administrative REST API tarafından kullanılmak üzere yapılandırılmadıysa, mweb sunucusunu yapılandırın.

Temel bir kayıt dosyasıyla mweb sunucusu için temel yapılandırma oluşturma hakkında daha fazla bilgi için bkz. [mweb sunucusu için temel yapılandırma](#).

2. **z/OS** 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. Try out a GET request on the `qmgr` resource by using basic authentication with the `mqadmin` user:

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

5. `mqsc` kaynağını kullanarak bir kuyruğu oluşturun, görüntüleyin, değiştirin ve silin:

Bu örnek, `QM1` kuyruk yöneticisini kullanır. Aynı adı taşıyan bir kuyruk yöneticisi yaratın ya da sisteminizde varolan bir kuyruk yöneticisini değiştirin.

- a) Yerel kuyruğu yaratmak için `mqsc` kaynağında bir POST isteği yapın:

İstek gövdesinde, yeni kuyruğun adı `Q1` olarak ayarlanır. Basic authentication is used, and an `ibm-mq-rest-csrf-token` HTTP header with an arbitrary value is set in the cURL REST request. 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 72 adımıyla oluşturulan yerel kuyruğu görüntülemek için `mqsc` kaynağında bir POST isteği yapın:

```
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) Make a POST request on the `mqsc` resource to resource to update the description of the queue:

```
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 tanımlamasını görüntülemek için `mqsc` kaynağında bir POST isteği yapın. İstek gövdesinde **responseParameters** özneliğini belirtin; böylece yanıt, açıklama alanını içerir:

```
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 bir POST isteği yapın:

```
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) Make a POST request on the `mqsc` resource to prove that the queue is deleted:

```
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 [REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma](#) ve [REST API ile belirteç tabanlı kimlik doğrulamasını kullanmabaşlıklı](#) konuya bakın.



- Learn more about using the administrative REST API and constructing URLs with query parameters: [“administrative REST APIkomutunu kullanma” sayfa 74.](#)
- Kullanılabilir administrative REST API kaynakları ve kullanılabilir tüm isteğe bağlı sorgu parametreleri için başvuru bilgilerine göz atın: [administrative REST API başvurusu.](#)
- Uzak sistemlerdeki IBM MQ nesnelere yönetmek için administrative REST API ' yi nasıl kullanacağını öğrenin: [“REST APIkomutunu kullanarak uzaktan yönetim” sayfa 76.](#)
- administrative REST API ' i MFTile nasıl kullanacağını öğrenin: [“MFTiçin REST API ile çalışmaya başlama” sayfa 73.](#)
- IBM MQ ileti alışverişi için RESTful arabirimi olan messaging REST API' u keşfedin: [REST APIkomutunu kullanarak ileti alışverişi.](#)
- Discover the IBM MQ Console, a browser-based GUI: [“Web konsolu kullanan yönetim” sayfa 87.](#)

MFTiçin REST API ile çalışmaya başlama

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

Başlamadan önce

- Örnekler, aktarımların bir listesini görüntülemek ve MFT aracı durumunu görüntülemek için REST isteklerini göndermek için cURL ' yi kullanır. Bu nedenle, bu görevi tamamlamak için sisteminizde kurulu olan cURL ' ye gereksinim duyarsınız.
- 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.

Yordam

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

- mqweb sunucusu administrative REST API, MFT, messaging REST APIya da IBM MQ Console için administrative REST API tarafından kullanılmak üzere yapılandırılmadıysa, mqweb sunucusunu yapılandırın. Temel bir kayıt dosyasıyla mqweb sunucusu için temel yapılandırma oluşturma hakkında daha fazla bilgi için bkz. [mqweb sunucusu için temel yapılandırma.](#)
- If the mqweb server is configured, ensure that step 8 of [mqweb sunucusu için temel yapılandırma](#) was completed to enable the administrative REST API for MFT.

2. 

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* , cıtmqweb' 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:

```
dspmqweb 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. agent kaynağında, ad, tip ve durum da içinde olmak üzere tüm araçlar hakkında temel ayrıntıları döndürmek için GET isteğinde bulunun.

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

5. **fteCreateTransfer** komutunu kullanarak görüntülemek için bazı aktarımlar oluşturun.

Mqweb sunucusu aktarımlara ilişkin bilgileri önbelleğe alır ve bir istek yapıldığında bu bilgileri döndürür. Bu önbellek, mqweb sunucusu yeniden başlatıldığında sıfırlanır. You can see whether the server was restarted by viewing the console.log and messages.log files, or on z/OS, looking at the output from the started task.

6. mqweb sunucusu başlatıldığından bu yana yapılan en çok dört aktarıma ilişkin ayrıntıları döndürmek için transfer kaynağından bir GET isteği yapın:

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

Sonraki adım

- Örnekler, isteği güvenli kılmak için temel kimlik doğrulamasını kullanır. Simge tabanlı kimlik doğrulamayı ya da istemci tabanlı kimlik doğrulamayı kullanabilirsiniz. Daha fazla bilgi için bkz. [Using token-based authentication with the REST API](#) ve [REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulaması kullanılması](#).
- Learn more about using the administrative REST API and constructing URLs with query parameters: [“administrative REST API komutunu kullanma” sayfa 74](#).
- Browse the reference information for the available administrative REST API for MFT resources and all the available optional query parameters: [administrative REST API başvurusu](#).
- IBM MQ ileti alışverişi için RESTful arabirimi olan messaging REST API' u keşfedin: [REST API komutunu kullanarak ileti alışverişi](#).
- Discover the IBM MQ Console, a browser-based GUI: [“Web konsolu kullanan yönetim” sayfa 87](#).

administrative REST API komutunu kullanma

administrative REST API'ı kullandığınızda, kuyruk yöneticileri ya da kuyruklar gibi çeşitli IBM MQ nesnelere gösteren URL' lerde HTTP yöntemlerini çağırırsınız. HTTP yöntemi (POST gibi), URL ile gösterilen nesnede gerçekleştirilecek işlem tipini temsil eder. Bu işlemle ilgili daha fazla bilgi, HTTP yönteminin bilgi yükünün bir parçası olarak ya da sorgu parametrelerinde kodlanabilir olarak sağlanabilir. İşlemin gerçekleştirilmesiyle ilgili bilgi, HTTP yanıtının gövdesi olarak döndürülebilecek.

Başlamadan önce

administrative REST API' u kullanmadan önce şu şeyleri göz önünde bulundurun:

- administrative REST API' i kullanabilmek için mqweb sunucusu ile kimlik doğrulaması gerekir. HTTP temel kimlik doğrulaması, istemci sertifikası kimlik doğrulaması ya da belirteç tabanlı kimlik doğrulaması kullanarak kimliğinizi doğrulayabilirsiniz. Bu kimlik doğrulama yöntemlerinin nasıl kullanılacağı hakkında daha fazla bilgi için IBM MQ Console ve REST API security başlıklı konuya bakın.
- REST API , büyük/küçük harfe duyarlıdır. Örneğin, kuyruk yöneticisi qmgr1 olarak adlandırıldıysa, aşağıdaki URL ' deki bir HTTP GET işlemi bilgi görüntülemez.

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

- IBM MQ nesne adlarında kullanılacak karakterler, bir URL ' de doğrudan kodlanabilir değil. Bu karakterleri doğru bir şekilde kodlamak için uygun URL kodlamasını kullanmanız gerekir:

- Sağa eğik çizgi,/, %2Folarak kodlanmalıdır.
- Yüzde işareti,%, %25olarak kodlanmalıdır.
- Bazı tarayıcıların işleyişi nedeniyle, nesnelere yalnızca nokta ya da eğik çizgi karakterlerini kullanarak adlandırmayın.

Bu görev hakkında

REST API ' u bir nesne üzerinde işlem yapmak için kullandığınızda, öncelikle o nesneyi göstermek için bir URL oluşturmanız gerekir. Her URL, isteğin gönderileceği anasistem adını ve kapağı tanımlayan bir önekle başlar. URL ' nin geri kalanı, kaynak olarak bilinen belirli bir nesneyi ya da nesne kümesini açıklar.

Kaynak üzerinde gerçekleştirilecek işlem, URL ' nin sorgu parametrelerinin gerekli gerekmediğini tanımlar mı, yoksa sorgu değiştirgilerinin mi gerekeceğini tanımlar. Ayrıca, kullanılan HTTP yöntemini ve URL ' ye ek bilgilerin gönderilip gönderilmeyeceğini ya da JSON formundan döndürülüp döndürülmeyeceğini de tanımlar. Ek bilgiler, HTTP isteğinin bir parçası olabilir ya da HTTP yanıtının bir parçası olarak döndürülebilirler.

URL 'yi oluşturduktan ve HTTP isteğini göndermek için isteğe bağlı bir JSON bilgi yükü yarattıktan sonra, HTTP isteğini IBM MQ' e gönderebilirsiniz. İsteğinizin programlama diline yerleşik HTTP uygulamasını kullanarak isteği gönderebilirsiniz. You can also send the requests by using command line tools such as cURL, or a web browser, or web browser add-on.

Önemli: You must, as a minimum, carry out steps “1.a” sayfa 75 and “1.b” sayfa 75.

Yordam

1. URL ' yi oluşturun:

a) Aşağıdaki komutu girerek URL öneğini 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](#)
- [/admin/queue](#)
- [/admin/subscription](#)
- [/admin/kanal](#)
- [/action/qmgr/{qmgrname}/mqsc](#)

Aşağıdaki Managed File Transfer kaynakları kullanılabilir:

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

Örneğin, kuyruk yöneticileriyle etkileşim kurmak için, aşağıdaki URL ' yi yaratmak üzere URL öneğine /qmgr ekleyin:

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

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

In the reference information for each object type, the optional segments can be identified in the URL by the braces that surround it { }.

For example, add the queue manager name QM1 to the URL to create the following URL:

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

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

Soru işareti ekleme,?, değişken adı, eşittir işareti =, bir değer ya da değer listesi URL ' ye.

Örneğin, QM1kuyruk yöneticisinin tüm özniteliklerini istemek için aşağıdaki URL ' yi yaratın:

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

e) URL ' ye isteğe bağlı sorgu parametreleri ekleyin.

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

2. URL ' de ilgili HTTP yöntemini çağırın. Herhangi bir isteğe bağlı JSON bilgi yükü belirtin ve kimlik doğrulaması için uygun güvenlik kimlik bilgilerini sağlayın. Örneğin:

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

REST API komutunu kullanarak uzaktan yönetim

Uzak kuyruk yöneticilerini ve bu kuyruk yöneticileriyle ilişkili IBM MQ nesnelere yönetmek için REST API ' i kullanabilirsiniz. Bu uzak yönetim, aynı sistemde bulunan, ancak mqweb sunucusuyla aynı IBM MQ kuruluşunda olmayan kuyruk yöneticilerini içerir. Bu nedenle, tüm IBM MQ ağınıza, yalnızca mqweb sunucusunu çalıştıran tek bir kurulumla yönetmek için REST API ' i kullanabilirsiniz. Uzak kuyruk yöneticilerini denetlemek için, administrative REST API ağ geçidini, mqweb sunucusu ile aynı kurulumda en az bir kuyruk yöneticisinin ağ geçidi kuyruk yöneticisi olarak işlev göreceği şekilde yapılandırmanız gerekir. Then, you can specify the remote queue manager in the REST API resource URL to perform the specified administrative action.

Başlamadan önce

administrative REST API ağ geçidini geçersiz kılarak uzaktan yönetimi önleyebilirsiniz. Daha fazla bilgi için [administrative REST API ağ geçidini yapılandırma](#) başlıklı konuya bakın.

administrative REST API ağ geçidini kullanmak için aşağıdaki koşulların yerine getirilmesi gerekir:

- mqweb sunucusu yapılandırılmalı ve başlatılmalıdır. mqweb sunucusunu yapılandırma ve başlatma hakkında daha fazla bilgi için bkz. [“administrative REST API ile çalışmaya başlama”](#) sayfa 71.
- Ağ geçidi kuyruk yöneticisi olarak yapılandırmak istediğiniz kuyruk yöneticisi, mqweb sunucusuyla aynı kurulumda olmalıdır.
- Denetlemek istediğiniz uzak kuyruk yöneticisi IBM MQ 8.0 ya da sonraki bir yayın düzeyiyle olmalıdır.
- İsteğinizde belirtilen tüm özniteliklerin, isteği gönderdiğiniz sistem için geçerli olduğundan emin olmanız gerekir. Örneğin, ağ geçidi kuyruk yöneticisi Windows üzeriyse ve uzak kuyruk yöneticisi z/OS üzeriyse, queue kaynağındaki bir HTTP GET isteği için dataCollection.statistics özniteliğinin döndürülmesini istemezsiniz.
- İsteğinizde belirtilen tüm özniteliklerin, isteği gönderdiğiniz IBM MQ düzeyi için geçerli olduğundan emin olmanız gerekir. Örneğin, uzak kuyruk yöneticisi IBM MQ 8.0 çalıştırıyorsa, queue kaynağındaki bir HTTP GET isteği için extended.enableMediaImageOperations özniteliğinin döndürülmesini istemezsiniz.
- Aşağıdaki desteklenen REST kaynaklarından birini kullanmanız gerekir:

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

Bir uzak kuyruk yöneticisini sorguladığınızda /qmgr kaynağı özniteliklere ilişkin yalnızca bir alt küme 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, uzak denetim için kuyruk yöneticilerini hazırlamanız gerekir. Yani, ağ geçidi kuyruk yöneticisi ile uzak kuyruk yöneticisi arasındaki iletim kuyruklarını, dinleyicileri ve göndereni ve alıcı kanallarını yapılandırmalısınız. Bundan sonra, kaynak URL adresinde kuyruk yöneticisini belirterek, uzak kuyruk yöneticisine bir REST isteği gönderebilirsiniz. Ağ geçidi kuyruk yöneticisi, mqRestGatewayQmgr öznitelikliğini, ağ geçidi kuyruk yöneticisinin adına ayarlamak için **setmqweb** komutunu kullanarak ya da istek ile gönderilen bir üstbilgide ağ geçidi kuyruk yöneticisinin adını 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 ve uzak kuyruk yöneticileri arasındaki iletişimi yapılandırın. Bu yapılandırma adımları, runmqsc ve PCF tarafından uzak denetimi yapılandırmak için gerekli olan adımlardır.
Bu adımlarla ilgili daha fazla bilgi için bkz. [“Uzak denetim için kuyruk yöneticilerinin yapılandırılması” sayfa 181.](#)
2. Uzak kuyruk yöneticilerindeki güvenliği yapılandırın:
 - a) Uzak kuyruk yöneticisinin üzerinde çalıştığı sistemde ilgili kullanıcı kimliklerinin var olduğunu doğrulayın. Uzak sistemde var olması gereken kullanıcı kimliği, REST API kullanıcısının rolüne bağlıdır:
 - REST API kullanıcısı MQWebAdmin ya da MQWebAdminRO grubunda yer aldıysa, mqweb sunucusunu başlatan kullanıcı kimliği uzak sistemde varolmalıdır. IBM MQ Appliance üzerinde, mqweb sunucusunu başlatan kullanıcı mqsystem olur.
 - REST API kullanıcısı MQWebUser grubundaysa, bu REST API kullanıcı kimliği uzak sistemde var olmalıdır.
 - b) Uzak kuyruk yöneticisinde uygun REST API kaynaklarına erişmek için ilgili kullanıcı kimliklerinin gereken yetki düzeylerine verildiğinden emin olun:
 - SYSTEM.ADMIN.COMMAND.QUEUE' a ileti koyma yetkisi.
 - SYSTEM.REST.REPLY.QUEUE' a ileti koyma yetkisi.
 - Uzaktan yönetim için tanımlanan iletim kuyruklarına erişim yetkisi.
 - Kuyruk yöneticisi özniteliklerini görüntüleme yetkisi.
 - REST isteklerini gerçekleştirme yetkisi. Daha fazla bilgi için, [REST API kaynakları başvuru konuları](#) başlıklı konunun Güvenlik Gereksinimleri bölümüne bakın.
3. Ağ geçidi olarak hangi yerel kuyruk yöneticisinin kullanılacağını yapılandırın. Bir varsayılan ağ geçidi kuyruk yöneticisini yapılandırabilir, bir HTTP üstbilgisinde ağ geçidi kuyruk yöneticisini belirtebilir ya da her iki yaklaşımın bir birleşimini de kullanabilirsiniz:

- **setmqweb** komutunu kullanarak varsayılan ağ geçidi kuyruk yöneticisini yapılandırın:

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

Burada *qmgrName* , ağ geçidi kuyruk yöneticisinin adıdır.

Bu ağ geçidi kuyruk yöneticisi, aşağıdaki deyimler doğru olduğunda kullanılır:

- Bir REST isteğinin `ibm-mq-rest-gateway-qmgr` üstbilgisinde bir kuyruk yöneticisi belirtilmez.
- REST API kaynak URL 'sinde belirtilen kuyruk yöneticisi yerel bir kuyruk yöneticisi değil.
- `ibm-mq-rest-gateway-qmgr` HTTP üstbilgisini ağ geçidi kuyruk yöneticisi adına ayarlayarak her REST isteğinde ağ geçidi kuyruk yöneticisini yapılandırın.

4. Kaynak URL ' de denetlemek istediğiniz uzak kuyruk yöneticisinin adını ekleyin.

Örneğin, remoteQMuzak kuyruk yöneticisinden kuyrukların listesini almak için şu URL adresini kullanın:

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

Sonuçlar

REST yanıtına sahip 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 ' yi kullanmakta güçlük çekmeniz durumunda:

- Uzak kuyruk yöneticisinin çalışır durumda olup olmadığını denetleyin.
- Komut sunucusunun uzak sistemde çalışır durumda olup olmadığını denetleyin.
- Kanal bağlantı kesme aralığının süresinin dolmadığından emin olun. Örneğin, bir kanal başlatıldıysa, ancak bir süre sonra sona erdirilirse. Bu, özellikle kanalları el ile başlatabilirseniz önemlidir.

Örnek

Aşağıdaki örnekte, iki makine üzerinde üç IBM MQ kurulumu vardır. Machine 1 üzerinde, bir Installation 1 ve bir Installation 2 vardır. Machine 2 üzerinde, bir Installation 3 vardır. Installation 1 için yapılandırılmış bir mqweb sunucusu. Her kurulumda tek bir kuyruk yöneticisi vardır ve bu kuyruk yöneticileri uzaktan yönetim için yapılandırılır. Yani, aşağıdaki dinleyiciler, kanallar ve kuyruklar yapılandırılıp başlatılmış olur:

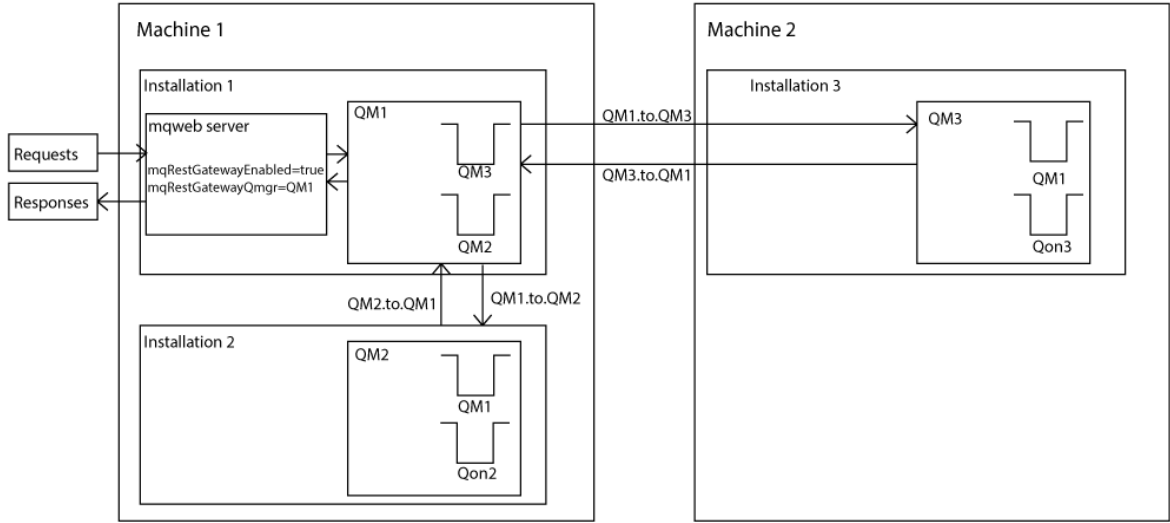
- Kuyruk yöneticisinde (QM1), Installation 1'da Machine 1'da:
 - Gönderen kanalı QM1.to.QM2
 - Günlük nesnesi kanalı QM2.to.QM1
 - Gönderen kanalı QM1.to.QM3
 - Günlük nesnesi kanalı QM3.to.QM1
 - İletim kuyruğu QM2
 - İletim kuyruğu QM3
 - 1414 kapısında yapılandırılan bir dinleyici
- Kuyruk yöneticisinde (QM2), Installation 2'da Machine 1'da:
 - Gönderen kanalı QM2.to.QM1
 - Günlük nesnesi kanalı QM1.to.QM2
 - İletim kuyruğu QM1
 - 1415 kapısında yapılandırılan bir dinleyici
- Kuyruk yöneticisi QM3, Installation 3'ta Machine 2 üzerinde:
 - Gönderen kanalı QM3.to.QM1
 - Alıcı kanalı QM1.to.QM3
 - İletim kuyruğu QM1
 - Varsayılan dinleyici

Kuyruk, Qon2 QM2 üzerinde tanımlanır ve QM3 üzerinde bir kuyruk Qon3 tanımlanır.

The user mquser is defined on both machines, is granted the MQWebAdmin role in the REST API, and is granted authority to access the appropriate queues on each queue manager.

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

Aşağıdaki çizge bu yapılandırmayı göstermektedir:



Şekil 14. Diagram of example configuration for remote administration by using the REST API.

Aşağıdaki REST isteği mqweb sunucusuna gönderilir:

```
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"
  }
]
}
```

Aşağıdaki REST isteği mqweb sunucusuna gönderilir:

```
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, UTC ' ye (Coordinated Universal Time; Eşgüdümlü Evrensel Saat) ve bir küme biçiminde döndürülür.

Tarih ve saat şu zaman damgası biçimiyle döndürülür:

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

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

Bu zaman damgasının doğruluğu garanti edilmez. Örneğin, mqweb sunucusu kaynak URL adresinde belirtilen kuyruk yöneticisiyle aynı saat diliminde başlatılmamışsa, zaman damgası doğru olmayabilir. Ayrıca, Gün Işığında Yararlanma Saati ayarlamaları gerekliyse, zaman damgası doğru olmayabilir.

REST API Hata işleme

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

Hata yanıt biçimi

Yanıt JSON biçiminde UTF-8 kodlamasında olur. İç içe geçmiş JSON nesnelerini içerir:

- `error` adlı tek bir JSON dizisi 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.

A unique identifier for the message of the form MQWBnnnnX. Bu tanıttıcı aşağıdaki öğelere sahiptir:

MQWB

İletinin IBM MQ Rest API 'sında kaynaklandığı gösteren bir önek.

nnnn

İletiyi tanımlayan benzersiz bir numara.

X

İletinin önem derecesini belirten tek bir harf:

- I bir ileti tamamen bilgilendirme amaçlı ise.
- W bir ileti bir sorunla ilgili uyarıysa.
- Bir ileti, bir hata oluştuğunu gösteriyorsa, E .
- S bir ileti önemli bir hatanın ortaya çıktığını gösteriyorsa.

ileti

Dizgi.
Hatanın açıklaması.

Açıklama

Dizgi.


Hatanın açıklaması.

işlem

Dizgi.

Hatayı çözmek için alınabilecek adımlara ilişkin bir açıklama.

qmgrName

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

Dizgi.

Hatayı deneyimleyen kuyruk yöneticisinin adı.

Bu alan messaging REST API için geçerli değildir.

completionCode

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

Numara.

Hata ile ilişkili MQ tamamlama kodu.

reasonCode

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

Numara.

Hata ile ilişkili MQ neden kodu.


Kural Dışı Durumlar

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

Dizi.

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

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

 IBM MQ 9.1.2' dan bu alan artık geri döndürülmez.

Kuyruk paylaşım gruplarıyla ilgili hatalar



In a queue sharing group, it is possible to specify an optional query parameter of **commandScope** for certain commands. Bu değıştirge, komutun kuyruk paylaşım grubundaki diğer kuyruk yöneticilerine yayılmasına olanak sağlar. Bu komutlardan herhangi biri bağımsız olarak başarısız olabilir ve kuyruk paylaşım grubu için bazı komutların başarılı olmasına ve bazı komutlara yol göstermesine neden olabilir.

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

Örnekler

- Aşağıdaki örnekte, var olmayan bir kuyruk yöneticisiyle ilgili bilgi alma girişimine ilişkin hata yanıtı gösterilmektedir:

```
"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 örnekte, bazı kuyruk yöneticilerine ilişkin var olmayan bir kuyruk paylaşım grubunda kuyruk silme girişimine ilişkin hata yanıtı gösterilmektedir:

```
"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 etkin değilse ve MFT REST API' u çağırdığınızda, 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"  
  }]  
}]
```

If MFT REST API services are enabled and the coordination queue manager is not set in the mqwebuser.xml file, you receive the following exception:

```
{  
  "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 için Documentation 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 keşif özelliği etkinleştirilerek görüntülenebilir.

Başlamadan önce

Önemli: API Discovery özelliği sabitlendi ve bu özelliği kullanmaya devam edebilirsiniz. Şu anda IBM MQ , mpOpenAPI özelliğinin kullanımını desteklemez.

API keşfi 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:

- Tarayıcıya aşağıdaki URL 'yi girerek REST API ' e 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.

- Aşağıdaki URL 'ye bir HTTP GET komutu vererek REST API ' in tamamını açıklayan tek bir Swagger 2 belgesini alın:

`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 adresinden doğru değerleri belirleyebilirsiniz. URL ' yi görüntülemek için **dspmqweb status** komutunu kullanın.

REST API ulusal dil desteđi

REST API , belirli niteliklere sahip, ulusal dilleri bir HTTP isteđinin bir parçası olarak belirtme yeteneđi sađlar.

Arka Plan

HTTP üstbilgileri , isteklerde belirli davranışların belirtilmesine ve yanıtlarda sađlanacak ek bilgilerde belirtilmesine olanak sađlar.

HTTP üstbilgilerinde, bu bilgilerin ulusal bir dilde döndürülmesi istenmektedir. REST API , mümkün olduđu yerlerde bu üstbilgiyi kabul eder.

Ulusal dil belirtilmesi

ACCEPT-LANGUAGE HTTP üstbilgisinde, bir ya da daha fazla dil etiketi sađlanabilir. İsteđe bađlı olarak, bir sırayı etiketlerle ilişkilendirebilir ve tercihe göre sıralanmış bir liste belirtimine olanak tanıyabilirsiniz. [Bu sayfa](#) , ilkeye ilişkin yararlı bir tartışmaya sahiptir.

REST API bu üstbilgiyi kabul eder, KABUL-DİL üstbilgisinden bir dil seçin ve bu dilde iletiler döndürür. ACCEPT-LANGUAGE üstbilgisi, REST API ' in destekleyebileceđi bir dil içermediđinde, iletiler varsayılan bir dilde döndürülür. Bu varsayılan dil, REST API web sunucusunun varsayılan yerel ayarına karşılık gelir.

“Hangi veriler çevrilir?” sayfa 84 bölümü, hangi verilerin çevrildiđini açıklar.

Yanıtlarda geçerli dili belirtir

REST API ' tan gelen yanıtlardaki CONTENT-LANGUAGE HTTP üstbilgisi, iletilerin döndürüldüđu dili gösterir.

Hangi veriler çevrilir?

Hata ve bilgi iletileri çevrilir, diđer metinler de çevrilmez.

- Bir kuyruk yöneticisinden döndürülen veriler çevrilmez; örneđin, REST API aracılıđıyla bir MQSC komutu yürütme durumunda, kuyruk yöneticisinin yanıtları kuyruk yöneticisinin yerel ayarında yer alıyor.
- The generated (Swagger) documentation for the REST API, as exposed via the apiDiscovery feature, is in English.

Hangi diller destekleniyor?

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

Çince (Basitleştirilmiş)

Denoted by the language tag zh_CN

Çince (Geleneksel)

Denoted by the language tag zh_TW

Çekçe

Denoted by the language tag cs

Fransızca

Denoted by the language tag fr

Macarca

Denoted by the language tag hu

İtalyanca

Denoted by the language tag it

Japonca

Denoted by the language tag ja

Korece

Denoted by the language tag ko

Lehçe

Denoted by the language tag pl

(Brezilya) Portekizce

Denoted by the language tag pt_BR

Rusça

Denoted by the language tag ru

İspanyolca

Denoted by the language tag es

Örnekler

Örneklerde, web sunucusunda bir İngilizce varsayılan ülke değeri vardır.

Desteklenen tek bir dil belirtilmesi

İstek üstbilgilerinde ACCEPT-LANGUAGE , frolarak ayarlıdır. Bu ayar, Fransızca (translatable) metin için Fransızca tercih edilen dil olarak belirtilir.

Yanıt üstbilgilerinde CONTENT-LANGUAGE , frolarak ayarlıdır. Bu ayar, yanıtta hata ve bilgi verici iletilerin Fransızca olarak olduğunu belirtir.

Dillerin listesini belirleme

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

Yanıt üstbilgilerinde CONTENT-LANGUAGE , frolarak ayarlıdır. Bu ayar, REST API ' in Amharic 'i desteklemediği için, yanıtta hata ve bilgi verici iletilerin Fransızca olarak olduğunu belirtir.

Desteklenmeyen tek bir dil belirtilmesi

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

Yanıt üstbilgilerinde CONTENT-LANGUAGE , enolarak ayarlıdır. Bu ayar, REST API ' in Amharic 'i desteklemediği için, yanıtta hata ve bilgi verici iletilerin İngilizce olduğunu belirtir.

REST API sürümler

The REST API version number forms part of the base URL for REST requests. Örneğin, `https://localhost:9443/ibmmq/rest/v2/admin/installation`. The version number is used to isolate clients from changes to the REST API that might be introduced in future releases.

V 9.2.0 IBM MQ 9.2.0 , REST API' un 2 sürümünü tanıtır. Bu sürüm artışı, administrative REST API, messaging REST API ve MFT REST API için geçerlidir. Bu sürüm artışı, REST API için kullanılan kaynak URL adresini değiştirir. Sürüm 2 'deki kaynak URL 'lerine ilişkin URL öneki şu URL' dir:

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

REST API ile tanımlanan bazı değişiklikler, varolan REST API işlevlerini değiştirebilir, bu durumda REST API ' u kullanan istemcilerin güncelleştirilmesi gerekebilir. Bu tür değişikliklerin istemcilerin güncellenmesini önlemek için, REST API sürüm numarası artırılır ve var olan işlev önceki sayıyla sabitlenmektedir. Var olan işlevi değiştirebilecek yeni işlev, yeni sürüm numarasından REST API değerine eklenir. Bu nedenle, istemciler güncellenmeden önceki sürümdeki REST API sürümünü kullanmaya devam edebilir.

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

- Removal of support for an existing attribute in the JSON that is sent to, or returned from, the REST API.
- Bir URL, HTTP komutu ya da üstbilgisinin kaldırılması. Örneğin, bir URL ya da üstbilgi yeniden adlandırılırsa ya da farklı bir komut kullanılırsa.
- Var olan bir URL ' ye gönderilen verilere yeni zorunlu bir JSON özneliği eklenmesi.

- Var olan bir URL ' ye gönderilen verilere yeni zorunlu HTTP üstbilgisi eklenmiştir.
- Var olan bir URL ' ye yeni bir zorunlu sorgu parametresi eklenmesine ek olarak.

Bir Long Term Support (LTS) yayınında var olan REST API işlevine bu değişiklik türü sunulduğunda, bu değişikliklerin ilki için REST API sürüm numarası artırılır. Any subsequent changes that are made within a Continuous Delivery (CD) release that might require changes to clients that use the REST API use the new version number.

This version number remains the same throughout subsequent CD releases until the next LTS release. Bu nedenle, sürüm numarası LTS yayın düzeyleri arasında en çok bir kez artar.

Sürüm numarası artırıldığında, var olan REST API işlevi eski sürüm numarasından sabitlenmektedir. Yani, LTS yayınında kullanılabilir olan var olan REST API işlevi eski sürüm numarasında kalır, ancak bu sürümde başka değişiklik yapılmamaktadır. REST API ' a eklenen her yeni işlev, yeni REST API sürümüne eklenir. However, any additions that are made to the REST API in CD releases before the version increase are not guaranteed to be included in the older version of the REST API.

Var olan istemciler, herhangi bir değişiklik gerektirmeden, eski sürüm numarasından REST API ' ı kullanmaya devam edebilir. REST API ' in eski sürümleri kullanımdan kaldırılabilir ve sonunda kaldırılabilir.

Bazı değişiklikler, REST API ' u kullanan istemcilerde değişiklik yapılmasını gerektirmez. Bu değişiklikler, sürüm numarasının artışlarıyla sonuçlanmaz. Bu nedenle, bu değişiklik tipleri kullanıma sunulduğunda REST API ' i kullanan tüm istemcilerin güncellenmesine gerek olmadığını doğrulayın. REST API ' da yapılan bu değişiklikler aşağıdaki değişiklikleri içerebilir:

- REST API' den döndürülen var olan verilere yeni bir JSON özniteliği eklenmiştir.
- Yeni bir URL eklendi.
- Var olan bir URL ' ye yeni bir HTTP yüklemi eklenmesi.
- Var olan bir URL ' ye yeni durum kodu eklenmesi.
- Var olan bir URL ' ye gönderilen verilere yeni isteğe bağlı JSON özniteliklerinin eklenmesi.
- Var olan bir URL ' de yeni sorgu parametrelerinin eklenmesi.
- Var olan bir URL ' ye gönderilen verilere yeni üstbilgilerin eklenmesi.
- REST API' den yeni üstbilgilerin döndürülmesi.

Yeni Continuous Delivery REST API işlevinde yapılan değişiklikler

Bir CD yayınına eklenen yeni REST API işlevi için, bu yeni işlevde yapılan tüm değişiklikler, REST API istemcilerinde yapılan değişikliklerin sürüm numarasını artırmamasını gerektirmektedir. Yani, yeni işlev, sonraki LTS yayınından önce sürüm numarasını artırmadan değişebilir. İşlev bir LTS yayınına eklendiğinde, REST API istemcilerinde değişiklik gerektirebilecek sonraki değişiklikler sürüm numarasını artırabilir.

Örnek

1. LTS yayın düzeyi X 'te, REST API sürüm 1 'de yer alıyor.
2. CD yayın düzeyinde X.0.1, yeni bir URL adresi desteği eklenmiştir. Bu değişiklik, REST API' u kullanan istemcilerde değişiklik yapılmasını gerektirmez. Bu nedenle, REST API sürüm 1 'de kalır.
3. CD X.0.2' de yeni bir URL adresi desteği eklenir. Bu değişiklik, REST API ' yı kullanan istemcilerde değişiklik yapılmasını gerektirmez. Bu nedenle, REST API sürüm 1 'de kalır.
4. LTS yayın Y ' de REST API , sürüm 1 'de yer alıyor.
5. CD yayın düzeyinde Y.0.1, var olan bir URL yeniden adlandırılır. Bu değişiklik, REST API' u kullanan istemcilerde değişiklik yapılmasını gerektirebilir. Therefore, a new version of the REST API is created as version 2. Yeniden adlandırılan URL, var olan tüm işlevlerle birlikte REST API' un 2. sürümünde yer alır. REST API ' a eklenen yeni işlevler sürüm 2 'ye eklenir. Sürüm 1, LTS yayın düzeyi Y ' de belirtilen düzeyde dengelenmiş olmaya devam eder.

6. CD yayın düzeyi Y.0.2, var olan başka bir URL yeniden adlandırıldı. Sürüm CD yayın Y ' de zaten artmış olduğundan, REST API sürüm 2 'de kalır. Sürüm 1, LTS yayın düzeyi Y ' de belirtilen düzeyde dengelenmiş olmaya devam eder.
7. LTS yayın düzeyinde Z ' de REST API , sürüm 2 'de kalır. Sürüm 1, LTS yayın düzeyi Y ' de belirtilen düzeyde dengelenmiş olmaya devam eder.

Web konsolu kullanan yönetim

Bir web konsolunu kullanarak temel yönetim görevlerini gerçekleştirebilirsiniz.

V 9.2.0 IBM MQ 9.2.0' tan, New Web Console olarak bilinen yeni bir web konsolu kullanılabilir, bkz. [“Yeni Web Konsoluna Hızlı Gezin”](#) sayfa 89.

You can, if you want, continue to use the Dashboard Web Console on Multiplatforms, see [“Konsol tipleri arasında geçiş yapma”](#) sayfa 109.

Not: Bir web konsolu kullandığınızda, kuyruk yöneticinizin herhangi birinde komut sunucusunu devre dışı bırakmayın. Bir kuyruk yöneticisi için komut sunucusu devre dışı bırakılırsa, web konsolu, komutların işlenmesine ilişkin uzun gecikmeler ile yanıt vermez duruma gelir. Kuyruk yöneticisine verilen komutların, komut sunucusunun devre dışı bırakıldığı zaman aşımı süresi.

İlgili görevler

V 9.2.0 Yeni Web Konsolunun İzlenmesi
[Gösterge Panosu Web Konsolunun İzlenmesi](#)

V 9.2.0 Web konsoluyla çalışmaya başlama

Web konsolu ile hızlı bir şekilde çalışmaya başlayın.

Başlamadan önce

Bu görevi tamamlamak için, **dspmqweb** komutunu kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

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

IBM i IBM üzerinde, komutların QSHELL içinde çalıştırılması gerekir.

Yordam

1. mqweb sunucusu, web konsolu tarafından kullanılmak üzere önceden yapılandırılmamışsa, mqweb sunucusunu yapılandırın.

Temel bir kayıt dosyasıyla mqweb sunucusu için temel yapılandırma oluşturma 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 web konsoluna ilişkin URI ' yi belirleyin:

```
dspmweb status
```

Komut, aşağıdakine benzer bir çıktı oluşturur:

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

Web konsolunun URI 'si console/sonakiyle sona eriyor.

4. Tarayıcıda “3” sayfa 88 URL adresini girerek web konsoluna bağlanın.

mqweb sunucusu ile sağlanan varsayılan sertifika güvenilir bir sertifika olmadığından, tarayıcı tarafından bir güvenlik kural dışı durumu üretilebilir. Web konsoluna ilerlemek için bu seçeneği belirleyin.

5. Web konsolunda oturum açın. Use the user name mqadmin, and the password mqadmin.

Sonraki adım

Varsayılan olarak, web konsolu kullanıcıların kimliklerini doğrulamak için simgeli tabanlı kimlik doğrulaması kullanır. İstemci sertifikası kimlik doğrulamasını da kullanabilirsiniz. Daha fazla bilgi için [REST API ve Web Konsoluya istemci sertifikası kimlik doğrulamasını kullanmabaşlıkları konuya bakın.](#)

z/OS z/OS ile ilgili kısıtlamalar

z/OS' ta kuyruk yöneticilerini yönetmek için IBM MQ Console kullanılırken aşağıdaki kısıtlamalar geçerlidir.

- z/OS üzerindeki kuyruk yöneticileri yaratılamaz, silinemez, başlatılamaz ya da durdurulamaz.
- z/OS üzerindeki kanal başlatıcıları başlatılamıyor ya da durdurulamıyor ve kanal başlatıcı durumu görüntülenmiyor.
- Dinleyiciler görüntülenemez ya da denetlenemez.
- Kanal komutlarını başlatma, ping, çözümlenme ve ilk duruma getirme komutları yalnızca CHLDISP (DEFAULT) ile yayınlanabilir.
- Yeni nesnelere yalnızca QSGDISP (QMGR) ile yaratılabilir.
- QSGDISP (GROUP) ile tanımlanan nesnelere görüntülenemez ya da yönetilemez.
- Kuyruk yöneticisi güvenliği yönetilemiyor.
- Sistem kaynağı kullanımı izlenemiyor.

İlgili kavramlar

[“Web konsolu kullanan yönetim” sayfa 87](#)

Bir web konsolunu kullanarak temel yönetim görevlerini gerçekleştirebilirsiniz.

İlgili görevler

[“Yerel kuyruk yöneticileriyle çalışılması” sayfa 92](#)

Yerel kuyruk yöneticilerini, Yönet görünümünün üst düzeyinden oluşturursun, yapılandırır ve

denetleyebilirsiniz



Multi IBM MQ for Multiplatforms ile ilgili kısıtlamalar

IBM MQ for Multiplatforms' ta kuyruk yöneticilerini yönetmek için IBM MQ Console kullanılırken aşağıdaki kısıtlamalar geçerlidir.

- You cannot use the IBM MQ Console to work with AMQP channels.
- You cannot use the IBM MQ Console to work with MQTT channels.

İlgili kavramlar

“Web konsolu kullanan yönetim” sayfa 87

Bir web konsolunu kullanarak temel yönetim görevlerini gerçekleştirebilirsiniz.

İlgili görevler

“Yerel kuyruk yöneticileriyle çalışılması” sayfa 92

Yerel kuyruk yöneticilerini, Yönet görünümünün üst düzeyinden oluşturursun, yapılandırır ve

denetleyebilirsiniz

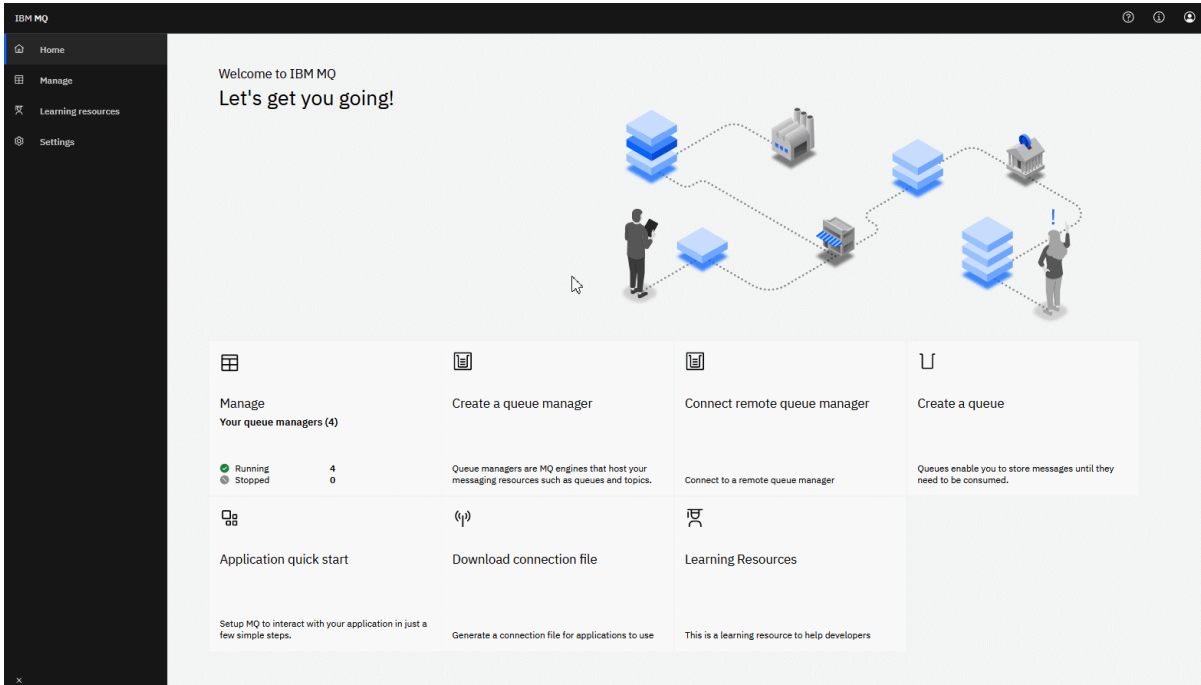


V 9.2.0 Yeni Web Konsoluna Hızlı Gezin

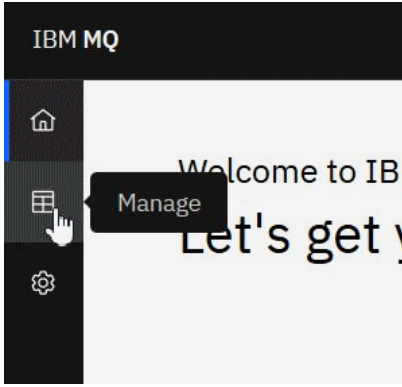
Hızlı gezinti, New Web Console ile çalışmaya başlamanızı sağlar. Kullanım iziyle ilgili daha ayrıntılı konular.

The New Web Console is the preferred web UI, but if you want to carry on using the existing console (the Dashboard Web Console), you can switch back to that, see [“Konsol tipleri arasında geçiş yapma” sayfa 109](#).

New Web Console 'da ilk oturum açtığınızda, giriş sayfasına götürülersiniz. Buradan, var olan kuyruk yöneticilerini yönetebilir, bir kuyruk yöneticisi ya da kuyruk yaratabilir, bazı eğitim konularına gidebilirsiniz ya da IBM Documentation'te IBM MQ ürün bilgilerini açabilirsiniz. Ayrıca, yeni ya da var olan kuyruk yöneticileri ile uygulamalar arasında ileti sistemini hızlı ve kolay bir şekilde ayarlamak için size yol gösteren, uygulamayı hızlı başlatma da başlatabilirsiniz.



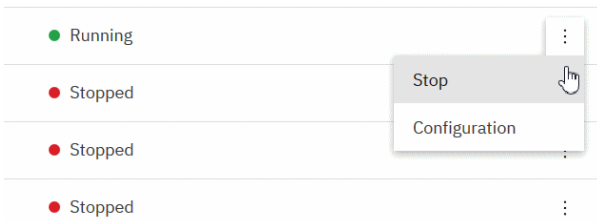
Alternatively, you can just click the manage icon to start managing IBM MQ objects straightaway.



Yönetme görünümü başlangıçta kuyruk yöneticilerini ve bunların yürürlükteki durumunu 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 sağlayan bir menüsü vardır; durdurulmuş bir kuyruk yöneticisini başlatabilir ya da silebilirsiniz.



Kuyruk yöneticisine ilişkin Yetki Kayıtları, Kimlik Doğrulama Bilgileri nesnelere ve Kanal Kimlik Doğrulaması kayıtları, yeni olanlar yaratabileceğiniz ve ekleyebileceğiniz kuyruk yöneticisinin **Yapılandırma** sayfasının **Güvenlik** etiketinden bulunabilir.


Gösterge panosunu açmak için, çalışmakta olan 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 tamamlayabilirsiniz:


Kuyruklar sekmesinde:

- Yeni kuyruklar yarat
- Var olan kuyrukları yapılandır 
- Varolan iletileri görüntülemek ve yeni iletiler yaratmak için kuyruk adını tıklatın.



Konular sekmesinde:

- Yeni konular yarat
- Var olan konuları yapılandır 
- Eşleşen abonelikleri görüntülemek için bir konu adını tıklatın

Abonelikler sekmesinde:

- Yeni yönetilen ya da yönetilmeyen abonelikler yarat
- Var olan abonelikleri yapılandır 

İletişim etiketinde:

- Dinleyiciler:
 - İletişimleri başlat, durdur ve yapılandır 
 - Yeni dinleyiciler yarat
- Kuyruk Yöneticisi Kanalları:
 - Kanalları başlatma, durdurma, ping ve yapılandırma kanalları 
 - Yeni kanallar yarat
 - Kanalları ilk durumuna getir (**Gelişmiş** menü öğesinden)
 - Kanallardaki belirsiz iletileri çözümler (**Gelişmiş** menü öğesinden)

- Uygulama Kanalları:
 - Kanalları başlatma, durdurma, ping ve yapılandırma kanalları
 - Yeni kanallar yarat
 - Kanalları ilk durumuna getir (**Gelişmiş** menü ögesinden)
 - Kanallardaki belirsiz iletileri çözümler (**Gelişmiş** menü ögesinden)

Yerel kuyruk yöneticileriyle çalışılması

Yerel kuyruk yöneticilerini, Yönet görünümünün üst düzeyinden oluşturursun, yapılandırır ve

denetleyebilirsiniz 

Bu görev hakkında

Multi Görünümün yönetilmesi, yerel kuyruk yöneticilerine eklenen IBM MQ Console ' in çalışmakta olduğu IBM MQ kuruluşu. Aynı sistemde IBM MQ ' un farklı kuruluşlarıyla ilişkili kuyruk yöneticileri listelenmiyor.

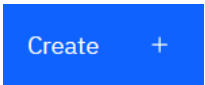
z/OS On z/OS, the Manage view lists the queue managers that are at the same version as the IBM MQ Console, and are defined on the system where the MQ Console is running. MQ Console ' a farklı bir sürümdeki kuyruk yöneticileri listelenmiyor.

Listeden çalışmak için listeden tek tek kuyruk yöneticileri seçebilirsiniz.


Not: IBM MQ Console , eşlenmiş veri kuyruğu yöneticilerini (RDQM ' ler) desteklemez.

Yordam


- Yeni bir yerel kuyruk yöneticisi yaratmak için:

- Kuyruk yöneticisi listesi görünümünde oluştur düğmesini  tıklattın.
- Yeni kuyruk yöneticisi için bir ad girin. Ad en çok 48 karakter içerebilir. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
- İsteğe bağlı: Kuyruk yöneticisinin dinlemesi için kullanılabilir bir TCP/IP kapısı girin. Kapı numarası 65535 ' i aşmamalıdır.
- Oluştur**' u tıklattın. Yeni kuyruk yöneticisi yaratılır ve başlatılır.


- Yerel kuyruk yöneticisini başlatmak için:



- Listede başlatmak istediğiniz kuyruk yöneticisini bulun.
- Menüden **StartBaşlat** ' ı seçin  seçeneğini tıklattın.

- Yerel kuyruk yöneticisini durdurmak için:

- Yerel kuyruk yöneticisi pencere bileşenindeki listeden durdurmak istediğiniz kuyruk yöneticisini seçin.
-  menüsünden **Durdur** ögesini seçin.

- Bir yerel kuyruk yöneticisini silmek için:

- Kuyruk yöneticisi çalışıyorsa, durdurun.
- Select **Yapılandır** from the menu  and select **Kuyruk yöneticisini sil**.

- c) Onay penceresine adını girerek kuyruk yöneticisini silmek istediğinizi onaylayın. Kuyruk yöneticisi ve ilişkili tüm nesnelere silinir.
- Yerel bir kuyruk yöneticisine ilişkin özellikleri görüntülemek ve düzenlemek için:
 - a) Kuyruk yöneticisinin çalışır durumda olduğundan emin olun ve kuyruk yöneticisi listesinde bu yöneticiyi bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Özellikler** sekmesinin seçili olduğundan emin olun. Özellikleri görüntüleyin ve gereken şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. Bir özelliğe ilişkin bilgi edinmek için, [Kuyruk yöneticisi özellikleri](#) içindeki özellik bilgilerini görüntüleyebilirsiniz.
- Yerel kuyruk yöneticisine ilişkin güvenlik ayarlarıyla çalışmak için:
 - a) Kuyruk yöneticisinin çalışır durumda olduğunu doğrulayın ve kuyruk yöneticisi listesinde bu yöneticiyi seçin.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Security** (Güvenlik) sekmesinin seçili olduğundan emin olun.
 - d) Kimlik doğrulama nesnelere, yetki kayıtlarıyla ya da kanal doğrulama nesnelere çalışabilirsiniz. Daha fazla bilgi için aşağıdaki konuları ziyaret edin:
 - [“Kimlik doğrulama bilgileri nesnelere çalışma”](#) sayfa 93
 - [“Kuyruk yöneticisi yetki kayıtlarıyla çalışılması”](#) sayfa 95
 - [“Kanal kimlik doğrulama kayıtlarıyla çalışma”](#) sayfa 96

V 9.2.0 Kimlik doğrulama bilgileri nesnelere çalışma


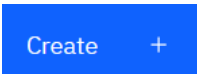
Konsol, bir kuyruk yöneticisine kimlik doğrulama bilgi nesnelere eklemek ve bunları silmek için kullanabilirsiniz. Ayrıca, özellikleri görüntüleyebilir ve ayarlayabilir ve nesnelere ilgili yetki kayıtlarını yönetebilirsiniz.



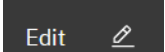



Bu görev hakkında

Kimlik doğrulama bilgileri görünümü, belirli bir kuyruk yöneticisi için var olan kimlik doğrulama bilgilerini listeler. Birlikte çalışmak için listeden tek tek kimlik doğrulama bilgileri seçebilirsiniz.

The queue manager authentication information forms part of IBM MQ support for Transport Layer Security (TLS). Bu nesnelere, LDAP sunucularında OCSP ya da Sertifika İptal Listeleri (CRL) kullanılarak sertifika iptal denetimini gerçekleştirmek için gereken tanımlamaları ve kullanıcı kimliği ve parola denetimini etkinleştirmek için gerekli tanımlamalar içerir.

Yordam

- Bir kuyruk yöneticisine ilişkin kimlik doğrulama bilgilerini görüntülemek için:
 - a) Kuyruk yöneticisinin çalışır durumda olduğunu doğrulayın ve kuyruk yöneticisi listesinde bu yöneticiyi seçin.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Security** (Güvenlik) sekmesinin seçili olduğundan emin olun.
 - d) Gezinme panosundan **Kimlik doğrulama bilgileri** seçeneğini belirleyin.
- Bir kimlik doğrulama bilgileri nesnesi eklemek için:
 - a) Kimlik doğrulama bilgileri liste görünümünde oluştur düğmesini  tıklattığınızda.

- 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 adı** belirtin. Bu ad, LDAP sunucusunun çalışmakta olduğu anasistemin anasistem adı, IPv4 noktalı onlu adresi ya da IPv6 onaltılı gösterimidir ve isteğe bağlı bir kapı numarası ile gösterilir. İsteğe bağlı olarak, LDAP sunucusuna erişen kullanıcı için bir kullanıcı adı ve parola belirleyebilirsiniz.
 - **OCSP** için, **OCSP yanıtlayıcı URL 'si** belirtin. Bu URL, sertifika iptali için denetlemek üzere kullanılan yanıtlayıcının URL adresidir. Bu değer, OCSP yanıtlayıcıya ilişkin ana makine adını ve kapı numarasını içeren bir HTTP URL 'si olmalıdır. OCSP yanıtlayıcısı, HTTP için varsayılan değeri olan 80 numaralı kapıyı kullanıyorsa, kapı numarası atlanabilir. HTTP URL adresleri RFC 1738 'de tanımlanır.
 - **IDPW OS** için, isteğe bağlı olarak bu kimlik doğrulama tipi için ek seçenekler de belirtebilirsiniz.
 - **IDPW LDAP** için, **LDAP sunucusu adı** ve **Kısa kullanıcı** adını belirtin. LDAP sunucusu adı, LDAP sunucusunun çalışmakta olduğu anasistemin anasistem adı, IPv4 noktalı onlu adresi ya da IPv6 onaltılı gösterimidir ve isteğe bağlı bir kapı numarası vardır. Kısa kullanıcı adı, LDAP kullanıcı kaydındaki, bağlantı için kısa ad olarak kullanılan alandır. İsteğe bağlı olarak, bu kimlik denetimi tipi için ek seçenekler belirleyebilirsiniz.
- e) **Ekle**'yi tıklatın.
- Bir kimlik doğrulama bilgileri nesnesini silmek için:
 - a) Listedeki silmek istediğiniz kimlik doğrulama bilgileri nesnesi için spanner simgesini  seçin.
 - b) Nesne özellikleri görünümünde, **Kimlik doğrulama bilgileri nesnesini sil** ögesini tıklatın.
 - c) Kimlik doğrulama bilgileri nesnesini silmek istediğinizi doğrulamak için **Sil** düğmesini tıklatın. Nesne silindi.
 - Bir kimlik doğrulama bilgileri nesnesinin özelliklerini görüntülemek ve düzenlemek için:
 - a) Listedeki görüntülemek istediğiniz kimlik doğrulama bilgileri nesnesi için spanner simgesini  seçin.
 - b) Görüntülenen özellikleri düzenlemek için **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.
 - Bir kimlik doğrulama bilgileri nesnesine ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Listedeki ilgili yetki kaydını görüntülemek istediğiniz kimlik doğrulama bilgileri nesnesi için spanner simgesini  seçin.
 - b) **Security** (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  tıklatın, yeni yetki kaydının ayrıntılarını belirtin ve **Oluştur** 'u tıklatın.


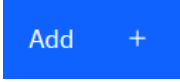
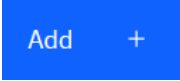


Kuyruk yöneticisi yetki kayıtlarıyla çalışılması

Kullanıcıların ve grupların, o kullanıcıya ya da gruba ilişkin bir yetki kaydı belirtilerek, kuyruk yöneticilerine erişimi denetleyebilirsiniz.

Bu görev hakkında

Yetki kayıtlarını kullanarak, bir ileti alışverişi kullanıcısının ya da ileti alışverişi kullanıcılarının belirli bir kuyruk yöneticisine sahip olduğu erişim erişimini ince ayarlayabilirsiniz. İki tip yetki kaydı vardır: Genel yetkileri denetleyen yetki kayıtları ve hangi kullanıcıların ve grupların kuyruk yöneticisi için nesne yaratabileceğini denetleyen yetki kayıtları yaratılır.

Yordam

- Bir kuyruk yöneticisine ilişkin yetki kayıtlarını görüntülemek için:
 - a) Kuyruk yöneticisinin çalışır durumda olduğunu doğrulayın ve kuyruk yöneticisi listesinde bu yöneticiyi seçin.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Security** (Güvenlik) sekmesinin seçili olduğundan emin olun.
 - d) Gezinme panosundan **Yetki kayıtları** seçeneğini belirleyin. Bu görünüm, genel yetki kayıtlarıyla ve yetki kayıtları yaratma ile çalışmanıza olanak tanıyan iki bölmede yetki kayıtlarını gösterir.
- Genel yetki kaydı eklemek için:
 - a) Yetki kayıtları liste görünümünde ekle düğmesini  tıklatın.
 - b) Bir kullanıcı ya da grup için bir yetki kaydı eklenip eklenmeyeceğini seçin.
 - c) Yetki kaydı eklemekte olduğunuz kullanıcı ya da grubun adını belirtin (yetki kaydı bunu adı olarak alır).
 - d) İzin vermek istediğiniz yetkileri seçin (bkz. yetkiler hakkında daha fazla bilgi için)
 - e) **Yarat** düğmesini tıklatın.
- Yaratma yetkisi kaydı eklemek için:
 - a) Yetki kayıtları listesi görünümünde ekle düğmesini  tıklatın.
 - b) Bir kullanıcı ya da grup için bir yetki kaydı eklenip eklenmeyeceğini seçin.
 - c) Yetki kaydı eklemekte olduğunuz kullanıcı ya da grubun adını belirtin (yetki kaydı bunu adı olarak alır).
 - d) Yaratma yetkisi vermekte olduğunuz nesne tiplerini seçin.
 - e) **Yarat** düğmesini tıklatın.
- Bir yetki kaydını silmek için:
 - a) Silmek istediğiniz yetki kaydına ilişkin menüyü açın  ve **Sil** seçeneğini belirleyin.
 - b) Kimlik doğrulama bilgileri nesnesini silmek istediğinizi doğrulamak için **Sil** düğmesini tıklatın. Nesne silindi.
- Bir yetki kaydına ilişkin özellikleri görüntülemek ve düzenlemek için:
 - a) Silmek istediğiniz yetki kaydı için  menüsünü açın ve **Düzenle** seçeneğini belirleyin.
 - b) Ayarları gerektiği şekilde değiştirin ve değişikliklerinizi kaydetmek için **Kaydet** düğmesini tıklatın.


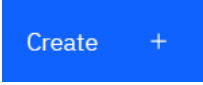
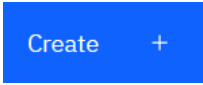


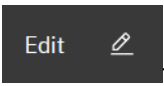
Bir kuyruk yöneticisine kanal doğrulama kayıtları eklemek ve silmek için IBM MQ Console 'i kullanabilirsiniz. Ayrıca, kanal doğrulama kayıtlarına ilişkin özellikleri de görüntüleyebilir ve ayarlayabilirsiniz.

Bu görev hakkında

Bir kanal düzeyinde sistemleri birbirine bağlama yetkisi verilen erişim üzerinde daha kesin bir denetim yapmak için kanal kimlik doğrulama kayıtlarını kullanabilirsiniz.

Güvenliği uygulamak için, kanallarınıza erişimi engellemek için kanal doğrulama kayıtlarını engellemeyi kullanabilirsiniz. Ayrıca, belirtilen kullanıcılara erişim izni vermek için adres eşleme kanalı kimlik doğrulama kayıtlarını da kullanabilirsiniz. Kanal kimlik doğrulama kayıtları hakkında daha fazla bilgi edinmek için bkz. [Kanal doğrulama kayıtları](#).

Yordam

- Bir kuyruk yöneticisine ilişkin kanal kimlik doğrulama bilgilerini görüntülemek için:
 - a) Kuyruk yöneticisinin çalışır durumda olduğunu doğrulayın ve kuyruk yöneticisi listesinde bu yöneticiyi seçin.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Security** (Güvenlik) sekmesinin seçili olduğundan emin olun.
 - d) Gezinme panosundan **Kanal kimlik doğrulaması** seçeneğini belirleyin.
- Kanal kimlik doğrulaması kaydı eklemek için:
 - a) Kanal kimlik doğrulaması bilgi listesi görünümünde oluştur düğmesini  tıklatın.
 - b) Kullanmak istediğiniz kural tipini seçin. Bir **İzin Ver, Öbek**ya da **Uyarmaseçeneğini** belirleyin.
 - c) Bir kanal kimlik doğrulama kuralını yapılandırdığınız kimlik tipini seçin. Seçtiğiniz kural tipine bağlı olarak farklı kimlik tipleri kullanılabilir.
 - d) Belirtmekte olduğunuz kimlik için gerekli bilgileri sağlayın. Varsayılan olarak, değerleri belirtmeniz için önerilen özellikler alt sınırı görüntülenir. **Tüm kullanılabilir seçenekleri gösterseçeneğini** belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
 - e) Kanal kimlik doğrulama kaydını oluşturmak için oluştur düğmesini  tıklatın. Kanal doğrulama kayıtlarına ilişkin kullanılabilir ayarlarla ilgili daha fazla bilgi için bkz. [Channel authentication records](#) ve [SET CHLAUTH](#)
- Bir kanal kimlik doğrulaması kaydını silmek için:
 - a) Silmek istediğiniz kanal kimlik doğrulama kaydının yanındaki spanner simgesini  tıklatın.
 - b) Kanal doğrulama düzenleme görünümünde, **Kanal kimlik doğrulama nesnesini sil**ögesini 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 spanner 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 tıklayın.

V 9.2.3 Uzak kuyruk yöneticisi eklenmesi

Uzak bir sistemde çalışan bir kuyruk yöneticisini yönetmek için IBM MQ Console ' yi kullanabilirsiniz.

Başlamadan önce

You must prepare the queue manager on the remote system so that it can be administered remotely, see step "1" sayfa 98 of ["Komut satırı kullanılarak uzak kuyruk yöneticisinin MQ Console ' ye bağlanması"](#) sayfa 98.

Ayrıca, IBM MQ Console' den uzak bağlantıları nasıl kullanabileceğini denetleyen bir yapılandırma dosyası da ayarlamanız gerekir. You create the configuration file by using the **setmqweb** command with the **remote** parameter (see [Uzak kuyruk yöneticisi bağlantı davranışını yapılandırma](#) and [setmqweb](#)). Yapılandırma dosyasını doğrudan düzenleyemezsiniz.

Bu görev hakkında

Uzak bağlantı ayrıntılarını belirtmek için JSON biçiminde bir istemci bağlantı tanımı çizelgesi (CCDT) kullanıyorsunuz. You can create a JSON CCDT by using a text editor (see step "2" sayfa 99 of ["Komut satırı kullanılarak uzak kuyruk yöneticisinin MQ Console ' ye bağlanması"](#) sayfa 98) or you can create one by using the IBM MQ Console.

Diğer bir seçenek olarak, doğrudan uzak kuyruk yöneticisini eklediğiniz gibi bağlantı ayrıntılarını belirterek IBM MQ Console ' den CCDT yaratabilirsiniz.

Uzak kuyruk yöneticisini gereken tüm görevler için komut satırı kullanarak (uzak kuyruk yöneticisini hazırlamaya ve CCDT oluşturmaya ek olarak) IBM MQ Console ' e de bağlanabilirsiniz. Bkz. ["Komut satırı kullanılarak uzak kuyruk yöneticisinin MQ Console ' ye bağlanması"](#) sayfa 98.

Yordam

- Varolan bir CCDT belirtilerek uzak kuyruk yöneticisi eklemek için:
 - a) Giriş sayfasından, **Uzak kuyruk yöneticisini bağla** seçeneğini tıklayın.
 - b) Uzak kuyruk yöneticisinin adını belirtin.
 - c) İsteğe bağlı olarak, kuyruk yöneticisi için benzersiz bir ad belirtin. Benzersiz bir ad belirlemezseniz, gerçek ad "remote-" öneki ile kullanılır.
 - d) **JSON CCDT kullanarak bağlan** seçeneğinin belirlendiğinden emin olun.
 - e) **Göz At** düğmesini tıklayın ve kullanmak istediğiniz JSON CCDT ' yi içeren dosyayı seçin.
 - f) Kullanıcı sayfasına gitmek ve isteğe bağlı olarak, uzak kuyruk yöneticisine bağlanmak için bir kullanıcı adı ve parola belirlemek üzere **İleri** düğmesini tıklayın. Bu bilgileri belirlemezseniz, kimlik doğrulama bilgileri uzak bağlantı yapılandırma dosyasından alınır.
 - g) Sertifika sayfasına gitmek için **İleri** düğmesini tıklayın. CCDT ' niz "transmissionSecurity" bilgisini belirtiyorsa, bu bilgi kullanılır. İsteğe bağlı olarak bir sertifikayı (base64 kodlu genel anahtar olarak) yapılandırabilirsiniz ve bu, genel güvenilirlik deposuna eklenir.

Sertifika, güvenilirlik deposuna eklenmeden önce geçici olarak *WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt* içinde saklanır. Bağlantı başarıyla eklenirse, sertifika bu yerden silinir.
 - h) Özet sayfasını görüntülemek için **İleri** düğmesini tıklayın. Önceki sayfaları yeniden ziyaret etmek ve düzeltmeleri yapmak için **Back** (Geri) düğmesini kullanabilirsiniz. Bu 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 yöneticisini bağla** seçeneğini tıklayın.

- b) Uzak kuyruk yöneticisinin adını belirtin.
- c) İsteğe bağlı olarak, kuyruk yöneticisi için benzersiz bir ad belirtin. Benzersiz bir ad belirlemezseniz, gerçek ad "remote-" öneki ile kullanılır.
- d) **El ile Giriş** seçeneğini belirleyin.
- e) Bağlantının kullanacağı istemci bağlantı kanalının adını girin.
- f) Uzak kuyruk yöneticisinin çalışmakta olduğu anasistemin adını belirtin. Uzak MQ kurulumları saptanırsa, anasistem adları görüntülenir ve bağlanmak istediğiniz uzak kuyruk yöneticisinin anasistemini seçebilirsiniz. Bazı ağ yapılandırmalarında, uzak MQ yönetim ortamlarının saptanması olanaklı değildir. Bu durumda, anasistem adını ve kapıyı el ile ekleyin.
- g) Kullanıcı sayfasına gitmek ve isteğe bağlı olarak, uzak kuyruk yöneticisine bağlanmak için bir kullanıcı adı ve parola belirlemek üzere **İleri** düğmesini tıklatın. Bu bilgileri belirlemezseniz, kimlik doğrulama bilgileri uzak bağlantı yapılandırma dosyasından alınır.
- h) Sertifika sayfasına gitmek için **İleri** düğmesini tıklatın. Açılan listeden bir SSL CipherSpec ögesini seçebilirsiniz. İsteğe bağlı olarak bir sertifikayı (base64 kodlu genel anahtar olarak) yapıştırebilirsiniz ve bu, genel güvenilirlik deposuna eklenir.
- Sertifika, güvenilirlik deposuna eklenmeden önce geçici olarak *WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt* içinde saklanır. Bağlantı başarıyla eklenirse, sertifika bu yerden silinir.
- i) Özet sayfasını görüntülemek için **İleri** düğmesini tıklatın. Önceki sayfaları yeniden ziyaret etmek ve düzeltmeleri yapmak için **Back** (Geri) düğmesini kullanabilirsiniz. Bu bilgilerden memnunsanız, uzak kuyruk yöneticisine bağlanmak için **Bağlan** düğmesini tıklatın.
- Belirlediğiniz bağlantı bilgileri, web dizininizdeki CCDT dosyasına yazılır. Yol şöyledir: *WLP_USER_DIR/generated.ccdt/ccdt-uniqueName*.

Sonuçlar

Uzak kuyruk yöneticisi, IBM MQ Console içindeki uzak kuyruk yöneticisi listesinde yer alır. Bağlantının başarılı olması koşuluyla, uzak kuyruk yöneticisinin nesnelere, yerel bir kuyruk yöneticisinin nesneleriyle aynı şekilde çalışabilirsiniz.

V 9.2.3 Komut satırı kullanılarak uzak kuyruk yöneticisinin MQ Console ' ye bağlanması

Komut satırındaki **setmqweb remote** komutunu kullanarak uzak kuyruk yöneticilerini MQ Console ' e bağlayabilirsiniz. Uzak kuyruk yöneticisi, MQ Console ile aynı sistemde farklı bir kurulumda çalışan bir kuyruk yöneticisi olabilir ya da farklı bir sistemde çalışan bir kuyruk yöneticidir.

Başlamadan önce

- mqweb sunucusunun, MQ Console ile uzak kuyruk yöneticisi bağlantılarına izin verecek şekilde yapılandırıldığından emin olun. Ek bilgi için [Uzak kuyruk yöneticisi bağlantıları davranışını yapılandırmabaşlıklı](#) konuya bakın.

Yordam

1. Uzak bağlantıları kabul etmek için yerel kuyruk yöneticisi QML ' yi yapılandırın:

- a) Kuyruk yöneticisinin uzak denetlenmesine izin vermek için bir sunucu bağlantısı kanalı yaratın.

Sunucu bağlantısı kanalı yaratmak için MQ Console komutunu kullanabilir ya da komut satırında **DEFINE CHANNEL** MQSC komutunu kullanabilirsiniz.

Örneğin, uzak kuyruk yöneticisi QM1 için bir sunucu bağlantı kanalı QM1 . SVRCONN yaratmak için şu komutları girin:

```
runmqsc QM1
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
```

DEFINE CHANNEL ve kullanılabilir seçenekler hakkında daha fazla bilgi için bkz. [KANAL TANIMLAMA](#).

- b) Sunucu bağlantısı kanalının, kanala uygun bir kullanıcı kimliği erişimine izin verdiğinden emin olun. This user ID must be the user ID that the mqweb server is started with on the system where the MQ Console makes the remote connection.

Uygun yetki kaydını yaratmak için MQ Console komutunu kullanabilir ya da komut satırında **SET CHLAUTH** MQSC komutunu kullanabilirsiniz.

Örneğin, bir kullanıcıyı yetkilendirmek için `exampleUser`, uzak kuyruk yöneticisi `QM1` için `QM1.SVRCONN` olanağına erişmek için şu komutu girin:

```
SET CHLAUTH(QM1.SVRCONN) TYPE(ADDRESSMAP) ADDRESS('*') MCAUSER('exampleUser')
```

Bu örnekte, **address** parametresi, `exampleUser` ' un herhangi bir IP adresinden bağlanabileceği şekilde ayarlanır. **address** parametresini, bunun yerine belirli bir IP adresine ayarlayabilirsiniz. For example, you might restrict access to only the IP address from which the MQ Console remotely connects to the queue manager. Bu komut için kullanılabilir seçeneklerle ilgili daha fazla bilgi için bkz. **SET CHLAUTH**.

c) **ALW**

Gelen ağ bağlantılarını kabul etmek için bir dinleyici yaratın:

Bir dinleyici yaratmak için MQ Console komutunu kullanabilir ya da komut satırında **DEFINE LISTENER** MQSC komutunu kullanabilirsiniz.

Örneğin, uzak kuyruk yöneticisi `QM1` için 1414 numaralı kapıda bir dinleyici `REMOTE.LISTENER` yaratmak için şu komutu girin:

```
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
```

- d) Dinleyicinin çalışır durumda olduğundan emin olun:

Dinleyiciyi başlatmak için MQ Console ' ı kullanabilir ya da komut satırında **START LISTENER** MQSC komutunu kullanabilirsiniz.

ALW For example, on AIX, Linux, and Windows to start the listener `REMOTE.LISTENER` for queue manager `QM1`, enter the following command:

```
START LISTENER(REMOTE.LISTENER)
```

z/OS For example, on z/OS, to start the listener, enter the following command:

```
START LISTENER TRPTYPE(TCP) PORT(1414)
```

z/OS üzerinde bir dinleyici başlatabilmeniz için kanal başlatıcı adres alanının başlatılması gerektiğini unutmayın.

2. Uzak kuyruk yöneticisi bağlantı bilgilerini içeren bir JSON CCDT dosyası yaratın:

- Uzaktan bağlanmak istediğiniz kuyruk yöneticisiyle aynı kuruluşla ilişkilendirilmiş MQ Console komutunu kullanarak, yerel bir kuyruk yöneticisi tanımlamasından bir CCDT dosyası oluşturun.

Home (Ana sayfa) panosundan **Download connection file** (Bağlantı dosyasını yükle) döşemesini tıklatın.

- Bağlantıyı tanımlayan bir JSON biçimi CCDT dosyası oluşturun. JSON biçimi CCDT yaratma hakkında daha fazla bilgi için [JSON biçimi CCDT ' nin yapılandırılması](#) başlıklı konuya bakın.

CCDT dosyası, `name`, `clientConnection` ve `type` bilgilerini içermelidir. Yani, CCDT dosyasının uzak kuyruk yöneticisi (`QM1`) için bağlantı verilerini içermesi ve yerel kuyruk yöneticisinin (`QML`) anasisteminde saklanmak zorunda olması gerekir. Bu anasistem, MQ Console ' i içeren anasistemle aynı. Özet olarak, yerel kuyruk yöneticisi `QML` ' nin uzak kuyruk yöneticisine `QM1` bağlanmasını sağlar.

İsteğe bağlı olarak, `transmissionSecurity` bilgileri gibi ek bilgileri de ekleyebilirsiniz. CCDT kanal özneteliği tanımlamalarıyla ilgili daha fazla bilgi için [CCDT kanal öznetelik tanımlamalarının tam listesibaşlıklı konuya](#) bakın.

Aşağıdaki örnekte, uzak kuyruk yöneticisi bağlantısı için temel bir JSON CCDT dosyası gösterilmektedir. Kanalın adını, "1" sayfa 98adımında yaratılan örnek sunucu bağlantısı kanalıyla aynı adı ve dinleyici tarafından kullanılan kapıyla aynı değere bağlantı noktasını ayarlar. Bağlantı anasistemi, uzak kuyruk yöneticisinin (QM1) çalıştığı sistemin anasistem adına ayarlıdır:

```
{
  "channel": [
    {
      "name": "QM1.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "example.com",
            "port": 1414
          }
        ]
      },
      "queueManager": "QM1"
    },
    {
      "transmissionSecurity": {
        "cipherSpecification": "",
        "certificateLabel": "",
        "certificatePeerName": ""
      },
      "type": "clientConnection"
    }
  ]
}
```

3. Use the **setmqweb remote** command to add the remote queue manager information to the MQ Console configuration. Uzak kuyruk yöneticisi bilgilerini görüntülemek istediğiniz kuruluşla ilişkili **setmqweb** komutunu kullanmanız gerekir.

As a minimum, to add a remote queue manager to the MQ Console you must provide the queue manager name, a unique name for the queue manager (to differentiate between other remote queue managers that might have the same queue manager name), and the CCDT URL for the queue manager. Uzak kuyruk yöneticisi bağlantısı için kullanılacak kullanıcı adı ve parola ya da güvenilirlik deposu ve anahtar deposunun ayrıntıları gibi, belirtebileceğiniz birkaç ek seçenek vardır. Parametrelerin tam listesi için **setmqweb remote** komutuyla belirtilenden daha fazla bilgi için bkz. [setmqweb](#).

Örneğin, örnek CCDT dosyasını kullanarak QM1örnek uzak kuyruk yöneticisini eklemek ve bağlantı için kullanılacak kullanıcı adının `exampleUser` olduğunu belirtmek için şu komutu girin:

```
setmqweb remote add -uniqueName "MACHINEAQM1" -qmgrName "QM1" -ccdtURL
"c:\myccdt\ccdt.json" -username "exampleUser" -password "password"
```

Sonuçlar

Uzak kuyruk yöneticisi, IBM MQ Consoleiçindeki uzak kuyruk yöneticisi listesinde yer alır. Bağlantının başarılı olması koşuluyla, uzak kuyruk yöneticisinin nesnelere, yerel bir kuyruk yöneticisinin nesneleriyle aynı şekilde çalışabilirsiniz.

V 9.2.0 IBM MQ nesneleriyle çalışma

Her IBM MQ kuyruk yöneticisinin kendisiyle ilişkili birkaç farklı nesne tipi vardır.

Bu görev hakkında

Konsolu, aşağıdaki IBM MQ nesnesi tipleriyle çalışmak için kullanabilirsiniz:

- Kuyruklar
- Konular
- Abonelikler

- İletişim nesneleri:
 - Dinleyiciler
 - Kuyruk yöneticisi kanalları
 - Uygulama kanalları

Yordam

Bir IBM MQ nesnesiyle çalışmak için:


1. Kuyruk yöneticisi listesi görünümünde, çalışmak istediğiniz nesnelere tıklanabilir olan kuyruk yöneticisini tıklatın.
2. Çalışmak istediğiniz nesne tipini seçmek için Kuyruklar, Konular, Abonelikler ya da İletişim sekmesini tıklatın.
3. Bu tip nesnelere çalışmaya ilişkin ayrıntılı yönergeler için aşağıdaki konulardan birine bakın.

Kuyruklarla Çalışma

Belirli bir kuyruk yöneticisi için var olan kuyrukları **Kuyruklar** sekmesinde görüntüleyebilirsiniz. Kuyruklar ekleyebilir ve silebilir, bir kuyruğa ileti ekleyebilir ya da iletileri temizleyebilir, iletilere göz atabilir, kuyruğun özelliklerini görüntüleyebilir ve bir kuyruğun özelliklerini ayarlayabilir ve kuyruğun yetki kayıtlarını yönetebilirsiniz.


Bu görev hakkında

Kuyruklar görünümü, belirli bir kuyruk yöneticisi için var olan kuyrukları listeler. Kuyruk yöneticisini tıklatarak ve **Kuyruklar** etiketini seçerek, kuyruk listesine erişmenizi sağlar. Birlikte çalışmak için listeden tek tek kuyruklar seçebilirsiniz.

 z/OS üzerindeki kuyruklara ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

Yordam

- Kuyruk eklemek için:

a) **Kuyruklar** sekmesinde, oluştur düğmesini  tıklatın.

b) Yaratmak istediğiniz kuyruk tipini seçin:

- Yerel kuyruk, ait olduğu kuyruk yöneticisinden gelen iletileri saklar.
- Diğer ad kuyruğu-aynı kuyruk yöneticisinde başka bir kuyruğa ilişkin gösterge.
- Uzak kuyruk-başka bir kuyruk yöneticisinde başka bir kuyruğa ilişkin gösterge.
- Model kuyruğu-dinamik bir kuyruk yöneticisi yaratıldığında kuyruk için kullanılan bir şablon.

c) Yaratmakta olduğunuz kuyruk tipi için gereken bilgileri belirtin. Varsayılan olarak, değerleri belirtmeniz için önerilen özellikler alt sınırı görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.

d) **Oluştur**'u tıklatın. Yeni kuyruk yaratıldı.

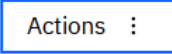


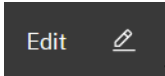

- Bir kuyruğa ileti koymak için:

a) Kuyruklar listesi görünümünde, listede ileti eklemek istediğiniz kuyruğu tıklatın. Bir model kuyruğu seçemezsiniz.

b) Oluştur düğmesini  tıklatın

c) Kuyruğa koymak istediğiniz iletiyi girin.

d) **Oluştur**'u tıklatın.


- Kuyruktan iletileri temizlemek için:
 - a) Kuyruklar listesinden iletilerin temizlenmesini istediğiniz yerel kuyruğu tıklatın.
 - b) Eylemler düğmesini  tıklatın ve **İletileri temizle** seçeneğini belirleyin.
 - c) Confirm that you want to clear the queue by clicking **İletileri temizle**.
- Kuyruklardaki iletilere göz atmak için, kuyruklar listesi görünümünde kuyruğu tıklatın. Kuyruktaki iletilerin listesi görüntülenir.
- Bir kuyruğu silmek için:
 - a) Silmek istediğiniz kuyruğun yanındaki spanner simgesini  tıklatın.
 - b) Düzenleme kuyruğu görünümünde, **Kuyruğu sil** düğmesini tıklatın.
 - c) Kuyruğu silmek istediğinizi doğrulamak için **Sildüğmesini** tıklatın. Kuyruk silindi.
- Bir kuyruğun özelliklerini görüntülemek ve düzenlemek için:
 - a) Düzenlemek istediğiniz kuyruğun yanındaki spanner 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. Özellikler hakkında bilgi için bkz. MQ Explorer belgelerindeki [Kuyruk özellikleri](#)
 - d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.
- Bir kuyruğa ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Yetki kayıtlarını düzenlemek istediğiniz, düzenlemek istediğiniz kuyruğun yanındaki spanner simgesini  tıklatın.
 - b) **Security** (Güvenlik) sekmesini tıklatın.
 - c) Kuyruk yöneticisi yetkisi kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın. Bkz. "[Kuyruk yöneticisi yetki kayıtlarıyla çalışılması](#)" sayfa 95.

Konularla çalışma

Konuları eklemek ve silmek, bir konunun özelliklerini görüntülemek ve ayarlamak için IBM MQ Console ögesini kullanabilirsiniz.

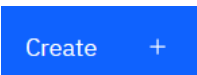
Bu görev hakkında



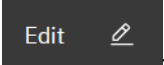
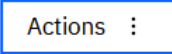
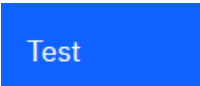
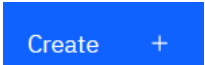


Konular görünümünde, belirli bir kuyruk yöneticisi için var olan konular listelenir. Konu listesine erişmek için bir kuyruk yöneticisini tıklatın ve **Konular** sekmesini seçin. Çalışmak için listeden tek tek konuları seçebilirsiniz.

 z/OS üzerindeki bir konuya ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

Yordam

- Bir konu eklemek için:

- a) **Konular** sekmesinde, oluştur düğmesini  tıklatın.
- b) Yaratmakta olduğunuz konu için gereken bilgileri belirtin. Varsayılan olarak, değerleri belirtmeniz için önerilen özellikler alt sınırı görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.

- c) **Oluştur**'u tıkkatın. Yeni konu yaratılır.
- Bir konuyu silmek için:
 - a) Silmek istediğiniz konunun yanındaki spanner simgesini  tıkkatın.
 - b) Kuyruk Düzenleme görünümünde **Konuyu sil**ögesini tıkkatın.
 - c) **Sil**seçeneğini tıkkatarak konuyu silmek istediğinizi onaylayın. Konu silindi.
- Bir konuya ilişkin özellikleri görüntülemek ve düzenlemek için:
 - a) Düzenlemek istediğiniz konunun yanındaki spanner simgesini  tıkkatın.
 - b) Düzenle düğmesini  tıkkatın
 - 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 MQ Gezgini belgelerindeki [Konu özellikleri](#) konusuna bakın.
 - d) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıkkatın.
- Bir konuyla ilgili bir iletiyi yayınlamak için en az bir eşleşen aboneliğiniz olmalıdır. Gerekirse, bir test aboneliği yaratabilirsiniz:
 - a) Konu listesinde yayınlanmasını istediğiniz konuyu tıkkatın.
 - b) Sonraki işleminiz, bu konunun eşleşen abonelikleri olup olmadığını bağıdır:
 - Eşleşen abonelik yoksa:
 - a. Eylemler düğmesini  tıkkatın ve **Konuyu test et**seçeneğini belirleyin.
 - b. Test düğmesini  tıkkatın. Test aboneliğine bir test iletisi yazılır.
 - Konu için eşleşen abonelikler varsa:
 - a. Abonelik adını tıkkatın.
 - b. Oluştur düğmesini  tıkkatın.
 - c. Yayınlamak istediğiniz iletiyi girin.
 - d. Put düğmesine  tıkkatın. İleti, eşleşen tüm aboneliklere yazılır.
- Bir konuya abone olmak için bkz. [“Aboneliklerle çalışma”](#) sayfa 103:
- Bir konuya ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Yetki kayıtlarını düzenlemek istediğiniz konunun yanındaki spanner simgesini () tıkkatın.
 - b) **Security** (Güvenlik) sekmesini tıkkatın.
 - c) Kuyruk yöneticisi yetki kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın, bkz. [“Kuyruk yöneticisi yetki kayıtlarıyla çalışılması”](#) sayfa 95.


V 9.2.0 **Aboneliklerle çalışma**

Abonelikleri eklemek ve silmek ve aboneliğin özelliklerini görüntülemek ve ayarlamak için IBM MQ Console ögesini kullanabilirsiniz.

Bu görev hakkında

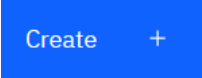
Abonelikler görünümü, belirli bir kuyruk yöneticisi için var olan abonelikleri listeler. Bir kuyruk yöneticisini tıklatıp **Abonelikler** sekmesini seçerek abonelikler listesine erişmenizi sağlar. Birlikte çalışmak için listeden tek tek abonelikler seçebilirsiniz.

Abonelikler hakkında daha fazla bilgi için [Aboneler ve abonelikler](#) ve [DEFINE SUB](#) başlıklı konuya bakın.


 z/OS' ta bir aboneliğe ilişkin yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

Yordam


- Abonelik eklemek için:

- Abonelikler** sekmesinde, oluştur düğmesini  tıklatın.
- Yönetilen ya da yönetilmeyen bir abonelik oluşturmak isteyip istemediğinizi seçin.
- Yaratmakta olduğunuz abonelik için gerekli bilgileri sağlayın. Varsayılan olarak, değerleri belirtmeniz için önerilen özellikler alt sınırı görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
- Oluştur**'u tıklatın. Yeni abonelik yaratılır.

- Bir aboneliği silmek için:

- Silmek istediğiniz aboneliğin yanındaki spanner simgesini  tıklatın.
- Kuyruk Düzenleme görünümünde **Aboneyi sil** ögesini tıklatın.
- Aboneliği silmek istediğinizi doğrulamak için **Sildü**ğmesini tıklatın. Abonelik silinir.

- Bir aboneliğin özelliklerini görüntülemek ve düzenlemek için:

- Düzenlemek istediğiniz aboneliğin yanındaki spanner simgesini  tıklatın.

- Düzenle düğmesini  tıklatın

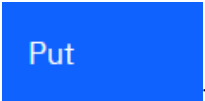
- Özellikleri gerektiği gibi düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir.
- Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.

- Konu aboneliğinin abone olduğu konuyla ilgili bir iletiyi yayınlamak için:

- Abonelik listesinde yayınlanmasını istediğiniz aboneliği tıklatın.

- Oluştur düğmesini  tıklatın.

- Yayınlamak istediğiniz iletiyi girin.

- Put düğmesine  tıklatın. İleti, yayınlamış olduğunuz konula eşleşen tüm aboneliklere yazılır.

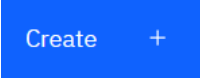



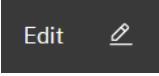


Dinleyicilerle çalışma

Dinleyici eklemek ya da silmek, dinleyici özelliklerini başlatmak ve durdurmak, dinleyici özelliklerini görüntülemek ve ayarlamak ve bir dinleyiciye ilişkin yetki kayıtlarını yönetmek için IBM MQ Console ' u kullanabilirsiniz.

Bu görev hakkında

İletişimciler görünümü, belirli bir kuyruk yöneticisi için varolan dinleyicileri görüntüler. Çalışmak için ayrı dinleyiciler seçebilirsiniz.

Yordam

- Dinleyici yaratmak için:
 - a) **İletişim** sekmesinde dinleyici görünümünün görüntülediğinden emin olun ve oluştur düğmesini  tıklatın.
 - b) Yaratmakta olduğunuz dinleyici için gereken bilgileri belirtin. Varsayılan olarak, değerleri belirtmeniz için önerilen özellikler alt sınırı görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
 - c) **Oluştur**'u tıklatın. Yeni dinleyici yaratılır.
- Bir dinleyici başlatmak için:
 - a) Listede başlatmak istediğiniz dinleyiciyi bulun.
 - b) Menüden **StartBaşlat** 'ı seçin  seçeneğini tıklatın.
- Bir dinleyiciyi durdurmak için:
 - a) Listede başlatmak istediğiniz dinleyiciyi bulun.
 - b)  menüsünden **Durdur** ögesini seçin.
- Bir dinleyiciye ilişkin özellikleri görüntülemek ve düzenlemek için:
 - a) Listedeki dinleyiciyi bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Özellikler** sekmesinin seçili olduğundan emin olun. Özellikleri düzenlemek için Düzenle düğmesini  tıklatın.
 - 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 belgelerindeki [Dinleyici özellikleri](#) başlıklı konuya 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. "[Kuyruk yöneticisi yetki kayıtlarıyla çalışılması](#)" sayfa 95.
- Bir dinleyiciyi silmek için:
 - a) Listedeki dinleyiciyi bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Dinleyiciyi sil** ögesini tıklatın.

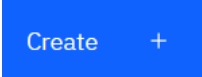




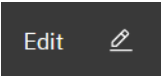
You can use the IBM MQ Console work with queue manager channels: you can add and delete queue manager channels, start and stop channels, reset and resolve channels, and ping channels. Ayrıca, bir kuyruk yöneticisi kanalının özelliklerini görüntüleyebilir ve ayarlayabilir ve kanala ilişkin yetki kayıtlarını yönetebilirsiniz.






Bu görev hakkında

Kuyruk yöneticisi kanalı, bir ağ üzerindeki kuyruk yöneticileri arasında ileti iletmek için kullanılan bir mantıksal iletişim bağlantısıdır. Kuyruk yöneticisi kanal görünümü, kaç kanalın çalıştırıldığını, kaç kez yeniden denendiğini ve kaç kişinin durdurulduğunu gösteren bir pano içerir.

z/OS 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 tıkklatın.
 - b) Yaratmak istediğiniz kuyruk yöneticisi kanalının tipini seçin ve sonraki düğmesine  tıkklatın.
 - c) Yaratmakta olduğunuz kanala ilişkin gerekli bilgileri sağlayın. Varsayılan olarak, değerleri belirtmeniz için önerilen özellikler alt sınırı görüntülenir. **Tüm kullanılabilir seçenekleri göster** seçeneğini belirleyerek tüm kullanılabilir özellikleri görüntüleyebilirsiniz.
 - d) **Oluştur**'u tıkklatın. Yeni kanal, **etkin değil** durumuyla oluşturulur.
- Bir kuyruk yöneticisi kanalını başlatmak için:
 - a) Listede başlatmak istediğiniz kanalı bulun.
 - b) Menüden **StartBaşlat** 'ı seçin  seçeneğini tıkklatın.
- Bir kuyruk yöneticisi kanalını durdurmak için:
 - a) Listede durmasını istediğiniz kanalı bulun.
 - b)  menüsünden **Durdur** ögesini seçin.
- Bir kuyruk yöneticisi kanalının özelliklerini görüntülemek için:
 - a) Listede kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Özellikler** sekmesinin seçili olduğundan emin olun. Özellikleri düzenlemek için Düzenle düğmesine  tıkklatın.
 - 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 belgelerindeki [Kanal özellikleri](#) konusuna bakın.
 - e) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıkklatın.
- Bir kuyruk yöneticisi kanalını ilk durumuna getirmek için:
 - a) Listede kanalı bulun.


- b) Menüden **Gelişmiş** ' i seçin  .
- c) **Reset** (İlk Duruma Getir) bölümünde bir ileti sıra numarası belirtin.
Başlatılamazsa, bir kanalı ilk durumuna getirmeniz gerekir; çünkü iki uç, gönderilecek bir sonraki iletinin sıra numarasını kabul etmemektedir. İleti sıra numarası bu numarayı belirtir.
- d) **Kanalı Sıfırla** öğesini tıklatın.
- Bir gönderici ya da sunucu kanalını çözmek için:
 - a) Listede kanalı bulun.
 - b) Menüden **Gelişmiş** ' i seçin  .
 - c) **Çözümle** kısmında, **İletileri iletim kuyruğuna geri yükle** ya da **İletileri at** öğesini tıklatarak yürürlükteki ileti grubunun kesinleştirilip kesinleştirilmeyeceğini ya da geri alınmayacağını seçin.
- Bir kuyruk yöneticisi kanalına ping komutu göndermek için:
 - a) Listede kanalı bulun.
 - b)  menüsünden **Ping** öğesini seçin.
- Bir kuyruk yöneticisi kanalına ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Listede kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin  .
 - c) **Security** (Güvenlik) sekmesini tıklatın.
 - d) Kuyruk yöneticisi yetki kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın, bkz. [“Kuyruk yöneticisi yetki kayıtlarıyla çalışılması” sayfa 95.](#)
- Bir kuyruk yöneticisi kanalını silmek için:
 - a) Listede kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin  .
 - c) **Kanalı sildüğmesini** tıklatın.

V 9.2.0 *Uygulama kanallarıyla çalışma*

Uygulama kanallarıyla çalışmak için IBM MQ Console ' ı kullanabilirsiniz: Kanal ekleyebilir ve silebilir, kanalları başlatabilir ve durdurabilir, kanalları sıfırlayabilir ve çözebilir, kanalları ve ping kanallarını çözebilirsiniz. Ayrıca, bir uygulama kanalının özelliklerini görüntüleyebilir ve ayarlayabilir ve kanala ilişkin yetki kayıtlarını yönetebilirsiniz.

Bu görev hakkında

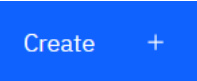
Uygulama kanalı, uygulamalar tarafından bir ağ üzerinde kuyruk yöneticisine bağlanmak için kullanılan mantıksal bir iletişim bağlantısıdır. Uygulama kanalı görünümü, kaç kanalın çalıştırıldığını, kaç kez yeniden denendiğini ve kaç kişinin durdurulduğunu gösteren bir pano içerir.

 **z/OS** 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  tıklatın.


b) Sonraki düğmeyi  tıklatın.

c) Yaratmakta olduğunuz kanala ilişkin gerekli bilgileri sağlayın. Varsayılan olarak, değerleri belirtmeniz için önerilen özellikler alt sınırı 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 **StartBaşlat** 'ı seçin  seçeneğini tıklatın.


• Bir uygulama kanalını durdurmak için:

a) Listede durmasını istediğiniz kanalı bulun.

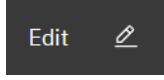
b)  menüsünden **Durdur** ögesini seçin.

• Bir uygulama kanalının özelliklerini görüntülemek için:

a) Listede kanalı bulun.

b) Menüden **Yapılandır** seçeneğini belirleyin .

c) **Özellikler** sekmesinin seçili olduğundan emin olun. Özellikleri düzenlemek için Düzenle düğmesini

 tıklatın.

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 belgelerindeki [Kanal özellikleri](#) konusuna bakın.

e) Yaptığınız değişiklikleri saklamak için **Tamam** düğmesini tıklatın.

• Bir uygulama kanalını ilk durumuna getirmek için:

a) Listede kanalı bulun.

b) Menüden **Gelişmiş** 'i seçin .

c) **Reset** (İlk Duruma Getir) bölümünde bir ileti sıra numarası belirtin.

Başlatılamazsa, bir kanalı ilk durumuna getirmeniz gerekir; çünkü iki uç, gönderilecek bir sonraki iletinin sıra numarasını kabul etmemektedir. İleti sıra numarası bu numarayı belirtir.

d) **Kanalı Sıfırla** ögesini tıklatın.

• Bir gönderici ya da sunucu kanalını çözmek için:




a) Listede kanalı bulun.

b) Menüden **Gelişmiş** 'i seçin .

c) **Çözümle** kısmında, **İletileri iletim kuyruğuna geri yükle** ya da **İletileri at** ögesini tıklatarak yürürlükteki ileti grubunun kesinleştirilip kesinleştirilmeyeceğini ya da geri alınmayacağını seçin.

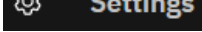
• Bir kanala ping komutu göndermek için:

a) Listede kanalı bulun.

- b)  menüsünden **Ping** ögesini seçin.
- Bir uygulama kanalına ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Listede kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Security** (Güvenlik) sekmesini tıklayın.
 - d) Kuyruk yöneticisi yetki kayıtları için açıklandığı gibi yetki kayıtlarıyla çalışın, bkz. [“Kuyruk yöneticisi yetki kayıtlarıyla çalışılması” sayfa 95.](#)
- Bir uygulama kanalını silmek için:
 - a) Listede kanalı bulun.
 - b) Menüden **Yapılandır** seçeneğini belirleyin .
 - c) **Kanalı sildüğmesini** tıklayın.

Web konsolu ayarları

Yeni Web konsolu için bazı genel ayarlar belirleyebilirsiniz.

Web konsolu ayarları görünümüne geçmek için ayarlar simgesini  tıklayın.

Aşağıdaki özellikleri denetlemek için ayarları kullanın:

- Kuyruk yöneticilerinin her on saniyede bir otomatik olarak yenilenmesi. Bu özellik açılabilir ya da kapatılabilir.
- Sistem nesnelere görüntülenip görüntülenmeyeceği. Bunu tüm nesne tipleri için belirleyebilir ya da nesne tiplerini tek tek seçebilirsiniz.
- İzleme bilgilerinin toplanıp toplanmayacağı.

Konsol tipleri arasında geçiş yapma



You can switch between the New Web Console (which is the default web console for IBM MQ 9.2) and the Dashboard Web Console (which was the web console for earlier versions of IBM MQ).

Bu görev hakkında

You switch between consoles by using the **setmqweb** command.

Dashboard Web Console'ı kullanmaya geri dönseniz, IBM MQ 9.1 belgelerinde [Dashboard Web Konsolu](#) 'e danışın. Kullanım yönergeleri için.

Bu görevi tamamlamak için, **dspmqweb** ve **setmqweb** komutlarını kullanabilmek üzere belirli ayrıcalıklara sahip bir kullanıcı olmanız gerekir:

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



Uyarı:

z/OS üzerinde **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, WLP_USER_DIR ortam değişkenini, değişken mqweb sunucusu yapılandırmanızı gösterecek şekilde ayarlamanız gerekir.

Bunu yapmak için aşağıdaki komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

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

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

Daha fazla bilgi için bkz. [mqweb sunucusu yaratılması](#).

Yordam

- Dashboard Web Console' e geçmek için aşağıdaki komutları kullanın:
 - Aşağıdaki komutu girerek mqConsoleEnableUnsafeYerleşik özelliğini true değerine ayarlayın:

```
setmqweb properties -k mqConsoleEnableUnsafeInline -v true
```

Bu özelliğin ayarlanmasını, CSP kural kümesini güvenli olmayan yerleşik etkinleştirmek için ayarlamaya dikkat edin; bu, yapılandırmanızı daha az güvenli hale getirebilecek, ancak Dashboard Web Console tarafından zorunlu kılınmaktadır.

- Switch the mqweb server to use the Dashboard Web Console by entering the following command:

```
setmqweb properties -k mqConsoleEarName -v com.ibm.mq.console
```

- mq web sunucusuna ilişkin varsayılan ayarlara geri dönmek için (New Web Console' ı içerir), aşağıdaki komutu kullanın:

```
setmqweb properties -r
```

- To revert to the New Web Console while preserving any other customizations you have made to the mqweb server, use the following commands:

- Aşağıdaki komutu girerek güvenlik yapılandırmasını geri yüklemek için mqConsoleEnableUnsafeYerleşik özelliğini false (yanlış) değerine ayarlayın:

```
setmqweb properties -k mqConsoleEnableUnsafeInline -v false
```

- Switch the mqweb server to use the New Web Console by entering the following command:

```
setmqweb properties -k mqConsoleEarName -v com.ibm.mq.webconsole
```

İlgili başvurular

[setmqweb](#)

Linux

Windows

IBM MQ Explorer komutunu kullanarak yönetim

IBM MQ Explorer , ağınızın yerel ya da uzak denetimini yalnızca Windows ya da Linux x86-64 çalıştıran bir bilgisayardan gerçekleştirmenizi sağlar.

IBM MQ for Windows and IBM MQ for Linux x86-64 provide an administration interface called the IBM MQ Explorer to perform administration tasks as an alternative to using control or MQSC commands. [Komut kümelerini karşılaştırma](#) , IBM MQ Explorer komutunu kullanarak yapabildiğinizi gösterir.

The IBM MQ Explorer allows you to perform local or remote administration of your network from a computer running Windows, or Linux x86-64, by pointing the IBM MQ Explorer at the queue managers and clusters you are interested in. Desteklenen herhangi bir platform z/OS dahilinde çalışmakta olan kuyruk yöneticilerine uzaktan bağlanabilir, tüm ileti sistemi omurgasının konsoldan görüntülenmesine, incelenmesine ve değiştirilmesine olanak sağlar.

To configure remote IBM MQ queue managers so that IBM MQ Explorer can administer them, see [“IBM MQ Explorer için önkoşul olan yazılımlar ve tanımlar” sayfa 113](#).

Bu, genellikle Windows ya da Linux x86-64 sistem etki alanı içinde yerel olarak ya da uzaktan IBM MQ için çalışma ortamının ayarlanması ve ayarlanması ile ilişkili görevleri gerçekleştirmenize olanak sağlar.

Linux' ta IBM MQ Explorer , birden fazla Eclipse kurulumunuz varsa başlatılamayabilir. Bu gerçekleşirse, diğer Eclipse kurulumu için kullandığınız bir kullanıcı kimliğini kullanarak IBM MQ Explorer ' ı başlatın.

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

IBM MQ Explorer can be installed either as part of the product installation (see [IBM MQ ürününün kurulması ve kaldırılması](#)), or from the stand-alone IBM MQ Explorer Fix Central ' tan kullanılabilir yükleme (bkz. [Installing and uninstalling IBM MQ Explorer as a stand-alone application on Linux and Windows](#)).

Linux

Windows

IBM MQ Explorer ile yapabildiğiniz

Bir dizi İçerik Görünümü ve Özellik iletişim kutusunu kullanarak yönetim görevlerini gerçekleştirmek için IBM MQ Explorer komutunu kullanabilirsiniz. IBM MQ Explorer ögesini, bir ya da daha çok Eclipse eklentisi yazarak da genişletebilirsiniz.

IBM MQ Explorer görevler

IBM MQ Explorer ile birlikte aşağıdaki görevleri gerçekleştirebilirsiniz:

- Bir kuyruk yöneticisi yaratın ve [silin](#) (yalnızca yerel makinenizde).
- [Kuyruk yöneticisini başlatma ve durdurma](#) (yalnızca yerel makineniz için).
- [IBM MQ nesnelere tanımlarını tanımlama, görüntüleme ve değiştirme](#) , kuyruklar ve kanallar gibi.
- [Kuyruklardaki iletilere göz atın](#).
- [Bir kanalı başlatma ve durdurma](#).
- Bir kanal, dinleyici, kuyruk ya da hizmet nesnelere ilişkin [durum bilgilerini görüntüleyin](#) .
- Bir kümedeki kuyruk yöneticilerini [görüntüleyin](#).
- [hangi uygulamaların, kullanıcıların ya da kanalların belli bir kuyruğu açık' i görmek için işaretleyin](#).
- [Yeni Küme Yarat sihirbazını kullanarak yeni bir kuyruk yöneticisi kümesi yaratın](#) .
- [Bir kümeye kuyruk yöneticisi eklenmesi Kuyruk Yöneticisi Ekle sihirbazını kullanarak](#).
- [Kimlik doğrulama bilgileri nesnesini yönetmenizi sağlar](#), Transport Layer Security (TLS) kanal güvenliği ile kullanılır.
- Kanal başlatıcıları, tetikleyici izleyicileri ve dinleyicileri yaratmanızı ve silmesini sağlar.
- [Komut sunucuları, kanal başlatıcıları, tetikleme izleyicileri ve dinleyiciler' i başlatın ya da durdurun](#).
- Belirli hizmetleri [bir kuyruk yöneticisi başlatıldığında otomatik olarak başlatılarak ayarlayın](#).
- [Kuyruk yöneticilerinin özelliklerini değiştirin](#).
- [Yerel varsayılan kuyruk yöneticisini değiştirme](#).
- [TLS sertifikalarını yönetmek, sertifikaları kuyruk yöneticileriyle ilişkilendirmek ve sertifika depolarını yapılandırmak ve ayarlamak \(yalnızca yerel makinenizde\) için \[strmqikm\]\(#\) \(ikeyman\) GUI 'sini çağırın seçeneğini tıklayın](#).
- [IBM MQ nesnelere JMS nesnelere oluşturun ve JMS nesnelere IBM MQ nesnelere](#).
- [Şu anda desteklenen tiplerden herhangi biri için JMS Connection Factory yarat](#) .
- Bir dinleyiciye ilişkin TCP kapı numarası ya da kanal başlatıcı kuyruğu adı gibi, herhangi bir hizmete ilişkin parametreleri [değiştirin](#).
- [Hizmet izleme işlemini başlatın ya da durdurun](#).

İçerik Görünümleri ve Özellik İletişim Kutuları

Bir dizi İçerik Görünümü ve Özellik iletişim kutusunu kullanarak yönetim görevlerini gerçekleştirmenizi sağlar.

İçerik Görünümü

İçerik Görünümü, aşağıdakileri görüntüleyebilen bir panodur:

- Öznitelikler ve IBM MQ ile ilgili yönetim seçenekleri.
- Öznitelikler ve bir ya da daha çok ilgili nesnelere ilişkin yönetim seçenekleri.
- Bir kümeyle ilişkili öznitelikler ve yönetim seçenekleri.

Özellik iletişim kutuları

Özellik iletişim kutusu, bir dizi alan içindeki bir nesneye ilişkin öznitelikleri görüntüleyen bir panodur.

You navigate through the IBM MQ Explorer using the Navigator view. Navigator , gerek duyduğunuz İçerik Görünümlerini seçmenize olanak tanır.

IBM MQ Explorer' in Genişletmesi

IBM MQ Explorer , bilgileri Eclipse çerçevesinin ve Eclipse ' in desteklediği diğer eklenti uygulamalarının bir stiliyle tutarlı bir şekilde sunar.

Through extending the IBM MQ Explorer, system administrators have the ability to customize the IBM MQ Explorer to improve the way they administer IBM MQ.

Daha fazla bilgi için bakınız: [Extending MQ Explorer](#).

IBM MQ Explorer' in kullanılmasının belirlenmesine karar verilmesi

Kuruluşunuzda IBM MQ Explorer ' in kullanılmasının belirlenmesine karar verilirken, bu konuda listelenen bilgileri göz önünde bulundurun.

Aşağıdaki noktalardan haberdar olmanız gerekir:

nesne adları

If you use lowercase names for queue managers and other objects with the IBM MQ Explorer, when you work with the objects using MQSC commands, you must enclose the object names in single quotation marks, or IBM MQ does not recognize them.

Büyük kuyruk yöneticileri

The IBM MQ Explorer works best with small queue managers. Tek bir kuyruk yöneticisinde çok sayıda nesne varsa, IBM MQ Explorer görünümü bir görünümde sunmak için gereken bilgileri çıkartırken gecikmeler yaşayabilirsiniz.

Kümelere

IBM MQ kümeleri potansiyel olarak yüzlerce ya da binlerce kuyruk yöneticisi içerebilir. IBM MQ Explorer , kuyruk yöneticilerini bir ağaç yapısı kullanarak bir kümede sunar. The physical size of a cluster does not affect the speed of the IBM MQ Explorer dramatically because the IBM MQ Explorer does not connect to the queue managers in the cluster until you select them.

IBM MQ Explorer' in ayarlanması

Bu bölüm, IBM MQ Explorer' yi ayarlamak için yapmanız gereken adımları özetlemektedir.

- [“IBM MQ Explorer için önkoşul olan yazılımlar ve tanımlar” sayfa 113](#)
- [“IBM MQ Explorer güvenliği” sayfa 113](#)
- [“Showing and hiding queue managers and clusters in IBM MQ Explorer” sayfa 116](#)
- [“Küme üyeliği ve IBM MQ Explorer” sayfa 117](#)
- [“IBM MQ Explorer için veri dönüştürme” sayfa 118](#)

IBM MQ Explorer için önkoşul olan yazılımlar ve tanımlar

IBM MQ Explorer' i kullanmaya çalışmadan önce aşağıdaki gereksinimleri karşıladığınızdan emin olun.

IBM MQ Explorer , yalnızca TCP/IP iletişim protokolünü kullanarak uzak kuyruk yöneticilerine bağlanabilir.

Şuna bakın:

1. Her uzaktan yönetilen kuyruk yöneticisinde bir komut sunucusu çalışıyor.
2. Her uzak kuyruk yöneticisinde uygun bir TCP/IP dinleyici nesnesi çalışıyor olmalıdır. Bu nesne, IBM MQ dinleyici ya da AIX and Linux sistemlerinde, inetd yardımcı programı olabilir.
3. Varsayılan olarak SYSTEM.ADMIN.SVRCONN, tüm uzak kuyruk yöneticilerinde bulunur.

Aşağıdaki MQSC komutunu kullanarak kanalı yaratabilirsiniz:

```
DEFINE CHANNEL(SYSTEM.ADMIN.SVRCONN) CHLTYPE(SVRCONN)
```

Bu komut, temel bir kanal tanımlaması yaratır. Daha karmaşık bir tanım (örneğin, güvenlik ayarlamak için) istiyorsanız, ek parametrelere gerek duyarsınız. Ek bilgi için [DEFINE CHANNEL](#) başlıklı konuya bakın.

4. Sistem kuyruğu, SYSTEM.MQEXPLORER.REPLY.MODEL(MODEL) var olmalıdır.

IBM MQ Explorergüvenliği

If you are using IBM MQ in an environment where it is important for you to control user access to particular objects, you might need to consider the security aspects of using the IBM MQ Explorer.

IBM MQ Explorer' i kullanma yetkisi

Herhangi bir kullanıcı IBM MQ Explorer' i kullanabilir, ancak kuyruk yöneticilerini bağlamak, bunlara erişmek ve yönetmek için bazı yetkiler gereklidir.

To perform local administrative tasks using the IBM MQ Explorer, a user is required to have the necessary authority to perform the administrative tasks. Kullanıcı, mqm grubunun bir üyesiye, kullanıcının tüm yerel yönetim görevlerini gerçekleştirme yetkisi vardır.

To connect to a remote queue manager and perform remote administrative tasks using the IBM MQ Explorer, the user executing the IBM MQ Explorer is required to have the following authorities:

- Hedef kuyruk yöneticisi nesnesindeki CONNECT yetkisi
- Hedef kuyruk yöneticisi nesnesi üzerinde SORGULAMA YETKISI
- Hedef kuyruk yöneticisi nesnesi için GÖRÜNTÜLEME yetkisi
- Kuyruk için SORGULAMA YETKISI, SYSTEM.MQEXPLORER.REPLY.MODEL
- Kuyruk için GÖRÜNTÜLEME yetkisi, SYSTEM.MQEXPLORER.REPLY.MODEL
- INPUT (alma) (get) yetkisi, SYSTEM.MQEXPLORER.REPLY.MODEL
- OUTPUT (put) yetkisi kuyruğa, SYSTEM.MQEXPLORER.REPLY.MODEL
- OUTPUT (put) yetkisi kuyruğa, SYSTEM.ADMIN.COMMAND.QUEUE
- Kuyruk, SYSTEM.ADMIN.COMMAND.QUEUE
- Seçilen işlemi gerçekleştirme yetkisi

Not: INPUT yetkisi, bir kuyruktan kullanıcı girişiyle ilgilidir (bir alma işlemi). OUTPUT yetkisi, kullanıcının bir kuyruktan çıkışa (put işlemi) ilişkindir.

To connect to a remote queue manager on IBM MQ for z/OS and perform remote administrative tasks using the IBM MQ Explorer, the following must be provided:

- Sistem kuyruğu için bir RACF profili, SYSTEM.MQEXPLORER.REPLY.MODEL
- Kuyruklara ilişkin bir RACF tanıtımı, AMQ.MQEXPLORER.*

Buna ek olarak, IBM MQ Explorer adlı kullanıcının aşağıdaki yetkilerine sahip olması gerekir:

- Sistem kuyruğu için RACF UPDATE yetkinizin, SYSTEM.MQEXPLORER.REPLY.MODEL
- RACF UPDATE authority to the queues, AMQ.MQEXPLORER.*
- Hedef kuyruk yöneticisi nesnesindeki CONNECT yetkisi
- Seçilen işlemi gerçekleştirme yetkisi
- MQCMDS sınıfındaki tüm hlq.DISPLAY.object profillerine ilişkin okuma yetkisi

IBM MQ nesnelere yetki verilmesine ilişkin bilgi için bkz. [AIX, Linux, and Windows sistemlerindeki bir IBM MQ nesnesine erişim verilmesi](#).

Bir kullanıcı, gerçekleştirme yetkisine sahip olmadığı bir işlemi gerçekleştirmeye çalışırsa, hedef kuyruk yöneticisi yetkilendirme hatası yordamlarını çağırır ve işlem başarısız olur.

IBM MQ Explorer içindeki varsayılan süzgeç, tüm IBM MQ nesnelere görüntülemektedir. Kullanıcının GÖRÜNTÜLEME yetkisi olmayan herhangi bir IBM MQ nesnesi varsa, yetki hataları oluşur. Yetki olayları kaydediliyorsa, görüntülenen nesne aralığını, kullanıcının GÖRÜNTÜLEME yetkisine sahip olduğu nesnelere sınırlayın.

IBM MQ Explorer' tan uzak kuyruk yöneticilerine bağlanmak için güvenlik

Kanalı, IBM MQ Explorer ve her uzak kuyruk yöneticisi arasında güvenli bir şekilde sağlamalısınız.

IBM MQ Explorer uzak kuyruk yöneticilerine MQI istemci uygulaması olarak bağlanır. Bu, her uzak kuyruk yöneticisinin bir sunucu bağlantı kanalı tanımlaması ve uygun bir TCP/IP dinleyicisinin olması gerektiği anlamına gelir. Sunucu bağlantı kanalınızı güvenceye almezseniz, kötü amaçlı bir uygulamanın aynı sunucu bağlantısı kanalına bağlanması ve kuyruk yöneticisi nesnelere sınırsız yetkisi elde edilmeleri mümkündür. Sunucu bağlantı kanalınızı güvenceye almak için, kanalın MCAUSER özniteliği için boş olmayan bir değer belirleyin, kanal 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 belirtirseniz, bu kanalı kullanarak kuyruk yöneticisiyle bağlantı kuran tüm programlar, adı belirtilen kullanıcının kimliğiyle çalışır ve aynı yetki düzeyine sahip olur. Kanal kimlik doğrulaması kayıtları kullanıyorsanız bu durum gerçekleşmez.

Using a security exit with the IBM MQ Explorer

IBM MQ Explorer komutunu kullanarak, varsayılan bir güvenlik çıkışı ve kuyruk yöneticisi özel güvenlik çıkışları belirtebilirsiniz.

Varsayılan bir güvenlik çıkışı tanımlayabilirsiniz; bu çıkış, IBM MQ Explorer' den gelen tüm yeni istemci bağlantıları için kullanılabilir. Bu varsayılan çıkış, bağlantı yapıldığında geçersiz kılınabilir. Ayrıca, tek bir kuyruk yöneticisi ya da bir kuyruk yöneticisi kümesi için bir güvenlik çıkışı tanımlayabilir ve bu da bir bağlantı yapıldığında yürürlüğe girer. Çıkışları, IBM MQ Explorer komutunu kullanarak belirtirsiniz. Daha fazla bilgi için IBM MQ Explorer Yardım 'a bakın.

TLS 'nin etkin olduğu MQI kanallarını kullanarak uzak kuyruk yöneticisine bağlanmak için IBM MQ Explorer ' in kullanılması



IBM MQ Explorer , bir MQI kanalını kullanarak uzak kuyruk yöneticilerine bağlanır. MQI kanalının TLS güvenliğini kullanarak güvenliğini sağlamak istiyorsanız, bir istemci kanal tanımlama çizelgesini kullanarak kanalı oluşturmanız gerekir.

İstemci kanalı tanımlama çizelgesi kullanılarak bir MQI kanalının nasıl oluşturulacağı hakkında bilgi için bkz. [IBM MQ MQI clients' a Genel Bakış](#).

When you have established the channel using a client channel definition table, you can use the IBM MQ Explorer to connect to a remote queue manager using TLS-enabled MQI channel, as described in [“Sistemdeki uzak kuyruk yöneticisini barındıran görevler” sayfa 114](#) and [“Tasks on the system that hosts the IBM MQ Explorer” sayfa 115](#).

Sistemdeki uzak kuyruk yöneticisini barındıran görevler

Uzak kuyruk yöneticisini bulandıran sistemde, aşağıdaki görevleri gerçekleştirin:

1. Bir sunucu bağlantısı ve istemci bağlantı çifti tanımlayın ve her iki kanalda da sunucu bağlantısında *SSLCIPH* özneliği için uygun değeri belirtin. *SSLCIPH* özneliğe ilişkin daha fazla bilgi için bkz. [Kanalları TLS ile koruma](#).
2. Kuyruk yöneticisinin @ipcc dizininde bulunan AMQCLCHL .TABkanal tanımlama tablosunu IBM MQ Explorer' i barındıran sisteme gönderin.
3. Belirlenen kapı üzerinde bir TCP/IP dinleyicisi başlatın.
4. CA ve kişisel TLS sertifikalarını, kuyruk yöneticisinin SSL dizinine yerleştirin:
 -  /var/mqm/qmgrs/+QMNAME+/SSL for AIX or Linux systems.
 -  Windows sistemleri için C:\Program Files\IBM\MQ\qmgrs\+QMNAME+\SSL .

Burada +QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.
5. Create a key database file of type CMS named key . kdb. Parolayı, **strmqikm** (iKeyman) içindeki seçeneği işaretleyerek bir dosyaya saklayın. GUI ' yi ya da -zu1a komutunu **runmqckm** komutuyla ya da **runmqakm** komutuyla birlikte kullanarak.
6. Önceki adımda yaratılan anahtar veritabanına sertifika kuruluşu (CA) sertifikalarını ekleyin.
7. Kuyruk yöneticisi için kişisel sertifikayı anahtar veritabanına içe aktarın.

For more detailed information about working with TLS on Windows systems, see [AIX, Linux, and Windows üzerinde TLS ile çalışma](#).

Tasks on the system that hosts the IBM MQ Explorer

IBM MQ Explorer' i bulunduran sistemde aşağıdaki görevleri gerçekleştirin:

1. key . jksadlı JKS tipinde bir anahtar veritabanı dosyası yaratın. Bu anahtar veri tabanı dosyası için bir parola belirleyin.

IBM MQ Explorer , TLS güvenliği için Java anahtar deposu dosyalarını (JKS) kullanır; dolayısıyla, IBM MQ Explorer için TLS ' yi yapılandırmak için kullanılan anahtar deposu dosyası bununla eşleşmelidir.
2. Önceki adımda yaratılan anahtar veritabanına sertifika kuruluşu (CA) sertifikalarını ekleyin.
3. Kuyruk yöneticisi için kişisel sertifikayı anahtar veritabanına içe aktarın.
4. Windows ve Linux sistemlerinde, sistem menüsünü, MQExplorer yürütülür dosyasını ya da **strmqcfcg** komutunu kullanarak IBM MQ Explorer ' yi başlatın.
5. IBM MQ Explorer araç çubuğundan, **Window-> Preferences**(Pencere-Tercihler) öğelerini tıklattıktan sonra **IBM MQ Explorer ' i genişletin** ve **SSL Client Certificate Stores** öğesini tıklatın. Enter the name of, and password for, the JKS file created in step 1 of “[Tasks on the system that hosts the IBM MQ Explorer](#)” sayfa 115, in both the Trusted Certificate Store and the Personal Certificate Store, then click **Tamam**.
6. **Tercihler** penceresini kapatın ve **Kuyruk Yöneticileri ' yi farenin sağ düğmesiyle tıklatın**. **Kuyruk Yöneticilerini Göster/Gizle** öğesini tıklattıktan sonra **Kuyruk Yöneticilerini Göster/Gizle** ekranındaki **Ekle** düğmesini tıklatın.
7. Kuyruk yöneticisinin adını yazın ve **Doğrudan bağlan** seçeneğini belirleyin. Next (İleri) düğmesini tıklatın.
8. **İstemci kanal tanımlama çizelgesini kullan (CCDT)** öğesini seçin ve uzak kuyruk yöneticisini bulunduran sistem üzerindeki “[Sistemdeki uzak kuyruk yöneticisini barındıran görevler](#)” sayfa 114 içindeki 2. adımda uzak kuyruk yöneticisinden aktardığınız kanal çizelge dosyasının yerini belirtin.
9. **Son** düğmesini tıklatın. Artık uzak kuyruk yöneticisine IBM MQ Explorer adresinden erişebilirsiniz.

IBM MQ Explorer ile başka bir kuyruk yöneticisiyle bağlantı kurulması

IBM MQ Explorer , ara kuyruk yöneticisi aracılığıyla bir kuyruk yöneticisine bağlanmanıza olanak tanır; bu yönetim kurulu IBM MQ Explorer ' un zaten bağlı olduğu bir kuyruk yöneticisine bağlanır.

Bu durumda, IBM MQ Explorer , PCF komut iletilerini ara kuyruk yöneticisine koyar ve aşağıdaki bilgileri belirtir:

- Hedef kuyruk yöneticisinin adı olarak, nesne tanımlayıcısındaki (MQOD) *ObjectQMGrAd* parametresi. Kuyruk adı çözümlemesine ilişkin ek bilgi için [Ad çözünürlüğü](#) başlıklı konuya bakın.
- İleti tanımlayıcısındaki (MQMD) *UserIdentifier* değıştirgesi yerel userId olarak.

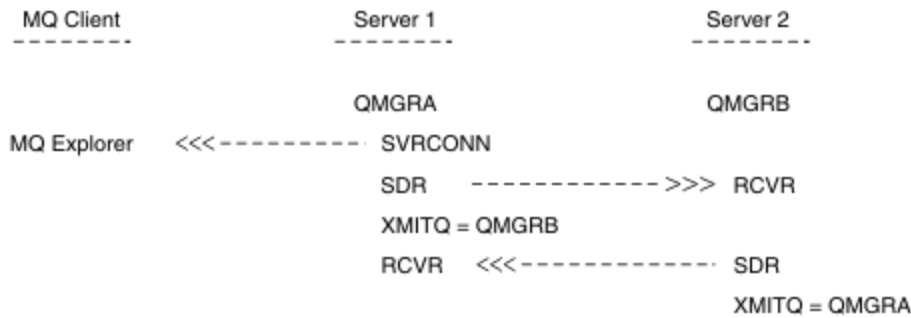
Baęlantı daha sonra bir ara kuyruk yöneticisi aracılığıyla hedef kuyruk yöneticisine baęlanmak için kullanıldıysa, userId ileti tanımlayıcısının (MQMD) *UserIdentifier* parametresine yeniden aktarılır. Hedef kuyruk yöneticisinde MCA dinleyicisinin bu iletiyi kabul etmesi için, MCAUSER öznitelięinin ayarlanması gerekir ya da userId önceden koyma yetkisine sahip olmalıdır.

Hedef kuyruk yöneticisinde bulunan komut sunucusu, ileti tanımlayıcısındaki (MQMD) *UserIdentifier* değıştirgesinde userId değerini belirten iletileri iletim kuyruęuna yerleřtiriyor. Bu durumda, userId ' nin başarılı olması için hedef kuyruk yöneticisinde yer alan yetkiyle önceden var olması gerekir.

Ařaęıdaki örnek, bir ara kuyruk yöneticisi aracılığıyla bir kuyruk yöneticisini IBM MQ Explorer' e nasıl baęlayacaęını göstermektedir.

Kuyruk yöneticisiyle uzak bir denetim baęlantısı kurun. Ařaęıdakilerden emin olun:

- Sunucudaki kuyruk yöneticisi etkindir ve tanımlı bir sunucu baęlantısı kanalı (SVRCONN) vardır.
- Dinleyici etkindir.
- Komut sunucusu etkin.
- SYSTEM.MQ EXPLORER.REPLY.MODEL kuyruęu yaratıldı ve yeterli yetkiye sahip olduęunuz bir kuyruk.
- Kuyruk yöneticisi dinleyicileri, komut sunucuları ve gönderen kanalları bařlatılır.



Bu örnekte:

- IBM MQ Explorer is connected to queue manager QMGRA (running on Server1) using a client connection.
- Queue manager QMGRB on Server2 can be now connected to IBM MQ Explorer through an intermediate queue manager (QMGRA)
- When connecting to QMGRB with IBM MQ Explorer, select QMGRA as the intermediate queue manager

In this situation, there is no direct connection to QMGRB from IBM MQ Explorer; the connection to QMGRB is through QMGRA.

Server2 üzerindeki kuyruk yöneticisi QMGRB , gönderen alıcı kanallarını kullanarak Server1 üzerinde QMGRA ' a baęlı. The channel between QMGRA and QMGRB must be set up in such a way that remote administration is possible; see [“Uzak denetim için kuyruk yöneticilerinin yapılandırılması” sayfa 181.](#)

Showing and hiding queue managers and clusters in IBM MQ Explorer

IBM MQ Explorer , aynı anda birden çok kuyruk yöneticisi görüntüleyebilir. Kuyruk Yöneticisi ağaç düęümüne ilişkin menüden seçilebilir/Gizle/Gizle (Kuyruk Yöneticisi ağaç düęümü menüsünden seçilebilir), başka bir (uzak) makine ile ilgili bilgileri görüntüleyip görüntülemeyeceğini seçebilirsiniz. Yerel kuyruk yöneticileri otomatik olarak algılanır.

Uzak kuyruk yöneticisini göstermek için:

1. **Kuyruk Yöneticileri** ağaç düęümünü sağ tıklatın ve **Kuyruk Yöneticilerini Göster/Gizle** seçeneğini belirleyin.

2. **Ekle**'yi tıkklatın. Kuyruk Yöneticileri Göster/Gizle panosu görüntülenir.
3. Sağlanan alanlardaki uzak kuyruk yöneticisinin adını ve anasistem adını ya da IP adresini girin.
Anasistem adı ya da IP adresi, uzak kuyruk yöneticisiyle bağlantı kurmak için varsayılan sunucu bağlantı kanalı olan SYSTEM.ADMIN.SVRCONNya da kullanıcı tanımlı bir sunucu bağlantısı kanalı.
4. **Bitir**'i tıkklatın.

Kuyruk Yöneticileri Göster/Gizle panosunda, tüm görünür kuyruk yöneticilerinin bir listesi de görüntülenir. Kuyruk yöneticilerini gezinme görünümünden gizlemek için bu panoyu kullanabilirsiniz.

IBM MQ Explorer , bir kümenin üyesi olan bir kuyruk yöneticisi görüntülense, küme algılanır ve otomatik olarak görüntülenir.

Uzak kuyruk yöneticileri listesini bu panodan dışa aktarmak için:

1. Kuyruk Yöneticileri Göster/Gizle panosunu kapatın.
2. Right-click the highest **IBM MQ** tree node in the Navigation pane of the IBM MQ Explorer, then select **Dışa Aktarma IBM MQ Explorer Ayarları**
3. **IBM MQ Explorer > IBM MQ Explorer Settings**öğesini tıkklatın.
4. **Bağlantı Bilgileri > Uzak kuyruk yöneticileri**öğelerini seçin.
5. Dışa aktarılan ayarları saklamak için bir dosya seçin.
6. Son olarak, uzak kuyruk yöneticisi bağlantı bilgilerini belirtilen kütüle dışa aktarmak için **Son** düğmesini tıkklatın.

Uzak kuyruk yöneticilerinin listesini almak için:

1. IBM MQ Explorer' un Gezinme bölmesinde en yüksek **IBM MQ** ağaç düğümünü sağ tıkklatın ve **IBM MQ Explorer Ayarlarını İçe Aktar**seçeneğini belirleyin.
2. **IBM MQ Explorer > IBM MQ Explorer Settings**öğesini tıkklatın.
3. **Göz At**düğmesini tıkklatın ve uzak kuyruk yöneticisi bağlantı bilgilerini içeren dosyanın yoluna gidin.
4. **Aç**'i tıkklatın. Dosya uzak kuyruk yöneticilerinin bir listesini içeriyorsa, **Bağlantı Bilgileri > Uzak kuyruk yöneticileri** kutusu seçilidir.
5. Son olarak, uzak kuyruk yöneticisi bağlantı bilgilerini IBM MQ Explorer' e (içe) aktarmak için **Son** düğmesini tıkklatın.

Küme üyeliği ve IBM MQ Explorer

IBM MQ Explorer , bir kümenin üyesi olan kuyruk yöneticilerine ilişkin bilgi gerektirir.

Kuyruk yöneticisi bir kümenin üyesiye, küme ağacı düğümü otomatik olarak doldurulacaktır.

If queue managers become members of clusters while the IBM MQ Explorer is running, then you must maintain the IBM MQ Explorer with up-to-date administration data about clusters so that it can communicate effectively with them and display correct cluster information when requested. Bunu yapmak için IBM MQ Explorer ' in aşağıdaki bilgilere gerek vardır:

- Havuz kuyruğu yöneticisinin adı
- Uzak bir kuyruk yöneticisince, havuz kuyruğu yöneticisinin bağlantı adı

Bu bilgiler ile IBM MQ Explorer şunları yapabilir:

- Kümedeki kuyruk yöneticilerinin listesini almak için havuz kuyruk yöneticisini kullanın.
- Kümenin üyesi olan ve desteklenen platformlarda ve komut düzeylerinde kuyruk yöneticilerini yönetin.

Aşağıdaki durumlarda yönetim olanaklı değildir:

- Seçilen havuz kullanılamaz duruma geliyor. IBM MQ Explorer , otomatik olarak alternatif bir havuza geçmiyor.
- Seçilen havuz TCP/IP üzerinden iletişim kurulamaz.

- Seçilen havuz, IBM MQ Explorer tarafından desteklenmeyen bir platform ve komut düzeyinde çalışan bir kuyruk yöneticisinde çalışıyor.

Denetlenebilen küme üyeleri yerel olabilir ya da TCP/IP kullanılarak iletişim kurulabilirlerse uzaktan kumanda edilebilir. IBM MQ Explorer , bir istemci bağlantısı kullanmadan, doğrudan bir kümenin üyesi olan yerel kuyruk yöneticilerine bağlanır.

IBM MQ Explorer için veri dönüştürme

IBM MQ Explorer , CCSID 1208 (UTF-8) içinde çalışır. Bu, IBM MQ Explorer ' un verileri uzak kuyruk yöneticilerinden doğru olarak görüntülemesini sağlar. Bir kuyruk yöneticisine doğrudan bağlanıp ya da bir ara kuyruk yöneticisi kullanarak IBM MQ Explorer , tüm gelen iletilerin CCSID 1208 (UTF-8) değerine dönüştürülmesini gerektirir.

IBM MQ Explorer ile kuyruk yöneticisi arasında, IBM MQ Explorer ' in tanımadığı bir CCSID ile bağlantı kurmaya çalışırsanız bir hata iletisi yayınlanır.

Desteklenen dönüştürmeler, [Kod sayfası dönüştürme](#)' de açıklanır.

Windows IBM MQ Taskbar uygulamasını kullanma (yalnızca Windows)

IBM MQ Görev çubuğu uygulaması, sunucudaki Windows sistem tepsisinde bir simge görüntüler. Bu simge, IBM MQ ' un geçerli durumunu ve bazı basit işlemleri gerçekleştirebileceğiniz bir menüyü sağlar.

Windows' ta, IBM MQ simgesi sunucudaki sistem tepsisinde bulunur ve renk kodlu durum simgesiyle üzerine yazılır; bu simge aşağıdaki anlamlardan birine sahip olabilir:

Yeşil

Doğru biçimde çalışıyor; şu anda uyarı yok

Mavi

Belirsiz; IBM MQ başlatıyor ya da sona erdiriliyor

Sarı

Uyarı; bir ya da daha fazla hizmet başarısız oldu ya da zaten başarısız oldu

Menüyü görüntülemek için IBM MQ simgesini sağ tıklatın. Menüden aşağıdaki işlemleri gerçekleştirebilirsiniz:

- IBM MQ Alert Monitor olanağını açmak için **Aç** düğmesini tıklatın.
- IBM MQ Taskbar uygulamasından çıkmak için **Exit** (Çık) düğmesini tıklatın.
- IBM MQ Explorer' ı başlatmak için **IBM MQ Explorer** ögesini tıklatın.
- IBM MQ' u durdurmak için **IBM MQ' u Durdur** seçeneğini tıklatın.
- IBM MQ Alert Monitor ile ilgili bilgileri görüntülemek için **Hakkında IBM MQ** seçeneğini tıklatın.

Windows IBM MQ uyarı izleyicisi uygulaması (yalnızca Windows)

The IBM MQ alert monitor is an error detection tool that identifies and records problems with IBM MQ on a local machine.

Uyarı izleme programı, bir IBM MQ sunucusunun yerel kuruluşunun geçerli durumuyla ilgili bilgileri görüntüler. It also monitors the Windows Advanced Configuration and Power Interface (ACPI) and ensures the ACPI settings are enforced.

IBM MQ uyarı izleyicisinden şunları yapabilirsiniz:

- Doğrudan IBM MQ Explorer ' a erişin
- Tüm bekleyen uyarılarla ilgili bilgileri görüntüle
- Yerel makineden IBM MQ hizmetini sona erdirin

- Ağ üzerinden uyarı iletilerini yapılandırılabilir bir kullanıcı hesabına ya da bir Windows iş istasyonuna ya da sunucuya yönlendirilir.

Yerel IBM MQ nesnelere çalışma

Yerel IBM MQ nesnelere, Message Queue Interface (MQI) olanağını kullanan uygulama programlarını desteklemek için denetleyebilirsiniz.

Bu görev hakkında

Bu bağlamda yerel yönetim, IBM MQ nesnelere oluşturma, görüntüleme, değiştirme, kopyalama ve silme anlamına gelir.

Bu bölümde açıklanan yaklaşımlara ek olarak, yerel IBM MQ nesnelere yönetmek için IBM MQ Explorer ' i kullanabilirsiniz. Daha fazla bilgi için bkz [“IBM MQ Explorerkomutunu kullanarak yönetim” sayfa 110.](#)

Yordam

- Yerel IBM MQ nesnelere yönetmenize yardımcı olması için aşağıdaki konularda bilgileri kullanın.
 - [MQI kullanan uygulama programları](#)
 - [“MQSC komutlarını kullanan yönetim” sayfa 11](#)
 - [“Kuyruk yöneticisi özniteliklerinin görüntülenmesi ve değiştirilmesi” sayfa 126](#)
 - [“Yerel kuyruklarla çalışma” sayfa 129](#)
 - [“Diğer ad kuyruklarıyla çalışılması” sayfa 141](#)
 - [“Model kuyruklarıyla çalışma” sayfa 142](#)
 - [“Hizmetlerle çalışma” sayfa 168](#)
 - [“Tetikleme için nesnelere yönetme” sayfa 175](#)

Kuyruk yöneticileriyle çalışılması

Bir kuyruk yöneticisini başlatmak ve durdurmak için denetim komutlarını kullanabilirsiniz. Kuyruk yöneticisi özniteliklerini görüntülemek ya da değiştirmek için MQSC komutlarını kullanabilirsiniz.

İlgili görevler

[Çoklu Platformlar üzerinde Kuyruk Yöneticileri Yaratılması](#)

Multi

Kuyruk Yöneticisinin Başlatılması

Bir kuyruk yöneticisi yarattığınızda, komutları ya da MQI çağrılarını etkinleştirmek için bir kuyruk yöneticisi başlatmalısınız.

Bu görev hakkında

You can start a queue manager by using the **strmqm** command. **strmqm** komutunun ve aksamalarının açıklaması için bkz. [strmqm](#).

Linux

Windows

Alternatively, on Windows and Linux (x86 and x86-64 platforms) systems, you can start a queue manager by using the IBM MQ Explorer.

Windows

On Windows you can start a queue manager automatically when the system starts using the IBM MQ Explorer. Daha fazla bilgi için bkz [“IBM MQ Explorerkomutunu kullanarak yönetim” sayfa 110.](#)

Yordam

- Bir kuyruk yöneticisini **strmqm** komutunu kullanarak başlatmak için, komutu girerek başlatmak istediğiniz kuyruk yöneticisinin adını girin.

Örneğin, QMBadlı bir kuyruk yöneticisini başlatmak için aşağıdaki komutu girin:

```
strmqm QMB
```

Not: Çalışmakta olduğunuz kuyruk yöneticisiyle ilişkilendirilmiş kurulumdan **strmqm** komutunu kullanmanız gerekir. You can find out which installation a queue manager is associated with using the `dspmq -o` installation command.

`strmqm` komutu, kuyruk yöneticisi çalışmaya başlayınca ve bağlantı isteklerini kabul etmeye hazır oluncaya kadar denetimi döndürmez.

•

To start a queue manager by using the IBM MQ Explorer, complete the following steps:

- IBM MQ Explorer'ni açın.
- Navigator görünümünde kuyruk yöneticisini seçin.
- Başlat'**ı tıkklatın.

Sonuçlar

Kuyruk yöneticisi başlatılır.

Kuyruk yöneticisi başlatma işlemi birkaç saniyeden daha uzun sürerse IBM MQ , başlatma ilerleyişini ayrıntılı olarak içeren bilgi iletileri iletir.

Kuyruk Yöneticisinin Durdurulması

Bir kuyruk yöneticisini durdurmak için **endmqm** komutunu kullanabilirsiniz. Bu komut, bir kuyruk yöneticisini durdurmak için dört yol sağlar: denetimli ya da susturulmuş, sona erdirme, anında kapatma, önleyici bir kapatma ve bekleme kapatma. Diğer bir seçenek olarak, Windows ve Linuxüzerinde, IBM MQ Explorerkomutunu kullanarak bir kuyruk yöneticisini durdurabilirsiniz.

Bu görev hakkında

Tek bir yönetim ortamı kuyruk yöneticisini **endmqm** komutuyla durdurmanın dört yolu vardır:

Denetimli (susturulmuş) sona erdirme

Varsayılan olarak, **endmqm** komutu, belirtilen kuyruk yöneticisinin susturulmuş bir şekilde sona erdirilmesini sağlar. Tüm bağlı uygulamaların bağlantısı kesilinceye kadar, susturulmuş bir sona erdirme bekler, bu nedenle tamamlanması biraz zaman alabilir.

Anında kapatma

Hemen sona erdirme işlemi için, yürürlükteki MQI çağrılarına izin verilir, ancak yeni çağrılar başarısız olur. Bu sona erdirme tipi, uygulamaların kuyruk yöneticisinden bağlantısını kesmesini beklemez.


Önleyici kapatma

Kuyruk yöneticisi hemen durdurulur. Bu kapatma tipini yalnızca olağan koşullarda (örneğin, bir kuyruk yöneticisi olağan bir **endmqm** komutunun sonucu olarak durdurulmadığında) kullanın.

Bekleme kapatıldı

Bu sona erdirme tipi, denetim yalnızca kuyruk yöneticisi durdurulduktan sonra size döndürülmesi dışında, denetimli sona erdirme işleminden eşdeğerdir.

endmqm komutu, çok eşgörünümlü bir kuyruk yöneticisinin tek bir yönetim ortamı kuyruk yöneticisini durdurduğu şekilde tüm yönetim ortamlarını durdurur. You can issue the **endmqm** on either the active instance, or one of the standby instances of a multi-instance queue manager. Ancak, kuyruk yöneticisini sona erdirmek için etkin yönetim ortamında **endmqm** komutunu yayınlamanız gerekir.


 IBM MQ 9.1.4' tan, kuyruk yöneticisini belirttiğiniz saniye sayısı kadar bir hedef zaman içinde sona erdirecek seçeneğiniz vardır, daha ayrıntılı bilgi için bkz. "[Hedef zaman içinde kuyruk yöneticisini sonlandırma](#)" sayfa 122 .

endmqm komutunun ve aksamlarının ayrıntılı bir açıklaması için bkz. [endmqm](#).

İpucu: Kuyruk yöneticisini sona erdirmeye ilgili sorunlar genellikle uygulamalar tarafından neden olur. Örneğin, uygulamalar:

- MQI dönüş kodlarını düzgün bir şekilde işaretlemeyin
- Susturma bildirimini istenmemesi
- Kuyruk yöneticisinden bağlantıyı kesmeden sona erdirin (bir MQDISC çağrısı yayınlayarak)

Kuyruk yöneticisini durdurmaya çalıştığınızda bir sorun oluşursa, Ctrl-C komutunu kullanarak **endmqm** komutundan ayrılabilirsiniz. Daha sonra başka bir **endmqm** komutu yayınlayabilirsiniz, ancak bu kez gereksinim duyduğunuz sona erdirmeye tipini belirten bir parametre ile birlikte bu kez daha fazla bilgi edinebilirsiniz.

 As an alternative to using the **endmqm** command, on Windows and Linux, you can stop a queue manager by using the IBM MQ Explorer to carry out either a controlled or an immediate shutdown.

Yordam

- To stop the queue manager by using the **endmqm** command, enter the command followed by the appropriate parameter, if required, and the name of the queue manager that you want to stop.

Not: Çalışmakta olduğunuz kuyruk yöneticisiyle ilişkilendirilmiş kurulumdan **endmqm** komutunu kullanmanız gerekir. Bir kuyruk yöneticisinin hangi kuruluşla ilişkilendirildiği hakkında bilgi almak için, **dspmqs** komutunu kullanın:

```
dspmqs -o installation
```

- Denetimli (susturulmuş) sona erdirmeye işlemi gerçekleştirmek için, **endmqm** komutunu aşağıdaki örnekte gösterildiği gibi girin; bu komut, QMBadlı bir kuyruk yöneticisini durdurur:

```
endmqm QMB
```

Alternatively, entering the **endmqm** command with the **-c** parameter, as shown in the following example, is equivalent to an **endmqm QMB** command.

```
endmqm -c QMB
```

Her iki durumda da, denetim hemen size gönderilir ve kuyruk yöneticisi durduğunda size bildirim gönderilmez. Komutun tüm uygulamalar duruncaya kadar beklemesini istiyorsanız ve denetimi size döndürmeden önce kuyruk yöneticisi sona erdiyse, aşağıdaki örnekte gösterildiği gibi yerine **-w** parametresini kullanın.

```
endmqm -w QMB
```

- To carry out an immediate shutdown, enter the **endmqm** command with the **-i** parameter as shown in the following example:

```
endmqm -i QMB
```

- To carry out a preemptive shutdown, enter the **endmqm** command with the **-p** parameter as shown in the following example:

```
endmqm -p QMB
```



Uyarı: Önleyici bir kapatma işlemi, bağlı uygulamalar için önceden kestirilemeyecek sonuçlar doğurabilir. Olağan bir **endmqm** komutu kullanılarak kuyruk yöneticisini durdurmak

için diğer tüm girişimler başarısız olursa bu seçeneği kullanmayın. **ALW** Önçekimsel kapatma işe yaramazsa, "[Kuyruk yöneticisini el ile durdurma](#)" sayfa 123 ' i deneyin.

- [Automatic client reconnection](#) isteğini istemek için, `-r` değiştirgesiyle **endmqm** komutunu girin. Bu değiştirge, istemcilerin kuyruk yöneticisi grubundaki diğer kuyruk yöneticilerine bağlanabilirliklerini yeniden kurmanın etkisine sahiptir.

Not: Bir kuyruk yöneticisinin varsayılan **endmqm** komutu kullanılarak sona erdirilmesiyle, otomatik istemci yeniden bağlantısı tetiklenmez.

- Etkin yönetim ortamını sona erdirdikten sonra, çok eşgörunümlü bir kuyruk yöneticisinin yedek yönetim ortamına aktarmak için, çok eşgörunümlü kuyruk yöneticisinin etkin somut örneğinde `-s` değiştirgesini içeren **endmqm** komutunu girin.
- Çok eşgörunümlü bir kuyruk yöneticisinin yedek yönetim ortamını sona erdirmek ve çalışmakta olan etkin yönetim ortamını bırakmak için, çok eşgörunümlü kuyruk yöneticisinin yedek yönetim ortamında `-x` değiştirgesini kullanarak **endmqm** komutunu girin.



On Windows and Linux, to stop the queue manager by using IBM MQ Explorer, complete the following steps:

- a) IBM MQ Explorer'ni açın.
- b) Select the queue manager from the Navigator View.
- c) **Durdurdüğmesini** tıklatın.
End Queue Manager (Kuyruk Yöneticisi Sonu) panosu görüntülenir.
- d) **Denetimliya** da **Hemenseçeneğini** belirleyin.
- e) **Tamam'**ı tıklatın.
Kuyruk yöneticisi durur.

İlgili görevler

[AIX üzerinde çok eşgörunümlü kuyruk yöneticilerine bakım düzeyi güncellemeleri uyguluyor](#)

[Linux üzerinde çok eşgörunümlü kuyruk yöneticilerine bakım düzeyi güncellemeleri uyguluyor](#)

[Windows üzerinde çok eşgörunümlü kuyruk yöneticilerine bakım düzeyi güncellemeleri uyguluyor](#)

Hedef zaman içinde kuyruk yöneticisini sonlandırma

Kuyruk yöneticisini, önemli kuyruk yöneticisi bakım görevlerini kesintiye uğratmadan ya da kesintiye uğratmadan, belirttiğiniz sayıda saniye için bir hedef süresiyle sona erdirebilirsiniz.

endmqm komutunu kullanırken bir hedef zamanı belirtmenin iki yolu vardır. **-t** seçeneği, kuyruk yöneticisinin sona erdirilmesinin aşamasını uzun bir süre uzatabilecek temel kuyruk yöneticisi bakım görevlerinin tamamlanmasını sağlar. **-tp** seçeneği, belirtilen hedef süreye uymak için gerekiyorsa, önemli kuyruk yöneticisi bakım görevlerini kesintiye uğratır.

When you specify a target time, the shutdown type of **-w**, **-i**, or **-p** indicates the starting shutdown type.

Not: An immediate shutdown is still orderly, differing from a controlled shutdown primarily in the way that any running applications are quiesced. **immediate** sona erdirme işlemi, kuyruk sıkıştırma ve zaman kaybına neden olabilecek NPMCLASS (yüksek) iletileri kalıcı olarak bakım işlemlerini gerektirir; ancak, bir süre sınırlı olarak kapatıldığında, bu ek işlemler hedef süreye müdahale ederken bu ek işlemler sona erdirilecektir.

Kuyruk yöneticisi, hedef saati karşılamak için, sona erdirme tipini gerektiği şekilde yönlendiriyor. Örneğin:

- **-w** ' ta başlayan 10 saniyelik bir **-t** hedefi, yedi saniye susturulmuş olabilir, kuyruk yöneticisinin hemen sona erdirilmesi de dahil olmak üzere, iki saniye sonra, ev bakımı da dahil olmak üzere, daha fazla ev bakımı yapılmaksızın hemen sona erdirilebilir:

```
endmqm -w -t 10 queue_manager
```

- 10 saniyelik bir **-tp** hedefi 7 saniye susturulabilir, kuyruk yöneticisinin hemen sona erdirilmesi, housekeeping de içinde olmak üzere, ikinci bir anında kapatma işlemi de dahil olmak üzere, iki saniye sonra IBM MQ işlemlerini sona erdirmek üzere sona erdirilebilir.

```
endmqm -c -tp 10 queue_manager
```

- A two second **-tp** target at **-i** might be one second immediate shutdown of the queue manager, including housekeeping, one second immediate shutdown without further housekeeping, then start ending IBM MQ processes:

```
endmqm -i -tp 2 queue_manager
```

- A one second target at **-w** could be 0.1 seconds at wait, for example, just long enough to send out IBM MQ return codes to connected applications, 0.9 seconds immediate shutdown of the queue manager, including housekeeping, then immediate shutdown without further housekeeping; then start ending IBM MQ processes.

İlgili başvurular

[endmqm \(uç kuyruk yöneticisi\)](#)

ALW Kuyruk yöneticisini el ile durdurma

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 120 komutuyla açıklandığı gibi **endmqm** komutunu kullanmaktadır. Standart bir şekilde kuyruk yöneticisini durduramazsanız, kuyruk yöneticisini el ile durdurmayı deneyebilirsiniz. Bunu yapmak için kullandığınız platforma bağlı olarak bu yol size bağlıdır.

Yordam

- **Windows**
Windows' ta kuyruk yöneticisini durdurmak için bkz. “Windows üzerinde kuyruk yöneticisini el ile durdurma” sayfa 123.
- **Linux** **AIX**
To stop a queue manager on AIX or Linux, see “AIX and Linux üzerinde kuyruk yöneticisini el ile durdurma” sayfa 124.

İlgili görevler

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

İlgili başvurular

[sonmqm](#)

Windows üzerinde kuyruk yöneticisini el ile durdurma

endmqm komutunu kullanarak Windows üzerinde bir kuyruk yöneticisini durduramazsanız, çalışmakta olan ve IBM MQ hizmetini durduran işlemleri sonlandırarak kuyruk yöneticisini el ile durdurmayı deneyebilirsiniz.

Bu görev hakkında

İpucu: Windows Task Manager ve **tasklist** komutu, görevlerle ilgili sınırlı bilgi verir. For more information to help to determine which processes relate to a particular queue manager, consider using a tool such as *Süreç Gezgini* (procexp.exe), which is available for download from the Microsoft website at <http://www.microsoft.com>.

Windows' ta kuyruk yöneticisini durdurmak için aşağıdaki adımları tamamlayın.

Yordam

1. List the names (IDs) of the processes that are running, by using the Windows Task Manager.
2. End the processes by using Windows Task Manager, or the **taskkill** command, in the following order (if they are running):

Çizelge 7. Çalıştırılıyorsa, durdurulacak Windows işlemleri	
İşlem adı	Tanım
AMQZMUC0	Kritik süreç yöneticisi
AMQZXMA0	Yürütme denetleyicisi
AMQZFUMA	OAM işlemi
AMQZLAA0	LQM araçları
AMQZLSA0	LQM araçları
AMQZMUFO	Yardımcı Program Yöneticisi
AMQZMGR0	Süreç denetleyicisi
AMQZMUR0	Yeniden başlatılabilir süreç yöneticisi
AMQFQPUB	Abone olma işlemi yayınla
AMQFCXBA	Aracı işçi süreci
AMQRPMPPA	Havuz oluşturma işlemi
AMQCRSTA	İş parçacıklı yanıtlayıcı olmayan iş süreci
AMQCRS6B	LU62 alıcı kanalı ve istemci bağlantısı
AMQRRMFA	Havuz işlemi (kümeler için)
AMQPCSEA	Komut sunucusu
RUNMQTRM	Bir sunucu için tetikleme izleyicinin başlatılması
RUNMQDLQ	Ölü-mektup kuyruğu işleyicini çağır
RUNMQCHI	Kanal başlatıcı işlemi
RUNMQLSR	Kanal dinleyicisi işlemi
AMQXSSVN	Paylaşılan bellek sunucuları

3. Windows Control Panel (Denetim Masası) içindeki **Administration tools > Services** (Yönetim Araçları-Hizmetler) seçeneklerini kullanarak IBM MQ hizmetini durdurun.
4. Tüm yöntemleri denediyseniz ve kuyruk yöneticisi durdurulmamışsa, sisteminizi yeniden başlatın.

Linux → AIX **AIX and Linux üzerinde kuyruk yöneticisini el ile durdurma**

endmqm komutunu kullanarak AIX ya da Linux üzerinde kuyruk yöneticisini durduramazsanız, çalışmakta olan ve IBM MQ hizmetini durduran işlemleri sonlandırarak kuyruk yöneticisini el ile durdurmayı deneyebilirsiniz.

Bu görev hakkında

AIX ya da Linux üzerinde kuyruk yöneticisini durdurmak için aşağıdaki adımları tamamlayın.

Kuyruk yöneticisini el ile durdurursanız, FFST alınabilir ve FDC dosyaları /var/mqm/errorsiçine yerleştirilebilir. Bu, kuyruk yöneticisinde bir hata olarak kabul edilmemelidir.

Kuyruk yöneticisi, el ile durdurmanın bu yöntemini kullanmanıza rağmen, olağan durumda yeniden başlatılacaktır.

Yordam

1. Find the process IDs of the queue manager programs that are still running by using the **ps** command. Örneğin, kuyruk yöneticisi QMNAMEolarak adlandırıldıysa, aşağıdaki komutu kullanın:

```
ps -ef | grep QMNAME
```

2. End any queue manager processes that are still running by using the **kill** command, specifying the PIDs discovered by using the **ps** command.

Bir süreci sona erdirmek için **kill -KILL <pid>** ya da eşdeğer **kill -9 <pid>** komutunu kullanın.

Öldürmek istediğiniz PID ' ler arasında çalışmak zorundasınız, birer birer, her seferinde bu komutu veriniz.

Önemli: 9 (SIGKILL) dışında bir sinyal kullanıyorsanız, işlem büyük olasılıkla durdurulmaz ve beklenmedik sonuçlar elde eder.

İşlemleri aşağıdaki sırada sona erdirin:

<i>Çizelge 8. Çalıştırılıyorsa, durdurulacakAIX and Linux işlemleri</i>	
İşlem adı	Tanım
amqzmuc0	Kritik süreç yöneticisi
amqzma0	Yürütme denetleyicisi
amqzfuma	OAM işlemi
amqzlaa0	LQM araçları
amqzlsa0	LQM araçları
amqzmuf0	Yardımcı Program Yöneticisi
amqzmur0	Yeniden başlatılabilir süreç yöneticisi
amqzmgr0	İşlem denetleyicisi
amqfqpub	Abone olma işlemi yayınla
amqfcxba	Aracı işçi süreci
amqrmppa	Havuz oluşturma işlemi
amqcrsta	İş parçacıklı yanıtlayıcı olmayan iş süreci
amqcrs6b	LU62 alıcı kanalı ve istemci bağlantısı
amqrrmfa	Havuz işlemi (kümeler için)
amqpcdeniz	Komut sunucusu
runmqtrm	Bir sunucu için tetikleme izleyicinin başlatılması
Runmqdlq	Ölü-mektup kuyruğu işleyicini çağır
runmqchi	Kanal başlatıcı işlemi
Runmqlsr	Kanal dinleyicisi işlemi

Not: Durdurmak için başarısız olan işlemleri sona erdirmek için **kill -9** komutunu kullanabilirsiniz.

Kuyruk yöneticisinin yeniden başlatılması

You can use the **strmqm** command to restart a queue manager, or on Windows and Linux x86-64 systems, you can restart a queue manager from IBM MQ Explorer.

Bu görev hakkında

strmqm komutunu kullanarak bir kuyruk yöneticisini yeniden başlatabilirsiniz. **strmqm** komutunun ve aksamalarının açıklaması için bkz. [strmqm](#).

On Windows and Linux x86-64 systems, you can restart a queue manager by using the IBM MQ Explorer in the same way as for starting a queue manager.

Yordam

- Bir kuyruk yöneticisini **strmqm** komutunu kullanarak yeniden başlatmak için, komutu girin ve ardından yeniden başlatmak istediğiniz kuyruk yöneticisinin adını girin.

Örneğin, `strmqm saturn.queue.manager` adlı bir kuyruk yöneticisini başlatmak için aşağıdaki komutu girin:

```
strmqm saturn.queue.manager
```

- To start a queue manager by using the IBM MQ Explorer, complete the following steps:

- a) IBM MQ Explorer'ni açın.
- b) Navigator görünümünde kuyruk yöneticisini seçin.
- c) **Başlat'**ı tıklatın.

Sonuçlar

Kuyruk yöneticisi yeniden başlatılır.

Kuyruk yöneticisini yeniden başlatma işlemi birkaç saniyeden daha uzun sürerse, IBM MQ sorun bilgileri, başlatma ilerleyişini sürekli olarak ayrıntılandırır.

Kuyruk yöneticisi özniteliklerinin görüntülenmesi ve değiştirilmesi

Kuyruk yöneticisi özniteliklerini görüntülemek ya da değiştirmek için MQSC komutlarını kullanabilirsiniz.

Bu görev hakkında

Bir kuyruk yöneticisine ilişkin kuyruk yöneticisi değiştirgelerini görüntülemek için **DISPLAY QMGR** komutunu ve yerel bir kuyruk yöneticisine ilişkin kuyruk yöneticisi değiştirgelerini değiştirmek için **ALTER QMGR** komutunu kullanın.

Yordam

- **runmqsc** komutunda belirtilen kuyruk yöneticisinin özniteliklerini görüntülemek için, **DISPLAY QMGR** MQSC komutunu kullanın:

```
DISPLAY QMGR
```

Aşağıdaki örnekte, bu komuttan tipik çıkış gösterilmektedir:

```
DISPLAY QMGR
1 : DISPLAY QMGR
AMQ8408: Display Queue Manager details.
```

QMNAME(QM1)	ACCTCONO(DISABLED)
ACCTINT(1800)	ACCTMQI(OFF)
ACCTQ(OFF)	ACTIVREC(MSG)
ACTVCONO(DISABLED)	ACTVTRC(OFF)
ALTDATE(2012-05-27)	ALTTIME(16.14.01)
AUTHOREV(DISABLED)	CCSID(850)
CHAD(DISABLED)	CHADEV(DISABLED)
CHDEXIT()	CHLEV(DISABLED)
CLWLDATA()	CLWLXIT()
CLWLLEN(100)	CLWLMRUC(999999999)
CLWLUSEQ(LOCAL)	CMDEV(DISABLED)
CMDLEVEL(800)	COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE)
CONFIGEV(DISABLED)	CRDATE(2011-05-27)
CRTIME(16.14.01)	DEADQ()
DEFXMITQ()	DESCR()
DISTL(YES)	INHIBTEV(DISABLED)
IPADDRV(IPV4)	LOCALEV(DISABLED)
LOGGEREV(DISABLED)	MARKINT(5000)
MAXHANDS(256)	MAXMSGL(4194304)
MAXPROPL(NOLIMIT)	MAXPRTY(9)
MAXUMSGS(10000)	MONACLS(QMGR)
MONCHL(OFF)	MONQ(OFF)
PARENT()	PERFMVEV(DISABLED)
PLATFORM(WINDOWSNT)	PSRTYCNT(5)
PSNPMMSG(DISCARD)	PSNPREV(NORMAL)
PSSYNCP(IFPER)	QMID(QM1_2011-05-27_16.14.01)
PSMODE(ENABLED)	REMOOTEV(DISABLED)
REPOS()	REPOSNL()
ROUTEREC(MSG)	SCHINIT(QMGR)
SCMDSERV(QMGR)	SSLCRLNL()
SSLCRYP()	SSLEV(DISABLED)
SSLFIPS(NO)	SSLKEYR(C:\Program Files\IBM\WebSphere
MQ\Data\mqgrs\QM1\ssl\key)	
SSLRKEYC(0)	STATACLS(QMGR)
STATCHL(OFF)	STATINT(1800)
STATMQI(OFF)	STATQ(OFF)
STRSTPEV(ENABLED)	SYNCP
TREELIFE(1800)	TRIGINT(999999999)

Not: SYNCP, salt okunur bir kuyruk yöneticisi öznitelidir.

ALL parametresi, **DISPLAY QMGR** komutundaki varsayılan parametredir. Kuyruk yöneticisi özniteliklerinin tümünü görüntüler. Çıkış, özellikle, size varsayılan kuyruk yöneticisi adını, ölü harf kuyruğu adını ve komut kuyruğu adını gösterir.

Bu kuyrukların şu komutu girerek varolduğunu doğrulayabilirsiniz:

```
DISPLAY QUEUE (SYSTEM.*)
```

Bu, kök SYSTEM.* ile eşleşen kuyrukların 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, kullanılmayan harf kuyruğunu değiştirir ve olayların engellenmesini sağlar.

ALTER QMGR komutunda belirtilmeyen parametreler, bu parametrelerin değişmeden bırakılmasına ilişkin var olan değerlerde sonuçlanıyor.

İlgili görevler

[Çoklu Platformlar üzerinde Kuyruk Yöneticileri Yaratılması](#)

İlgili başvurular

[Kuyruk yöneticisine ilişkin öznitelikler](#)

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

[QMGR GÖRÜNTÜLE](#)

Kuyruk Yöneticisinin Silinmesi

You can delete a queue manager using the **dltmqm** command. Diğer bir seçenek olarak, Windows ve Linux sistemlerinde, kuyruk yöneticisini silmek için IBM MQ Explorer ' i kullanabilirsiniz.

Başlamadan önce



Uyarı:

- Kuyruk yöneticisi silindiğinde, kuyruk yöneticisiyle ilişkili tüm kaynakları da sildiğiniz için, kuyruk yöneticisi tüm kuyruklar ve bunların iletileri ve tüm nesne tanımlamaları da içinde olmak üzere, köklü bir adımdır. **dltmqm** komutunu kullanırsanız, fikrinizi değiştirmenize olanak tanıyan bir bilgi istemi görüntülenmez; Enter tuşuna bastığınızda tüm ilişkili kaynaklar kaybedilir.
- **Windows** Windows üzerinde, bir kuyruk yöneticisinin silinmesi, kuyruk yöneticisini otomatik başlatma listesinden (“Kuyruk Yöneticisinin Başlatılması” sayfa 119 ' da açıklanmıştır) da kaldırır. Komut tamamlandığında, bir IBM MQ queue manager ending iletisi görüntülenir; kuyruk yöneticisinin silindiği söylenmez.
- Bir küme kuyruk yöneticisinin silinmesi, küme kuyruk yöneticisinin kümeden kaldırılmasına neden olmaz. Daha fazla bilgi için [dltmqm](#) içindeki kullanım notlarına bakın.

Bu görev hakkında

You can delete a queue manager by using the **dltmqm** command. **dltmqm** komutunun ve aksamalarının açıklaması için bkz. [dltmqm](#). Yalnızca güvenilir denetimcilerin bu komutu kullanma yetkisine sahip olduğundan emin olun. (Güvenliğe ilişkin bilgi için bkz. [AIX, Linux, and Windows üzerinde güvenliğin ayarlanması](#).)

Alternatively, on Windows and Linux (x86 and x86-64 platforms) systems, you can delete a queue manager by using the IBM MQ Explorer.

Yordam

- Bir kuyruk yöneticisini **dltmqm** komutunu kullanarak silmek için aşağıdaki adımları tamamlayın:
 - a) Kuyruk yöneticisini durdurun.
 - b) Şu komutu verin:

```
dltmqm QMB
```

Not: Çalışmakta olduğunuz kuyruk yöneticisiyle ilişkilendirilmiş kurulumdan **dltmqm** komutunu kullanmanız gerekir. You can find out which installation a queue manager is associated with using the `dspmqr -o installation` command.

- IBM MQ Explorer komutunu kullanarak bir kuyruk yöneticisini silmek için aşağıdaki adımları tamamlayın:

- a) IBM MQ Explorer' nı açın.
- b) Navigator görünümünde kuyruk yöneticisini seçin.
- c) Kuyruk yöneticisi durdurulmamışsa, durdurun.
Kuyruk yöneticisini durdurmak için, yöneticiyi farenin sağ düğmesiyle tıklatın ve **Durdur** ' u tıklatın.
- d) Kuyruk yöneticisini silin.
Kuyruk yöneticisini silmek için, yöneticiyi farenin sağ düğmesiyle tıklatın ve **Sil** ögesini seçin.

Sonuçlar

Kuyruk yöneticisi silindi.

MQI kanalları durduruluyor

Bir sunucu bağlantısı kanalına karşı STOP CHANNEL komutunu verdiğinizde, istemci bağlantı kanalını durdurmak için hangi yöntemin kullanılacağını seçebilirsiniz. Bu, bir MQGET bekleme çağrısını yayınlayan bir istemci kanalının denetlenebilir ve kanalın nasıl durdurulacağı ve ne zaman durdurulacağı konusunda karar verebilirsiniz.

STOP CHANNEL komutu üç kip ile yayınlanabilir; bu, kanalın nasıl durdurulacağı anlamına gelir:

Susturma

Yürürlükteki iletiler işlendikten sonra kanalı durdurur.

Sohbet paylaşımı etkinleştirilirse, IBM MQ MQI client , durdurma isteğinin zamanında farkında olur; bu süre, ağır hızına bağlıdır. İstemci uygulaması, IBM MQ' e daha sonra yapılan bir çağrı yayınının verilmesi sonucunda durdurma isteğinin farkında olur.

Kuvvet

Kanalı hemen durdurur.

Sonlandır

Kanalı hemen durdurur. Kanal bir işlem olarak çalışıyorsa, kanal işlemini sonlandırabilir ya da kanal bir iş parçacığı, iş parçacığı olarak çalışıyorsa, bu işlem, kanal işlemini sonlandırabilir.

Bu çok aşamalı bir süreç. Kip sonlandırma kullanılırsa, sunucu bağlantı kanalını durdurma girişiminde bulunulursa, önce kip susturma, sonra kip zorlamalı olarak ve kip sonlandırma işlemi ile gerekiyorsa. Müşteri, sonlandırmanın farklı aşamaları sırasında farklı dönüş kodları alabilir. Süreç ya da iş parçacığı sona erdirilirse, istemci bir iletişim hatası alır.

Uygulamaya döndürülen dönüş kodları, yayınlanan MQI çağrısına göre değişiklik gösterir ve STOP CHANNEL komutu yayınlandı. İstemci bir MQRC_CONNECTION_QUIESCING ya da MQRC_CONNECTION_BROKEN dönüş kodu alır. Bir istemci MQRC_CONNECTION_QUIESCING işlemini saptarsa, yürürlükteki hareketi tamamlamayı ve sona erdirmeyi dener. Bu, MQRC_CONNECTION_BROKEN ile mümkün değildir. İstemci işlemi tamamlamazsa ve yeterince hızlı sonlandırmazsa, birkaç saniye sonra CONNECTION_BROKEN değeri alır. MODE (FORCE) ya da MODE (TERMINATE) içeren STOP CHANNEL komutu, MOLE (QUIESCE) değerinden daha büyük bir CONNECTION_BROKEN ile 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](#) başlıklı konuya bakın.

İlgili başvurular

[Kuyruklara ilişkin adlandırma kısıtları](#)

[Diğer nesnelere ilişkin adlandırma kısıtlamaları](#)

DEFINE QLOCAL ile yerel kuyruk tanımlama

Bir uygulama için, yerel kuyruk yöneticisi, uygulamanın bağlı olduğu kuyruk yöneticidir. Yerel kuyruk yöneticisi tarafından yönetilen kuyruklara, o kuyruk yöneticisinin yerel olduğu söyleniyor.

Bu görev hakkında

Yerel kuyruk yaratmak için **DEFINE QLOCAL** MQSC komutunu kullanın. Varsayılan yerel kuyruk tanımlamasında tanımlı olan varsayılan değeri kullanabilir ya da kuyruk özelliklerini varsayılan yerel kuyruklardaki gibi değiştirebilirsiniz.

Not: Varsayılan yerel kuyruk SYSTEM.DEFAULT.LOCAL.QUEUE ve sistem kuruluşunda yaratılır.

Yordam

- Yerel bir kuyruk yaratmak için, aşağıdaki örnekte gösterildiği gibi **DEFINE QLOCAL** komutunu girin. Bu örnekte, **DEFINE QLOCAL** komutu ORANGE.LOCAL.QUEUE :
 - Bir öncelik sırası temelinde işler, koyar, koyar ve işler için etkinleştirilir.
 - Bu bir *normal* kuyruğdur; bir başlatma kuyruğu ya da iletim kuyruğu değildir ve tetikleyici iletileri oluşturmaz.
 - Kuyruk derinliği üst sınırı 5000 iletilerdir; ileti uzunluğu üst sınırı 4194304 byte 'tır.

```
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. Açıklamadaki değer dışında, örnekte gösterilen tüm öznitelik değerleri varsayılan değerlerdir. Bu örnekler, şekil amacıyla eklenmiştir. Varsayılan değerlerin istediğiniz ya da değiştirilmediğinden eminseniz, bunları atlayabilirsiniz. Ayrıca bkz. [“Varsayılan nesne özniteliklerinin DISPLAY QUEUE ile görüntülenmesi” sayfa 130.](#)
2. **USAGE (NORMAL)** , bu kuyruğun bir iletim kuyruğu olmadığını belirtir.
3. ORANGE.LOCAL.QUEUE, bu komut başarısız olur. Bir kuyruğun var olan tanımlamasının üzerine yazmak istiyorsanız **REPLACE** özniteliğini kullanın, ancak aynı zamanda bkz. [“ALTER QLOCAL ya da DEFINE QLOCAL ile yerel kuyruk özniteliklerinin değiştirilmesi” sayfa 132.](#)

İlgili başvurular

[QLOCAL ' I TANIMLA](#)

Varsayılan nesne özniteliklerinin DISPLAY QUEUE ile görüntülenmesi

Bir IBM MQ nesnesi tanımlandığında varsayılan nesneden alınan öznitelikleri görüntülemek için **DISPLAY QUEUE** komutunu kullanabilirsiniz.

Bu görev hakkında

Bir IBM MQ nesnesi tanımladığınızda, bu nesne varsayılan nesneden belirtmediğiniz öznitelikleri alır. Örneğin, bir yerel kuyruk tanımladığınızda, kuyruk, tanımda çıkardığınız öznitelikleri varsayılan yerel kuyruktan (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, ancak **DEFINE** komutunda kuyruğun tipini belirtmeniz gerekir; örneğin, QLOCAL, QALIAS, QMODEL ya da QREMOTE.

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, yürürlükteki kuyruk derinliğini, yani kuyruktaki ileti sayısını içerir. Bu, görüntülemek için yararlı bir özniteliğe sahip, çünkü kuyruk derinliğini izleyerek, kuyruğun dolu olmamasını sağlayabilirsiniz.

İlgili başvurular

[GÖRÜNTÜLE](#)

[TANIMLAR](#)

Yerel kuyruk tanımlamasının DEFINE QLOCAL ile kopyalanması

DEFINE QLOCAL komutundaki **LIKE** özniteliğini kullanarak bir kuyruk tanımlamasını kopyalayabilirsiniz.

Bu görev hakkında

You can use the **DEFINE** command with the **LIKE** attribute to create a queue with the same attributes as the specified queue, rather than those of the system default local queue. Ayrıca, bir kuyruk tanımlamasını kopyalamak için **DEFINE** komutunun bu formunu kullanabilir, ancak özgün özniteliklerde bir ya da daha fazla değişiklik yerine başkasını koyabilirsiniz.

Notlar:

1. Bir **DEFINE** komutunda **LIKE** özniteliğini kullandığınızda, yalnızca kuyruk özniteliklerini kopyalısınız. Kuyruklardaki iletileri kopyalamazsınız.
2. Yerel bir kuyruk tanımladığınızda, **LIKE** belirtilmeden önce aşağıdaki gibi olur:

```
DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Yordam

- Sistem varsayılan yerel kuyruğundan değil, belirtilen kuyrukla aynı özniteliklere sahip bir kuyruk yaratmak için, aşağıdaki örnekte gösterildiği gibi **DEFINE** komutunu girin.

Kuyruğu yaratırken tam olarak girildiği şekilde kopyalanacak kuyruğun adını girin. Ad küçük harfli karakterler içeriyorsa, adı tek tırnak işareti içine alın.

Bu örnek, queueTURUNCU kuyrukla aynı özniteliklere sahip bir kuyruk yaratır.-KILITLE.Sistem varsayılan yerel kuyruğundan değil, QUEUE:(kuyruk):

```
DEFINE QLOCAL (MAGENTA.QUEUE) +
LIKE (ORANGE.LOCAL.QUEUE)
```

- Bir kuyruk tanımlamasını kopyalamak için, bir ya da daha çok değişikliği özgün özniteliklerine koyun; örneğin, aşağıdaki örnekte gösterildiği gibi **DEFINE** komutunu girin.

This command copies the attributes of the queue ORANGE.LOCAL.QUEUE to the queue THIRD.QUEUE, but specifies that the maximum message length on the new queue is to be 1024 bytes, rather than 4194304:

```
DEFINE QLOCAL (THIRD.QUEUE) +
```

```
LIKE (ORANGE.LOCAL.QUEUE) +  
MAXMSGL(1024);
```

İlgili başvurular

TANIMLAR

ALTER QLOCAL ya da DEFINE QLOCAL ile yerel kuyruk özniteliklerinin değiştirilmesi

You can change queue attributes in two ways, by using either the **ALTER QLOCAL** command or the **DEFINE QLOCAL** command with the **REPLACE** attribute.

Bu görev hakkında

Belirtilen yeni tanımlamayla var olan bir tanımı değiştirmek için, **ALTER** ve **DEFINE** komutlarının **REPLACE** özniteliğini kullanabilirsiniz. The difference between using **ALTER** and **DEFINE** is that **ALTER** with **REPLACE** does not change unspecified parameters, but **DEFINE** with **REPLACE** sets all the parameters.

Yordam

- Kuyruk özniteliklerini değiştirmek için, aşağıdaki örneklerde gösterildiği gibi, **ALTER** komutunu ya da **DEFINE** komutunu kullanın.
Bu örneklerde, kuyruktaki ileti uzunluğu üst sınırı ORANGE.LOCAL.QUEUE (kuyruk) değeri 10.000 byte 'a düştü.
 - **ALTER** komutunu kullanarak:

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

Bu komut, ileti uzunluğu üst sınırı olan tek bir özniteliği değiştirir; diğer tüm öznitelikler de aynı kalır.

- **DEFINE** komutunu **REPLACE** seçeneğiyle birlikte kullanarak, örneğin:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

Bu komut yalnızca ileti uzunluğu üst sınırını değil, aynı zamanda diğer tüm öznitelikleri de değiştirir; bu da varsayılan değerlerine sahip olur. Örneğin, kuyruk önceden engellenmiş ise, bu, kuyruk SYSTEM.DEFAULT.LOCAL.QUEUE.

Varolan bir kuyruktaki ileti uzunluğu üst sınırını azaltıyorsanız, var olan iletiler bundan etkilenmez. Ancak yeni mesajların yeni kriterlere uygun olması gerekir.

İlgili başvurular

ALTER kuyrukları

ALTER QLOCAL

TANIMLAR

QLOCAL ' I TANIMLA

Yerel kuyruğun CLEAR QLOCAL ile temizlenmesi

Yerel bir kuyruğu temizlemek için **CLEAR QLOCAL** komutunu kullanabilirsiniz.

Başlamadan önce

Şu durumda bir kuyruğu temizleyemezsiniz:

- Eşitleme noktası altında kuyruğa yerleştirmeye ilişkin kesinleştirilmemiş iletiler var.
- Şu anda bir uygulama kuyruğu açık durumda.

Bu görev hakkında

If you want to clear a local queue by using the **CLEAR QLOCAL** command, the name of the queue must be defined to the local queue manager.

Not: Fikrinizi deęiřtirmenize olanak tanıyan bir bilgi istemi yoktur; Enter tuřuna bastığınızda iletiler kaybedilir.

Yordam

İletileri yerel bir kuyruktan temizlemek için, ařaęıdaki örnekte gösterildięi gibi **CLEAR QLOCAL** ' i kullanın.

Bu örnekte, tüm iletiler MAGENTA.QUEUE:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

İlgili başvurular

[QLOCAL ' I TEMİZLE](#)

Yerel kuyruęun DELETE QLOCAL ile silinmesi

Yerel bir kuyruęu silmek için **DELETE QLOCAL MQSC** komutunu kullanabilirsiniz.

Bu görev hakkında

Kesinleřtirilmemiř iletiler içerirse, kuyruk silinemez.

Bir kuyruęun bir ya da daha çok kesinleřtirilmiř ileti varsa ve kesinleřtirilmemiř iletiler varsa, bu ileti yalnızca **PURGE** seçeneęini belirttiğinizde silinebilir. Silme iřlemi, adlandırılmıř kuyruklardaki kesinleřtirilmiř iletiler olsa bile devam eder ve bu iletiler de temizlenir.

PURGE yerine **NOPURGE** belirtilmesi, kuyruęun kesinleřtirilmiř iletiler içerip içermedięi için silinmemesini saęlar.

Yordam

- Yerel bir kuyruęu silmek için, ařaęıdaki örnekte gösterildięi gibi **DELETE QLOCAL** komutunu kullanın. Bu örnek, PINK.QUEUE (Kuyruk):

```
DELETE QLOCAL (PINK.QUEUE) NOPURGE
```

Bu örnek, PINK.QUEUE kuyruęunda kesinleřtirilmiř iletiler olsa bile:

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

İlgili başvurular

[QLOCAL SIL](#)

Örnek program ile kuyruklara göz atma

IBM MQ , kuyrukta bulunan iletilerin içerięine bakmak için kullanabileceğiniz bir örnek kuyruk tarayıcısı saęlar.

Bu görev hakkında

Tarayıcı, ařaęıdaki konumlarda hem kaynak hem de yürütülür biçimlerde saęlanır; burada **MQ_INSTALLATION_PATH** , IBM MQ ' in kurulu olduęu üst düzey dizini temsil eder.

Windows

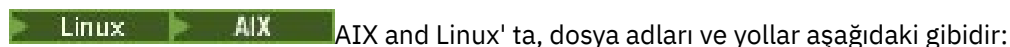
Windows' ta, örnek kuyruk tarayıcısının dosya adları ve yolları ařaęıdaki gibidir:

Kaynak

`MQ_INSTALLATION_PATH\tools\c\samples\`

Yürütülebilir

`MQ_INSTALLATION_PATH\tools\c\samples\bin\amqsbcg.exe`



Kaynak

`MQ_INSTALLATION_PATH/samp/amqsbcg0.c`

Yürütülebilir

`MQ_INSTALLATION_PATH/samp/bin/amqsbcg`

Yordam

- Örnek programı çalıştırmak için aşağıdaki örnekteki gibi bir komut girin.
Örnek program için iki giriş değişirgesi, iletilerin göz atılacağı kuyruğun adı ve bu kuyruğun sahibi olan kuyruk yöneticisi gereklidir. Örneğin:

```
amqsbcg SYSTEM.ADMIN.QMGREVENT.tpp01 saturn.queue.manager
```

Sonuçlar

Bu komuttaki tipik sonuçlar aşağıdaki örneklerde gösterilir:

```
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'00000000000000000000000000000000000000000000000000000'
   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 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 yapılmaksızın kullanılabilir.

Linux **AIX** AIX and Linux sistemlerinde, birden çok gigabayta ya da terabayta ilişkin kuyruk dosyaları yaratabilmeniz için büyük dosya desteğini belirttik olarak etkinleştirmeniz gerekir. Bunun nasıl yapacağına ilişkin bilgi için işletim sistemi belgelerinize 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 hakkında bilgi almak için işletim sistemi belgelerinizi denetleyin.

Kuyruklar için gereksinim duyduğunuz depolama miktarını planlamaya ilişkin bilgi için, platforma özgü performans raporları için [MQ Performans belgeleri](#) ' ne bakın.

V 9.2.0 IBM MQ 9.1.5' tan, yerel ve model kuyruklarında yeni bir öznitelik kullanarak kuyruk dosyalarının boyutunu denetleyebilirsiniz. Ek bilgi için [“IBM MQ kuyruk dosyalarının değiştirilmesi” sayfa 135](#) başlıklı konuya bakın.

Multi **V 9.2.0** **IBM MQ kuyruk dosyalarının değiştirilmesi**

IBM MQ 9.2.0' tan, yerel ve model kuyruklarındaki bir özniteliği kullanarak kuyruk dosyalarının boyutunu denetleyebilirsiniz. Bir kuyruk dosyasının yürürlükteki büyüklüğünü ve şu an için büyüyecek büyüklük üst sınırını (o dosyada kullanılmakta olan öbek büyüklüğüne dayalı olarak) iki kuyruk durumu özniteliklerini kullanarak görüntüleyebilirsiniz.

Kuyruk dosyalarını değiştirmek için kullanılan öznitelik

Yerel ve model kuyruklarındaki öznitelik şunlardır:

ÜST BÜYÜKLÜK

Kuyruk tarafından, megabayt cinsinden kullanılan kuyruk dosyasının büyüklük üst sınırını belirtir.

Ek bilgi için [MAXFSIZE](#) ve [“IBM MQ kuyruk dosyasının büyüklüğünün değiştirilmesi” sayfa 136](#) konusuna bakın.

The PCF attribute for this attribute is MQIA_MAX_Q_FILE_SIZE; see [Değişiklik, Kopyalama ve Oluşturma Kuyruğu](#).

Kuyruk durumundaki iki öznitelik şunlardır:

CURFSIZE

Kuyruk dosyasının geçerli boyutunu megabayt cinsinden, en yakın megabayta yuvarlanmış olarak görüntüler.

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

Bu öznitelige ilişkin PCF özniteliği MQIA_CUR_Q_FILE_SIZE ' dir.

CURMAXFS

Bir kuyrukta kullanılan yürürlükteki blok büyüklüğü göz önüne alındığında, kuyruk dosyasının büyüyebileceği büyüklük üst sınırını en yakın megabayta yuvarlayarak büyüyebileceğini gösterir.

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

Bu özneteliğe ilişkin PCF özneteliği MQIA_CUR_MAX_FILE_SIZE ' dir.

Bu iki PCF özneteliklerine ilişkin ek bilgi için bkz. [Sorgulama Kuyruğu](#) ve [Kuyruk Sorgulama Kuyruğu \(yanıt\)](#) .

Bu öznetelikleri, MQSC komutlarını, IBM MQ Explorer ve administrative REST APIMQSC komutlarını kullanarak ayarlayabilir ve görüntüleyebilirsiniz.

Not: You can only display the value of MAXFSIZE in the IBM MQ Console; you cannot configure the value.

Blok boyutu ve ayrıntı düzeyi

Kuyruk dosyaları, bloklar adı verilen kesimlere ayrılır. Bir kuyruk dosyası büyüklüğü üst sınırını artırmak için, kuyruk yöneticisi tarafından kuyruk büyüklüğü ya da öge boyu sıklığını değiştirmek gerekebilir.

Büyük bir MAXFSIZE değeriyle yeni tanımlanmış bir kuyruk yaratılırsa, kuyruk uygun bir öbek büyüklüğüyle yaratılır. Ancak, varolan bir kuyruğun MAXFSIZE değeri artırıldıysa, örneğin, ALTER QLOCAL komutunu kullanarak, kuyruk yöneticisinin kuyruğu yeniden yapılandırması için kuyruğun boşaltılmasını sağlamanız gerekebilir.

Ek bilgi için [“Bir IBM MQ kuyruk dosyasının ne kadar veri depolayabileceği hesaplanması” sayfa 137](#) başlıklı konuya bakın.



Uyarı: Bazı kütük sistemleri ve işletim sistemleri, tüm kütük sisteminin büyüklüğü ya da tek bir kütüğün büyüklüğü ile ilgili sınırlar içerir. İşletmelerinizin kullandığı sistemlerle ilgili sınırları denetlemeniz gerekir.

İlgili başvurular

[KUYRANLAR](#)

[GÖRÜNTÜLE](#)

[QSTATUS GÖRÜNTÜLE](#)

Muti **V 9.2.0** **IBM MQ kuyruk dosyasının büyüklüğünün değiştirilmesi**

Bir kuyruk dosyasının büyüklük üst sınırını artırabilir ya da azaltabilirsiniz.

Başlamadan önce

Kuyruk dosyası için yeni bir büyüklük ayarlama önce, değiştirmek istediğiniz kuyruk dosyasının büyüklüğünü görmek için [DISPLAY QLOCAL](#) komutunu kullanın. Örneğin, aşağıdaki 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 da, kuyruk dosyasının büyüklük üst sınırının 2,088,960 MB ' lik varsayılan değeri olduğunu gösterir.

Bu görev hakkında

Aşağıdaki yordamlarda aşağıdaki işlemlerin nasıl olduğunu gösteriniz:

- Kuyruk dosyasının büyüyebileceği büyüklük üst sınırını azaltın.
- Kuyruk dosyasının büyüyebileceği büyüklük üst sınırını artırın.



Uyarı: Uygulamalarınızın yazılı olduğu ve performansın olası etkisinin göz önüne alınmadan kuyruk dosyalarının boyutunu artırmanız için dikkatli olmanız gerekir. Çok büyük bir kuyruk dosyasında rasgele olan iletilere erişilmesi çok yavaş olabilir.

Bir kuyruk dosyasının maksimum boyutunu, varsayılan dışında artırmayı düşünüyorsanız, ilinti tanıtıcıları ve IBM MQ classes for JMS seçici dizgileri gibi ileti seçicilerini kullanmanıza dikkat edilmelidir. Daha büyük kuyruk dosyaları, kuyruğa ilk olarak ilk çıkış erişimi için daha uygundur.

Tek tek kuyruk dosyalarında çok büyük miktarda verinin olması, yalnızca döngüsel günlük kaydı için yapılandırılmış kuyruk yöneticilerinde ya da ortam görüntülemenin tek tek kuyruk için etkinleştirilmemiş olduğu kuyruk yöneticilerinde yapılmalıdır.

Kuyruk yöneticisinin işlemi etkileyebileceğinden, sistem kuyruklarının büyüklüğünü sınırlamamalısınız.

Yordam

1. Kuyruk dosyası büyüklüğü üst sınırını küçült

- a) Issue the following command to create a local file called SMALLQUEUE, with a size of 500 gigabytes :

```
DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
  2 : DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
AMQ8006I: IBM MQ queue created
```

ve şu iletiyi alırsınız: AMQ8006I:

Not: Bir kuyruğu, dosyadaki veri miktarından daha küçük bir değerle yapılandırırsanız, yeni iletiler kuyruğa konamaz.

Bir uygulama, yeterli alanı olmayan bir kuyruk dosyasına ileti yerleştirmeye çalışırsa, uygulama MQRC_Q_SPACE_NOT_AVASIAL dönüş kodunu alır. Kuyruktan yok edici sayıda ileti okunduğunda, uygulamalar kuyruğa yeni iletiler yerleştirmeye başlayabilir.

2. Kuyruk dosyası büyüklüğü üst sınırını artırın.

- a) LARGEQUEUE adlı yerel bir dosya yaratmak için 5 terabayt büyüklüğünde bir aşağıdaki komutu verin:

```
DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
  3 : DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
AMQ8006I: IBM MQ queue created
```

Multi V 9.2.0 Bir IBM MQ kuyruk dosyasının ne kadar veri depolayabileceği hesaplanması

Bir kuyruksa saklanabilen veri miktarı, kuyruğun bölüneceği tek tek öbeklerin büyüklüğünün büyüklüğünün sınırlanmasıdır.

Blok boyutu ve ayrıntı düzeyi

Varsayılan blok büyüklüğü 512 bayttır. İki terabayttan daha büyük kuyruk dosyalarını desteklemek için, kuyruk yöneticisinin öbek büyüklüğünü artırması gerekir.

The block size is automatically calculated when you configure the ÜST BÜYÜKLÜK for a queue, but the revised block size cannot be applied to the queue if the queue already has messages on it. Kuyruk boş olduğunda, kuyruk yöneticisi yapılandırılan MAXFSIZE ' yi desteklemek için otomatik olarak blok büyüklüğünü değiştirir.

DISPLAY QSTATUS komutunun yeni bir özneliği (CURMAXFS), yeni bir öbek büyüklüğü kullanacak şekilde değiştirildiğini doğrulamanıza olanak sağlayan yeni bir özneliği vardır.

Aşağıdaki örnekte, CURMAXFS değeri 4177920, kuyruk dosyasının şu anda yaklaşık olarak dört terabayta kadar büyüyebildiğini doğrulamaktadır. Kuyruksa yapılandırılan MAXFSIZE değeri, CURMAXFS değerinden büyükse, kuyruk yöneticisinin öbek büyüklüğünü yeniden yapılandırmadan önce kuyruk yöneticisi yine de kuyruğu boşaltılması için bekliyor.

```
DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
  2 : DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
AMQ8450I: Display queue status details
        QUEUE(LARGEQUEUE)                TYPE(Queue)
        CURMAXFS(4177920)                CURDEPTH(100000)
```

Kuyruk dosyası büyüklüğünün denetlenmesi

DISPLAY QSTATUS komutundaki CURFSIZE özniteliğini kullanarak, disk üzerinde bir kuyruk dosyasının yürürlükteki büyüklüğünü megabayt olarak görüntüleyebilirsiniz. This can be useful on platforms such as the IBM MQ Appliance, where it is not possible to access the file system directly.

```
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 ileti kaldırıldığında, CURFSIZE özniteliği hemen düşmeyebilir.

Genellikle, bir kuyruk dosyasındaki kullanılmayan alan yalnızca aşağıdaki durumlarda yayınlanır:

- Kuyruk açık hiçbir uygulama olmadığında
- Kuyruk yöneticisi günlüğüne 1000 yazdıktan sonra ya da
- Kuyruk yöneticisi kapatılıyor

İlgili başvurular

[KUYRANLAR](#)

[QSTATUS GÖRÜNTÜLE](#)

Uzak Kuyruklarla Çalışılması

Uzak kuyruk, uzak kuyruğun yerel tanımlarından biri. Yani, uzak kuyruk yöneticisindeki bir kuyruğa gönderme yapan yerel kuyruk yöneticisinde bir tanımlama.

Uzak bir kuyruğu yerel bir konumdan tanımlamanıza gerek yoktur; ancak, uzak kuyruğun uzak kuyruğa gönderme yapması, uzak kuyruğun bulunduğu kuyruk yöneticisinin tanıtıcısıyla nitelenmiş bir ad belirtmek yerine, uzak kuyruğa yerel olarak tanımlanmış bir adla gönderme yapabilmesini sağlar.

Uzak kuyruklar için yerel tanımlamalar nasıl çalışır

Bir uygulama yerel bir kuyruk yöneticisine bağlanır ve sonra bir MQOPEN çağrısını yayımlar. Açık aramada, belirtilen kuyruk adı, yerel kuyruk yöneticisinde bulunan bir uzak kuyruk tanımlamasıyla aynı. Uzak kuyruk tanımlaması, hedef kuyruğun, hedef kuyruk yöneticisinin ve isteğe bağlı olarak bir iletim kuyruğunun adını sağlar. Uzak kuyruğa ileti yerleştirmek için, uygulama MQOPEN çağrısından döndürülen tanıtıcıyı belirterek bir MQPUT çağrısını yayımlar. Kuyruk yöneticisi, iletinin başlangıcındaki bir iletim üstbilgisinde uzak kuyruk adını ve uzak kuyruk yöneticisi adını kullanır. Bu bilgi, iletiyi ağdaki doğru hedef noktaya yönlendirmek için kullanılır.

Denetimci olarak, uzak kuyruk tanımlamasını değiştirerek iletinin hedef noktasını denetleyebilirsiniz.

Aşağıdaki örnekte, bir uygulamanın uzak kuyruk yöneticisine ait bir kuyruğa nasıl ileti yerleştirdiğini gösteren örnek gösterilmektedir. Uygulama bir kuyruk yöneticisine bağlanıyor; örneğin, saturn.queue.manager. Hedef kuyruğun sahibi başka bir kuyruk yöneticisi var.

On the MQOPEN call, the application specifies these fields:

Alan değeri	Açıklama
<i>ObjectName</i> SIYAN.REMOTE.QUEUE	Uzak kuyruk nesnesinin yerel adını belirler. Bu, hedef kuyruğu ve hedef kuyruk yöneticisini tanımlar.
<i>ObjectType</i> (Kuyruk)	Bu nesneyi kuyruk olarak tanımlar.
<i>ObjectQmgrName</i> Boş ya da saturn.queue.manager	Bu alan isteğe bağlıdır. Boş bırakılırsa, yerel kuyruk yöneticisinin adı kabul edilir. (Bu, uzak kuyruk tanımlamasının var olduğu kuyruk yöneticidir.)

Bundan sonra, uygulama bu kuyruğa bir ileti koymak için bir MQPUT çağrısı yayınlar.

Yerel kuyruk yöneticisinde, aşağıdaki MQSC komutlarını kullanarak uzak kuyruğun yerel tanımlamasını yaratabilirsiniz:

```
DEFINE QREMOTE (CYAN.REMOTE.QUEUE) +
DESCR ('Queue for auto insurance requests from the branches') +
RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE) +
RQMNAME (jupiter.queue.manager) +
XMITQ (INQUOTE.XMIT.QUEUE)
```

Burada:

QREMOTE (CYAN.REMOTE.QUEUE)

Uzak kuyruk nesnesinin yerel adını belirler. This is the name that applications connected to this queue manager must specify in the MQOPEN call to open the queue AUTOMOBILE.INSURANCE.QUOTE.QUEUE on the remote queue manager jupiter.queue.manager.

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 belirler. Bu, CYAN.REMOTE.QUEUE kuyruk adını belirten uygulamalar tarafından gönderilen iletiler için gerçek hedef kuyruğdur. AUTOMOBILE.INSURANCE.QUOTE.QUEUE kuyruğu, uzak kuyruk yöneticisinde yerel kuyruk olarak tanımlanmalıdır.

RQMNAME (jupiter.queue.manager)

AUTOMOBILE.INSURANCE.QUOTE.QUEUE hedef kuyruğunun sahibi olan uzak kuyruk yöneticisinin adını belirtir.

XMITQ (INQUOTE.XMIT.QUEUE)

İletim kuyrukunun adını belirler. Bu isteğe bağlıdır; bir iletim kuyruğunun adı belirtilmediyse, uzak kuyruk yöneticisiyle aynı adı taşıyan bir kuyruk kullanılır.

In either case, the appropriate transmission queue must be defined as a local queue with a **Usage** attribute specifying that it is a transmission queue (USAGE(XMITQ) in MQSC commands).

İletileri uzak bir kuyruğa koymanın alternatif bir yolu

Uzak kuyruğun yerel tanımlamalarının kullanılması, iletileri uzak bir kuyruğa koymanın tek yolu değildir. Applications can specify the full queue name, including the remote queue manager name, as part of the MQOPEN call. Bu durumda, uzak kuyruğun yerel tanımlamalarına gerek yoktur. Ancak bu, uygulamaların çalıştırma zamanında uzak kuyruk yöneticisinin adını bilmesi ya da erişime sahip olması gerektiği anlamına gelir.

Diğer komutların uzak kuyruklarla kullanılması

Bir uzak kuyruk nesnesinin özniteliklerini görüntülemek ya da değiştirmek için MQSC komutlarını kullanabilir ya da uzak kuyruk nesnesini silebilirsiniz. Örnek:

- Uzak kuyruğun özniteliklerini görüntülemek için:

```
DISPLAY QUEUE (CYAN.REMOTE.QUEUE)
```

- Etkinleştirilecek uzak kuyruğu 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 tanımı:

```
DELETE QREMOTE (CYAN.REMOTE.QUEUE)
```

Not: Bir uzak kuyruğu sildiğinizde, yalnızca uzak kuyruğun yerel temsilini silmiş olur. Uzak kuyruğun kendisi ya da üzerinde herhangi bir ileti silmez.

Uzak kuyruk tanımlamalarının diğer adlar olarak kullanılması

Başka bir kuyruk yöneticisinde kuyruk bulmanın yanı sıra, kuyruk yöneticisi diğer adları ve yanıtlama kuyruğu diğer adları için uzak kuyruğun yerel tanımlamasını da kullanabilirsiniz. Her iki diğer ad tipi, uzak bir kuyruğun yerel tanımlamasıyla çözümler. İletinin hedefine ulaşması için uygun kanalları ayarlamamız gerekir.

Kuyruk yöneticisi diğer adları

Diğer ad, bir iletide belirtildiği gibi, hedef kuyruk yöneticisinin adının, ileti rotasındaki bir kuyruk yöneticisi tarafından değiştirildiğini belirten işlemdir. Kuyruk yöneticisi diğer adları önemlidir, çünkü bir kuyruk yöneticisi ağı içindeki iletilerin hedef hedefini denetlemek için bu adları kullanabilirsiniz.

Bunu yapmak için, denetim noktasındaki kuyruk yöneticisinde uzak kuyruk tanımlamasını değiştirerek bunu yapabilirsiniz. Gönderme uygulaması, belirtilen kuyruk yöneticisi adının bir diğer ad olduğunun farkında değil.

Kuyruk yöneticisi diğer adlarıyla ilgili ek bilgi için [Diğer adlar nedir?](#) başlıklı konuya bakın.

Yanıtlama kuyruğu diğer adları

İsteğe bağlı olarak, bir uygulama, bir kuyruğa *istek iletisi* koyduğunda, yanıtlanacak bir kuyruğun adını belirtebilir.

İletiyi işleyen uygulama, yanıt kuyruğun adını çıkardığında, gerekirse *yanıt iletisinin* nereye gönderileceğini bilir.

Yanıt kuyruğu diğer adı, bir istek iletisinde belirtildiği gibi, bir yanıt kuyruğunun ileti rotasındaki bir kuyruk yöneticisinde değiştirildiği işlemdir. Gönderme uygulaması, belirtilen yanıtlama kuyruğu adının bir diğer ad olduğunun farkında değil.

Yanıtlama kuyruğu diğer adı, yanıt kuyruğu adını ve isteğe bağlı olarak kuyruk yöneticisini değiştirmenize olanak sağlar. Bu, yanıt iletileri için hangi rotanın kullanıldığını denetlemenize olanak sağlar.

İstek iletileri, yanıt iletileri ve yanıt kuyrukları hakkında daha fazla bilgi için bkz. [İleti tipleri](#) ve [Yanıt kuyruğu ve kuyruk yöneticisi](#).

Yanıtlama kuyruğu diğer adlarına ilişkin ek bilgi için [Yanıt kuyruğu diğer adları ve kümeler](#) başlıklı konuya bakın.

Diğer ad kuyruklarıyla çalışılması

Başka bir kuyruğa ya da konuya dolaylı olarak gönderme yapmak için bir diğer ad kuyruğu tanımlayabilirsiniz.



Uyarı: Dağıtım listeleri, konu nesnelarini 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.

Diğer ad kuyruğunun gönderme yaptığı kuyruk aşağıdakilerden biri olabilir:

- Yerel bir kuyruk (bkz. [“DEFINE QLOCAL ile yerel kuyruk tanımlama” sayfa 129](#)).
- Uzak kuyruğun yerel tanımlaması (bkz. [“Uzak Kuyruklarla Çalışılması” sayfa 138](#)).
- 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.

Diğer ad kuyruğu tanımlanması

Aşağıdaki komut 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 komut, ALPHA uygulaması için etkinleştirilen ve devre dışı bırakılan bir diğer adı tanımlar:

```
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 komutların diğer ad kuyruklarıyla kullanılması

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 başvurular

[QALIAS DEĞİŞTİR](#)

[QALIAS TANIMLAYIN](#)

[QDİĞER ADI SIL](#)

[Dağıtım listeleri](#)

Model kuyruklarıyla çalışma

Kuyruk yöneticisi, bir uygulama tarafından model kuyruğu olarak tanımlanmış bir kuyruk adı belirleyen bir uygulamadan gelen bir MQI çağrısı alırsa, *dinamik kuyruk* yaratır. Kuyruk yaratıldığında, kuyruk yöneticisi yeni dinamik kuyruğun adını içerir. *Model kuyruğu*, bu şablondan yaratılan dinamik kuyrukların özniteliklerini belirleyen bir şablondur. Model kuyrukları, uygulamalar için gerekli olan kuyruklar yaratması için uygun bir yöntem sağlar.

Model kuyruğu tanımlanması

You use the **DEFINE QMODEL** command to define a model queue with a set of attributes in the same way that you define a local queue. Model kuyrukları ve yerel kuyruklar, yaratılan dinamik kuyrukların geçici mi kalıcı mı, yoksa kalıcı mı olduğunu belirleyebileceğiniz model kuyruklarında aynı öznitelikler

kümesine sahiptir. (Kuyruk yöneticisi yeniden başlatma işlemlerinde kalıcı kuyruklar sürdürür, geçici sorgular sürdürmez.) 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, **SYSTEM.DEFAULT.MODEL.QUEUE** varsayılan kuyruğu.

Model kuyruklarını tanımladığınızda, bunları yerel kuyruklarla aynı şekilde tanımladığınızda, **LIKE** ve **REPLACE** özniteliklerini kullanabilirsiniz.

Model kuyruklarıyla diğer komutların kullanılması

Bir model kuyruğunun özniteliklerini görüntülemek ya da değiştirmek ya da model kuyruğu nesnesini silmek için uygun **MQSC** komutlarını kullanabilirsiniz. Örneğin:

Örnek kuyruğun özniteliklerini görüntülemek için **DISPLAY QUEUE** komutunu kullanın:

```
DISPLAY QUEUE (GREEN.MODEL.QUEUE)
```

Bu modelden yaratılan herhangi bir dinamik kuyruğa yerleştirmeyi 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

[ALTER QMODEL](#)

[QMODEL ' I TANIMLA](#)

[QMODEMI SIL](#)

[GÖRÜNTÜLE](#)

Ölü-harfli kuyruklarla çalışma

Her kuyruk yöneticisinin genellikle bir yerel kuyruğu, bir ölü-mektup kuyruğu olarak kullanması için, doğru hedefe teslim edilemeyen iletilerin daha sonra alınabilmesi için saklanabileceği bir kuyruktur. Kuyruk yöneticisine, ölü-mektup kuyruğundan bahsedin ve bir ölü-mektup kuyruğunda bulunan iletilerin nasıl işleneceğini belirtin. Ölü harf kuyruklarının kullanılması, iletilerin teslim edildiği sırayı etkileyebilir, bu nedenle bunları kullanmamayı tercih edebilirsiniz.

To tell the queue manager about the dead-letter queue, specify a dead-letter queue name on the **crtmqm** command (**crtmqm -u DEAD.LETTER.QUEUE**, for example), or by using the **DEADQ** attribute on the **ALTER QMGR** command to specify one later. Bu işlemi kullanmadan önce, kuyruk-harf kuyruğunu tanımlamanız gerekir.

SYSTEM.DEAD.LETTER.QUEUE (Kuyruk) ürünle birlikte kullanılabilir. Kuyruk yöneticisi yarattığınızda bu kuyruk otomatik olarak yaratılır. Gerekirse bu tanımlamayı değiştirebilir ve yeniden adlandırabilirsiniz.

Bir ölü-mektup kuyruğunda aşağıdakiler dışında özel bir gereksinim yoktur:

- Yerel bir kuyruk olmalıdır
- Its MAXMSGL (maximum message length) attribute must enable the queue to accommodate the largest messages that the queue manager has to handle **artı** the size of the dead-letter header (MQDLH)

Ölü harf kuyruklarının kullanılması, iletilerin teslim edildiği sırayı etkileyebilir, bu nedenle bunları kullanmamayı tercih edebilirsiniz. USEDQL kanal özniteliğini, iletiler teslim edilemediğinde, ölü-harf kuyruğunun kullanılıp kullanılmadığını belirlemek için ayarlırsınız. Bu öznitelik, kuyruk yöneticisinin bazı işlevlerinin ölü-mektup kuyruğunu kullanması için yapılandırılabilirken, diğer işlevler de yapılandırılmayabilir. Farklı MQSC komutlarındaki USEDQL kanal özniteliğinin kullanımıyla ilgili ek bilgi için [DEFINE CHANNEL](#), [DISPLAY CHANNEL](#), [ALTER CHANNEL](#) ve [DISPLAY CLUSQMGR](#) başlıklı konuya bakın.

IBM MQ , bir ölü-mektup kuyruğunda bulunan iletilerin nasıl işleneceğini ya da kaldırılacağı hakkında bilgi vermenizi sağlayan, çıkmaz bir kuyruk işleyicisi sağlar. Bkz. [“Processing messages on an IBM MQ dead-letter queue”](#) sayfa 144.

İlgili kavramlar

[Ölü-harf kuyrukları](#)

İlgili görevler

[Teslim edilmeyen iletiler sorun giderme](#)

İlgili başvurular

[ALTER QMGR](#)

[crtmqm \(kuyruk yöneticisi yarat\)](#)

Processing messages on an IBM MQ dead-letter queue

İletileri bir ölü-mektup kuyruğunda (DLQ) işlemek için, IBM MQ varsayılan bir DLQ işleyicisi sağlar. İşleyici, tanımladığınız bir kural çizelgesindeki girişlere karşı DLQ 'daki iletilerde eşleşir.

İletiler, kuyruk yöneticilerine, ileti kanalı araçlarına (MCA 'lar) ve uygulamalara göre bir DLQ'ya yerleştirilebilir. All messages on the DLQ must be prefixed with a *dead-letter üstbilgisi* structure, MQDLH. Bir kuyruk yöneticisi ya da ileti kanalı aracısı tarafından DLQ 'ya ilişkin iletiler her zaman bu üstbilgiye sahip olur; DLQ'ya ileti yerleştiren uygulamalar bu üstbilgiyi sağlamalıdır. MQDLH yapısının *Reason* (Neden) alanı, iletinin DLQ 'da neden olduğunu belirten bir neden kodu içerir.

Tüm IBM MQ ortamlarında, iletileri düzenli olarak DLQ üzerinde işlemek için bir yordama gerek vardır. IBM MQ , `runmqdlq` komutunu kullanarak çağırdığınız *ölü-harfli kuyruk işleyicisi* (DLQ işleyicisi) adı verilen varsayılan bir yordama sağlar.

DLQ 'daki iletilerin işlenmesine ilişkin yönergeler, kullanıcı tarafından yazılan *kural çizelgesi* aracılığıyla DLQ işleyicisine sağlanır. Yani, DLQ işleyicisi, kurallar çizelgesindeki girişlere karşı DLQ 'daki iletilerle eşleşir; bir DLQ iletilisi kural çizelgesindeki bir girişle eşleştiğinde, DLQ işleyicisi bu girişle ilişkili işlemi gerçekleştirir.

İlgili kavramlar

[Ölü-harf kuyrukları](#)

İlgili görevler

[Teslim edilmeyen iletiler sorun giderme](#)

IBM i **IBM i' ta ölü-harfli kuyruk işleyicisi**

IBM i 'in ölme harfli kuyruk işleyicisi nedir ve bunu nasıl başlatıyorsunuz?

Bazen *teslim edilemeyen ileti kuyruğu* olarak da bilinen bir *kuyruk-çıkış kuyruğu* (DLQ), hedef kuyruklarına teslim edilemeyen iletiler için bir tutma kuyruğudur. Bir ağdaki her kuyruk yöneticisinin ilişkili bir DLQ 'ya sahip olması gerekir.

Not: Genellikle bir DLQ 'ya ileti yerleştirmekten kaçınmak tercih edilir. DLQ 'ların kullanımı ve önlenmesi hakkında bilgi için bkz. [“Ölü-harfli kuyruklarla çalışma”](#) sayfa 143.

Kuyruk yöneticileri, ileti kanalı araçları ve uygulamalar DLQ ' ya ileti yerleştirebilir. All messages on the DLQ must be prefixed with a *dead-letter üstbilgisi* structure, MQDLH. DLQ 'ya bir kuyruk yöneticisi tarafından ya da bir ileti kanalı aracısı tarafından gönderilen iletiler her zaman bir MQDLH' ye sahiptir. DLQ ' ya ileti koymak için her zaman bir MQDLH belirtin. MQDLH yapısının *Reason* (Neden) alanı, iletinin DLQ ' da neden olduğunu belirten bir neden kodu içerir.

Tüm IBM MQ ortamlarında, DLQ ' da iletileri işlemek için düzenli olarak çalışan bir yordam olmalıdır. IBM MQ , STRMQMDLQ komutunu kullanarak çağırdığınız *-Ölü harf kuyruğu işleyicisi* (DLQ işleyicisi) adı verilen varsayılan bir yordamı sağlar. Kullanıcı tarafından yazılan *kurallar çizelgesi* , DLQ ' ya ilişkin iletileri işlemek için DLQ işleyicisine ilişkin yönergeleri sağlar. DLQ işleyicisi, kural çizelgesindeki girişlere karşı DLQ ' daki iletilerle eşleşir. Bir DLQ iletilisi, kural çizelgesindeki bir girişle eşleştğinde, DLQ işleyicisi bu girişle ilişkili işlemi gerçekleştirir.

DLQ işleyicisini çağırma

DLQ işleyicisini çağırmak için STRMQMDLQ komutunu kullanın. İşlemek istediğiniz DLQ ' yı ve kullanmak istediğiniz kuyruk yöneticisini iki şekilde adlayabilirsiniz:

- Komut isteminden STRMQMDLQ ' un deęiřtirgeleri olarak. Örneęin:

```
STRMQMDLQ UDLMSGQ(ABC1.DEAD.LETTER.QUEUE) SRCMBR(QRULE) SRCFILE(library/QTXTSRC)
MQMNAME(MY.QUEUE.MANAGER)
```

- Kurallar tablosunda. Örneęin:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE)
```

Not: Kurallar çizelgesi, herhangi bir adı alabilen kaynak fiziksel dosya içindeki bir üyedir.

Örnekler, varsayılan kuyruk yöneticisinin iyelięindeki ABC1 . DEAD . LETTER . QUEUEadlı DLQ için geçerlidir.

DLQ ya da kuyruk yöneticisini gösterildięi gibi belirtmezseniz, kuruluş için varsayılan kuyruk yöneticisi, o kuyruk yöneticisine ait DLQ ' yla birlikte kullanılır.

STRMQMDLQ komutu, girişini kural çizelgesinden alır.

DLQ işleyicisini çalıştırmak için, DLQ ' ya hem kendisine, hem de DLQ üzerindeki iletilerin iletileceęi ileti kuyruklarına erişmeniz için yetki verilmelidir. Ayrıca, dięer kullanıcıların kimlięini, DLQ ' nun ileti bağlamındaki kullanıcı kimlięinin yetkisi olan kuyruklara ileti konmasını sağlamak için, bu kullanıcıların kimlięini de varsaymak için yetki verilmelidir.

İlgili kavramlar

[Ölü-harf kuyrukları](#)

İlgili görevler

[Teslim edilmeyen iletiler sorun giderme](#)

IBM i *IBM i üzerinde DLQ işleyici kuralları çizelgesi*

Gelmeyen ileti kuyruęu işleyicisi kural çizelgesi, DLQ işleyicisinin IBM i DLQ ' ya gelen iletileri nasıl işleyeceęini tanımlar.

DLQ işleyicisi kural çizelgesi, DLQ işleyicisinin DLQ ' ya gelen iletileri nasıl işleyeceęini tanımlar. Bir kural çizelgesinde iki tip giriş vardır:

- Çizelgedeki isteęe baęlı ilk giriş *denetim verilerini* içerir.
- Çizelgedeki dięer tüm girişler, DLQ işleyicisinin izlemesine ilişkin *kurallardır* . Her kural, bir iletinin eşleřtirildięi bir *örüntüden* (ileti özellikleri kümesi) ve DLQ ' daki bir ileti belirtilen örüntüyle eşleřtięinde yapılacak bir *iřlemden* 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 (*QueueManagerAdı* | ' ' (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 değiştirgesi olarak DLQ ' nun adını, denetim verilerinin kurallar çizelgesine eklenmesine alternatif olarak girebilirsiniz. Hem kurallar çizelgesinde hem de **STRMQMDLQ** komutuna girişte herhangi bir değer belirtilirse, **STRMQMDLQ** komutunda belirtilen değer önceliklidir.

Not: Kurallar çizelgesinde bir denetim verileri girişi varsa, bu giriş çizelgedeki ilk giriş olmalıdır.

DLQ rules (patterns and actions) on IBM i

IBM i ' in ölü-mektup kuyruğu kurallarının her birine ilişkin örüntüler ve işlemlerin bir açıklaması.

Bir DLQ işleyicisi kural çizelgesinden örnek bir kural vardır:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Bu kural, DLQ işleyicisine, MQPUT ve MQPUT1 engellendiğinden, hedef kuyruğuna (DLQ) yerleştirilecek her kalıcı iletiyi göndermek için 3 girişimde bulunmaya yardımcı olur.

Bu kısım, bir kurala dahil edebildiğiniz anahtar sözcükleri açıklar. Aşağıdakileri unutmayın:

- Bir anahtar sözcüğün varsayılan değeri (varsa) altı çizilir. Çoğu anahtar sözcük için, varsayılan değer * (yıldız işareti), herhangi bir değerle eşleşir.
- Dikey çizgi (|) alternatifleri ayırır. Bunlardan yalnızca birini belirtebilirsiniz.
- ACTION dışındaki tüm anahtar sözcükler isteğe bağlıdır.

Bu kısım, örüntü eşleştirme anahtar sözcüklerinin (DLQ 'daki iletilerin eşleştirilmesiyle karşılaştırılır) bir tanımlamayla başlar. Daha sonra, işlem anahtar sözcüklerini (DLQ işleyicinin eşleşen bir iletiyi işleyeceğini saptayanlar) açıklanır.

IBM i IBM i üzerinde DLQ örüntü eşleştirme anahtar sözcükleri

Örüntü eşleştirme anahtar sözcükleri bir örnekte açıklanmıştır. IBM i ileti kuyruğundaki iletilerin eşleştirileceği değerleri belirlemek için bu anahtar sözcükleri kullanın. Kalıpla eşleşen tüm anahtar sözcükler isteğe bağlıdır.

APPLIDAT (*ApplIdentityData* | * (varsayılan)

İleti tanımlayıcısı MQMD 'de belirtilen DLQ 'daki iletinin *ApplIdentityData* (Veri) değeri.

APPLNAME (*PutApplAdi* | * (varsayılan)

DLQ üzerindeki iletinin tanımlayıcısının (MQMD) *PutApplName* alanında belirtildiği 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 gönderildiği ileti kuyruğunun adı.

DESTQM (*QueueManagerAdi* | * (varsayılan)

İletin gönderildiği ileti kuyruğunun kuyruk yöneticisi adı.

FEEDBACK (*Geribildirim* | * (varsayılan)

MsgType değeri MQMT_REPORT olduğunda, *Geribildirim* raporun doğasını açıklar.

Sembolik adlar kullanabilirsiniz. Örneğin, hedef kuyruklarına varışlarının doğrulanmasını gerektiren DLQ 'da bu 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 gerektiren 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 'da kalıcı olan 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 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 (*QueueManagerAdi* | * (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 (*QueueManagerAd* | & DESTQM | & REPLYQM | ' (varsayılan))

Bir iletinin iletildiği kuyruğun kuyruk yöneticisi.

QueueManagerAd

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 üzerinde DLQ kural çizelgesi kuralları*

IBM i ölü harf kuyruğu kuralları çizelgesi, sözdizimiyle, yapısıyla ve içerikleriyle ilgili özel kurallara uymalıdır.

- Bir kural çizelgesinin en az bir kural içermesi gerekir.
- Anahtar sözcükler herhangi bir sırada olabilir.
- Bir anahtar sözcük yalnızca herhangi bir kuralda bir kez eklenebilir.
- Anahtar sözcükler büyük ve küçük harfe duyarlı değildir.
- Bir anahtar sözcük ve onun değiştirge değeri, en az bir boşluk ya da virgül ile diğer anahtar sözcüklerden ayrılmalıdır.
- Herhangi bir sayıda boşluk, bir kuralın başında ya da sonunda ve anahtar sözcükler, noktalama işaretleri ve değerler arasında olabilir.
- Her kural yeni bir satırda başlamalıdır.
- Taşınabilirlik için, çizginin önemli uzunluğunun 72 karakterden uzun olmamalıdır.
- Kuralın sonraki satırın ilk boş olmayan karakterinden devam ettiğini göstermek için, bir satırda son boş olmayan karakter olarak artı işaretini (+) kullanın. Kuralın sonraki satırın başlangıcından devam ettiğini göstermek için eksi işaretini (-) bir satırda son boş olmayan karakter olarak kullanın. Anahtar sözcükler ve değiştirgelerde devam karakterleri oluşabilir.

Örneğin:

```
APPLNAME('ABC+  
D')
```

'ABCD' ile sonuçlanır.

```
APPLNAME('ABC-  
D')
```

'ABC D' nin sonuçları.

- Yıldız işareti (*) ile başlayan açıklama satırları, kural çizelgesinin herhangi bir yerinde oluşabilir.
- Boş satırlar yoksayıılır.
- DLQ işleyici kuralları çizelgesindeki her giriş bir ya da daha çok anahtar sözcükten ve ilişkili değiştirgelerinden oluşur. Değiştirgelerin sözdizimi kurallarına uygun olması gerekir:
 - Her parametre değerinin en az bir anlamlı karakter içermesi gerekir. Tırnak içine alınan değerlerdeki sınırlayıcı tırnak işaretleri anlamlı olarak kabul edilmez. Örneğin, bu değiştirgeler 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 deęiřtirgeler önemli karakterler içermedięi için geçersizdir:

FORMAT (' ')
 FORMAT ()
 FORMAT ()
 FORMAT

- Genel arama karakterleri desteklenir. Soru işareti (?) herhangi bir tek karakterin yerine, sonda boşluk dışında bir boşluk dışında kullanabilirsiniz. Sıfır ya da daha çok bitişik karakter yerine yıldız (*) işareti kullanabilirsiniz. Yıldız imi (*) ve soru işareti (?), **her zaman** parametre değerlerinde genel arama karakteri olarak yorumlanır.
- Bu anahtar sözcüklerin parametrelerinde genel arama karakterleri içeremez: ACTION, HEADER, RETRY, FWDQ, FWDQM ve PUUTUT.
- Parametre değerlerinde sondaki boşluklar ve DLQ üzerindeki iletilerde karşılık gelen alanlarda, joker karakter eşleşmeleri gerçekleştirilirken önemli değildir. Ancak, tırnak içindeki dizgiler içinde baştaki ve gömülü boşluklar, genel arama karakterlerine önem vermekle önemlidir.
- Sayısal parametreler soru işareti (?) genel arama karakteri içeremez. Sayısal bir parametrenin tamamının yerine yıldız (*) işareti ekleyebilirsiniz, ancak yıldız işareti sayısal bir parametrenin parçası olarak içerilemez. Örneğin, bunlar geçerli sayısal deęiřtirgelerdir:

MSGTYPE (2)	Yalnızca yanıt iletileri seçilebilir
MSGTYPE (*)	Herhangi bir ileti tipi seçilebilir
MSGTYPE (' * ')	Herhangi bir ileti tipi seçilebilir

Ancak, sayısal bir parametrenin parçası olarak bir yıldız işareti (*) içerdüğinden MSGTYPE (' 2 * ') geçerli değildir.

- Sayısal parametreler 0-999 999 999 aralığında olmalı. Parametre değeri bu aralıktaki değerse, anahtar sözcüğün ilişkili olduğu alanda geçerli olmasa da kabul edilir. Sayısal parametreler için simgesel adlar kullanabilirsiniz.
- Bir dizilim değeri, anahtar sözcüğün ilgili olduğu MQDLH ya da MQMD ' deki alandan kısaysa, değer, alanın uzunluğuna kadar boşluklarla doldurulur. Yıldız imleri hariç olmak üzere değer, alandan daha uzunsa bir hataya neden olur. Örneğin, 8 karakterlik bir alana ilişkin tüm geçerli dizgi değerleri şunlardır:

' ABCDEFGH '	8 karakter
' A*C*E*G*I '	Yıldız imleri dışında 5 karakter
' *A*C*E*G*I*K*M*O* '	Yıldız imleri dışında 8 karakter

- Boşluk, küçük harf karakterleri ya da nokta (.), eğik çizgi (?), alt çizgi (_) ve yüzde imi (%) içeren dizgiler tek tırnak işareti içine alınmalıdır. Tırnak işareti içine alınmayan küçük harfli karakterler büyük harfe çevrilir. Dizgide bir tırnak işareti varsa, hem başlangıç hem de bitiş sonunu göstermek için iki tek tırnak işareti kullanılmalıdır. Dizilimin uzunluğu hesaplanırken, çift tırnak işaretlerinin her bir oluşumu tek bir karakter olarak sayılır.

IBM üzerinde DLQ kural çizelgesinin işlenmesi

Ölü harf kuyruğu işleyicisi, IBM i ölü-harfli kuyruğunda bir iletiyle eşleşen bir örüntüye sahip bir kural için kural çizelgesini arar.

Arama, tablodaki ilk kuralla başlar ve tablo aracılığıyla sırayla devam eder. Eşleşen bir örüntüye sahip bir kural bulunduğunda, kural çizelgesi bu kuraldan işlemi dener. DLQ işleyicisi, bu kuralı uygulamak için

her girişimde bulunduğu, bir kural için yeniden deneme sayısını 1 artırır. İlk deneme başarısız olursa, yapılan girişimler sayısı, RETRY anahtar sözcüğünde belirtilen sayıyla eşleşinceye kadar yinelenir. Tüm denemeler başarısız olursa, DLQ işleyicisi çizelgede sonraki eşleşen kuralı arar.

Bu işlem, bir işlem başarılı oluncaya kadar sonraki eşleştirme kuralları için yinelenir. Her eşleştirme kuralı RETRY anahtar sözcüğünde belirtilen sayıda kez denendiğinde ve tüm girişimler başarısız olduğunda, ACTION (IGNORE) varsayılır. Eşleşen bir kural bulunmazsa, ACTION (IGNORE) da varsayılır.

Not:

1. Eşleşen kural örüntüleri yalnızca, bir MQDLH ile başlayan DLQ 'daki iletiler için aranır. MQDLH ile başlamayan iletiler düzenli aralıklarla hata olarak bildirilir ve DLQ 'da süresiz olarak kalır.
2. Bir kuralın yalnızca bir eylemden oluşabilmesi için, tüm örüntü anahtar sözcükleri varsayılan olarak varsayılan değer olabilir. Ancak, kuyrukta MQDLHs içeren ve çizelgeye ilişkin diğer kurallara uygun olarak işlenmemiş olan tüm iletilere yalnızca işlem kuralları uygulanmaktadır.
3. Kural çizelgesi, DLQ işleyicisi başlatıldığında doğrulanır ve o sırada işaretlenen hatalar olur. (DLQ işleyicisi tarafından verilen hata iletileri Messages and reason codesiçinde açıklanmıştır.) Kural çizelgesinde istediğiniz zaman değişiklik yapabilirsiniz, ancak DLQ işleyicisi yeniden başlatılıncaya kadar bu değişiklikler yürürlüğe girmez.
4. DLQ işleyicisi, iletilerin içeriğini, MQDLH 'yi ya da ileti tanımlayıcısını değiştirmez. DLQ işleyicisi her zaman, iletileri MQPMO_PASS_ALL_CONTEXT ileti seçeneği ile diğer kuyruklara koyar.
5. Kural çizelgesinde art arda sözdizimi hataları tanınmamış olabilir; kural çizelgesinin geçerlilik denetimi, yinelenen hataların oluşturulmasını ortadan kaldırır.
6. DLQ işleyicisi, DLQ 'yı MQOO_INPUT_AS_Q_DEF seçeneğiyle açar.
7. Aynı kural çizelgesini kullanarak, DLQ işleyicisinin birden çok eşgörünümü aynı kuyruğa koşut olarak koşut zamanlı olarak çalışabilir. Ancak, bir DLQ ile DLQ işleyicisi arasında bire bir ilişki olması daha olağan bir durum.

IBM i

IBM üzerinde tüm DLQ iletilerinin işlenmesini sağlama

The dead-letter queue handler keeps a record of all messages on the IBM i DLQ that have been seen but not removed.

DLQ işleyicisini, DLQ 'dan gelen iletilerin küçük bir alt kümesini çıkarmak için süzgeç olarak kullanırsanız, DLQ işleyicisi, DLQ üzerinde işlem yapmadığı iletilerin bir kaydını tutar. DLQ işleyicisi, DLQ 'ya gelen yeni iletilerin, ilk giren ilk giren (FIFO) olarak tanımlansa bile, DLQ'ya yeni gelen iletilerin görüleceğini garanti edemez. Kuyruk boş değilse, DLQ, tüm iletileri denetlemek için düzenli aralıklarla yeniden taranır.

Bu nedenlerle, DLQ 'nın mümkün olduğunca az sayıda ileti içerdiğini doğrulayın. Atılmayan ya da başka kuyruklara iletilenmeyen iletiler (herhangi bir nedenle) kuyrukta birikmesine izin verilirse, DLQ işleyicisinin iş yükü artar ve DLQ 'nın kendisi doldurma tehlikesiyle karşı karşılanır.

DLQ işleyicinin DLQ 'yı boşaltmasını sağlamak için belirli ölçümler yapabilirsiniz. Örneğin, DLQ 'da ileti bırakan ACTION (IGNORE) kullanmayın. (Tablodaki diğer kurallar tarafından açık bir şekilde ele alınmayan iletiler için ACTION (IGNORE) varsayılanını unutmayın.) Bunun yerine, başka bir şekilde yoksaymak istediğiniz iletiler için, iletileri başka bir kuyruğa taşıyan bir işlem kullanın. Örneğin:

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

Benzer şekilde, tabloda daha önceki kurallar tarafından ele alınmamış iletileri işlemek için bir catchall tablolarında son kuralı yapın. Örneğin, tabloda son kural şöyle bir şey olabilir:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Bu, tablodaki son kurala kadar geçen iletilerin, el ile işlenebileceği REALLY . DEAD . QUEUEkuyruğuna iletilebilmesine neden olur. Böyle bir kuralınız yoksa, iletilerin DLQ 'da süresiz olarak kalabileceğini kabul edin.

IBM i' ta bir ölü-harflı kuyruk işleyicisi kural çizelgesi için örnek kod. Bu örnek kural çizelgesi, tek bir denetim-veri girişi 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)
```

DLQ işleyicisini çağırma

Invoke the dead-letter queue handler using the Runmqdlq command. İşlemek istediğiniz DLQ ' yı ve kullanmak istediğiniz kuyruk yöneticisini iki şekilde adlayabilirsiniz.

İki yol aşağıdaki gibidir:

- Komut isteminden Runmqdlq ' e parametre olarak. Örneğin:

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER <qrule.ru1
```

- Kurallar tablosunda. Örneğin:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

Örnekler, ABC1.DEAD.LETTER.QUEUE(kuyruk yöneticisi) ABC1.QUEUE.MANAGER

DLQ ya da kuyruk yöneticisini gösterildiği gibi belirtmezseniz, kuruluş için varsayılan kuyruk yöneticisi, o kuyruk yöneticisine ait DLQ ' yla birlikte kullanılır.

The Runmqdlq command takes its input from stdin ; you associate the rules table with Runmqdlq by redirecting stdin from the rules table.

DLQ işleyicisini çalıştırmak için, DLQ ' ya hem kendisine, hem de DLQ üzerindeki iletilerin iletileceği ileti kuyruklarına erişmeniz için yetki verilmelidir. DLQ işleyicisi için, ileti bağlamındaki kullanıcı kimliğinin yetkisi olan kuyruklara ileti yerleştirmesi için, diğer kullanıcıların kimliğini de üstlenmek için yetkilendirilmiş olmanız gerekir.

runmqdlq komutuna ilişkin ek bilgi için [runmqdlq](#) başlıklı konuya bakın.

İlgili kavramlar

[Ölü-harf kuyrukları](#)

İlgili görevler

[Teslim edilmeyen iletiler 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
```

V 9.2.3 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 bağlanacağınızı tanımlamak üzere ayarlamamız gerekir. Örneğin:

```
export MQSERVER="SYSTEM.DEF.SVRCONN/TCP/myappliance.co.uk(1414)"
```

DLQ işleyici kuralları çizelgesi

Ölü-harfli kuyruk işleyicisi kural çizelgesi, DLQ işleyicisine ilişkin iletilerin DLQ ' ya nasıl işleneceğini tanımlar.

Bir kural çizelgesinde iki tip giriş vardır:

- Çizelgedeki ilk giriş (isteğe bağlı) *denetim verileri* içerir.
- Çizelgedeki diğer tüm girişler, izlenecek DLQ işleyicisine ilişkin *kurallar* ' dır. Her kural, bir iletinin karşılaştırıldığı bir *örüntüden* (ileti özellikleri kümesi) ve DLQ üzerindeki bir ileti belirtilen örüntüyle eşleştirmede alınacak bir *işlem* ile oluşur. Bir kural çizelgesinde en az bir kural olmalıdır.

Kural çizelgesindeki her giriş bir ya da daha çok anahtar sözcükten oluşur.

İlgili kavramlar

Ölü-harf kuyrukları

İlgili görevler

Teslim edilmeyen iletiler 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. runmqd1q komutu için parametre olarak sağladığınız herhangi bir INPUTQ değeri, kurallar çizelgesindeki herhangi bir INPUTQ değerini geçersiz kılar.
2. runmqd1q 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, runmqd1q komutu için değiştirge olarak verilen addır.
4. runmqd1q 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 (QueueManagerAdi| ' ' (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 başlatmabaşlıklı konuya](#) bakın.

Denetim verilerini kural çizelgesine dahil etmek yerine, runmqdlq komutunun giriş deęiřtirgeleri olarak DLQ ve kuyruk yöneticisi adlarını belirtmeniz gerekir. Hem kurallar çizelgesinde hem de runmqdlq komutuna giriş olarak bir deęer belirtirseniz, runmqdlq komutunda belirtilen deęer önceliklidir.

Kurallar çizelgesine bir denetim verileri giriři eklerseniz, bu giriş çizelgedeki **ilk** giriş olmalıdır.

DLQ kuralları (örüntüler ve işlemler)

Kalıpla eşleşen anahtar sözcüklerin (ileti kuyruğundaki iletilerin eşleştięi 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 eşleştirileceęi deęerleri belirtmek için kullandığınız örüntü eşleştirme anahtar sözcükleri aşağıdaki gibidir. (Tüm örüntü eşleştirme anahtar sözcükleri isteęe baęlıdır):

APPLIDAT (ApplIdentityData|* (varsayılan)

DLQ 'daki iletinin ileti tanımlayıcısında (MQMD) belirtilen *ApplIdentityVerileri* deęeri.

APPLNAME (PutApplAdi|* (varsayılan))

DLQ 'daki iletinin tanımlayıcısının (MQMD) *PutApplName* alanında belirtildięi şekilde, MQPUT ya da MQPUT1 çağırısını yayınlayan uygulamanın adı.

APPLTYPE (PutApplTip|* (varsayılan))

DLQ 'daki iletinin ileti tanımlayıcısında (MQMD) belirtilen *PutApplTip* deęeri.

DESTQ (QueueName|* (varsayılan))

İletin gönderildięi ileti kuyruğunun adı.

DESTQM (QueueManagerAdi|* (varsayılan))

İletin gönderildięi ileti kuyruğunun kuyruk yöneticisinin adı.

FEEDBACK (Geribildirim|* (varsayılan))

MsgType deęeri 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))

İletinin 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))

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 kuyruğunun adı.

REPLYQM (*QueueManagerAdı* | * (varsayılan))

DLQ 'daki iletinin ileti tanımlayıcısında (MQMD) belirtildiği 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 (*QueueManagerAd* | & DESTQM | & REPLYQM | ' (varsayılan))

İletinin iletileceği kuyruğun kuyruk yöneticisi.

QueueManagerAd

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 kuralları çizelgesi kuralları

Ölü-mektup kuyruğu işleyici kuralları çizelgesinin sözdizimi, yapısı ve içeriği bu sözleşmeye uygun olmalıdır.

Kural çizelgesinin aşağıdaki kurallara uyması gerekir:

- Bir kural çizelgesinin en az bir kural içermesi gerekir.

- Anahtar sözcükler herhangi bir sırada olabilir.
- Bir anahtar sözcük herhangi bir kuralda yalnızca bir kez eklenebilir.
- Anahtar sözcükler büyük ve küçük harfe duyarlı değildir.
- Bir anahtar sözcük ve onun değiştirge değeri, en az bir boşluk ya da virgül ile diğer anahtar sözcüklerden ayrılmalıdır.
- Bir kuralın başında ya da sonunda ve anahtar sözcükler, noktalama işaretleri ve değerler arasında herhangi bir sayı boşluk olabilir.
- Her kural yeni bir satırda başlamalıdır.
- Windows sistemlerinde, çizelgedeki son kural satır başı/satır besleme karakteriyle bitmelidir. Bu işlemi, kuralın sonundaki Enter tuşuna basmanızı sağlayarak, çizelgenin son satırının boş bir çizgi olmasını sağlayarak bunu başarabilirsiniz.
- Taşınabilirlik nedenleriyle, çizginin önemli uzunluğunun 72 karakterden uzun olmamalıdır.
- Kuralın, sonraki satırın ilk boş olmayan karakterinden devam ettiğini belirtmek için, satırdaki son boş olmayan karakter olarak artı işaretini (+) kullanın. Kuralın sonraki satırın başlangıcından devam ettiğini belirtmek için, satırdaki son boş olmayan karakter olarak eksi işaretini (-) kullanın. Anahtar sözcükler ve değiştirgelerde devam karakterleri oluşabilir.

Örneğin:

```
APPLNAME('ABC+
D')
```

'ABCD' ile sonuçlanır ve

```
APPLNAME('ABC-
D')
```

'ABC D' nin sonuçları.

- Yıldız işareti (*) ile başlayan açıklama satırları, kural çizelgesinin herhangi bir yerinde oluşabilir.
- Boş satırlar yoksayılr.
- DLQ işleyici kuralları çizelgesindeki her giriş bir ya da daha çok anahtar sözcükten ve ilişkili değiştirgelerinden oluşur. Değiştirgelerin sözdizimi kurallarına uygun olması gerekir:
 - Her parametre değerinin en az bir anlamlı karakter içermesi gerekir. Tırnak içine alınmış değerlerdeki sınırlandırıcı tek tırnak işaretleri anlamlı olarak kabul edilmez. Örneğin, bu değiştirgeler 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 değiştirgeler önemli karakterler içermediği için geçersizdir:

```
FORMAT(' ')
FORMAT( )
FORMAT()
FORMAT
```

- Genel arama karakterleri desteklenir. Sondaki boşluk dışında tek bir karakter yerine soru işaretini (?) kullanabilirsiniz; sıfır ya da daha fazla bitişik karakter yerine yıldız (*) işaretini kullanabilirsiniz. Yıldız imi (*) ve soru işareti (?), **her zaman** parametre değerlerinde genel arama karakteri olarak yorumlanır.

- Şu anahtar sözcüklerin parametrelerinde genel arama karakterleri bulunamaz: ACTION, HEADER, RETRY, FWDQ, FWDQM ve PUTAUT.
- Parametre değerlerinde sondaki boşluklar ve DLQ üzerindeki iletilerde karşılık gelen alanlarda, joker karakter eşleşmeleri gerçekleştirilirken önemli değildir. Ancak, tek tırnak içine alınmış dizgiler içinde baştaki ve gömülü boşluklar, genel arama karakteri eşleşmeleri için önemlidir.
- Sayısal parametreler soru işareti (?) genel arama karakteri içeremez. Sayısal bir değiştirgenin bir parçası olarak değil, sayısal bir değiştirgenin tamamı yerine yıldız (*) işaretini kullanabilirsiniz. Örneğin, bunlar geçerli sayısal değiştirgelerdir:

MSGTYPE (2)	Yalnızca yanıt iletileri seçilebilir
MSGTYPE (*)	Herhangi bir ileti tipi seçilebilir
MSGTYPE ('*')	Herhangi bir ileti tipi seçilebilir

Ancak, sayısal bir parametrenin parçası olarak bir yıldız işareti (*) içerdiğinden MSGTYPE ('*2*') geçerli değildir.

- Sayısal parametreler 0-999 999 999 aralığında olmalı. Parametre değeri bu aralıktaki değerse, anahtar sözcüğün ilişkili olduğu alanda geçerli olmasa da kabul edilir. Sayısal parametreler için simgesel adlar kullanabilirsiniz.
- Bir dizilim değeri, anahtar sözcüğün ilgili olduğu MQDLH ya da MQMD 'deki alandan kısaysa, değer, alanın uzunluğuna kadar boşluklarla doldurulur. Yıldız imleri hariç olmak üzere değer, alandan daha uzunsa bir hataya neden olur. Örneğin, bunlar 8 karakterlik bir alana ilişkin geçerli dizgi değerleridir:

'ABCDEFGH'	8 karakter
'A*C*E*G*I'	Yıldız imleri dışında 5 karakter
'*A*C*E*G*I*K*M*O*'	Yıldız imleri dışında 8 karakter

- Boşluk, küçük harf karakterleri ya da nokta (.), eğik çizgi (?), alt çizgi (_) ve yüzde imi (%) içeren dizgileri tek tırnak işareti içine alın. Tek tırnak içine alınmayan küçük harfli karakterler büyük harfe çevrilir. Dizgide bir tırnak işareti varsa, hem başlangıç hem de sonunu belirtmek için iki tek tırnak işareti kullanın. Dizilimin uzunluğu hesaplanırken, çift tırnak işaretlerinin her bir oluşumu tek bir karakter olarak sayılır.

DLQ kuralları çizelgesinin nasıl işlendiği

Ölü-harfli kuyruk işleyicisi, kuralın DLQ 'da bir iletiyle eşleştiği bir kural için kural çizelgesini arar.

Arama, tablodaki ilk kuralla başlar ve tablo aracılığıyla sırayla devam eder. DLQ işleyicisi, eşleştirme örüntüyle bir kural bulunduğunda, bu kuraldan işlem alır. DLQ işleyicisi, kural uygulandığında, bir kuralla ilişkin yeniden deneme sayısını 1 artırır. İlk deneme başarısız olursa, DLQ işleyicisi RETRY anahtar sözcüğünde belirtilen sayıyla eşleşmeye çalışılincaya kadar yeniden dener. Tüm denemeler başarısız olursa, DLQ işleyicisi çizelgede sonraki eşleşen kuralı arar.

Bu işlem, bir işlem başarılı oluncaya kadar sonraki eşleştirme kuralları için yinelenir. Her eşleştirme kuralı RETRY anahtar sözcüğünde belirtilen sayıda kez denendiğinde ve tüm girişimler başarısız olduğunda, ACTION (IGNORE) varsayılır. Eşleşen bir kural bulunmazsa, ACTION (IGNORE) da varsayılır.

Not:

1. Eşleşen kural örüntüleri yalnızca, bir MQDLH ile başlayan DLQ 'daki iletiler için aranır. MQDLH ile başlamayan iletiler düzenli aralıklarla hata olarak bildirilir ve DLQ 'da süresiz olarak kalır.
2. Bir kuralın yalnızca bir eylemden oluşabilmesi için, varsayılan olarak tüm örüntü anahtar sözcüklerine izin verilebilir. Ancak, kuyrukta MQDLHs içeren ve çizelgeye ilişkin diğer kurallara uygun olarak işlenmemiş olan tüm iletilere yalnızca işlem kuralları uygulanmaktadır.
3. Kural çizelgesi, DLQ işleyicisi başlatıldığında doğrulanır 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 her zaman, iletileri MQPMO_PASS_ALL_CONTEXT ileti seçeneği ile diğer kuyruklara koyar.
5. Kural çizelgesi, geçerlilik denetimi sırasında yinelenen hataların oluşturulmasını ortadan kaldırmak için tasarlandığı için, kural çizelgesinde art arda sözdizimi hataları tanınmayabilir.
6. DLQ işleyicisi, DLQ ' yı MQOO_INPUT_AS_Q_DEF seçeneğiyle açar.
7. Aynı kural çizelgesini kullanarak, DLQ işleyicisinin birden çok eşgörünümü aynı kuyruğa koşturarak koştuzamanlı olarak çalışabilir. Ancak, bir DLQ ile DLQ işleyicisi arasında bire bir ilişki olması daha olağan bir durum.

İlgili kavramlar

Ölü-harf kuyrukları

İlgili görevler

Teslim edilmeyen iletiler sorun giderme

Tüm DLQ iletilerinin işlenmesini sağlama

İleti kuyruğu işleyicisi, DLQ üzerinde görülen ancak kaldırılmamış olan tüm iletilerin kaydını tutar.

DLQ işleyicisini, DLQ ' dan gelen iletilerin küçük bir alt kümesini çıkarmak için süzgeç olarak kullanırsanız, DLQ işleyicisi, DLQ üzerinde işlem yapmadığı iletilerin kaydını tutmak zorunda kalabilir. DLQ işleyicisi, DLQ 'ya gelen yeni iletilerin, ilk giren ilk çıkış (FIFO) olarak tanımlansa da, DLQ' ya yeni iletilerin görülmesini garanti edemez. Kuyruk boş değilse, DLQ, tüm iletileri denetlemek için düzenli aralıklarla yeniden taranır.

Bu nedenlerle, DLQ 'nun olabildiğince az sayıda ileti içerdiğinden emin olmaya çalışın; atılmayan ya da başka kuyruklara iletilen iletiler (herhangi bir nedenle) kuyrukta birikmesine izin verilirse, DLQ işleyicisinin iş yükü artar ve DLQ' nun kendisi doldurabilir.

DLQ işleyicinin DLQ ' yı boşaltmasını sağlamak için belirli ölçümler yapabilirsiniz. Örneğin, DLQ ' da ileti bırakan ACTION (IGNORE) kullanmayın. (Tablodaki diğer kurallar tarafından açık bir şekilde ele alınmayan iletiler için ACTION (IGNORE) varsayılanını unutmayın.) Bunun yerine, başka bir şekilde yoksaymak istediğiniz iletiler için, iletileri başka bir kuyruğa taşıyan bir işlem kullanın. Örnek:

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

Benzer şekilde, tabloda daha önceki kurallar tarafından ele alınmamış iletileri işlemek için bir catchall tablolarında son kuralı yapın. Örneğin, tabloda son kural şöyle bir şey olabilir:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Bu ileti, çizelgedeki son kurala kadar, el ile işlenebileceği REALLY . DEAD . QUEUEkuyruğuna iletilecek iletiler iletir. Böyle bir kuralınız yoksa, iletilerin DLQ ' da süresiz olarak kalabileceğini kabul edin.

Bir DLQ işleyicisi kural çizelgesi örneği

runmqdlq komutu için, tek bir denetim-veri girişi ve birkaç kural içeren, örnek bir kuyruk kuralları çizelgesi.

```
*****
*   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

[Ölü-harf kuyrukları](#)

İlgili görevler

[Teslim edilmeyen iletiler sorun giderme](#)

İlgili başvurular

[runmqdlq \(çalıştırma-harfli kuyruk işleyicisi çalıştır\)](#)

Yönetimle ilgili konularla çalışma

Yönetimle ilgili konuları yönetmek için MQSC komutlarını kullanın.

Bu komutlarla ilgili ayrıntılı bilgi için [MQSC komutları](#) başlıklı konuya bakın.

İlgili kavramlar

[Yönetimle ilgili konu nesneleri](#)

[“Yönetimle ilgili konu tanımlanması” sayfa 162](#)

Bir yönetim konusu yaratmak için **DEFINE TOPIC** MQSC komutunu kullanın. Yönetimle ilgili bir konu tanımlarken, isteğe bağlı olarak her konu özniteliğini ayarlayabilirsiniz.

[“Yönetimle ilgili konu nesnesi özniteliklerinin görüntülenmesi” sayfa 162](#)

Bir denetim konusu nesnesini görüntülemek için **DISPLAY TOPIC** MQSC komutunu kullanın.

[“Denetim konusu özniteliklerinin değiştirilmesi” sayfa 163](#)

You can change topic attributes in two ways, using either the **ALTER TOPIC** command or the **DEFINE TOPIC** command with the **REPLACE** attribute.

[“Denetim konusu tanımlamasının kopyalanması” sayfa 164](#)

DEFINE komutundaki **LIKE** özniteliğini kullanarak bir konu tanımını kopyalayabilirsiniz.

[“Denetim konusu tanımlamasının silinmesi” sayfa 164](#)

Bir yönetim konusunu silmek için **DELETE TOPIC** MQSC komutunu kullanabilirsiniz.

Yönetimle ilgili konu tanımlanması

Bir yönetim konusu yaratmak için **DEFINE TOPIC** MQSC komutunu kullanın. Yönetimle ilgili bir konu tanımlarken, isteğe bağlı olarak her konu özniteliğini ayarlayabilirsiniz.

Belirtik olarak ayarlanmayan konunun herhangi bir özniteliği, sistem kuruluşu kurulduğunda oluşturulan varsayılan yönetim konusundan (**SYSTEM.DEFAULT.TOPIC**) devralınır.

Örneğin, izleyen **DEFINE TOPIC** komutu, aşağıdaki özelliklere sahip **ORANGE . TOPIC** adlı bir konuyu tanımlar:

- **TURUNCU** konu dizgisine çözülüyor. Konu dizgilerinin nasıl kullanılabileceğiyle ilgili bilgi için [Konu dizgilerinin birleştirilmesi](#) başlıklı konuya bakın.
- **ASPARENT** olarak ayarlanan herhangi bir öznitelik, bu konunun üst konularının tanımladığı şekilde özniteliği kullanır. Bu işlem, kök konu **SYSTEM.BASE.TOPIC** bulunduğu sürece konu ağacını yinelemeli. Ek bilgi için [Konu ağaçları](#) başlıklı konuya bakın.

```
DEFINE TOPIC (ORANGE.TOPIC) +
TOPICSTR (ORANGE) +
DEFPRTY(ASPARENT) +
NPMSGDLV (ASPARENT)
```

Not:

- Konu dizgisinin değeri dışında, gösterilen tüm öznitelik değerleri varsayılan değerlerdir. Burada yalnızca bir şekil olarak gösterilir. Varsayılan değerlerin istediğiniz ya da değiştirilmediğinden eminseniz, bunları atlayabilirsiniz. Ayrıca bkz. [“Yönetimle ilgili konu nesnesi özniteliklerinin görüntülenmesi” sayfa 162.](#)
- Aynı kuyruk yöneticisiyle ilgili olarak **ORANGE.TOPIC** adlı bir yönetim konuunuz varsa, bu komut başarısız olur. Bir konunun var olan tanımlamasının üzerine yazmak istiyorsanız **REPLACE** özniteliğini kullanın, ancak [“Denetim konusu özniteliklerinin değiştirilmesi” sayfa 163](#) konusuna da bakın.

İlgili başvurular

[KONUYU TANIMLA](#)

Yönetimle ilgili konu nesnesi özniteliklerinin görüntülenmesi

Bir denetim konusu nesnesini görüntülemek için **DISPLAY TOPIC** MQSC komutunu kullanın.

Tüm konuları görüntülemek için şunu kullanın:

```
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 aşağıdaki gibi görüntüler:

```
AMQ8633: Display topic details.  
TOPIC(ORANGE.TOPIC) TYPE(LOCAL)  
TOPICSTR(ORANGE) DEFPRTY(ASPARENT)  
NPMSGDLV(ASPARENT)
```

Yürütme sırasında kullanılmakta olan ASPARENT değerlerini görüntülemek için **DISPLAY TPSTATUS** komutunu kullanın. Örneğin, şunları kullanın:

```
DISPLAY TPSTATUS(ORANGE) DEFPRTY NPMSGDLV
```

Komut, aşağıdaki ayrıntıları görüntüler:

```
AMQ8754: Display topic status details.  
TOPICSTR(ORANGE) DEFPRTY(0)  
NPMSGDLV(ALLAVAIL)
```

Bir denetim konusu tanımlarken, belirttiğiniz varsayılan denetim konusundan (SYSTEM.DEFAULT.TOPIC. Bu varsayılan özniteliklerin ne olduğunu görmek için aşağıdaki komutu kullanın:

```
DISPLAY TOPIC (SYSTEM.DEFAULT.TOPIC)
```

İlgili başvurular

[KONUYU GÖRÜNTÜLE](#)

[TANITIM](#)

Denetim konusu özniteliklerinin değiştirilmesi

You can change topic attributes in two ways, using either the **ALTER TOPIC** command or the **DEFINE TOPIC** command with the **REPLACE** attribute.

Örneğin, ORANGE.TOPIC, 5 olması için aşağıdaki komutlardan birini kullanın.

- **ALTER** komutunu kullanarak:

```
ALTER TOPIC(ORANGE.TOPIC) DEFPRTY(5)
```

Bu komut tek bir özniteliği değiştirir; bu konuya, bu konuya ilişkin varsayılan öncelik önceliği 5 'e teslim edilir; diğer tüm öznitelikler aynı kalır.

- **DEFINE** komutunu kullanarak:

```
DEFINE TOPIC(ORANGE.TOPIC) DEFPRTY(5) REPLACE
```

Bu komut, bu konuya teslim edilen iletilerin varsayılan önceliğini değiştirir. Diğer tüm özniteliklere varsayılan değerleri verilir.

Bu konuya gönderilen iletilerin önceliğini değiştiriyorsanız, var olan iletiler bundan etkilenmez. Ancak, yeni bir ileti, yayınlama uygulaması tarafından sağlanmıyorsa, belirtilen önceliği kullanın.

İlgili başvurular

[KONUYU DEĞİŞTİR](#)

KONUYU GÖRÜNTÜLE

Denetim konusu tanımlamasının kopyalanması

DEFINE komutundaki **LIKE** özneliğini kullanarak bir konu tanımını kopyalayabilirsiniz.

Örneğin:

```
DEFINE TOPIC (MAGENTA.TOPIC) +  
LIKE (ORANGE.TOPIC)
```

Bu komut bir konu yaratır, **MAGENTA.TOPIC**, özgün konuyla aynı özneliklerle **ORANGE.TOPIC**, sistemin varsayılan denetim konularından değil. Konuyu oluşturduğunuz sırada tam olarak girildiği şekilde kopyalanacak konunun adını girin. Ad küçük harfli karakterler içeriyorsa, adı tek tırnak işareti içine alın.

Bir konu tanımlamasını kopyalamak için **DEFINE** komutunun bu formunu da kullanabilirsiniz, ancak özgün özneliklerin özneliklerinde değişiklik yapabilirsiniz. Örneğin:

```
DEFINE TOPIC (BLUE.TOPIC) +  
TOPICSTR (BLUE) +  
LIKE (ORANGE.TOPIC)
```

You can also copy the attributes of the topic **BLUE.TOPIC** to the topic **GREEN.TOPIC** and specify that when publications cannot be delivered to their correct subscriber queue they are not placed onto the dead-letter queue. Örneğin:

```
DEFINE TOPIC (GREEN.TOPIC) +  
TOPICSTR (GREEN) +  
LIKE (BLUE.TOPIC) +  
USEDLQ (NO)
```

İlgili başvurular

KONUYU TANIMLA

Denetim konusu tanımlamasının silinmesi

Bir yönetim konusunu silmek için **DELETE TOPIC MQSC** komutunu kullanabilirsiniz.

Örneğin:

```
DELETE TOPIC (ORANGE.TOPIC)
```

Artık uygulamalar, **ORANGE.TOPIC**. Konunun açık olan yayınlama uygulamaları, çözümlenen konu dizisini yayınlamaya devam edebilir. Bu konuda önceden yapılmış olan abonelikler, konu silindikten sonra yayınları almaya devam eder.

Bu konu nesnesine gönderme yapan ancak çözülmüş konu dizgisini kullanan uygulamalar, bu örnekte 'TURUNCU' olarak gösterilen 'TURUNCU' başlıklı konuyu kullanmaya devam eder. Bu durumda, özellikler, konu ağacındaki bir konu nesnesindeki özellikleri devralır. Ek bilgi için [Konu ağaçları](#) başlıklı konuya bakın.

İlgili başvurular

KONUYU SIL

Aboneliklerle çalışma

Abonelikleri yönetmek için **MQSC** komutlarını kullanın.

Abonelikler, **SUBTYPE** öznelisinde tanımlanan üç tipten biri olabilir:

ADMIN

Bir kullanıcı tarafından yönetimsel olarak tanımlanır.

YETKİLİ SUNUCU

Kuyruk yöneticileri arasında yayınları yönlendirmek için dahili olarak oluşturulmuş bir abonelik.

API

Örneğin, MQI MQSUB çağrısını kullanarak programsal olarak yaratıldı.

Bu komutlarla ilgili ayrıntılı bilgi için [MQSC komutları](#) başlıklı konuya bakın.

İlgili kavramlar

[“Yönetimle ilgili abonelik tanımlanması” sayfa 165](#)

Denetim aboneliği yaratmak için **DEFINE SUB** MQSC komutunu kullanın. Varsayılan yerel abonelik tanımlanmasında tanımlı olan varsayılan değeri de kullanabilirsiniz. Ya da, abonelik özelliklerini varsayılan yerel abonelikten (SYSTEM.DEFAULT.SUB , sistem kurulduğunda yaratılmış olur.

[“Aboneliklerin özneliklerini görüntüleme” sayfa 166](#)

Kuyruk yöneticisi tarafından bilinen herhangi bir aboneliğin yapılandırılmış özneliklerini görüntülemek için **DISPLAY SUB** komutunu kullanabilirsiniz.

[“Yerel abonelik özneliklerinin değiştirilmesi” sayfa 167](#)

You can change subscription attributes in two ways, using either the **ALTER SUB** command or the **DEFINE SUB** command with the **REPLACE** attribute.

[“Yerel abonelik tanımlanmasının kopyalanması” sayfa 167](#)

DEFINE komutundaki **LIKE** özneliğini kullanarak bir abonelik tanımlanmasını kopyalayabilirsiniz.

[“Yerel aboneliği silme” sayfa 167](#)

You can use the MQSC command **DELETE SUB** to delete a local subscription.

Yönetimle ilgili abonelik tanımlanması

Denetim aboneliği yaratmak için **DEFINE SUB** MQSC komutunu kullanın. Varsayılan yerel abonelik tanımlanmasında tanımlı olan varsayılan değeri de kullanabilirsiniz. Ya da, abonelik özelliklerini varsayılan yerel abonelikten (SYSTEM.DEFAULT.SUB , sistem kurulduğunda yaratılmış olur.

Örneğin, aşağıdaki **DEFINE SUB** komutu, bu özelliklere sahip TURUNCU adlı bir aboneliği tanımlar:

- Sürekli abonelik, kuyruk yöneticisini yeniden başlatma, sınırsız süre bitimi ile devam eder anlamına gelir.
- TURUNCU konu dizisinde, yayınlama uygulamalarının ayarlandığı şekilde, ileti öncelikleri ile 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 (ASPUB)
```

Not:

- Abonelik ve konu dizisi adının eşleşmesi gerekmez.
- Hedef ve konu dizisinin değerleri dışında, gösterilen tüm öznelik değerleri varsayılan değerlerdir. Burada yalnızca bir şekil olarak gösterilir. Varsayılan değerlerin istediğiniz ya da değiştirilmediğinden eminseniz, bunları atlayabilirsiniz. Ayrıca bkz. [“Aboneliklerin özneliklerini görüntüleme” sayfa 166](#).
- Aynı kuyruk yöneticisiyle ilgili olarak TURUNCU ile yerel bir aboneliğiniz varsa, bu komut başarısız olur. Bir kuyruğun var olan tanımlanmasının üzerine yazmak istiyorsanız **REPLACE** özneliğini kullanın, ancak aynı zamanda bkz. [“Yerel abonelik özneliklerinin değiştirilmesi” sayfa 167](#).
- SUBQ KUYRUĞU yoksa, bu komut başarısız olur.

İlgili başvurular

[ALT](#)

Aboneliklerin özniteliklerini görüntüleme

Kuyruk yöneticisi tarafından bilinen herhangi bir aboneliğin yapılandırılmış özniteliklerini görüntülemek için **DISPLAY SUB** komutunu kullanabilirsiniz.

Örneğin, şunları kullanın:

```
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 üzerinde çalıştığı çözümlenen konu dizgisidir. Bir abonelik, bir konu nesnesini kullanmak için tanımlandığında, o nesneden konu dizgisi, abonelik yapılırken sağlanan konu dizgisine önek olarak kullanılır. SUBID, abonelik yaratıldığında kuyruk yöneticisi tarafından atanan benzersiz bir tanıttıcıdır. Bazı abonelik adları uzun ya da farklı bir karakter kümelerinde kullanışsız olabileceği için, bu özellik görüntülenmek üzere yararlı bir özniteliğe sahip olabilir.

Aboneliklerin görüntülenmesine ilişkin diğer bir yöntem SUBID ' yi kullanmandır:

```
DISPLAY SUB +  
SUBID(414D51204141412020202020202020EE921E4E20002A03) +  
TOPICSTR +  
DURABLE
```

Bu komut, aşağıdakiyle aynı çıkışa 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 bir yetkili sunucu ya da TÜRÜ **SUBTYPE** belirtin.

Yürütme ortamı özniteliklerini görüntülemek için DISPLAY SBSTATUS komutunu kullanabilirsiniz. Örneğin, şu komutu kullanın:

```
DISPLAY SBSTATUS(ORANGE) NUMMSGs
```

Aşağıdaki çıkış görüntülenir:

```
AMQ8099: IBM MQ subscription status inquired.  
SUB(ORANGE)  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
NUMMSGs(0)
```

Bir yönetici aboneliği tanımladığınızda, bu abonelik varsayılan abonelikten belirtik olarak belirtmediğiniz herhangi bir özniteliği alır; bu, SYSTEM.DEFAULT.SUBadlı bir aboneliktenexplicitlyolarak adlandırılır. Bu varsayılan özniteliklerin ne olduğunu görmek için aşağıdaki komutu kullanın:

```
DISPLAY SUB (SYSTEM.DEFAULT.SUB)
```

İlgili başvurular

[GÖRÜNTÜLE](#)

Yerel abonelik özniteliklerinin değiştirilmesi

You can change subscription attributes in two ways, using either the **ALTER SUB** command or the **DEFINE SUB** command with the **REPLACE** attribute.

Örneğin, TURUNCU olarak adlandırılan bir aboneliğe teslim edilen iletilerin önceliğini 5 olarak değiştirmek istiyorsanız, aşağıdaki komutlardan birini kullanın:

- **ALTER** komutunu kullanarak:

```
ALTER SUB(ORANGE) PUBPRTY(5)
```

Bu komut tek bir özniteliği değiştirir, bu aboneliğe 5 'e teslim edilen iletilerin önceliğini gösterir; diğer tüm öznitelikler de aynı kalır.

- **DEFINE** komutunu kullanarak:

```
DEFINE SUB(ORANGE) PUBPRTY(5) REPLACE
```

Bu komut, yalnızca bu aboneliğe teslim edilen iletilerin önceliğini değil, diğer tüm özniteliklerin de varsayılan değerlerini değiştirmesini sağlar.

Bu aboneliğe gönderilen iletilerin önceliğini değiştiriyorsanız, var olan iletiler etkilenmez. Ancak, belirlenen önceliğe ilişkin yeni iletiler de vardır.

İlgili başvurular

[ALTER SUB](#)

[ALT](#)

Yerel abonelik tanımlamasının kopyalanması

DEFINE komutundaki **LIKE** özniteliğini kullanarak bir abonelik tanımlamasını kopyalayabilirsiniz.

Örneğin:

```
DEFINE SUB(BLUE) +  
LIKE(ORANGE)
```

You can also copy the attributes of the sub REAL to the sub THIRD.SUB, and specify that the correlID of delivered publications is THIRD, rather than the publishers correlID. Örneğin:

```
DEFINE SUB(THIRD.SUB) +  
LIKE(BLUE) +  
DESTCORL(ORANGE)
```

İlgili başvurular

[ALT](#)

Yerel aboneliği silme

You can use the MQSC command **DELETE SUB** to delete a local subscription.

```
DELETE SUB(ORANGE)
```

SUBID seçeneğini kullanarak da bir aboneliği silebilirsiniz:

```
DELETE SUB SUBID(414D51204141412020202020202020EE921E4E20002A03)
```

İlgili başvurular

[SUB SIL](#)

Abonelikte iletilerin denetlenmesi

Bir abonelik tanımlandığında, bu abonelik bir kuyrukla ilişkilendirilir. Bu abonelikte eşleşen yayınlanan iletiler bu kuyruğa konabiliyor.

Bu görev hakkında

Aşağıdaki **runmqsc** komutlarının yalnızca ileti alan abonelikleri gösterdiğine dikkat edin.

Bir aboneliğin şu anda kuyruğa alınmış olan iletileri denetlemek için aşağıdaki adımları izleyin:

Yordam

1. Bir abonelik tipi `DISPLAY SBSTATUS(sub_name) NUMMSG` için kuyruğa alınan iletileri denetlemek için bkz. "[Aboneliklerin özniteliklerini görüntüleme](#)" sayfa 166.
2. **NUMMSG** değeri sıfırdan büyükse, abonelikle ilişkili kuyruğu `DISPLAY SUB(sub_name) DEST` yazarak belirleyin.
3. Using the name of the queue returned you can view the messages by following the technique described in "[Örnek program ile kuyruklara göz atma](#)" sayfa 133.

İlgili başvurular

[SBSTATUS GÖRÜNTÜLE](#)

Hizmetlerle çalışma

Hizmet nesneleri, ek işlemlerin bir kuyruk yöneticisinin parçası olarak yönetilebileceği bir araçtır. Hizmetler ile, kuyruk yöneticisi başladığında ve sona erdirildiğinde başlatılmış ve durdurulmuş programlar tanımlayabilirsiniz. IBM MQ services are always started under the user ID of the user who started the queue manager.

Yeni bir IBM MQ hizmet tanımı tanımlamak için, MQSC komutu `DEFINE SERVICE` komutunu kullanın.

Hizmet nesneleri aşağıdaki tiplerden biri olabilir:

Sunucu

Sunucu, **SERVTYPE** parametresine sahip bir hizmet nesnesidir ve `SUNUCU` olarak belirtilmiştir. Sunucu hizmeti nesnesi, belirlenen bir kuyruk yöneticisi başlatıldığında yürütülen bir programın tanımlamasıdır. Sunucu hizmeti nesneleri, genellikle uzun bir süre boyunca çalıştırılan programları tanımlar. Örneğin, bir sunucu hizmeti nesnesi, bir tetikleme izleme işlemi yürütmek için kullanılabilir (örneğin, **runmqtrm**).

Bir sunucu hizmeti nesnesinin yalnızca bir eşgörünümü koşut zamanlı olarak çalışabilir. The status of running server service objects can be monitored using the MQSC command, **DISPLAY SVSTATUS**.

Komut

Komut, **SERVTYPE** parametresi `COMMAND` olarak belirtilen bir hizmet nesnesidir. Command service objects are similar to server service objects, however multiple instances of a command service object can run concurrently, and their status cannot be monitored using the MQSC command **DISPLAY SVSTATUS**.

MQSC komutu **STOP SERVICE** yürütülürse, program durdurulmadan önce, **START SERVICE** MQSC komutu tarafından başlatılmış olan programın etkin olup olmadığını saptamak için denetim yapılmaz.

İlgili başvurular

[SVSTATUS GÖRÜNTÜLE](#)

[Hizmeti başlat](#)

[Hizmeti durdur](#)

Hizmet Nesnesi Tanımlama

You define a service object with the MQSC command **DEFINE SERVICE**.

Tanımlamak için gereksinim dumanız gereken öznitelikler şunlardır:

SERVTYPE

Hizmet nesnesinin tipini tanımlar. Olası değerler aşağıdaki gibidir:

SERVER

Bir sunucu hizmeti nesnesi.

Bir kerede tek bir sunucu hizmeti nesnesinin tek bir eşgörünümü yürütülebilir. The status of server service objects can be monitored using the MQSC command, **DISPLAY SVSTATUS**.

Komut

Bir komut hizmeti nesnesi.

Bir komut hizmeti nesnesinin birden çok eşgörünümü koşutzamanlı olarak yürütülebiliyor. Bir komut hizmeti nesnelerinin durumu izlenemez.

STARTCMD

Hizmeti başlatmak için yürütülen program. Programın tam olarak nitelenmiş bir yolu belirtilmelidir.

BAŞLATMA

Başlangıç programına geçirilen bağımsız değişkenler.

STDERR

Hizmet programının standart hatasının (stderr) yeniden yönlendirilmesi gereken bir kütüğün yolunu belirtir.

KALDIR

Hizmet programının standart çıkışının (stdout) yeniden yönlendirilmesi gereken bir kütüğün yolunu belirtir.

STOPCMD

Hizmeti durdurmak için yürütülen program. Programın tam olarak nitelenmiş bir yolu belirtilmelidir.

STOPARG

Durdurma programına bağımsız değişkenler geçirildi.

CONTROL

Hizmetin nasıl başlatılacağını ve durdurulduğunu belirtir:

El ile

Hizmet otomatik olarak başlatılamaz ya da otomatik olarak durdurulmaz. Bu, **START SERVICE** ve **STOP SERVICE** komutlarının kullanımıyla denetlenir. Bu varsayılan değerdir.

MMGR

Kuyruk yöneticisi başlatılıp durdurulmuş olduğundan, tanımlanmakta olan hizmet başlatılıp durdurulur ve durdurulur.

BAŞLATMA

Hizmet, kuyruk yöneticisi başlatıldığı gibi başlatılacak, ancak kuyruk yöneticisi durdurulduğunda durdurulacak şekilde istenmez.

İlgili kavramlar

[“Hizmetlerin yönetilmesi” sayfa 170](#)

CONTROL parametresini kullanarak, bir hizmet nesnesinin bir eşgörünümü başlatılır ve kuyruk yöneticisi tarafından otomatik olarak durdurulabilir ya da durdurularak MQSC komutları **START SERVICE** ve **STOP SERVICE** kullanılarak başlatılabilir ve durdurulabilir.

İlgili başvurular

[HİZMET](#)

[SVSTATUS GÖRÜNTÜLE](#)

[Hizmeti başlat](#)

[Hizmeti durdur](#)

Hizmetlerin yönetilmesi

CONTROL parametresini kullanarak, bir hizmet nesnesinin bir eşgörünümü başlatılır ve kuyruk yöneticisi tarafından otomatik olarak durdurulabilir ya da durdurularak MQSC komutları **START SERVICE** ve **STOP SERVICE** kullanılarak başlatılabilir ve durdurulabilir.

Bir hizmet nesnesinin bir eşgörünümü başlatıldığında, kuyruk yöneticisi hata günlüğüne, hizmet nesnesinin adını ve başlatılan işlemin işlem tanıtıcısını içeren bir ileti yazılır. Aşağıda, bir sunucu hizmeti nesnesi için bir günlük girişi örneği verilmiştir:

```
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.
```

Bir komut hizmeti nesnesi için örnek bir günlük girişi aşağıdaki gibidir:

```
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 sona erdirmiş işleminin işlem tanıtıcısını içeren bir ileti yazılır. Bir sunucu hizmeti nesnesi için örnek günlük girişi aşağıdaki gibidir:

```
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 başvurular

[“Ek ortam değişkenleri” sayfa 171](#)

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.

[Hizmeti durdur](#)

[Hizmeti başlat](#)

Ek ortam deęişkenleri

Bir hizmet başlatıldığında, hizmet işleminin başlatıldığı ortam kuyruk yöneticisinin ortamından devralınır. Tanımlamak istediğiniz deęişkenleri `service.env` ortamı geçersiz kılma dosyalarından birine ekleyerek hizmet sürecinin ortamında ayarlanacak ek ortam deęişkenleri tanımlanabilir.

Ortam deęişkenlerini ekleyebileceğiniz dosyalar

Ortam deęişkenlerini ekleyebileceğiniz iki olası dosya vardır:

Makine kapsamı `service.env` dosyası

Bu dosyanın yeri:

- **Linux** **AIX** AIX and Linux sistemlerinde `/var/mqm`.
- **Windows** Windows sistemlerinde kuruluş sırasında seçilen veri dizini.

Kuyruk yöneticisi kapsamı `service.env` dosyası

Bu dosya, kuyruk yöneticisi veri dizininde bulunur. Örneğin, QMNAME adlı bir kuyruk yöneticisi için ortam geçersiz kılma dosyasının yeri:

- **Linux** **AIX** AIX and Linux sistemlerinde, `/var/mqm/qmgrs/QMNAME/service.env`
- **Windows** Windows sistemlerinde,
`C:\ProgramData\IBM\MQ\qmgrs\QMNAME\service.env`

Her iki dosya da (varsa) işlenir ve kuyruk yöneticisi kapsam dosyasındaki tanımlamalar, makine kapsamı dosyasındaki tanımlamalardan önceliklidir.

`service.env` içinde belirtilebilir ortam deęişkenleri.

`service.env` içinde herhangi bir ortam deęişkeni belirtilebilir. Örneğin, IBM MQ hizmeti bir dizi komut çalıştırırsa, `service.env` dosyasında `PATH` kullanıcı deęişkenini ayarlamak yararlı olabilir. Deęişkeni ayarladığınız deęerler ortam deęişkenleri 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 Telemetri) kendi sınıf yollarını ayarlar. `service.env` içinde tanımlanan `CLASSPATH` buna eklenir.

Dosyada tanımlı deęişkenlerin biçimi olan `service.env`, ad ve deęer deęişkeni çiftlerinin bir listesidir. Her deęişken yeni bir satırda tanımlanmalıdır ve her deęişken, beyaz alan da içinde olmak üzere, belirtik olarak tanımlandığı şekilde alınır.

`service.env` örneęi

```
#####  
##  
## <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 başvurular

“Hizmet tanımlarında değiştirilebilir eklemeler” sayfa 172

Bir hizmet nesnesi tanımlamasında, simgelerin yerine konması mümkündür. Hizmet programı yürütüldüğünde yerine koyulan simgeler, otomatik olarak genişletilmiş metniyle değiştirilir. Yerine koyma simgeleri, aşağıdaki ortak belirteçler listesinden ya da dosyada tanımlı olan değişkenlerden (service.env) alınabilir.

Hizmet tanımlarında değiştirilebilir eklemeler

Bir hizmet nesnesi tanımlamasında, simgelerin yerine konması mümkündür. Hizmet programı yürütüldüğünde yerine koyulan simgeler, otomatik olarak genişletilmiş metniyle değiştirilir. Yerine koyma simgeleri, aşağıdaki ortak belirteçler listesinden ya da dosyada tanımlı olan değişkenlerden (service.env) alınabilir.



Aşağıda, bir hizmet nesnesinin tanımındaki simgeleri yerine koymak için kullanılacak ortak simgeler yer alıyor:

MQ_INSTALL_PATH

IBM MQ ' un kurulu olduğu yer.

MQ_DATA_PATH

IBM MQ veri dizininin konumu:

-  AIX and Linux sistemlerinde, IBM MQ veri dizini yeri /var/mqm/olur.
-  Windows sistemlerinde, IBM MQ veri dizininin yeri, IBM MQ kurulumu sırasında seçilen veri dizidir.

QMNAME

Yürürlükteki kuyruk yöneticisi adı.

MQ_HIZMET_ADI

Hizmetin adı

MQ_SERVER_PID

Bu simge yalnızca **STOPARG** ve **STOPCMD** bağımsız değişkenleri tarafından kullanılabilir.

Sunucu hizmeti nesnelere ilişkin bu simge, **STARTCMD** ve **STARTARG** bağımsız değişkenlerinin başlattığı işlemin işlem tanıtıcısıyla değiştirilir. Ters durumda, bu simge 0 ile değiştirilir.

MQ_Q_MGR_DATA_PATH

Kuyruk yöneticisi verileri dizininin yeri.

MQ_Q_MGR_DATA_ADI

Kuyruk yöneticisinin dönüştürülen adı. Ad dönüşümlerine ilişkin ek bilgi için [IBM MQ dosya adlarını anlamabaşlıklı konuya](#) bakın.

Değiştirilebilir ekler kullanmak için, simgeyi + karakterleri içinde **STARTCMD**, **STARTARG**, **STOPCMD**, **STOPARG**, **STDOUT** ya da **STDERR** dizgilerinin herhangi birine yerleştirin. Buna örnek olarak bkz. “Hizmet nesnelere ilişkin kullanımlarına ilişkin örnekler” sayfa 172.

Hizmet nesnelere ilişkin kullanımlarına ilişkin örnekler

Bu bölümdeki hizmetler, aksi belirtilenler dışında, UNIX stil yolu ayırıcı karakterleriyle yazılır.

Sunucu hizmeti nesnesinin kullanılması

Bu örnekte, bir tetikleme izleyiciyi başlatmak için sunucu hizmeti nesnesinin nasıl tanımlanacak, kullanılacağı ve değiştirileceği gösterilmektedir.

1. A server service object is defined, using the **DEFINE SERVICE** MQSC command:

```
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 is a sample program provided with IBM MQ which requests the queue manager to break all connections for the process ID. amqsstop , PCF komutlarını oluşturur, bu nedenle komut sunucusu çalışır durumda olmalıdır.

+MQ_SERVER_PID+ , durdurma programına geçirilen süreç tanıtıcısını gösteren bir simgedir.

Ortak belirteçlerin listesi için bkz. “Hizmet tanımlarında değiştirilebilir eklemeler” sayfa 172 .

2. Kuyruk yöneticisi bir sonraki başlatılırsa, sunucu hizmeti nesnesinin bir eşgörünümü yürütülür. Ancak, sunucu hizmeti nesnesinin bir eşgörünümünü hemen **START SERVICE** MQSC komutu ile başlatacağız:

```
START SERVICE(S1)
```

3. The status of the server service process is displayed, using the **DISPLAY SVSTATUS** MQSC command:

```
DISPLAY SVSTATUS(S1)
```

4. Bu örnek şimdi sunucu hizmeti nesnesinin nasıl değiştirileceğini ve sunucu hizmeti işleminin el ile yeniden başlatılarak güncellemelerin nasıl seçileceğini gösterir. Sunucu hizmeti nesnesi, başlatma kuyruğu JUPITER.INITIATION.QUEUE olarak belirtilmek üzere değiştirilir. **ALTER SERVICE** MQSC komutu kullanılır:

```
ALTER SERVICE(S1) +  
STARTARG('-m +QMNAME+ -q JUPITER.INITIATION.QUEUE')
```

Not: Çalışmakta olan bir hizmet, yeniden başlatılıncaya kadar, hizmet tanımında herhangi bir güncelleme toplamaz.

5. The server service process is restarted so that the alteration is picked up, using the **STOP SERVICE** and **START SERVICE** MQSC commands:

```
STOP SERVICE(S1)
```

Takip eden:

```
START SERVICE(S1)
```

Sunucu hizmeti işlemi yeniden başlatılır ve “4” sayfa 173 içinde yapılan değişiklikleri alır.

Not: MQSC komutu (**STOP SERVICE**) yalnızca hizmet tanımında bir **STOPCMD** bağımsız değişkeni belirtilirse kullanılabilir.

İlgili başvurular

ALTER HİZMETİ

HİZMET

SVSTATUS GÖRÜNTÜLE

[Hizmeti başlat](#)

[Hizmeti durdur](#)

Komut hizmeti nesnesinin kullanılması

Bu örnek, bir kuyruk yöneticisi başlatıldığında ya da durdurulduğunda, girişlerin işletim sisteminin sistem günlüğüne yazan bir programı başlatmak için bir komut hizmeti nesnesinin nasıl tanımlanacağı gösterilmektedir.

1. The command service object is defined, using the **DEFINE SERVICE** MQSC command:

```
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')
```

Burada:

logger is the AIX or Linux system supplied command to write to the system log.
+QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.

İlgili başvurular

[HİZMET](#)

Bir kuyruk yöneticisi yalnızca bir kuyruk yöneticisi sona erdiğinde, bir komut hizmeti nesnesi kullanılması

Bu örnek, bir kuyruk yöneticisi yalnızca durdurulduğunda, girişlerin işletim sisteminin sistem günlüğüne yazan bir programı başlatmak için bir komut hizmeti nesnesinin nasıl tanımlanacak olduğunu gösterir.

1. The command service object is defined, using the **DEFINE SERVICE** MQSC command:

```
DEFINE SERVICE(S3) +
CONTROL(QMGR) +
SERVTYPE(COMMAND) +
STOPCMD('/usr/bin/logger') +
STOPARG('Queue manager +QMNAME+ stopping')
```

Burada:

logger is a sample program provided with IBM MQ that can write entries to the operating system's system log.
+QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.

İlgili başvurular

[HİZMET](#)

Diğer bağımsız değişkenler için daha fazla

Bu örnek, bir kuyruk yöneticisi başlatıldığında runserv adlı programı başlatmak için bir sunucu hizmeti nesnesinin nasıl tanımlanacak olduğunu gösterir.

Bu örnek, Windows biçim yolu ayırıcı karakterleriyle yazılmıştır.

Başlangıç programına geçirilecek bağımsız değişkenlerden biri, boşluk içeren bir dizilimdir. Bu bağımsız değişkenin tek bir dizgi olarak geçirilmesi gerekir. Bunu elde etmek için, komut hizmeti nesnesini tanımlamak için aşağıdaki komutta gösterildiği gibi çift tırnak imi kullanılır:

1. Sunucu hizmeti nesnesi, **DEFINE SERVICE** MQSC komutu kullanılarak tanımlanır:

```
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\" , boşluk içeren, tek bir dizgi olarak geçirilecek bir dizgidir.

İlgili başvurular

HİZMET

Bir hizmeti otomatik olarak başlatma

Bu örnekte, kuyruk yöneticisi başlatıldığında Tetikleme İzleyicisi 'nin otomatik olarak başlatılması için kullanılacak bir sunucu hizmeti nesnesinin nasıl tanımlanabileceği gösterilmektedir.

1. Sunucu hizmeti nesnesi, **DEFINE SERVICE** MQSC komutu kullanılarak tanımlanır:

```
DEFINE SERVICE(TRIG_MON_START) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('runmqtm') +
STARTARG('-m +QMNAME+ -q +IQNAME+')
```

Burada:

+QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.

+IQNAME+ , kullanıcı tarafından, kullanıma hazırlama kuyruğunun adını temsil eden service.env kütüklerinden birinde tanımlanan bir ortam değişkenidir.

İlgili başvurular

HİZMET

Tetikleme için nesnelere yönetme

IBM MQ , kuyruklardaki belirli koşullar karşılandığında otomatik olarak bir uygulama başlatmanızı sağlar. Örneğin, kuyruklardaki ileti sayısı belirli bir sayıya ulaştığında bir uygulamayı başlatmak isteyebilirsiniz. Bu olanağa *tetikleme* adı verilir. Tetiklemeyi destekleyen nesnelere tanımlamanız gerekir.

Tetikleme, Tetikleyicileri kullanarak IBM MQ uygulamalarının başlatılması konusunda ayrıntılı olarak açıklanmıştır.

Tetikleme için bir uygulama kuyruğu tanımlama

Uygulama kuyruğu, ileti alışverişi için uygulamalar tarafından, MQI aracılığıyla kullanılan bir yerel kuyruktur. Tetikleme, uygulama kuyruğunda tanımlanacak kuyruk özniteliklerinin sayısını gerektirir.

Tetikleme, **Trigger** özneliği (MQSC komutlarındaki tetikleme) tarafından etkinleştirilir. In this example, a trigger event is to be generated when there are 100 messages of priority 5 or greater on the local queue MOTOR.INSURANCE.QUEUE, as follows:

```
DEFINE QLOCAL (MOTOR.INSURANCE.QUEUE) +
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
MAXMSGL (2000) +
DEFPSIST (YES) +
INITQ (MOTOR.INS.INIT.QUEUE) +
TRIGGER +
TRIGTYPE (DEPTH) +
TRIGDPH (100)+
TRIGPRI (5)
```

Burada:

QLOCAL (MOTOR.INSURANCE.QUEUE)

Tanımlanmakta olan uygulama kuyruğunun adı.

PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

Bir tetikleme izleme programı tarafından başlatılacak uygulamayı tanımlayan süreç tanımlamasının adıdır.

MAXMSGL (2000)

Kuyruktaki ileti uzunluğu üst sınırı.

DEFPSIST (YES)

Bu kuyruktaki iletilerin varsayılan olarak kalıcı olduğunu belirtir.

INITQ (MOTOR.INS.INIT.QUEUE)

Kuyruk yöneticisinin tetikleme iletisini koyacağı başlangıç kuyruğunun adı.

TRIGGER

Tetikleme özneliği değeri mi?

TRIGTYPE (DEPTH)

Gerekli önceliğin (TRIGMPRI) ileti sayısı TRIGDPTH ' de belirtilen sayıya ulaştığında tetikleme olayının üretileceğini belirtir.

TRIGDPTH (100)

Bir tetikleme olayı oluşturmak için gereken ileti sayısı.

TRIGMPRI (5)

Bir tetikleme olayı oluşturulup oluşturulmayacağınıza karar vermek için kuyruk yöneticisi tarafından sayılacak iletilerin önceliğidir. Yalnızca, 5 ya da üzeri önceliğe sahip iletiler sayılır.

Başlatma kuyruğu tanımlanması

Bir tetikleme olayı ortaya çıktığında, kuyruk yöneticisi bir tetikleme iletisini uygulama kuyruğu tanımlamasında belirtilen başlangıç kuyruğuna koyar. Başlatma kuyruklarının özel ayarları yoktur, ancak yerel kuyruk MOTOR.INS.INIT.QUEUE :

```
DEFINE QLOCAL(MOTOR.INS.INIT.QUEUE) +
GET (ENABLED) +
NOSHARE +
NOTRIGGER +
MAXMSGL (2000) +
MAXDEPTH (1000)
```

Süreç tanımlama

Bir süreç tanımlaması yaratmak için DEFINE PROCESS komutunu kullanın. Süreç tanımlaması, iletileri uygulama kuyruğundan işlemek için kullanılacak uygulamayı tanımlar. Uygulama kuyruğu tanımlaması, kullanılacak işlemi ve dolayısıyla, uygulama kuyruğunu, iletilerin işlemek üzere kullanılacak uygulamayla ilişkilendirir. This is done through the PROCESS attribute on the application queue MOTOR.INSURANCE.QUEUE. Aşağıdaki MQSC komutu gereken işlemi tanımlıyor: MOTOR.INSURANCE.QUOTE.PROCESS, bu örnekte tanımlanır:

```
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 adı.

DESCR ('Insurance request message processing')

Bu tanımın ilişkilendirdiği uygulama programını açıklar. Bu metin, DISPLAY PROCESS komutunu kullandığınızda görüntülenir. Bu, işlemin ne yaptığını tanımlamanıza yardımcı olabilir. Dizide boşluk kullanırsanız, dizgiyi tek tırnak işareti içine almalısınız.

APPLTYPE (UNIX)

Başlatılacak uygulama tipidir.

APPLICID ('/u/admin/test/IRMP01')

Uygulama yürütülür dosyasının adı, tam olarak nitelenmiş bir dosya adı olarak belirtilir. Windows sistemlerinde, tipik bir APPLICID değeri c:\app1\test\irmp01.exe olur.

USERDATA ('open, close, 235')

Kullanıcı tanımlı veridir ve uygulama tarafından kullanılabilir.

Süreç tanımlaması özniteliklerini görüntüleme

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)
```

Ayrıca, bir süreç tanımlamasını silmek için, MQSC komutu ALTER PROCESS deyimini kullanarak, varolan bir süreç tanımlamasını değiştirebilir ve DELETE PROCESS komutunu 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 yüklemeye önce bir düzenleyicide güncelleyebilmeniz için dosya insan tarafından okunabilir. Dosyayı düzenlerseniz, biçimini değiştirmemelisiniz.
2. IBM MQ 9.1.0 lanağında, **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 kullanılabilecek deęiřtirgelerle ilgili ek bilgi için [dmpmqmsg](#) kısmına bakın.

Windows makinesi kullanarak Linux üzerinde dmpmqmsg yardımcı programını kullanma örneęi

Bir Linux makinesinde, aynı kuyruk yöneticisinde başka bir kuyruęa (Q2) taşımak istedięiniz iletileri (Q1) içeren bir kuyruk yöneticiniz vardır. **dmpmqmsg** yardımcı programını bir Windows makinesinden başlatmak istiyorsunuz.

Kuyruk (Q1), örnek **amqspu**t (yerel kuyruk yöneticisi) ya da **amqspu**tc (uzak kuyruk yöneticisi) uygulaması kullanılarak eklenen dört iletiye sahiptir.

Linux makinesinde řunları görürsünüz:

```
display ql(Q1) CURDEPTH
      2 : display ql(Q1) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q1)
      TYPE(QLOCAL)
      CURDEPTH(4)
```

MQSERVER ortam deęiřkenini Linux içindeki kuyruk yöneticisini gösterecek řekilde ayarlayın. Örneęin:

```
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\mqqlload.txt içinde saklayın.

MQSERVER tarafından oluşturulan Linux anasisteminde ve kapısında çalışan *QM_VER* kuyruk yöneticisine uzak istemci olarak baęlanın. Baęlantıyı uzak istemci olarak řu öznitelięi kullanarak elde edebilirsiniz: -c.

```
dmpmqmsg -m QM_VER -i Q1 -f c:\temp\mqqlload.txt -c
Read      - Files:    0  Messages:    4  Bytes:    22
Written   - Files:    1  Messages:    4  Bytes:    22
```

c:\temp\mqqlload.txt çıkış kütüęü, **dmpmqmsg** yardımcı programının anladığı bir biçimi kullanarak metin içeri.

Windows makinesinde, Linux makinesindeki yükleme kuyruęunu (Q2) Windows makinesindeki bir dosyadan yüklemek için **dmpmqmsg** komutunu (-i seçeneęi yerine -o seçeneęini kullanarak) verin:

```
dmpmqmsg -m QM_VER -o Q2 -f c:\temp\mqqlload.txt -c
Read      - Files:    1  Messages:    4  Bytes:    22
Written   - Files:    0  Messages:    4  Bytes:    22
```

Linux makinesinde, kuyruktan artık dosyadan geri yüklenen dört ileti olduęunu 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 ql(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(QLLOCAL)
      CURDEPTH(0)
```

Komutun ve deęiřtirgelerinin açıklaması için bkz. [dmpmqmsg](#) .

İlgili kavramlar

“[dmpmqmsg yardımcı programını kullanma örnekleri](#)” sayfa 179

dmpmqmsg yardımcı programını (önceden **qload**) kullanabileceğiniz basit yöntemler. This utility is incorporated into the product from IBM MQ 8.0.

dmpmqmsg yardımcı programını kullanma örnekleri

dmpmqmsg yardımcı programını (önceden **qload**) kullanabileceğiniz basit yöntemler. This utility is incorporated into the product from IBM MQ 8.0.

Daha önce **qload** yardımcı programı SupportPac MO03olarak kullanılabilmişti.

Kuyruğun bir dosyaya boşa alınması

Kuyruktaki iletileri saklamak için, komut satırında aşağıdaki seçenekleri kullanın:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile
```

Bu komut, kuyruktan gelen iletilerin bir kopyasını alır ve belirtilen kütükada saklar.

Kuyruğu bir dosya dizisine boşa al

Dosya adında bir insert karakteri kullanarak bir kuyruğu bir dosya dizisine boşa çıkarabilirsiniz. Bu kipte her ileti yeni bir dosyaya yazılır:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile%n
```

This command unloads the queue to files, myfile1, myfile2, myfile3, and so on.

Dosyadan bir kuyruk yükle

“[Kuyruğun bir dosyaya boşa alınması](#)” sayfa 179' ta kaydetmiş olduğunuz iletilerle bir kuyruğu yeniden yüklemek için, komut satırında aşağıdaki seçenekleri kullanın:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

This command unloads the queue to files, myfile1, myfile2, myfile3, and so on.

Bir dosya dizisinden kuyruk yükle

Dosya adında bir insert karakteri kullanarak, bir dosya dizisinden kuyruk yükleyebilirsiniz. Bu kipte her ileti yeni bir dosyaya yazılır:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

This command loads the queue to files, myfile1, myfile2, myfile3, and so on.

İletileri bir kuyruktan başka bir kuyruğa kopyala

Replace the file parameter in “[Kuyruğun bir dosyaya boşa alınması](#)” sayfa 179, with another queue name and use the following options:

```
dmpmqmsg -m QM1 -i Q1 -o Q2
```

Bu komut, bir kuyruktan gelen 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
```

İletileri bir kuyruktan başka bir kuyruğa taşı

“Dosyadan bir kuyruk yükle” sayfa 179 üzerinde bir çeşitleme. Yalnızca bir kuyruğu göz attıran `-i` (küçük harf) ve bir kuyruktan otomatik olarak yok eden `-I` (büyük harf) arasındaki ayrımı not edin:

```
dmpmqmsg -m QM1 -I Q1 -o Q2
```

Bir kuyruktan eski iletileri bir kuyruktan başka bir kuyruğa taşı

Bu örnek, yaş seçimine ilişkin kullanımı gösterir. İletiler, bir yaş aralığı içinde ya da daha küçük olan, daha küçük ya da daha eski bir ileti olabilir.

```
dmpmqmsg -m QM1 -I Q1 -o Q2 -T1440
```

Kuyruklardaki ileti çağlarını görüntüler

Komut satırında aşağıdaki seçenekleri kullanın:

```
dmpmqmsg -m QM1 -i Q1 -f stdout -dT
```

İleti kütüğüyle çalış

Having unloaded the message from your queue, as in “[Kuyruğun bir dosyaya boşa alınması](#)” sayfa 179, you might want to edit the file.

Ayrıca, kuyruğun boşaltılması sırasında belirtmediğiniz görüntü seçeneklerinden birini kullanmak için dosyanın biçimini değiştirmek de isteyebilirsiniz.

You can use the **dmpmqmsg** utility to reprocess the file into the required format even after the unload of the queue has taken place. Komut satırındaki aşağıdaki seçenekleri kullanın.

```
dmpmqmsg -f c:\oldfile -f c:\newfile -dA
```

Komutun ve parametrelerinin açıklaması için bkz. [dmpmqmsg](#) .

Uzak IBM MQ nesneleriyle çalışma

Uzak kuyruk yöneticilerindeki IBM MQ nesnelерini, MQSC komutlarını, PCF komutlarını ya da administrative REST API komutunu kullanarak yönetebilirsiniz. Bu yöntemlerin herhangi birini kullanmadan önce, uzak kuyruk yöneticisine komutların gönderilebilmesi ve yerel kuyruk yöneticisi

tarafından alınan yanıtların gönderilebilmesi için yerel kuyruk yöneticisi ile uzak kuyruk yöneticisi arasındaki iletim kuyruklarını ve kanalları tanımlamanız gerekir. Diğer bir seçenek olarak, bir kuyruk yöneticisi kümesi yapılandırabilir ve daha sonra aynı uzak denetim yöntemlerini kullanabilirsiniz.

Bu görev hakkında

Uzak denetim için kuyruk yöneticilerini hazırlamak üzere, yerel kuyruk yöneticisinde aşağıdaki nesnelere yapılandırmanız gerekir:

- Bir dinleyici.
- Uzak kuyruk yöneticisi adına sahip bir iletim kuyruğu.
- Uzak kuyruk yöneticisine ilişkin bağlantı ayrıntılarına sahip bir gönderen kanalı.
- Uzak kuyruk yöneticisinde gönderen kanalıyla aynı adı taşıyan bir alıcı kanalı.

Uzak kuyruk yöneticisinde aşağıdaki nesnelere de yapılandırmanız gerekir:

- Bir dinleyici.
- Yerel kuyruk yöneticisinin adına sahip bir iletim kuyruğu.
- Yerel kuyruk yöneticisine ilişkin bağlantı ayrıntılarına sahip bir gönderen kanalı.
- Yerel kuyruk yöneticisinde gönderen kanalıyla aynı adı taşıyan bir alıcı kanalı.

Bu nesnelere yapılandırılmasıyla ilgili daha fazla bilgi için bkz. [“Uzak denetim için kuyruk yöneticilerinin yapılandırılması” sayfa 181](#).

Diğer bir seçenek olarak, bir kuyruk yöneticisi kümesi de yapılandırabilirsiniz. *Küme*, kuyruk yöneticilerinin karmaşık iletim kuyruğu, kanal ve kuyruk tanımları olmadan tek bir ağ üzerinden doğrudan iletişim kurabilecekleri bir şekilde ayarlanan bir kuyruk yöneticilerinden oluşan bir gruptur. Kümeler kolayca ayarlanabilir ve genellikle bir şekilde mantıksal olarak ilişkili olan ve veri ya da uygulamaları paylaşmak zorunda olan kuyruk yöneticilerine sahip olur. En küçük küme bile sistem yönetimi maliyetlerini düşürür.

Bir kümede kuyruk yöneticisi ağı oluşturulması, geleneksel dağıtılmış kuyruklama ortamı oluşturmaktan daha az tanımlamayı içerir. Daha az tanımlamaya sahip olarak, ağınıza daha hızlı ve kolay bir şekilde ayarlayabilir ya da değiştirebilir ve tanımlarınızda hata yapma riskini azaltabilirsiniz.

Bir küme oluşturmak için, her kuyruk yöneticisi için bir küme gönderen (CLUSDR) ve bir küme alıcısı (CLUSTRCVR) tanımlamasına gereksinim duyarsınız. İletim kuyruğu tanımlamalarına ya da uzak kuyruk tanımlarına gerek yoktur. Uzak yönetim ilkeleri bir küme içinde kullanıldığında aynıdır, ancak tanımlamalar kendilerini büyük ölçüde basitleştirmektedir.

Küme yapılandırılmasına ilişkin ek bilgi için [Kuyruk yöneticisi kümesinin yapılandırılması](#) başlıklı konuya bakın.

Yordam

- Uzak IBM MQ nesnelere nasıl yönetileceği hakkında bilgi için aşağıdaki alt konulara bakın:
 - [“Uzak denetim için kuyruk yöneticilerinin yapılandırılması” sayfa 181](#)
 - [“Uzak denetim için komut sunucusunun yönetilmesi” sayfa 185](#)
 - [“Uzak kuyruk yöneticisinde MQSC komutları verilmesi” sayfa 186](#)
 - [“Kodlanmış karakter kümeleri arasında veri dönüştürme” sayfa 188](#)

Uzak denetim için kuyruk yöneticilerinin yapılandırılması

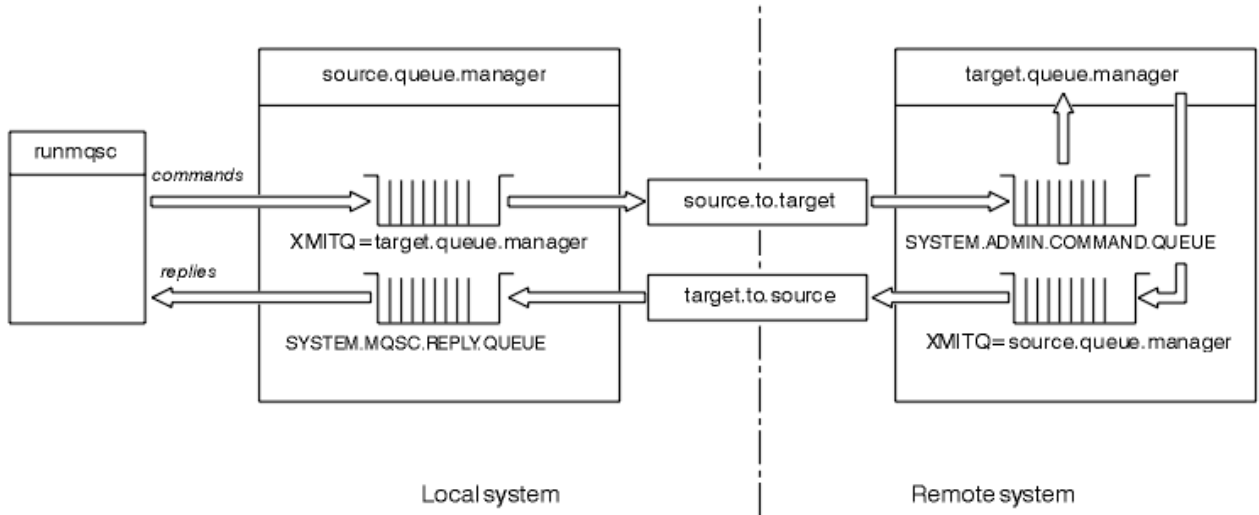
You can administer a remote queue manager from a local queue manager by using the administrative REST API, MQSC, or PCF commands. Uzak kuyruk yöneticisi aynı sistemde, farklı bir kurulumda ya da aynı ortamdaki farklı bir sistemde ya da farklı bir IBM MQ ortamıyla aynı olabilir. Bir kuyruk yöneticisini yerel bir kuyruk yöneticisinden uzaktan denetlemeden önce, her kuyruk yöneticisinden bir gönderici ve alıcı kanalı, bir dinleyici ve bir iletim kuyruğu oluşturmanız gerekir. Bu kanallar ve kuyruklar, komutların uzak kuyruk yöneticisine gönderilmesini ve yerel kuyruk yöneticisinde alınacak yanıtların gönderilmesini sağlar. Bu

kuyrukları ve kanalları yaratma yordamı, administrative REST API, MQSC ya da PCF komutlarını kullanmak isteyip istememediğiniz aynıdır.

Başlamadan önce

- The following procedure uses example queue managers `source.queue.manager`, and `target.queue.manager`. Bu adımları izlemek için sisteminizde bu kuyruk yöneticilerini yaratmanız ve başlatmanız ya da ilgili adımlarda kendi kuyruk yöneticisi adlarınızın yerine geçmeniz 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 da kullanabilirsiniz, ancak uygun adımlarda kapı değerlerinizin yerine geçmeniz gerekir.
- Yordamda bulunan komutları yerel olarak ya da Telnet gibi bir ağ olanağı üzerinden çalıştırmanız gerekir.

Bu görev hakkında



Şekil 15. Uzak denetim için kanalların ve kuyrukların ayarlanması

Şekil 15 sayfa 182 , uzak denetim için gereksinim duyduğunuz kuyruk yöneticilerinin, kuyrukların ve kanalların yapılandırıldığını gösterir:

- The object `source.queue.manager` is the source queue manager from which you can issue administrative REST API, MQSC, or PCF commands, and to which the results of these commands are returned.
- `target.queue.manager` nesnesi, komutları işleyen ve herhangi bir işletmen iletimi oluşturan hedef kuyruk yöneticisinin adıdır.
- Komutlar, uzak kuyruk yöneticisiyle aynı adı taşıyan iletim kuyruğuna yerleştirilmesini sağlar. Bu durumda, `target.queue.manager`. İletim kuyruğu, MCA ' nın bunları alması ve uzak kuyruk yöneticisine göndermeden önce iletileri geçici olarak tutan özelleştirilmiş bir yerel kuyruktur.
- The commands get sent by the `source.to.target` channel to the `SYSTEM.ADMIN.COMMAND.QUEUE` on the remote queue manager. Kanalın her bir ucunda ayrı bir tanım vardır. Bir ucu gönderici, diğer ucu da alıcı. İki tanım aynı ada sahip olmalı ve birlikte tek bir iletim kanalı oluşturur.
- Komut çıkışı, komutun gönderildiği yerel kuyruk yöneticisiyle aynı adı taşıyan uzak iletim kuyruğuna konmuştur. Bu durumda, `source.queue.manager`.
- Çıkış, `target.to.source` kanalı tarafından uygun bir yanıt kuyruğuna gönderilir. Bu ileti, özgün komut tarafından alınır ve çıkarılır.

Yordam

1. On the remote system queue manager, ensure that the command queue SYSTEM.ADMIN.COMMAND.QUEUE is present. Bu kuyruk, bir kuyruk yöneticisi yaratıldığında varsayılan olarak yaratılır.
2. Uzak sistemde, komut sunucusunun kuyruk yöneticisinde çalıştığını doğrulayın. Komut sunucusu çalışmıyorsa, uzak denetim olanağı olanaklı değildir:

- a) Kuyruk yöneticisi için **runmqsc** ' u başlatın. For example, for queue manager target.queue.manager, enter the following command:

```
runmqsc target.queue.manager
```

- b) Komut sunucusunun durumunu görüntülemek için aşağıdaki komutu girin:

```
DISPLAY QMSTATUS CMDSERV
```

- c) Aşağıdaki komutu girerek **runmqsc** ' dan çıkın:

```
end
```

- d) Komut sunucusu başlatılmamışsa, başlatın. For example, for queue manager target.queue.manager, enter the following command:

```
strmqcsv target.queue.manager
```

3. Yerel kuyruk yöneticisinin kanallarını, dinleyicisini ve iletim kuyruğunu tanımlayın:

- a) Kuyruk yöneticisi için **runmqsc** ' u başlatın. For example, for queue manager source.queue.manager, enter the following command:

```
runmqsc source.queue.manager
```

- b) Gönderen kanalını tanımlayın. Bu gönderen kanalı, uzak kuyruk yöneticisinde alıcı kanalıyla aynı adı olmalıdır. Örneğin, uzak kuyruk yöneticisinin IP adresi ve dinleyicinin kapı numarası için **CONNNAME** değerini 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 kanalı, uzak kuyruk yöneticisinde gönderen kanalıyla aynı adı olmalıdır. Örneğin, şu komutu girin.

```
DEFINE CHANNEL ('target.to.source') +  
CHLTYPE(RCVR) +  
TRPTYPE(TCP)
```

- d) Yerel kuyruk yöneticisinde dinleyici 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 başlat. Örneğin, şu komutu girin.

```
START LISTENER ('source.queue.manager')
```

g) Aşağıdaki komutu girerek **runmqsc** ' dan çıkın:

```
end
```

4. Uzak kuyruk yöneticisinin kanallarını, dinleyicisini ve iletim kuyruğunu tanımlayın:

a) Kuyruk yöneticisi için **runmqsc** ' u başlatın. For example, for queue manager `target.queue.manager`, enter the following command:

```
runmqsc target.queue.manager
```

b) Gönderen kanalını tanımlayın. Bu gönderen kanalı, yerel kuyruk yöneticisindeki alıcı kanalıyla aynı ada sahip olmalıdır. Örneğin, yerel kuyruk yöneticisinin IP adresi ve dinleyiciye ilişkin kapı numarası için **CONNNAME** değerini 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 günlük nesnesi kanalı, yerel kuyruk yöneticisinde bulunan gönderen kanalıyla aynı adı 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 başlat. Örneğin, şu komutu girin.

```
START LISTENER ('target.queue.manager')
```

g) Aşağıdaki komutu girerek **runmqsc** ' dan çıkın:

```
end
```

5. Yerel sistemde gönderen kanalını başlatın:

a) Kuyruk yöneticisi için **runmqsc** ' u başlatın. For example, for queue manager `source.queue.manager`, enter the following command:

```
runmqsc source.queue.manager
```

b) Gönderici kanalını başlatın. Örneğin, şu komutu girin.

```
START CHANNEL ('source.to.target')
```


c) Aşağıdaki komutu girerek **runmqsc** ' dan çıkın:

```
end
```

6. Uzak sistemde gönderen kanalını başlatın:

a) Kuyruk yöneticisi için runmqsc programını başlatın. For example, for queue manager target.queue.manager, enter the following command:

```
runmqsc target.queue.manager
```

b) Gönderici kanalını başlatın. Örneğin, şu komutu girin.

```
START CHANNEL ('target.to.source')
```

c) Aşağıdaki komutu girerek **runmqsc** ' dan çıkın:

```
end
```

7. Yerel sistemden uzak kuyruk yöneticisine bir MQSC komutu göndererek yapılandırmanın başarıyla tamamlandığını sına sına:

a) Start **runmqsc** for the remote queue manager from the local system. Örneğin, şu komutu girin.

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

b) Aşağıdaki komutu girerek uzak kuyruk yöneticisinde kuyukları görüntüleyin:

```
DISPLAY QUEUE (*)
```

Başarıda, uzak kuyruk yöneticisinden gelen kuyukların listesi görüntülenir.

c) Bu adımlar işe yaramazsa, her iki sistemdeki kanalların çalışır durumda olup olmadığını denetleyin. Kanallar çalışmıyorsa ve başlatılamazsa, kanalların ve iletim kuyuklarının doğru şekilde yapılandırıldığını ve komut sunucusunun çalıştığını doğrulayın. Örneğin, gönderen kanalları için doğru CONNAME ' in belirtildiğini ve iletim kuyuklarının doğru adlara sahip olup olmadığını denetleyin. Ayrıca, sorunun çözülmesine yardımcı olabilecek güvenlik kural dışı durumları olup olmadığını görmek için kuyruk yöneticisi günlüklerini de denetleyin.

Sonuçlar

Kuyruk yöneticilerinizin konfigürasyonu, uzak kuyruk yöneticisini yerel sistemden uzaktan denetlemek üzere tanımlanmıştır.

Sonraki adım

- MQSC komutlarını kullanarak uzaktan yönetim hakkında daha fazla bilgi edinin: [“Uzak kuyruk yöneticisinde MQSC komutları verilmesi” sayfa 186](#)
- Denetim programlarını PCF komutları kullanarak yazmakla ilgili daha fazla bilgi edinin: [“IBM MQ Programları Komut Biçimlerinin Kullanılması” sayfa 24.](#)
- Uzak yönetim için administrative REST API komutunu kullanma hakkında daha fazla bilgi edinin: [“REST APIkomutunu kullanarak uzaktan yönetim” sayfa 76.](#)

Uzak denetim için komut sunucusunun yönetilmesi

Her kuyruk yöneticisinin kendisiyle ilişkilendirilmiş bir komut sunucusu vardır. Komut sunucusu, uzak kuyruk yöneticilerinden gelen komutları ya da uygulamalardan PCF komutlarından herhangi bir komutu işler. Bu komut, kuyruk yöneticisine gönderilen komutları işlenmek üzere gösterir ve bir tamamlanma kodu ya da işletmen iletişi döndürür. Komut sunucusunun durumunu başlatabilir, durdurabilir ve görüntüleyebilirsiniz. Bir komut sunucusu, PCF komutlarının, MQAI ' nin ve ayrıca uzak yönetim için de yer alan tüm denetim için zorunludur.

Başlamadan önce

Kuyruk yöneticisi özniteliğinin değerine bağlı olarak, **SCMDSERV**, kuyruk yöneticisi başlatıldığında komut sunucusu otomatik olarak başlatılır ya da el ile başlatılmalıdır. Komut sunucusu otomatik olarak başlatılırsa, komut sunucusunu başlatmak ve durdurmak için strmqcsv ya da endmqcsv komutlarını kullanamazsınız. You can change the value of the **SCMDSERV** attribute by using the MQSC command **ALTER QMGR**. Varsayılan olarak, komut sunucusu otomatik olarak başlatılır.

Kuyruk yöneticisinin durdurulması, kendisiyle ilişkili komut sunucusunu da sona erdirir.

Yordam

- Komut sunucusunun durumunu görüntüle:
 - a) Start **runmqsc** for the appropriate queue manager by entering the following command:

```
runmqsc target.queue.manager
```

Burada `target.queue.manager`, komut sunucusunun görüntülenmekte olduğu kuyruk yöneticidir.

- b) Aşağıdaki MQSC komutunu girerek komut sunucusu durumunu görüntüleyin:

```
DISPLAY QMSTATUS CMDSERV
```

- c) Aşağıdaki komutu girerek **runmqsc** ' dan çıkın:

```
end
```

- Komut sunucusu otomatik olarak başlatılacak şekilde ayarlanmadıysa, aşağıdaki komutu girerek komut sunucusunu başlatın:

```
strmqcsv target.queue.manager
```

Burada `target.queue.manager`, komut sunucusunun başlatılmakta olduğu kuyruk yöneticidir.

- Komut sunucusu otomatik olarak başlatılacak şekilde ayarlanmadıysa, aşağıdaki komutu girerek komut sunucusunu durdurun:

```
endmqcsv target.queue.manager
```

Burada `target.queue.manager`, komut sunucusunun durdurulmakta olduğu kuyruk yöneticidir.

Varsayılan değer olarak, komut sunucusu denetimli bir şekilde durur. Komutta -i işaretini ekleyerek komut sunucusunu hemen durdurabilirsiniz.

Uzak kuyruk yöneticisinde MQSC komutları verilmesi

Uzak denetim için kuyruk yöneticilerini yapılandırdıktan sonra, uzak bir kuyruk yöneticisine MQSC komutlarını çalıştırmak için yerel bir sistemde **runmqsc** komutunun belirli bir biçimini kullanabilirsiniz. Her komut, uzak kuyruk yöneticisinin SYSTEM.ADMIN.COMMAND.QUEUEadlı komut kuyruğuna bir Escape PCF olarak gönderilir. Yanıtlar, SYSTEM.MQSC.REPLY.QUEUE kuyruğunda alınır.

Başlamadan önce

Bir kuyruk yöneticisini uzaktan denetlemek için MQSC komutlarını kullanarak, kanalları, iletim kuyruklarını, dinleyicileri ve komut sunucusunu yapılandırmak için "[Uzak denetim için kuyruk yöneticilerinin yapılandırılması](#)" sayfa 181 içindeki adımları tamamlamanız gerekir.

Yordam

1. Komut sunucusunun uzak kuyruk yöneticisinde çalıştığından emin olun.
Bir kuyruk yöneticisine komut sunucusunun nasıl başlatılabilmesiyle ilgili bilgi için bkz. [“Uzak denetim için komut sunucusunun yönetilmesi” sayfa 185.](#)
2. Kaynak kuyruk yöneticisinde, MQSC komutlarını aşağıdaki iki yoldan biriyle çalıştırabilirsiniz:

- **runmqsc** komutunu aşağıdaki komutlarla başlayarak etkileşimli olarak kullanın:

- **z/OS** Uzak kuyruk yöneticisi z/OSüzeriniyse, aşağıdaki komutu girin:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager
```

- **Multi** Uzak kuyruk yöneticisi z/OS' de değilse, şu komutu girin:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

- Komut kütüğünden:

- a. Uzak sistemde çalıştırılacak MQSC komutlarını bir metin dosyasında, her satır için bir komut olarak koyun.
- b. **runmqsc** komutundaki -v işaretini kullanarak yerel kuyruk yöneticisine ilişkin MQSC komutlarınızı doğrulayın. -v işareti, komutların geçerli olup olmadığını denetler, ancak bunları çalıştırmaz. Uzak kuyruk yöneticisi için geçerli olan ancak yerel kuyruk yöneticisi için geçerli olmayan komutların bazı komutların başarısız olabileceğini unutmayın:

```
runmqsc -v source.queue.manager < myCmdFile.in > results.out
```

The `myCmdFile.in` contains the MQSC commands to check, and the `results.out` file contains the verification results for the commands.

- c. Aşağıdaki komutlardan birini girerek uzak kuyruk yöneticisinde komut kütüğünü çalıştırın:

- **z/OS** Uzak kuyruk yöneticisi z/OSüzeriniyse, aşağıdaki komutu girin:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager < myCmdFile.in >  
results.out
```

- **Multi** Uzak kuyruk yöneticisi z/OS' de değilse, şu komutu girin:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager < myCmdFile.in >  
results.out
```

Kullanılan parametreler aşağıdaki parametrelerle ilgili olabilir:

-w saniye

MQSC komutlarının dolaylı kipte çalıştırıldığını, komutların komut sunucusu giriş kuyruğuna konduğunu ve sırayla yürütüleceğini belirtir.

saniye değişkeni, uzak kuyruk yöneticisinden bir yanıt için beklenecek süreyi (saniye olarak) belirtir. Bu süreden sonra alınan yanıtlar atılır, ancak MQSC komutları uzak kuyruk yöneticiliklerinde çalışmaya devam eder. Komut zamanaşımına neden olduğunda yerel kuyruk yöneticisinde aşağıdaki ileti üretilir:

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

MQSC komutlarını vermeyi durdurduğunuzda, yerel kuyruk yöneticisi, gelen tüm zaman aşımına uğramış yanıtları görüntüler ve diğer yanıtları atar.

-x

Uzak kuyruk yöneticisinin bir z/OS kuyruk yöneticisi olduğunu belirtir.

-m *localQMGrAdi*

Uzak kuyruk yöneticisine komut sunmak için kullanmak istediğiniz yerel kuyruk yöneticisinin adını belirtir.

Sonraki adım

MQSC komutlarını uzaktan çalıştırmakta zorlanırsanız:

- Uzak kuyruk yöneticisinin çalışır durumda olup olmadığını denetleyin.
- Komut sunucusunun uzak sistemde çalışır durumda olup olmadığını denetleyin.
- Kanal bağlantı kesme aralığının süresinin dolmadığından emin olun. Örneğin, bir kanal başlatıldıysa, ancak bir süre sonra sona erdirilirse. Bu, özellikle kanalları el ile başlatabilirseniz önemlidir.
- Yerel kuyruk yöneticisinden gönderilen isteklerin hedef kuyruk yöneticisi için bir anlam verdiğinden emin olun. Örneğin, uzak kuyruk yöneticisinde desteklenmeyen deęiřtirgeleri içeren istekler.
- Ayrıca, [MQSC komutlarıyla ilgili sorunların çözümlenmesi](#) başlıklı konuya da bakın.

Kodlanmış karakter kümeleri arasında veri dönüřtürme

IBM MQ tanımlı biçimlerde (yerleşik biçimler olarak da bilinir) ileti verileri, her iki karakter kümesinin tek bir dil ya da benzer dillerdeki bir grup ile ilişkisi olması koşuluyla, bir kodlanmış karakter takımından kuyruk yöneticisi tarafından dönüřtürülebilmektedir.

Örneğin, her ikisi de Batı Avrupa dilleri için geçerli olduęu için, CCSID (CCSID) 850 ve 500 olan kodlanmış karakter kümeleri arasında dönüřtürme desteklenir.

ASCII ' ye EBCDIC newline (NL) karakter dönüřtürmeleri için bkz. [Tüm kuyruk yöneticileri](#).

Desteklenen dönüřtürmeler [Veri dönüřtürme işlemi](#) ' nde tanımlanır.

V 9.2.0

From IBM MQ 9.2.0, converting between CCSIDs 37 and 500 is supported on the IBM MQ Appliance, Windows, Linux, and macOS.

Kuyruk yöneticisi, iletileri yerleşik biçimlerde dönüřtüremedięinde

CCSID ' leri farklı ulusal dil gruplarını gösteriyorsa, kuyruk yöneticisi iletileri yerleşik biçimlerde otomatik olarak dönüřtüremez. Örneğin, CCSID 850 ile CCSID 1025 arasındaki dönüřtürme (Kiril alfabesinde kullanan diller için EBCDIC kodlanmış karakter kümesi), kodlanmış karakter kümesindeki karakterlerin birçoęu dięerinde gösterilemiyorsa desteklenmez. Farklı ulusal dillerde çalışan bir kuyruk yöneticisi aęınız varsa ve kodlanmış karakter kümelerinin bazıları arasında veri dönüřtürme desteklenmiyorsa, varsayılan dönüřtürmeyi etkinleřtirebilirsiniz.

ccsid_part2.tbl ' in geçerli olduęu platformlar için ek bilgi için [ccsid_part2.tbl](#) ' i kullanarak [“Varsayılan veri dönüřtürmenin belirtilmesi”](#) sayfa 192 ' a bakın. [ccsid_part2.tbl](#) dosyasının geçerli olduęu platformlarda varsayılan veri dönüřtürme işlemi, [“Varsayılan veri dönüřtürme”](#) sayfa 189 ' de anlatılır.

Geliřtirilmiř Unicode veri dönüřtürme desteęi

IBM MQ 9.0 öncesinde, ürünün önceki sürümleri, Basic Birden Çok Dilli Uçak (U+FFFF ' un üzerindeki kod noktaları) dıřındaki Unicode kod noktalarını içeren verilerin dönüřtürülmesini desteklemiyordu. Unicode data conversion support was limited to code points defined in the Unicode 3.0 standard, encoded in either UTF-8 or UCS-2, a 2-byte fixed-width subset of UTF-16.

Ürün IBM MQ 9.0 ' den, veri dönüřtürmede Unicode 8.0 standartında tanımlanan tüm Unicode karakterleri destekler. Bu, vekil çiftler de dahil olmak üzere UTF-16 için tam destek içerir (X ' D800 ' ileabovarasındaki 2 baytlık UTF-16 karakter çifti, U+FFFF ' un üzerinde bir Unicode kod noktasını temsil eden X ' DFFF ').

Karakter sıralarının birleştirilmesi, bir CCSID 'deki önceden oluşturulmuş karakterin başka bir CCSID' deki birleştirme karakteri dizisiyle eşlendiği durumlarda da desteklenir.

Bazı platformlarda, Unicode ve CCSID 'ler 1388, 1390, 1399, 4933, 5488 ve 16884 kodlarına veri dönüştürme işlemi genişletildi ve bu CCSID' ler için tanımlanmış olan tüm kod noktalarını desteklemek için, Unicode tamamlayıcı uçaklarda kod noktalarıyla eşleştirilen tüm kod noktalarını destekler.

CCSID 1390, 1399 ve 16884 vakalarında bu, JIS X 0213 (JIS2004) standardındaki karakterler içerir.

Unicode ve altı yeni CCSID ' ye (1374 ile 1379 arası) dönüştürme için de destek eklenmiştir.

ccsid_part2.tbl dosyası

IBM MQ 9.0 , ek bir dosya (ccsid_part2.tbl) sağlanıyor.

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 Windowsüzerinde, ccsid_part2.tbl varsayılan olarak *MQDataRoot\conv\table* dizininde bulunur. Bununla birlikte, IBM MQ for Windows üzerinde, desteklenen tüm kod kümeleri de kaydedilir.

Linux IBM MQ for Linuxüzerinde ccsid_part2.tbl , *MQDataRoot/conv/table* dizininde bulunur ve desteklenen kod kümeleri, IBM MQtarafından sağlanan dönüştürme çizelgelerinde tutulur.

Although the ccsid_part2.tbl file replaces the existing ccsid.tbl file used in previous versions of IBM MQ to supply additional CCSID information, the ccsid.tbl file continues to be parsed by IBM MQ and must therefore not be deleted.

Daha fazla bilgi için bkz [“ccsid_part2.tbl dosyası” sayfa 190.](#)

ccsid.tbl dosya

ccsid_part2.tbl kütüğünün geçerli olduğu altyapılarda, ccsid.tbl kütüğü aşağıdaki amaçlar için kullanılır:

- **AIX** AIX' ta, desteklenen kod kümeleri işletim sistemi tarafından dahili olarak tutulur.
- Bu, ek kod kümelerini belirtir. Ek kod kümeleri belirtmek için, ccsid.tbl dosyasını düzenlemeniz gerekir (bunun nasıl yapacağına ilişkin yönergeler dosyada sağlanır).
- Herhangi bir varsayılan veri dönüştürmesini belirtir.

ccsid.tbl' ta kaydedilen bilgileri güncelleştirebilirsiniz; örneğin, işletim sisteminizin gelecekteki bir yayın düzeyi ek kodlanmış karakter kümelerini destekliyorsa bunu yapmak isteyebilirsiniz.

Varsayılan veri dönüştürme

IBM MQ 9.0 ' den varsayılan veri dönüştürme yöntemi aşağıdaki altyapılarda değiştirildi:

- **Linux** Linux -tüm sürümler
- **Windows** Windows

See “Varsayılan veri dönüştürmenin belirtilmesi” sayfa 192 using `ccsid_part2.tbl` for further information.

Veri dönüştürmenin normalde desteklenmediği iki makine arasında kanal ayarlarsanız, kanalların çalışabilmesi için varsayılan veri dönüştürme özelliğini etkinleştirmeniz gerekir.

Varsayılan veri dönüştürme işlemini geçerli kılmak için `ccsid_part2.tbl` ' un geçerli olduğu altyapılarda, varsayılan EBCDIC CCSID değerini ve varsayılan ASCII CCSID değerini belirtmek için `ccsid.tbl` dosyasını düzenleyin. Bunun nasıl yapacağına ilişkin yönergeler dosyada yer alır. Bunu, kanalları kullanarak bağlanabilecek tüm makinelerde yapmanız gerekir. 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 hem EBCDIC ya da hem de ASCII ' dir, karakter verileri, dönüştürme yapılmadan hedef uygulamaya geçirilir.
- Bir CCSID, ASCII kodlanmış karakter takımını, diğeri ise EBCDIC kodlanmış karakter kümesini temsil ediyorsa, IBM MQ , `ccsid.tbl` içinde tanımlanan varsayılan veri dönüştürme CCSID 'sini kullanarak verileri dönüştürür.

Not: Dönüştürülen karakterleri, ileti için belirlenen kodlanmış karakter kümesinde aynı kod değerlerine sahip olanlara ve varsayılan kodlanmış karakter takımında sınırlamaya çalışın. Yalnızca IBM MQ nesne adları için geçerli olan karakter kümesini kullanıyorsanız (IBM MQ nesnelерinde adlandırma) genel olarak, bu şartı yerine getireceksiniz. Japonya 'da kullanılan EBCDIC CCSID 'ler 290, 930, 1279 ve 5026 ile özel durumlar ortaya çıkar; burada, küçük harflerin diğeri EBCDIC CCSID 'lerde kullanılanlardan farklı kodları vardır.

Kullanıcı tanımlı biçimlerde iletilerin dönüştürülmesi

Kuyruk yöneticisi, kullanıcı tanımlı biçimlerdeki iletileri bir kodlanmış karakter kümesinden diğeri dönüştüremez. Verileri kullanıcı tanımlı bir biçimde dönüştürmeniz gerekiyorsa, her biçim 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 Yazma verileri-dönüştürme çıkışları başlıklı konuya bakın.

Kuyruk yöneticisi CCSID 'si değiştiriliyor

Kuyruk yöneticisinin CCSID 'sini değiştirmek için ALTER QMGR komutunun CCSID özniteliğini kullandıysanız, komut sunucusu ve kanal programları da içinde olmak üzere tüm çalışan uygulamaların durdurup yeniden başlatıldığından emin olmak için kuyruk yöneticisini durdurun ve yeniden başlatın.

Kuyruk yöneticisi CCSID değeri değiştirildiğinde çalışmakta olan uygulamalar varolan CCSID 'yi kullanmaya devam ettiği 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 diğeri 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 kümeleri arasında veri dönüştürme” sayfa 188.
- 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. tblsirasında IBM MQ , karşılaşılan tüm CCSID ' leri IBM MQkomut 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  IS08859-1
  923  923  0  923  1  3  0  IS08859-15
 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 Transferkomutunu yönetmek için Managed File Transfer komutlarını kullanın. Ayrıca, bazı yönetim görevleri için IBM MQ Explorer ' yi de kullanabilirsiniz.

Aracı komut kuyruğuna bir ileti yerleştirerek aktarma işlemini başlat

Kaynak aracının komut kuyruğuna bir dosya aktarma iletisi koyarak da bir dosya aktarımı başlatabilirsiniz. Örnek bir komut kuyruğu adı SYSTEM.FTE.COMMAND.AGENT01. İletinin doğru kaynak aracısının komut kuyruğuna ulaşmasını sağlamanız gerekir; ileti, XML ' deki kaynak bilgilerle eşleşmeyen bir aracı tarafından alındıysa, ileti reddedilir.

Aktarma isteği XML ' i FileTransfer.xsd şemasına uygun olmalıdır ve < request> ögesini kök öge olarak kullanmalıdır. Bir aktarma isteği iletisinin yapısı ve içeriğiyle ilgili bilgi için [Dosya aktarma isteği ileti biçimi](#) başlıklı konuya bakın. Bir aracı komut kuyruğuna aktarma isteği iletisini nasıl yerleştirdiğiniz, göreve özgü bir görevdir. Örneğin, kuyruğa programlı olarak bir ileti koymak için IBM MQ Java API ' yı kullanabilirsiniz.

MFT aracısının başlatılması

Dosya aktarımı için bir Managed File Transfer aracısını kullanmadan önce aracıyı başlatmanız gerekir.

Bu görev hakkında

Komut satırından bir Managed File Transfer Agent işlemi başlatabilirsiniz. Bu durumda, sistem oturumunu kapattığınızda aracı işlemi durur.

ALW AIX, Linux, and Windows işletim sisteminde, bir aracıyı, sistemde oturum kapattığınızda çalışmaya devam edecek şekilde ve dosya aktarımlarını almaya devam edebilen bir aracı yapılandırabilirsiniz.

z/OS z/OS'ta, aracıyı etkileşimli bir oturuma gerek kalmadan JCL' den başlatılan bir görev olarak başlayacak şekilde yapılandırabilirsiniz.

Bir aracı çalışırken kurtarılamayan bir hata saptarsa, ilk hata verilerini yakalama (FDC) oluşturulduğunu ve aracının durdurulduğunu unutmayın.

Yordam

- Bir aracıyı komut satırından başlatmak için **fteStartAgent** komutunu kullanın. Daha fazla bilgi için bkz. [fteStartAgent](#).

ALW

Bir aracıyı, sistemde oturumu kapattığınızda çalışmaya devam edecek şekilde yapılandırmak için:

– **Windows** Windows' ta aracıyı bir Windows hizmeti olarak çalışacak şekilde yapılandırın. Daha fazla bilgi için bkz [“Windows aracısı olarak MFT aracısının başlatılması”](#) sayfa 193.

– **Linux** **AIX** AIX and Linux' ta, bir komut dosyası kullanarak aracıyı yeniden başlatma sırasında otomatik olarak başlayacak şekilde yapılandırın. Daha fazla bilgi için bkz [“AIX and Linux sistemi başlatma sırasında bir MFT aracısının başlatılması”](#) sayfa 195.

z/OS

z/OS'ta, aracıyı etkileşimli bir oturuma gerek kalmadan JCL' den başlatılan bir görev olarak başlayacak şekilde yapılandırın.

Daha fazla bilgi için bkz [“z/OS üzerinde bir MFT aracısının başlatılması”](#) sayfa 197.

Windows Windows aracısı olarak MFT aracısının başlatılması

Bir aracıyı Windows hizmeti olarak başlatabilirsiniz; böylece, Windows oturumunu kapattığınızda aracınız çalışmaya devam eder ve dosya aktarımlarını alabilir.

Bu görev hakkında

Windows'ta, bir aracıyı komut satırından başlattığınızda, aracı işlemi Windows' ta oturum açmak için kullandığınız kullanıcı adını kullanarak çalışır. Sistemde oturum kapattığınızda, aracı işlemi durur. Aracının durmasını önlemek için bir aracının Windows hizmeti olarak çalışacak şekilde yapılandırılmasını sağlar. Bir Windows hizmeti olarak çalıştırılması, Windows ortamı başlatıldığında ya da yeniden başlatıldığında otomatik olarak başlatılacak araçları yapılandırmanıza da olanak sağlar.

Windows hizmeti olarak çalışan bir aracıyı başlatmak için aşağıdaki adımları tamamlayın. Aracıyı bir Windows hizmeti olarak çalıştırmak için, desteklenen Windows sürümlerinden birinde Managed File Transfer komutunu çalıştırıyor olmanız gerekir. Desteklenen ortamların listesi için [IBM MQ için Sistem Gereksinimleri](#) başlıklı konuya bakın.

Tam adımlar, bir aracı oluşturup oluşturmadığınızı ya da bir aracı oluşturup oluşturmadığınızı size bağlıdır. Her iki seçenek de aşağıdaki adımlarda anlatılır.

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 aracı AGENT1 yaratılır. The Windows service runs using a user name of fteuser, which has an associated password ftepassword.

```
fteCreateAgent -agentName AGENT1 -agentQMGR QMGR1 -s -su fteuser -sp ftepassword
```

You can optionally specify a name for the service after the **-s** parameter. Bir ad belirtmezseniz, hizmet `mqmftAgentAGENTQMGR` adını taşır; burada `AGENT`, belirttiğiniz aracı adı ve `QMGR`, aracı kuyruk yöneticisi adıdır. Bu örnekte, hizmet için varsayılan ad `mqmftAgentAGENT1QMGR1`' dir.

Not: **-su** parametresini kullanarak belirttiğiniz Windows kullanıcı hesabı, **Log on as a service** haklarına sahip olmalıdır. Bunu nasıl yapılandırabilmeye ilişkin bilgi için bkz. [Guidance for running an MFT agent or logger as a Windows service](#).

Daha fazla bilgi için bkz. [fteCreateAgent](#), [fteCreateCDAgent](#): Create a Connect:Direct bridge Agent ya da [fteCreateBridgeAgent](#) (bir MFT iletişim kuralı köprüsü aracı oluşturun ve yapılandırın).

2. If you followed the previous step to create an agent, run the MQSC commands that are generated by the **fteCreateAgent**, **fteCreateCDAgent**, or **fteCreateBridgeAgent** command. Bu komutlar, aracının gereksinim duyduğu IBM MQ kuyruklarını oluşturur.
Örneğin, `AGENT1` adlı aracı, `QMGR1` adlı bir aracı kuyruk yöneticisi ve `COORDQMGR1` adlı bir eşgüdümleme kuyruk yöneticisi için aşağıdaki 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 aracıyı çalıştırdığınızda durdurun ve daha sonra, yapılandırmasını değiştirin.

a) Aşağıdaki örnek, `AGENT1` adlı bir aracıyı kullanır. 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 bakınız: [fteModifyAgent: run an MFT Agent as a Windows service](#).

4. Start your agent using the **fteStartAgent** command. Diğer bir seçenek olarak, hizmeti başlatmak için Windows masaüstü başlatma menüsünden seçilen Denetim Masası 'nda (Control Panel) Denetim Masası 'ndan (Administrative Tools) bulunan Windows Services aracını kullanabilirsiniz.

```
fteStartAgent AGENT1
```

Windows oturumunuz kapatılsa da hizmet çalışmaya devam eder. service' in sona erdirildikten sonra Windows yeniden başlatıldığında hizmetin de yeniden başlatılmasını sağlamak için, Windows Services aracındaki **Startup Type** (Başlatma Tipi) alanı varsayılan olarak **Automatic** (Otomatik) olarak ayarlanır. Windows yeniden başlatıldığında hizmetin yeniden başlatılmasını istemiyorsanız, bunu **El İle** 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 belirtmiş olup olmamasından bağımsız olarak, **-i** parametresini kullanarak çalıştırır. The **-i** parameter stops the agent immediately without completing any transfers that are in progress. Bunun nedeni, Windows hizmetinin bir sınırlamasıdır.

Sonraki adım

Windows hizmetinizi başlatmaya ilişkin sorunlarınız varsa, bkz. [Guidance for running an MFT agent or logger as a Windows service](#). Bu konuda ayrıca, Windows hizmet günlüğü dosyalarının yeri de açıklanır.

Linux

AIX

AIX and Linux sistemi başlatma sırasında bir MFT aracısının başlatılması

Bir Managed File Transfer Agent , AIX and Linux sisteminde sistem başlatma sırasında başlayacak şekilde yapılandırılabilir. Oturumu kapattığınızda, aracınız çalışmaya devam eder ve dosya aktarımlarını alabilir.

Bu Managed File Transfer komutlarından birini kullanarak bir aracı yarattıktan ve yapılandırdığınızda; **fteCreateAgent**, **fteCreateCDAgent**, ya da **fteCreateBridgeAgent** komutlarından birini kullanarak, aşağıdaki komutu yürüten bir komut dosyası kullanarak AIX and Linux ,reboot,rebootmachinesya da:makinelerinde otomatik olarak başlatılacak şekilde yapılandırabilirsiniz.

```
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

Burada *mq_install_root* , gerekli Managed File Transfer kuruluşunun kök dizinidir; varsayılan değer şudur: /opt/mqm ve *agent_name* , başlatılacak Managed File Transfer Agent adıdır. Bu komut dosyası kullanımı, işletim sistemine bağlı olarak değişir. Örneğin, Linux altında ek seçenekler vardır.

Linux

Linux

Linux sistemleri için, sistem önyükleme işlemi sırasında uygulamaları başlatabileceğiniz birden çok yol vardır. Genel olarak, aşağıdaki adımları dikkate alın:

1. İçindekiler ile /etc/rc.mqmft adlı 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. Dosya yürütülebilir dosyasını yapın, örneğin:

```
chmod 755 /etc/rc.mqmft
```

3. Sonraki satırı `/etc/inittab`' a ekleyin:

```
mqmft:5:boot:/etc/rc.mqmft
```

Linux 'ta önyükleme sırasında bir aracıyı başlatmanın diğer yolları arasında, komut dosyası satırlarının `/etc/rc.d/rc.local` dosyasına eklenmesi ya da Linux SuSe' de, komut dosyası satırlarının `/etc/init.d/boot.local` dosyasına eklenmesi yer alır. Ortamınız için en iyi şekilde çalışan yöntemi seçmeniz gerekir. Desteklenen belirli Linux dağıtımlarında başlatma sırasında bir aracıyı başlatmanın diğer yolları hakkında daha fazla bilgi için aşağıda yer alan bilgiler şunlardır:

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. Aşağıdaki satırları `rc.rclocal` dosyasına 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
```

Starting Managed File Transfer agents on Linux with systemd

Linux

Aşağıdaki yordamı gerçekleştirin:

1. `/etc/systemd/` sistem klasöründe bir dosya oluşturun ve bunu adı (örneğin, `<agentname>.service`) yazın. Aşağıdaki içeriği ekleyin; burada `<agentname>` , `MFT_AGT_LNX_0` olur.

```
# 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. Hizmetin geçerli kılınmasını sağlamak 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
```

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

LTS

z/OS üzerindeki bir MFT aracı, aşağıdaki senaryolarda istemci bağlantısı kullanarak z/OS üzerindeki bir kuyruk yöneticisine bağlanabilir:

- MFT aracı, APAR PH56722 uygulanmış olarak IBM MQ 9.2.0 Uzun Süreli Destek 'dedir ve IBM MQ Advanced for z/OS Value Unit Editionya da IBM MQ Advanced for z/OS ürün tanıtıcısıyla (PID) ilişkilendirilmiştir.
- MFT aracı IBM MQ 9.2.0 adresinde yer almakta ve IBM MQ Advanced for z/OS Value Unit Edition PID 'si ile ilişkilendirilmiştir

IBM MQ ürünlerinin ayrıntıları, ilişkili PID değerleri ve dışa aktarma sınıflandırmaları için bkz. [IBM MQ ürün tanıtıcıları ve dışa aktarma bilgileri](#) .

Bir MFT kuruluşuyla ilişkili PID ' nin nasıl ayarlanacağına ilişkin bilgi için bkz. **fteSetProductID** .

Aracının altında çalıştığı PID, aracı başlatılırken günlükte görüntülenir.

Not: Başka bir PID altında çalışan z/OS üzerindeki bir MFT aracı, yalnızca bağ tanımlama kipi bağlantısı kullanarak 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 aracısını durdurma” sayfa 198

JCL ' den başlatılan bir görev olarak z/OS üzerinde bir Managed File Transfer Agent çalıştırıyorsanız, aracı **fteStopAgent** komutuna ek olarak **MODIFY** ve **STOP** z/OS işlecini de kabul eder.

[MFT agent.properties dosyası](#)

MFT araçlarını listeleme

Belirli bir kuyruk yöneticisiyle kayıtlı Managed File Transfer araçlarını, komut satırını ya da IBM MQ Explorer komutunu kullanarak listeleyebilirsiniz.

Bu görev hakkında

Aracıları komut satırını kullanarak listelemek için bkz. [fteListAgents](#) komutu.

To list agents using the IBM MQ Explorer, in the Navigator view click **Aracılar** under the coordination queue manager name.

Bir aracı **fteListAgents** komutu tarafından listelenmiyorsa ya da IBM MQ Explorer içinde görüntülenmiyorsa, sorunu bulmak ve düzeltmek için aşağıdaki konuyla ilgili tanımlama akış şemasını kullanın: [MFT aracınız fteListAgents komutu tarafından listelenmiyorsa ne yapmanız gerekir?](#)

MFT aracısını durdurma

Bir Managed File Transfer aracısını komut satırından durdurabilirsiniz. Bir aracıyı durdurduğunuzda, aracıyı durdurup durdurulurken, durdurulmadan önce aracının yürürlükteki dosya aktarımını tamamlanmasına izin veriniz. Bir aracıyı hemen durdurmak için komut satırındaki **-i** parametresini de belirtebilirsiniz. Aracı durdurulduğunda, o aracıyı yeniden başlatıncaya kadar dosya aktarmak için kullanamazsınız.

Başlamadan önce

Bir kuyruk yöneticisiyle ilişkilendirilmiş araçların adlarını denetlemek istiyorsanız, araçları IBM MQ Explorer ya da komut satırını kullanarak listeleyebilir, bkz. [fteListAgents](#) komutu.

Bu görev hakkında

Bir aracıyı komut satırından durdurmak için [fteStopAracısı](#)' na bakın.

Aracıyı Windows hizmeti olarak çalışacak şekilde yapılandırdıysanız, **fteStopAgent** komutunu çalıştırarak Windows hizmetini de durdurursun. Diğer bir seçenek olarak, hizmeti Windows Services aracını kullanarak durdurarak aracıyı durdurabilirsiniz. Daha fazla bilgi için [“Windows aracı olarak MFT aracısının başlatılması” sayfa 193](#) başlıklı konuya bakın.

z/OS z/OS üzerinde bir MFT aracısını durdurma

JCL ' den başlatılan bir görev olarak z/OS üzerinde bir Managed File Transfer Agent çalıştırıyorsanız, aracı **fteStopAgent** komutuna ek olarak **MODIFY** ve **STOP** z/OS işlecini de kabul eder.

Başlatılan bir görev, belirli bir kullanıcı kimliği altında çalıştığı ve kullanıcıların oturum kapatmalarından etkilenmediği için kullanılır.

Not: Başlatılan görevler tipik olarak, oturum açma ayrıcalıklarına sahip olmayan bir denetimci kullanıcı altında çalıştırılır ve bu nedenle, aracının altında çalıştığı kullanıcı olarak z/OS sisteminde oturum

açılmaz. The **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, cannot be issued for that agent.

From IBM MQ 9.0.2 and IBM MQ 9.0.0 Fix Pack 1, you can use the agent property **adminGroup** with Managed File Transfer agents on z/OS. Bir güvenlik yöneticisi grubu tanımlayabilirsiniz; örneğin, MFTADMIN ve daha sonra, başlatılan görev kullanıcı kimliğini ve denetimci TSO kimliklerini bu gruba ekleyin. Aracı özellikleri dosyasını düzenleyin ve **adminGroup** özelliğini, bu güvenlik yöneticisi grubunun adı olacak şekilde ayarlayın.

```
adminGroup=MFTADMIN
```

Members of this group can then issue the **fteStartAgent**, **fteStopAgent**, and **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, for the agent that is running as a started task.

Daha fazla bilgi için, [MFT agent.properties dosyası](#) içindeki **adminGroup** özelliğine bakın.

z/OS MODIFY komutu (F) kullanılarak denetlenen aracı sona erdirmesi

The **MODIFY** command allows you to stop an agent in a controlled way as an alternative to the **fteStopAgent** command. Aracı şu anda devam etmekte olan aktarımları tamamlar, ancak aracı yeni bir aktarma başlatmaz.

Örneğin:

```
F job_name,APPL=STOP
```

Burada *iş_adi* , aracı işleminin çalışmakta olduğu işidir.

Immediate agent shutdown by using the z/OS STOP command (P)

STOP komutu, **-i** parametresiyle **fteStopAgent** komutunu kullanarak hemen bir duraklamaya eşdeğerdır. Aracı şu anda bir dosyayı aktarmakta olsa bile aracı durdurulur.

Örneğin:

```
P job_name
```

Burada *iş_adi* , aracı işleminin çalışmakta olduğu işidir.

İlgili başvurular

“z/OS üzerinde bir MFT aracısının başlatılması” sayfa 197

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.

The [MFT agent.properties file](#)

Yeni bir dosya aktarımı başlatılıyor

IBM MQ Explorer ' tan ya da komut satırından yeni bir dosya aktarımı başlatabilir ve bir grupta tek bir dosya ya da birden çok dosya aktarmayı seçebilirsiniz.

Bu görev hakkında

Komut satırından yeni bir kütük aktarımı başlatmak için [fteCreateTransfer command](#) başlıklı konuya bakın.

IBM MQ Explorer' ta **Yeni Yönetilen Dosya Aktarması Yarat** sihirbazını kullanarak yeni bir dosya aktarımı başlatmak için aşağıdaki adımları kullanın:

Yordam

1. Navigator görünümünde, **Yönetilen Dosya Aktarması**' yı tıklatın. **Yönetilen Dosya Aktarma Merkezi** , İçerik görünümünde görüntülenir.
2. Tüm koordinasyon kuyruğu yöneticilerinizin tümü Navigator görünümünde görüntülenir. Aktarım için kullanmak istediğiniz aracının kayıtlı olduğu eşgüdümleme kuyruk yöneticisinin adını genişletin. Aktarma için kullanmak istediğiniz bir eşgüdümleme kuyruk yöneticisine bağlıysa, Navigator görünümündeki eşgüdüm kuyruk yöneticisi adını sağ tıklatın ve **Bağlantıyı Kesdüğmesini** tıklatın. Daha sonra, kullanmak istediğiniz koordinasyon kuyruğu yöneticisinin adını sağ tıklatın ve **Bağlan**' ı tıklatın.
3. Aşağıdaki yöntemlerden birini kullanarak **Create New Managed File Transfer** (Yeni Yönetilen Dosya Aktarması Yarat) sihirbazını başlatın:
 - a) Navigator görünümünde, aşağıdaki düğümlerden herhangi birinin adını sağ tıklatın: ilgili koordinasyon kuyruğu yöneticisi, **Aktarma Şablonları**, **Aktarma Günlüğü** ya da **Bekleyen Aktarımlar**. Daha sonra, sihirbazı başlatmak için **Yeni Aktarım** düğmesini tıklatın.
 - b) **Dosya > Yeni > Diğer > Yönetilen Dosya Aktarım Sihirbazları > Yeni Aktarım Sihirbazı** öğelerini tıklatın.
4. Sihirbaz panolarındaki yönergeleri izleyin. Her bir pano için bağlama duyarlı yardım da vardır. Windows' ta bağlama duyarlı yardıma erişmek için F1tuşuna basın. Linux' ta Ctrl+F1 ya da Shift+F1tuşuna basın.

Aktarma tanımlaması dosyalarının kullanılması

Dosya aktarma yaratmak için kullanılacak bir aktarma tanımlaması dosyası belirleyebilirsiniz. Aktarma tanımlaması dosyası, aktarma yaratmak için gereken bilgilerin bazılarını ya da tümünü tanımlayan bir XML dosyasıdır.

Aktarma tanımlaması dosyaları, tek bir aktarma işleminde birden çok kaynak dosyası ve birden çok hedef dosya belirlemek istediğinizde kullanışlıdır. Karmaşık bir dosya aktarımı sunmak için bir aktarma tanımlaması dosyası kullanabilirsiniz. Aktarma tanımlama dosyasını yeniden kullanılabilir ve paylaşabilirsiniz.

Bir aktarma tanımlaması dosyası için iki biçim kullanabilirsiniz ve bu biçimler biraz farklılık gösterirken, her ikisi de FileTransfer.xsd şemasına uygun olur. Bu şemayı Managed File Transfer kurulumunun samples\schema dizininde bulabilirsiniz.

Aktarma tanımlaması kütüklerinin aşağıdaki iki biçimi desteklenir:

- Bir aktarıma ilişkin kaynak ve hedef dosyaların tanımlaması. Bu tanım kök olarak bir **transferSpecifications** ögesi kullanır.
- Kaynak ve hedef dosyalar, kaynak ve hedef araçlar da içinde olmak üzere, tüm aktarmanın tanımı. Bu tanım kök olarak bir **request** ögesi kullanır.
 - Bu biçimi içeren dosyalar, **-gt** parametresi kullanılarak **fteCreateTransfer** komutundan oluşturulabilir.

Aşağıdaki örnekte, aktarıma ilişkin yalnızca kaynak ve hedef dosyaları belirten bir aktarma tanımlaması dosyası biçimi gösterilmektedir:

```
<?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>
```


Bu aktarma tanımlaması dosyası biçimini göndermek 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, aktarma işlemi için gereken tüm bilgileri belirten bir aktarma tanımlaması 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 komutundaki **-gt** parametresini kullanarak bu biçimi içeren bir dosya oluşturabilirsiniz. Bu biçim ile bir aktarma tanımlaması dosyası gönderdiğinizde, komut satırında başka bir şey belirtmenize gerek yoktur:

```
fteCreateTransfer -td c:\definitions\example2.xml
```

Komut satırı ile ilgili kaynak ve hedef aracı bilgilerini, aktarma tanımlaması dosyasına ek olarak normal değiştirmelerde geçirebilirsiniz. Örneğin:

```
fteCreateTransfer -da AGENT9 -dm agent9qm -td c:\definitions\example2.xml
```

This example uses the command-line options to override the destination agent defined inside the transfer definition file with **AGENT9** and the destination queue manager defined in the transfer definition file as **agent9qm**.

Tanımlanan biçimlerin her ikisi de bir ya da daha fazla < item > ögesi içerebilir. < öge > ögesi hakkında daha fazla bilgi için bkz. [Dosya aktarma isteği ileti biçimi](#). Bu aktarma öğelerinden her biri, aktarma davranışını denetleyebilmek için ek öznitelikleri olan bir kaynak ve hedef dosya çiftini tanımlar. Örneğin, aşağıdaki davranışı belirtebilirsiniz:

- Aktarmanın sağlama toplamı kullanıp kullanmadığını
- Aktarmanın metin mi, yoksa ikili mi olduğunu
- Aktarmanın tamamlanmasından sonra kaynak dosyanın silinip silinmeyeceği
- Dosya varsa, hedef dosyanın üzerine yazılıp yazılmayacağını

Aktarma tanımlama kütüklerinin kullanılmasının bir avantajı da, komut satırından kullanılacak ek seçenekleri belirleyebilmenizi sağlar. Örneğin, iletiyi dosya aktarma işlemleri gerçekleştirirken, bir aktarma tanımlaması dosyası kullanarak groupId özniteliğini belirtebilirsiniz. Bu öznitelik, kuyruktan okunan iletilerin IBM MQ grup tanıtıcısını belirtir. Aktarma tanımlaması dosyalarının başka bir avantajı da, her dosya çifti için farklı seçenekler belirleyebilmenizi sağlar. Örneğin, bir sağlama toplamının kullanılıp kullanılmadığını ya da dosyanın metin ya da ikili kipte mi aktarılacağını, dosya temelinde mi

belirleyebilirsiniz. Komut satırını kullanırsanız, bir aktarımda bulunan 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 bir z/OS sistemine dosya aktarmak için aşağıdaki 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, kaynak araçından textTransferTest.txt dosyasını, metin kipindeki hedef araçtaki //TEXT.TRANS.TEST veri kümesine aktarır. Bu aktarım, kaynak verileri kaynak aracının varsayılan kodlamasından (kaynak kodlama özniteliği belirtilmez) kod sayfasına dönüştürür: IBM-1047.

Zamanlanmış bir dosya aktarma işlemi yaratılması

IBM MQ Explorer¹ den ya da komut satırından yeni bir dosya aktarımı zamanlayabilirsiniz. Zamanlanmış aktarma, bir grupta tek ya da birden çok dosya içerebilir. Zamanlanan bir dosya aktarımı işlemi bir kez gerçekleştirebilir ya da aktarma işlemi birden çok kez yineleyebilirsiniz.

Bu görev hakkında

Bir dosya aktarma zamanlamasını bir kez oluşturmak ya da aşağıdaki aralıklarla gerçekleşecek şekilde ayarlayabilirsiniz:

- Her dakika
- Saatlik
- Günde bir
- Haftada bir
- Ayda bir

- Yıllık

Daha sonra, aşağıdaki noktalarda durdurulacak geçişleri belirtebilirsiniz:

- Tanımlı bir saat ve tarihte
- Tanımlı bir oluşum sayısından sonra

Diğer bir seçenek olarak, geçişlerin sonsuza kadar devam ettiğini de belirtebilirsiniz.

Zamanlanmış bir aktarma her gün aynı saatte çalıştırılırsa, zaman çizelgesinin saat değişiminde gerçekleşeceği zamanı ayarlamak için aracı özellikleri dosyasındaki **adjustScheduleTimeForDaylightSaving** özneteliğini kullanın. Daha fazla bilgi için bkz. [MFT agent.properties](#) dosyası.

To create a new scheduled file transfer using the command line, use the scheduling parameters (**-tb**, **-ss**, **-oi**, **-of**, **-oc**, and **-es**) for the `fteCreateAktarma` komutu.

IBM MQ Explorer' ta **Yeni Yönetilen Dosya Aktarması Yarat** sihirbazını kullanarak zamanlanmış yeni bir dosya aktarımı yaratmak için aşağıdaki adımları kullanın:

Yordam

1. Navigator görünümünde, **Yönetilen Dosya Aktarması**' yı tıklatın. **Yönetilen Dosya Aktarma Merkezi** , İçerik görünümünde görüntülenir.
2. Tüm koordinasyon kuyruğu yöneticilerinizin tümü Navigator görünümünde görüntülenir. Aktarım için kullanmak istediğiniz aracının kayıtlı olduğu eşgüdümleme kuyruk yöneticisinin adını genişletin. Aktarma için kullanmak istediğiniz bir eşgüdümleme kuyruk yöneticisine bağlıysa, Navigator görünümündeki eşgüdüm kuyruk yöneticisi adını sağ tıklatın ve **Bağlantıyı Kesdüğmesini** tıklatın. Daha sonra, kullanmak istediğiniz koordinasyon kuyruğu yöneticisinin adını sağ tıklatın ve **Bağlan**' ı tıklatın.
3. Aşağıdaki yöntemlerden birini kullanarak **Create New Managed File Transfer** (Yeni Yönetilen Dosya Aktarması Yarat) sihirbazını başlatın:
 - a) Navigator görünümünde, aşağıdaki düğümlerden herhangi birinin adını sağ tıklatın: ilgili koordinasyon kuyruğu yöneticisi, **Aktarma Şablonları**, **Aktarma Günlüğü** ya da **Bekleyen Aktarımlar**. Daha sonra, sihirbazı başlatmak için **Yeni Aktarım** düğmesini tıklatın.
 - b) **Dosya > Yeni > Diğer > Yönetilen Dosya Aktarım Sihirbazları > Yeni Aktarım Sihirbazı** öğelerini tıklatın.
4. Sihirbaz panolarındaki yönergeleri izleyin. Ensure that you select the **Zamanlama aktarımı etkinleştir** check box and enter your schedule details on the **Zamanla** tab. Zamanlanan dosya aktarımları, aktarma işlemi etkileyebilecek herhangi bir sorun yoksa, zamanlamanın başlangıç saatinden bir dakika içinde başlar. Örneğin, ağınlı ya da aracınızla zamanlanan aktarma başlatılmasını önleyen sorunlar olabilir. Her bir pano için bağlama duyarlı yardım sağlanır. Windows' ta bağlama duyarlı yardıma erişmek için F1tuşuna basın. Linux' ta Ctrl+F1 ya da Shift+F1tuşuna basın.

Sonuçlar

Zamanlanmış dosya aktarımlarıyla ilgili iletilere ilişkin bilgi için [Zamanlanmış dosya aktarma günlüğü ileti biçimleri](#) başlıklı konuya bakın.

Bekleyen dosya aktarımlarıyla çalışma

IBM MQ Explorer' den beklemekte olan zamanlanmış dosya aktarımlarını görüntüleyebilirsiniz. **Beklemedeki Aktarmalar** penceresinde, bağlı olduğunuz eşgüdüm kuyruk yöneticisiyle kayıtlı olan tüm beklemedeki aktarımlar görüntülenir.


Bu görev hakkında

Henüz başlamamış olan zamanlanmış bir dosya aktarımının durumunu görüntülemek için aşağıdaki adımları kullanın:

Yordam

1. Navigator görünümünde **Yönetilen Dosya Aktarması** nesnesini açın. **Yönetilen Dosya Aktarma Merkezi** , İçerik görünümünde görüntülenir.
2. Tüm eşgüdüm kuyruk yöneticilerinizin tümü Navigator görünümünde görüntülenir. Zamanlanan aktarım için kullandığınız eşgüdüm kuyruk yöneticisinin adını açın. Bağlandığınız eşgüdüm kuyruk yöneticisini değiştirmek istiyorsanız, Navigator görünümünde kullanmak istediğiniz koordinasyon kuyruğu yöneticisinin adını sağ tıklayın ve **Bağlan'** ı tıklayın.
3. **Beklemedeki Aktarımlar'** ı tıklayın. İçerik görünümünde **Bekleyen Aktarımlar** penceresi görüntülenir.
4. **Beklemedeki Aktarmalar** penceresinde, zamanlanmış dosya aktarımlarınızla ilgili aşağıdaki ayrıntılar görüntülenir:
 - a) **Ad** Zamanlanan dosya aktarımının numarası. Bu numara otomatik olarak atanır.
 - b) **Kaynak** Kaynak aracıya ilişkin ad.
 - c) **Kaynak Dosya** 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 aktarımında, denetimcinin seçilen saat diliminde başlatılması için zamanlanan saat ve tarih. Görüntülenen saat dilimini değiştirmek için, **Pencere > Tercihler > IBM MQ Explorer > Yönetilen Dosya Aktarması** seçeneklerini tıklayın ve **Saat dilimi:** listesinden bir alternatif saat dilimi seçin. **Tamam'** ı tıklayın.
 - g) **Yinele Her** Zamanlanan aktarımı, aktarmayı yinelemek istediğiniz belirtilen aralığı, bir sayı olarak ifade edilen aralık.
 - 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) **Yineleme Tarihi:** Zamanlanmış aktarımı yinelemeyi seçtiyseniz, yinelenen dosya aktarımının durmasını istediğiniz zaman ayrıntılarıyla ilgili ayrıntıları belirleyin. Örneğin, belirli bir tarih ve saat ya da belirli bir oluşum sayısından sonra.

Sonuçlar

Beklemedeki Aktarımlar penceresinde görüntülenenleri yenilemek için, İçerik görünümü araç çubuğundaki Yenile düğmesini  tıklayın.

Bekleyen bir dosya aktarımı iptal etmek için, aktarma işlemi farenin sağ düğmesiyle tıklayın ve **İptal** düğmesini tıklayın. Bir aktarmanın iptal edilmesi, dosya aktarma isteğini tamamen atar.

Dosya aktarımlarının tetiklenmesi

Aktarma işlemi gerçekleşmeden önce, doğru olması gereken bazı tetikleme koşullarını bir kütük aktarımı için ayarlayabilirsiniz. Tetikleme koşulları doğru değilse, dosya aktarımı gerçekleşmez ve bir günlük iletisi isteğe bağlı olarak, aktarma işlemi gerçekleştirilmesi için gönderilir. Daha sonra, dosya aktarma isteği atılır. Örneğin, yalnızca kaynak aracısının bulunduğu sistemdeki adlandırılmış bir dosyanın belirli bir büyüklüğün üzerinde olması ya da kaynak aracının bulunduğu sistemde belirli bir adlandırılmış dosya varsa, bir dosya aktarma işlemi ayarlayabilirsiniz. You can set up a triggered file transfer from either the IBM MQ Explorer or from the command line.

Bu görev hakkında

Bir tetikleme koşulunun karşılanabilmesi için bir kaynağı sürekli olarak izleyebilirsiniz. Kaynak izleme ile ilgili daha fazla bilgi için bkz. [“MFT kaynaklarının izlenmesi” sayfa 210.](#)

Ayarlayabileceğiniz üç farklı tetikleyici koşul var. 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 boyuttan geçiyorsa (boyut, bayt, KB, MB ya da GB cinsinden ifade edilebilir). Bu ölçü birimleri 2^{10} kuralını kullanır; örneğin, 1 KB, 1024 bayta eşittir ve 1 MB, 1024 KB 'ye eşittir.

Yukarıdaki listede tetikleme tipleri iki şekilde birleştirilebilir:

- Tek bir koşulla, kaynak aracının bulunduğu sistemde birden çok dosya belirleyebilirsiniz. Bu, belirtilen dosyalardan herhangi birinin koşulu karşıladığında (Boole işleci OR) aktarımı tetikler.
- Birden çok koşul belirtebilirsiniz. Bu, aktarma işlemini yalnızca tüm koşullar karşılanırsa (Boole işleci AND) tetikler.

Tetiklenen bir aktarımı zamanlanmış bir aktarımla da birleştirebilirsiniz. Ek bilgi için [Zamanlanmış dosya aktarma yaratılması](#) başlıklı konuya bakın. Bu durumda, tetikleme koşulları, zamanlamanın başlatılacağı sırada ya da zamanlamanın her başlatıldığı zaman çizelgesi için bir yineleme zamanlamasında değerlendirilir.

Tetiklenen aktarımlar, protokol köprüsü araçlarında desteklenmez.

Komut satırını kullanarak tetiklenen bir dosya aktarımı yaratmak için, [fteCreateTransfer](#) komutundaki **-tr** parametresini kullanın.

IBM MQ Explorer' ta **Yeni Yönetilen Dosya Aktarması Yarat** sihirbazını kullanarak zamanlanmış bir dosya aktarımı yaratmak için aşağıdaki adımları kullanın:

Yordam

1. Navigator görünümünde, **Yönetilen Dosya Aktarması**' yı tıklatın. **Yönetilen Dosya Aktarma Merkezi** , İçerik görünümünde görüntülenir.
2. Tüm koordinasyon kuyruğu yöneticilerinizin tümü Navigator görünümünde görüntülenir. Zamanlanan aktarım için kullandığınız eşgüdüm kuyruk yöneticisinin adını açın. Bağlandığınız eşgüdüm kuyruk yöneticisini değiştirmek istiyorsanız, Navigator görünümünde kullanmak istediğiniz koordinasyon kuyruğu yöneticisinin adını sağ tıklatın ve **Bağlan** ' ı tıklatın.
3. Aşağıdaki yöntemlerden birini kullanarak **Create New Managed File Transfer** (Yeni Yönetilen Dosya Aktarması Yarat) sihirbazını başlatın:
 - a) Navigator görünümünde, aşağıdaki düğümlerden herhangi birinin adını sağ tıklatın: ilgili koordinasyon kuyruğu yöneticisi, **Aktarma Şablonları**, **Aktarma Günlüğü** ya da **Bekleyen Aktarımlar**. Daha sonra, sihirbazı açmak için **Yeni Aktarım** ' ı tıklatın.
 - b) **Dosya > Yeni > Diğer > Yönetilen Dosya Aktarım Sihirbazları > Yeni Aktarım Sihirbazı** öğelerini tıklatın.
4. Sihirbaz panolarındaki yönergeleri izleyin. **Tetikleyiciler** etiketindeki **Tetiklenen aktarımı etkinleştir** onay kutusunu işaretlediğinizden emin olun ve tetikleme ayarlamak için bu sekmede bulunan alanları doldurun. Her bir pano için bağlama duyarlı yardım sağlanır. Windows' ta bağlama duyarlı yardıma erişmek için F1tuşuna basın. Linux' ta **Ctrl+F1** ya da **Shift+F1**tuşlarına basın.


Devam etmekte olan dosya aktarımlarını izleme

Devam etmekte olan bir dosya aktarımında, IBM MQ Explorerindeki **Yönetilen Dosya Aktarması- Yürürlükteki Aktarma İlerleme Durumu** sekmesini kullanarak izleyebilirsiniz. Bu dosya aktarma işlemi IBM MQ Explorer ya da komut satırından başlatılabilir. Bu sekme, zamanlanan aktarımların başladığı noktadaki zamanlanan aktarımların ilerleyişini de görüntüler.

Bu görev hakkında

Uzak bir sistemdeki bir koordinasyon kuyruk yöneticisiyle ilişkili aktarımları izlemek için IBM MQ Explorer ' i kullanmak istiyorsanız, [“Configuring IBM MQ Explorer to monitor a remote coordination queue manager” sayfa 207](#) konusundaki yönergeleri izleyin.

Previous file transfer information is not retained after you stop and restart IBM MQ Explorer. Yeniden başlatma sırasında, geçmiş aktarımlara ilişkin bilgiler **Geçerli Aktarım Aşaması** sekmesinden temizlenir.


IBM MQ Explorer ' un açık olduğu herhangi bir noktada **Tamamlanan aktarımları kaldır**  seçeneğini kullanarak tamamlanan aktarımları temizleyebilirsiniz.


Yordam

After you have started a new file transfer using IBM MQ Explorer or the command line, you can monitor the progress of your transfer in the **Yürürlükteki Aktarım Aşaması** tab. Devam etmekte olan her aktarma işlemi için aşağıdaki bilgiler görüntülenir:

- Kaynak.** Dosyayı kaynak sistemden aktarmak için kullanılan aracının adı.
- Hedef.** Hedef sistemde dosyayı almak için kullanılan aracının adı.
- Yürürlükteki dosya.** Aktarılmakta olan dosyanın adı. Önceden aktarılmış olan tek tek dosyanın parçası B, KiB, MiB' de görüntülenir. GiBya da TiB ile birlikte, dosyanın toplam boyutu parantez içinde yer alan bir dosya boyudur. Görüntülenen ölçüm birimi, dosyanın boyutuna bağlıdır.
B, saniye başına byte 'tır. KiB/s saniye başına kibibayttır; burada 1 kibibayt 1024 bayta eşittir. MiB/s, 1 mebibayt 'ın 1 048 576 bayta eşit olduğu saniye başına mebibayttır. GiB/s, 1 gibibyte 'ın 1 073 741 824 bayta eşit olduğu saniye başına gibibayttır. TiB/s, 1 tebibayt 'ın 1 099 511 627 776 bayta eşit olduğu saniye başına tebibayttır.
- Dosya numarası.** Birden fazla dosya aktarıyorsanız, bu sayı, aktarımın toplam grup grubu tarafından ne kadar uzaklarda olduğunu gösterir.
- İlerleme Durumu.** İlerleme çubuğu, geçerli dosya aktarımının tamamlanma yüzdesi olarak nasıl tamamlandığını gösterir.
- Oran.** Dosyanın KiB/s cinsinden aktarılma hızı (kilobayt/saniye); burada 1 kibibayt 1024 bayta eşittir.)
- Başlatıldı (seçilen saat dilimi).** Dosya aktarma işlemi, denetimcinin seçilen saat diliminde sunulma saati. Görüntülenen saat dilimini değiştirmek için, **Pencere > Tercihler > IBM MQ Explorer > Yönetilen Dosya Aktarması** seçeneklerini tıklatın ve **Saat dilimi:** listesinden bir alternatif saat dilimi seçin. **Tamam'**ı tıklatın.
Aktarma işlemi dosyayı aktarırken bir kurtarma durumuna girerse, başlatılan zaman, dosya aktarımına devam edilen süreyi yansıtacak şekilde güncellenir.

Sonuçlar

Bu sekme, düzenli olarak bilgilerini otomatik olarak yeniler, ancak **Yürürlükteki Aktarım Aşaması** sekmesinde görüntülenenin yenilenmiş bir görünümünü zorlamak için İçerik görünümü araç çubuğunda **Yenile**  simgesini tıklatın.

Dosya aktarımlarını **Yürürlükteki Aktarım Aşaması** sekmesinden silmek için İçerik görünümü araç çubuğundaki **Tamamlanan aktarımları kaldır**  simgesini tıklatın. Bu düğmenin tıklatılması, dosya aktarma ayrıntılarını yalnızca etiketten kaldırır; yürürlükteki ya da zamanlanmış bir aktarımı durdurmaz ya da iptal etmez.

Kapattıktan sonra **Yürürlükteki Aktarım Aşaması** sekmesine geri dönmek isterseniz, **Pencere > Görünümü Göster > Diğer > Diğer > Yönetilen Dosya Aktarması-Yürürlükteki Aktarma İlerlemesi** simgesini tıklatarak sekmeyi görüntüleyebilirsiniz. **Tamam'**ı tıklatın.

Sonraki adım

Ayrıca, özel dosya aktarımı izleme için uygulamalar geliştirmek de mümkündür. This can be accomplished by creating a subscription to the appropriate Managed File Transfer administrative topic (either programmatically or administratively), and the monitor application can then receive Managed File Transfer file transfer activity publications on the topic. Abonelik konusu ve yayın iletisi biçimiyle ilgili daha fazla bilgi için [Dosya aktarma ilerleme durumu iletisi örnekler](#) başlıklı konuya bakın.

İlgili görevler

[“Configuring IBM MQ Explorer to monitor a remote coordination queue manager” sayfa 207](#)

Uzak bir sistemde çalışan bir koordinasyon kuyruk yöneticisiyle ilişkilendirilmiş dosya aktarımlarını izlemek için IBM MQ Explorer ' i kullanın. IBM WebSphere MQ 7.5'ta ya da daha sonra, IBM MQ Explorer' yi çalıştırabilecek bir sisteme gerek duyarsınız. Uzak koordinasyon kuyruğu yöneticisine bağlanabilmek için IBM MQ Explorer bileşeninin kurulması gerekir.

[“Aktarma Günlüğü 'nde dosya aktarımlarının durumunu görüntüleme” sayfa 208](#)

Dosya aktarımlarının ayrıntılarını IBM MQ Exploreriçindeki **Aktarma Günlüğü** ' ne kullanarak görüntüleyebilirsiniz. Bu aktarımlar, komut satırından ya da IBM MQ Explorerkomutundan başlatılmış olabilir. Ayrıca, **Aktarma Günlüğü** ' ta görüntülenenleri özelleştirebilirsiniz.

Configuring IBM MQ Explorer to monitor a remote coordination queue manager

Uzak bir sistemde çalışan bir koordinasyon kuyruk yöneticisiyle ilişkilendirilmiş dosya aktarımlarını izlemek için IBM MQ Explorer ' i kullanın. IBM WebSphere MQ 7.5'ta ya da daha sonra, IBM MQ Explorer' yi çalıştırabilecek bir sisteme gerek duyarsınız. Uzak koordinasyon kuyruğu yöneticisine bağlanabilmek için IBM MQ Explorer bileşeninin kurulması gerekir.

Bu görev hakkında

Varsayımlar: Uzak bağlantılar için izin vermek üzere kuyruk yöneticisini yapılandırarak, uzak koordinasyon kuyruğu yöneticisine bağlanma yetkisi.

Bunun nasıl yapılandırılacağı hakkında daha fazla bilgi için [Kanal kimlik doğrulamasıyla istemci kipinde bir kuyruk yöneticisine bağlanma ve MFT' e özgü kaynaklara ilişkin yetkilerin yönetilmesi](#) başlıklı konuya bakın.

Windows ya da Linuxişletim sistemini çalıştırmayan bir sistemdeki kuyruk yöneticilerini ve dosya aktarımlarını izlemek için, IBM MQ Explorer ' u uzak sisteme bağlanmak için aşağıdaki adımları yapılandırın:

Yordam

1. Yerel IBM MQ Explorer' i başlatın.
2. IBM MQ Explorer yüklendiğinde, **Yönetilen Dosya Aktarması** klasörünü farenin sağ düğmesiyle tıklatın ve **Yeni yapılanış** öğesini seçin.
3. Sihirbazda ilerleyin, Coordination and Commands kuyruk yöneticisini seçin ve yapılanış için bir ad tanımlayın.
4. Tanımlamayı tamamlamak için **finish** (Son) düğmesini tıklatın.
5. Tanım sona erdiğinde, tanımlamayı farenin sağ düğmesiyle tıklatın ve **Bağlan** öğesini seçin.

Sonuçlar

Şimdi IBM MQ Explorer ' u başlatın ve eşgüdüm kuyruk yöneticisiyle ilişkilendirilmiş Managed File Transfer ağına ilişkin aktarma etkinliğini izlemek için bu programı kullanın.

İlgili görevler

[“Devam etmekte olan dosya aktarımlarını izleme” sayfa 205](#)

Devam etmekte olan bir dosya aktarımında, IBM MQ Exploreriçindeki **Yönetilen Dosya Aktarması- Yürürlükteki Aktarma İlerleme Durumu** sekmesini kullanarak izleyebilirsiniz. Bu dosya aktarma işlemi IBM MQ Explorer ya da komut satırından başlatılabilir. Bu sekme, zamanlanan aktarımların başladığı noktadaki zamanlanan aktarımların ilerleyişini de görüntüler.


[“Aktarma Günlüğü 'nde dosya aktarımlarının durumunu görüntüleme” sayfa 208](#)

Dosya aktarımlarının ayrıntılarını IBM MQ Exploreriçindeki **Aktarma Günlüğü** ' ne kullanarak görüntüleyebilirsiniz. Bu aktarımlar, komut satırından ya da IBM MQ Explorerkomutundan başlatılmış olabilir. Ayrıca, **Aktarma Günlüğü** ' ta görüntülenenleri özelleştirebilirsiniz.

Aktarma Günlüğü 'nde dosya aktarımlarının durumunu görüntüleme

Dosya aktarımlarının ayrıntılarını IBM MQ Exploreriçindeki **Aktarma Günlüğü** ' ne kullanarak görüntüleyebilirsiniz. Bu aktarımlar, komut satırından ya da IBM MQ Explorerkomutundan başlatılmış olabilir. Ayrıca, **Aktarma Günlüğü** ' ta görüntülenenleri özelleştirebilirsiniz.

Yordam



1. Navigator görünümünde **Yönetilen Dosya Aktarması** nesnesini açın ve aktarma günlüğünü görüntülemek istediğiniz koordinasyon kuyruk yöneticisinin adını genişletin.
2. Navigator görünümünde **Aktarım Günlüğü** ' ne tıklayın. The **Aktarma Günlüğü** is displayed in the Content view.
3. **Aktarma Günlüğü** penceresi, dosya aktarımlarınızla ilgili aşağıdaki ayrıntıları görüntüler:
 - a) **Kaynak** Kaynak dosyanın bulunduğu sistemdeki aracıyı adı.
 - b) **Hedef** Dosyayı aktarmak istediğiniz aracıya ilişkin aracı adı.
 - c) **Tamamlanma Durumu** Dosya aktarımının durumu. Durum şu değerlerden biri olabilir: "Started", "In progress", "Successful", "Partially Successful", "Cancelled" ya da "Failed".
 - d) **Sahip** Aktarma isteğini sunan anasistemdeki kullanıcı kimliği.
 - e) **Başlatıldı (seçilen saat dilimi)** Dosya aktarma isteğinin Managed File Transfer aracısı tarafından kabul edildiği tarih ve tarih, yöneticinin seçilen saat diliminde gösterilir. Görüntülenen saat dilimini değiştirmek için, **Pencere > Tercihler > IBM MQ Explorer > Managed File Transfer** seçeneklerini tıklayın ve **Saat dilimi:** listesinden bir alternatif saat dilimi seçin. **Tamam**'ı tıklayın.
 - f) **State Kaydedildi (seçilen saat dilimi)** (Bu sütun varsayılan olarak görüntülenmez. You can choose to display the column by using the **Aktarma Günlüğü Kolonlarını Yapılandır**  window.) Yönetici tarafından seçilen saat diliminde tamamlanma durumunun kaydedildiği saat ve tarih.
 - g) **İş Adı** An identifier specified by the user by using the **-jn** parameter of **fteCreateTransfer** or in an Ant script
 - h) **Aktarma Tanıtıcısı** Kütük aktarıma ilişkin benzersiz tanıtıcı.
 - i) **Bağlan: Doğrudan** Details about **İşlem Numarası, İşlem Adı, Birincil Düğüm, İkincil Düğüm, Kaynak Tipi** and **Hedef Tipi** are listed.

Sonuçlar

Not: APAR IC99545 için IBM MQ 8.0.0 Fix Pack 1 içinde Aktarım Günlüğüne ilişkin iç biçim değiştirildi. As a result, if an IBM MQ Explorer is upgraded to V8.0.0.1 or later, and then restored to V8.0.0.0, no audit XML is displayed for transfers that took place while IBM MQ Explorer was at V8.0.0.1. Bu aktarımlar için **Özellikler** penceresindeki XML panosu boş bir metin kutusu içerecektir.

Tamamlanan bir aktarıma ilişkin ek ayrıntıları görüntülemek için, ilgilendiğiniz aktarımı artı işaretini (+) tıklatarak genişletin. Daha sonra, bu aktarımda içerilen tüm kaynak ve hedef dosya adlarını görebilirsiniz. Ancak, aktarma işlemi devam etmekte ve birçok dosyadan oluşuyorsa, yalnızca şu ana kadar aktarılmış olan dosyaları görüntüleyebilirsiniz.

Aktarma Günlüğü ' ta görüntülenenleri yenilemek için, İçerik görünümü araç çubuğundaki **Yenile**

düğmesini  tıklayın. The file transfer information in the Transfer Log remains in the log after you stop and restart the IBM MQ Explorer. Tamamlanan tüm dosya aktarımlarını günlüğün içinden silmek istiyorsanız, İçerik görünümü araç çubuğunda **Tamamlanan Aktarımları Kaldır**  simgesini tıklayın.

Tamamlanan bir dosya aktarımı günlüğünden silmek için, aktarma işlemi farenin sağ düğmesiyle tıklayın ve **Sildüğmesini** tıklayın. Bir aktarımı silerseniz, devam etmekte olan ya da zamanlanan bir aktarımı durdurmaz ya da iptal etmez; yalnızca depolanan geçmiş verilerini siliyorsanız, bu işlemi iptal edebilirsiniz.

Bir aktarımın benzersiz tanıtıcısını panoya kopyalamak için, o aktarımı sağ tıklayın ve **Kimliği Kopyala** ' yı tıklayın.

Aktarım için meta veriler ve eksiksiz denetim XML ' i, **Özellikler** işleminin altında beliren menüden kullanılabilir.

İlgili görevler

[“Devam etmekte olan dosya aktarımlarını izleme” sayfa 205](#)

Devam etmekte olan bir dosya aktarımında, IBM MQ Exploreriçindeki **Yönetilen Dosya Aktarması-Yürürlükteki Aktarma İlerleme Durumu** sekmesini kullanarak izleyebilirsiniz. Bu dosya aktarma işlemi IBM MQ Explorer ya da komut satırından başlatılabilir. Bu sekme, zamanlanan aktarımların başladığı noktadaki zamanlanan aktarımların ilerleyişini de görüntüler.

[“Aktarma Günlüğünün Yapılandırılması” sayfa 209](#)

Hangi bilgilerin görüntüleneceğini ve bilgilerin IBM MQ Exploreriçindeki **Aktarma Günlüğü** ' nde nasıl görüntüleneceğini yapılandırabilirsiniz.

[“Durdurulan aktarımların kurtarılması için bir zamanaşımı ayarlanması” sayfa 291](#)

Bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan, durdurulan dosya aktarımları için bir aktarım kurtarma zamanaşımı ayarlayabilirsiniz. Ayrıca, tek bir aktarım için aktarma kurtarma zaman aşımı da ayarlayabilirsiniz. Belirli bir süre için saniye cinsinden, bir kaynak aracının durmuş bir dosya aktarımını kurtarmaya çalıştığı ve aktarma zamanaşımına ulaştığında aktarma başarısız olduğunda aktarma başarısız olursa, saniye cinsinden belirli bir süre değerini ayarlamalıdır.

Aktarma Günlüğünün Yapılandırılması

Hangi bilgilerin görüntüleneceğini ve bilgilerin IBM MQ Exploreriçindeki **Aktarma Günlüğü** ' nde nasıl görüntüleneceğini yapılandırabilirsiniz.

Bu görev hakkında


Aktarım Günlüğü ' nde sütunların sırasını yeniden düzenlemek için, taşımak istediğiniz sütunun başlığını tıklatın ve sütunu yeni konumuna sürükleyin. Yeni kolon sırası yalnızca siz sonraki durağa kadar alıkonur ve IBM MQ Explorer' yi yeniden başlatıncaya kadar saklanır.

Aktarım Günlüğü ' ndeki girdileri süzgeçten geçirmek için **Görüntülenen günlük girdilerini süzgeçten geçir** alanına bir dize girin. Günlükteki tüm girişleri geri yüklemek için, alandan girdiğiniz dizgiyi silin. Bu alanda geçerli herhangi bir Java düzenli ifadesini kullanabilirsiniz. Daha fazla bilgi için bkz. [MFT tarafından kullanılan düzenli ifadeler](#).

Aktarma Günlüğü 'nde hangi sütunların görüntüleneceğini özelleştirmek için **Aktarma Günlüğü**

Kolonlarını Yapılandır  kullanın. **Aktarma Günlüğü Kolonlarını Yapılandır** penceresini başlatmak ve kullanmak için aşağıdaki adımları kullanın.

Yordam

1. İçerik görünümünde **Aktarma Günlüğü** ' in açık olduğundan emin olun. İçerik görünümü araç çubuğunda **Aktarım Günlüğü Sütunlarını Yapılandır**  düğmesini tıklatın. **Aktarma Günlüğü Kolonlarını Yapılandır** penceresi açılır.
2. **Aktarma Günlüğü** görünümünü özelleştirmek için göstermek ya da gizlemek istediğiniz sütunlara ilişkin tek tek onay kutularını işaretleyin ya da bu kutuların işaretini kaldırın. You can click **Tümünü Seç**, then **Tamam** to select all of the check boxes or **Tüm Seçimi Kaldır**, then **Tamam** to clear all of the check boxes.

İlgili görevler

[“Devam etmekte olan dosya aktarımlarını izleme” sayfa 205](#)

Devam etmekte olan bir dosya aktarımında, IBM MQ Exploreriçindeki **Yönetilen Dosya Aktarması-Yürürlükteki Aktarma İlerleme Durumu** sekmesini kullanarak izleyebilirsiniz. Bu dosya aktarma işlemi IBM MQ Explorer ya da komut satırından başlatılabilir. Bu sekme, zamanlanan aktarımların başladığı noktadaki zamanlanan aktarımların ilerleyişini de görüntüler.

[“Aktarma Günlüğü 'nde dosya aktarımlarının durumunu görüntüleme” sayfa 208](#)

Dosya aktarımlarının ayrıntılarını IBM MQ Explorerindeki **Aktarma Günlüğü** ' ne kullanarak görüntüleyebilirsiniz. Bu aktarımlar, komut satırından ya da IBM MQ Explorerkomutundan başlatılmış olabilir. Ayrıca, **Aktarma Günlüğü**' ta görüntülenenleri özelleştirebilirsiniz.

MFT kaynaklarının izlenmesi

Managed File Transfer kaynaklarını izleyebilirsiniz; örneğin, bir kuyruk ya da dizin. Bu kaynaktaki bir koşul yerine getirildiğinde, kaynak izleme programı bir görev başlatır (örneğin, dosya aktarımı). You can create a resource monitor by using the **fteCreateMonitor** command or the **İzleme Programları** view in the Managed File Transfer plug-in for IBM MQ Explorer.

Bu görev hakkında

Managed File Transfer kaynak izlemesi aşağıdaki terminolojiyi kullanır:

Kaynak izleme programı

Kaynak izleme programı, kaynak içeriklerinin değişip değişmediğini görmek için, önceden tanımlanmış bir düzenli aralıklarla bir kaynağı (dizin ya da kuyruk gibi) yoklayan bir işlemdir. Varsa, içerikler bu izleme programına ilişkin koşullar kümesiyle karşılaştırılır. Bir eşleşme varsa, bu izleme programına ilişkin görev başlatılır.

Kaynak

Kaynak izleme programının tetikleme koşullarıyla karşılaştırılacak her yoklama aralığında incelediği sistem kaynağı. Kuyruklar, dizinler ya da içiçe yerleştirilmiş dizin yapıları izlenen kaynak olabilir.

Koşul ve Tetikleme koşulu

Koşul, değerlendirilen bir ifadedir (tipik olarak, izlenen kaynağın içeriğine karşı). İfade doğru (true) olarak değerlendirilirse, koşul genel tetikleyici koşuluna katkıda bulunur.

Tetikleme koşulu genel koşuldur ve tüm koşullar karşılandığında karşılanır. Tetikleme koşulu yerine getirildiğinde görevin devam edebilmesi gerekir.

Görev

Görev, tetikleme koşulu ya da koşul kümesi yerine getirildiğinde başlatılan işlemdir. Desteklenen görevler, dosya aktarma ve komut çağrılaridir.

Tetikleme dosyası

Tetikleyici dosyası, bir görevin (genellikle bir aktarımda) başlayabileceğini göstermek için izlenen bir dizine yerleştirilen bir dosyadır. Örneğin, işlenecek tüm dosyaların bilinen bir yere vardığını ve aktarılabilceğini ya da başka bir şekilde işlem yapıp aktarılabilceğinin göstergesi olabilir. Tetikleme dosyasının adı, değişken yerine koyma değeri kullanılarak aktarılacak kütükleri belirtmek için kullanılabilir. Daha fazla bilgi için bkz [“Customizing MFT resource monitor tasks with variable substitution” sayfa 221.](#)

Tetikleme dosyası, hazır dosya ya da go dosyası olarak da bilinir. Ancak, bu belgede genellikle tetikleyici dosyası olarak anılır.

Kaynak izleme, protokol köprüsü araçlarında ya da Connect:Direct köprü araçlarındatarafından desteklenmez.

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 aracıyla ilişkilendirilir ve yalnızca aracı çalışırken etkindir. İzleme aracı durduğunda, kaynak izleme programı da durur. Kaynak izleme programı yaratıldığında aracı zaten çalışıyorsa, kaynak izleme programı hemen başlar. İzleme aracı, kaynak izleyici tarafından başlatılan görevin kaynak aracı 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 9.2.2 Long Term Supportve IBM MQ 9.2.2öncesinde Continuous Delivery için, kaynak izleme programını durdurmanın tek yolu, izleme işlemini çalıştıran aracıyı durdurmandır. Bir kaynak izleme programını yeniden başlatmak için aracıyı tümüyle yeniden başlatmanız gerekir. IBM MQ 9.2.2' 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 213.](#)

Bir aracıda yaratılabilecek kaynak izleme programı sayısı üzerinde herhangi bir kısıtlama yoktur ve tümü aynı öncelikle 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 özniteliği 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.

V 9.2.0 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.

V 9.2.0 IBM MQ 9.1.3' den bir dosya aktarılmadıysa, 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 geçmişini temizliyor” sayfa 237.](#)

Kaynaklar

Managed File Transfer içindeki kaynak izleyicileri aşağıdaki iki tip kaynağın içeriğini yoklayabilir:

Dizinler ya da içiçe izin 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 izin izlenir. Alt dizinleri incelemek için **fteCreateTransfer** komutunda özinyeleme 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. [“Customizing MFT resource monitor tasks with variable substitution” sayfa 221.](#)

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. [“Bir dizinin izlenmesi ve değişken yerine koyma değerinin kullanılması” sayfa 219.](#)

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: Bir MFT kaynağının yapılandırılması” sayfa 220.](#)

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** deęiştirgesinin deęeri olarak iletiyorsunuz. **fteCreateMonitor** çalıştırıldığında, görev XML belgesini okur. **fteCreateMonitor** çalıştırıldıktan sonra, görev XML dosyasında yapılan deęişiklikler izleme programı tarafından kullanılmaz.

Bir dosya aktarma görevini kullanırken, bir göreve kaç tetikleyici koşulunun toplu olarak dahil edileceğini seçebilirsiniz. Varsayılan deęer, bir tetikleyici koşulunun bir görevi başlatması içindir. Toplu olarak tek bir görevde toplanacak tetikleyici koşullarının sayısını seçmek için [fteCreateMonitor](#) komutunu **-bs** seçeneęiyle 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 215.

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, içerięi kaynak dosya yolları ve isteęe baęlı olarak hedef dosya yolları için ayrıştırılır. Bu dosya yolları, belirlediğiniz görev aktarma XML dosyasındaki dosya öğelerini tanımlamak için kullanılır ve aracıya tek aktarma isteęi olarak sunulur. Kaynak izleme programının tanımlaması, tetikleyici içerięinin 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. "[Tetikleme Dosyasının Kullanılması](#)" sayfa 229.

Kaynak izleme programlarının başlatılması ve durdurulması

Long Term Supportve IBM MQ 9.2.Öncesinde Continuous Delivery için, 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 aracısının başlatılması](#)" sayfa 193 ve "[MFT aracısını durdurma](#)" sayfa 198.

V 9.2.2 IBM MQ 9.2.2olanaęından, **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 araçıda 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 deęilse ve gereksiz yere çalışmasını istemiyorsanız, bu nedenle deęerli sistem kaynaklarını tüketin.

V 9.2.2 Ek bilgi için [MFT kaynak izleme programının başlatılması ve MFT kaynak izleme programının durdurulması](#)başlıklı konuya bakın.

Çizelge 9. Hangi komutun çalıştırıldığına bağlı olarak bir kaynak izleme programının davranışı	
Komut	Kaynak izleme programının davranışı
V 9.2.2 fteStartMonitor	Aracı çalışıyorsa, kaynak izleme programı durdurulmuşsa başlatılır.
V 9.2.2 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 yapılanışını bir XML dosyasına aktarmak için **fteCreateMonitor** komutunu **-ox** değiştirilmesiyle ve kaynak izleyicisi yapılanışını bir XML dosyasından içe aktararak kaynak izleyicisini geri yüklemek için **-ix** değiştirilmesiyle 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 236.

Kaynak izleyicisi günlük kaydı

IBM MQ 9.1.0' den Managed File Transfer , kaynak izleyici günlük kaydını içerir. Daha fazla bilgi için bkz [“Logging MFT resource monitors”](#) sayfa 231.

İlgili kavramlar

[“Customizing MFT resource monitor tasks with variable substitution”](#) sayfa 221

Etkin bir kaynak izleyicisinin tetikleme koşulları yerine getirildiğinde, tanımlanan görev çağrılır. Aktarma ya da komut görevini her seferinde aynı hedef aracıya ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımını da değiştirebilirsiniz. Bunu yapmak için, görev tanımlaması XML ' ine değişken adları ekleyin. İzleme programı, tetikleme koşullarının yerine getirildiğini ve görev tanımlamasının değişken adları içerdiğini saptadığında, değişken adlarını değişken değerleriyle değiştirir ve daha sonra, görevi çağırır.

İlgili görevler

[“Komutları ve komut dosyalarını başlatmak için MFT izleme görevlerinin yapılandırılması”](#) sayfa 215
Kaynak izleme programları, ilişkili görevleri olarak dosya aktarımlarını gerçekleştirmek için sınırlı değildir. İzleyici 'yi, yürütülür 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 ' ini düzenleyerek, ilgili komut çağırısı parametreleriyle (bağımsız değişkenler ve özellikler gibi) bir ya da daha çok komut ögesi içermesini sağlar.

[“Örnek: Bir MFT kaynağının yapılandırılması”](#) sayfa 220

Bir kaynak izleme programı tarafından izlenecek kaynak olarak **-mq** kuyruğunu kullanarak **fteCreateMonitor** komutunu kullanarak IBM MQ kuyruğuspecifykomutunu kullanarak belirtebilirsiniz.

[“Kuyruğun izlenmesi ve deęişken yerine koyma deęerinin kullanılması” sayfa 226](#)

You can monitor a queue and transfer messages from the monitored queue to a file by using the **fteCreateMonitor** command. İzlenen kuyruktan okunan ilk iletiden herhangi bir IBM MQ ileti özelliğinin deęeri, görev XML tanımlamasındaki yerine koyulabilir ve aktarma davranışını tanımlamak için kullanılır.

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 görevleri olarak dosya aktarımlarını gerçekleştirmek için sınırlı değildir. İzleyici 'yi, yürütülür programlar, Ant komut dosyaları ya da JCL işleri de içinde olmak üzere, izleme aracısından dięer komutları çağırarak için de yapılandırabilirsiniz. Komutları çağırarak için, izleme görevi tanımlaması XML ' ini düzenleyerek, ilgili komut çağırısı parametreleriyle (bağımsız deęişkenler ve özellikler gibi) bir ya da daha çok komut ögesi içermesini sağlar.

Bu görev hakkında

İzleme aracısının aramasını istediğiniz yürütülebilir programın, Ant komut dosyasının ya da JCL işinin dosya yolu, izleme aracısının commandPath ' ine eklenmelidir. Komut yolu özelliği hakkında bilgi için bkz. [commandPath MFT property](#).

Görev tanımlaması XML belgesini aşağıdaki yollardan birini kullanarak yaratabilirsiniz:

- Create the task definition XML document manually according to the `FileTransfer.xsd` schema.
- Oluşturulan bir XML belgesini, görev tanımlamanızın temeli olarak kullanın.

Bir aktarma görevi ya da komut görevi isteyip istemediğiniz, görev tanımlamasının bir `<request>` kök ögesiyle başlaması gerekir. `<request>` alt ögesinin `<managedTransfer>` ya da `<managedCall>` olması gerekir. You would typically choose `<managedCall>` when there is a single command or script to run, and `<managedTransfer>` if you want the task to include a file transfer and optionally up to four command calls.

Yordam

- Görev tanımlaması XML belgesini `FileTransfer.xsd` şemasına göre el ile yaratmak için bkz. [“Bir görev tanımlaması XML ' inin şemaya göre el ile yaratılması” sayfa 215](#).
- Oluşturulan bir belgeyi deęiştirerek bir görev tanımı oluşturmak için, **fteCreateTransfer -gt** parametresi tarafından oluşturulan XML belgesini düzenleyin. Daha fazla bilgi için bkz [“Oluşturulan bir belgeyi deęiştirerek bir görev tanımlaması belgesi oluşturma” sayfa 217](#).

Bir görev tanımlaması XML ' inin şemaya göre el ile yaratılması

Bir görev tanımlaması XML dosyasını `FileTransfer.xsd` şemasına göre el ile yaratabilirsiniz.

Bu görev hakkında

The schema `FileTransfer.xsd` can be found in the `MQ_INSTALLATION_PATH/mqft/samples/schema`. Bu şemayla ilgili ek bilgi için [Dosya aktarma isteęi ileti biçimibaşlıklı konuya](#) bakın.

Örnek

The following example shows an example task definition XML document saved as `cleanuptask.xml`, which uses the `<managedCall>` element to call an Ant script called `RunCleanup.xml`. The `RunCleanup.xml` Ant script must be located on the `commandPath` of the monitoring agent.

```
<?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` dosyasında adı belirtilen Ant komut dosyasıyla yapılandırılmış Managed File Transfer Agent ögesini belirtir.

`<call><command>` . . . yapısı, çalıştırmak istediğiniz yürütülebilir dosyayı ya da komut dosyasını tanımlar. Komut isteğe bağlı bir `type` özniteliğini alır; bu öznitelik aşağıdaki değerlerden birine sahip olabilir:

antscript

Ayrı bir JVM ' de bir Ant komut dosyasını çalıştırın.

Yürütülebilir

Yürütülebilir bir programı çağırın.

jcl

JCL işini çağırın.

`type` özniteliğini atlarsanız, varsayılan değer olan yürütülebilir 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 dosyası ya da programı arar.

The `retrycount` attribute specifies the number of times to try calling the program again if the program does not return a success return code. Bu öznitelige atanan değer negatif olmamalıdır. `retrycount` özniteliğini belirtmezseniz, varsayılan değer olarak sıfır kullanılır.

The `retrywait` attribute specifies the time to wait, in seconds, before trying the program invocation again. Bu öznitelige atanan değer negatif olmamalıdır. `retrywait` özniteliğini belirtmezseniz, varsayılan değer olarak sıfır kullanılır.

The `successrc` attribute is an expression used to determine when the program invocation successfully runs. Komut için dönüş kodu bu ifade kullanılarak değerlendirildi. Değer, Boole OR ya da işareti (&) belirtmek için dikey çubuk (|) karakteriyle birleştirilen bir ya da daha fazla ifadeden oluşabilir. Boole AND işlecini belirtir. Her ifade aşağıdaki ifade tiplerinden biri olabilir:

- Süreç dönüş kodu ile sayı arasındaki eşitlik testini belirtmek için bir sayı.
- Sayı ile süreç dönüş kodu arasındaki testten daha büyük bir testi belirtmek için, önekli bir sayı (>) önekli bir sayı.
- Başında karakteri olmayan bir sayı öneki (<) Sayı ile süreç dönüş kodu arasındaki testten daha az bir testi belirtmek için.
- Sayı ile işlem dönüş kodu arasındaki eşit olmayan sınamayı göstermek için, önünlem işareti karakteri (!) olan bir sayı. Örneğin: > 2 & < 7 &! 5 | 0 | 14, başarılı olmak için şu dönüş kodları olarak yorumlanır: 0, 3, 4, 6, 14. Diğer tüm dönüş kodları başarısız olduğu şeklinde yorumlanır.

successrc özneliğini belirtmezseniz, varsayılan değer olarak sıfır kullanılır. Bu, komutun, bir sıfır kodunu döndürmesi durumunda başarılı bir şekilde çalıştırılacağına ilişkin olarak değerlendirileceği anlamına gelir.

Bir Ant komut dosyası için, tipik olarak <target> ve <property> öğelerini belirtiyorsunuz. <target> öğesi değerlerinin, Ant komut dosyasındaki hedef adlarla eşleşmesi gerekir.

Yürütülebilir programlar için, <argument> öğelerini belirtebilirsiniz. İç içe geçmiş bağımsız değişken öğeleri, program çağrısının bir parçası olarak çağrılmakta olan programa geçirilecek bağımsız değişkenleri belirtir. Program bağımsız değişkenleri, bağımsız değişken öğelerinin belirttiği değerlerden bağımsız değişken öğelerinin saptanması için oluşturulur. Bir program çağrısının iç içe yerleştirilmiş öğeleri olarak sıfır ya da daha fazla bağımsız değişken öğesi belirtebilirsiniz.

The administrator defines and starts the monitor as normal using the task definition XML document that includes the <managedCall> element. 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 kullanarak (bu örnekte /tasks/cleanuptask.xml) çalıştırdığınız yerel dosya sisteminde olmalıdır. cleanuptask.xml belgesi yalnızca kaynak izleyicisini yaratmak için kullanılır. cleanuptask.xml belgesi başvurularının (Ant komut dosyaları ya da JCL işleri) izleme aracısının komut yolunda olması gereken görevler. İzleme programı tetikleme koşulu yerine getirildiğinde, görev tanımlaması XML 'indeki değişkenlerin yerine, izleme programından gelen gerçek değerler konur. So for example \${FilePath} is replaced in the request message sent to the agent with /monitored/cleanup.go. İstek iletisi, aracı komut kuyruğuna yerleştirilir. Komut işlemcisi, isteğin bir program çağrısı için olduğunu algılar ve belirtilen programı başlatır. antscript tipi bir komut çağrılırsa, yeni bir JVM başlatılır ve Ant görevi yeni JVM altında çalışır. Değişken yerine koyma değerinin kullanılmasına ilişkin ek bilgi için [Değişken yerine koyma değeri olan görevlerin uyarlanması](#) başlıklı konuya bakın.

İlgili kavramlar

“Customizing MFT resource monitor tasks with variable substitution” sayfa 221

Etkin bir kaynak izleyicisinin tetikleme koşulları yerine getirildiğinde, tanımlanan görev çağrılır. Aktarma ya da komut görevini her seferinde aynı hedef aracıya ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımını da değiştirebilirsiniz. Bunu yapmak için, görev tanımlaması XML 'ine değişken adları ekleyin. İzleme programı, tetikleme koşullarının yerine getirildiğini ve görev tanımlamasının değişken adları içerdiğini saptadığında, değişken adlarını değişken değerleriyle değiştirir ve daha sonra, görevi çağırır.

İlgili başvurular

[Dosya aktarma isteği iletme biçimi](#)

[commandPath MFT özelliği](#)

Oluşturulan bir belgeyi değiştirerek bir görev tanımlaması belgesi oluşturma

You can create the monitor task definition document by modifying the XML document generated by the **-gt** option of **fteCreateTransfer**.

Bu görev hakkında

The generated document has a `<request>` followed by `<managedTransfer>` element. Bu görev tanımlamasını geçerli bir `<managedCall>` yapısına dönüştürmek için aşağıdaki adımları izleyin:

Yordam

1. `<managedTransfer>` başlangıç ve bitiş etiketlerini `<managedCall>` etiketleriyle değiştirin.
2. Herhangi bir `<schedule>` ögesini ve alt düğümü kaldırın.
3. `<sourceAgent>` başlangıç ve bitiş etiketlerini, izleme aracı yapılandırma ayrıntılarıyla eşleştirmek için `<agent>` ile değiştirin.
4. `<destinationAgent>` ve `<trigger>` ögelerini kaldırın.
5. `<item>` ögelerini kaldırın.
6. Herhangi bir `preSourceCall`, `postSourceCall`, `preDestinationCall`ya da `postDestinationCall` ögesini kaldırın.
7. `<transferSet>` ögesi içinde yeni bir `<call>...</call>` yapısı ekleyin. Bu yapı, aşağıdaki örnekteki 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

Ayrıca, tüm dosya aktarma ayrıntıları da dahil olmak üzere `<managedTransfer>` ögesini de saklayabilir ve en çok dört komut çağrısı ekleyebilirsiniz. Bu durumda, `<metaDataSet>` ve `<item>` ögeleri arasında aşağıdaki çağrı elemanlarından herhangi bir seçim ekleyebilirsiniz:

Kaynak Arama

Aktarma işlemi başlatılmadan önce kaynak aracıda bir program çağırın.

postSourceArama

Aktarma işlemi tamamlandıktan sonra, kaynak aracıdaki bir programı çağırın.

preDestinationCall

Aktarma işlemi başlatılmadan önce hedef aracıda bir program çağırın.

postDestinationCall

Aktarma işlemi tamamlandıktan sonra hedef aracıda bir program çağırın.

Bu ögelerin her biri, daha önceki örnekte açıklandığı gibi `<command>` ögesi yapısını alır. `FileTransfer.xsd` şeması, çeşitli çağrı ögeleri tarafından kullanılan tipleri tanımlar.

Aşağıdaki örnekte, bir görev tanımlaması belgesinde önceden `SourceCall`, `postSourceCall`, `preDestinationCall` ve `postDestinationCall` gösterilmektedir:

```
:
<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">

```

Aktarmaya farklı tipte komut tiplerini karıştırabilirsiniz. Bağımsız değişken, hedef ve özellik öğeleri isteğe bağlıdır.

Bir dizinin izlenmesi ve değişken yerine koyma değerinin kullanılması

fteCreateMonitor komutunu kullanarak bir dizini izleyebilirsiniz. Bir yerine koyma değişkeninin değeri, görev XML tanımlamasındaki yerine koyulabilir ve aktarma davranışını tanımlamak için kullanılır.

Bu görev hakkında

Bu örnekte, kaynak aracıya AGENT_HOP adı verilir. AGENT_HOP izleyicileri /test/monitoredolarak adlandırıldığı dizin. Ajan her 5 dakikada bir dizini yok ediyor.

Dizine bir .zip dosyası yazıldıktan sonra, dosyayı dizine yazan uygulama aynı dizine bir tetikleyici dosyası yazar. Tetikleme dosyasının adı, .zip dosyasının adıdır, ancak dosya uzantısı farklı olur. For example, after the file file1.zip is written to the directory, the file file1.go is written to the directory. The resource monitor monitors the directory for files that match the pattern *.go then uses variable substitution to request a transfer of the associated .zip file.

Yordam

1. İzleme programının, tetiklendiğinde gerçekleştirdiği görevi tanımlayan görev XML ' ini 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>

```

Tetikleme dosyasıyla ilişkili değerlerle değiştirilen değişkenler, **koyu** harfiyle vurgulanır. Bu görev XML dosyası /home/USER1/task.xml dosyasına kaydedildi.

2. /test/monitored dizinini izlemek için bir kaynak izleme programı yaratın.

Aşağıdaki komutu gönderin:

```
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/monitored` dizinine yazar, daha sonra `jump.go` dosyasını dizine yazar.
4. İzleme programı, `jump.go` dosyasının varlığı tarafından tetiklenir. Aracı, tetikleyici dosya ile ilgili bilgileri görev XML ' ine koyar.

Bu, görev XML ' inin dönüştürülmekte olduğu görev ile sonuçlanır:

```
<?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

Görev XML ' i tarafından tanımlanan aktarma gerçekleştirilir. `jump.zip` dosyası, `AGENT_HOP` tarafından `/test/monitored` dizininden okunur ve `AGENT_SKIP` ' in çalışmakta olduğu sistemde bulunan `/out/jump.zip` adlı dosyaya aktarılır.

İlgili kavramlar

“Customizing MFT resource monitor tasks with variable substitution” sayfa 221

Etkin bir kaynak izleyicisinin tetikleme koşulları yerine getirildiğinde, tanımlanan görev çağrılır. Aktarma ya da komut görevini her seferinde aynı hedef aracıya ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımını da değiştirebilirsiniz. Bunu yapmak için, görev tanımlaması XML ' ine değişken adları ekleyin. İzleme programı, tetikleme koşullarının yerine getirildiğini ve görev tanımlamasının değişken adları içerdiğini saptadığında, değişken adlarını değişken değerleriyle değiştirir ve daha sonra, görevi çağırır.

İlgili görevler

“Komutları ve komut dosyalarını başlatmak için MFT izleme görevlerinin yapılandırılması” sayfa 215

Kaynak izleme programları, ilişkili görevleri olarak dosya aktarımlarını gerçekleştirmek için sınırlı değildir. İzleyici 'yi, yürütülür programlar, Ant komut dosyaları ya da JCL işleri de içinde olmak üzere, izleme aracısından diğer komutları çağırarak için de yapılandırabilirsiniz. Komutları çağırarak için, izleme görevi tanımlaması XML ' ini düzenleyerek, ilgili komut çağrısı parametreleriyle (bağımsız değişkenler ve özellikler gibi) bir ya da daha çok komut ögesi içermesini sağlar.

İlgili başvurular

fteCreateMonitor: bir MFT kaynak izleme programı yaratır

Örnek: Bir MFT kaynağının yapılandırılması

Bir kaynak izleme programı tarafından izlenecek kaynak olarak **-mq** kuyruğunu kullanarak **fteCreateMonitor** komutunu kullanarak IBM MQ kuyruğuspecifykomutunu kullanarak belirtebilirsiniz.

Bu görev hakkında

Bu örnekte, izlenecek kaynak kuyruğdur *MONITORED_QUEUE*. Bu kuyruk, izleme aracısının kuyruk yöneticisinde (*QM_NEPTUNE*) olmalıdır. Kuyruğun izlendiği koşul, eksiksiz bir ileti grubunun varlığıdır. Koşul karşılanırsa, gerçekleştirilecek görev *task.xml* dosyasında tanımlıdır.

Not: Tek bir kuyruğu izlemek için birden çok kaynak izleme programı yaratmayın. Daha sonra, önceden kestirilemeyecek bir davranış ortaya çıkar.

Yordam

Aşağıdaki komutu yazın:

```
fteCreateMonitor -ma AGENT_NEPTUNE -mn myMonitor -mm QM_NEPTUNE -mq MONITORED_QUEUE  
-mt task.xml -tr completeGroups -pi 5 -pu minutes
```

İzleyici, *completeGroups* koşulunun doğru olup olmadığını görmek için kuyruğun her beş dakikada bir kuyruğunu denetler. Kuyrukta bir ya da daha çok tam grup varsa, izleme programı her bir tam grup için bir kez *task.xml* dosyasında tanımlanan görevi çalıştırır.

Customizing MFT resource monitor tasks with variable substitution

Etkin bir kaynak izleyicisinin tetikleme koşulları yerine getirildiğinde, tanımlanan görev çağrılır. Aktarma ya da komut görevini her seferinde aynı hedef aracıya ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımını da değiştirebilirsiniz. Bunu yapmak için, görev tanımlaması XML 'ine değişken adları ekleyin. İzleme programı, tetikleme koşullarının yerine getirildiğini ve görev tanımlamasının değişken adları içerdiğini saptadığında, değişken adlarını değişken değerleriyle değiştirir ve daha sonra, görevi çağırır.



Uyarı: Değişken adları büyük ve küçük harfe duyarlı değildir.

Yerine koyma değeri için kullanılan değişkenler yalnızca pozitif tetikleyici koşulları için kullanılabilir. Yalnızca *match* ve *fileSize* tetikleme koşulları, değişkenlerin yerine konabilmesine neden olur. Bir *noMatch* koşulu kullanılıyorsa ve görev tanımlamasında yerine koyma değişkeni adları varsa, görev çağrılmaz ve izleme programı, 110 dönüş kodunu ve BFGDM0060Ehata iletisini yükseltir.

İzlenen kaynak bir kuyruksa

İzlenen kuyruktan okunabilecek ilk iletiden herhangi bir IBM MQ ileti özelliğinin değeri, görev XML tanımındaki yerine koyulabilir.

Kullanıcı tanımlı ileti özelliklerinin başına *usr.* öneki eklenir, ancak bu öneki değişken adına eklemeyin. Değişken adlarından önce bir dolar işareti (\$) karakteri eklenmiş ve kaşlı ayraç { } içine alınmalıdır.

Örneğin, $\${destFileName}$, kaynak kuyruktan okunmak üzere ilk iletinin *usr.destFileName* ileti özelliğinin değeri ile değiştirilir. Daha fazla bilgi için bkz. [Kaynak kuyruklardaki iletilerden MFT tarafından okunanMQ ileti özellikleri](#) ve [“Kuyruğun izlenmesi ve değişken yerine koyma değerinin kullanılması” sayfa 226](#).

Bir değişken ileti özelliği olarak tanımlanmazsa, izleme programı bir BFGDM0060E hatasını bildirir ve dönüş kodu 110 (Monitor görev değişkeni yerine koyma işlemi başarısız oldu) değerini döndürür. Bunun yanı sıra aracı, olay günlüğüne (*outputN.log*) aşağıdaki hata iletisini yazar:

```
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 orta ya da ayrıntılı kaynak izleyicisi günlüğe kaydetme etkinleştirilmişse, izleme programı aracının (*resmoneventN.log*) kaynak izleyicisi olay günlüğüne aşağıdaki iletiyi yazar:

```
BFGDM0060E: A monitor task could not complete as a variable substitution  
<variable name> was not present.
```

Kaynak izleyicisi günlüğe kaydetme ile ilgili daha fazla bilgi için bkz. [“Logging MFT resource monitors” sayfa 231](#) .

Aşağıdaki çizelge, varsayılan değer olarak hangi yerine koyma değişkenlerinin sağlansa olduğunu gösterir. Örneğin, `{AGENTNAME}` , kaynak izleyici aracısının adıyla değiştirilir.

<i>Çizelge 10. Varsayılan olarak sağlanan yerine koyma değişkenleri</i>	
Değişken	Tanım
AGENTNAME	Kaynak izleyici aracısının adı.
queueName	İzlenmekte olan kuyruğun adı.
Kodlama	Kuyrukta ilk iletinin karakter kodlaması ya da bir gruptaki ilk iletinin karakter kodlaması.
MESSAGEID	Kuyrukta ilk iletinin IBM MQ ileti tanıtıcısı ya da gruptaki ilk ileti.
GRUPID	Yalnızca tek bir ileti bulunursa, grubun IBM MQ grup kimliği ya da ileti tanıtıcısı. Bu değişken yalnızca, tam grupları izliyorsanız ayarlıdır.
CurrentTimeDamg ası	İzleme programının tetiklediği yerel saati temel alan bir zaman damgası. Zaman damgası değeri, aracı için benzersizdir.
CurrentTimeStamp UTC	İzleme programının tetiklediği, UTC saat diliminde zamana dayalı bir zaman damgası. Zaman damgası değeri, aracı için benzersizdir.

İzlenen kaynak bir diziyse

Aşağıdaki çizelge, görev XML tanımlamasında yerine konabilecek değişken adları kümesini göstermektedir.

<i>Çizelge 11. Yerine konabilen değişkenler</i>	
Değişken	Tanım
FilePath	Tetikleyici dosyasının tam yol adı.
FileName	Tetikleyicinin dosya adı kısmı.
LastModifiedSa ati	Tetikleme dosyasının en son değiştirildiği saat. Bu süre, aracının çalışmakta olduğu saat diliminin yerel saati olarak ifade edilir ve ISO 8601 saati olarak biçimlendirilir.
LastModifiedTa rihi	Tetikleme dosyasının en son değiştirildiği tarih. Bu tarih, aracının çalıştırıldığı saat diliminin yerel tarihi olarak ifade edilir ve ISO 8601 tarihi olarak biçimlendirilir.
LastModifiedTi meUTC	Tetikleme dosyasının en son değiştirildiği saat. Bu süre, yerel saat UTC saat dilimine dönüştürülmüş ve ISO 8601 saati olarak biçimlendirilmiş olarak ifade edilir.
LastModifiedD ateUTC	Tetikleme dosyasının en son değiştirildiği tarih. Bu tarih, yerel tarih UTC saat dilimine dönüştürülmüş ve ISO 8601 tarihi olarak biçimlendirilmiş olarak ifade edilir.
AgentName	Kaynak izleyici aracısının adı.
CurrentTimeDa mgası	İzleme programının tetiklediği yerel saati temel alan bir zaman damgası. Zaman damgası değeri, aracı için benzersizdir.
CurrentTimeSt ampUTC	İzleme programının tetiklediği UTC saat diliminde zamana dayalı bir zaman damgası. Zaman damgası değeri, aracı için benzersizdir.

İzlenen kaynak bir tetikleme dosyaysa

Aşağıdaki çizelge, kaynak izleme programı, aktarılması gereken dosyaları saptamak için bir tetikleme dosyasının içeriğini kullanırken yerine konabilecek değişken adları kümesini göstermektedir.

Çizelge 12. Tetikleme dosyası kullanılırken yerine konabilen değişkenler	
Değişken	Tanım
contentSource	Kaynak dosyanın tam yol adı.
contentDestination	Hedef dosyanın tam yol adı.

Değişken adlarından önce bir dolar işareti (\$) karakteri eklenmeli ve kaşlı ayraç içine alınmış olmalıdır: { }. Örneğin, \${FilePath} , eşleşen tetikleyici dosyasının tam olarak nitelenmiş dosya yolu ile değiştirilir.

Daha fazla iyileştirme sağlamak için değişken adlara uygulanabilen iki özel anahtar sözcük vardır. Bu bilgiler şunlardır:

belirteç

Yerine koyulacak simge dizini (soldan 1 'den başlayarak, sağdan -1 'den başlayarak)

ayırıcı

Değişken değerini tokenize etmek için tek bir karakter. Varsayılan değer, AIX and Linux altyapılarında eğik çizgi karakteri (/) ya da Windows altyapılarında ters eğik çizgi karakteri (\); ancak ayırıcı, değişken değerinde görünebilecek geçerli herhangi bir karakter olabilir.

Bir değişken adında ayırıcı anahtar sözcüğü belirtilirse, değişken değeri, ayırıcı karaktere göre simgelere bölünür.

Belirteç anahtar sözcüğünün atandığı değer, değişken adını değiştirmek için kullanılacak simgeyi seçmek üzere bir dizin olarak kullanılır. Simge dizini, değişkenin ilk karakterine göredir ve 1 'de başlar. Simge anahtar sözcüğü belirtilmediyse, değişkenin tamamı eklenir.

İleti XML ' inde bir aracı adına değiştirilen değerler, büyük ve küçük harfe duyarlı olmayan bir şekilde işlenir. Tüm Managed File Transfer Agent adları büyük harflerdir. If the value Paris is substituted into an agent attribute in the message XML, this value is interpreted as a reference to the agent PARIS.

İlgili kavramlar

“Örnekler: Kaynak izleme programı tanımlamaları için değişken yerine koyma değeri” sayfa 223 XML ve IBM MQ Explorerkullanan kaynak izleme programı tanımlamaları için değişken yerine koyma örnekleri.

İlgili görevler

[Değişken yerine koyma değeri birden çok dosyanın tek bir dosya adına gitmesine neden olursa ne yapılır?](#)

Örnekler: Kaynak izleme programı tanımlamaları için değişken yerine koyma değeri

XML ve IBM MQ Explorerkullanan kaynak izleme programı tanımlamaları için değişken yerine koyma örnekleri.

Değişken yerine koyma değerinin nasıl çalıştığını gösteren örnekler

Eşleşen tetikleme dosyasının dosya yolunun Windows üzerinde c:\MONITOR\REPORTS\Paris\Report2009.doc ve AIX and Linux platformlarında /MONITOR/REPORTS/Paris/Report2009.doc olduğunu varsayarsak, değişkenlerin yerine aşağıdaki tabloda gösterildiği gibi konur.

<i>Çizelge 13. Değişkenlerin yerine nasıl konacağını</i>	
Değişken belirtimi	Değişken yerine koyma işleminden sonra
<code>\${FilePath}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux :/MONITOR/REPORTS/Paris/Report2009.doc
<code>\${FilePath{token=1}{separator=.}}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux :/MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=2}{separator=.}}</code>	Windows : doc AIX and Linux : doc
<code>\${FilePath{token=3}}</code>	Windows RAPORLAR AIX and Linux : Paris

Ayrıca, aşağıdaki tabloda gösterildiği gibi, değişkenin son karakterine göre belirteçleri seçmek için negatif bir simge dizini de belirleyebilirsiniz. The examples in the table use the same variable value, c:\MONITOR\REPORTS\Paris\Report2009.doc on Windows and /MONITOR/REPORTS/Paris/Report2009.doc on AIX and Linux.

<i>Çizelge 14. Negatif simge dizini kullanma örnekleri</i>	
Değişken belirtimi	Değişken yerine koyma işleminden sonra
<code>\${FilePath}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux :/MONITOR/REPORTS/Paris/Report2009.doc
<code>\${FilePath{token=-2}{separator=.}}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux :/MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=-2}{separator=\}}</code>	Windows : Paris AIX and Linux : Paris
<code>\${FilePath{token=-4}}</code>	Windows : İZLEME AIX and Linux : İZLEME

The variables that are used for substitution are only available for the following positive trigger conditions and the noSizeDeğişiklik option, which is an exception to the positive trigger condition rule:

- eşleşen
- fileSize
- noSizeDeğişikliği

Bir noMatch koşulu kullanılıyorsa ve görev tanımlamasında yerine koyma değişkeni adları varsa, görev çağrılmaz ve izleme programı, 110 dönüş kodunu ve BFGDM0060Ehata iletisini yükseltir.

XML kullanan örnek

Aşağıdaki örnek görev tanımlaması XML, aktarım için kaynak aracı olarak izleme aracı adını kullanır (Paris), aktarma için hedef aracı adı olarak dosya yolunda en son dizin adını kullanır (Report2009) ve aktarılan dosyayı, .rptuzantılı tetikleyici dosya adının kökü olacak şekilde 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 dönüştürülmekte olduğu görev ile sonuçlanır:

```
<?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" />
    <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 özniteindeki `{FilePath{token=-2}}` değişkeni, Parisdeğeriyle değiştirildi. This value is treated in a not case-sensitive way and interpreted as a reference to the agent PARIS.

IBM MQ Explorerkullanan örnekler

IBM MQ Exploreraracılığıyla bir kaynak izleme programı yaratırken, izleme programı özellikleri ve tetikleme koşulları belirtildikten sonra, izleme programına aktarma öğeleri eklemek için bu seçenek verilir. The following examples demonstrate how the `{FilePath}` and `{FileName}` variables can be used in the "**Aktarım öğesi ekle panosu ekle**" to customize transfers resulting from a resource monitor match.

Örnek 1

Bir tetikleme koşulu karşılandığında kaynak dosyayı başka bir yere aktarmak için, `{FilePath}` değişkeni kullanılabilir:

- Kaynak **Dosya adı** 'nın `{FilePath}` olmasını ayarlayın.

- Hedef için **Tip** açılan menüsünden **Dizin** seçeneğini belirleyin.
- Hedef **Dosya adı** 'nın, kaynak dosyanın aktarılacağı konum olarak ayarlayın; örneğin, bu C:\MFT\out\olabilir.

Örnek 2

Kaynak dosyayı başka bir yere aktarmak ve dosyanın uzantısını değiştirmek için, `{FileName}` değişkeni `{FilePath}` değişkeniyle birlikte kullanılabilir:

Aşağıdaki örnekte, kaynak dosyanın dosya yolunun C:\MONITOR\REPORTS\Paris\Report2009.doc' a eşit olduğu varsayılır:

- Kaynak **Dosya adı** 'nın `{FilePath}` olmasını ayarlayın.
- Hedef **Dosya adı** 'nın, kaynak dosyanın aktarılacağı, ardından `{FileName}{token=1}{separator=.}` tarafından ve ardından dosyanın yeni uzantısı olmasını istediğiniz konumun olmasını belirleyin. Örneğin, kaynak dosya adıyla C:\MFT\out\Report2009.rpt ' a eşitlenecek olan bu C:\MFT\out\{FileName}{token=1}{separator=.}.rpt olabilir.

Örnek 3

Aktarımın hedefini belirlemek için kaynak dosyanın dosya yolunun bir kısmını kullanmak için, `{FilePath}` değişkeni simge ve ayırıcı belirtilerle birlikte kullanılabilir.

Aşağıdaki örnekte, kaynak dosyanın dosya yolunun C:\MONITOR\REPORTS\Paris\Report2009.doc' a eşit olduğu varsayılır.

Dosyanın hedefini belirlemek için kaynak dosya yolunun bir kısmını kullanmak mümkündür. Dosya, kaynak dosyanın konumuna bağlı olarak bir klasöre aktarılacaktır, C:\MONITOR\REPORTS\Paris\Report2009.doc dosyasının yol örneğini kullanarak, bu örnekte Paris , aşağıdaki gibi bir dosya yapılabilir:

- Kaynak **Dosya adı** 'nın `{FilePath}` olmasını ayarlayın.
- Hedef **Dosya adı** 'nın, her bir konuma ilişkin klasörlerin yer aldığı hedef olacak şekilde ayarlayın ve daha sonra, dosya yolunun hedef kısmını ve dosya adını ekleyin. Örneğin, kaynak dosya adıyla C:\MFT\out\Paris\Report2009.doc ' a eşitlenecek olan bu C:\MFT\out\{FilePath}{token=-2}{separator=\}\{FileName} olabilir.

İlgili kavramlar

“Customizing MFT resource monitor tasks with variable substitution” sayfa 221

Etkin bir kaynak izleyicisinin tetikleme koşulları yerine getirildiğinde, tanımlanan görev çağrılır. Aktarma ya da komut görevini her seferinde aynı hedef aracıya ya da aynı hedef dosya adıyla çağırmanın yanı sıra, yürütme sırasında görev tanımını da değiştirebilirsiniz. Bunu yapmak için, görev tanımlaması XML ' ine değişken adları ekleyin. İzleme programı, tetikleme koşullarının yerine getirildiğini ve görev tanımlamasının değişken adları içerdiğini saptadığında, değişken adlarını değişken değerleriyle değiştirir ve daha sonra, görevi çağırır.

İlgili görevler

Değişken yerine koyma değeri birden çok dosyanın tek bir dosya adına gitmesine neden olursa ne yapılır?

Kuyruğun izlenmesi ve değişken yerine koyma değerinin kullanılması

You can monitor a queue and transfer messages from the monitored queue to a file by using the **fteCreateMonitor** command. İzlenen kuyruktan okunan ilk iletiden herhangi bir IBM MQ ileti özelliğinin değeri, görev XML tanımlamasındaki yerine koyulabilir ve aktarma davranışını tanımlamak için kullanılır.

Bu görev hakkında

Bu örnekte, kaynak aracıya, QM_VENUS ile bağlantı kuran AGENT_VENUS adı verilir. AGENT_VENUS monitörlerinin START_QUEUE adı verilen ve QM_VENUS üzerinde bulunan kuyruğun bulunduğu kuyruk. Ajan her 30 dakikada bir kuyruğa girsin.

Kuyruk kuyruğuna tam bir ileti grubu yazıldığında, izleme görevi, bir ileti grubunu bir dosyaya gönderir. Tüm bunlar kuyruk yöneticisinde QM_MARS ' e bağlanan bir hedef araçtan biridir. İletilerin grubunun aktarıldığı dosyanın adı, gruptaki ilk iletide usr . fileName IBM MQ ileti özelliği tarafından tanımlanır. İleti grubunun gönderileceği aracının adı, gruptaki ilk iletide usr . toAgent IBM MQ ileti özelliği tarafından tanımlanır. usr . toAgent üstbilgisi ayarlanmadıysa, hedef aracı için kullanılacak varsayılan değer Agent_magenta değeridir.

useGroups="true" belirttiğinizde, groupId=" \${GROUPID}" belirtmezseniz, aktarma yalnızca kuyruğun ilk iletisine alınır. Örneğin, fileName oluşturmak için değişken yerine koyma değerini kullanıyorsanız, bu nedenle a . txt içeriğinin doğru olmayabileceği bir durum olabilir. Bunun nedeni, izleme programı tarafından fileName tarafından oluşturulduğundan kaynaklanır, ancak aktarma aslında fileName adlı dosyayı oluşturması gereken bir ileti alır.

Yordam

1. İzleme programının, tetiklendiğinde gerçekleştirdiği görevi tanımlayan görev XML ' ini 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 değerleriyle değiştirilen değişkenler, **kalin** olarak vurgulanır. Bu görev XML dosyası /home/USER1/task.xml dosyasına kaydedildi.

2. START_QUEUE kuyruğunu izlemek için bir kaynak izleme programı yaratın.

Aşağıdaki komutu gönderin:

```
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 ileti grubunu START_QUEUE kuyruğuna yazar.

Bu gruptaki ilk iletide şu IBM MQ ileti özellikleri kümesi vardır:

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. Tam grup yazıldığında izleme programı tetiklenir. Aracı, IBM MQ ileti özelliklerini görev XML ' ine koyar.

Bu, görev XML ' inin dönüştürülmekte olduğu görev ile sonuçlanır:

```
<?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 aktarım gerçekleştirilir. START_QUEUE by AGENT_VENUS tarafından START_QUEUE tarafından okunan tam ileti 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 aktarılacaksa, benzer bir tekniği daha önce bu konuda anlatılan bir yöntemle kullanabilirsiniz.

1. Create the monitor as described previously, specifying the **-tr completeGroups** parameter on the **fteCreateMonitor** command.
2. Görev XML ' inde aşağıdakileri belirtin:

```

<queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>

```

Ancak, iletileri kaynak kuyruğa yerleştirdiğinizde, bunları bir IBM MQ grubuna yerleştirmeyin. Her iletiye IBM MQ ileti özellikleri ekleyin. Örneğin, her ileti için benzersiz bir dosya adı değeri olan `usr.filename` özelliğini belirtin. Bu, Managed File Transfer Agent ' in kaynak kuyruğunda her bir iletiyi ayrı bir grup olarak ele almasına neden olur.

İletiyeye dosya aktarımları için izleyici yeniden deneme davranışını yapılandırma

Bir kaynak izleme programı tarafından tetiklenen bir ileti-dosya aktarımı başarısız olursa ve kuyrukta izleme programını tetikleyen ileti grubunu bırakırsa, o aktarma sonraki yoklama aralıklarında yeniden gönderilir. Aktarımın yeniden sunulma sayısı, izleme aracısının **monitorGroupRetryLimit** özelliğiyle sınırlıdır.

Bu görev hakkında

Her yeni iletiyle dosya aktarma işlemi tetiklendiğinde, aktarma görevi için yeni bir aktarma tanıtıcısı yaratılır.

If the agent is restarted the monitor triggers a transfer again even if the number of times the transfer has been triggered has exceeded the value of **monitorGroupRetryLimit** in the `agent.properties` file. **monitorGroupRetryLimit** özelliğinin değeri, kuyruğun üzerinde hala ileti grubu varsa, bir izleme programının bir iletiyi yeniden aktarma işlemini tetiklemesinin en çok kaç kez tetikleneceğini ifade eder. Bu özelliğin varsayılan değeri 10 'tır. Bu özelliğin değeri herhangi bir pozitif tamsayı değerine ya da -1 değerine ayarlanabilir. Bu özellik için -1 değeri belirlenirse, izleme programı, aktarma işlemini tetikleme koşulu karşılanmadıkça, sınırsız sayıda aktarmayı yeniden tetikler.

If a transfer attempt causes the number of times that the transfer has been triggered to exceed the value of **monitorGroupRetryLimit**, the agent writes an error to its event log.

Tek bir ileti tek bir gruptu gibi işlenir ve aktarma her yoklama aralığında yeniden tetiklenir, ancak ileti kuyrukta kalır ve aktarma işlemi tetiklendiğinde, **monitorGroupRetryLimit** değerinden daha küçük bir iletidir.

İzleme aracısında **monitorGroupRetryLimit** özelliğini ayarlamak için aşağıdaki adımları gerçekleştirin:

Yordam

1. Stop the monitoring agent, using the **fteStopAgent** command.
2. İzleme aracı için `agent.properties` dosyasını düzenleyerek aşağıdaki satırı ekleyin:

```
monitorGroupRetryLimit=number_of_retries
```

`agent.properties` dosyası, `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/monitoring_agent_namedizininde` bulunur.

3. Start the monitoring agent, using the **fteStartAgent** command.

İlgili görevler

“Örnek: Bir MFT kaynağının yapılandırılması” sayfa 220

Bir kaynak izleme programı tarafından izlenecek kaynak olarak **-mq** kuyruğunu kullanarak **fteCreateMonitor** komutunu kullanarak IBM MQ kuyruğuspecificykomutunu kullanarak belirtebilirsiniz.

Tetikleme Dosyasının Kullanılması

Tek bir aktarma isteğinde aktarılacak bir kütük kümesi tanımlamak için, kaynak izleme programında bir tetikleme dosyasının içeriğini kullanabilirsiniz. Eşleşen bir tetikleyici dosyası her saptandığında, kaynak dosya yolları için ve isteğe bağlı olarak hedef dosya yolları için içeriği ayrıştırılır. Bu dosya yolları daha sonra, belirttiğiniz görev aktarımı XML dosyasındaki dosya öğelerini tanımlamak için kullanılır; bu dosya, aracıya tek bir aktarma isteği olarak sunulur. Kaynak izleyicisinin tanımlaması, tetikleme içeriğinin etkinleştirilip etkinleştirilmeyeceğini belirler.

-tc (tetikleme içeriği) parametresini belirterek bir izleme programı yaratırken dosya içeriğini tetiklemeyi etkinleştirebilirsiniz. Bu **-tc** parametresi yalnızca dosya tetikleyicisi seçenekleri eşleştir ve noSizeDeğişikliği için geçerlidir. İzleme programı oluşturma hakkında daha fazla bilgi için bkz. **fteCreateMonitor**: bir MFT kaynak izleme programı yaratın.

Bir tetikleyici içerik dosyası kullanılırken, her satırın varsayılan biçimi aşağıdakilerden biri olabilir:

- Tek bir kaynak dosya yolu ya da
- Bir kaynak dosya yolu ve bir hedef dosya yolu, virgülle ayrılmış olarak

Burada beyaz alan karakterlerin, dosya yollarının bir parçası olarak işlendiği yerdir. **fteCreateMonitor** komutundaki **-tcr** ve **-tcc** parametrelerini belirterek varsayılan satır biçimini değiştirmek mümkündür. Daha fazla bilgi için bkz “Gelişmiş seçenekler” sayfa 230.

Bir tetikleyici dosyası ayrıştırıldıktan sonra, dosya yollarının bir listesi oluşturulur ve belirttiğiniz aktarma görevi XML 'ine uygulanır. Tüm izleme programlarında olduğu gibi, aktarma görevi XML 'inin biçimi, **fteCreateTransfer** komutu tarafından oluşturulan, tek bir öğe ya da dosya tanımlı olan eksiksiz bir aktarım görevi XML 'idir. Tek parça, kaynak ve hedef dosya yolları için değiştirme olarak `${contentSource}` ve isteğe bağlı olarak `${contentDestination}` değerini kullanmalıdır. İzleme programı, aktarma görevi XML dosyasını, tetikleme dosyasındaki her bir hat (dosya yolu) için bir dosya ögesi içerecek şekilde genişletir.

-tc parametresi, her bir tetikleyici dosyası için bir aktarma isteği belirttiğinden, **-bs** parametresiyle dosya içeriğini tetikleme işlemi kullanamazsınız.

Örnek

Aşağıdaki örnek, `trig` 'ta sona eren ve o dosyadaki dosya yollarını okuyan bir dosyayı tetikleyecek 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` adı verilen bir dosya yaratır. Örneğin:

```
<item checksumMethod="MD5" mode="binary">
  <source disposition="leave" recursive="false">
    <file>${contentSource}</file>
  </source>
</item>
```

The **fteCreateMonitor** command scans for files that end in `trig` in the `/home/trigdir` directory and uses the contents to create a single transfer request that is based on the `task.xml` for all paths in that trigger file. Tetikleme dosyasının biçimi, virgül ayırıcı içermeyen her satırda tek bir dosya yolu (yalnızca kaynak) olmalıdır. Örneğin:

```
/home/file/first.txt
/home/file/second.txt
/home/different/third.txt
:
```

All files are delivered to the `/file/destdir` directory with its file name and not its file path, that is, `/home/file/first.txt` is delivered to `/file/destdir/first.txt`.

Alternatively, if you change the **-dd /file/destdir** parameter in the **fteCreateTransfer** command to **-df \${contentDestination}** and the format of the content of a trigger file to *kaynak dosya yolu,hedef dosya yolu*, you can define different destination paths for the same destination agent. Örneğin:

```
/home/file/first.txt,/home/other/sixth.txt
```

Hedef konum, sonra `/home/other/sixth.txt` olur.

Yerine koyma değişkenleri simgeleştirilebilir. Örneğin, `${contentDestination{token=-1}}` kullanarak dosya adı kısmını sağlanan yoldan ayırabilirsiniz. Therefore, if the **fteCreateTransfer** destination is defined as **-df /file/destdir/\${contentDestination{token=-1}}**, the new destination for `/home/file/first.txt` is `/file/destdir/sixth.txt`.

Gelişmiş seçenekler

You can change the default line format for the content of the trigger file by using the **-tcr** *regex* parameter. Gerekli satır biçimiyle eşleşen düzenli bir ifade sağlayın ve bir ya da iki yakalama grubu sağlasın. İlk yakalama grubu kaynak ve ikinci, isteğe bağlı, yakalama grubu hedeftir. Örneğin:

- Kaynak ve hedef yol, bir tire işaretiyle ayrılır:

```
((?:[^-]+)-((?:[^-]+))
```

Bu örnekte, ayırıcı üç yerde tanımlanır ve tire, -, herhangi bir karaktere değiştirilebilir. Özel karakterlerden kurtulduğunuzdan emin olun.

- Kaynak ve hedef yollar, sondaki boşlukları içeren bir virgülle ayrılır. Bir sayı işaretiyle (#) gösterilen yorumlar yoksayılır.

```
((?:[^\,]+),((?:[^\,]+)) *(?:#\.*)+
```

Dosya yolları sayı işaretini (#) içeremez. Genellikle bir girdi şu şekildedir: /home/source/from.txt, /home/destination/to.txt # some comment.

-**tcx** parametresini kullanıyorsanız, ifadenin hataları algılayabilmesi ve tetikleyici dosyalarını doğru olarak ayrıştırabilmesi için düzenli ifadenin iyi tasarlanmış ve sınındığından emin olun.

You can reverse the order of the capture by using the **-tcc destSrc** parameter. Bu parametreyi belirtirseniz, ilk yakalama grubu hedef dosya yolu ve ikinci grup kaynak dosya yoludur.

Hataların nasıl işlendiği

Boş tetikleyici dosyası

Tetikleme dosyası boşsa, sonuç dosya aktarımlarından olmaz. Yani, izleme programı bir aktarma isteği yaratır, ancak dosya ögesi belirtilmez.

Dosyayı hata ile tetikle

Tetikleme kütüğündeki herhangi bir giriş beklenen biçime karşı ayrıştırılmazsa, aktarma isteği oluşturulamaz. Bir izleme programı hata günlüğü yayınlandı ve hata, olay günlüğünde de günlüğe kaydedildi. Tetikleme dosyası işlendi olarak işaretlendi ve izleme programı, dosya güncelleninceye kadar dosyayı işlemeyi yeniden denemiyor.

Aktarma görevi XML ' i yanlış eşleşiyor

Aktarma görevi XML 'in tetikleme dosyasıyla eşleşmesi gerekir; bu, aktarma görevi XML 'in hem{contentSource} hem de \${contentDestination} sahip olması durumunda, bu izleme programına ilişkin tüm tetikleyici dosyaları kaynak ve hedef dosya yollarına ve benzer şekilde tersine çevirmelidir. İlk durumda, tetikleyici dosyası yalnızca kaynak dosya yolunu sarf ettiyse, izleme programı \${contentDestination} yerine koyma hatasını bildirir.

Örnekler

Aşağıdaki örnek, bir tetikleme 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
```

-**tcx** parametresi, bir boşluk karakteriyle ayrılmış herhangi bir karakter dizisine ilişkin iki yakalama grubunu tanımlar. **-tcc destSrc** parametresi ve seçeneği, yakalama gruplarının hedef olarak işleneceğini ve 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
-tcx "((?:[^\ ])+) ((?:[^\ ])+)" -tcc destSrc
```

Logging MFT resource monitors

Kaynak izleme programlarıyla ilgili tanımlama bilgilerini günlüğe kaydetmeyi kullanarak edinebilirsiniz.

Bu görev hakkında

Kaynak izleyicisi günlüğe kaydetmeyi denetlemek için **fteSetAgentLogLevel** komutunu ya da `agent.properties` dosyasını kullanarak kaynak izleme programları için günlüğe kaydetme özelliğini kullanabilirsiniz.

Var olan izleme noktalarının bilgileri yakalamak için hala kullanıldığını unutmayın.

Kaynak izleyicisi günlükleri, `resmoneventN.log` adlı bir dosyaya yazılır; burada *N* bir sayı belirtir; örneğin, `resmonevent0.log`. Olay günlüğü dosyaları, bir izleme programı bir kaynak (örneğin, bir dizin ya da kuyruk) yoklarken gerçekleşen birkaç işlemi kaydeder.



Uyarı: Bir aracının tüm kaynak izleme programları aynı günlük dosyasına yazılır.

Bir resmoneventN. Log dosyasının bazı örnek çıkışı için bkz. [MFT dizin kaynağı izleyiciniz dosyaları tetiklemediyse ne yapmanız gerekir?](#).

Aşağıdaki çizelge, kaynak izleyicisinin günlük dosyasına yazdığı olayların tipini listeler. Üçüncü sütunda, en düşük düzeyin INFO ve en yüksek düzeyin VERBOSE olduğu her bir olayı yakalamak için gereken günlük düzeyi açıklanmaktadır.

Daha yüksek bir günlük düzeyi ayarına dikkat edin, alt düzey olaylar da yazar. Örneğin, günlük düzeyi ILIMLI olarak ayarlanıyor, BİLGİ düzeyi olayları da yazar, ancak ayrıntılı düzey olayları değil.

Numara	Olay	Günlük düzeyi	Tanım
1	İzleyici Yaratıldı	BİLGİ	Bir kaynak izleme programı yaratıldı.
2	İzleme Silindi	BİLGİ	Bir kaynak izleyicisi silinmiş.
3	İzleme Durduruldu	BİLGİ	Bir kaynak izleme programı durduruldu.
4	İzleme Başlatıldı	BİLGİ	Bir kaynak izleme programı başlatıldı.
5	Yoklamayı Başlat	BİLGİ	Bir kaynak izleyicisi yeni bir yoklama döngüsü başlattı.
6	Anketi Sona Gönder	BİLGİ	Kaynak izleyicisi yoklama döngüsü sona erdi.
7	Örüntü Eşleşmesi	Ayrıntılı	Tetikleyici izleyicisi dizinindeki bir kütük ya da kuyrukta belirtilen örüntüyle eşleşen bir ileti bulundu.
8	Kalip Uyuşmazlığı	Ayrıntılı	Tetikleme izleme dizinindeki eşleşmeyen dosya ya da kuyrukta belirtilen örüntüyle eşleşmeyen bir ileti bulundu.
9	Aktarma İsteği	BİLGİ	Kaynak izleyicisi tarafından bir aktarma işlemi başlatıldı.
10	Dizin çok derin	Ayrıntılı	Kaynak izleme programı tarafından izlenen dizin, yoklama için kaynak izleyicisi yapısınındaki belirlenen sayıdan daha fazla alt dizin içeriyor.
11	Dosya kilitlendi	orta düzeyde	Kaynak izleme programı tarafından izlenen tetikleyici dosyası başka bir işlem tarafından kilitlenmiş.
12	Dosya boyutu küçük	orta düzeyde	Tetikleyici dosyası, kaynak izleyicisi yapısınındaki belirtilen boyuttan küçük.
13	Dosya boyutu dengesiz	orta düzeyde	Tetikleme dosyası, kaynak izleyicisi yapısınından beklenenden daha sık değiştiriliyor.
14	Çok fazla sandık var	orta düzeyde	Bir kaynak izleyicisi dengesiz bir tetikleme dosyasını çok kez yokladı.
15	Öğeler Eşleştirildi	BİLGİ	Kaynak izleme programı tarafından kullanılan 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 oluşturdu.

Numara	Olay	Günlük düzeyi	Tanım
18	Aktarma İsteği	BİLGİ	Aktarma isteği kaynak izleme programı tarafından gönderildi.
19	İzleme Programı Başlatma Başarısız Oldu	orta düzeyde	Kaynak izleme programı başlatılamadı.
20	Temizlenen Geçmiş	BİLGİ	İzleme geçmişi bilgileri temizlendi.
21	İzleme Geçmişini Temizleme Başarısız Oldu	BİLGİ	İzleme geçmişi bilgilerini temizleme girişi başarısız oldu.
22	Aktarma Tanıtıcısı	BİLGİ	Aktarma isteğinin tanıtıcısı izleme programı tarafından gönderildi.
23	Toplu İş	BİLGİ	Eşleştirilen öğelere ilişkin aktarma isteklerinin toplam sayısı: N , burada N bir sayıdır.

Yordam

- Kaynak izleyicisi günlük kaydını açmak ve kapatmak için **fteSetAgentLogLevel** ' i kullanmak üzere, **logMonitor** değiştirgesinin açıklaması için [fteSetAgentLogDüzyey](#) konusuna bakın ve farklı seçenekleri nasıl kullandığınızı gösteren örnekler.
- Kaynak izleyicisi günlüğe kaydetme özelliğini denetlemek üzere `agent.properties` dosyasını kullanmak için, aşağıdaki günlüğe kaydetme etkinliklerini gerçekleştirmenize olanak tanıyan ek özelliklerin bir açıklaması için bkz. [MFT agent.properties dosyası](#) :
 - Oturum açmayı aç ya da kapat
 - Her bir günlük dosyasının boyutunu sınırla
 - Kaynak izleyicilerinin oluşturabileceği günlüklerin sayısını sınırla

Örnek

The following sample message sets 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

[fteSetAgentLogDüzyey](#) komutu

[The MFT agent.properties file](#)

V 9.2.2 MFT Kaynak İzleme Programının Başlatılması

IBM MQ 9.2.2' 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 kısıtlanması](#) 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 araçta kaynak izleyici başlatmak için **fteStartMonitor** komutunu aşağıdaki gibi girin:

```
fteStartMonitor -mn monitor_name -ma agent_name
```

- Farklı bir makinede çalışan bir araçta 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** değiştirgesi isteğe bağlıdır; tersi durumda, aracı kuyruk yöneticisini **-mm** değiştirgesiyle 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 210

Managed File Transfer kaynak izleme özelliğine ilişkin temel kavramlara genel bakış.

İlgili görevler

“MFT Kaynak İzleme Programının Durdurulması” sayfa 235

IBM MQ 9.2.2' den **fteStopMonitor** komutunu kullanarak bir aracıyı durdurmaya ya da yeniden başlatmaya gerek kalmadan kaynak izleyicilerini durdurabilirsiniz.

İlgili başvurular

[fteStartİzleme Programı \(MFT kaynak izleme programının başlatılması\)](#)

IBM MQ 9.2.2' 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 dizesel 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 çalıştırdıktan 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 araçta durdurmak için **fteStopMonitor** komutunu aşağıdaki gibi girin:

```
fteStopMonitor -mn monitor_name -ma agent_name
```

- Farklı bir makinede çalışan bir araçta 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 210

Managed File Transfer kaynak izleme özelliğine ilişkin temel kavramlara genel bakış.

İlgili görevler

“MFT Kaynak İzleme Programının Başlatılması” sayfa 233

IBM MQ 9.2.2' 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 değiştirgesi 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 ya da her kaynak izleyicisi için **fteListMonitors -ox** komutunu çalıştırmak üzere ayrı bir komut dosyası kullanmak yerine, kullanışlıdır.

Yordam

- Bir kaynak izleyicisinin tanımlamasını bir XML dosyasına aktararak yedeklemek için aşağıdaki komutlardan birini kullanın:
 - **-ox** parametresiyle **fteCreateMonitor** komutu.
 - **-ox** parametresiyle **fteListMonitors** komutu.
- **-ox** değiştirgesini kullanırken, aşağıdaki örnekte gösterildiği gibi **-ma** ve **-mn** değiştirgelerini 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 gösterildiği gibi **fteListMonitors** komutunu **-od** değiştirilmesiyle birlikte kullanın:

```
fteListMonitors -od /usr/mft/resmonbackup
```

Kaynak izleyicilerini toplu olarak yedeklerken geçerli bir hedef dizin belirtmelisiniz. Hedef yol belirtilmemesi, aşağıdaki örnekte gösterildiği gibi bir hata iletiliştir:

BFGCL0762E: Çıkış dizini belirtilmedi. Geçerli bir yol belirterek komutu yeniden çalıştırın.

-od parametresi **-ox** parametresiyle birleştirilmemelidir; tersi durumda, aşağıdaki hata iletiliştir:

BFGCL0761E: Hem '-od', hem de '-ox' değiştiricilerini birlikte belirtmek geçerli değildir.

Yedeğin içereceği belirli bir kaynak izleme programı kümesi tanımlayabilirsiniz. Örneğin, bir aracının adını belirtmek için **-ma** değiştiricisini kullanarak, aşağıdaki örnekte gösterildiği gibi, o aracıya ilişkin tüm kaynak izleyicilerini yedekleyebilirsiniz:

```
fteListMonitors -ma AGENT1 -od /usr/mft/resmonbackup
```

Aracı adlarını ve/veya izleme programı adlarını eşleştirmek için kullanılacak bir kalıp tanımlarken yıldız işareti (*) karakteri ekleyerek genel arama karakteri eşleştirmesini de kullanabilirsiniz. Aşağıdaki örnek, belirli bir kalıpla eşleşen ve adı belirli bir kalıpla eşleşen tüm kaynak izleyicilerini yedekler:

```
fteListMonitors -ma AGENT* -mn MON* -od /usr/mft/resmonbackup
```

Komut çalışırken, aşağıdaki ilerleme raporu iletilerini görüntüler:

Toplam *sayı* eşleşen kaynak izleyicisi tanımlaması bulundu.
Dosya sistemine saklanan *sayı* kaynak izleme programı tanımlamasının *dizini* .

Ayrıntılı seçeneği 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 belirli bir dizindeki bir XML dosyasına aktararak bir kaynak izleyicisini yedeklemek için, **-od** değiştirilmesiyle **fteListMonitors** komutunu kullanın:

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -od /usr/mft/resmonbackup
```

-od değiştiricisinin tek bir kaynak izleme programını yedeklemek için kullanılması, çıkış kütüğü adının *agent name.monitor name.xml* biçiminde olması dışında, **-ox** değiştiricisinin kullanılmasına benzer.

- Kaynak izleyicisi tanımlamalarını bir yedekten geri yüklemek için, aşağıdaki örnekte gösterildiği gibi **fteCreateMonitor** komutunu **-ix** değiştirilmesiyle birlikte kullanın:

```
fteCreateMonitor -ix file name
```

-od değiştiricisinin kullanılmasına ilişkin ek örnekler için [fteListMonitors: list MFT resource itors](#) (İzleyiciler: liste kaynak izleyicileri) konusuna bakın.

İlgili başvurular

fteCreateMonitor: MFT kaynak izleme programı yaratılması

[fteList](#) İzleme Programları: liste MFT kaynak izleme programları

V9.2.0 Kaynak izleyicisi geçmişini temizliyor

Bir hata nedeniyle daha önce aktarılmamış bir dosya için başka bir dosya aktarma isteğinin sunulabilmesi için kaynak izleme programının geçmişini temizleyebilirsiniz. Kaynak izleyicisi geçmişini 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 geçmişini temizleyen kullanıcının aşağıdaki çizelgede gösterildiği gibi uygun yetkiye sahip olması gerekir.

İzleyici geçmişini 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 geçmişini 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 210).

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 iletisi 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 iletisi görüntülenir. Örneğin:

```
BFGUI0016E Belirtilen kaynak izleme programının geçmişi temizlenemedi-neden: 2059
```

Dosya aktarma şablonlarıyla çalışılması

Yinelenen ya da karmaşık aktarımlar için ortak dosya aktarma ayarlarını saklamak için dosya aktarma şablonlarını kullanabilirsiniz. **fteCreateTemplate** komutunu kullanarak komut satırından bir aktarma şablonu yaratın ya da **Yönetilen Dosya Aktarması için Yeni Şablon Yarat** sihirbazını kullanarak bir aktarma şablonu yaratmak için IBM MQ Explorer komutunu kullanın ya da **Aktarma ayarlarını şablon olarak sakla** onay kutusunu seçerek bir dosya aktarma işlemi yaratırken bir şablon kaydedin. **Aktarma Şablonları** penceresi, Managed File Transfer ağınızda yarattığınız aktarma şablonlarının tümünü görüntüler.

Bu görev hakkında

Komut satırından bir aktarma şablonu yaratmak için [fteCreateTemplate](#) komutunu kullanın. Daha sonra, komut satırında oluşturduğunuz bir aktarım şablonunu göndermek istediğinizde, IBM MQ Explorer'deki **Gönder** düğmesini tıklatın.


To view transfer templates in the IBM MQ Explorer, use the following steps:

Yordam

1. Navigator görünümünde **Yönetilen Dosya Aktarması** nesnesini açın. **Yönetilen Dosya Aktarma Merkezi**, İçerik görünümünde görüntülenir.
2. Tüm eşgüdüm kuyruk yöneticilerinizin tümü Navigator görünümünde listelenir. Zamanlanan aktarım için kullandığınız eşgüdüm kuyruk yöneticisinin adını açın. Bağlandığınız eşgüdüm kuyruk yöneticisini değiştirmek istiyorsanız, Navigator görünümünde kullanmak istediğiniz koordinasyon kuyruğu yöneticisinin adını sağ tıklatın ve **Bağlan** 'ı tıklatın.
3. **Şablonları Aktar** 'ı tıklatın. İçerik görünümünde **Aktarma Şablonları** penceresi görüntülenir.
4. **Aktarma Şablonları** penceresi, dosya aktarımlarınızla ilgili aşağıdaki ayrıntıları listeler:
 - a) **Ad** Dosya aktarma şablonunuzun adı.
 - b) **Kaynak** Dosyayı kaynak sistemden aktarmak için kullanılan aracının adı.
 - c) **Kaynak Dosya** Anasistemde aktarılacak kütüğün adı.
Bu alanı görüntülemek için aktarma şablonu bilgilerini genişletin.
 - d) **Hedef** Hedef sistemde dosyayı almak için kullanılan aracıyı adı.

- e) **Hedef Dosya** Hedef sisteme aktarıldıktan sonra dosyanın adı.
Bu alanı görüntülemek için aktarma şablonu bilgilerini genişletin.
- f) **Zamanlanmış Başlatma (seçilen saat dilimi)** Dosya aktarımında, denetimcinin kullandığı saat diliminde başlatılacağı tarih ve saat. Görüntülenen saat dilimini değiştirmek için, **Pencere > Tercihler > IBM MQ Explorer > Managed File Transfer** seçeneklerini tıklatın ve **Saat dilimi:** listesinden bir alternatif saat dilimi seçin. **Tamam'**ı tıklatın.
- g) **Tetikleme Olayları** Dosyayı başlatmak için dosya aktarımı tetikleyen olay tipi. Tip şu değerlerden biri olabilir: var, var değil ya da aşıldı.

Sonuçlar

Aktarma Şablonları penceresinde görüntülenenleri yenilemek için, İçerik görünümü araç çubuğundaki Yenile düğmesini  tıklatın.

Bir aktarma şablonunu göndermek ve şablona tanımlı aktarma işlemini başlatmak için, şablon adını farenin sağ düğmesiyle tıklatın ve **Sun** ögesini seçin.

Bir aktarma şablonunu değiştirmek için, şablon adını farenin sağ düğmesiyle tıklatın ve **Düzenle** ögesini seçin. Özgün şablonda yer alan tüm dosyalar, özgün şablondaki bir grubun parçası olarak eklenmemiş olsa da, aktarma grubunun bir parçası olarak listelenir. Şablondan bir dosyayı kaldırmak istiyorsanız, gruptan dosya belirtimini seçmeniz ve **Seçilmeyi kaldır'**ı tıklatmanız gerekir. Şablona yeni dosya belirtileri eklemek isterseniz, şablon panelindeki alanları kullanın ve **Grubuna ekle** düğmesini tıklatın. Düzenlemelerinizi gerçekleştirdiğinizde, düzenlenen şablona yeni bir ad vermeniz istenir.

Aktarma şablonundan kütük aktarımı yaratmak için, şablon adını farenin sağ düğmesiyle tıklatın ve **Yeni Aktarma Olarak Düzenle** ögesini seçin.

Bir aktarma şablonunun kopyasını yaratmak için, şablon adını farenin sağ düğmesiyle tıklatın ve **Çoğalt** ögesini seçin. Yinelenen aktarım şablonu, özgün şablonla aynı adla otomatik olarak kaydedilir ve "(kopyala)" ile eklenir.

Bir aktarma şablonunu silmek için, şablon adını farenin sağ düğmesiyle tıklatın ve **Sil** ögesini seçin.

İlgili görevler

[“IBM MQ Explorer kullanarak dosya aktarma şablonu yaratılması” sayfa 240](#)

IBM MQ Explorer ' tan ya da komut satırından bir dosya aktarma şablonu yaratabilirsiniz. Daha sonra, şablon ayrıntılarını kullanarak yeni dosya aktarımları yaratmak için bu şablonu kullanabilir ya da dosya aktarımı başlatmak için şablonu sunabilirsiniz.

İlgili başvurular

[fteCreateTemplate](#): yeni dosya aktarma şablonu yarat

[fteListŞablonları](#)

[fteDeleteŞablonları](#)

IBM MQ Explorer kullanarak dosya aktarma şablonu yaratılması

IBM MQ Explorer ' tan ya da komut satırından bir dosya aktarma şablonu yaratabilirsiniz. Daha sonra, şablon ayrıntılarını kullanarak yeni dosya aktarımları yaratmak için bu şablonu kullanabilir ya da dosya aktarımı başlatmak için şablonu sunabilirsiniz.

Bu görev hakkında

Komut satırından bir dosya aktarma şablonu yaratmak için [fteCreateTemplate](#) komutunu kullanın.

IBM MQ Explorer' ta **Yönetilen Dosya Aktarması için Yeni Şablon Yarat** sihirbazını kullanarak bir dosya aktarma şablonu yaratmak için aşağıdaki adımları kullanın:

Yordam

1. Navigator görünümünde, **Yönetilen Dosya Aktarması**' yı tıklatın. **Yönetilen Dosya Aktarma Merkezi** , İçerik görünümünde görüntülenir.
2. Tüm koordinasyon kuyruğu yöneticilerinizin tümü Navigator görünümünde görüntülenir. Zamanlanan aktarım için kullandığınız eşgüdüm kuyruk yöneticisinin adını açın. Bağlandığınız eşgüdüm kuyruk yöneticisini değiştirmek istiyorsanız, Navigator görünümünde kullanmak istediğiniz koordinasyon kuyruğu yöneticisinin adını sağ tıklatın ve **Bağlan'** ı tıklatın.
3. **Aktarma Şablonları** ögesini farenin sağ düğmesiyle tıklatıp **Yeni Şablon** ögesini seçerek **Yönetilen Dosya Aktarması için Yeni Şablon Yarat** sihirbazını başlatın.
4. Sihirbaz panolarındaki yönergeleri izleyin. Her bir pano için bağlama duyarlı yardım sağlanır. Windows' ta bağlama duyarlı yardıma erişmek için F1tuşuna basın. Linux' ta Ctrl+F1 ya da Shift+F1tuşuna basın.

Gereken tüm aktarım ayrıntılarını içeren bir şablon yarattıysa, bu onay kutusu önceden seçilmediyse, **Aktarma Özeti** sayfasındaki **Aktarma ayarlarını şablon olarak sakla** onay kutusunu işaretlediğinizden emin olun. Ayrıca, Ad alanında şablon için bir ad girin. Gerekli aktarım ayrıntılarının tümünü içermeyen bir şablon oluşturursanız, **Aktarma ayarlarını şablon olarak sakla** onay kutusu otomatik olarak sizin için işaretler.

İlgili görevler

“Dosya aktarma şablonlarıyla çalışılması” sayfa 239

Yinelenen ya da karmaşık aktarımlar için ortak dosya aktarma ayarlarını saklamak için dosya aktarma şablonlarını kullanabilirsiniz. **fteCreateTemplate** komutunu kullanarak komut satırından bir aktarma şablonu yaratın ya da **Yönetilen Dosya Aktarması için Yeni Şablon Yarat** sihirbazını kullanarak bir aktarma şablonu yaratmak için IBM MQ Explorer komutunu kullanın ya da **Aktarma ayarlarını şablon olarak sakla** onay kutusunu seçerek bir dosya aktarma işlemi yaratırken bir şablon kaydedin. **Aktarma Şablonları** penceresi, Managed File Transfer ağınızda yarattığınız aktarma şablonlarının tümünü görüntüler.

İlgili başvurular

fteCreateTemplate: yeni dosya aktarma şablonu yarat

[fteListŞablonları](#)

[fteDeleteŞablonları](#)

Dosya aktarma şablonu tanımlamasının yedeklenmesi

Dosya aktarma şablonları, aktarıma ilişkin kaynak ve hedef dosya belirtilmelerini tanımlayan bir XML belgesi içerir. You can use this XML file as input to the **fteCreateTemplate** command to recreate a file transfer template.

Bu görev hakkında

Bir aktarma şablonuna ilişkin kaynak ve hedef kütük belirtilmelerini içeren XML belgesini yedeklemek için, [fteCreateTransfer](#) komutu komutunu ya da IBM MQ Explorer komutunu kullanın. Bir aktarma şablonu XML biçimli yedek dosyası yaratmak için aşağıdaki adımları kullanın:

Yordam

- Yöntem bir: Yeni bir dosyaya aktarma şablonu XML iletisi oluşturmak için bir [fteCreateTransfer](#) komutundaki **-gt** parametresini kullanın.
- Method two: create the template using IBM MQ Explorer.
Aktarma şablonu özeti sayfasına geçtiğinizde:
 - a) *İleti XML önizlemesini iste'* yi kopyalayın.
 - b) Bu aktarma şablonu XML iletisini yeni bir dosyaya saklar.
- Üç yöntem: var olan şablonları yedeklemek için IBM MQ Explorer kullanın.

- a) **Yönetilen Dosya Aktarması** > **Kuyruk Yöneticisi Adı** > **Aktarma Şablonları**' ya gidin.
- b) Aktarma panosunda, yedeklenmeleri gereken şablonu vurgulayın, sağ tıklatın ve açılan menüden **Düzenle** seçeneğini belirleyin.
- c) **Aktarma şablonu özeti** sayfasına geçinceye kadar **İleri** düğmesini tıklatın.
- d) **İleti XML önizlemesini iste'** yi kopyalayın.
- e) Bu aktarma şablonu XML iletisini yeni bir dosyaya saklar.

Sonuçlar

You can use the transfer template XML message file, created by one of the above methods, as input to the `fteCreateŞablonu` command. Bu komutu nasıl kullanmanıza ilişkin ayrıntılar için **fteCreateTemplate** komutuna bakın.

İlgili başvurular

[fteCreateŞablon komutu](#)

[fteListTemplates komutu](#)

Ktklerden iletilere veri aktarma

Bir dosyadan tek bir iletiye ya da birden çok iletiye, bir IBM MQ kuyruğunda veri aktarmak için Managed File Transfer ' un dosya-hedef özelliğini kullanabilirsiniz.

To perform file-to-message and message-to-file transfers both the source and destination agent of the transfer must either be at IBM WebSphere MQ 7.5 , or later, or at IBM WebSphere MQ File Transfer Edition 7.0.3, or later. İletiyeye dosya aktarımları hakkında bilgi için bkz. [“İletilerden dosyalara veri aktarılması” sayfa 250.](#)

Bir dosya için dosya aktarıma ilişkin hedef aracı, bir iletişim kuralı 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 küçük aktarma tipleri desteklenmektedir:

- Tek bir dosyadan tek bir iletiye kadar. İletinin bir IBM MQ grubu tanıtıcısı kümesi yok.
- Tek bir dosyadan birden çok iletiye, dosyayı belirli bir uzunluğun iletilerine böler. 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.
- Onaltılık sınırlayıcıda ikili bir dosyayı bölerek, tek bir dosyadan birden çok iletiye kadar. İletilerin tümü aynı IBM MQ grup tanıtıcısına sahiptir.

Bir ikili dosyayı sınırlayıcı olarak byte dizeleri kullanarak bölmek istiyorsanız, **fteCreateTransfer** komutunun **-sqdb** parametresini kullanın. Ek bilgi için [-sqdb parameterbaşlıklı](#) konuya bakın.

Varsayılan olarak, bir dosya-ileti aktarımı tarafından oluşturulan iletiler kalıcıdır. İletiler kalıcı olmayan ya da hedef kuyruk tarafından tanımlanan kalıcılık değerine sahip olacak şekilde ayarlanabilir.

Bir dosyanın birden çok iletiye bölüneceğini belirtirseniz, dosyadan yaratılan tüm iletilerin aynı IBM MQ grup tanıtıcısına sahip olur. Bir dosyanın birden çok iletiye bölüneceğini belirtmezseniz, dosyadan yalnızca bir ileti oluşturulur ve bu ileti IBM MQ grup kimliği kümesine sahip değildir.

Dosyaları büyük iletilere ya da çok sayıda küçük iletiye aktarıyorsanız, bazı IBM MQ ya da Managed File Transfer özelliklerini değiştirmeniz gerekebilir. Hakkında bilgi için bkz. [İleti boyutu ile ilişkili MQ özniteliklerini ve MFT özelliklerini ayarlama kılavuzu.](#)

Not: Hedef kuyruk kümelenmiş bir kuyruksa ya da kümelenmiş bir kuyruğa ilişkin diğer ad ise, aracı özelliği `enableClusterQueueInputOutput true` olarak ayarlanmadıysa, bir dosyayı kuyruğa aktarırken bir hata iletisi alır. Daha fazla bilgi için bkz. [Hedef kuyruk kümelenmiş bir kuyruksa ya da kümelenmiş bir kuyruğa diğer ad ise ne yapılır](#) konusuna bakın.

Aracıyı, ileti-ileti aktarımları gerçekleştirecek şekilde yapılandırma

Varsayılan olarak, araçları ileti-ileti ya da ileti-dosya aktarımları gerçekleştiremez. Bu işlevi etkinleştirmek için, `enableQueueInputOutput` adlı aracı özelliğini `true` değerine ayarlamalısınız. IBM MQ kümelenmiş kuyruklarına yazmayı etkinleştirmek için, `enableClusterQueueInputOutput` adlı aracı özelliğini de `true` değerine ayarlamalısınız.

Bu görev hakkında

`enableQueueInputOutput` özelliği `true` olarak ayarlanmış bir hedef aracıya bir dosya için ileti aktarımı gerçekleştirmeyi denerseniz, aktarma başarısız olur. Koordinasyon kuyruk yöneticisine yayınlanan aktarma günlüğü iletisi 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.
```

Aracıyı kuyruklardan yazma ve bu kuyruklardan okuma işlemlerini gerçekleştirmek 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. Hedef aracıyı **fteStartAgent** komutunu kullanarak başlatın.

Örnek: Tek bir dosyayı tek bir iletiye aktarma

You can specify a queue as the destination of a file transfer by using the **-dq** parameter with the **fteCreateTransfer** command. Kaynak dosya, hedef kuyruktaki ileti uzunluğu üst sınırı değerinden küçük olmalıdır. Hedef kuyruğun, hedef aracının bağlandığı kuyruk yöneticisiyle aynı kuyruk yöneticisine olması gerekmez, ancak bu iki kuyruk yöneticisi iletişim kurabilmelidir.

Bu görev hakkında

Kaynak dosya adı `/tmp/single_record.txt` olan ve kaynak aracıyla, `AGENT_NEPTUNE` ile aynı sistemde yer alır. Kaynak aracı, `AGENT_NEPTUNE`, `QM_NEPTUNE` kuyruk yöneticisini kullanır. Hedef aracı `AGENT_VENUS` ve bu aracı kuyruk yöneticisi `QM_VENUS` ile bağlantı kurar. Hedef kuyruk, `RECEIVING_QUEUE`, kuyruk yöneticisi `QM_CIR` üzerinde bulunur. `QM_CIVA`, `QM_VENUS` kuyruk yöneticisi ile aynı IBM MQ ağında yer alan ve bunlara erişilebilirler.

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 görevli tarafından kullanılan kuyruk yöneticisine farklı bir kuyruk yöneticisinde yer alıyorsa, **-dq** parametresinin değerini şu biçimde belirtmeniz gerekir: `kuyruk_adi@kuyruk_yonetme_adi`. Değer içinde `@queue_manager_name` değerini belirtmezseniz, hedef aracı hedef kuyruğun hedef aracı

kuyruk yöneticisinde bulunduğunu varsayar. Bu kural dışı durum, enableClusterQueueInputOutput aracı özelliğinin true değerine ayarlandığını göstermektedir. Bu durumda, hedef aracı, kuyruğun nerede olduğunu belirlemek için standart IBM MQ çözme yordamlarını kullanır.

Kaynak aracı, AGENT_NEPTUNE, verileri /tmp/single_record.txt kütüğünden okur ve bu verileri AGENT_VENUS hedef aracısına aktarır. Hedef aracı, AGENT_VENUS, verileri, RECEIVING_QUEUE@QM_CIVA kuyruğunda kalıcı bir iletiye gönderir. İletinin bir IBM MQ grubu tanıtıcısı kümesi yok.

Örnek: Tek bir dosyayı uzunluğa göre birden çok iletiye böler

Bir dosyayı, **fteCreateTransfer** komutunun **-qs** parametresini kullanarak birden çok IBM MQ iletisine bölümlersiniz. Dosya, her biri tek bir iletiye yazılan, değişmez uzunluklu kısımlara bölünmektedir.

Bu görev hakkında

Kaynak dosya /tmp/source.file olarak adlandırılır ve 36 KB boyutlarında olur. Kaynak dosya, AGENT_NEPTUNE kaynak aracıyla aynı sistemde bulunur. Kaynak aracı, AGENT_NEPTUNE, kuyruk yöneticisi QM_NEPTUNE ile bağlantı kurar. Hedef aracı AGENT_CIVA, kuyruk yöneticisi QM_MERKÜL ile bağlantı kurar. Hedef kuyruk, RECEIVING_QUEUE, kuyruk yöneticisi QM_CIR üzerinde de bulunur. Aktarım, kaynak dosyayı 1 KB boyutunda olan bölümlere ayırır ve bu bölümlerin her birini RECEIVING_QUEUE 'daki 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
```

Kaynak aracı, AGENT_NEPTUNE, verileri /tmp/source.file kütüğünden okur ve bu verileri hedef aracıya aktarır, AGENT_CIVA. Hedef aracı, AGENT_CIVA, verileri RECEIVING_QUEUE@QM_CIVA kuyruğunda otuz altı 1 KB 'lık kalıcı iletilerine yazar. Bu iletilerin tümü aynı IBM MQ grup tanıtıcısına sahiptir ve gruptaki son ileti, IBM MQ LAST_MSG_IN_GROUP işaret kümesine sahiptir.

Örnek: Düzenli ifade sınırlayıcısı kullanılarak bir metin dosyasını birden çok iletiye böler

Verili bir Java düzenli ifadesinin her eşleşmesine dosyayı bölerek, tek bir metin dosyasını birden çok iletiye aktarın. Bunu yapmak için **fteCreateTransfer** komutunun **-dqdt** parametresini kullanın.

Bu görev hakkında

Dosya, her biri ayrı bir iletiye yazılan değişken uzunluktaki kısımlara bölünmektedir. Metin dosyası, dosyadaki metnin verili bir düzenli ifayla eşleştiği her noktada bölünür. Kaynak dosya /tmp/ names.text olarak adlandırılır ve aşağıdaki içeriklere sahiptir:

```
Jenny Jones,John Smith,Jane Brown
```

Dosyanın bölüneceğini belirten düzenli ifade, virgül karakteridir (,).

Kaynak dosya, QM_NEPTUNE kuyruk yöneticisine bağlanan AGENT_NEPTUNE kaynak aracıyla aynı sistemde bulunuyor. The destination queue, ALACAK_KUYRUĞU, is located on the queue manager QM_CIVA. QM_CIVA, hedef aracı AGENT_CIVA tarafından da kullanılan kuyruk yöneticidir. Aktarım, kaynak dosyayı bölümlere ayırır ve bu bölümlerin her birini RECEIVING_QUEUE 'da 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
```

Kaynak aracı, AGENT_NEPTUNE, verileri /tmp/names.text kütüğünden okur ve bu verileri hedef aracıya aktarır, AGENT_CIVA. Hedef aracı, AGENT_CIVA, verileri *RECEIVING_QUEUE* kuyruğunda üç kalıcı iletiye yazar. Bu iletilerin tümü aynı IBM MQ grup tanıtıcısına sahiptir ve gruptaki son ileti, IBM MQ *LAST_MSG_IN_GROUP* işaret kümesine sahiptir.

İletilerde bulunan veriler aşağıdaki gibidir.

- İlk mesaj:

```
Jenny Jones
```

- İkinci ileti:

```
John Smith
```

- Üçüncü mesaj:

```
Jane Brown
```

Örnek: Bir metin dosyasını düzenli ifade sınırlayıcısıyla bölme ve iletilerde sınırlayıcı da dahil olmak üzere

Verili bir Java düzenli ifadesinin her eşleşmesinde dosyayı böleterek, birden çok iletiye tek bir metin dosyası aktarın ve sonuçtaki iletilerin düzenli ifade eşleşmesini ekleyin. Bunu yapmak için **fteCreateTransfer** komutuna ilişkin **-dqdt** ve **-qi** parametrelerini kullanın.

Bu görev hakkında

Kuyruklardaki birden çok iletiye tek bir metin dosyası aktarabilirsiniz. Dosya, her biri ayrı bir iletiye yazılan değişken uzunluktaki kısımlara bölünmektedir. Metin dosyası, dosyadaki metnin verili bir düzenli ifayla eşleştiği her noktada bölünür. Kaynak dosya /tmp/customers.text olarak adlandırılır ve aşağıdaki içeriklere 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
```

The regular expression that specifies where to split the file is `Customer\snumber:\s\d+`, which matches the text "Müşteri numarası: " followed by any number of digits. Komut satırında belirtilen düzenli ifadelerin, düzenli ifadeyi değerlendirmesini önlemek için çift tırnak işareti içine alınması gerekir. Düzenli ifade, Java düzenli ifadesi olarak değerlendirilir. Daha fazla bilgi için bkz. [MFT tarafından kullanılan düzenli ifadeler](#).

Varsayılan olarak, bir düzenli ifadenin eşleştirebileceği karakter sayısı beş olarak ayarlanır. Bu örnekte kullanılan düzenli ifade, beş karakterden uzun dizgilerle eşleşir. Beş karakterden daha uzun olan eşleşmeleri etkinleştirmek için aracı özellikleri dosyasını **maxDelimiterMatchLength** özelliğini içerecek şekilde düzenleyin.

Varsayılan olarak, düzenli ifayla eşleşen metin iletilerde yer almaz. Bu örnekte olduğu gibi, iletilerde düzenli ifadenin eşleşen metni dahil etmek için **-qi** parametresini kullanın. Kaynak dosya, QM_NEPTUNE

kuyruk yöneticisine bağlanan kaynak aracı AGENT_NEPTUNE ile aynı sistemde bulunur. Hedef kuyruk, RECEIVING_QUEUE, kuyruk yöneticisi QM_CIR üzerinde bulunur. QM_CIVA, aynı zamanda hedef aracı AGENT_CIVA tarafından kullanılan kuyruk yöneticidir. Aktarım, kaynak dosyayı bölümlere ayırır ve bu bölümlerin her birini RECEIVING_QUEUE ' da bir iletiye yazar.

Yordam

1. Aşağıdaki komutu kullanarak hedef aracıyı durdurun:

```
fteStopAgent AGENT_MERCURY
```

2. AGENT_CIVA için aracı özellikleri dosyasına aşağıdaki satırı ekleyin:

```
maxDelimiterMatchLength=25
```

Not: `maxDelimiterMatchLength` değerinin artırılması başarımı 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
```

Kaynak aracı, AGENT_NEPTUNE, verileri /tmp/customers.text kütüğünden okur ve bu verileri hedef aracıya aktarır, AGENT_CIVA. Hedef aracı, AGENT_CIVA, verileri RECEIVING_QUEUE kuyruğunda üç kalıcı iletiye yazar. Bu iletilerin tümü aynı IBM MQ grup tanıtıcısına sahiptir ve gruptaki son ileti, IBM MQ LAST_MSG_IN_GROUP işaret kümesine sahiptir.

İletilerde bulunan veriler aşağıdaki gibidir.

- İlk mesaj:

```
Customer name: John Smith  
Customer contact details: john@example.net  
Customer number: 314
```

- İkinci ileti:

```
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
```

Example: Setting IBM MQ message properties on a file-to-message transfer

You can use the `-qmp` parameter on the `fteCreateTransfer` command to specify whether IBM MQ message properties are set on the first message written to the destination queue by the transfer. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesine ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiyle ilgili bilgileri almak için izin verir.

Bu görev hakkında

fteCreateTransfer komutuna `-qmp true` parametresini ekleyin. Bu örnekte, komutu gönderen kullanıcının MQMD kullanıcı kimliği `larmer` olur.

Yordam

Aşağıdaki 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.WMQFTETransferId=414cbaedefa234889d999a8ed09782395ea213ebbc9377cd
usr.WMQFTETransferMode=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
```

Örnek: Dosya-ileti aktarımı için kullanıcı tanımlı özelliklerin ayarlanması

Kullanıcı tanımlı meta veriler, aktarma işlemi tarafından hedef kuyruğa yazılan ilk iletide bir IBM MQ ileti özelliği olarak ayarlanır. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesine ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiyle ilgili bilgileri almak için etkinleştirmesini 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
```

In addition to the standard set of IBM MQ message properties, the user-defined property is set in the message header of the first message written by the destination agent, `AGENT_SATURN`, to the queue, `MY_QUEUE`, on queue manager, `MyQM`. Üstbilgi, aşağıdaki değere ayarlanır:

```
usr.account=123456
```

`usr` öneki, kullanıcı tanımlı meta veri adının başına eklenir.

Örnek: Dosya-ileti aktarımı için kullanıcı tanımlı bir ileti özelliği ekleme

İleti-dosya tarafından 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 birini kullanabilirsiniz:

- Aktarma isteğindeki **-md** parametresini belirleyin. Daha fazla bilgi için, bkz. [“Örnek: Dosya-ileti aktarımı için kullanıcı tanımlı özelliklerin ayarlanması” sayfa 247.](#)
- Bir Ant görevi kullanın; fte: filecopy ya da fte:filemove seçeneğini kullanabilirsiniz. Aşağıdaki örnek bir fte: filecopy görevi:

```
<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 izleme programı ve değişken yerine koyma değeri kullanın. Aşağıdaki örnek, bazı aktarma görevi XML ' lerini gösterir:

```
<?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/MonitorDefinit ion ./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

[“Example: Setting IBM MQ message properties on a file-to-message transfer” sayfa 246](#)

You can use the **-qmp** parameter on the **fteCreateTransfer** command to specify whether IBM MQ message properties are set on the first message written to the destination queue by the transfer. IBM MQ ileti özellikleri, bir uygulamanın işlenecek iletileri seçmesine ya da IBM MQ Message Descriptor (MQMD) ya da MQRFH2 üstbilgilerine erişmeden bir iletiyle ilgili bilgileri almak için izin verir.

İlgili başvurular

[fte: filecopy Ant görevi](#)

[fte: filemove Ant görevi](#)

Dosya-ileti aktarımı başarısız oldu

Aracı, dosya verilerini hedef kuyruğa yazma işlemini başlattıktan sonra, bir dosyayla ileti aktarımı başarısız olursa, aracı, bir hatanın olduğu iletileri tüketen bir uygulamaya belirtmek için kuyruğa bir ileti yazar.

Bir hata oluşursa, hedef kuyruğa yazılan ileti:

- Boşluk
- Aracı tarafından hedef kuyruğa yazılan önceki iletiyle aynı IBM MQ grup tanıtıcısına sahip
- IBM MQ LAST_MSG_IN_GROUP işaret kümesine sahiptir
- İleti özellikleri etkinleştirilmişse, ek IBM MQ ileti özellikleri içerir. For more information, see the topic [Hedef kuyruklara yazılan iletilerde MFT tarafından ayarlananMQ ileti özellikleri](#).

Örnek

Aşağıdaki komut çalıştırılarak bir aktarma isteğinde bulunulması gerekir:

```

fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq RECEIVING_QUEUE
-qmp true -qs 1K /tmp/source1.txt

```

source1.txt dosyası 48 KB 'dir. Aktarma, bu dosyayı 1 KB 'lik iletilere böler ve bu iletileri hedef kuyruk RECEIVING_QUEUE 'ye yazar.

Aktarma işlemi devam ederken, aracı RECEIVING_QUEUE için 16 ileti yazdıktan sonra, kaynak aracıda bir hata oluşur.

Aracı, RECEIVING_QUEUE 'ye boş bir ileti yazar. Standart ileti özellikleri kümesinin yanı sıra, boş ileti aşağıdaki ileti özellikleri kümesine ayarlanmıştır:

```
usr.WMQFTEResultCode = 40  
usr.WMQFTESupplement = BFGTR0036I: The transfer failed to complete successfully.
```

V 9.2.2 **V 9.2.0.2** From IBM MQ 9.2.0 Fix Pack 2 ve IBM MQ 9.2.2, when a transfer from a file fails, because of a delimiter size check error, only one empty message is sent. Ayrıca, hedef aracıdaki küme büyüklüğünü aşan sınırlayıcıya bağlı olarak aktarma başarısızlığı nedeniyle kullanıcı özellikleri de bu iletiye eklenir.

İletilerden dosyalara veri aktarılması

The message-to-file feature of Managed File Transfer enables you to transfer data from one or more messages on an IBM MQ queue to a file, bir veri kümesi (z/OSüzerinde), or a user file space. If you have an application that creates or processes IBM MQ messages, you can use the message-to-file capability of Managed File Transfer to transfer these messages to a file on any system in your Managed File Transfer network.

Dosya iletiyle aktarılmasıyla ilgili bilgi için bkz. "[Ktklerden iletilere veri aktarma](#)" sayfa 242.



Uyarı: Bir iletiyle dosya aktarımı için kaynak aracı, bir iletişim kuralı köprüsü aracı ya da bir Connect:Direct köprü aracı olamaz.

IBM MQ ileti verilerini bir dosyaya aktarabilirsiniz. Aşağıdaki ileti tipleri desteklenmektedir:

- Tek bir iletiden tek bir dosyaya
- Birden çok iletiden tek bir dosyaya
- Aynı IBM MQ grup tanıtıcısına sahip birden çok iletiden tek bir dosyaya sahip.
- Dosyaya yazılan her iletiden veriler arasında bir metin ya da ikili sınırlayıcı da dahil olmak üzere, birden çok iletiden tek bir dosyaya.

Büyük iletilerden ya da birçok küçük iletiden dosya aktarıyorsanız, bazı IBM MQ ya da Managed File Transfer özelliklerini değiştirmeniz gerekebilir. Hakkında daha fazla bilgi için bkz. [İleti boyutu ile ilişkili MQ özniteliklerinin ve MFT özelliklerinin ayarına ilişkin yönergeler](#).

IBM MQ 9.1.0'tan, dosya aktarımında, kaynak aracıya, IBM MQ' un önceki sürümlerindeki yıkıcı olmayan GET ' in tersine, kaynak kuyruktan gelen iletileri göz atar. İletiler, tüm iletilerden (ileti gruplaması kullanılıyorsa, bir gruptaki) göz atıldıktan ve hedef dosyaya yazılan veriler sonrasında kaynak kuyruktan kaldırılır. Bu, aktarma başarısız olursa ya da iptal edildiyse, iletilerin kaynak kuyruktan kalmasına izin verir. Bu değişiklik nedeniyle, dosya aktarımlarına ileti çalıştırma yetkisi de ile birlikte BROWSE yetkisi de sağlanmalıdır.

From IBM MQ 9.0.0 Fix Pack 2 and IBM MQ 9.0.4, Managed File Transfer is updated to restore the comparison check, previously removed by [APAR IT18213](#), of the transfer identifier and the value of the groupId attribute within the transfer request XML payload. Bu iki tanıtıcı eşdeğerse, kaynak aracı, ileti-to-file aktarımı için giriş kuyruğunda yapılan ilk MQGET denemesi için tanıtıcı eşleşme seçeneği olarak (bir grup tanıtıcısı eşleştirme seçeneğinin aksine) tanıtıcıyı kullanır.

Aracıyı dosya aktarımları gerçekleştirecek şekilde yapılandırma

Varsayılan olarak, araçlar dosyaya ya da dosyaya ileti, aktarımlar için ileti gerçekleştiremez. Bu işlevi etkinleştirmek için, enableQueueInputOutput aracı özelliğini true değerine ayarlamanız gerekir.

Bu görev hakkında

enableQueueInputOutput özelliği true olarak ayarlanmış bir kaynak aracıdan dosya aktarımı için ileti gerçekleştirmeye çalışırsanız, aktarma başarısız olur. Koordinasyon kuyruk yöneticisine yayınlanan aktarma günlüğü ileti 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.
```

Aracıyı kuyruklardan yazma ve bu kuyruklardan okuma işlemlerini gerçekleştirmek 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. **fteStartAracısı** komutunu kullanarak kaynak aracıyı başlatın.

Örnek: Kuyruktan tek bir dosyaya aktarma

You can specify an IBM MQ queue as the source of a file transfer by using the **-sq** parameter with the **fteCreateTransfer** command.

Bu görev hakkında

Kaynak veriler, `START_QUEUE` kuyruğunda üç iletide yer alır. Bu kuyruk, kaynak aracının kuyruk yöneticisinde (`QM_NEPTUNE`) olmalıdır.

Yordam

Aşağıdaki komutu yazın:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE
                 -da AGENT_VENUS -df /out/three_to_one.txt
                 -sq START_QUEUE
```

The data in the messages on the queue `START_QUEUE` is written to the file `/out/three_to_one.txt` on the system where `AGENT_VENUS` is running.

Örnek: Kuyruktan bir ileti grubunun tek bir dosyaya aktarılması

You can specify a single complete group on an IBM MQ queue as the source of a file transfer by using the **-sq** and **-sqgi** parameters with the **fteCreateTransfer** command.

Bu görev hakkında

Bu örnekte, kuyrukta `START_QUEUE` kuyruğunda on ileti vardır. Bu kuyruk, kaynak aracının kuyruk yöneticisinde (`QM_NEPTUNE`) olmalıdır. İlk üç ileti, IBM MQ grup kimliği `41424b3ef3a22020202020202020202020202020202020201111` olan bir gruba ait; bu grup tam bir grup değil. Sonraki beş ileti, IBM MQ grup kimliği `41424b3ef3a220202020202020202020202020202020202222` olan bir gruba aittir; bu grup tamamdır. Kalan iki ileti, IBM MQ grup kimliği `41424b3ef3a220202020202020202020202020202020203333` olan bir gruba aittir; bu grup tamamdır.

Yordam

Aşağıdaki komutu yazın:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS
                 -df /out/group.txt -sqgi -sq START_QUEUE
```

The data in the messages belonging to the first complete group on the queue *START_QUEUE*, the group with IBM MQ group ID 41424b3ef3a220222, is written to the file */out/group.txt* on the system where *AGENT_VENUS* is running.

Örnek: Her iletiden veri önüne bir metin sınırlayıcı ekleme

When you are transferring in text mode from a source queue to a file, you can specify that a text delimiter is inserted before the data from individual messages by using the **-sq**, **-sqdt** and **-sqdp** parameters with the **fteCreateTransfer** command.

Bu görev hakkında

Bu örnekte, *START_QUEUE* kuyruğunda dört ileti vardır. Bu kuyruk, kaynak aracının kuyruk yöneticisin; *QM_NEPTUNE*' dir. Her iletiden gelen verilerin önüne eklenecek metin sınırlayıcısı, 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, *START_QUEUE* tarafından *START_QUEUE*, *AGENT_NEPTUNE* tarafından *START_QUEUE* üzerindeki her dört iletinin her birinden başlayarak veri başına eklenir. Bu veriler hedef dosyaya (*/out/output.txt*) yazılıdır.

Örnek: Her iletiden gelen veriler arasında ikili sınırlayıcı ekleme

When transferring in binary mode from a source queue to a file, you can specify that a binary delimiter is inserted after the data from individual messages by using the **-sq**, **-sqdb**, and **-sqdp** parameters with the **fteCreateTransfer** command.

Bu görev hakkında

Bu örnekte, *START_QUEUE* kuyruğunda üç ileti vardır. Bu kuyruk, kaynak aracının kuyruk yöneticisin; *QM_NEPTUNE*' dir. Her iletiden gelen verilerin yerleştirileceği ikili sınırlayıcı, virgülle ayrılmış bir onaltılı byte 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, kaynak aracı *AGENT_NEPTUNE* tarafından *START_QUEUE* ile ilgili üç iletinin her birinden verilerin sonuna eklenir. Bu veriler hedef dosyaya (*/out/binary.file*) yazılıdır.

Kuyruğun izlenmesi ve değişken yerine koyma değerinin kullanılması

You can monitor a queue and transfer messages from the monitored queue to a file by using the **fteCreateMonitor** command. İzlenen kuyruktan okunan ilk iletiden herhangi bir IBM MQ ileti özelliğinin değeri, görev XML tanımlamasındaki yerine koyulabilir ve aktarma davranışını tanımlamak için kullanılır.

Bu görev hakkında

Bu örnekte, kaynak aracıya, QM_VENUS ile bağlantı kuran AGENT_VENUS adı verilir. AGENT_VENUS monitörlerinin START_QUEUE adı verilen ve QM_VENUS üzerinde bulunan kuyruğun bulunduğu kuyruk. Ajan her 30 dakikada bir kuyruğa girsin.

Kuyruk kuyruğuna tam bir ileti grubu yazıldığında, izleme görevi, bir ileti grubunu bir dosyaya gönderir. Tüm bunlar kuyruk yöneticisinde QM_MARS ' e bağlanan bir hedef aracıdan biridir. İletilerin grubunun aktarıldığı dosyanın adı, gruptaki ilk iletide `usr.fileName` IBM MQ ileti özelliği tarafından tanımlanır. İleti grubunun gönderileceği aracının adı, gruptaki ilk iletide `usr.toAgent` IBM MQ ileti özelliği tarafından tanımlanır. `usr.toAgent` üstbilgisi ayarlanmadıysa, hedef aracı için kullanılacak varsayılan değer `Agent_magenta` değeridir.

`useGroups="true"` belirttiğinizde, `groupId="{GROUPID}"` belirtmezseniz, aktarma yalnızca kuyruğun ilk iletisine alınır. Örneğin, `fileName` oluşturmak için değişken yerine koyma değerini kullanıyorsanız, bu nedenle `a.txt` içeriğinin doğru olmayabileceği bir durum olabilir. Bunun nedeni, izleme programı tarafından `fileName` tarafından oluşturulduğundan kaynaklanır, ancak aktarma aslında `fileName` adlı dosyayı oluşturması gereken bir ileti alır.

Yordam

1. İzleme programının, tetiklendiğinde gerçekleştirdiği görevi tanımlayan görev XML ' ini 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 değerleriyle değiştirilen değişkenler, **kalin** olarak vurgulanır. Bu görev XML dosyası `/home/USER1/task.xml` dosyasına kaydedildi.

2. START_QUEUE kuyruğunu izlemek için bir kaynak izleme programı yaratın.

Aşağıdaki komutu gönderin:

```
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 ileti grubunu START_QUEUE kuyruğuna yazar.

Bu gruptaki ilk iletide şu IBM MQ ileti özellikleri kümesi vardır:

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. Tam grup yazıldığında izleme programı tetiklenir. Aracı, IBM MQ ileti özelliklerini görev XML ' ine koyar. Bu, görev XML ' inin dönüştürülmekte olduğu görev ile sonuçlanır:

```

<?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 aktarım gerçekleştirilir. START_QUEUE by AGENT_VENUS tarafından START_QUEUE tarafından okunan tam ileti 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 aktarılacaksa, benzer bir tekniği daha önce bu konuda anlatılan bir yöntemle kullanabilirsiniz.

1. Create the monitor as described previously, specifying the **-tr completeGroups** parameter on the **fteCreateMonitor** command.
2. Görev XML ' inde aşağıdakileri belirtin:

```

<queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>

```

Ancak, iletileri kaynak kuyruğa yerleştirdiğinizde, bunları bir IBM MQ grubuna yerleştirmeyin. Her iletiye IBM MQ ileti özellikleri ekleyin. Örneğin, her ileti için benzersiz bir dosya adı değeri olan `usr.filename` özelliğini belirtin. Bu, Managed File Transfer Agent ' in kaynak kuyruğunda her bir iletiyi ayrı bir grup olarak ele almasına neden olur.

Örnek: IBM MQ ileti özellikleri kullanılarak bir iletiyle dosya aktarımı başarısız oldu

`usr.ReturnCode` IBM MQ ileti özelliğini sıfır dışında bir değere ayarlayarak, aktarma işleminin başarısız olması için bir iletinin başarısız olmasına neden olabilirsiniz. You can also specify supplementary information about the reason for the failure by setting the `usr.Supplement` IBM MQ message property.

Bu görev hakkında

Bu örnekte, INPUT_QUEUE kuyruğu ile /home/user/output.filekütüğü arasında bir aktarma işlemi devam ediyor.

Bir kullanıcı ileti yaratıyor ve bunları INPUT_QUEUE kuyruğunda yerleştiriyor. Kaynak aracı, INPUT_QUEUE kuyruğundan ileti tüketiyor ve aktarma verilerini hedef aracıya gönderiyor. Hedef aracı, bu verileri /home/user/output.filedosyasına yazıyor.

INPUT_QUEUE kuyruğuna ileti yazan kullanıcı, devam etmekte olan aktarımı durdurmak istiyor ve hedef dosyaya önceden yazılmış olan verileri siler.

Yordam

1. Kullanıcı, aşağıdaki IBM MQ ileti özellikleri kümesine sahip kuyruk 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 kuyruktan ileti işlemeyi durdurur. Hedef aracı, hedef dizine yazılmış olan tüm dosya verilerini siler.
3. Kaynak aracı, aktarma hatasını raporlayan eşgüdüm kuyruk yöneticisine bir aktarma günlüğü iletisi gönderir.

İleti aşağıdaki bilgileri içerir:

```
<?xml version="1.0" encoding="UTF-8"?>  
<transaction version="1.00"  
      ID="414d5120514d3120202020202020202020202020e970d4920008702" agentRole="sourceAgent"  
      xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"  
      xsi:noNamespaceSchemaLocation="TransferLog.xsd"  
      xmlns="">  
  <action time="2008-11-02T21:28:09.593Z">progress</action>  
  <sourceAgent agent="FTEAGENT" QMgr="QM1">  
    <systemInfo architecture="x86" name="Windows 7"  
      version="6.1 build 7601 Service Pack 1" />  
  </sourceAgent>  
  <destinationAgent agent="FTEAGENT" QMgr="QM1">  
    <systemInfo architecture="x86" name="Windows 7"  
      version="6.1 build 7601 Service Pack 1" />  
  </destinationAgent>  
  <originator>  
    <hostName>reportserver.com</hostName>  
    <userID>USER1</userID>  
    <mqmdUserID>USER1 </mqmdUserID>  
  </originator>  
  <transferSet index="0" size="1"  
    startTime="2008-11-02T21:28:09.281Z"  
    total="1">  
    <item mode="binary">  
      <source>  
        <queue>INPUT_QUEUE@QM1</queue>  
      </source>  
      <destination exist="error">  
        <file>/home/user/output.file</file>  
      </destination>  
      <status resultCode="1">  
        <supplement>Cancelling transfer - sent wrong data.</supplement>  
      </status>  
    </item>  
  </transferSet>  
</transaction>
```

Protokol köprüsü

İletişim kuralı köprüsü, Managed File Transfer (MFT) ağınızın, yerel etki alanınızda ya da uzak bir konumdaki MFT ağınızın dışında bir dosya sunucusunda saklanan dosyalara erişmesini sağlar. Bu dosya sunucusu, FTP, FTPS ya da SFTP ağ protokollerini kullanabilir. Her dosya sunucusunun en az bir adanmış aracısına gerek vardır. Adanmış aracı, protokol köprüsü aracısı olarak bilinir. Bir köprü aracısı, birden çok dosya sunucusuyla etkileşimde bulunabilir.

İletişim kuralı köprüsü, Managed File Transfer' un Hizmet bileşeninin bir parçası olarak kullanılabilir. Farklı dosya sunucularına bağlanan, MFT çalıştıran tek bir sistemde birden çok adanmış aracıya sahip olabilirsiniz.

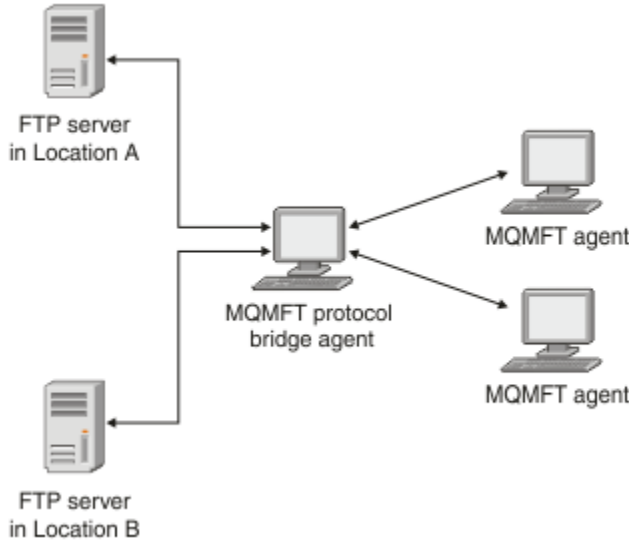
Birden çok uç noktaya eşzamanlı olarak dosya aktarmak için bir iletişim kuralı köprüsü aracısını kullanabilirsiniz. MFT , dosyaları aktarmak istediğiniz farklı protokol dosyası sunucularını

tanımlamak için düzenleyebileceğiniz `ProtocolBridgeProperties.xml` adlı bir dosya sağlar. **fteCreateBridgeAgent** komutu, varsayılan protokol dosyası sunucusunun ayrıntılarını sizin için `ProtocolBridgeProperties.xml` 'e ekler. Bu dosya, Protokol köprüsü özellikleri dosya biçimiinde açıklanmıştır.

Protokol köprüsü aracısını kullanarak aşağıdaki işlemleri gerçekleştirebilirsiniz:

- FTP, FTPS ya da SFTP kullanarak MFT ağından uzak sunucuya dosya yükleyin.
- Download files from a remote server, using FTP, FTPS, or SFTP, to the MFT network

Not: Protokol köprüsü aracısı, dosyalara mutlak dosya yolu tarafından erişilmesine izin veren yalnızca FTP, FTPS ya da SFTP sunucularını destekleyebilir. Aktarma isteğinde görel bir dosya yolu belirtilirse, protokol köprüsü aracısı, iletişim kuralı sunucusunda oturum açmak için kullanılan ana dizine dayalı olarak görel yolu mutlak dosya yoluna dönüştürmeyi dener. Yalnızca geçerli dizine dayalı dosyalara erişime izin veren bu iletişim kuralı sunucuları, protokol köprüsü aracısı tarafından desteklenmiyor.



Çizgede, farklı konumlarda iki FTP sunucusu gösterilir. FTP sunucuları, dosyaları Managed File Transfer araçları ile değiş tokuş etmek için kullanılır. İletişim kuralı köprüsü aracısı, FTP sunucuları ile MFT ağının geri kalanı arasında ve her iki FTP sunucusuyla iletişim kuracak şekilde yapılandırılmış olmalıdır.

İletişim kuralı köprüsü aracısına ek olarak MFT ağınızda başka bir aracıya sahip olduğundan emin olun. Protokol köprüsü aracısı yalnızca FTP, FTPS ya da SFTP sunucusuna yapılan bir köprüdür ve aktarılan dosyaları yerel diske yazmaz. FTP, FTPS ya da SFTP sunucusuna dosya aktarmak istiyorsanız, dosya aktarımı için hedef ya da kaynak olarak protokol köprüsü aracısını (FTP, FTPS ya da SFTP sunucusunu gösteren) ve başka bir standart aracıyı karşılık gelen kaynak ya da hedef olarak kullanmanız gerekir.

Protokol köprüsünü kullanarak dosya aktardığınızda, köprünün, aktarmak istediğiniz dosyaları içeren kaynak dizini ya da kaynak dizini okuma izni olmalıdır. Örneğin, yalnızca yürütme izinleri (d -- x -- x -- x) olan `/home/fte/bridge` dizininden dosya aktarmak istiyorsanız, bu dizinden denediğiniz aktarma işlemleri 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

Bir iletişim kuralı köprüsü aracısı, standart bir MFT aracısı gibidir. **fteCreateBridgeAgent** komutunu kullanarak bir iletişim kuralı köprüsü aracısı oluşturun. Bir iletişim kuralı köprüsü aracısını, Protokol köprüsü özellikleri dosya biçimiinde açıklanan `ProtocolBridgeProperties.xml` dosyasını kullanarak yapılandırabilirsiniz. Daha önceki bir sürümü kullanıyorsanız, aracıyı, `Advanced Agent properties: Protocol bridge` ve `Advanced Agent properties: Protocol bridge Agent logging` içinde açıklanan belirli protokol köprüsü özelliklerini kullanarak yapılandırın. For all versions, you can also configure a

credential mapping as described in [“Bir dosya sunucusuna ilişkin kimlik bilgilerinin eşlenmesi” sayfa 264](#). Belirli bir protokol dosyası sunucusu için bir protokol köprüsü aracısını yapılandırdıktan sonra, o aracıyı yalnızca bu amaçla kullanabilirsiniz.

Protokol köprüsü kurtarması

Protokol köprüsü aracısı, dosya sunucusu kullanılmadığı için dosya sunucusuna bağlanamazsa, dosya sunucusu kullanılabilir duruma gelinceye kadar tüm dosya aktarma istekleri kuyruğa alınır. Aracı yanlış kimlik bilgilerini kullandığı için protokol köprüsü aracısı dosya sunucusuna bağlanamazsa, aktarma başarısız olur ve aktarma günlüğü iletisi 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 yapılandırılmış bir iletişim kuralı dosya sunucusuysa (kullanıcılar, iletişim kuralı dosya sunucusuna dosya yükleyebilir, ancak karşıya yüklenen dosyaları herhangi bir şekilde değiştiremezler; yalnızca bir kez yazı yazabilir), aktarılan dosyalar doğrudan hedefe yazılır. Başka bir deyişle, aktarma sırasında bir sorun oluşursa, kısmen yazılmış dosyalar hedef protokol dosyası sunucusunda kalır ve Managed File Transfer bu dosyaları silemez ya da düzenleyemez. Bu durumda, aktarma başarısız olur.

Defining properties for protocol file servers using the ProtocolBridgeProperties.xml file

Aracı yapılandırma dizininde Managed File Transfer tarafından sağlanan ProtocolBridgeProperties.xml dosyasını kullanmak ve kullanmak istediğiniz bir ya da daha fazla protokol dosyası sunucusunun özelliklerini tanımlayın. Bu,filesdosyasını kullanarak dosyaları aktarmak için.

Bu görev hakkında

fteCreateBridgeAgent komutu, *MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name* aracı yapılandırma dizininde ProtocolBridgeProperties.xml dosyasını oluşturur. Komut, komut çalıştırıldığında varsayılan değer belirlenmişse, varsayılan protokol dosya sunucusu için dosyada bir giriş de yazar.

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 use
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://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.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="99999999"
      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 ek kimlik bilgileri dosyası işleme için el ile oluşturulmalıdır. Varsayılan olarak bu dosya ProtocolBridgeCredentials.xml adını alır ve ana sayfa içinde bulunur. Aracıyı başlatan kullanıcının dizini. Örneğin, bu kullanıcı aracıyı başlattıysa Yer: \$HOME/ProtocolBridgeCredentials.xml

Kimlik bilgileri dosyası kullanıyorsanız:

1. Bir tane yaratılmasına ilişkin ek bilgi için aşağıdaki metne bakın.
2. Kimlik bilgileri dosyası, sınırlı izinlere sahip bir dizinde olmalıdır. Örneğin, diğer kullanıcılar için okuma erişimi olmamalıdır.
3. Başlatılan aracıya ilişkin kullanıcı kimliği için \$HOME ortam değişkeninde kimlik bilgileri dosyasına ilişkin dizinin konumunu belirtin ya da ProtocolBridgeProperties.xml dosyasını düzenleyin ve aşağıdaki konumu belirtin:

```
<tns:credentialsFile path="/test/ProtocolBridgeCredentials.xml"/>
```

Varsayılan olmayan protokol sunucuları eklemek istiyorsanız, bu dosyaların özelliklerini tanımlamak için bu dosyayı düzenleyin. Bu örnek, ek bir FTP sunucusu ekler.

Not: Protokol köprüsü aracı dosya kilitlemesini desteklemiyor. Bunun nedeni, Managed File Transfer' un bir dosya sunucusundaki dosya kilitleme mekanizmasını desteklememesinden kaynaklanır.

Yordam

1. Define a protocol file server by inserting the following lines into the file as a child element of <tns:serverProperties>:

```
<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 , protokol dosyası sunucunuzun adıdır
- host , protokol dosyası sunucusunun anasistem adı ya da IP adresidir.
- port , protokol dosyası sunucusunun kapı numarasıdır
- platform , protokol dosyası sunucusunun çalıştığı altyapıdır
- timeZone , protokol dosyası sunucusunun çalıştığı saat dilimidir.
- locale , iletişim kuralı dosya sunucusunda kullanılan dildir

- `fileEncoding` , protokol dosyası sunucusunun karakter kodlamasıdır
- `listFormat` , protokol dosyası sunucusundan döndürülen dosya listeleme biçimidir.
- `limitedWrite` , bir dosya sunucusuna yazılırken varsayılan kipin izlenip izlenmeyeceğini belirler; bu, geçici bir dosya oluşturmak ve aktarma tamamlandığında o dosyayı yeniden adlandırmak olur. Yalnızca yazma olarak yapılandırılmış bir dosya sunucusu için, dosya doğrudan son adıyla yaratılır. Bu özelliğin değeri `true` (doğru) ya da `false` (yanlış) olabilir. İletişim kuralı köprüsü araçlarında `limitedWrite` özneliği ve `doNotUseTempOutputFile` aracı özneliği birlikte kullanılır. Geçici dosyaları kullanmak istiyorsanız, `doNotUseTempOutputFile` değerini ayarlamamanız ve `limitedWrite` değerini `false` olarak ayarlamamanız gerekir. Diğer herhangi bir ayar bileşimi, geçici dosyaların kullanılmayacağı anlamına gelir.
- `maxListFileNames` , dosya adları için iletişim kuralı dosya sunucusunda bir dizin taranırken toplanan ad sayısı üst sınırdır.
- `maxListDirectoryLevels` , dosya adları için iletişim kuralı dosya sunucusunda bir dizin taranırken, yinelenen dizin düzeyi üst sınısıdır.

Bu özneliklerle ilgili daha fazla ayrıntı için, bunlar zorunlu ya da isteğe bağlı ve varsayılan değerleri de dahil olmak üzere, [Protokol köprüsü özellikleri dosya biçimi](#) konusuna bakın.

İlgili başvurular

[Protokol köprüsü özellikleri dosya biçimi](#)

[MFT tarafından kullanılan düzenli ifadeler](#)

İletişim kuralı dosya sunucusu özellikleri arıyor: ProtocolBridgePropertiesExit2

Çok sayıda protokol dosyası sunucunuz varsa, aktarlarda gönderme yapılan protokol dosyası sunucusu özelliklerini aramak için `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` arabirimini gerçekleştirebilirsiniz. Bir `ProtocolBridgeProperties.xml` dosyasını korumak için tercihte bu arabirimi uygulayabilirsiniz.

Bu görev hakkında

Managed File Transfer , iletişim kuralı dosya sunucusu özelliklerini arayan bir kullanıcı çıkışı sağlar. Daha fazla bilgi için bkz. "[Protokol dosyası sunucusu özelliklerini aramak için örnek kullanıcı çıkışının kullanılması](#)" sayfa 261.

İletişim kuralı köprüsü özelliklerinin görüneceği herhangi bir kullanıcı çıkışı, `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` arabirimini gerçekleştirmelidir. Daha fazla bilgi için bkz. [ProtocolBridgePropertiesExit.java](#) arabirimi.

Birden çok iletişim kuralı sunucusu özelliklerini birbirine zincirleyebilir, diğer kullanıcı çıkışlarına benzer bir şekilde birlikte çıkar. Çıkışlar, aracı özellikleri dosyasında `protocolBridgePropertiesExitClasses` özelliğini kullanarak belirttikleri sırayla çağrılır. Başlatma yöntemlerinin tümü ayrı olarak döndürülür; bir ya da daha çok `false` değeri döndürülürse, aracı başlatılmaz. Hata, aracı olay günlüğüne raporlanır.

Tüm çıkışlar için `getProtocolServerProperties` yöntemleri için yalnızca bir genel sonuç döndürülür. Yöntem sonuç kodu olarak bir özellik nesnesi döndürürse, bu değer döndürülen sonuçtur ve sonraki çıkışların `getProtocolServerProperties` yöntemlerinin çağrılmaz. Yöntem, sonuç kodu olarak boş değer döndürürse, sonraki çıkışa ilişkin `getProtocolServerProperties` yöntemi çağrılır. Sonraki çıkış yoksa, boş sonuç döndürülür. Boş değerli genel sonuç kodu, protokol köprüsü aracısının arama hatası olarak kabul edilir.

`ProtocolBridgePropertiesExit2.java` arabirimini kullanmanız önerilir; ancak, `ProtocolBridgePropertiesExit.java` arabirimiyle ilgili bilgi için bkz. "[İletişim kuralı dosya sunucusu özellikleri arıyor: ProtocolBridgePropertiesExit](#)" sayfa 262.

Çıkışınızı çalıştırmak için aşağıdaki adımları tamamlayın:

Yordam

1. Protokol sunucusu özellikleri kullanıcı çıkışı derleyin.
2. Derlenmiş çıkışı ve paket yapısını içeren bir Java arşiv (JAR) dosyası oluşturun.
3. İletişim kuralı köprüsü aracısının `exits` dizinine çıkış sınıfını içeren JAR dosyasını yerleştirin. Bu dizin, `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` dizininde bulunur.
4. Edit the property file of the protocol bridge agent to include the property `protocolBridgePropertiesExitClasses`. 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` tarafından belirtilen çıkış sınıflarının ilk kullanıma hazırlama () yöntemine String olarak geçirilir. Daha fazla bilgi için bkz. `MFT agent.properties` dosyası.

Protokol dosyası sunucusu özelliklerini aramak için örnek kullanıcı çıkışının kullanılması

Managed File Transfer , iletişim kuralı dosya sunucusu özelliklerini aralayan bir kullanıcı çıkışı sağlar.

Bu görev hakkında

İletişim kuralı köprüsü özelliklerini gösteren örnek bir kullanıcı çıkışı, `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` dizininde ve [Örnek protokol köprüsü özellikleri kullanıcı çıkışı](#) başlıklı konuda sağlanmıştır.

`SamplePropertiesExit2.java` çıkışı, protokol sunucularına ilişkin özellikleri içeren bir özellikler dosyasını okur. Özellikler dosyasındaki her girişin biçimi aşağıdaki gibidir:

```
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. Derlenmiş 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. Edit the `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` file to contain the line:

```
protocolBridgePropertiesExitClasses=SamplePropertiesExit2
```

5. Create a protocol bridge properties file, for example `protocol_bridge_properties.properties`, in the directory `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent`. Bu dosyayı, girdileri şu biçimde içermek için düzenleyin:

```
serverName=type://host:port
```

6. Edit the `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties` file to contain the line:

```
protocolBridgePropertiesConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/protocol_bridge_properties.properties
```

protocol_bridge_properties.properties dosyasının mutlak yolunu kullanmanız gerekir.

7. Start the protocol bridge agent by using the **fteStartAgent** command.

İlgili kavramlar

[“Protokol köprüsü” sayfa 255](#)

İletişim kuralı köprüsü, Managed File Transfer (MFT) ağınızın, yerel etki alanınızda ya da uzak bir konumdaki MFT ağınızın dışında bir dosya sunucusunda saklanan dosyalara erişmesini sağlar. Bu dosya sunucusu, FTP, HTTPS ya da SFTP ağ protokollerini kullanabilir. Her dosya sunucusunun en az bir adanmış aracısına gerek vardır. Adanmış aracı, protokol köprüsü aracısı olarak bilinir. Bir köprü aracısı, birden çok dosya sunucusuyla etkileşimde bulunabilir.

İlgili başvurular

[ProtocolBridgePropertiesExit.java arabirimi](#)

[Örnek protokol köprüsü özellikleri kullanıcı çıkışı](#)

[The MFT agent.properties file](#)

[fteCreateBridgeAgent \(bir MFT iletişim kuralı köprüsü aracısı oluşturun ve yapılandırın\)](#)

İletişim kuralı dosya sunucusu özellikleri aranıyor: ProtocolBridgePropertiesExit

Çok sayıda protokol dosyası sunucunuz varsa, aktarlarda gönderme yapılan protokol dosyası sunucusu özelliklerini aramak için `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` arabirimini gerçekleştirilebilirsiniz.

Bu görev hakkında

Bir `ProtocolBridgeProperties.xml` dosyasını korumak için tercihte `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` arabirimini uygulayabilirsiniz. `ProtocolBridgePropertiesExit2.java` arabirimini kullanmanız önerilir, ancak `ProtocolBridgePropertiesExit.java` arabirimi de desteklenmektedir. If you have an existing implementation of the `ProtocolBridgePropertiesExit.java` interface from IBM WebSphere MQ File Transfer Edition, you can use it in IBM WebSphere MQ 7.5 or later. `ProtocolBridgePropertiesExit2.java` içindeki `getCredentialYer` yöntemi, ana dizininiz olan `ProtocolBridgeCredentials.xml` dosyasının varsayılan konumunu kullanır.

Not: IBM WebSphere MQ File Transfer Edition (FTE) artık desteklenen bir ürün değildir. IBM MQ'da FTE'den Managed File Transfer bileşenine geçiş için [Migrating Managed File Transfer](#) başlıklı konuya bakın.

Any user exit that looks up protocol bridge properties must implement the interface `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit`:

```
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 özelliklerini birbirine zincirleyebilir, diğer kullanıcı çıkışlarına benzer bir şekilde birlikte çıkar. Çıkışlar, aracı özellikleri dosyasında `protocolBridgePropertiesExitClasses` özelliğini kullanarak belirttikleri sırayla çağrılır. Başlatma yöntemlerinin tümü ayrı olarak döndürülür; bir ya da daha çok false değeri döndürülürse, aracı başlatılmaz. Hata, aracı olay günlüğüne raporlanır.

Tüm çıkışlar için `getProtocolServerProperties` yöntemleri için yalnızca bir genel sonuç döndürülür. Yöntem sonuç kodu olarak bir özellikler nesnesi döndürürse, bu değer döndürülen sonuçtur ve sonraki çıkışların `getProtocolServerProperties` yöntemlerinin çağrılmaz. Yöntem, sonuç kodu olarak boş değer döndürürse, sonraki çıkışa ilişkin `getProtocolServerProperties` yöntemi çağrılır. Sonraki çıkış yoksa, boş sonuç döndürülür. Boş değerli genel sonuç kodu, protokol köprüsü aracısının 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 oluşturun.
3. İletişim kuralı köprüsü aracısının `exits` dizinine çıkış sınıfını içeren JAR dosyasını yerleştirin.

Bu dizin, `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` dizininde bulunur.

4. Edit the property file of the protocol bridge agent to include the property `protocolBridgePropertiesExitClasses`.

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` tarafından belirtilen çıkış sınıflarının ilk kullanıma hazırlama () yöntemine String olarak geçirilir. Daha fazla bilgi için bkz. `MFT agent.properties` dosyası.

Bir dosya sunucusuna ilişkin kimlik bilgilerinin eşlenmesi

Map user credentials in Managed File Transfer to user credentials on the file server by using the default credential mapping function of the protocol bridge agent or by writing your own user exit. Managed File Transfer , kullanıcı kimlik bilgileri eşlemesini gerçekleştiren bir örnek kullanıcı çıkışı sağlar.

***ProtocolBridgeCredentials.xml* dosyasını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerinin eşlenmesi**

Map user credentials in Managed File Transfer to user credentials on the file server by using the default credential mapping function of the protocol bridge agent. Managed File Transfer , kimlik bilgileri bilgilerinizi içermek için düzenleyebileceğiniz bir XML dosyası sağlar.

Bu görev hakkında

`ProtocolBridgeCredentials.xml` dosyası, kullanıcı tarafından el ile oluşturulmalıdır. Varsayılan olarak, bu dosyanın yeri protokol köprüsü aracısını başlatan kullanıcının ana dizinidir; ancak bu, aracı tarafından erişilebilen dosya sisteminde herhangi bir yerde saklanabilir. Farklı bir yer belirtmek için, `<credentialsFile>` ögesini `ProtocolBridgeProperties.xml` dosyasına ekleyin. Örneğin,

```
<tns:credentialsFile path="/example/path/to/ProtocolBridgeCredentials.xml"/>
```

Bir iletişim kuralı köprüsü aracısını kullanmadan önce, bu dosyayı anasistem, kullanıcı ve kimlik bilgileri bilgilerinizi içerecek şekilde düzenleyerek kimlik bilgileri eşlemesini ayarlayın. Daha fazla bilgi ve örnek için bkz. [Protokol köprüsü kimlik bilgileri dosya biçimi](#).

z/OS When you create the `ProtocolBridgeCredentials.xml` file on a z/OS platform using IBM WebSphere MQ 7.5 or earlier, you must set a file tag before you edit the file. Dosyayı ASCII içeriği olarak işaretlemek için aşağıdaki komutu çalıştırın:

```
chtag -t -c IS08859-1 ProtocolBridgeCredentials.xml
```

Not: **z/OS** z/OS' ta, iletişim kuralı köprüsü kimlik bilgileri dosyasını bir veri kümesinde saklayabilirsiniz; burada `.xml` dosyasının adı kullanıcı tarafından belirtilebilir.

Yordam

1. Edit the line `<tns:server name="server name">` to change the value of the name attribute to the server name in the `ProtocolBridgeProperties.xml` file.

IBM WebSphere MQ File Transfer Edition 7.0.4 ve öncesi için oluşturulan protokol köprüsü araçlarının bir `ProtocolBridgeProperties.xml` dosyası (ya da ilgili kullanıcı çıkışları) yoktur, bu nedenle IBM WebSphere MQ File Transfer Edition 7.0.4 Fix Pack 1 için ve daha sonra sunucu adı sunucunun ana makine adını otomatik olarak atamaktadır. Bu nedenle, güncellenmiş bir `ProtocolBridgeCredentials.xml` dosyasını `<sunucu>` girişleriyle kullanıyorsanız, sunucunun ana makine adına karşılık gelen bir ad eşleşir.

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. Insert user ID and credential information into the file as child elements of `<tns:server>`.
Aşağıdaki öğelerden birini ya da birçoğunu dosyaya ekleyebilirsiniz:

- Protokol dosyası sunucusu bir FTP, FTPS ya da SFTP sunucusuysa, aktarımı isteyen kullanıcının kimliğini doğrulamak için parolaları kullanabilirsiniz. Aşağıdaki satırları dosyaya ekleyin:

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  serverPassword="Server Password">
</tns:user>
```

Sonra, özniteliklerin değerini değiştirin.

- name , MFT aktarma isteğiyle ilişkilendirilmiş MQMD kullanıcı kimliğiyle eşleştirmek için kullanılan bir Java düzenli ifadesidir.
- serverUserId , oturum açma kullanıcı kimliği olarak protokol dosyası sunucusuna geçirilen değerdir. serverUserId özniteliği belirtilmezse, yerine MFT aktarma isteğiyle ilişkilendirilmiş MQMD kullanıcı kimliği kullanılır.
- serverPassword , serverUserId ile ilişkili paroladır.

name özniteliği bir Java düzenli ifadesi içerebilir. The credential mapper attempts to match the MQMD user ID of the MFT transfer request to this regular expression. Protokol köprüsü aracısı, MQMD kullanıcı kimliğiyle, <tns: user> öğelerinin ad (name) öznitelideki düzenli ifadeye eşleştirmeyi dener (öğelerin dosyada var olması gerekir). Eşleşme bulunduğunda, protokol köprüsü aracısı daha fazla eşleşme aramaz. Bir eşleşme bulunursa, ilgili serverUserId ve serverPassword değerleri, oturum açma kullanıcı kimliği ve parolası olarak protokol dosyası sunucusuna geçirilir. MQMD kullanıcı kimliği eşleşmeleri büyük ve küçük harfe duyarlıdır.

- Protokol dosyası sunucusu bir SFTP sunucusuysa, aktarımı isteyen kullanıcının kimliğini doğrulamak için genel ve özel anahtarları kullanabilirsiniz. Aşağıdaki satırları dosyaya ekleyin ve özniteliklerin değerini değiştirin. <tns: user> ögesi bir ya da daha çok <tns:privateKey> öge 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 eşleştirmek için kullanılan bir Java düzenli ifadesidir.
- serverUserId , oturum açma kullanıcı kimliği olarak protokol dosyası sunucusuna geçirilen değerdir. serverUserId özniteliği belirtilmezse, yerine MFT aktarma isteğiyle ilişkilendirilmiş MQMD kullanıcı kimliği kullanılır.
- hostKey , oturum açılırken sunucunun döndürdüğü beklenen anahtardır
- key , serverUserId' nin özel anahtardır.
- keyPassword , ortak anahtarları oluşturmak için anahtarın parolasıdır
- associationName , izleme ve günlüğe kaydetme amaçlarını tanımlamak için kullanılan bir değerdir.

name özniteliği bir Java düzenli ifadesi içerebilir. The credential mapper attempts to match the MQMD user ID of the MFT transfer request to this regular expression. Protokol köprüsü aracısı, MQMD kullanıcı kimliğiyle, <tns: user> öğelerinin ad (name) öznitelideki düzenli ifadeye eşleştirmeyi dener (öğelerin dosyada var olması gerekir). Eşleşme bulunduğunda, protokol köprüsü aracısı daha fazla eşleşme aramaz. If a match is found, the corresponding serverUserId and key values are used to authenticate the MFT user with the protocol file server. MQMD kullanıcı kimliği eşleşmeleri büyük ve küçük harfe duyarlıdır.

Bir iletişim kuralı köprüsü aracısıyla özel anahtarlar kullanma hakkında daha fazla bilgi için bkz. [“Örnek: Bir iletişim kuralı köprüsü aracısının, UNIX SFTP sunucusu ile özel anahtar kimlik bilgilerini kullanacak şekilde yapılandırılması” sayfa 268.](#)

Not: 

Aktarma isteđi komut kuyruđuna yazıldıđında, kaynak aracı komut kuyruđu bir z/OS ya da IBM i sisteminde olması durumunda, MQMD kullanıcı kimliđi büyük harfe dönüştürülebilir. Sonuç olarak, aynı kaynak kullanıcı için MQMD kullanıcı kimliđi, özgün vakadaki kimlik bilgileri çıkışa gelebilir ya da aktarma isteđinde belirtilen kaynak aracıya bađlı olarak büyük harfe dönüştürülebilirdi. Varsayılan kimlik bilgileri eşleme çıkışı, sađlanan MQMD kullanıcı kimliđine karşı büyük ve küçük harfe duyarlı eşleşmeler gerçekleştirir; bu, eşleme dosyasında izin vermenize gerek duyabilirsiniz.

Çıkış sınıflarını kullanarak bir dosya sunucusuna ilişkin kimlik bilgilerinin eşlenmesi

If you do not want to use the default credential mapping function of the protocol bridge agent, you can map user credentials in Managed File Transfer to user credentials on the file server by writing your own user exit. Kimlik bilgileri eşleme kullanıcı çıkışlarını yapılandırırsanız, bunlar varsayılan kimlik bilgileri eşleme işlevinin yerini alır.

Bu görev hakkında

Managed File Transfer , kullanıcı kimlik bilgileri eşlemesini gerçekleştiren bir örnek kullanıcı çıkışı sađlar. Daha fazla bilgi için bkz [“Örnek protokol köprüsü kimlik bilgileri kullanıcı çıkışının kullanılması” sayfa 267.](#)

Eşleme iletişim kuralı köprüsü kimlik bilgileri için bir kullanıcı çıkışı aşıđıdaki arabirimlerden birini gerçekleştirmelidir:

- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`, which allows a protocol bridge agent to transfer files to and from one default protocol file server
- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2`, which allows you to transfer files to and from multiple endpoints.

`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, diđer kullanıcı çıkışlarına benzer bir şekilde zincirleme olarak birlikte bađlanabilir. Çıkışlar, aracı özellikleri dosyasında `protocolBridgeCredentialConfiguration` özelliđini kullanarak belirtildikleri sırayla çağrılır. Başlatma yöntemlerinin tümü ayrı olarak döndürülür; bir ya da daha çok false deđeri döndürülürse, aracı başlatılmaz. Hata, aracı olay günlüđüne raporlanır.

Tüm çıkışların `mapMQUserTanıtıcı` yöntemleri için aşıđıdaki gibi yalnızca bir genel sonuç döndürülür:

- Yöntem, sonuç kodu olarak `USER_SUCFULLY_MAPPED` ya da `USER_DENIED_Access` deđerini döndürürse, bu deđer döndürülen sonuç olur ve sonraki çıkışlara ilişkin `mapMQUserTanıtıcı` yöntemleri çağrılmaz.
- Yöntem sonuç kodu olarak `NO_MAPPING_FOUND` deđerini döndürürse, sonraki çıkışa ilişkin `mapMQUserTanıtıcı` yöntemi çağrılır.
- Sonraki bir çıkış yoksa, `NO_MAPPING_FOUND` sonucu döndürülür.
- `USER_DENIE_ACCESS` ya da `NO_MAPPING_FOUND` ' in genel sonuç kodu, köprü aracı tarafından bir aktarma hatası olarak kabul edilir.

Çıkışınızı çalıştırmak için aşıđıdaki adımları tamamlayın:

Yordam

1. Protokol köprüsü kimlik bilgileri kullanıcı çıkışını derleyin.
2. Derlenmiş çıkışı ve paket yapısını içeren bir Java arşiv (JAR) dosyası oluşturun.
3. Çıkış sınıfını içeren JAR dosyasını, köprü aracısının `exits` dizinine yerleştirin. Dizin, `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` dizininde yer alıyor.

4. Edit the property file of the protocol bridge agent to include the property `protocolBridgeCredentialExitClasses`. Bu özelliğin değeri için, protokol köprüsü kimlik bilgisi çıkış yordamını uygulayan sınıfların virgülle ayrılmış bir listesini belirtin. Çıkış sınıfları, bu listede belirtildikleri sırayla çağrılır. Daha fazla bilgi için bkz. `MFT agent.properties` dosyası.
5. Protokol köprüsü aracısının özellik dosyasını aşağıdaki içermeye uygun olarak düzenleyin:

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

Bir aracı için `agent.properties` dosyası, `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/bridge_agent_name` dizininizde olur.

`agent.properties` dosyasını değiştirirseniz, değişiklikleri almak için aracıyı yeniden başlatmanız gerekir.

6. İsteğe bağlı olarak `protocolBridgeCredentialConfiguration` özelliğini belirtebilirsiniz. Bu özellik için belirttiğiniz değer, `protocolBridgeCredentialExitClasses` tarafından belirtilen çıkış sınıflarının `initialize()` yöntemine `String` nesnesi olarak geçirilir. Daha fazla bilgi için bkz. `MFT agent.properties` dosyası.
7. Protokol köprüsü aracısını **`fteStartAgent`** komutuyla başlatın.

Örnek protokol köprüsü kimlik bilgileri kullanıcı çıkışının kullanılması

Managed File Transfer , kullanıcı kimlik bilgileri eşlemesini gerçekleştiren bir örnek kullanıcı çıkışı sağlar.

Bu görev hakkında

A sample protocol bridge credential exit is provided in the `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` directory and in the topic [Örnek protokol köprüsü kimlik bilgileri kullanıcı çıkışı](#). Bu örnek, `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` arabirimini temel alır.

`SampleCredentialExit.java` çıkışı, sunucu kullanıcı kimliklerine ve sunucu parolarına yönelik aktarma istekleriyle ilişkili MQMD kullanıcı kimlikleriyle eşleşen 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. Edit the `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` file to contain the line:

```
protocolBridgeCredentialExitClasses=SampleCredentialExit
```

5. Protokol köprüsü aracısının özellik dosyasını aşağıdaki içermeye uygun olarak düzenleyin:

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

Bir aracı için `agent.properties` dosyası, `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` dizininizde olur.

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 bilgisi özellikler dosyası (`credentials.properties`) oluşturun ve bunları, aşağıdaki biçime eklemek için düzenleyin:

```
mqUserId=serverUserId,serverPassword
```

7. Edit the `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` file to contain the line:

```
protocolBridgeCredentialConfiguration=MQ_DATA_PATH/mqft/  
config/coordination_queue_manager/agents/bridge_agent_name/credentials.properties
```

`credentials.properties` dosyasının mutlak yolunu kullanmanız gerekir.

8. Start the protocol bridge agent by using the **fteStartAgent** command.

Örnek: Bir iletişim kuralı köprüsü aracısının, UNIX SFTP sunucusu ile özel anahtar kimlik bilgilerini kullanacak şekilde yapılandırılması

Bu örnek, `ProtocolBridgeCredentials.xml` dosyasını nasıl oluşturabileceğiniz ve yapılandırabileceğiniz gösterilir. Bu örnek, tipik bir örnektir ve ayrıntılar altyapınıza göre değişiklik gösterebilir, ancak ilkeler aynı kalır.

Bu görev hakkında

Yordam

1. SFTP sunucusu ile kimlik doğrulaması yapmak için kullanılacak genel ve özel bir anahtar oluşturun. Örneğin, bir Linux anasisteminde, genel/özel anahtar çiftini yaratmak için 'openssh' paketinin bir parçası olarak sağlanan aracı **ssh-keygen**' yı kullanabilirsiniz. Varsayılan olarak, bağımsız değişken olmadan, **ssh-keygen** komutu, iki anahtar dosyası için bir konum ve geçiş tümcesine ilişkin bilgi istemlerini kullanır; bu, varsayılan olarak adlara varsayılan değer olarak gelir:

```
id_rsa      <-- Private key  
id_rsa.pub  <-- Public key
```



Uyarı: If you are using the **ssh-keygen** command from a recent version of OpenSSH, such as that supplied with RHEL 8, the key format used is not compatible with the protocol bridge agent, and transfer attempts to the SFTP server fail with the message:

```
BFGBR0216E: Authentication to protocol server 'sftp.host.address' failed  
because of invalid private key.
```

OpenSSH' un bu yeni sürümleriyle uyumlu bir özel anahtar yaratmak için, **ssh-keygen** komutuna aşağıdaki bağımsız değişkenle birlikte anahtar biçimini belirtin:

```
ssh-keygen -m PEM
```

`id_rsa` özel anahtarının içeriği, aşağıdaki ilk ve son satırlara sahip olur:

```
-----BEGIN RSA PRIVATE KEY-----  
.....  
-----END RSA PRIVATE KEY-----
```

bu da protokol köprüsü aracısıyla uyumludur.

- Copy the entire contents of the `id_rsa.pub` file into the `~/.ssh/authorized_keys` file of the SFTP user on the SFTP server.

Bu dosyayla ilgili dosya izinlerinin ve `~/.ssh` dizininin, anahtar doğrulamaya izin vermek üzere SFTP sunucusu için uygun şekilde ayarlandığından emin olun. Bu izinler genellikle:

```
~/.ssh Mode 700
~/.ssh/authorized_keys Mode 600
```

- Managed File Transfer , MD5 algoritması kullanılarak oluşturulan bir anasistem ssh parmak izi gerektirir. SFTP sunucusunun anasistem ssh parmak izini almak için aşağıdaki komutlardan birini çalıştırın.

- Red Hat® Enterprise Linux sürüm 6.x ve aşağısında 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 çıktısı aşağıdaki örneğe benzer olacaktır:

```
2048 MD5:64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27 no comment (RSA)
```

Extract the hexadecimal portion only of the output to use as the `hostKey` in the `ProtocolBridgeCredentials.xml` file (see step “4” sayfa 269). Bu nedenle, bu örnekte `64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27` u ayıklayacaksınız.

- İletişim kuralı köprüsü aracı sisteminde, `ProtocolBridgeCredentials.xml` dosyasını düzenleyin. Aşağıdaki örnekte italik olarak gösterilen değerleri kendi değerlerinizle değiştirin:

```
<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:

- Agent_name* , protokol köprüsü aracısının adıdır.
- SFTP_host_name* , `ProtocolBridgeProperties.xml` dosyasında gösterildiği gibi SFTP sunucusunun adıdır.
- mq_User_ID* , aktarma isteğiyle ilişkili MQMD kullanıcı kimliğidir.
- SFTP_user_ID* , 2. adımda kullanıldığı şekliyle SFTP kullanıcı kimliğidir. SFTP ' ye iletilen değer, oturum açma kullanıcı kimliği olarak işlev görmektedir.
- ssh_host_finger* , 3. adımda toplanan parmak izidir.

- *ad* izleme ve günlüğe kaydetme amacıyla kullanılacak belirtebileceğiniz bir addir.
- *pas_tümcecik* , 1. adımda ssh-keygen içinde sağladığınız geçiş tümcecidir.
- *ID_rsa dosyasının tam içeriği* , oluşturulan *id_rsa* kütüğünün 1. adımdan tam olarak içeriğidir. Bir bağlantı hatasını önlemek için aşağıdaki girişlerin her ikisini de içerdiğinizden emin olun:

```
-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----
```

< tns: privatekey> ögesini yineleyerek ek tuşlar ekleyebilirsiniz.

5. Aracı önceden başlatılmamışsa, protokol köprüsü aracısını başlatın. Diğer bir seçenek olarak, protokol köprüsü aracısı düzenli olarak ProtocolBridgeCredentials.xml dosyasını yoklar ve değişiklikleri alır.

FTPS sunucusu için bir protokol köprüsünü yapılandırma

Bir FTP sunucusunu yapılandığınız 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 iletişim kuralı 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) Güvenilirlik deposu dosyasına ilişkin **-bts** parametresi. Komut, yalnızca sunucu kimlik doğrulamasının gerekli olduğunu varsayar ve güvenilir depo dosyasının yerini belirtmeniz gerekir.

FTPS protokolünün açık biçimi, varsayılan olarak **fteCreateBridgeAgent** komutu tarafından yapılandırılır, ancak protokol köprüsü özellikleri dosyasını değiştirerek örtük formu yapılandırabilirsiniz. İletişim kuralı köprüsü, her zaman pasif modda FTPS sunucularına bağlanır.

fteCreateBridgeAgent komutuyla ilgili daha fazla bilgi için bkz. [fteCreateBridgeAgent \(MFT iletişim kuralı köprüsü aracısını oluşturma ve yapılandırma\)](#).

If you need instructions about how to create truststore files, see the IBM Developer article, [IBM WebSphere MQ File Transfer Edition' ta Secure Sockets Layer bağlantırlığını yapılandırma](#), or see the information about the keytool at the [Oracle anahtar aracı belgeleri](#).

Not: IBM WebSphere MQ File Transfer Edition (FTE) artık desteklenen bir ürün değildir. IBM MQ'da FTE' den Managed File Transfer bileşenine geçiş için [Migrating Managed File Transfer](#) başlıklı konuya bakın.

2. Protokol köprüsü özellikleri dosyasında bir <ftpsServer> ögesi içinde FTPS sunucusu özelliklerini tanımlayın: ProtocolBridgeProperties.xml. Daha fazla bilgi için, bkz. [“Defining properties for protocol file servers using the ProtocolBridgeProperties.xml file” sayfa 257](#). Ayrıca, protokol köprüsü özellikleri dosyasını düzenleyerek istemci kimlik denetimini etkinleştirebilirsiniz. Tüm yapılandırma seçeneklerine ilişkin ayrıntılar için bkz. [Protocol bridge properties file format](#).
3. Map user credentials in Managed File Transfer to user credentials on the FTPS server either by using the default credential mapping function of the protocol bridge agent or by writing your own user exit. Daha fazla bilgi için bkz [“Bir dosya sunucusuna ilişkin kimlik bilgilerinin eşlenmesi” sayfa 264](#).
4. Varsayılan olarak, güvenilir depo dosyası JKS biçimine sahip olarak yapılandırılır; biçimi değiştirmek isterseniz, protokol köprüsü özellikler dosyasını düzenleyin.

Örnek

Protokol köprüsü özellikleri dosyasında bir FTPS sunucusu için örnek bir giriş aşağıdaki gibi gösterilir:

```
<tns:serverProperties xmlns:tns="http://wqfite.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wqfite.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">
  <tns:defaultServer name="ftpsserver.mycompany.com" />

  <tns:ftpsServer name="ftpsserver.mycompany.com" host="ftpsserver.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 kısımlarıyla ilgili bilgi için bkz. [FTPS sunucusu desteği protokol köprüsünün temelinde](#).

V9.2.1 Tek tek dosya sunucularına dosya aktarımları sayısını sınırlamaya ilişkin senaryolar ve örnekler

Düzeltilen protokol köprüsü aracısı, bazı örneklerle birlikte **maxActiveDestinationTransfers** ve **failTransferWhenCapacityReached** öznitelikleriyle birlikte çalışır.

Protokol köprüsü aracısının maxActiveDestinationTransfers değerine dayalı olarak çalıştığını gösteren senaryolar

1. senaryo

İletişim kuralı köprüsü aracısına ilişkin ProtocolBridgeProperties.xml dosyası, iki dosya sunucusu tanımlaması içerir:

- Genel **maxActiveDestinationTransfers** özniteliğini ayarlamadınız.
- **maxActiveDestinationTransfers** özniteliğini hem fileServerA hem de FileServerB üzerinde ayarlamadınız.
- Protokol köprüsü aracısını **maxDestinationTransfers** özniteliğini varsayılan değere ayarladınız.

Protokol köprüsü aracısı **maxDestinationTransfers** özniteliğini varsayılan değer olan 25 değerine ayarladıysanız, aşağıdaki:

- Hedef aracı, iki yönetilen aktarma işlemini fileServerA' e işlemeye başlar.
- Her ikisi de transferler tamamlandı.

Bu zaman noktasında, istemci fileServerA ' in başarısız olduğunu fark eder ve ProtocolBridgeProperties.xml dosyasında fileServerA için aşağıdaki değerleri ayarlar:

```
maxActiveDestinationTransfers = 0
failTransferWhenCapacityReached = true
```

- fileServerA için başka bir aktarım daha gelir ve fileServerB için birkaç kişi gönderilir: Önceki adımda ayarlanan özelliklere dayalı olarak, fileServerA ' a yönetilen aktarım reddedilir ve başarısız olarak işaretlenir, ancak fileServerB için yapılan aktarımlar, var olan standart akışta işlenir.
- Bir süre sonra, istemci fileServerA 'in yeniden çalıştığını öğrenir, böylece istemci ProtocolBridgeProperties.xml' ta önceden eklenen değeri kaldırır ya da yorumlar. fileServerA için yeni bir yönetilen aktarım geliyor ve var olan standart akışta ele alınır.

2. senaryo

- Bir dosya sunucusu için **maxActiveDestinationTransfers** özneliğini ayarladınız ve **failTransferWhenCapacityReached** özneliğini ayarlamadınız.
- Protokol köprüsü aracı, dosya sunucusuna bu yönetilen aktarma sayısı için hedef aracı olarak işlev görmektedir.
- **maxActiveDestinationTransfers** özneliğinin değeri 1 azaltılır.

İletişim kuralı köprüsü aracı, yapılandırmasını dinamik olarak günceller ve **maxActiveDestinationTransfers** değerini hala etkin durumdayken yeni değere ayarlar. Devam etmekte olan yönetilen aktarımlar bu güncellemeye etkilenmez ve tamamlamaya izin verilir.

3. senaryo

Bir iletişim kuralı köprüsü aracı için ProtocolBridgeProperties.xml dosyası, iki dosya sunucusu tanımlaması içerir:

- Genel **maxActiveDestinationTransfers** özneliğini ayarlamadınız.
- **failTransferWhenCapacityReached** özneliğini ayarlamadınız.
- `fileServerA`'ta **maxActiveDestinationTransfers** ' a 1 olarak ayarlandınız.
- `fileServerB` üzerinde **maxActiveDestinationTransfers** özneliğini ayarlamadınız.

Protokol köprüsü aracısının **maxDestinationTransfers** özneliği 5 değerine ayarlıysa:

- İletişim kuralı köprüsü aracısından `fileServerA` değerine kadar etkin hedef aktarma sayısı üst sınırı 1 'dir (hedef aracıda 5 hedef aktarım yuvası olsa da, yönetilen aktarımlar için yalnızca 1 kullanılabilir `fileServerA`).

Bu, `fileServerA` başarısız olduğunda yararlı olur. `fileServerA` yeniden çalıştırıldıktan sonra, izin verilen hedef aktarımların tam kapasitede olmasına izin vermek için **maxActiveDestinationTransfers** değeri 5 'e artırılabilir.

- İletişim kuralı köprüsü aracısından `fileServerB` ' a kadar etkin hedef aktarma sayısı üst sınırı 5 'tür.

Bu dosya sunucusu için **maxActiveDestinationTransfers** belirlenmediği için, protokol köprüsü aracı, yönetilen aktarımlar için hedef aktarımlarının 5 'i tüm hedef yuvalarını kullanabilir.

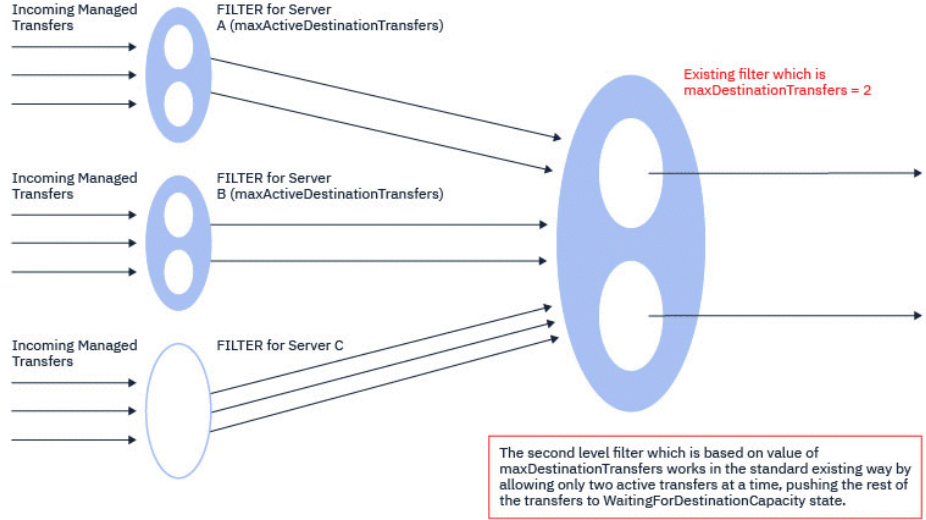
4. senaryo

Aşağıdaki şemada:

- `agent.properties` dosyasında **maxDestinationTransfers** özneliğini 2 olarak ayarladınız.
- `fileServerA`'da **maxActiveDestinationTransfers** ' yi 2 olarak ayarladınız.
- You have set the **maxActiveDestinationTransfers** attribute to 2 on `fileServerB`.
- `fileServerC` üzerinde **maxActiveDestinationTransfers** özneliğini ayarlamadınız.

SCENARIO
maxDestinationTransfers=2
fileServerA :
maxActiveDestinationTransfers=2
fileServerB :
maxDestinationTransfers=2
fileServerC :
maxDestinationTransfers not set

In both cases A and B, only two managed transfers are let in by the maxActiveDestinationTransfers filter. The rest of the transfers go into the WaitingForDestinationFileServerCapacity state.
In the case of C, the filter gives everything a passthrough since no value is set.



Çizge gösterdikçe, **maxActiveDestinationTransfers** ve **maxDestinationTransfers** öznelikleri birbirinden bağımsızdır.

Sunucuların her biri için **maxActiveDestinationTransfers** değerleri denetlenir. Bu değere bağlı olarak, aktarımların daha fazla devam etmesine izin verilir ya da **WaitingForDestinationFileServerCapacity** durumuna yönlendirilir.

İzin verilen aktarımlar, **maxDestinationTransfers**' e karşı var olan standart denetim akışından geçilir.

5. senaryo



Uyarı: You should be careful when setting the values of the **maxActiveDestinationTransfers** attributes as you must keep the value of the **maxDestinationTransfers** attribute in mind.

Bunu yapmazsanız, aşağıdaki metinde açıklanan bir durum ortaya çıkabilir:

- Genel **maxActiveDestinationTransfers** özneliği için bir değer ayarlamadınız.
- agent.properties dosyasında **maxDestinationTransfers= 2** olarak bir değer belirlediniz.
- fileServerA üzerinde **maxActiveDestinationTransfers= 2** olarak bir değer ayarladınız.
- fileServerB' ta **maxActiveDestinationTransfers** için bir değer ayarlamadınız.

Aşağıdaki olay sırasının gerçekleşeceğini varsayın:

- İletişim kuralı köprüsü aracısı, bir dosyayı fileServerA' e aktarma isteği alır. Protokol köprüsü aracısı şu anda bir şey yapmıyor, bu nedenle bu yönetilen aktarma isteğini kabul eder.

Aktarma yuvaları şu anda şöyle görünür:

- Hedef Aktarımlar: 1
- fileServerA için Hedef Aktarımlar: 1
- fileServerB için hedef aktarımlar: 0

- Şimdi, protokol köprüsü aracısı, fileServerA ile ilgili yönetilen bir aktarıma ilişkin hedef aracı olarak işlev görmesi 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 meşgul olduğundan, aracı fileServerA ' a yapılan aktarımlardan biri tamamlanincaya kadar aracı daha fazla yönetilen aktarımlara katılamaz.

- Kısa bir süre sonra fileServerA başarısız olur ve bu da iki yönetilen aktarıma neden kurtarma işlemini gerçekleştirmesini sağlar. Bu yönetilen aktarmaların kullandığı Destination transfer yuvaları bu süre içinde kullanımda kalır.
- Daha sonra, protokol köprüsü aracı, fileServerBdosyasını aktarmak için bir istek alır. Destination Transfers for fileServerB yuvalarında bu aktarıma ilişkin bir alan vardır; ancak, aracı için tüm Destination Transfer yuvaları kullanılır ve bu nedenle aktarma, daha sonra yeniden denenebilmesi için birikim birikine yerleştirilir.

Sonuç olarak, fileServerB 'a aktarma işlemi, fileServerA ' a yapılan aktarımların en az biri tamamlanincaya ve Destination Transfer yuvasını serbest bırakıncaya kadar engellenir.

Bu durumun oluşmasını önlemek için:

- Dosya sunucularındaki **maxActiveDestinationTransfers** değerini, **maxDestinationTransfers** değerinden küçük olacak şekilde ayarlayın; böylece, serbest yuvalar kalır ya da
- Tüm uç sunucular arasında **maxActiveDestinationTransfers** özniteliğinin değerini eşit olarak dağıtın.

Protokol köprüsü aracısının maxActiveDestinationTransfers özniteliğinin değerlerine dayalı davranışı

Not: Aşağıdaki tabloda listelenen tüm hata durumlarında, **maxActiveDestinationTransfers** özniteliği geçerli olmayan bir değere ayarlanmışsa, protokol köprüsü aracı bu özniteliğin ayarlanmadığını varsayar.

maxActiveDestinationTransfers	Örnek değer	Açıklama
Belirlenmedi	Belirlenmedi	Transferler her zamanki gibi gidiyor. * ftp * uç noktası için aktarma sayısına bir sınır konmaz.
Belirtildi	0	Bu belirli * ftp * uç noktasına aktarma izni yok.
Negatif değer	-1	output0.log Value -1 oturum açma hatası negatif olmayan bir tamsayı için geçerli değil. Protokol köprüsü aracı, özniteliğin ayarlanmamış olduğunu varsayar.
Tamsayı olmayan değer	abc	output0.log Value abc oturum açma hatası, bir tamsayı için geçerli değil. Protokol köprüsü aracı, özniteliğin ayarlanmamış olduğunu varsayar.
Boş	""	maxActiveDestinationTransfers özniteliğinin ' ' değeri negatif olmayan bir tamsayı için geçerli değil.
Belirtildi	5	Bu * ftp * uç noktası için herhangi bir zaman noktasında yalnızca beş etkin aktarımın çalıştırılmasına izin verir. Aşırı aktarımlar, failTransferWhenCapacityReached özniteliğinin değerine dayalı olarak yeniden denenir ya da reddedilir.

Behavior of the protocol bridge agent for the combination of the `maxActiveDestinationTransfers` and `failTransferWhenCapacityReached` attributes

<code>failTransferWhenCapacityUlaşıl</code> an değeri	<code>maxActiveDestinationTransfers</code> değeri	Sonuç
Yanlış	3	Bu uç sunucuya üç etkin aktarıma izin verilir. Başka transferler tekrar denenir.
Doğru	3	Bu uç sunucuya üç etkin aktarıma izin verilir. Başka aktarımlar reddedilir ve başarısız olarak işaretlenir.
Belirlenmedi	3	Default value yanlış for <code>failTransferWhenCapacityReached</code> is considered. Sonuç olarak, bu uç sunucuya üç etkin aktarıma izin verilir. Başka transferler tekrar denenir.
Boole değeri dışındaki değerler	Belirtildi	Error logged in output.log. <code>failTransferWhenCapacity</code> için belirtilen değer Boole değeri değil. The default value for <code>failTransferWhenCapacityReached</code> is considered.

Behavior of the protocol bridge agent for the combination of the `maxDestinationTransfers` and `failTransferWhenCapacityReached` attributes

<code>failTransferWhenCapacityUlaşıl</code> an değeri	<code>maxDestinationAktarmalar</code> değeri	Sonuç
Doğru	10	Eşzamanlı etkin aktarımların sayısı 10 'a ulaştığında, protokol köprüsü aracısı 11 th tarafından yönetilen aktarım başarısız olur.
Yanlış	10	Var olan bir davranış. Eşzamanlı etkin aktarımların sayısı 10 'a ulaştığında, 11 th yönetilen aktarımı, yuva serbest bırakılacak bir yuva için kuyruklanır.
Belirlenmedi	10	Var olan davranış

hata iletileri

Var olan ileti:

BFGS0082I

İletişim kuralı köprüsü aracı, **maxDestinationTransfers** öznelteisinde tanımlı aktarma sayısı üst sınırını zaten çalıştırdığında, protokol köprüsü aracı aktarımı reddettiğinde kaynak aracının output0.log dosyasında oturum açmış olur.

Yeni iletiler:

BFGS0085I

Protokol köprüsü aracı yönetilen bir aktarımı reddettiğinde ve yeniden denemediğinde kaynak aracının output0.log dosyasında oturum açmaktadır.

BFGS0086I

Protokol köprüsü aracı bir yönetilen aktarımı reddettiğinde ve yeniden denediğinde, kaynak aracının output0.log dosyasında oturum açılır ve hedef öge dosya sunucusu adını içermez.

BFGS0084E

Is logged in the Explorer and audit.xml file when the protocol bridge agent rejects, for exceeding the maximum number of concurrent transfers specified in the **maxActiveDestinationTransfers** attribute, and marks a managed transfer as failed.

BFGS0087E

Is logged in the Explorer and audit.xml file when the protocol bridge agent rejects, for exceeding the maximum number of destination transfers specified in the **maxActiveDestinationTransfers** attribute, and marks a managed transfer as failed.

BFGS0088W

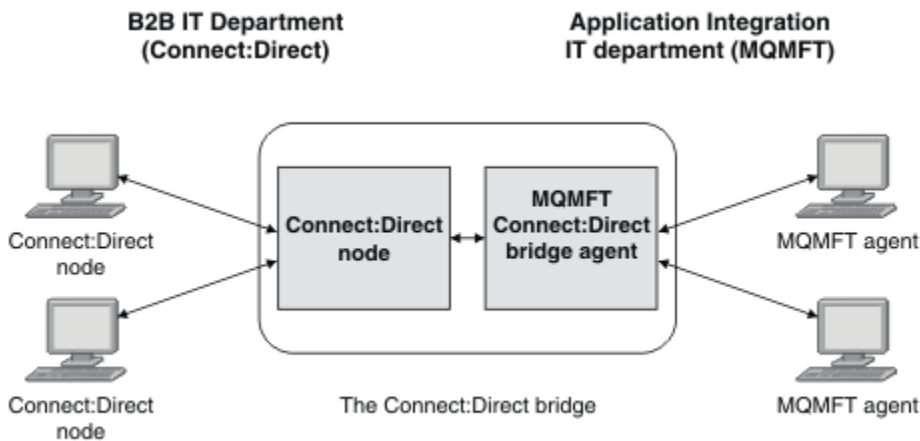
Is logged in the output0.log, when the value of the **maxActiveDestinationTransfers** attribute exceeds the value of the **maxDestinationTransfers** attribute.

BFGS0089I

Hedef protokol köprüsü aracısının output0.log dosyasında, IBM MQ 9.2.1 konumunda olmayan bir kaynak aracı çalıştığında ya da daha sonra günlüğe kaydedilmektedir.

Connect:Direct köprüsü

Var olan bir IBM Sterling Connect:Direct ağına dosya aktarabilir ya da bu ağdan dosya aktarabilirsiniz. Use the Connect:Direct bridge, which is a component of Managed File Transfer, to transfer files between MFT and IBM Sterling Connect:Direct.



Bu çizgede, iki bölüm, B2B IT departmanı ve Uygulama Tümüleşimi BT departmanı arasında bir MFT Connect:Direct köprüsü gösterilmektedir. B2B BT departmanı, dosyaları şirketin iş ortaklarına aktarmak için Connect:Direct ' u kullanır. Uygulama Tümüleşimi BT bölümü, ileti sistemi altyapısı olarak IBM MQ 'yi kullanır ve bu nedenle, dosya aktarımı çözümü olarak son zamanlarda Managed File Transfer ' yi seçmiştir.



Using the MFT Connect:Direct bridge, the two departments can transfer files between the Connect:Direct network in the B2B IT department and the MFT network in the Application Integration IT department. The Connect:Direct bridge is a component of Managed File Transfer, which includes an MFT agent that

communicates with a Connect:Direct node. MFT aracı, Connect:Direct düğümü ile aktarımlara ayrılmıştır ve Connect:Direct köprüsü aracı olarak bilinir.

Connect:Direct köprüsü, Managed File Transfer' in Service and Agent bileşenlerinin bir parçası olarak kullanılabilir ve aşağıdaki görevler için kullanılabilir:


1. Use Managed File Transfer commands to initiate a transfer of a file, or multiple files, from an MFT agent to a Connect:Direct node.
2. Use Managed File Transfer commands to initiate a transfer of a file, or multiple files, from a Connect:Direct node to an MFT agent.
3. Kullanıcı tanımlı bir Connect:Direct işlemini başlatan bir dosya aktarımı başlatmak için Managed File Transfer komutlarını kullanın.
4. Bir MFT dosya aktarma isteği sunmak için Connect:Direct işlemini kullanın.

A Connect:Direct bridge can transfer files to or from only Connect:Direct nodes. Connect:Direct köprüsü, yalnızca Connect:Direct işlemi tarafından gönderilen bir aktarımın bir parçası olarak dosyaları yerel dosya sistemine aktarabilir.

 z/OS sistemindeki bir Connect:Direct düğümünde bulunan bir veri kümesinden ya da birdatadüğümünde bulunan bir veri kümesinden aktarmak için Connect:Direct köprüsünü kullanabilirsiniz. Yalnızca Managed File Transfer araçlarını içeren veri kümesi aktarımlarıyla karşılaştırıldığında, davranışlarda bazı farklılıklar vardır. Daha fazla bilgi için  Veri kümelerini Connect:Direct düğümlerine vefromdüğümlerine aktarılması başlıklı konuya bakın.

Desteklenen platformlar

Connect:Direct köprüsü, bir MFT Connect:Direct köprüsü aracısından ve bir Connect:Direct düğümlerinden oluşur. The agent is supported on Windows and Linux for x86-64. The node is supported on the platforms that are supported for IBM Sterling Connect:Direct for Windows and IBM Sterling Connect:Direct for UNIX. For instructions on creating the Connect:Direct bridge agent and configuring a Connect:Direct node for the agent to communicate with, see [Connect:Direct köprüsünü yapılandırma](#).

The Connect:Direct bridge can transfer files to and from Connect:Direct nodes that are running as part of a Connect:Direct for Windows or Connect:Direct for UNIX  Ya da z/OS için Connect:Direct Service installation. Desteklenen Connect:Direct sürümlerine ilişkin ayrıntılar için [System Requirements for IBM MQweb 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 aynı 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.

Bir dosyanın bir Connect:Direct düğümüne aktarılması

You can transfer a file from a Managed File Transfer agent to a Connect:Direct node using the Connect:Direct bridge. Specify a Connect:Direct node as the destination of the transfer by specifying

the Connect:Direct bridge agent as the destination agent and specifying the destination file in the form `connect_direct_node_name:file_path`.

Başlamadan önce

Bir dosyayı aktarmadan önce, Managed File Transfer bileşeninin bir bileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Daha fazla bilgi için [Connect:Direct köprüsünün yapılandırılması](#) başlıklı konuya bakın.

Bu görev hakkında

Bu örnekte, Connect:Direct köprüsü aracısının adı CD_BRIDGE olarak adlandırılır. Kaynak aracıya FTE_AGENT adı verilir ve WMQFTE sürümü herhangi bir sürüm olabilir. Hedef Connect:Direct düğümü CD_NODE1 olarak adlandırılır. Aktarılabilecek kütük, FTE_AGENT dosyasının bulunduğu sistemde /home/helen/file.log dosya yolunda bulunur. Dosya, CD_NODE1 ' un çalıştığı sistemde /files/data.log dosya yoluna aktarılır.

Yordam

1. Use the `fteCreateTransfer` command with the value for the **-df** (destination file) parameter in the form `connect_direct_node_name:file_path` and the value of the **-da** (destination agent) parameter specified as the name of the Connect:Direct bridge agent.

Not: The Connect:Direct node specified by `connect_direct_node_name` is the node that you want the file to be transferred to, not the Connect:Direct node that operates as part of the Connect:Direct bridge.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
-df CD_NODE1:/files/data.log /home/helen/file.log
```

Daha fazla bilgi için bakınız: [fteCreateTransfer: start a new file transfer](#).

2. Kaynak aracı FTE_AGENT, dosyayı Connect:Direct köprü aracısı CD_BRIDGE ' e aktarır. Dosya, `cdTmpDir` aracı özelliği tarafından tanımlanan konumda, Connect:Direct köprü aracısının çalıştığı sistemde geçici olarak saklanır. The Connect:Direct bridge agent transfers the file to the Connect:Direct node CD_NODE1.

İlgili kavramlar

“Connect:Direct köprüsü” sayfa 276

Var olan bir IBM Sterling Connect:Direct ağına dosya aktarabilir ya da bu ağdan dosya aktarabilirsiniz. Use the Connect:Direct bridge, which is a component of Managed File Transfer, to transfer files between MFT and IBM Sterling Connect:Direct.

İlgili görevler

“Transferring a file from a Connect:Direct node” sayfa 278

Connect:Direct düğümünü kullanarak bir dosyayı Connect:Direct düğümünden Managed File Transfer Agent ' a aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprü aracısını belirterek ve kaynak belirtimini `connect_direct_node_name:file_path` biçiminde belirterek, aktarımın kaynağı olarak bir Connect:Direct düğümü belirtebilirsiniz.

İlgili başvurular

[The MFT agent.properties file](#)

Transferring a file from a Connect:Direct node

Connect:Direct düğümünü kullanarak bir dosyayı Connect:Direct düğümünden Managed File Transfer Agent ' a aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprü aracısını belirterek ve kaynak belirtimini `connect_direct_node_name:file_path` biçiminde belirterek, aktarımın kaynağı olarak bir Connect:Direct düğümü belirtebilirsiniz.

Başlamadan önce

Bir dosyayı aktarmadan önce, Managed File Transfer bileşeninin bir bileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Bkz. [Connect:Direct köprüsünü yapılandırma](#).

Bu görev hakkında

Bu örnekte, Connect:Direct köprüsü aracısının adı CD_BRIDGE olarak adlandırılır. The destination agent is called FTE_AGENT and can be any version of Managed File Transfer. Kaynak Connect:Direct düğümü CD_NODE1 olarak adlandırılır. The file to be transferred is located at the file path /home/brian/in.file on the system where CD_NODE1 is located. Dosya, FT_AGENT ' in çalıştığı sistemde /files/out.file dosya yoluna aktarılır.

Yordam

Use the **fteCreateTransfer** command with the value for the source specification in the form *connect_direct_node_name:file_path* and the value of the **-sa** parameter specified as the name of the Connect:Direct bridge agent.

Not: The Connect:Direct node specified by *connect_direct_node_name* is the node that you want the file to be transferred from, not the Connect:Direct node that operates as part of the Connect:Direct bridge. Örneğin:

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_AGENT  
-df /files/out.file CD_NODE1:/home/brian/in.file
```

Daha fazla bilgi için bakınız: **[fteCreateTransfer](#)**: start a new file transfer.

Sonuçlar

Connect:Direct köprü aracı CD_BRIDGE, dosyayı Connect:Direct düğümünden CD_NODE1' den 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 köprüsünün aktarılmasını tamamladığında, Connect:Direct köprüsü daha sonra dosyayı hedef aracıya FT_AGENT adlı hedef aracıya gönderir ve geçici konumdan dosyayı siler.

İlgili kavramlar

“Connect:Direct köprüsü” sayfa 276

Var olan bir IBM Sterling Connect:Direct ağına dosya aktarabilir ya da bu ağdan dosya aktarabilirsiniz. Use the Connect:Direct bridge, which is a component of Managed File Transfer, to transfer files between MFT and IBM Sterling Connect:Direct.

İlgili başvurular

[The MFT agent.properties file](#)

Bir veri kümesinin z/OS üzerindeki bir Connect:Direct düğümüne aktarılması

You can transfer a data set from a Managed File Transfer agent on z/OS to a Connect:Direct node on z/OS by using a Connect:Direct bridge that is located on a Windows or Linux system.

Başlamadan önce

Bir dosyayı aktarmadan önce, Managed File Transfer bileşeninin bir bileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Bkz. [Connect:Direct köprüsünü yapılandırma](#).

Bu görev hakkında

Bu örnekte, aktarımın hedefini belirtmek için **-df** parametresi kullanılır. The parameter **-df** is valid for use when the source agent of the transfer is any version of Managed File Transfer. Kaynak aracı IBM WebSphere MQ File Transfer Edition 7.0.4 ya da daha sonraki bir yayın düzeyiyse, bunun yerine **-ds** parametresini kullanabilirsiniz. Kaynak aracıya FTE_ZOS1 adı verilir ve bir IBM WebSphere MQ File

Transfer Edition 7.0.3 aracıdır. Connect:Direct köprüsü aracısının adı CD_BRIDGE ve bir Linux sisteminde bulunur. Hedef Connect:Direct düğümü CD_ZOS2 olarak adlandırılır. Hem kaynak aracı, hem de hedef Connect:Direct düğümü, z/OS sistemlerinde bulunur. The data set to be transferred is located at //FTEUSER.SOURCE.LIB on the system where FTE_ZOS1 is located. The data set is transferred to the data set //CDUSER.DEST.LIB on the system where CD_ZOS2 is located.

Not: IBM WebSphere MQ File Transfer Edition (FTE) artık desteklenen bir ürün değildir. IBM MQ'da FTE'den Managed File Transfer bileşenine geçiş için [Migrating Managed File Transfer](#) başlıklı konuya bakın.

Yordam

1. Use the fteCreateTransfer command with the value for the **-df** parameter in the form: `connect_direct_node_name:data_set_name;attributes` and the value of the **-da** (destination agent) parameter specified as the name of the Connect:Direct bridge agent.

The Connect:Direct node specified by `connect_direct_node_name` is the node that you want the data set to be transferred to, not the Connect:Direct node that operates as part of the Connect:Direct bridge.

`data_set_name` tarafından belirtilen veri kümesi adı mutlak değil, mutlak olmalıdır. Connect:Direct , veri kümesi adının önekini kullanıcının adıyla öneksiz yapmaz.

```
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 bakınız: **fteCreateTransfer: start a new file transfer.**


2. Kaynak aracı FTE_ZOS1 , verileri, veri kümesindeki verileri Connect:Direct köprü aracısı CD_BRIDGE 'e aktarır. Veriler geçici olarak, cdTmpDir aracısı özelliği tarafından tanımlanan konumda Connect:Direct köprü aracısının çalıştığı sistemde düz bir dosya olarak saklanır. The Connect:Direct bridge agent transfers the data to the Connect:Direct node CD_ZOS2. Aktarma işlemi tamamlandığında, düz dosya, Connect:Direct köprü aracısının çalıştığı sistemden silinir.

İlgili kavramlar

“Connect:Direct köprüsü” sayfa 276

Var olan bir IBM Sterling Connect:Direct ağına dosya aktarabilir ya da bu ağdan dosya aktarabilirsiniz. Use the Connect:Direct bridge, which is a component of Managed File Transfer, to transfer files between MFT and IBM Sterling Connect:Direct.

İlgili görevler

 [Transferring data sets to and from Connect:Direct nodes](#)

İlgili başvurular

 [BPXWDYN özellikleri, MFT ile birlikte kullanmamanız gerekir](#)

Birden çok dosyanın bir Connect:Direct düğümüne aktarılması

Connect:Direct köprüsünü kullanarak, birden çok dosyayı bir Managed File Transfer Agent düğümünden bir Connect:Direct düğümüne aktarabilirsiniz. Bir Connect:Direct düğümünü çoklu dosya aktarımın hedefi olarak kullanmak için, hedef aracı olarak Connect:Direct köprüsü aracısını belirtin ve hedef dizini `connect_direct_node_name:directory_path` biçiminde belirtin.

Başlamadan önce

Dosyaları aktarmadan önce, Managed File Transfer bileşeninin bir bileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Bkz. [Connect:Direct köprüsünü yapılandırma](#).

Bu görev hakkında

Bu örnekte, kaynak aracıya FTE_AGENT olarak adlandırılır. Connect:Direct köprüsü aracısının adı CD_BRIDGE olarak adlandırılır. Hedef Connect:Direct düğümü CD_NODE1 olarak adlandırılır. Aktarılacak

dosyalar, FTE_AGENT 'in bulunduğu sistemde /home/jack/data.log, /logs/log1.txt ve /results/latest ' dir. Dosyalar, CD_NODE1 ' un çalıştığı sistemde /in/files dizinine aktarılır.

Yordam

Use the `fteCreateTransfer` command with the value for the **-dd** (destination directory) parameter in the form `connect_direct_node_name:directory_path`. Connect:Direct köprüsü aracısının adı olarak **-da** (hedef aracı) parametresinin değerini belirtin.

Not: The Connect:Direct node specified by `connect_direct_node_name` is the node that you want the files to be transferred to, not the Connect:Direct node that operates as part of the Connect:Direct bridge.

```
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 bakınız: **[fteCreateTransfer](#)**: start a new file transfer.

Sonuçlar

Kaynak aracı FTE_AGENT, ilk dosyayı Connect:Direct köprü aracı CD_BRIDGE 'e aktarır. Connect:Direct köprüsü aracı, dosyayı geçici olarak `cdTmpDir` özelliği ile tanımlanan konumda saklar. Dosya, kaynak araçından Connect:Direct köprüsünden tamamen aktarıldığında, Connect:Direct köprü aracı, dosyayı `cdNode` aracı özelliği tarafından tanımlanan Connect:Direct düğümüne gönderir. Bu düğüm, dosyayı hedef Connect:Direct düğümüne CD_NODE1 'e gönderir. The Connect:Direct bridge agent deletes the file from the temporary location when the transfer between the two Connect:Direct nodes is complete. Bu işlem, belirtilen her kaynak dosya için yinelenir.

İlgili kavramlar

[“Connect:Direct köprüsü” sayfa 276](#)

Var olan bir IBM Sterling Connect:Direct ağına dosya aktarabilir ya da bu ağdan dosya aktarabilirsiniz. Use the Connect:Direct bridge, which is a component of Managed File Transfer, to transfer files between MFT and IBM Sterling Connect:Direct.

İlgili görevler

[“Bir dosyanın bir Connect:Direct düğümüne aktarılması” sayfa 277](#)

You can transfer a file from a Managed File Transfer agent to a Connect:Direct node using the Connect:Direct bridge. Specify a Connect:Direct node as the destination of the transfer by specifying the Connect:Direct bridge agent as the destination agent and specifying the destination file in the form `connect_direct_node_name:file_path`.

[“Joker karakterler kullanarak birden çok dosya Connect:Direct 'e aktarılması” sayfa 282](#)

Birden çok dosyayı bir Managed File Transfer aracısından bir Connect:Direct düğümüne aktarmak için Connect:Direct köprüsünü kullanın. **fteCreateTransfer** komutuna sağladığınız kaynak belirtiminde genel arama karakterleri kullanabilirsiniz. Genel arama karakterleri içeren tüm Managed File Transfer aktarmalarında olduğu gibi, yalnızca dosya yolunun son bölümü bir genel arama karakteri içerebilir. Örneğin, `/abc/def*` geçerli bir dosya yolu ve `/abc*/def` geçerli değil.

[“Transferring a file from a Connect:Direct node” sayfa 278](#)

Connect:Direct düğümünü kullanarak bir dosyayı Connect:Direct düğümünden Managed File Transfer Agent 'a aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprü aracısını belirterek ve kaynak belirtimini `connect_direct_node_name:file_path` biçiminde belirterek, aktarımın kaynağı olarak bir Connect:Direct düğümü belirtebilirsiniz.

[“Birden çok dosyanın bir Connect:Direct düğümünden aktarılması” sayfa 282](#)

Connect:Direct köprüsünü kullanarak, birden çok dosyayı bir Connect:Direct düğümünden bir Managed File Transfer Agent düğümüne aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprüsü aracısını belirterek ve `connect_direct_node_name:file_path` biçiminde bir ya da daha fazla kaynak belirtimi belirtilerek, birden çok dosya aktarımın kaynağı olarak bir Connect:Direct düğümü belirtebilirsiniz.

İlgili başvurular

[The MFT agent.properties file](#)

Birden çok dosyanın bir Connect:Direct düğümünden aktarılması

Connect:Direct köprüsünü kullanarak, birden çok dosyayı bir Connect:Direct düğümünden bir Managed File Transfer Agent düğümüne aktarabilirsiniz. Kaynak aracı olarak Connect:Direct köprüsü aracısını belirterek ve `connect_direct_node_name:file_path` biçiminde bir ya da daha fazla kaynak belirtimi belirtilerek, birden çok dosya aktarımın kaynağı olarak bir Connect:Direct düğümü belirtebilirsiniz.

Başlamadan önce

Bir dosyayı aktarmadan önce, Managed File Transfer bileşeninin bir bileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Bkz. [Connect:Direct köprüsünü yapılandırma](#).

Bu görev hakkında

Bu örnekte, Connect:Direct köprüsü aracısının adı CD_BRIDGE olarak adlandırılır. Hedef aracıya FTE_Z adı verilir ve bir z/OS sisteminde çalışıyor. Kaynak Connect:Direct düğümü CD_NODE1 olarak adlandırılır. The files to be transferred are located at the file paths /in/file1, /in/file2, and /in/file3 on the system where CD_NODE1 is located. The files are transferred to the partitioned data set //OBJECT.LIB on the system where FTE_Z is running.

Yordam

`fteCreateTransfer` komutunu, `connect_direct_node_name:file_path` formundaki kaynak belirtilere ve Connect:Direct köprü aracısının adı olarak belirtilen **-sa** parametresinin değerini kullanarak kullanın.

Not: The Connect:Direct node specified by `connect_direct_node_name` is the node that you want the files to be transferred from, not the Connect:Direct node that operates as part of the Connect:Direct bridge.

```
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 bakınız: [fteCreateTransfer: start a new file transfer](#).

Sonuçlar

The Connect:Direct bridge agent CD_BRIDGE requests the first file from the Connect:Direct node CD_NODE1. 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 köprüsünün aktarılmasını tamamladığında, Connect:Direct köprüsü dosyayı hedef aracıya FTE_Z hedef aracısına gönderir ve dosyayı geçici konumdan siler. Bu işlem, belirtilen her kaynak dosya için yinelenir.

İlgili kavramlar

[“Connect:Direct köprüsü” sayfa 276](#)

Var olan bir IBM Sterling Connect:Direct ağına dosya aktarabilir ya da bu ağdan dosya aktarabilirsiniz. Use the Connect:Direct bridge, which is a component of Managed File Transfer, to transfer files between MFT and IBM Sterling Connect:Direct.

İlgili başvurular

[The MFT agent.properties file](#)

Joker karakterler kullanarak birden çok dosya Connect:Direct ' e aktarılması

Birden çok dosyayı bir Managed File Transfer aracısından bir Connect:Direct düğümüne aktarmak için Connect:Direct köprüsünü kullanın. **fteCreateTransfer** komutuna sağladığınız kaynak belirtiminde genel arama karakterleri kullanabilirsiniz. Genel arama karakterleri içeren tüm Managed File Transfer aktarmalarında olduğu gibi, yalnızca dosya yolunun son bölümü bir genel arama karakteri içerebilir. Örneğin, `/abc/def*` geçerli bir dosya yolu ve `/abc*/def` geçerli değil.

Başlamadan önce

Bir dosyayı aktarmadan önce, Managed File Transfer bileşeninin bir bileşeni olan Connect:Direct köprüsünü yapılandırmanız gerekir. Daha fazla bilgi için [Connect:Direct köprüsünün yapılandırılması](#) başlıklı konuya bakın.

Bu görev hakkında

Bu örnekte, kaynak aracıya FTE_AGENT olarak adlandırılır ve Connect:Direct köprü aracı CD_BRIDGE olarak adlandırılır. Hedef Connect:Direct düğümü CD_NODE1 olarak adlandırılır. Aktarılabacak dosyalar, FT_AGENT ' in bulunduğu sistemde /reports dizininde bulunur. Yalnızca, report ile başlayan, ardından iki karakter ve .log sonekiyle başlayan dosyalar aktarılır. Örneğin, /reports/report01.log dosyası aktarılsa da /reports/report1.log dosyası aktarılmaz. Dosyalar, CD_NODE1 ' un çalıştığı sistemde /home/fred dizinine aktarılır.

Yordam

1. Use the fteCreateTransfer command with the value for the **-dd** (destination directory) parameter in the form *connect_direct_node_name:directory_path*. **-da** (hedef aracı) parametresi için, Connect:Direct köprü aracısını belirtin.

Not: The Connect:Direct node specified by *connect_direct_node_name* is the node that you want the files to be transferred to, not the Connect:Direct node that operates as part of the Connect:Direct bridge.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/home/fred "/reports/report?? .log"
```

Daha fazla bilgi için bakınız: **fteCreateTransfer: start a new file transfer**.

2. The source agent FTE_AGENT transfers the first file that matches the pattern /reports/report?? .log to the Connect:Direct bridge agent CD_BRIDGE. Connect:Direct köprüsü aracı, dosyayı geçici olarak cdTmpDir özelliği ile tanımlanan konumda saklar. Dosya, kaynak araçından Connect:Direct köprüsünden tamamen aktarıldığında, Connect:Direct köprü aracı, dosyayı cdNode aracı özelliği tarafından tanımlanan Connect:Direct düğümüne gönderir. Bu düğüm, dosyayı hedef Connect:Direct düğümüne CD_NODE1 ' e gönderir. The Connect:Direct bridge agent deletes the file from the temporary location when the transfer between the two Connect:Direct nodes is complete. Bu işlem, /reports/report?? .log genel arama karakteriyle eşleşen her kaynak dosya için yinelenir.

Not: /reports/report?? .log örüntüleriyle eşleşen dosyaların listesi, FTE_AGENT kaynak aracısının bulunduğu sistemin işletim sistemine bağlı olarak değişiklik gösterir.

- Kaynak aracı, Windows işletim sistemiyle çalışan bir sistemde bulunuyorsa, kalıp eşleştirme büyük ve küçük harfe duyarlı değildir. The pattern matches all files in the /reports directory with a file name of the form report followed by two characters and a suffix of .log, regardless of the case that the letters are in. Örneğin, Report99 .Log bir eşleşir.
- Kaynak aracı, Linux ya da UNIX işletim sistemi olan bir sistemde bulunuyorsa, kalıp eşleştirme büyük ve küçük harfe duyarlıdır. Örüntü yalnızca, /reports dizinindeki dosya adı report ve ardından iki karakter ve .log soneki olan dosyalarla eşleşir. Örneğin, reportAB .log bir eşleşmeydir, ancak reportAB .LOG ve Report99 .Log eşleşmez.

Connect:Direct düğümlerine vefromdüğümlerine aktarımlar için kurtarma ve yeniden başlatma

Managed File Transfer might be unable to connect to your IBM Sterling Connect:Direct node during a transfer; for example, if the node becomes unavailable. Managed File Transfer , aktarma işlemini gerçekleştirmeyi dener ya da aktarma başarısız olur ve hata iletisi üretilir.

Connect:Direct düğümü kullanılamaz duruma gelirse

Connect:Direct düğümü kullanılamaz duruma gelirse; örneğin, bir ağ ya da güç kesintisi nedeniyle, Managed File Transfer aşağıdaki şekillerde bir dosya aktarımı kurtarır:

- If Managed File Transfer has not previously successfully connected to the Connect:Direct node as part of this transfer request, the transfer is tried again for a length of time determined by the values of the **cdMaxConnectionRetries** and **recoverableTransferRetryInterval properties**. Bu özellikler, Connect:Direct köprü aracı için agent.properties dosyasında belirtilir. Başarısız girişimler sayısı **cdMaxConnectionRetries property** değerine ulaştığında, aktarma başarısız olur ve bir hata iletisi üretilir. Varsayılan değer olarak, aktarma işlemi, girişimler arasında 60 saniye ile süresiz olarak denir.
- Managed File Transfer , bu aktarma isteğinin bir parçası olarak önceden Connect:Direct düğümüne başarıyla bağlandıysa, **cdMaxPartialWorkConnectionRetries** ve **recoverableTransferRetryInterval** özelliklerinin değerleri tarafından belirlenen süre boyunca aktarma işlemi yeniden denir. Başarısız girişimler sayısı **cdMaxPartialWorkConnectionRetries** özelliğinin değerine ulaştığında, aktarma başarısız olur ve bir hata iletisi üretilir. Varsayılan değer olarak, aktarma işlemi, girişimler arasında 60 saniye ile süresiz olarak denir.
- Belirli Connect:Direct düğümü hatası tipleri için, örneğin, zorla durdurulan düğüm, düğüm kurtarıldığında Connect:Direct işlemleri Held Due to Error (HE) durumuna geçmektedir. Düğüm kurtarıldıktan sonra, Managed File Transfer , dosya aktarımı ile ilgili tüm Connect:Direct işlemlerini otomatik olarak sürdürür ve HE durumunda olur.
- Aktarma başarısız olursa, aktarıyla ilgili geçici dosyalar, Connect:Direct köprüsünü barındıran sistemden silinir. Bu geçici dosyaların konumu, **cdTmpDir** özelliği tarafından tanımlanır.
- Aktarım Managed File Transfer 'tan Connect:Direct' a ve bir kaynak silme işlemi belirtilirse, aktarma başarısız olursa kaynak dosyalar silinmez.

Connect:Direct düğümü kullanıcı kimlik bilgileri geçersizse

Kullanıcının kimlik bilgileri düğüm tarafından reddedildiğinden Managed File Transfer Connect:Direct düğümüne bağlanmazsa, aktarma başarısız olur ve bir hata iletisi üretilir. Bu durumda, Connect:Direct düğümü için doğru kullanıcı kimlik bilgilerini sağlamış olup olmadığınızı denetleyin. Daha fazla bilgi için bakınız: [Mapping credential for Connect:Direct](#).

Connect:Direct köprüsü aracı kullanılamaz duruma gelirse

Connect:Direct köprüsü aracı kullanılamaz duruma gelirse, devam eden dosya aktarımları standart Managed File Transfer aktarımları ile aynı şekilde kurtarılır. Daha fazla bilgi için bkz [“MFT kurtarma ve yeniden başlatma”](#) sayfa 290.

İlgili kavramlar

[“Connect:Direct köprüsü”](#) sayfa 276

Var olan bir IBM Sterling Connect:Direct ağına dosya aktarabilir ya da bu ağdan dosya aktarabilirsiniz. Use the Connect:Direct bridge, which is a component of Managed File Transfer, to transfer files between MFT and IBM Sterling Connect:Direct.

[“MFT kurtarma ve yeniden başlatma”](#) sayfa 290

Aracınız ya da kuyruk yöneticiniz herhangi bir nedenle kullanılamazsa, örneğin bir güç ya da ağ hatası nedeniyle, Managed File Transfer bu senaryolarda aşağıdaki gibi kurtarılır:

İlgili görevler

[Connect:Direct köprüsünü yapılandırma](#)

İlgili başvurular

[The MFT agent.properties file](#)

Kullanıcı tanımlı bir Connect:Direct işleminin dosya aktarma isteğinden sunulması

Connect:Direct köprü aracısından, kullanıcı tanımlı Connect:Direct işlemini, dosya aktarımın bir parçası olarak çağıran bir aktarım için aktarma isteği sunabilirsiniz.

By default, when you submit a file transfer request for a transfer that goes through the Connect:Direct bridge, the Connect:Direct bridge agent generates the Connect:Direct process that is used to transfer the file to or from the remote Connect:Direct node.

However, you can configure the Connect:Direct bridge agent to instead call a user-defined Connect:Direct process by using the `ConnectDirectProcessDefinition.xml` file.

ConnectDirectProcessDefinition.xml dosyası

fteCreateCDAgent komutu, `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` aracı yapılandırma dizininde `ConnectDirectProcessDefinitions.xml` dosyasını yaratır. Before you can call user-defined Connect:Direct processes from the Connect:Direct bridge agent, you must set up process definitions by editing this file.

Dosya, bir aktarımın parçası olarak çağrılan bir ya da daha fazla Connect:Direct işleminin yerini içeren bir ya da daha fazla süreç kümesini tanımlar. Her süreç kümesi bir dizi koşul içerir. Aktarma işlemi, süreç kümesinin tüm koşullarını karşılıyorsa, aktarma işlemi tarafından hangi Connect:Direct işlemlerinin çağrıldığı belirtilmek üzere işlem kümesi kullanılır. Daha fazla bilgi için bkz [“ConnectDirectProcessDefinition.xml dosyası kullanılarak başlatılacak Connect:Direct işlemi belirtme” sayfa 285.](#)

İçsel simgesel değişkenler

You can use the intrinsic symbolic variables that are defined by Managed File Transfer to substitute values into user-defined Connect:Direct processes. To follow the Connect:Direct naming convention, all intrinsic symbolic variables used by Managed File Transfer have the format %FTE followed by five uppercase alphanumeric characters.

Dosyaları Connect:Direct düğümünden Connect:Direct köprü sistemine aktarmak için bir işlem oluştururken, Connect:Direct işleminde DOSYA değeri olarak %FTETFILE adlı iç değişkeni 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 ve ağından aktarma işlemleri için kullandığı geçici dosya yollarını içerir.

İçsel simgesel değişkenler hakkında 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`.

ConnectDirectProcessDefinition.xml dosyası kullanılarak başlatılacak Connect:Direct işlemi belirtme

Managed File Transfer aktarımın bir parçası olarak başlatılacak Connect:Direct işlemi belirtin. Managed File Transfer , süreç tanımlamalarını belirlemek için düzenleyebileceğiniz bir XML dosyası sağlar.

Bu görev hakkında

fteCreateCDAgent komutu, `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` aracı yapılandırma dizininde `ConnectDirectProcessDefinitions.xml` dosyasını yaratır. Before you can call user-defined Connect:Direct processes from the Connect:Direct bridge agent, you must set up process definitions by editing this file.

Connect:Direct köprüsü aracılığıyla aktarımın bir parçası olarak çağırmak istediğiniz her işlem için aşağıdaki adımları gerçekleştirin:

Yordam

1. Connect:Direct köprüsü aracısının aktarımın 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 düzenleyicide açın.
 3. Bir `<processSet>` ögesi yaratın.
 4. `<processSet>` ögesinin içinde bir `<condition>` ögesi yaratın.
 5. `<condition>` ögesinin içinde, aktarma isteğinin, Adım 1 'de tanımladığınız Connect:Direct işlemi çağırma için eşleşmesi gereken bir koşulu tanımlayan bir ya da daha çok öge yaratın. Bu ögeler `<match>` ögeleri ya da `<defined>` ögesi olabilir.
 - Bir değişkenin değerinin bir kalıpla eşleşmesi gerektiğini belirtmek için bir `<match>` ögesi kullanın. `<match>` ögesini aşağıdaki özniteliklerle yaratın:
 - `variable` -değeri karşılaştırılan değişkenin adı. Değişken, içsel bir simgedir. Daha fazla bilgi için bakınız: [Yerine koyma değişkenleri, kullanıcı tanımlı Connect:Direct süreçleriyle kullanılır.](#)
 - `value` -belirtilen değişkenin değeriyle karşılaştırılacak örüntü.
 - İsteğe bağlı: `pattern` - `value` özniteliğinin değeri tarafından kullanılan örüntünün tipi. Bu örüntü tipi `wildcard` ya da `regex` olabilir. Bu öznitelik isteğe bağlıdır ve varsayılan değer `wildcard` 'dir.
 - Bir değişkenin tanımlanmış bir değere sahip olması gerektiğini belirtmek için bir `<defined>` ögesi kullanın. `<defined>` ögesini aşağıdaki öznitelikle yaratın:
 - `variable` -tanımlı bir değeri olması gereken değişkenin adı. Değişken, içsel bir simgedir. Daha fazla bilgi için bakınız: [Yerine koyma değişkenleri, kullanıcı tanımlı Connect:Direct süreçleriyle kullanılır.](#)
- `<condition>` ögesi içinde belirtilen koşullar, mantıksal AND ile birleştirilir. Connect:Direct köprüsü aracısının bu `<processSet>` ögesi tarafından belirtilen işlemi çağırmasıyla ilgili tüm koşullar karşılanmalıdır. Bir `<condition>` ögesi belirtmezseniz, süreç kümesi tüm aktarmalarla eşleşir.
6. `<processSet>` ögesinin içinde bir `<process>` ögesi yaratın.
 7. `<process>` ögesinin içinde bir `<transfer>` ögesi yaratın.

The transfer element specifies the Connect:Direct process that the Connect:Direct bridge agent calls as part of the transfer. `<transfer>` ögesini aşağıdaki öznitelikle yaratın:

 - `process` - Adım 1 'de tanımladığınız Connect:Direct işleminin yeri. Bu dosyanın konumu, mutlak bir yol ile ya da `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` diziniyle görel bir konumdur.

Sonuçlar

Bir koşul eşleşmesi aranırken, Connect:Direct köprü aracısı, dosyanın başlangıcından dosyanın sonuna kadar arama yapar. Bulunan ilk eşleşme, kullanılan ilk eşleşir.

İlgili görevler

[Connect:Direct köprüsünü yapılandırma](#)

İlgili başvurular

[Connect:Direct süreç tanımlaması dosyası biçimi](#)

[fteCreateCDAgent: bir Connect:Direct köprüsü aracısı oluşturma](#)

Using intrinsic symbolic variables in Connect:Direct processes that are called by Managed File Transfer

Bir Managed File Transfer aktarımından kullanıcı tanımlı bir Connect:Direct işlemi çağırabilir ve süreç tanımlamasındaki içsel simgesel değişkenler kullanarak Connect:Direct işlemine aktarımdan bilgi geçirebilirsiniz.

Bu görev hakkında

Bu örnek, kullanıcı tanımlı bir Connect:Direct işlemine Managed File Transfer aktarımından bilgi geçirmek için içsel simgesel değişkenleri kullanır. Managed File Transfer tarafından kullanılan içsel simgesel değişkenlerle ilgili daha fazla bilgi için bakınız: [Substitute definases for use with user-definive Connect:Direct processes.](#)

Bu örnekte, dosya bir Managed File Transfer Agent ' den bir Connect:Direct köprüsü düğümüne aktarılır. Aktarımın ilk bölümü Managed File Transfer tarafından gerçekleştirilir. Aktarımın ikinci bölümü, kullanıcı tarafından tanımlanan bir Connect:Direct işlemi tarafından gerçekleştirilir.

Yordam

1. İçsel simgesel değişkenleri kullanan bir Connect:Direct işlemi yaratın.

```
%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 bir metin dosyasına şu konumdaki bir dosyaya kaydedin: *MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent/Example.cdp*
3. Edit the *ConnectDirectProcessDefinition.xml* file to include a rule that calls the Connect:Direct process that you created in Step 1.

```
<?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 olarak TOBERMORY içeren Connect:Direct köprü aracısına bir aktarma isteği gönderilirse, *Example.cdp* Connect:Direct işlemi çağrılır.

4. Adım 3 'te *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 değerinin TOBERMORY değerine ayarlı olması gerekir. Bu komut, *ConnectDirectProcessDefinition.xml* dosyasında ayarlanan koşulla eşleşir.

5. Managed File Transfer , kaynak dosyayı, Connect:Direct köprü aracısıyla aynı sistemde bulunan geçici bir konuma aktarır.
6. Connect:Direct köprüsü aracı, aktarma isteği ve yapılandırma bilgilerindeki bilgilerden intrinsic simgesel değişkenlerin değerlerini ayarlar.
İç simgesel değişkenler, aşağıdaki değerlere ayarlanır:
 - %FTEPNAME=*process_name* -Bu değer, Connect:Direct köprü aracı tarafından oluşturulan 8 karakterlik bir süreç adıdır.
 - %FTESNODE=TOBERMORY-Bu değer, **fteCreateTransfer** komutunun **-df** değiştirgesinden ayarlanır.
 - %FTEPUSER, =*primary_node_user* -Bu bilgiler ConnectDirectCredentials.xml dosyasından alınır.
 - %FTEPPASS=*primary_node_user_password* -Bu bilgi, ConnectDirectCredentials.xml dosyasından alınır.
 - %FTESUSER, =*secondary_node_user* -Bu bilgi, ConnectDirectCredentials.xml dosyasından alınır.
 - %FTESPASSPAS=*secondary_node_user_password* -Bu bilgiler ConnectDirectCredentials.xml dosyasından alınır.
 - %FTEFFILE =*geçici_konumu* - This value is the temporary location of the file on the same system as the Connect:Direct bridge agent.
 - %FTEFDISP=leave-Bu değer, **fteCreateTransfer** komutunun **-sd** değiştirgesinden ayarlanır.
 - %FTETFILE=/home/bulgaria/destination.txt -Bu değer, **fteCreateTransfer** komutunun **-df** değiştirgesinden ayarlanır.
 - %FTETDISP=overwrite-Bu değer, **fteCreateTransfer** komutunun **-de** değiştirgesinden ayarlanır.
7. Connect:Direct işlemi, Connect:Direct köprüsü düğümünde başlatılır. Connect:Direct transfers the file from the temporary location on the Connect:Direct bridge system to the destination /home/bulgaria/destination.txt on the system where the Connect:Direct node TOBERMORY is running.

İlgili kavramlar

“Kullanıcı tanımlı bir Connect:Direct işleminin dosya aktarma isteğinden sunulması” sayfa 284
Connect:Direct köprü aracısından, kullanıcı tanımlı Connect:Direct işlemi, dosya aktarımın bir parçası olarak çağıran bir aktarım için aktarma isteği sunabilirsiniz.

İlgili başvurular

Kullanıcı tanımlı Connect:Direct süreçleriyle kullanım için yerine koyma değişkenleri

Managed File Transfer aktarma isteklerini sunmak için Connect:Direct işlemlerinin kullanılması

Bir Connect:Direct işleminden Connect:Direct köprü aracısına bir aktarma isteği sunabilirsiniz. Managed File Transfer , Connect:Direct sürecindeki bir **RUN TASK** deyiminden çağrılacak komutlar sağlar.

Managed File Transfer , Connect:Direct işlemleriyle kullanılmak üzere aşağıdaki komutları sağlar:

fteetiketi

Bu komutu, aktarıma ilişkin gerekli denetim bilgilerini yaratmak için **ftebxfer** ya da **ftecxfer** komutunun önünde yer alan bir adımda belirleyin. Bu komut, aktarmanın kaynak belirtimini parametre olarak alır. Kaynak belirtiminin biçimiyle ilgili bilgi için bkz. **fteCreateTransfer**: yeni bir dosya aktarımı başlatın.

ftebxfer

Aktarma isteğinin sunulduğu kuyruk yöneticisi, komutu gönderen Connect:Direct düğümüyle aynı sistemde yer alan bir dosya aktarma isteği yaratmak için bu komutu belirleyin. Bu komut, **fteCreateTransfer** komutuyla aynı parametreleri alır. Bu parametreler hakkında bilgi için bkz. **fteCreateTransfer**: yeni bir dosya aktarımı başlatın. Bu komutun ek bir parametresi de vardır:

-qmgrname

Gereklidir. Komutu teslim etmek için kuyruk yöneticisinin adı.

Ftecsxer

Aktarma isteğinin sunulduğu kuyruk yöneticisi, komutu sunan Connect:Direct düğümüne farklı bir sistemde yerleştirildiğinde, bir dosya aktarma isteği yaratmak için bu komutu belirtin. Bu komut, **fteCreateTransfer** komutuyla aynı parametreleri alır. Parametrelerle ilgili bilgi için bkz. **fteCreateTransfer**: yeni bir dosya aktarımı başlatın. Bu komut için üç ek parametre de vardır:

-qmgrname

Gereklidir. Komutu teslim etmek için kuyruk yöneticisinin adı.

-connname

Gereklidir. Komutun IBM MQ CONNAME biçiminde belirtildiği şekilde teslim olması için kuyruk yöneticisinin anasistemi ve kapısı. Örneğin, host.example.com(1337).

-channelname

İsteğe Bağlı. Komutun gönderileceği kuyruk yöneticisine bağlanmak için kullanılacak kanalın adı. Bu belirtilmediyse, varsayılan değer olarak SYSTEM.DEF.SVRCONN kullanılır.

İlgili görevler

[“Connect:Direct İstekçisi 'ni kullanarak Managed File Transfer ' i çağıran bir Connect:Direct işleminin yaratılması ve sunulması” sayfa 289](#)

Connect:Direct Requester, Managed File Transfer' i çağıran bir Connect:Direct işlemi yaratmak ve göndermek için kullanabileceğiniz bir grafik kullanıcı arabirimidir.

İlgili başvurular

Örnek: [MFT komutlarını çağıran bir Connect:Direct işlem dosyası](#)

Connect:Direct İstekçisi 'ni kullanarak Managed File Transfer ' i çağıran bir Connect:Direct işleminin yaratılması ve sunulması

Connect:Direct Requester, Managed File Transfer' i çağıran bir Connect:Direct işlemi 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 yaratılacağı ele alınmıştır. Aktarma isteğinin sunulduğu kuyruk yöneticisi, komutu sunan Connect:Direct düğümüne farklı bir sistemde bulunuyorsa **ftecxfer** komutunu kullanın. Aktarma isteğinin sunulduğu kuyruk yöneticisi, komutu gönderen Connect:Direct düğümüne aynı sistemde bulunuyorsa **ftebxfer** komutunu kullanın. **ftecxfer** komutu, aktarımın kaynak aracısının aracı kuyruk yöneticisine istemci bağlantısı yapar. **ftecxfer** komutunu çağırmadan önce, **ftetag** komutunu çağırmanız ve kaynak belirtim bilgilerini geçmeniz gerekir. Bu, işlemin Managed File Transfer' tan başlatılan aktarımlar ile aynı şekilde günlüğe kaydedilmesini ve denetlenmesini sağlar.

Yordam

1. Connect:Direct Requester 'ı başlatın.
2. Panonun **Nodes** (Düğümler) etiketinde, işlemin birincil düğümü olarak kullanılan Connect:Direct düğümünü seçin.
3. **Dosya > Yeni > İşlem** seçeneklerini belirleyin. **Süreç Özellikleri** penceresi açılır.
4. **Ad:** alanına sürecin adını yazın.
5. **Snode > Name:** (Snode-> Ad:) listesinden ikincil düğümü seçin.
6. **Snode > İşletim Sistemi:** listesinden ikincil düğüm için işletim sistemini seçin.
7. İsteğe bağlı: Bu pencerede gereksinim duyduğunuz diğer bilgileri tamamlayın.
8. **Tamam**'i tıklatın. **Süreç Özellikleri** penceresi kapanır.
9. Managed File Transfer **ftetag** komutunu çalıştıran bir deyim oluşturun.
 - a) **End** deyimindeki **Process** (İşlem) penceresini sağ tıklatın.

- b) **Ekle** > **Görevi Çalıştır** seçeneklerini belirleyin. **Görev Deyimi Çalıştır** penceresi açılır.
- c) **Label:** (Etiket:) alanına Tag yazın.
- d) **İsteğe Bağlı Değişirgeler ya da Komutlar** alanına `pgm(MQ_INSTALLATION_PATH/bin/ftetag) args(source_specification)` yazın. *source_specification* biçimi hakkında daha fazla bilgi için bkz. **ftCreateTransfer:** start a new file transfer.
- e) **Tamam**'ı tıklatın. **Run Task Deyimi** penceresi kapanır.
10. Managed File Transfer **ftecxfer** ya da **ftebxfer** komutunu çalıştıran bir deyim oluşturun.
- a) **End** deyimindeki **Process** (İşlem) penceresini sağ tıklatın.
- b) **Ekle** > **Görevi Çalıştır** seçeneklerini belirleyin. **Görev Deyimi Çalıştır** penceresi açılır.
- c) **Label:** (Etiket:) alanına Transfer yazın.
- d) In the **İsteğe bağlı değişirgeler ya da komutlar** field, type `pgm(MQ_INSTALLATION_PATH/bin/ftecxfer) args(parameters)` or `pgm(MQ_INSTALLATION_PATH/bin/ftebxfer) args(parameters)` depending on which command you choose. **ftecxfer** ve **ftebxfer** komutları tarafından kullanılan parametreler, **ftCreateTransfer** komutu tarafından kullanılan parametrelerle aynıdır, ayrıca **ftecxfer** ve **ftebxfer** 'e özgü bazı ek parametreler de aynıdır. Daha fazla bilgi için bkz. **ftCreateTransfer:** start a new file transfer ve “Managed File Transfer aktarma isteklerini sunmak için Connect:Direct işlemlerinin kullanılması” sayfa 288.
- e) **Tamam**'ı tıklatın. **Run Task Deyimi** penceresi kapanır.
11. İsteğe bağlı: Gereksinim duyduğunuz ek deyimleri oluşturun.
12. İşlemi sunun.
- a) **İşlem** penceresini sağ tıklatın.
- b) **Gönder** seçeneğini belirleyin. **Connect:Direct Attach** penceresi açılır.
- c) İşlemi çalıştırmak için kullanmak üzere kullanıcı adını ve parolayı girin.
- d) **Tamam**'ı tıklatın.

İlgili kavramlar

“Managed File Transfer aktarma isteklerini sunmak için Connect:Direct işlemlerinin kullanılması” sayfa 288

Bir Connect:Direct işleminden Connect:Direct köprü aracısına bir aktarma isteği sunabilirsiniz. Managed File Transfer , Connect:Direct sürecindeki bir **RUN TASK** deyiminden çağrılacak komutlar sağlar.

Working with MFT from IBM Integration Bus

FTEOutput ve FTEInput düğümlerini kullanarak IBM Integration Bus ' tan Managed File Transfer ile çalışabilirsiniz.

- Use the FTEInput node to transfer a file across the network using Managed File Transfer and then process that file as part of an Integration Bus flow.
- Bir Integration Bus akışı tarafından çıkışa ait bir dosyayı ağdaki başka bir yere aktarmak için FTEOutput düğümünü kullanın.

Aracı aracısına ya da aracı aracısından 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

Aracınız ya da kuyruk yöneticiniz herhangi bir nedenle kullanılamazsa, örneğin bir güç ya da ağ hatası nedeniyle, Managed File Transfer bu senaryolarda aşağıdaki gibi kurtarılır:

- Typically, if there is a problem while a file is being transferred, Managed File Transfer recovers and restarts that file transfer after the problem is repaired.

- Aracı ya da kuyruk yöneticisi kullanılamaz durumdayken, aktarılmakta ya da değiştirilmekte olan bir dosya silinirse ya da değiştirilirse, aktarma başarısız olur ve aktarma günlüğüne hatayla ilgili ayrıntılar sağlayan bir ileti alır.
- Bir aracı işlemi dosya aktarımı sırasında başarısız olursa, aracı yeniden başlattığınızda aktarma devam eder.
- Bir aracı, Agent kuyruk yöneticisiyle bağlantıyı kaybederse, aracı kuyruk yöneticisine yeniden bağlanmayı denerken bekler. Aracı, kuyruk yöneticisine başarıyla yeniden bağlandığında, yürürlükteki aktarma devam eder.
- Aracı herhangi bir nedenle durdurulduysa, bir aracıyla ilişkilendirilmiş kaynak izleme programları yoklamayı durdurur. Aracı kurtarıldığında, izleme programları da yeniden başlatılır ve kaynak yoklama işlemi sürdürülür.
- For a file transfer with a source disposition of `sil`, if a recovery occurs after all the data is sent from a source agent to a destination agent, the source file is unlocked before deletion. Bu kilit açma, kaynak dosyanın dosya silinmeden önce değiştirebileceği anlamına gelir. Bu nedenle, kaynak dosyayı silmek güvensiz 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 değiştirilmemiş olduğunu doğrulayın ve kaynak dosyayı el ile silin.

You can check the status of your transfers in the IBM MQ Explorer. Herhangi bir aktarma Stalledolarak görünüyorsa, durdurulan durum, aracıyla ya da aktarımda yer alan iki aracı arasındaki bir sorunu ifade ettiği için, düzeltici işlemi yapmanız gerekebilir.

İlgili görevler

[“Durdurulan aktarımların kurtarılması için bir zamanaşımı ayarlanması” sayfa 291](#)

Bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan, durdurulan dosya aktarımları için bir aktarım kurtarma zamanaşımı ayarlayabilirsiniz. Ayrıca, tek bir aktarım için aktarma kurtarma zaman aşımı da ayarlayabilirsiniz. Belirli bir süre için saniye cinsinden, bir kaynak aracının durmuş bir dosya aktarımını kurtarmaya çalıştığı ve aktarma zamanaşımına ulaştığında aktarma başarısız olduğunda aktarma başarısız olursa, saniye cinsinden belirli bir süre değerini ayarlamalıdır.

Durdurulan aktarımların kurtarılması için bir zamanaşımı ayarlanması

Bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan, durdurulan dosya aktarımları için bir aktarım kurtarma zamanaşımı ayarlayabilirsiniz. Ayrıca, tek bir aktarım için aktarma kurtarma zaman aşımı da ayarlayabilirsiniz. Belirli bir süre için saniye cinsinden, bir kaynak aracının durmuş bir dosya aktarımını kurtarmaya çalıştığı ve aktarma zamanaşımına ulaştığında aktarma başarısız olduğunda aktarma başarısız olursa, saniye cinsinden belirli bir süre değerini ayarlamalıdır.

Bu görev hakkında

IBM MQ 9.1' tan, aracının `agent.properties` dosyasına bir aktarma kurtarma zamanaşımı parametresi ekleyerek, bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan bir aktarım kurtarma zamanaşımı ayarlayabilirsiniz. Ayrıca, komut satırından tek bir aktarım için ya da IBM MQ Explorer ile ya da Apache Ant görevlerini kullanarak aktarma kurtarma zamanaşımı ayarlayabilirsiniz. `agent.properties` dosyasında bir aktarım kurtarma zamanaşımı değeri ayarlandıysa, tek bir aktarım için aktarım kurtarma zaman aşımını ayarlama ayarı `agent.properties` dosyasındaki değeri geçersiz kılar.

Aktarma kurtarma zamanaşımı için üç seçenek vardır:

- Aracı, başarılı bir şekilde tamamlanıncaya kadar, durdurulan aktarımı kurtarmayı denemeye devam eder. Bu, aktarma kurtarma zamanaşımı ayarlanmazsa, aracının varsayılan davranışı ile aynıdır.
- Aracı, kurtarma işlemi girildikten hemen sonra, aktarma işlemini başarısız olarak işaretler.
- Aracı, aktarımın başarısız olduğu şekilde işaretlenmeden önce, belirtilen süre için durdurulan aktarımı yeniden denemeyi sürdürür.

Dosya aktarma kurtarma zamanasını ayarının belirlenmesi isteğe bağlıdır. Ayarlamadıysanız, aktarımlar varsayılan davranışı izler. This is the same as the default behavior of a Managed File Transfer source agent before IBM MQ 9.1, where the agent keeps trying to recover a stalled transfer until it is successful.

İlgili kavramlar

“MFT kurtarma ve yeniden başlatma” sayfa 290

Aracınız ya da kuyruk yöneticiniz herhangi bir nedenle kullanılamazsa, örneğin bir güç ya da ağ hatası nedeniyle, Managed File Transfer bu senaryolarda aşağıdaki gibi kurtarılır:

Aktarma kurtarma zamanasını kavramları

Bir kaynak aracının duraklayan bir dosya aktarımı kurtarmaya çalıştığı süreyi saniye cinsinden ayarlayabilirsiniz. Aracı yeniden deneme aralığı için zamanasını değerine ulaştığında aktarma başarısız olursa, aktarma başarısız olur.

Kurtarma zamanasını önceliği

A transfer recovery timeout value for an individual transfer specified through the **fteCreateTransfer**, **fteCreateTemplate**, or **fteCreateMonitor** commands, or by using IBM MQ Explorer, or specified in the **fte:filespec** nested element, takes precedence over the value that is specified for the **transferRecoveryTimeout** parameter in the `agent.properties` file for the source agent.

Örneğin, **fteCreateTransfer** komutu **-rt** parametresi ve değer çifti olmadan başlatılırsa, kaynak aracı AGENT1 , kurtarma zamanasını davranışını belirlemek için bir **transferRecoveryTimeout** değeri için `agent.properties` dosyasını denetler:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

`agent.properties` dosyasındaki **transferRecoveryTimeout** parametresi ayarlanmadıysa ya da -1 olarak ayarlandıysa, aracı varsayılan davranışı izler ve başarılı oluncaya kadar aktarımı kurtarmaya çalışır.

Ancak, **fteCreateTransfer** komutu **-rt** parametresini içeriyorsa, bu parametrenin değeri `agent.properties` dosyasındaki değerden öncelikli olarak uygulanır ve aktarım için kurtarma zamanasını ayarı olarak kullanılır:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

Kurtarma zamanasını sayacı

Kurtarma zamanasını sayacı, aktarma durumu kurtarma durumuna girdiğinde başlar. Aktarım günlüğü iletili, aktarma durumunun kurtarılacak şekilde değiştirildiğini ve durumun değiştirildiği kaynak aracı saat süresine işaret etmek üzere `Log/agent_name/transfer_ID` konu dizisiyle SYSTEM.FTE konusuna yayınlanır. Aktarma yeniden deneme aralığı içinde sürdürülürse ve kurtarma zamanasını (sayacı < =kurtarma zamanasını) ulaşmazsa, sayaç 0 olarak sıfırlanır, aktarım kurtarma işlemi girerse yeniden başlamaya hazır olur.

Sayaç, kurtarma zamanasını (sayaç == kurtarma zamanasını) için ayarlanan üst sınır değerine ulaşırsa, aktarma durakları ve kaynak aracı, aktarımı başarısız olarak bildirir. Bu aktarma hatası tipi, aktarma işlemi için kurtarma zamanasını değerine ulaşılması nedeniyle, KURTARMA ZAMAN AŞIIMI (69ileti kodundan kaynaklanır. Başka bir aktarım günlüğü iletili SYSTEM.FTE konusu, aktarımın başarısız olduğunu ve bir iletiyi, dönüş kodunu ve kaynak aracının olay günlüğünü içerdiğini belirtmek için `Log/agent_name/transfer_ID` konu dizisiyle birlikte. Kaynak aracının olay günlüğü, kurtarma sırasında aşağıdaki olaylardan herhangi biri gerçekleştiğinde bir iletiyle güncellenir:

- Kurtarma zamanasını parametresi -1 değerinden daha büyük bir değere ayarlandığında, aktarma işlemi kurtarma işlemi girer. Aracının olay günlüğü güncellenir ve **TransferId** için kurtarma süreölçerinin başlangıcını ve kaynak aracının kurtarma zamanasını işlemi başlatmadan önce bekleyeceği süreyi belirtir.

- Kurtarma işlemi sürdürüldüğünde, kaynak aracının olay günlüğü, kurtarma işlemi olan **TransferId** ' in sürdürülmesi gerektiğini belirtmek için yeni bir iletiyle güncellenir.
- Kurtarma işlemi zamanaşımına uğradığında, kaynak aracının olay günlüğü, kurtarma zamanaşımı nedeniyle kurtarma işlemi sırasında başarısız olan **TransferId** ' i belirtmek üzere güncellenir.

Bu günlük iletileri, kullanıcılar (aboneler ve kaydediciler), aktarma kurtarma zamanaşımı nedeniyle başarısız olan aktarımları tanımlamalarını sağlar.

Kurtarma zamanaşımı sayacı her zaman kaynak aracıda yer alıyor. Ancak, hedef aracı kaynak aracıdan zamanında bilgi alamazsa, aktarma işlemi gerçekleştirilmesi için kaynak aracıya bir istek gönderebilir. Kurtarma zamanaşımı seçeneğinin belirlendiği bir aktarma için, kaynak aracı, hedef aracıdan gelen isteği aldığı anda kurtarma zamanaşımı sayacı başlatır.

Kurtarma zamanaşımı seçeneğini, başarısız olan ve kısmen tam olarak aktarımları kullanmayan aktarımlar için el ile işleme gerekli olmaya devam eder.

Birden çok dosya için tek bir aktarma isteğinin yayınlandığı ve dosyaların bir kısmının başarıyla tamamlandığı, ancak yalnızca kısmen tamamlanan aktarma kümeleri için, aktarma işlemi beklendiği gibi tamamlanmadığı için hala başarısız olarak işaretlenir. Kaynak aracı, kısmen tamamlanmış olan dosyayı aktarırken zamanaşımına uğramış olabilir.

Hedef aracı ve dosya sunucusunun, dosya aktarımlarını kabul etmek için hazır ve bir durumda olduğundan emin olun.

Tüm küme için aktarma isteğini yeniden yayınlamamanız gerekir; ancak, bazı dosyalar ilk aktarma girişiminden hedefte kaldığı için sorunları önlemek için, yeni isteği Varsa üzerine yaz seçeneğiyle yayınlatabilirsiniz. Bu, kütüklerin hedefe yeniden yazılmadan önce, önceki aktarma girişimindeki eksik kütük kümesinin yeni aktarımın bir parçası olarak temizlenmesini sağlar.

V 9.2.0 IBM MQ 9.1.5' tan, ilk aktarma girişimi başarısız olduktan sonra, bir hedefte bırakılan parça dosyalarını el ile kaldırmak artık gerekli değildir. Aktarma için bir aktarma kurtarma zamanaşımı ayarlandıysa, aktarma kurtarma zamanaşımına uğradığında kaynak aracı, aktarımı RecoveryTimedOut durumuna taşır. Aktarma yeniden uyumlulaştırıldıktan sonra, hedef aracı, aktarma sırasında yaratılan tüm parça dosyalarını kaldırır ve kaynak aracıya bir tamamlanma iletisi gönderir.

İzlemeler ve iletiler

İzleme noktaları tanılama amacıyla içerilir. Kurtarma zamanaşımı değeri, yeniden deneme aralığı başlangıcı, sürdürme süresi ve sayacın ilk durumuna getirilmesi ve aktarmanın zamanaşımına uğrayıp zamanaşımına uğramadığı, günlüğe kaydedilip kaydedilmediği. Bir sorun ya da beklenmeyen bir davranışta, kaynak aracı çıkış günlüğünü ve izleme dosyalarını toplayabilir ve sorun gidermeye yardımcı olmak için IBM desteği tarafından istenildiğinde bunları sağlayabilirsiniz.

İletiler aşağıdaki durumlarda bilgilendirilecektir:

- Aktarım sırasında kurtarma işlemi başlar ([BFGTR0081I](#))
- Bir aktarım işlemi, kurtarma işlemi sırasında zamanaşımına uğradığı için sonlandırıldı ([BFGSS0081E](#))
- Kurtarma işlemi tamamlandıktan sonra Atransfer devam eder ([BFGTR0082I](#))

İlgili kavramlar

“MFT kurtarma ve yeniden başlatma” sayfa 290

Aracınız ya da kuyruk yöneticiniz herhangi bir nedenle kullanılamazsa, örneğin bir güç ya da ağ hatası nedeniyle, Managed File Transfer bu senaryolarda aşağıdaki gibi kurtarılır:

Bir kaynak aracıya ilişkin tüm aktarımlar için aktarım kurtarma zamanaşımını ayarlama

You can set a transfer recovery timeout that applies to all the transfers for a source agent by adding the **transferRecoveryTimeout** parameter to the `agent.properties` file.

Bu görev hakkında

Bir kaynak aracıya ilişkin tüm aktarımlar için geçerli olan bir ağaç sörfü kurtarma zaman aşımı ayarlamak için, **transferRecoveryTimeout** için parametre ve değer çiftini [agent.properties](#) dosyasına ekliyorsunuz.

transferRecoveryTimeout parametresi için üç seçenek vardır:

-1

Aracı, aktarma işlemi başarılı oluncaya kadar, durdurulan aktarımı kurtarmayı denemeye devam eder. Bu seçeneği kullanmak, özellik ayarlanmadığında aracının varsayılan davranışına eşdeğerdir.

0

Aracı, kurtarma işlemini girer girmez dosya aktarımlarını durdurur.

>0

Aracı, belirtilen pozitif tamsayı değeri tarafından belirlenen süre saniye cinsinden, durdurulan aktarım işlemini kurtarma girişiminde bulunmaya devam eder.

`agent.properties` dosyasında yaptığınız değişiklikler, yalnızca aracı yeniden başlatıldıktan sonra yürürlüğe girmektedir.

Gerekirse, tek bir aktarım için `agent.properties` dosyasındaki aktarma kurtarma zaman aşımı değerini geçersiz kılabilirsiniz. Daha fazla bilgi için bkz [“Tek tek aktarımlar için aktarım kurtarma zaman aşımını ayarlama”](#) sayfa 294.

Yordam

- Aracının, durdurulan aktarımı, başarıyla tamamlanmadıkça kurtarmayı denemeye devam ettiğini belirtmek için, aşağıdaki örnekte gösterildiği gibi, aktarım kurtarma zaman aşımını değerini **-1** olarak ayarlayın:

```
transferRecoveryTimeout=-1
```

- Aracının kurtarma işlemi sırasında hemen başarısız olarak aktarıldığını belirtmek için, aşağıdaki örnekte gösterildiği gibi, aktarma kurtarma zaman aşımını değeri **0** değerini belirtin:

```
transferRecoveryTimeout=0
```

- Aracının, aktarma işlemi başarısız olarak işaretlenmeden önce, durdurulan bir aktarımı yeniden denemeye devam ettiğini belirtmek için, yeniden denemeye devam etmek üzere aracıya vermek istediğiniz süre için, saniye cinsinden bir aktarma kurtarma zaman aşımını değeri ayarlayın.

Örneğin, bir aktarım kurtarma zaman aşımını değerinin 21600 ayarlanması, aracının kurtarma işlemini gireceği zamandan itibaren altı saat boyunca kurtarmayı denediği anlamına gelir:

```
transferRecoveryTimeout=21600
```

Bu parametre için maksimum değer 999999999' dir.

Tek tek aktarımlar için aktarım kurtarma zaman aşımını ayarlama

Komut satırından tek bir aktarım için ya da IBM MQ Explorer ile ya da Apache Ant görevlerini kullanarak aktarma kurtarma zaman aşımını ayarlayabilirsiniz. `agent.properties` dosyasında bir aktarım kurtarma zaman aşımını değeri ayarlandıysa, tek bir aktarım için aktarım kurtarma zaman aşımını ayarlayarak `agent.properties` dosyasında ayarlanan değeri geçersiz kılar.

Bu görev hakkında

Ayrı ayrı aktarma için aktarma kurtarma zaman aşımını parametresini aşağıdaki durumlarda belirleyebilirsiniz:

- Creating a transfer either by using the **fteCreateTransfer** command, or by using IBM MQ Explorer.

- Creating a transfer template either by using the **fteCreateTemplate** command, or by using IBM MQ Explorer.
- Creating a resource monitor either by using the **fteCreateMonitor** command, or by using IBM MQ Explorer.
- Dosyaları kopyalamak ya da taşımak için [fte: filecopy](#) ya da [fte: filemove](#) Ant görevlerini kullanın.

Tek bir aktarım için bir aktarma kurtarma zamanaşımı değeri ayarladıysanız, bu değer `agent.properties` dosyasında ayarlanan aktarım kurtarma zamanaşımı değerini geçersiz kılar (bkz. "[Bir kaynak aracıya ilişkin tüm aktarımlar için aktarım kurtarma zamanaşımını ayarlama](#)" sayfa 293).

Yordam

- Aktarım kurtarma zamanaşımını ayarlamak üzere **fteCreateTransfer** ya da **fteCreateTemplate** komutunu kullanmak için, **-rt** parametresi için uygun seçeneği belirtin:

-1

Aracı, aktarma işlemi başarılı oluncaya kadar, durdurulan aktarımı kurtarmayı denemeye devam eder. Bu seçeneği kullanmak, özellik ayarlanmadığında aracının varsayılan davranışına eşdeğerdir.

0

Aracı, kurtarma işlemi girer girmez dosya aktarımlarını durdurur.

>0

Aracı, saniye cinsinden belirtilen süre için durdurulan aktarımı kurtarmayı denemeye devam eder.

fteCreateTransfer komutuna ilişkin örnekler

```
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 komutuna ilişkin örnekler

```
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. If you set the **-rt** parameter with the **fteCreateTransfer** command and also set the **-gt** parameter, the recovery timeout parameter is included in the XML document with the transfer definition that is generated when you run the **fteCreateTransfer** command. Bundan sonra, kaynak izleme programı bu XML belgesini kullanarak **fteCreateMonitor** komutunu çalıştırırken kullanır. Aşağıdaki örnekte, aktarma kurtarma zamanaşımı ayrıntıları `task.xml` dosyasına eklenecektir:

```
fteCreateMonitor -ma AgentName -md C:\mqmft\monitors -mn Monitor_Name -mt task.xml -tr
"fileSize>=5MB,*.zip"
```

- Aktarma kurtarma zamanaşımını ayarlamak için IBM MQ Explorer New Transfer, New Monitor ya da New Template sihirbazı sayfasını kullanmak için **Transfer Recovery Timeout** (saniye) alanında gerekli seçeneği belirleyin:

Kaynak Aracı Olarak

If you select **Kaynak Aracı Olarak**, the **transferRecoveryTimeout** parameter value from the `agent.properties` file is used if it is set, otherwise the default behavior for transfer recovery timeout is applied.

Sayısal liste kutusu

Sayısal liste kutusuna saniye cinsinden bir saat girerseniz, aracı belirtilen süre için durdurulan aktarımı kurtarmayı denemeye devam eder.

Yok

Yok seçeneğini belirlerseniz, aktarma kurtarma zamanaşımı değeri belirlenmez ve aracı, aktarma başarılı oluncaya kadar durdurulan aktarımı kurtarma girişiminde bulunmaya devam eder.

- Kurtarma zamanaşımını Ant görevlerini kullanarak ayarlamak için. include the **transferRecoveryTimeout** option and value, with the **fte:filecopy** or **fte:filemove** elements for moving or copying files, for example:

fte:filecopy için örnek

```
<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>
```

fte:filemove için örnek

```
<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>
```

Linux

Windows

AIX

YönetmeMQ Telemetry

MQ Telemetry , IBM MQ Explorer ya da bir komut satırı kullanılarak yönetilir. Telemetri kanallarını yapılandırmak, telemetri hizmetini denetlemek ve IBM MQ' a bağlı MQTT istemcilerini izlemek için kaşifi kullanın. Configure the security of MQ Telemetry using JAAS, TLS and the IBM MQ object authority manager.

IBM MQ Explorer komutunu kullanarak yönetme

Telemetri kanallarını yapılandırmak, telemetri hizmetini denetlemek ve IBM MQ' a bağlı MQTT istemcilerini izlemek için kaşifi kullanın. Configure the security of MQ Telemetry using JAAS, TLS and the IBM MQ object authority manager.

Komut satırını kullanarak yönetme

MQ Telemetry , [MQSC komutlarını kullanarak](#) komut satırında tam olarak denetlenebilir.

MQ Telemetry belgelerinde, IBM MQ Telemetry Transport v3 Client uygulamasının temel kullanımını gösteren örnek komut dosyaları da vardır.

Örnekleri kullanmadan önce [IBM MQ Telemetry Transport örnek programları](#) içindeki örnekleri okuyun ve anlayın.

İlgili kavramlar

[MQ Telemetry](#)

İlgili başvurular

[MQXR özellikleri](#)

Yapılandırılması

MQ Telemetry 'u el ile yapılandırmak için bu adımları izleyin. Yalnızca konuk kullanıcı kimliğini kullanan basit bir yapılandırmaya gereksinim duyarsanız, bunun yerine IBM MQ Explorer' ta MQ Telemetry destek sihirbazını çalıştırabilirsiniz.

Başlamadan önce

Yalnızca basit bir yapılandırmaya gereksinim duyarsanız, IBM MQ Explorer içinde MQ Telemetry desteğini kullanmayı düşünün. Bu destek, bir sihirbazı ve örnek komut yordamını `sampleMQM` içerir. Bu kaynaklar, konuk kullanıcı kimliğini kullanarak ilk yapılandırmayı ayarlar. Bkz. [Verifying the installation of MQ Telemetry by using IBM MQ Explorer](#) ve [IBM MQ Telemetry Transport örnek programları](#).

Farklı bir kimlik doğrulama yöntemi kullanan daha karmaşık bir yapılandırmaya gereksininiz varsa, bu görevdeki adımları kullanın. Aşağıdaki başlangıç adımlarıyla başlayın:

1. IBM MQ ve MQ Telemetry özelliğinin nasıl kurulabilmesiyle ilgili bilgi için bkz. [MQ Telemetry için kuruluşa ilgili dikkat edilmesi gereken noktalar](#).
2. Kuyruk yöneticisi yarat ve başlat. Kuyruk yöneticisi bu görevde `qMgr` olarak anılır.
3. Bu görevin bir parçası olarak, telemetri (MQXR) hizmetini yapılandırıyorsunuz. MQXR özellik ayarları, platforma özgü özellikler dosyasında depolanır: `mqxr_win.properties`. Hemen hemen tüm ayarların MQSC denetim komutları ya da IBM MQ Explorer yoluyla yapılandırılabilceğinden, MQXR özellikler dosyasını doğrudan düzenlemenize gerek yoktur. Dosyayı doğrudan düzenlemeye karar verdiyseniz, değişikliklerinizi yapmadan önce kuyruk yöneticisini durdurun. Bkz. [MQXR özellikleri](#).

Bu görev hakkında

Follow the steps in this task to configure MQ Telemetry manually using different authorization schemes.

Yordam

1. Telemetri numuneleri 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 kez başlatıldığında otomatik olarak yaratılır ve konuk kullanıcı kimliğini kullanmak için ayarlanır. However this task configures MQ Telemetry to use a different authorization scheme. Bu görev için, telemetri (MQXR) hizmetini başlatmadan önce SYSTEM.MQTT.TRANSMIT.QUEUE' u yaratır ve ona erişimi yapılandırın.

Aşağıdaki komutu çalıştırın:

```
echo "DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000)" | runmqsc qMgr
```

3. Varsayılan iletim kuyruğunu belirleyin.

SYSTEM.MQTT.TRANSMIT.QUEUE varsayılan iletim kuyruğıyse, iletileri doğrudan MQTT istemcilerine göndermek daha kolay olur. Ters durumda, IBM MQ iletilerini alan her istemci için bir uzak kuyruk tanımlaması eklemeniz gerekir; bkz. [“İstemciye doğrudan ileti gönderme” sayfa 302](#). Varsayılan iletim kuyruğunun değiştirilmesiyle var olan yapılanışınıza müdahale edebilirsiniz.

Telemetri (MQXR) hizmeti ilk kez başlatıldığında, kuyruk yöneticisi için varsayılan iletim kuyruğı olarak SYSTEM.MQTT.TRANSMIT.QUEUE değerini belirlemez. Bu ayarı yapılandırmak için, varsayılan iletim kuyruğı özelliğini değiştirirdiniz. Bunu, IBM MQ Explorer komutunu kullanarak ya da aşağıdaki komutu çalıştırarak yapın:

```
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 verme” sayfa 304 içindeki bir yordamı izleyin. Kullanıcı kimlikleri, yayınları yayınlama, abone olma ve MQTT istemcilerine gönderme yetkisine sahiptir.
5. Telemetri (MQXR) hizmetini kurun.

```
cat /opt/<install_dir>/mqxr/samples/installMQXRService_unix.mqsc | runmqsc qMgr
```

Ayrıca, “SYSTEM.MQXR.SERVICEoluşturma” sayfa 298içindeki örnek kodla da bakın.

6. Hizmeti başlatın.

```
echo "START SERVICE(SYSTEM.MQXR.SERVICE) " | runmqsc qMgr
```

Kuyruk yöneticisi başlatıldığında telemetri (MQXR) hizmeti otomatik olarak başlatılır. Kuyruk yöneticisi zaten çalışır durumda olduğundan, bu görev bu görevde el ile başlatılır.

7. IBM MQ Explorerolanağını kullanarak, telemetri kanallarını MQTT istemcilerinden gelen bağlantıları kabul edecek şekilde yapılandırın.

Telemetri kanalları, kimliklerinin “4” sayfa 298adımında tanımlanan kullanıcı kimliklerinden biri olması gibi yapılandırılmalıdır.

Ayrıca bkz. DEFINE CHANNEL (MQTT).

8. Örnek istemciyi çalıştırarak yapılandırmayı doğrulayın.

telemetri kanalıyla çalışabilmek için örnek müşteri için kanal, müşteriye yayınları yayınlama, abone olma ve alma yetkisi vermelidir. Örnek istemci, varsayılan olarak 1883 numaralı bağlantı noktasındaki telemetri kanalına bağlanır. Ayrıca bkz. [IBM MQ Telemetry Transport örnek programları](#).

SYSTEM.MQXR.SERVICEoluşturma

SYSTEM.MQXR.SERVICEkomutunu oluşturmak için **runMQXRService** komutunu kullanın.

LTS

```
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+"') +
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')
```

V 9.2.4

```
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 9.2.4** IBM MQ 9.2.4 ' dan itibaren, **-sf** işareti, TLS kanalı geçiş tümcelerinden şifreleme için kullanılacak anahtarları içeren kimlik bilgileri anahtar dosyası adını sağlar. Daha fazla bilgi için bakınız: [Encryption of passpsases for MQTT TLS kanals.](#)

MQ Telemetry 'u el ile yapılandırmak için bu adımları izleyin. Yalnızca konuk kullanıcı kimliğini kullanan basit bir yapılandırmaya gereksinim duyarsanız, bunun yerine IBM MQ Explorer' ta MQ Telemetry destek sihirbazını çalıştırabilirsiniz.

Başlamadan önce

Yalnızca basit bir yapılandırmaya gereksinim duyarsanız, IBM MQ Explorer içinde MQ Telemetry desteğini kullanmayı düşünün. Bu destek, bir sihirbazı ve örnek komut yordamını `sampleMQM` içerir. Bu kaynaklar, konuk kullanıcı kimliğini kullanarak ilk yapılandırmayı ayarlar. Bkz. [Verifying the installation of MQ Telemetry by using IBM MQ Explorer](#) ve [IBM MQ Telemetry Transport örnek programları](#).

Farklı bir kimlik doğrulama yöntemi kullanan daha karmaşık bir yapılandırmaya gereksininiz varsa, bu görevdeki adımları kullanın. Aşağıdaki başlangıç adımlarıyla başlayın:

1. IBM MQ ve MQ Telemetry özelliğinin nasıl kurulabilmesiyle ilgili bilgi için bkz. [MQ Telemetry için kuruluşla ilgili dikkat edilmesi gereken noktalar](#).
2. Kuyruk yöneticisi yarat ve başlat. Kuyruk yöneticisi bu görevde `qMgr` olarak anılır.
3. Bu görevin bir parçası olarak, telemetri (MQXR) hizmetini yapılandırıyorsunuz. MQXR özellik ayarları, platforma özgü özellikler dosyasında depolanır: `mqxr_win.properties`. Hemen hemen tüm ayarların MQSC denetim komutları ya da IBM MQ Explorer yoluyla yapılandırılabilceğinden, MQXR özellikler dosyasını doğrudan düzenlemenize gerek yoktur. Dosyayı doğrudan düzenlemeye karar verdiyseniz, değişikliklerinizi yapmadan önce kuyruk yöneticisini durdurun. Bkz. [MQXR özellikleri](#).

Bu görev hakkında

Follow the steps in this task to configure MQ Telemetry manually using different authorization schemes.

Yordam

1. Telemetri numuneleri 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 kez başlatıldığında otomatik olarak yaratılır ve konuk kullanıcı kimliğini kullanmak için ayarlanır. However this task configures MQ Telemetry to use a different authorization scheme. Bu görev için, telemetri (MQXR) hizmetini başlatmadan önce SYSTEM.MQTT.TRANSMIT.QUEUE' u yaratır ve ona erişimi yapılandırın.

Aşağıdaki komutu çalıştırın:

```
echo DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000) | runmqsc qMgr
```

3. Varsayılan iletim kuyruğunu belirleyin.

SYSTEM.MQTT.TRANSMIT.QUEUE varsayılan iletim kuyruğıyse, iletileri doğrudan MQTT istemcilerine göndermek daha kolay olur. Ters durumda, IBM MQ iletilerini alan her istemci için bir uzak kuyruk tanımlaması eklemeniz gerekir; bkz. [“İstemciye doğrudan ileti gönderme” sayfa 302](#). Varsayılan iletim kuyruğunun değiştirilmesiyle var olan yapılanışınıza müdahale edebilirsiniz.

Telemetri (MQXR) hizmeti ilk kez başlatıldığında, kuyruk yöneticisi için varsayılan iletim kuyruğı olarak SYSTEM.MQTT.TRANSMIT.QUEUE değerini belirlemez. Bu ayarı yapılandırmak için, varsayılan iletim kuyruğı özelliğini değiştirirdiniz. Bunu, IBM MQ Explorer komutunu kullanarak ya da aşağıdaki komutu çalıştırarak yapın:

```
echo ALTER QMGR DEFQXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```

4. Bir ya da daha çok kullanıcı kimliği yaratmak için “MQTT istemcilerinin IBM MQ nesnelere erişmesi için yetki verme” sayfa 304 içindeki bir yordamı izleyin. Kullanıcı kimlikleri, yayınları yayınlama, abone olma ve MQTT istemcilerine gönderme yetkisine sahiptir.
5. Telemetri (MQXR) hizmetini kurun.

```
type installMQXRService_win.mqsc | runmqsc qMgr
```

Ayrıca, “Oluşturma SYSTEM.MQXR.SERVICE” sayfa 300 içindeki örnek kodla da bakın.

6. Hizmeti başlatın.

```
echo START SERVICE(SYSTEM.MQXR.SERVICE) | runmqsc qMgr
```

Kuyruk yöneticisi başlatıldığında telemetri (MQXR) hizmeti otomatik olarak başlatılır. Kuyruk yöneticisi zaten çalışır durumda olduğundan, bu görev bu görevde el ile başlatılır.

7. IBM MQ Explorerolanağını kullanarak, telemetri kanallarını MQTT istemcilerinden gelen bağlantıları kabul edecek şekilde yapılandırın.

Telemetri kanalları, kimliklerinin “4” sayfa 300adımında tanımlanan kullanıcı kimliklerinden biri olması gibi yapılandırılmalıdır.

Ayrıca bkz. [DEFINE CHANNEL \(MQTT\)](#).

8. Örnek istemciyi çalıştırarak yapılandırmayı doğrulayın.

telemetri kanalıyla çalışabilmek için örnek müşteri için kanal, müşteriye yayınları yayınlama, abone olma ve alma yetkisi vermelidir. Örnek istemci, varsayılan olarak 1883 numaralı bağlantı noktasındaki telemetri kanalına bağlanır. Ayrıca bkz. [IBM MQ Telemetry Transport örnek programları](#).

Oluşturma SYSTEM.MQXR.SERVICE

SYSTEM.MQXR.SERVICEkomutunu oluşturmak için **runMQXRService** komutunu kullanın.

V9.2.4

```
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: V9.2.4 IBM MQ 9.2.4 ' dan itibaren, **-sf** işareti, TLS kanalı geçiş tümcelerinden şifreleme için kullanılacak anahtarları içeren kimlik bilgileri anahtar dosyası adını sağlar. Daha fazla bilgi için bakınız: [Encryption of passpsases for MQTT TLS kanals.](#)

Linux

Windows

AIX

MQTT istemcilerine ileti göndermek için dağıtımli kuyruğa alma yapılandırılması

IBM MQ uygulamaları, MQTT v3 istemcileri iletilerini bir istemci tarafından oluşturulan aboneliğe yayınlamak ya da doğrudan bir ileti göndererek gönderebilir. Hangi yöntem kullanılırsa kullanılırsa, ileti SYSTEM.MQTT.TRANSMIT.QUEUE' e yerleştirilir ve telemetri (MQXR) hizmeti tarafından istemciye gönderilir. SYSTEM.MQTT.TRANSMIT.QUEUE' ta bir ileti yerleştirmenin çeşitli yolları vardır.

Publishing a message in response to an MQTT client subscription

Telemetri (MQXR) hizmeti, MQTT istemcisi adına bir abonelik oluşturur. İstemci, istemci tarafından gönderilen aboneliğe eşleşen yayınların hedefidir. Telemetri hizmetleri, eşleşen yayınları istemciye geri iletir.

Bir MQTT istemcisi, kuyruk yöneticisi adı *ClientIdentifier* olarak ayarlanmış bir kuyruk yöneticisi olarak IBM MQ 'ye bağlanır. İstemciye gönderilecek yayınların hedefi bir iletim kuyruğudur (*SYSTEM.MQTT.TRANSMIT.QUEUE*). The telemetry service forwards messages on *SYSTEM.MQTT.TRANSMIT.QUEUE* to MQTT clients, using the target queue manager name as the key to a specific client.

Telemetri (MQXR) hizmeti, kuyruk yöneticisi adı olarak *ClientIdentifier* komutunu kullanarak iletim kuyruğunu açar. Telemetri (MQXR) hizmeti, istemci aboneliğiyle eşleşen yayınları iletmek için, kuyruğun nesne tanıtıcısını MQSUB çağrısına iletir. In the object name resolution, the *ClientIdentifier* is created as the remote queue manager name, and the transmission queue must resolve to *SYSTEM.MQTT.TRANSMIT.QUEUE*. Standart IBM MQ nesne adı çözünürlüğünü kullanarak, *ClientIdentifier* şu şekilde çözümlenir; bkz. Çizelge 16 sayfa 301.

1. *ClientIdentifier* hiçbir şey ile eşleşmiyor.

ClientIdentifier, uzak bir kuyruk yöneticisi adıdır. Yerel kuyruk yöneticisi adı, kuyruk yöneticisi diğer adı ya da iletim kuyruğu adı ile eşleşmiyor.

Kuyruk adı tanımlı değil. Şu anda, telemetri (MQXR) hizmeti, kuyruğun adı olarak *SYSTEM.MQTT.PUBLICATION.QUEUE* 'yi ayarlar. Bir MQTT v3 istemcisi kuyrukları desteklemez, bu nedenle, çözümlenen kuyruk adı istemci tarafından yoksayılr.

Yerel kuyruk yöneticisi özelliği, Varsayılan iletim kuyruğu, ad *SYSTEM.MQTT.TRANSMIT.QUEUE* olarak ayarlanmalıdır; böylece, yayın, istemciye gönderilecek *SYSTEM.MQTT.TRANSMIT.QUEUE* 'a konmalıdır.

2. *ClientIdentifier*, *ClientIdentifier* adlı bir kuyruk yöneticisi diğer adıyla eşleşiyor.

ClientIdentifier, uzak bir kuyruk yöneticisi adıdır. Kuyruk yöneticisi diğer adı adıyla eşleşir.

Kuyruk yöneticisi diğer adı, uzak kuyruk yöneticisi adı olarak *ClientIdentifier* ile tanımlanmalıdır.

Kuyruk yöneticisi diğer adı tanımlamasındaki iletim kuyruğu adını ayarlayarak, varsayılan iletim için *SYSTEM.MQTT.TRANSMIT.QUEUE* değere ayarlanmalıdır.

Çizelge 16. Bir MQTT kuyruk yöneticisi diğer adının çözülmesi					
<i>ClientIdentifier</i>	Giriş		Çıkış		
	Kuyruk yöneticisi adı	Kuyruk adı	Kuyruk yöneticisi adı	Kuyruk adı	İletim kuyruğu
Hiçbir şey ile	<i>ClientIdentifier</i>	<i>tanımsız</i>	<i>ClientIdentifier</i>	<i>tanımsız</i>	Varsayılan iletim kuyruğu. <i>SYSTEM.MQTT.TRANSMIT.QUEUE</i>
<i>ClientIdentifier</i> adlı bir kuyruk yöneticisi diğer adıyla eşleşiyor	<i>ClientIdentifier</i>	<i>tanımsız</i>	<i>ClientIdentifier</i>	<i>tanımsız</i>	<i>SYSTEM.MQTT.TRANSMIT.QUEUE</i>

Ad çözümlenmesiyle ilgili daha fazla bilgi için [Ad çözünürlüğü](#) başlıklı konuya bakın.

Herhangi bir IBM MQ programı aynı konuya yayınlanabilir. Yayın, konuya aboneliği olan MQTT v3 istemcileri de dahil olmak üzere abonelerine gönderilir.

Bir yönetim konusu bir kümede yaratılırsa, *CLUSTER(clusterName)* özniteliği ile, kümedeki herhangi bir uygulama istemciye yayınlanabilir; örneğin:

```
echo DEFINE TOPIC('MQTTExamples') TOPICSTR('MQTT Examples') CLUSTER(MQTT) REPLACE | runmqsc qMgr
```

Şekil 16. Windows üzerinde bir küme başlığı tanımlama

Not: SYSTEM.MQTT.TRANSMIT.QUEUE bir küme özniteliği vermemek için.

MQTT istemci aboneleri ve yayıncıları farklı kuyruk yöneticilerine bağlanabilirler. Aboneler ve yayıncılar aynı kümenin bir parçası olabilir ya da bir yayınlama/abone olma hiyerarşisi ile bağlantıda olabilir. The publication is delivered from the publisher to the subscriber using IBM MQ.

İstemciye doğrudan ileti gönderme

Abonelik yaratan ve abonelik konularıyla eşleşen bir yayını alan bir istemciye alternatif olan bir alternatif, doğrudan bir MQTT v3 istemcisine bir ileti gönderir. MQTT V3 istemci uygulamaları doğrudan ileti gönderemez; ancak, IBM MQ uygulamaları gibi başka bir uygulama da gönderilebilir.

IBM MQ uygulaması, MQTT v3 istemcisinin `ClientIdentifier` 'ı (`ClientIdentifierClientIdentifierClientIdentifier`)knowbiliyor olmalıdır. As MQTT v3 clients do not have queues, the target queue name is passed to the MQTT v3 application client `messageArrived` method as a topic name. Örneğin, bir MQI programında, istemciyle `ObjectQmgrAdı` olarak bir nesne tanımlayıcısı yaratın:

```
MQOD.ObjectQmgrName = ClientIdentifier ;  
MQOD.ObjectName = name ;
```

Şekil 17. Bir MQTT v3 istemci hedefine ileti göndermek için MQI Nesnesi tanımlayıcısı

Uygulama JMS kullanılarak yazıldıysa, noktadan noktaya iletişim hedefi yaratın; örneğin:

```
javax.jms.Destination jmsDestination =  
(javax.jms.Destination)jmsFactory.createQueue  
("queue://ClientIdentifier/name");
```

Şekil 18. MQTT v3 istemcisine ileti göndermek için JMS hedefi

Bir MQTT istemcisine istekte bulunulmayan bir ileti göndermek için uzak kuyruk tanımlamasını kullanın. Uzak kuyruk yöneticisi adı, istemcinin `ClientIdentifier` 'ine çözümlenmelidir. İletim kuyruğu `SYSTEM.MQTT.TRANSMIT.QUEUE` 'e çözümlenmelidir; bkz. Çizelge 17 sayfa 302. Uzak kuyruk adı herhangi bir değer olabilir. İstemci bunu bir konu dizgisi olarak alır.

Çizelge 17. Bir MQTT istemcisi uzak kuyruk tanımlamasının ad çözümlemesi

Giriş		Çıkış		
Kuyruk adı	Kuyruk yöneticisi adı	Kuyruk adı	Kuyruk yöneticisi adı	İletim kuyruğu
Uzak kuyruk tanımlamasının adı	Boş ya da yerel kuyruk yöneticisi adı	Bir konu dizgisi olarak kullanılan uzak kuyruk adı	<code>ClientIdentifier</code>	<code>SYSTEM.MQTT.TRANSMIT.QUEUE</code>

If the client is connected, the message is sent directly to the MQTT client, which calls the `messageArrived` method; see `messageArrived` yöntemi.

İstemcinin kalıcı bir oturumla bağlantısı kesildiyse, ileti SYSTEM.MQTT.TRANSMIT.QUEUE içinde saklanır; bkz. MQTT durumsuz ve durumlu oturumlar. İstemci oturuma yeniden bağlandığında, istemci istemciye iletilir.

If you send a non-persistent message it is sent to the client with en çok bir kez quality of service, QoS=0. Kalıcı bir iletiyi doğrudan istemciye gönderdiğinizde, varsayılan olarak bu ileti tam olarak bir kez hizmet kalitesi QoS=2 ile gönderilir. İstemcinin bir kalıcılık mekanizması bulunmayabilir gibi, istemci doğrudan gönderilen iletiler için kabul ettiği hizmet kalitesini düşürebilir. 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 istemci tanıtıcısı, yetkilendirme ve kimlik doğrulaması

The telemetry (MQXR) service publishes, or subscribes to, IBM MQ topics on behalf of MQTT clients, using MQTT channels. IBM MQ yöneticisi, IBM MQ yetki kimliği için kullanılan MQTT kanal kimliğini yapılandırır. Yönetici, kanal için ortak bir kimlik tanımlayabilir ya da kanala bağlı bir istemcinin Kullanıcı Adı ya da ClientIdentifier' ini kullanabilir.

Telemetri (MQXR) hizmeti, istemci tarafından sağlanan Kullanıcı adı'yı kullanarak ya da bir istemci sertifikası kullanarak istemcinin kimliğini doğrulayabilir. Kullanıcı Adı' in kimliği, istemci tarafından sağlanan bir parola kullanılarak doğrulanır.

Özetlemek için: Müşteri kimliği, istemci kimliğinin seçimidir. Bağlama bağlı olarak, istemci ClientIdentifier, Username(Kullanıcı Adı), yönetici tarafından oluşturulan ortak bir istemci kimliği ya da istemci sertifikası ile tanımlanır. Özgünlük denetimi için kullanılan istemci tanıtıcısının, yetkilendirme için kullanılan tanıtıcı olması gerekmez.

MQTT istemci programları, sunucuya bir MQTT kanalı kullanılarak gönderilen Kullanıcı adı ve Parola'yı ayarlar. Ayrıca, bağlantıyı şifrelemek ve bağlantıyı doğrulamak için gereken TLS özelliklerini de ayarlayabilirler. Yönetici, MQTT kanalının doğrulanıp doğrulanmayacağını ve kanalın nasıl doğrulanıp doğrulanmayacağını belirler.

Bir MQTT istemcisine IBM MQ nesnelere erişim yetkisi vermek için, istemcinin ClientIdentifier ya da Kullanıcı Adı' ini yetkilendirin ya da ortak bir istemci kimliğini yetkilendirin. Bir istemcinin IBM MQ'a bağlanmasına izin vermek için, Kullanıcı adı'nın kimliğini doğrulayın ya da bir istemci sertifikası kullanın. Configure JAAS to authenticate the Kullanıcı Adı, and configure TLS to authenticate a client certificate.

İstemcide bir Parola ayarlıysa, VPN'yi kullanarak bağlantıyı şifreleyin ya da parolayı özel tutmak için MQTT kanalını TLS'yi kullanacak şekilde yapılandırın.

İstemci sertifikalarını yönetmek zordur. Bu nedenle, parola kimlik doğrulamasıyla ilgili riskler kabul edilebilir bir durumsa, parola doğrulaması genellikle istemcilerin kimliğini doğrulamak için kullanılır.

İstemci sertifikasını yönetmenin ve depolamanın güvenli bir yolu varsa, sertifika kimlik doğrulamasına güvenmek mümkün olur. Ancak, sertifikaların telemetrenin kullanıldığı ortam tiplerinde güvenli bir şekilde yönetilebilir olması çok ender bir durum. Bunun yerine, istemci sertifikalarını kullanan aygıtların kimlik doğrulaması, sunucudaki istemci parolalarının doğrulanarak tamamlanır. Ek karmaşıklık nedeniyle, istemci sertifikalarının kullanımı son derece hassas uygulamalarla sınırlandırılmıştır. İki kimlik doğrulama biçiminin kullanılması iki etkenli kimlik doğrulaması olarak adlandırılır. Parola gibi faktörlerden birini bilmeniz gerekir; örneğin, bir sertifika gibi.

chip-ve-pin cihazı gibi son derece hassas bir uygulamada, dahili donanım ve yazılıma müdahale etmek için üretim sırasında cihaz kapalı olarak kilitlenir. Aygıt güvenilen, zaman sınırlanmış bir istemci sertifikası kopyalanmıştır. Aygıt, kullanılabilmesi için yere konuşlandırılır. Aygıt her kullanımında, parola kullanılarak ya da akıllı karttan başka bir sertifika kullanılarak daha fazla kimlik doğrulaması gerçekleştirilir.

Linux Windows AIX MQTT istemci kimliği ve yetkilendirmesi

IBM MQ nesnelere erişim yetkisi için istemci tanıtıcısını (Kullanıcı adı) ya da ortak bir istemci kimliğini kullanın.

IBM MQ yöneticisi, MQTT kanalının kimliğini seçmek için üç seçeneğe sahiptir. Yönetici, istemci tarafından kullanılan MQTT kanalını tanımlarken ya da değiştirirken seçimi yapar. Kimlik, IBM MQ konularına erişimi yetkilendirmek için kullanılır. Seçim aşağıdaki sırada yapılır:

1. İstemci tanıtıcısı (bkz. [USECLNTID](#)).
2. Yöneticinin kanal için sağladığı kimlik (kanaldaki MCAUSER). Bkz. [MCAUSER](#)).
3. Önceki seçeneklerden hiçbiri geçerli değilse, MQTT istemcisinden ([Kullanıcı adı](#)) geçirilen [Kullanıcı adı](#) , [MqttConnectOptions](#) sınıfının bir özneliğinden biri olur. İstemcinin hizmete bağlamadan önce ayarlanması gerekir. Varsayılan değeri boş değerdir).

Beladan kaçının: Bu süreç tarafından seçilen kimlik bundan sonra, örneğin DISPLAY CHSTATUS (MQTT) komutuna, istemcinin MCAUSER adıyla anılır. Bunun, (2) numaralı seçeneğe başvuruda bulunan kanalın MCAUSER ile aynı kimlik olması gerekmediği konusunda dikkat edin.

Use the IBM MQ **setmqaut** command to select which objects, and which actions, are authorized to be used by the identity associated with the MQTT channel. Örneğin, aşağıdaki kod, QM1kuyruk yöneticisi yöneticisi tarafından sağlanan bir kanal kimliğine MQTTClientyetki 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
```

Linux

Windows

AIX

MQTT istemcilerinin IBM MQ nesnelere erişmesi

İçin yetki verme

MQTT istemcilerine IBM MQ Nesnelere abone olma ve abone olma yetkisi vermek için bu adımları izleyin. Bu adımlar, dört alternatif erişim denetimi örüntülerini izlemektedir.

Başlamadan önce

MQTT istemcilerinin, bir telemetri kanalına bağlandığında bir kimlik atanarak IBM MQ içindeki nesnelere erişme yetkisi vardır. The IBM MQ Administrator configures the telemetry channel using IBM MQ Explorer to give a client one of three types of identity:

1. `ClientIdentifier`
2. `Kullanıcı adı`
3. Yöneticinin kanala atadığı ad.

Hangi tip kullanılırsa kullanılırsa, kimlik kurulu yetkilendirme hizmeti tarafından birincil kullanıcı olarak IBM MQ olarak tanımlanmalıdır. The default authorization service on Windows or Linux is called the Object Authority Manager (OAM). OAM kullanıyorsanız, kimlik bir kullanıcı kimliği olarak tanımlanmalıdır.

Bir istemci ya da istemci grubu vermek için, IBM MQ' ta tanımlanan konuları yayınlama ya da abone olma izni vermek için bu kimliği kullanın. Bir MQTT istemcisi bir konuya abone olmuşsa, elde edilen yayınları alma izni vermek için kimliği kullanın.

Her biri bireysel erişim izinlerine gereksinim duyan, on binlerce MQTT istemcisiyle bir sistemi yönetmek zordur. Tek bir çözüm, ortak kimlikleri tanımlamak ve tek tek MQTT istemcilerini ortak kimliklerden biriyle ilişkilendirmek. Farklı izin birleşimleri tanımlamak için gereksinim duyduğunuz kadar çok ortak kimlik tanımlayın. Bir diğer çözüm de, binlerce kullanıcıyla işletim sisteminden daha kolay bir şekilde başa çıkabilen kendi yetkilendirme hizmetinizi yazmanız.

You can combine MQTT clients into common identities in two ways, using the OAM:

1. Her biri, denetimcinin IBM MQ Explorer ' ı kullanarak ayırdığı farklı bir kullanıcı kimliği ile birden çok telemetri kanalı tanımlayın. Farklı TCP/IP kapı numaralarını kullanarak bağlanan istemciler farklı telemetri kanallarıyla ilişkilendirilir ve farklı kimlikler atanır.
2. Tek bir telemetri kanalı tanımlayın, ancak her bir istemciye küçük bir kullanıcı kimlikleri kümesinden bir `Kullanıcı adı` seçin. Sistem yöneticisi, istemci `Kullanıcı Adı` ' i kimliği olarak seçmek için telemetri kanalını yapılandırır.

Bu görevde, nasıl ayarlandığından bağımsız olarak, telemetri kanalının kimliği *mqttUser* olarak adlandırılır. İstemcilerin derlemleri farklı kimlikler kullanırsa, her istemci toplaması için bir tane olmak üzere birden çok *mqttUsers* kullanın. Görev OM ' yi kullandıkça, her *mqttUser* bir kullanıcı kimliği olmalıdır.

Bu görev hakkında

Bu görevde, belirli gereksinimlere göre uyarlayabileceğiniz dört erişim denetimi kalıbından oluşan bir seçiminiz vardır. Kalıplar, erişim denetiminin ayrıntı düzeylerine göre farklılık gösterir.

- “Erişim denetimi yok” sayfa 305
- “Kaba erişim denetimi” sayfa 305
- “Orta-parçalı erişim denetimi” sayfa 305
- “Parçalı erişim denetimi” sayfa 306

The result of the models is to assign *mqttUsers* sets of permissions to publish and subscribe to IBM MQ, and receive publications from IBM MQ.

Erişim denetimi yok

MQTT istemcilerine IBM MQ yönetim yetkisi verilir ve herhangi bir nesne üzerinde herhangi bir işlem gerçekleştirebilirler.

Yordam

1. Tüm MQTT istemcilerinin kimliği olarak hareket etmek için *mqttUser* kullanıcı kimliği oluşturun.
2. *mqm* grubuna *mqttUser* ekleyin; bkz. [Windows üzerinde bir gruba kullanıcı ekleme](#) ya da [Creating and managing groups on Linux](#)

Kaba erişim denetimi

MQTT istemcilerinin yayınlama, abone olma ve MQTT istemcilerine ileti gönderme yetkileri vardır. Diğer işlemleri gerçekleştirme ya da diğer nesnelere erişme yetkisi yoktur.

Yordam

1. Tüm MQTT istemcilerinin kimliği olarak hareket etmek için *mqttUser* kullanıcı kimliği oluşturun.
2. Tüm konuları yayınlamak ve tüm konuları abone etmek ve yayınları MQTT istemcilerine göndermek için *mqttUser* ' i yetkilendirin.

```
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-parçalı erişim denetimi

MQTT istemcileri, farklı konu kümelerine abone olmak ve abone olmak ve MQTT istemcilerine ileti göndermek için farklı gruplara ayrılmıştır.

Yordam

1. Yayınlama/abone olma konu ağacında birden çok kullanıcı kimliği (*mqttUsers*) ve birden çok yönetimle görevli konu yaratın.
2. Farklı konulara farklı *mqttUsers* ' e 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 grup *mqtt* oluşturun ve tüm *mqttUsers* ' ları gruba ekleyin.
4. Authorize *mqtt* to send topics to MQTT clients.

```
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

Parçalı erişim denetimi

MQTT istemcileri, gruplara nesnelere üzerinde eylemler gerçekleştirmeleri için yetki veren, var olan bir erişim denetimi sistemine dahil edilir.

Bu görev hakkında

Bir kullanıcı kimliği, gerektirdiği yetkiler doğrultusunda bir ya da daha çok işletim sistemi grubuna atanmış. IBM MQ uygulamaları yayınlandıysa ve MQTT istemcileriyle aynı konu alanına abone olduysa, bu modeli kullanın. Gruplar, Publish X, Subscribe Y ve mqtt olarak adlandırılır.

Publish X

Publish X gruplarının üyeleri *topicX* a yayınlanabilir.

Subscribe Y

Subscribe Y gruplarının üyeleri *topicY* e abone olabilir.

mqtt

Members of the *mqtt* group can send publications to MQTT clients.

Yordam

1. Yayınlama/abone olma konu ağacında birden çok yönetici konu için ayrılmış birden çok grup, Publish X ve Subscribe Y oluşturun.
2. Create a group mqtt.
3. Birden çok kullanıcı kimliği (*mqttUsers*) yaratın ve yetkilendirildikleri şeye bağlı olarak, kullanıcıları herhangi bir gruba ekleyin.
4. Authorize different Publish X and Subscribe X groups to different topics, and authorize the *mqtt* group to send messages to MQTT clients.

```
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

Bir parola kullanarak MQTT istemci kimlik

doğrulaması

İstemci parolasını kullanarak Kullanıcı adı 'nın kimliğini doğrulayın. İstemcinin kimliğini doğrulamak ve konulara abone olmak için istemciyi yetkilendirmek için kullanılan kimliğe farklı bir kimlik kullanarak doğrulayabilirsiniz.

Telemetri (MQXR) hizmeti, istemcinin Kullanıcı adı kimliğini doğrulamak için JAAS 'ı kullanır. JAAS , MQTT istemcisi tarafından sağlanan Parola 'yı kullanır.

client yöneticisi, istemcinin bağlandığı MQTT kanalını yapılandırarak Kullanıcı adı 'nın doğrulanıp doğrulanmayacağını ya da kimlik doğrulaması yapılamayacağı konusunda IBM MQ yöneticisi karar verir. İstemciler farklı kanallara atanabilir ve her kanal, istemcilerinin kimliklerini farklı şekillerde doğrulamak üzere yapılandırılabilir. JAAS 'ı kullanarak, istemcinin kimliğini doğrulamak zorunda olan ve isteğe bağlı olarak istemciyi doğrulayabilecek yöntemleri yapılandırabilirsiniz.

Kimlik doğrulaması için kimlik seçimi, yetkilendirme için kimlik seçmesini etkilemez. Yönetimle ilgili kolaylıklar için bir yetkilendirme için ortak bir kimlik ayarlamak isteyebilirsiniz, ancak her bir kullanıcının kimliğini kullanması için kimlik doğrulaması yapmak isteyebilirsiniz. Aşağıdaki yordama göre, tek tek kullanıcıların ortak bir kimliğe sahip olması için kimlik doğrulama adımları özetlenmiştir:

1. The IBM MQ administrator sets the MQTT channel identity to any name, such as MQTTClientUser, using IBM MQ Explorer.

2. IBM MQ yöneticisi, MQTTCClient ' e herhangi bir konuyu yayınlayıp abone olmak için yetki verir:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTCClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTCClient -all +pub +sub
```

3. MQTT istemci uygulaması geliştiricisi, sunucuya bağlanmadan önce bir MqttConnectOptions nesnesi yaratır ve Kullanıcı adı ve Parola seçeneklerini ayarlar.
4. The security developer creates a JAAS LoginModule to authenticate the Kullanıcı Adı with the Parola and includes it in the JAAS configuration file.
5. The IBM MQ administrator configures the MQTT channel to authenticate the Username of the client using JAAS.

Linux Windows AIX TLS ' yi kullanarak MQTT istemci kimlik 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 'yi kullanarak MQTT istemci kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 308](#)
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 görevler

[TLS kullanan istemci kimlik doğrulaması için MQTT istemci yapılandırması](#)

Linux Windows AIX **TLS 'yi 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.

Set the property, com.ibm.mq.MQTT.ClientAuth of a TLS telemetry channel to REQUIRED to force all clients connecting on that channel to provide proof that they have verified digital certificates. İstemci sertifikalarının kimliği, sertifika yetkililerinden alınan sertifikalar kullanılarak doğrulanır ve güvenilir bir kök sertifikasına yol sağlar. İstemci sertifikası kendinden onaylı bir sertifika ise ya da bir sertifika yetkilisinden alınan bir sertifika tarafından imzalandıysa, istemcinin genel olarak imzalanmış sertifikaları ya da sertifika yetkilisi, sunucuda güvenli bir şekilde saklanmalıdır.

Genel olarak imzalanan istemci sertifikasını ya da sertifikayı telemetri kanalı anahtar deposunda bulunan sertifika yetkilisinden yerleştirin. Sunucuda, genel olarak imzalanmış sertifikalar, ayrı bir güvenilirlik deposunda değil, özel olarak imzalanmış sertifikalarla aynı anahtar dosyasına saklanır.

Sunucu, sahip olduğu tüm genel sertifikalar ve şifreleme takımlarını kullanarak gönderildiği istemci sertifikalarının imzasını doğrular. Sunucu, anahtar zincirini doğrular. Kuyruk yöneticisi, sertifikayı sertifika iptal listesine göre test etmek üzere yapılandırılabilir. Kuyruk yöneticisi iptali ad listesi özelliği SSLCRLNL' dir.

İstemcinin gönderdiği sertifikalardan herhangi biri sunucu anahtar depodaki bir sertifika tarafından doğrulanırsa, istemcinin kimliği doğrulanır.

The IBM MQ administrator can configure the same telemetry channel to use JAAS to check the UserName or ClientIdentifier of the client with the client Parola.

Birden çok telemetri kanalı için aynı anahtar deposunu kullanabilirsiniz.

Aygıttaki parola korumalı istemci anahtar deposunda en az bir dijital sertifikana ilişkin doğrulama, istemciyi sunucu tarafından doğrulatabiliyor. The digital certificate is only used for authentication by IBM MQ. İstemcinin TCP/IP adresini doğrulamak ya da yetki ya da hesap için istemcinin kimliğini belirlemek için kullanılmaz. Sunucu tarafından kabul edilen istemcinin kimliği, istemcinin Kullanıcı Adı ya da ClientIdentifier 'sidir ya da IBM MQ yöneticisi tarafından yaratılan bir kimdir.

İstemci kimlik doğrulaması için TLS şifre takımlarını da kullanabilirsiniz. SHA-2 şifreleme takımlarını kullanmayı planlıyorsanız, bkz. [“System requirements for using SHA-2 cipher suites with MQTT channels” sayfa 310.](#)

İlgili kavramlar

“TLS kullanılarak kanal kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 309

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 TANIMLA \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

[CipherSpecs ve CipherSuites](#)

Linux Windows AIX TLS ' yi kullanarak telemetri kanalı kimlik 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. changeitvarsayılan 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.

Linux Windows AIX TLS kullanılarak kanal 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.

telemetri kanalının sunucuda kullanacağı anahtar deposunda, özel anahtisiyle imzalanmış olan sunucunun dijital sertifikasını depp edin. Anahtar zincirini istemciye iletmek istiyorsanız, anahtar depodaki anahtar zincirinde bulunan sertifikaları saklayın. TLS ' yi kullanmak için IBM MQ gezginini kullanarak telemetri kanalını yapılandırın. Anahtar deposunun yolunu ve anahtar deposuna erişmek için geçiş tümcesini belirtin. Kanalın TCP/IP kapı numarasını ayarlamadıysanız, TLS telemetri kanalı kapı numarası varsayılan değer olarak 8883 'e ayarlanır.

Kanal kimlik doğrulaması için TLS şifre takımlarını da kullanabilirsiniz. SHA-2 şifreleme takımlarını kullanmayı planlıyorsanız, bkz. [“System requirements for using SHA-2 cipher suites with MQTT channels” sayfa 310.](#)

İlgili kavramlar

“TLS 'yi kullanarak MQTT istemci kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 308 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 TANIMLA \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

[CipherSpecs ve CipherSuites](#)

Linux Windows AIX System requirements for using SHA-2 cipher suites with MQTT channels

SHA-2 şifreleme takımlarını destekleyen bir Java sürümünü kullanıyorsanız, MQTT (teletext) kanallarınızı ve istemci uygulamalarınızı güvenceye almak için bu paketleri kullanabilirsiniz.

For IBM MQ 8.0 , which includes the telemetry (MQXR) service, the minimum Java version is Java 7 from IBM, SR6. SHA-2 cipher suites are supported by default in Java 7 from IBM, SR4 onwards. Bu nedenle, MQTT (teletext) kanallarınızı güvenli kılmak için telemetri (MQXR) hizmetiyle SHA-2 şifre takımlarını kullanabilirsiniz.

Farklı bir JRE ile bir MQTT istemcisi çalıştırıyorsanız, bunun SHA-2 şifreleme takımlarını da desteklediğinden emin olmanız gerekir.

İlgili kavramlar

[Telemetri \(MQXR\) hizmeti](#)

“TLS kullanılarak kanal kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 309

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 TANIMLA \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

Linux Windows AIX telemetri kanallarında yayın gizliliği

Telemetri kanallarında her iki yönde gönderilen MQTT yayınlarının gizliliği, bağlantı üzerinden iletimleri şifrelemek için TLS kullanılarak güvenli kılınmaktadır.

Telemetri kanallarına bağlanan MQTT istemcileri, kanalda iletilen yayınların gizliliğini simetrik anahtar şifrelemesi kullanarak güvenli hale getirmeleri için TLS 'yi kullanır. Uç noktaların kimliği doğrulanmadığı için, kanal şifrelemesine tek başına güvenemezsiniz. Güvenli gizliliği sunucu ya da 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 [“TLS 'yi kullanarak telemetri kanalı kimlik doğrulaması” sayfa 309' a](#) danışın.

Sunucu doğrulanmadan TLS bağlantılarını şifrelemek, orta ve orta şiddetteki saldırılara karşı bağlantı oluşturur. Değiştirdiğiniz bilgiler kulak misafirine karşı korunsa da, kiminle değiş tokuş edildiğinizi bilmenize gerek yok. Ağı denetlemezseniz, IP iletimlerinizi ya da uç nokta olarak maskeleyiş işlemi sırasında bir kişiye maruz kalmakta olduğunuz ortaya çıkar.

Anonim TLS 'yi destekleyen bir Diffie-Hellman anahtar değiş tokuş CIPHERSpec kullanarak sunucuyu doğrulamadan şifrelenmiş bir TLS bağlantısı oluşturabilirsiniz. İstemci ile sunucu arasında paylaşılan ve TLS iletimlerini şifrelemek için kullanılan ana güvenlik dizgisi, özel olarak imzalanmış bir sunucu sertifikası değiş tokuş olarak kurulur.

Anonim bağlantılar güvensiz olduğu için, çoğu TLS somutlaması anonim CIPHERSpecs özelliğini kullanmak için varsayılan değer değildir. Bir telemetri kanalı tarafından TLS bağlantısı için bir istemci isteği kabul edilirse, kanala geçiş tümceciği tarafından korunan bir anahtar deposu olmalıdır. Varsayılan olarak, TLS somutlamaları anonim CIPHERSpecs kullanmadığı için, anahtar deposunun istemcinin doğrulayabileceği özel olarak imzalanmış bir sertifika içermesi gerekir.

Anonim CIPHERSpecs değerini kullanırsanız, sunucu anahtar deposu var olmalıdır, ancak özel olarak imzalanmış herhangi bir sertifika içermemelidir.

Şifrelenmiş bir bağlantı kurmanın başka bir yolu da, istemciye güven sağlayıcısının yerine kendi uygulamanızı koymanızdır. Güvenilir sağlayıcınız sunucu sertifikasının kimliğini doğrulamazdı, ancak bağlantı şifrelenir.



Uyarı: When using TLS with MQTT you can use large messages, however, there might be a possible performance impact when doing so. MQTT , küçük iletiler (genellikle 1KB ve 1MB boyutlarında) işlenmek üzere eniyelenir.

Linux

Windows

AIX

MQTT Java istemcilerinin ve telemetri

kanallarının TLS yapılandırması

Telemetri kanalını ve MQTT Java istemcisini doğrulamak için TLS 'yi yapılandırın ve bunlar arasında iletilerin aktarımı şifrelenir. MQTT Java istemcileri, TLS 'yi 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 aği üzerinden TCP/IP 'yi kullanarak MQTT istemcilerini telemetri kanallarına bağlayabilirsiniz.

TCP/IP üzerinden TLS iletişim kuralını kullanmak için bir Java MQTT istemcisi ile telemetri kanalı arasında bağlantı yapılandırabilirsiniz. Güvenli olan, JSSE 'yi kullanmak için TLS 'yi nasıl yapılandırmanıza 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. İstemciyle kuyruk yöneticisi arasındaki iletileri şifrele; bkz. [“TLS 'yi kullanarak MQTT istemci kimlik doğrulaması” sayfa 307](#)
2. Bir MQTT istemcisini yalnızca güvenilir bir telemetri kanalına bağlayın. İstemciyle kuyruk yöneticisi arasındaki iletileri şifrele; bkz. [“TLS 'yi kullanarak telemetri kanalı kimlik doğrulaması” sayfa 309.](#)
3. İstemciyle kuyruk yöneticisi arasındaki iletileri şifrele; bkz. [“telemetri kanallarında yayın gizliliği” sayfa 310.](#)

JSSE yapılandırma parametreleri

Bir TLS bağlantısının konfigürasyonunun tanımlanını değiştirmek için JSSE parametrelerini değiştirin. JSSE yapılandırma parametreleri üç küme halinde düzenlenmiştir:

1. [MQ Telemetry kanalı](#)
2. [MQTT Java istemcisi](#)
3. [JRE](#)

IBM MQ Explorer 'ı kullanarak telemetri kanalı parametrelerini yapılandırın.

MqttConnectionOptions.SSLProperties öznitelide MQTT Java Client parametrelerini ayarlayın. JRE

güvenlik deęiřtirmelerini istemci ve sunucu üzerindeki JRE güvenlik dizinindeki dosyaları düzenleyerek deęiřtirin.

MQ Telemetry channel

IBM MQ Explorer 'ı kullanarak tüm telemetri kanalı TLS parametrelerini ayarlayın.

ChannelName

ChannelName , tüm kanallarda zorunlu bir parametredir.

Kanal adı, belirli bir kapı numarasıyla ilişkilendirilen kanalı tanımlar. MQTT istemcilerini yönetmenize yardımcı olacak ad kanalları.

PortNumber

PortNumber , tüm kanallarda isteęe baęlı bir parametredir. Bu, TCP kanalları için 1883 , TLS kanalları için de 8883 deęerine ayarlanır.

Bu kanalla ilişkilendirilen TCP/IP kapı numarası. MQTT istemcileri, kanal için tanımlanan kapıyı belirterek bir kanala baęlanır. Kanalda TLS özellikleri varsa, istemci TLS iletişim kuralını kullanarak baęlanmalıdır; örneęin:

```
MQTTClient mqttClient = new MqttClient( "ssl://www.example.org:8884", "clientId1");
mqttClient.connect();
```

KeyFileAd

KeyFileAd , TLS kanalları için gerekli bir parametredir. TCP kanalları için atlanmalıdır.

KeyFileAdı , sağladığınız sayısal sertifikalar içeren Java anahtar deposunun yoludur. Sunucuda anahtar deposu tipi olarak JKS, JCEKS ya da PKCS12 ' yi kullanın.

Anahtar deposu tipini, aşağıdaki dosya uzantılarından birini kullanarak tanımlayın:

- .jks
- .jceks
- .p12
- .pkcs12

Diđer dosya uzantılarına sahip bir anahtar deposunun JKS anahtar deposu olduęu varsayılır.

Sunucudaki bir anahtar deposu tipini, istemcideki diđer anahtar deposu tipleriyle birleřtirebilirsiniz.

Sunucunun özel sertifikasını anahtar depoya yerleřtirin. Sertifika, sunucu sertifikası olarak bilinir. Sertifika, bir imzalama yetkilisi tarafından imzalanmış bir sertifika zincirinin ya da sertifika zincirinin bir parçası olabilir.

Bir sertifika zinciri kullanıyorsanız, ilişkili sertifikaları sunucu anahtar deposuna yerleřtirin.

Sunucu sertifikası ve sertifika zincirindeki tüm sertifikalar, sunucunun kimlięini doęrulamak üzere istemcilere gönderilir.

ClientAuth öęesini Requiredolarak ayarladıysanız, anahtar deposunun istemcinin kimlięini doęrulamak için gereken sertifikaları içermesi gerekir. İstemci kendinden onaylı bir sertifika ya da bir sertifika zinciri gönderir ve istemciyi, bu malzemenin anahtar depodaki bir sertifikaya ilişkin ilk doęrulamaları ile doęrulanır. Bir sertifika zinciri kullanarak, bir sertifika, farklı istemci sertifikalarıyla yayınlansa bile, birçok istemciyi doęrulayabilir.

PassPhrase

PassPhrase , TLS kanalları için gerekli bir parametredir. TCP kanalları için atlanmalıdır.

Anahtar deposunu korumak için geçiř tümcecisi kullanılır.

ClientAuth

ClientAuth , isteğe bağlı bir TLS parametresidir. İstemci kimlik doğrulaması için varsayılan değer olarak kullanılır. TCP kanalları için atlanmalıdır.

İstemcinin telemetri kanalına bağlanmasını izin vermeden önce, telemetri (MQXR) hizmetinin istemcinin kimliğini doğrulamasına izin vermek için ClientAuth seçeneğini belirleyin.

ClientAuthseçeneğini ayarsanız, istemci sunucuya TLS ' yi kullanarak bağlanmalı ve sunucuyu doğrulamalıdır. ClientAuthayarına yanıt olarak istemci, dijital sertifikasını sunucuya gönderir ve diğer sertifikalar anahtar deposuyla birlikte gönderilir. Dijital sertifikası, istemci sertifikası olarak bilinir. Bu sertifikaların kimliği, kanal anahtar deposunda ve JRE cacerts mağazasında tutulan sertifikalara göre doğrulanır.

CipherSuite

CipherSuite , isteğe bağlı bir TLS parametresidir. Varsayılan olarak, tüm etkin CipherSpecs ' ı deneyecek şekilde ayarlanır. TCP kanalları için atlanmalıdır.

If you want to use a particular CipherSpec, set CipherSuite to the name of the CipherSpec that must be used to establish the TLS connection.

Telemetri hizmeti ve MQTT istemcisi, her bir uçta etkinleştirilen tüm CipherSpecs 'nden ortak bir CipherSpec ' i kararlaştırabiliyor. Bağlantının her iki ucunda ya da her iki ucunda belirli bir CipherSpec belirtilirse, bu, diğer uçta CipherSpec ile eşleşmelidir.

JSSSE ' ye ek sağlayıcılar ekleyerek ek şifreleri kurun.

Federal Bilgi İşleme Standartları (FIPS)

FIPS, isteğe bağlı bir ayardır. Varsayılan olarak ayarlanmaz.

Kuyruk yöneticisinin özellikler panosunda ya da **runmqsc**komutunu kullanarak SSLFIPSayarlayın. SSLFIPS , yalnızca FIPS onaylı algoritmaların kullanılıp kullanılmayacağını belirtir.

İptal adlistesi

İptal ad listesi isteğe bağlı bir ayardır. Varsayılan olarak ayarlanmaz.

Kuyruk yöneticisinin özellikler panosunda ya da **runmqsc**komutunu kullanarak SSLCRLNL ' yi ayarlayın. SSLCRLNL , sertifika iptal konumlarını sağlamak için kullanılan kimlik doğrulama bilgileri nesnelerinin ad listesini belirtir.

TLS özelliklerini ayarlayan başka bir kuyruk yöneticisi değiştirgesi kullanılmadı.

MQTT Java istemcisi

MqttConnectionOptions.SSLProperties içinde Java istemcisi için TLS özelliklerini ayarlayın; 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 açıklanmıştır. MQTT istemci kitaplıklarına ilişkin istemci API belgelerine ilişkin bağlantılar için bkz. [MQTT istemci programlama başvurusu](#).

Protokol

Protokol isteğe bağlıdır.

Protokol, telemetri sunucusu ile müzakere halinde seçilir. Belirli bir protokole gereksinim duyarsanız, bir protokol seçebilirsiniz. Telemetri sunucusu, iletişim kuralını desteklemiyorsa bağlantı başarısız olur.

ContextProvider

ContextProvider isteğe bağlıdır.

KeyStore

KeyStore isteğe bağlıdır. İstemcinin kimlik doğrulamasını zorunlu kılacak sunucuda ClientAuth ayarlandıysa, bu değeri yapılandırın.

İstemci sayısal sertifikasını, özel anahtarını kullanarak, anahtar deposuna yerleştirin. Anahtar deposu yolunu ve parolasını belirtin. Tip ve sağlayıcı isteğe bağlıdır. JKS, varsayılan tiptir ve IBMJCE varsayılan sağlayıcısıdır.

Yeni bir anahtar deposu sağlayıcısı ekleyen bir sınıfa gönderme yapmak için farklı bir anahtar deposu sağlayıcısı belirtin. Pass the name of the algorithm used by the keystore provider to instantiate the KeyManagerÜreticisi by setting the key manager name.

TrustStore

TrustStore isteğe bağlıdır. JRE cacerts mağazasına güvendiğiniz tüm sertifikaları yerleştirebilirsiniz.

İstemci için farklı bir güvenilir deponun olmasını istiyorsanız, güvenli deponun konfigürasyonunu tanımlayın. Sunucu, önceden cacerts içinde saklanan kök sertifikasına sahip, iyi bilinen bir sertifika kuruluşu tarafından verilen bir sertifikayı kullanıyorsa, güvenilir deponun konfigürasyonunu tanımlayamayabilirsiniz.

Sunucunun genel olarak imzalanmış sertifikasını ya da güvenli depona kök sertifikayı ekleyin ve güvenilirlik deposu yolunu ve parolasını belirtin. JKS, varsayılan tiptir ve IBMJCE varsayılan sağlayıcısıdır.

Yeni bir güvenilirlik deposu sağlayıcısı ekleyen bir sınıfa gönderme yapmak için farklı bir güvenilirlik deposu sağlayıcısı belirtin. Güvenilirlik deposu sağlayıcısı tarafından kullanılan algoritmanın adını, güvenilirlik yöneticisi adını ayarlayarak TrustManagerÜreticisi ' yi somutlaştırmak için kullanın.

JRE

Java güvenliğinin hem istemci, hem de sunucu üzerinde TLS davranışını etkileyen diğer yönleri JRE ' de yapılandırılır. Windows üzerindeki yapılandırma dosyaları *Java Installation Directory*\jre\lib\security içinde yer alıyor. IBM MQ ile birlikte gönderilen JRE ' yi kullanıyorsanız, yol aşağıdaki çizelgede gösterildiği gibidir:

Çizelge 18. JRE TLS yapılandırma dosyaları için altyapıya göre dosya yolları	
Altyapı	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, tanınmış sertifika yetkililerinin kök sertifikalarını içerir. Bir güvenilirlik deposu belirtmediğiniz sürece, cacerts varsayılan olarak kullanılır. cacerts deposunu kullanıyorsanız ya da bir güvenilirlik deposu sağlamıyorsa, güvenlik gereksinimlerinizi karşılamak için cacerts içindeki imzalayanlar listesini gözden geçirmeniz ve düzenlemeniz gerekir.

You can open cacerts using the IBM MQ command `stmqikm`.which runs the IBM Key Management utility. Open cacerts as a JKS file, using the password `changeit`. Dosyayı güvenli kılmak için parolayı değiştirin.

Güvenlik sınıflarının yapılandırılması

Ek güvenlik sağlayıcıları ve diğer varsayılan güvenlik özelliklerini kaydetmek için `java.security` dosyasını kullanın.

İzinler

Kaynaklara verilen izinleri değiştirmek için `java.policy` dosyasını kullanın. `javaws.policy`, `javaws.jar` için izin verir.

Şifreleme gücü

Bazı JRES 'ler güç şifrelemesi azaltılmış. Anahtarları anahtar depolarına aktaramıyorsanız, azaltılmış güvenlik düzeyi şifreleme nedeni olabilir. Either, try starting **ikeyman** using the **strong** command, or download strong, but limited jurisdiction files from [IBM geliştirici setleri](#), Güvenlik bilgileri.

Önemli: Ülkenizde, şifreleme yazılımların başka bir ülkeye alınması, bulundurmaları, kullanılması ya da yeniden dışa aktarılması konusunda sınırlamalar olabilir. Sınırlanmamış ilke dosyalarını karşıdan yüklemeyen ya da karşıdan yüklemeyen önce, ülkenizin yasalarını denetlemeniz gerekir. İzin verilip verilmediğini belirlemek için şifreleme yazılımını içe aktarma, bulundurma, kullanma ve yeniden ihracata ilişkin ilkeleri ve düzenlemeleri denetleyin.

İstemcinin herhangi bir sunucuya bağlanmasına izin vermek için güven sağlayıcısını değiştirin

Bu örnek, bir güven sağlayıcısının nasıl ekleneceğini ve MQTT istemci kodundan nasıl başvurucağını gösterir. Bu örnek, istemci ya da sunucu için kimlik doğrulaması gerçekleştirmez. Sonuçtaki TLS bağlantısı, kimlik doğrulaması yapılmadan şifrelenir.

Şekil 19 sayfa 315 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 19. 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 20. `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 21. `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 22. AcceptAllX509TrustManager.java

Linux

Windows

AIX

Telemetri kanalı JAAS yapılandırması

İstemci tarafından gönderilen `Kullanıcı` adı'nın kimliğini doğrulamak için JAAS 'ı yapılandırın.

IBM MQ yöneticisi, JAAS 'ı kullanarak istemci kimlik doğrulaması gerektiren MQTT kanallarını yapılandırır.

JAAS kimlik doğrulamasını gerçekleştirecek her kanal için bir JAAS yapılandırmasının adını belirtin.

Kanallar, aynı JAAS yapılandırmasını kullanabilir ya da farklı JAAS yapılandırmalarını kullanabilir.

Yapılandırmalar `WMQData directory\qmgrs\qMgrName\mqxr\jaas.config` içinde tanımlanır.

`jaas.config` dosyası JAAS yapılandırma adı tarafından düzenlenir. Her bir yapılandırma adı altında Oturum Açma yapılandırmalarının bir listesi yer alıyor; bkz. "[Örnek jaas.config dosyası](#)" sayfa 317.

JAAS , dört standart oturum açma modülü sağlar. Standart NT ve UNIX Oturum Açma modülleri sınırlı değerlerdir.

JndiLoginBirimi

JNDI (Java Naming and Directory Interface) altında yapılandırılan bir dizin hizmetine karşı kimlik doğrulama gerçekleştirir.

Krb5LoginModule

Kerberos protokollerini kullanarak kimlik doğrulama gerçekleştirir.

NTLoginModule

Geçerli kullanıcı için NT güvenlik bilgilerini kullanarak kimlik doğrulama gerçekleştirir.

UnixLoginBirimi

Geçerli kullanıcı için UNIX güvenlik bilgilerini kullanarak kimlik doğrulama gerçekleştirir.

NTLoginModule ya da UnixLoginModule kullanılmasıyla ilgili sorun, telemetri (MQXR) hizmetinin, MQTT kanalının kimliği değil, mqm kimliği ile çalıştırıldığı. mqm , kimlik doğrulaması için NTLoginModule ya da UnixLoginModule ' a aktarılan kimliktir ve istemcinin kimliği değildir.

Bu sorunu çözmek için, kendi Oturum Açma modülünüzü yazın ya da diğer standart Oturum Açma modüllerini kullanın. A sample JAASLoginModule . java is supplied with MQ Telemetry. Bu, `javax.security.auth.spi.LoginModule` arabiriminin bir uygulamasıdır. Kendi kimlik doğrulama yönteminizi geliştirmek için bunu kullanın.

Sağladığınız yeni LoginModule sınıflarının, telemetri (MQXR) hizmetinin sınıf yolunda olması gerekir. Sınıflarınızı, sınıf yolunda bulunan IBM MQ dizinlerine yerleştirmeyin. Kendi dizinlerinizi oluşturun 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 bilgiler içerebilir. CLASSPATH değişkeninde değişkenleri kullanamazsınız; örneğin, CLASSPATH=%CLASSPATH% yanlış. Telemetri (MQXR) hizmeti kendi sınıf yolunu (classpath) ayarlar. service.env içinde tanımlanan CLASSPATH bu dosyaya eklenir.

Telemetri (MQXR) hizmeti, MQTT kanalına bağlı bir istemci için Kullanıcı adı ve Parola'yı döndüren iki geri arama sağlar. The Kullanıcı Adı and Parola are set in the MqttConnectOptions object. Kullanıcı Adı ve Parolaerişimine nasıl erişileceğini gösteren bir örnek için bkz. "[Örnek JAASLoginModule.Login\(\) yöntemi](#)" sayfa 317 .

Örnek jaas.config dosyası

namedyapılandırması, MQXRConfigyapılandırması içeren bir JAAS yapılandırma dosyası örneği

```
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'yı alacak şekilde kodlanmış bir JAAS Oturma Açma modülü 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.

Viewing IBM MQ objects in use by AMQP clients

Farklı IBM MQ kaynaklarını, örneğin, bağlantılar ve abonelikler için AMQP istemcileri tarafından kullanılan şekilde görüntüleyebilirsiniz.

Bağlantılar

AMQP hizmeti başlatıldığında yeni Hconns oluşturulur ve kuyruk yöneticisine bağlanır. Bu Hconns havuzu, AMQP istemcileri ileti yayınlarken kullanılır. Hconns uygulamasını [DISPLAY CONN](#) komutunu kullanarak görüntüleyebilirsiniz. Örneğin:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (APPLDESC LK 'IBM MQ Advanced Message Queuing Protocol*')
```

Bu komut, istemciye özgü Hconns programlarını da gösterir. Boş bir istemci tanıtıcısı özneliğine sahip olan Hconns (Hconns), havuzda kullanılan Hconns

Bir AMQP istemcisi bir AMQP kanalına bağlandığında, kuyruk yöneticisine yeni bir Hconn bağlantısı kurulur. Bu Hconn, AMQP istemcisinin yarattığı abonelikler için zamanuyumsuz olarak ileti tüketmek için kullanılır. [DISPLAY CONN](#) komutunu kullanarak belirli bir AMQP istemcisi tarafından kullanılan Hconn 'u görüntüleyebilirsiniz. Örneğin:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_abcd1234')
```

İstemciler tarafından yaratılan abonelikler

Bir AMQP istemcisi bir konuya abone olduğunda, yeni bir IBM MQ aboneliği yaratılır. Abonelik adı aşağıdaki bilgileri içerir:

- İstemcinin adı. İstemci paylaşılan bir aboneliğe katılmışsa, paylaşımın adı kullanılır.
- İstemcinin abone olduğu konu örneği
- Bir örnek. Bu örnek, istemci paylaşılmayan bir abonelik oluşturduysa `private` ya da istemci paylaşılan bir aboneliğe katılmışsa `share` olur.

Belirli bir AMQP istemcisi tarafından kullanımda abonelikleri görüntülemek için, [DISPLAY SUB](#) komutunu çalıştırın ve `private` örneğine süzgeç uygulayın:

```
DISPLAY SUB(':private:*')
```

Birden çok istemci tarafından kullanımda olan paylaşılan abonelikleri görüntülemek için **DISPLAY SUB** komutunu çalıştırın ve share öneğine süzgeç uygulayın:

```
DISPLAY SUB(' :share:*')
```

Paylaşılan abonelikler birden çok AMQP istemcisi tarafından kullanılabilir için, paylaşılan abonelikten iletileri tüketmekte olan istemcileri görüntülemek isteyebilirsiniz. Bu işlemi, abonelik kuyruğunda bir tanıtıcısı açık olan Hconns ' leri listeleyerek yapabilirsiniz. Şu anda bir paylaşımı kullanmakta olan 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(414D5120514D312020202020202020707E0A565C2D0020)
SUB(:private:recv_e298452:public)
DEST(SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
```

2. Bu kuyruğun üzerindeki 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. Tanıtıcıların her biri için, tutamaç 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, yetki kimliği ve kimlik doğrulaması

Diğer IBM MQ istemci uygulamaları gibi, AMQP bağlantılarını çeşitli şekillerde güvenli hale getirebilirsiniz. AMQP bağlantılarını IBM MQ' e güvenli kılmak için aşağıdaki güvenlik özelliklerini kullanabilirsiniz:

- [Kanal doğrulama kayıtları](#)
- [Bağlantı kimlik doğrulaması](#)
- Kanal MCA kullanıcı yapılandırması
- IBM MQ yetki tanımları
- [TLS bağlantırlığı](#)

Bir güvenlik perspektifinden, bir bağlantı oluşturmak aşağıdaki iki adımdan oluşur:

- Bağlantının devam edip etmeyeceğini belirleme
- Daha sonraki yetki denetimleri için uygulamanın hangi IBM MQ kimliğine ilişkin karar vereceğine karar verilmesi

Aşağıdaki bilgiler, bir AMQP istemcisi bağlantı yapmaya çalıştığında, farklı IBM MQ yapılandırmalarını ve üzerinde çalışılan adımları özetlemektedir. Tüm IBM MQ yapılandırmaları, anlatılan tüm adımları kullanmaz. Örneğin, bazı yapılandırmalar şirket güvenlik duvarının içindeki bağlantılar için TLS kullanmaz ve bazı yapılandırmalar TLS kullanır, ancak kimlik doğrulama için istemci sertifikalarını kullanmaz. Birçok ortam özel ya da özel JAAS birimleri kullanmaz.

Bağlantı kurulması

Aşağıdaki adımlarda, bir AMQP istemcisi tarafından bir bağlantı kurulduğunda ne olacağını açıklanır. Bu adımlar, bağlantının devam edip etmeyeceğini ve uygulamanın yetki denetimi için hangi IBM MQ kimliğinden devralacağını belirler:

1. İstemci IBM MQ ' e TLS bağlantısı açarsa ve sertifika veriyorsa, kuyruk yöneticisi istemci sertifikasını doğrulamayı dener.
2. İstemci kullanıcı adı ve parola kimlik bilgileri sağlıyorsa, kuyruk yöneticisi AMQP SASL çerçevesi alır ve MQ CONNAUTH yapılanışı denetlenir.
3. MQ kanal kimlik doğrulama kuralları denetlenir (örneğin, IP adresinin ve TLS sertifika DN ' inin geçerli olup olmadığı)
4. Kanal kimlik doğrulama kuralları aksini belirlemiyorsa, Kanal MCAUSER değeri verilir.
5. Bir JAAS modülü yapılandırıldıysa, çağrılır
6. MQ CONNECT authority check applied to resulting MQ user ID.
7. Varsayılan bir IBM MQ kimliği ile bağlantı kuruldu.

İleti yayınlanması

Aşağıdaki adımlarda, bir AMQP istemcisi tarafından bir ileti yayınlandığında neler olacağını açıklanır. Bu adımlar, bağlantının devam edip etmeyeceğini ve uygulamanın yetki denetimi için hangi IBM MQ kimliğinden devralacağını belirler:

1. AMQP bağlantı ekleme çerçevesi kuyruk yöneticisinde geliyor. Bağlantı sırasında kurulan MQ kullanıcı kimliği için, belirtilen konu dizgisine ilişkin IBM MQ yayınlama yetkisi denetlenir.
2. İleti, belirtilen konu dizgisine yayınlandı.

Konu örüntülerine abone olma

Aşağıdaki adımlarda, bir AMQP istemcisi bir konu örüntülerine abone olduğunda ne olacağını açıklamalı. Bu adımlar, bağlantının devam edip etmeyeceğini ve uygulamanın yetki denetimi için hangi IBM MQ kimliğinden devralacağını belirler:

1. AMQP bağlantı ekleme çerçevesi kuyruk yöneticisinde geliyor. Bağlantı sırasında kurulan MQ kullanıcı kimliği için, belirtilen konu örüntüye ilişkin IBM MQ abone olma yetkisi denetlenir.
2. Abonelik yaratıldı.

AMQP istemci kimliği ve yetkilendirmesi

IBM MQ nesnelere erişim yetkisi için, AMQP istemci tanıtıcısını, AMQP kullanıcı adını ya da kanalda tanımlı olan ortak bir istemci kimliğini ya da kanal doğrulama kuralındaki bir ortak istemci kimliğini kullanın.

Denetimci, AMQP kanalını tanımlarken ya da değiştirirken, kuyruk yöneticisi CONNAUTH ayarını yapılandırarak ya da kanal doğrulama kurallarını tanımlayarak seçimi yapar. Kimlik, IBM MQ konularına erişimi yetkilendirmek için kullanılır. Seçim, aşağıdakine dayalı olarak yapılır:

1. Kanal USECLNTID öznitelidir.
2. Kuyruk yöneticisi CONNAUTH kuralının ADOPTCTX özniteliği.
3. Kanalda tanımlı MCAUSER özniteliği.
4. Eşleşen bir kanal kimlik doğrulama kuralının USERSRC özniteliği.

Beladan kaçının: Bu işlem tarafından seçilen kimlik bundan sonra, örneğin, istemcinin MCAUSER (MCAUSER) komutu olarak, örneğin DISPLAY CHSTATUS (AMQP) komutuna başvurulmaktadır. Bunun, (2) numaralı seçeneğe atıfta bulunulan kanalın MCAUSER ile aynı kimlik olması gerektiği konusunda dikkat edin.

Use the IBM MQ **setmqaut** command to select which objects, and which actions, are authorized to be used by the identity associated with the AMQP channel. For example, the following commands authorize a channel identity AMQPClient, provided by the administrator of queue manager QM1:

```
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 istemci kullanıcı adını doğrulayın. İstemcinin kimliğini doğrulamak ve konulara abone olmak için istemciyi yetkilendirmek için kullanılan kimlikten farklı bir kimlik kullanarak 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 yordam, yerel işletim sistemi kullanıcılarına ve parolalarına karşı tek tek kullanıcıların kimliğini doğrulamak için kullanılan örnek adımları özetlemekte ve başarılı olursa, AMQPUser ortak kimliğini benimsiyor:

1. The IBM MQ administrator sets the AMQP channel MCAUSER identity to any name, such as AMQPUser, using IBM MQ Explorer.
2. IBM MQ yöneticisi, AMQPUser ' e herhangi bir konuyu yayınlayıp abone olmak için yetki verir:

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPUser -all +pub +sub +connect
```

3. The IBM MQ administrator configures an IDPWOS CONNAUTH rule to check the user name and password presented by the client. CONNAUTH kuralı CHCKCLNT (REQUIREN) ve ADOPTCTX (NO) ayarlamalıdır.

Not: Kanal doğrulama kurallarını kullanmanız ve kuyruk yöneticisiyle bağlantı üzerinde daha fazla denetime izin vermek için ayrıcalıkları olmayan bir kullanıcıya MCAUSER kanal özniteliğini ayarlamanız önerilir.

Kanallarda yayın gizliliği

AMQP kanallarının her iki yönde gönderdiği AMQP yayınlarının gizliliği, bağlantı üzerinden iletimleri şifrelemek için TLS ' yi kullanarak güvenli kılınmaktadır.

AMQP kanallarına bağlanan AMQP istemcileri, kanalda iletilen yayınların gizliliğini simetrik anahtar şifrelemesi kullanarak güvenli hale getirmeleri için TLS ' yi kullanır. Uç noktaların kimliği doğrulanmadığı için, kanal şifrelemesine tek başına güvenemezsiniz. Güvenli gizliliği sunucu ya da karşılıklı kimlik doğrulamasıyla birleştirin.

TLS ' nin kullanılmasına bir alternatif olarak, IPsec gibi bir tür Sanal Özel Ağ (VPN), 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 kurmuyorsunuz. AMQP istemcilerini, VPN ağı üzerinden TCP/IP kullanarak AMQP kanallarına bağlayabilirsiniz.

Sunucu doğrulanmadan TLS bağlantılarını şifrelemek, orta ve orta şiddetteki saldırılara karşı bağlantı oluşturur. Değiştirdiğiniz bilgiler kulak misafirine karşı korunsa da, kiminle değiş tokuş edildiğinizi bilmenize gerek yok. Ağı denetlemezseniz, IP iletimlerinizi ya da uç nokta olarak maskeleyiş işlemi sırasında bir kişiye maruz kalmakta olduğunuz ortaya çıkar.

Anonim TLS 'yi destekleyen bir Diffie-Hellman anahtar değişimini CipherSpec olanağını kullanarak, sunucuyu doğrulamadan şifrelenmiş bir TLS bağlantısı oluşturabilirsiniz. İstemci ile sunucu arasında paylaşılan ve TLS iletimlerini şifrelemek için kullanılan ana güvenlik dizgisi, özel olarak imzalanmış bir sunucu sertifikası değiş tokuş olarak kurulur.

Anonim bağlantılar güvensiz olduğu için, çoğu TLS somutlaması anonim şifreleme Belirtileri 'ni kullanma varsayılan değeri değildir. Bir AMQP kanalı tarafından TLS bağlantısı için bir istemci isteği kabul edilirse, kanala geçiş tümceciği tarafından korunan bir anahtar deposu bulunmalıdır. Varsayılan olarak, TLS somutlamaları anonim CipherSpecs kullanmadığı için, anahtar deposunun istemcinin doğrulayabileceği özel olarak imzalanmış bir sertifika içermesi gerekir.

Anonim CipherSpecs kullanırsanız, sunucu anahtar deposu var olmalıdır, ancak özel olarak imzalanmış herhangi bir sertifika içermemelidir.

Şifrelenmiş bir bağlantı kurmanın başka bir yolu da, istemciye güven sağlayıcısının yerine kendi uygulamanızı koymanız. Güvenilir sağlayıcınız sunucu sertifikasının kimliğini doğrulamazdı, ancak bağlantı şifrelenir.

AMQP istemcilerinin TLS ile yapılandırılması

Ağ üzerinde akan verileri korumak ve istemcinin bağlı olduğu kuyruk yöneticisinin kimliğini doğrulamak için AMQP istemcilerini TLS 'yi kullanacak şekilde yapılandırabilirsiniz.

Bir AMQP istemcisinden bir AMQP kanalına bağlantı için TLS 'yi kullanmak üzere kuyruk yöneticisinin TLS 'ye yapılandırıldığından emin olmanız gerekir. [Kuyruk yöneticilerindeki TLS 'nin yapılandırılması](#) , kuyruk yöneticisinin TLS sertifikalarını okuduğu anahtar deposunun nasıl yapılandırılacağı açıklanır.

Kuyruk yöneticisi bir anahtar deposuyla yapılandırıldığında, istemcilerin bağlanacağı AMQP kanalının TLS özniteliklerini yapılandırmalısınız. AMQP kanallarında, TLS yapılandırmasıyla ilgili dört öznitelik vardır:

SSLCAUTH

SSLCAUTH özniteliği, kuyruk yöneticisinin kimliğini doğrulamak için bir istemci sertifikası sunmak üzere bir AMQP istemcisi gerektirip gerektirmeyeceğini belirtmek için kullanılır.

SSLCIPH

SSLCIPH özniteliği, kanalın TLS akışındaki verileri kodlamak için kullanması gereken şifreyi belirtir.

SSLPEER

SSLPEER özniteliği, bir bağlantı izin verilecekse, istemci sertifikasının eşleşmesi gereken ayırt edici adı (DN) belirtmek için kullanılır.

CERTLABEL

CERTLABEL, kuyruk yöneticisinin istemciye sunması gereken sertifikayı belirtir. Kuyruk yöneticisinin anahtar deposu birden çok sertifika içerebilir. Bu öznitelik, bu kanala bağlantılar için kullanılacak sertifikayı belirtmenize olanak sağlar. CERTLABEL belirlenmezse, kuyruk yöneticisi CERTLABEL özniteliğine karşılık gelen etiketle, kuyruk yöneticisi anahtar havuzunda bulunan sertifika kullanılır.

AMQP kanalınızı TLS öznitelikleriyle yapılandırdığınızda, aşağıdaki komutu kullanarak AMQP hizmetini yeniden başlatmalısınız:

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

Bir AMQP istemcisi, TLS tarafından korunan bir AMQP kanalına bağlandığında, istemci kuyruk yöneticisi tarafından sunulan sertifikaya ilişkin kimliği doğrular. Bunu yapmak için, AMQP istemcinizi kuyruk yöneticisinin sertifikasını içeren bir güvenli depo ile yapılandırmalısınız. Bunu yapma adımları,

kullanmakta olduğunuz AMQP istemcisine bağlı olarak değişir. Çeşitli AMQP istemcileri ve API 'leri hakkında bilgi almak için, ilgili AMQP istemci belgelerine bakın.

AMQP istemcilerinin kuyruk yöneticisinden çıkarılması

AMQP istemcilerinin kuyruk yöneticisinden bağlantısını kesmek istiyorsanız, PURGE CHANNEL komutunu çalıştırın ya da AMQP istemcisine olan bağlantıyı durdurun.

- **PURGE CHANNEL** komutunu çalıştırın. Örnek:

```
PURGE CHANNEL(MYAMQP) CLIENTID('recv_28dbb7e')
```

- Diğer bir seçenek olarak, AMQP istemcisinin istemciyi kesmek için kullandığı bağlantıyı durdurmak için aşağıdaki adımları izleyin:

1. Find the connection that the client is using by running the **DISPLAY CONN** command. Örnek:

```
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. Örnek:

```
STOP CONN(707E0A565F2D0020)
```

Çoklu yayını yönetme

Çok hedefli iletilerin boyutunu küçültür ve veri dönüştürmeyi etkinleştirme gibi IBM MQ Multicast denetim görevleri hakkında bilgi edinmek için bu bilgileri kullanın.

Çok noktaya yayın ile çalışmaya başlama

IBM MQ Multicast konuları ve iletişim bilgileri nesnelere çalışmaya başlamak için bu bilgileri kullanın.

Bu görev hakkında

IBM MQ Multicast ileti sistemi, konuları grup adreslerine eşleyerek ileti göndermek için ağı kullanır. Aşağıdaki görevler, gerekli IP adresinin ve kapının çoklu yayın ileti sistemi için doğru bir şekilde yapılandırılıp yapılandırılıp yapılandırılıp yapılandırılmadığını test etmek için hızlı bir yoldur.

Çok noktaya gönderim için COMMINFO nesnesi yaratılması

İletişim bilgileri (COMMINFO) nesnesi, çok hedefli iletimle ilişkili öznitelikleri içerir. COMMINFO nesne değiştirgeleriyle ilgili ek bilgi için [DEFINE COMMINFO](#) başlıklı konuya bakın.

Çok noktaya gönderim için bir COMMINFO nesnesi tanımlamak için aşağıdaki komut satırı örneğini kullanın:

```
DEFINE COMMINFO(MC1) GRPADDR(group address) PORT(port number)
```

Burada *MC1* , COMMINFO nesnesinin adı, *grup adresi* , grubunuzun çok hedefli IP adresi ya da DNS adı ve *kapı numarası* , iletimi için kullanılan kapıdır (varsayılan değer 1414 'tür).

MC1 adında yeni bir COMMINFO nesnesi yaratılır; bu ad, bir sonraki örnekte TOPIC nesnesi tanımlanırken belirtmeniz gereken addır.

Çoklu yayın için TOPIC nesnesi yaratılması

Konu, bir yayınlama/abone olma iletişimde yayınlanan bilgilerin konusu ve bir konu TOPIC nesnesi yaratılarak tanımlanmıştır. KONU NESNELERİNİN, çoklu yayınlama kullanılıp kullanılmayacağını tanımlayan iki parametre vardır. Bu değişirgeler şunlardır: **COMMINFO** ve **MCAST**.

- **COMMINFO** Bu parametre, çok hedefli iletişim bilgileri nesnesinin adını belirtir. COMMINFO nesne değişirgeleriyle ilgili ek bilgi için [DEFINE COMMINFO](#) başlıklı konuya bakın.
- **MCAST** Bu parametre, konu ağacındaki bu konumda çoklu yayının izin verilip verilmeyeceğini belirtir.

Çok noktaya gönderim için KONU nesnesi tanımlamak için aşağıdaki komut satırı örneğini kullanın:

```
DEFINE TOPIC(ALLSPORTS) TOPICSTR('Sports') COMMINFO(MC1) MCAST(ENABLED)
```

ALLSPOR adı verilen yeni bir KONU nesnesi yaratılır. It has a topic string *Spor*, its related communication information object is called *MC1* (which is the name you specified when defining a COMMINFO object in the previous example), and multicast is enabled.

Çok noktaya yayın yayınlama/abone olma testi

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 KONU nesnesinin konu dizgisidir ve *QM1* , kuyruk yöneticisinin adıdır.

3. Komut satırı 2 'de aşağıdaki komutu girin:

```
amqssubc Sports QM1
```

Burada *Spor* ve *QM1* , “2” sayfa 324 adımıyla kullanılanla aynıdır.

4. Komut satırı 1 'de Hello world girin. If the port and IP address that are specified in the COMMINFO object are configured correctly; the amqssubc sample, which is listening on the port for publications from the specified address, outputs Hello world at command line 2.

IBM MQ Multicast konu topolojisi

IBM MQ Multicast konu topolojisini anlamak için bu örneği kullanın.

IBM MQ Multicast desteği, her bir alt ağacın, toplam sıradüzen 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 şemada, olası iki çok noktaya gönderim veri akışı vardır:

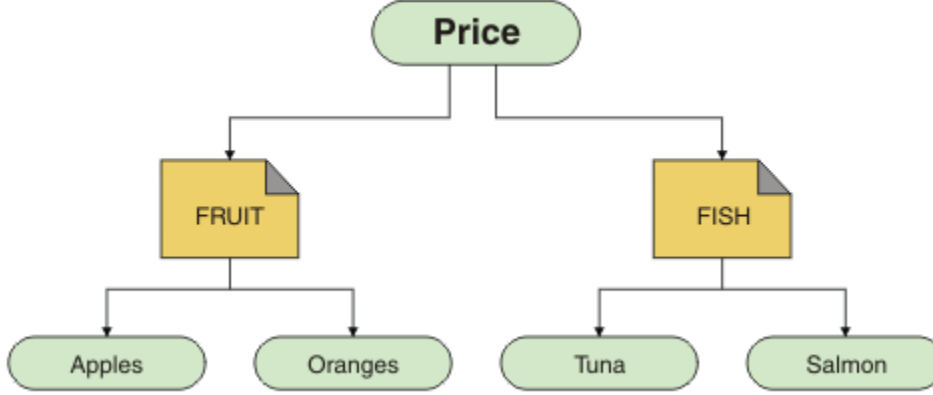
```
DEF COMMINFO(MC1) GRPADDR(239.XXX.XXX.XXX
)
```

```
DEF COMMINFO(MC2) GRPADDR(239.YYY.YYY.YYY)
```

Burada 239.XXX.XXX.XXX ve 239.YYY.YYY.YYY, geçerli çoklu yayın adresleridir.

Bu konu tanımları, aşağıdaki çizgede gösterildiği gibi bir konu ağacı oluşturmak için kullanılır:

```
DEFINE TOPIC(FRUIT) TOPICSTRING('Price/FRUIT') MCAST(ENABLED) COMMINFO(MC1)
DEFINE TOPIC(FISH) TOPICSTRING('Price/FISH') MCAST(ENABLED) COMMINFO(MC2)
```



Her bir çoklu yayın iletişim bilgileri (COMMINFO) nesnesi, grup adresleri farklı olduğu için farklı bir veri akışını temsil eder. In this example, the FRUIT topic is defined to use COMMINFO object MC1, the FISH topic is defined to use COMMINFO object MC2, and the Price node has no multicast definitions.

IBM MQ Multicast, konu dizgileri için 255 karakter sınırlaması içerir. Bu sınırlama, ağaç içindeki düğümlerin ve yaprak düğümlerin adları ile bakım yapılması gerektiği anlamına gelir; düğüm ve yaprak düğümlerin adları çok uzunsa, konu dizgisi 255 karakteri geçebilir ve [2425 \(0979\) \(RC2425\): MQRC_TOPIC_STRING_ERROR](#) neden kodunu döndürebilir. Daha uzun konu dizgilerinin başarım üzerinde zararlı etkisi olabileceği için, konu dizgilerinin mümkün olduğunca kısa olması önerilir.

Çok hedefli iletilerin boyutunu denetleme

IBM MQ ileti biçimi hakkında bilgi edinmek ve IBM MQ iletileri boyutunu azaltmak için bu bilgileri kullanın.

IBM MQ iletilerinde, ileti tanımlayıcısında yer alan özniteliklere ilişkin bir dizi öznitelik vardır. Küçük iletiler için, bu öznitelikler veri trafiğin çoğunu gösterebilir ve iletim hızı üzerinde önemli bir zarar verici etkiye sahip olabilir. IBM MQ Multicast, bu özniteliklerin herhangi biri ile birlikte iletilirse, kullanıcının yapılandırabilmesini sağlar.

İleti özniteliklerinin varlığı, konu dizgisinden farklı olarak, COMMINFO nesnesinin gönderilip gönderilmemesine bağlı olup olmadığına bağlıdır. Bir öznitelik iletilmezse, alan uygulaması varsayılan bir değer uygular. Varsayılan MQMD değerleri, MQMD_XX_ENCODE_CASE_ONE default değeriyle aynı olmak zorunda değildir ve [Çizelge 19 sayfa 326'](#) ta açıklanmıştır.

COMMINFO nesnesi, MQMD alanlarının ve kullanıcı özelliklerinin birçoğunun iletiliye nasıl akıp geçeceğini denetleyen MCPROP özniteliğini içerir. Bu özniteliğin değerini uygun bir düzeye ayarlayarak, IBM MQ Multicast iletilerinin boyutunu denetleyebilirsiniz:

MCPROP

Çok hedefli özellikler, MQMD özelliklerinin ve kullanıcı özelliklerinin kaç tanesi iletiliye akıp geçeceğini denetler.

TÜMÜ

Tüm kullanıcı özellikleri ve MQMD 'nin tüm alanları iletilir.

Yanıt

Yalnızca kullanıcı özellikleri ve iletileri yanıtlamak için kullanılan MQMD alanları iletilir. Bu özellikler şunlardır:

- MsgType
- MessageId
- CorrelId
- ReplyToQ
- ReplyToQmgr

KULLANICI

Yalnızca kullanıcı özellikleri iletilir.

YOK

Hiçbir kullanıcı özelliği ya da MQMD alanı iletilmedi.

ŞİRKET

Bu değer, iletinin iletiminin uyumlu bir kipte (RMM) gerçekleştirilmesine neden olur; bu kip, yürürlükteki XMS uygulamaları ve IBM Integration Bus RMM uygulamaları ile bazı işlemler arası işlemler yapılmasına olanak tanır.

V 9.2.0 **V 9.2.0** XMS .NET Çok hedefli ileti alışverişi (RMMkullanılarak) IBM MQ 9.2 ' den kullanımdan kaldırılmıştır ve ilerideki bir XMS .NETyayınında kaldırılacaktır.

Çok hedefli ileti öznitelikleri

İleti öznitelikleri, MQMD gibi çeşitli yerlerden (örneğin, MQRFH2) ve ileti özelliklerindeki alanlara gelebilir.

Aşağıdaki çizelge, iletiler MCPROP değerine (bu bölümde daha önce açıklanan) tabi olarak gönderildiğinde ve bir öznitelik gönderilmezse kullanılan varsayılan değer olarak gönderildiğinde ne olacağını gösterir.

<i>Çizelge 19. İleti alışverişi öznitelikleri ve bunların çok hedefli olarak nasıl ilişkilendirildikleri</i>		
Öznitelik	Çoklu yayın kullanan işlem	Varsayılan değer iletilmezse
TopicString	Her Zaman Dahil	Burada geçerli değil
MQMQ StrucId	İletilmedi	Burada geçerli değil
MQMD Sürümü	İletilmedi	Burada geçerli değil
Rapor	Varsayılan değilse dahil edilir	0
MsgType	Varsayılan değilse dahil edilir	MQMT_DATAGRAM
Son kullanma tarihi	Varsayılan değilse dahil edilir	0
Geribildirim	Varsayılan değilse dahil edilir	0
Kodlama	Varsayılan değilse dahil edilir	MQENC_NORMAL (equiv)
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_PERSISTENT
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
ReplyToQMgr	Varsayılan değilse dahil edilir	Boş
UserIdentifier	Varsayılan değilse dahil edilir	Boş

Çizelge 19. İleti alışverişi öznelikleri ve bunların çok hedefli olarak nasıl ilişkilendirildikleri (devamı var)

Öznitelik	Çoklu yayın kullanan işlem	Varsayılan değer iletilmezse
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	Burada geçerli değil
MsgSeqNumarası	Çıkarıldı	Burada geçerli değil
Görelî Konum	Çıkarıldı	Burada geçerli değil
MsgFlags	Çıkarıldı	Burada geçerli değil
OriginalLength	Çıkarıldı	Burada geçerli değil
UserProperties	Dahil	Burada geçerli değil

İlgili başvurular

[ALTER KOMUT](#)

[DEFINE KOMUTU](#)

Multicast ileti sistemi için veri dönüştürmenin etkinleştirilmesi

Use this information to understand how data conversion works for IBM MQ Multicast messaging.

IBM MQ Multicast paylaşılan, bağlantısız bir iletişim kuralıdır ve bu nedenle her istemcinin veri dönüştürmesi için belirli istekler hazırlamaması mümkün değildir. Aynı çoklu yayın akışına abone olan her istemci aynı ikili verileri alır; bu nedenle, IBM MQ veri dönüştürme gerekiyorsa, dönüştürme her istemcide yerel olarak gerçekleştirilir.

Karma bir platform kuruluşunda, istemcilerin büyük bölümü, verileri iletme uygulamasının yerel biçimi olmayan bir biçimde gerektirmektedir. Bu durumda, multicast COMMINFO nesnesinin **CCSID** ve **ENCODING** değerleri, verimlilik için ileti iletiminin kodlamasını tanımlamak için kullanılabilir.

IBM MQ Multicast, aşağıdaki yerleşik biçimlere ilişkin ileti bilgi yükünün veri dönüştürmesini destekler:

- MQADMIN
- MQOLAY
- MQPCF
- MQRFH
- MQRFH2
- MQSTR

Bu biçimlere ek olarak, kendi biçimlerinizi de tanımlayabilir ve bir [MQDXP-Data-conversion exit parameter](#) veri dönüştürme çıkışı kullanabilirsiniz.

Veri dönüştürme programlarıyla ilgili bilgi için bakınız: [Data conversion in the MQI for Multicast Messaging](#).

Veri dönüştürme hakkında daha fazla bilgi için bkz. [Veri dönüştürme](#).

Veri dönüştürme çıkışları ve ClientExitPathile ilgili daha fazla bilgi için, istemci yapılandırma dosyasının [ClientExitYol](#) kısmına bakın.

Çok hedefli uygulama izleme

IBM MQ Multicast 'ı yönetme ve izleme hakkında bilgi edinmek için bu bilgileri kullanın.

Çok hedefli trafiğe ilişkin yürürlükteki yayıncıların ve abonelerin durumu (örneğin, gönderilen ve alınan ileti sayısı ya da kaybedilen ileti sayısı), istemciden düzenli olarak sunucuya iletilir. Durum alındığında, COMMINFO nesnesinin **COMMEV** özniteliği, kuyruk yöneticisinin SYSTEM.ADMIN.PUBSUB.EVENT. Olay iletisine, alınan durum bilgileri yer alır. Bu bilgiler, bir sorunun kaynağını bulmada çok değerli bir tanılama yardımı sağlar.

Kuyruk yöneticisine bağlı uygulamalara ilişkin bağlantı bilgilerini görüntülemek için **DISPLAY CONN** MQSC komutunu kullanın. **DISPLAY CONN** komutuna ilişkin ek bilgi için [DISPLAY CONN](#) konusuna bakın.

Yayıncıların ve abonelerin durumunu görüntülemek için MQSC komutunu **DISPLAY TPSTATUS** 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 yayıncılarının ve abonelerinin izlenmesinde önemli bir öğedir. The reliability indicator (the **MSGREL** field that is returned on the Publish or Subscribe status commands) is an IBM MQ indicator that illustrates the percentage of transmissions that have no errors. Sometimes messages have to be retransmitted due to a transmission error, which is reflected in the value of **MSGREL**. İletim hatalarının olası nedenleri arasında yavaş aboneler, yoğun ağlar ve ağ kesintileri yer alır. **COMMEV**, COMMINFO nesnesi kullanılarak yaratılan çoklu yayın tanıtıcıları için olay iletilerinin oluşturulup oluşturulmayacağını ve olası üç değerden birine ayarlanıp ayarlanmayacağını denetler:

DEVRE DIŞI

Olay iletileri yazılmaz.

Etkinleştirildi

Olay iletileri her zaman, COMMINFO **MONINT** değiştirgesinde tanımlı bir sıklıkla yazılır.

KURAL DIŞI DURUM

İleti güvenilirliği güvenilirlik eşliğinin altındaysa olay iletileri yazılır. %90 ya da daha az bir ileti güvenilirlik düzeyi, ağ yapılandırmasıyla ilgili bir sorun olabileceğini ya da Yayınlama/Abone Olma uygulamalarının bir ya da daha fazlasının çok yavaş çalıştığını gösterir:

- **MSGREL (100, 100)** değeri, kısa vadede ya da uzun süreli zaman diliminde herhangi bir sorun olmadığını gösterir.
- **MSGREL (80, 60)** değeri, iletilerin %20 'nin şu anda sorun yaşamakta olduğunu, ancak 60 'ın uzun vadeli değeri üzerinde bir gelişme olduğunu belirtir.

İstemciler, kuyruk yöneticisine tek hedefli bağlantı bozulduğunda bile çoklu yayın trafiğini iletmeye ve almaya devam edebilir; bu nedenle veriler güncel olmayabilir.

Çok hedefli ileti güvenilirliği

IBM MQ Multicast aboneliğinin ve ileti geçmişinin nasıl ayarlanacağını öğrenmek için bu bilgileri kullanın.

Çok hedefli iletim hatasının çok önemli bir öğesi, iletilen verilerin arabelleğe alınmasını (bağlantının kesilen ucunda tutulacak iletilerin bir geçmişi) IBM MQ ile olur. This process means that no buffering of messages is required in the putting application process because IBM MQ provides the reliability. Bu tarihin boyutu, aşağıdaki bilgilerde anlatıldığı gibi, iletişim bilgileri (COMMINFO) nesnesi aracılığıyla yapılandırılır. Daha büyük bir iletim arabelleği, gerekirse yeniden iletilecek daha fazla iletim geçmişi olduğu anlamına gelir, ancak çoklu yayının niteliği nedeniyle %100 güvenli teslim desteklenemez.

IBM MQ Multicast ileti geçmişi, iletişim bilgileri (COMMINFO) nesnesinde **MSGHIST** özniteliğe göre denetlenir:

MSGHIST

Bu değer, NACK ' lar (negatif onaylar) durumunda, yeniden iletimleri işlemek için sistem tarafından tutulan, kilobayt cinsinden ileti geçmişi miktarıdır.

0 değeri, en az güvenilirlik düzeyini verir. Varsayılan değer 100 KB 'dir.

The IBM MQ Multicast new subscription history is controlled in the communication information (COMMINFO) object by the **NSUBHIST** attribute:

NSUBHIST

Yeni abone geçmişi, bir yayın akışına katılan bir abonenin şu anda kullanılabilir olduğu kadar veri alıp almadığını ya da yalnızca abonelik süresinden yapılan yayınları aldığını denetler.

YOK

NONE değeri, vericinin yalnızca abonelik zamanından yapılan yayını iletmesine neden olur. Varsayılan değer, 'NONE ' değeridir.

TÜMÜ

ALL değeri, vericinin, bilindiği gibi konunun çok geçmiş tarihini yeniden aktarmasına neden olur. bazı durumlarda bu durum, alıkonan yayınlara da benzer bir davranış verebilir.

Not: Büyük bir konu geçmişi varsa, ALL değerinin kullanılması, tüm konu geçmişinin yeniden iletilmesinden dolayı büyük bir konu geçmişi, başarımlar üzerinde zarar verebilir.

İlgili başvurular

DEFINE KOMUTU

ALTER KOMUT

Gelişmiş çok noktaya gönderim görevleri

.ini dosyalarını yapılandırma ve IBM MQ LLM ile birlikte çalışabilirlik gibi gelişmiş IBM MQ çoklu yayın yönetimi görevleri hakkında bilgi edinmek için bu bilgileri kullanın.

Multicast kuruluşunda güvenlik açısından dikkat edilmesi gereken noktalar için bkz. [Multicast security](#).

Çok hedefli ve çok hedefli yayınlama/abone olma etki alanları arasında köprü oluşturma

Çok hedefli olmayan bir yayıncı IBM MQ Multicast etkinleştirilmiş bir konuya yayınlandığında ne olacağını anlamak için bu bilgileri kullanın.

Çok hedefli olmayan bir yayıncı, **MCAST** etkin ve **BRIDGE** etkin olarak tanımlanmış bir konuya yayınlarsa, kuyruk yöneticisi iletiyi çok hedefli olarak doğrudan dinlemiş olabilecek tüm abonelere iletir. Çok noktaya yayın yapan bir yayıncı, çoklu yayın özelliğinin etkinleştirilmemiş konulara yayınlamaz.

Var olan konular, bir konu nesnesinin **MCAST** ve **COMMINFO** parametreleri ayarlanarak çoklu yayın özelliği etkinleştirilebilir. Bu parametrelerle ilgili ek bilgi için [İlk çoklu yayın kavramları](#) başlıklı konuya bakın.

COMMINFO nesnesi **BRIDGE** özneliği, çok hedefli uygulamalar kullanmayan uygulamalardan yayınlar denetler. **BRIDGE** ENABETLE olarak ayarlanırsa ve konunun **MCAST** parametresi de ETKIN olarak ayarlandıysa, çoklu yayın kullanmayan uygulamalardan gelen yayınlar, yapılan uygulamalara köprülülük olarak ayarlanır. **BRIDGE** parametresine ilişkin ek bilgi için [DEFINE COMBILGIS](#) başlıklı konuya bakın.

Çoklu yayın için .ini dosyaları yapılandırılıyor

.ini dosyalarındaki IBM MQ Multicast alanlarını anlamak için bu bilgileri kullanın.

Ek IBM MQ Multicast yapılandırması bir ini dosyasında yapılabilir. Kullanmanız gereken belirli bir ini dosyası, uygulama tipine bağlıdır:

- İstemci: `MQ_DATA_PATH /mqclient.ini` dosyasını yapılandırma.
- Kuyruk yöneticisi: `MQ_DATA_PATH /qmgrs/QMNAME/qm.ini` dosyasını yapılandırma.

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 Multicast 'un davranışını ince ayarlamaya 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 iletişim kuralı kullanılarak gönderilir. Ancak, ağ öğeleri, IP kipinde olduğu gibi 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 destek desteği olan ağ öğeleri, güvenilir çoklu yayın paketi dağıtımında yardımcı olur. Bu kip, PGM standardına tam olarak uyumludur.

IPVersion

IPv4

Yalnızca IPv4 iletişim kuralını kullanarak iletişim kurun. Bu varsayılan değerdir.

IPv6

Yalnızca IPv6 iletişim kuralını kullanarak iletişim kurun.

HERHANGİ BİRİ

Communicate using IPv4, IPv6, or both, depending on which protocol is available.

her ikisi

Hem IPv4 hem de IPv6 kullanılarak iletişimi destekler.

LimitTransOranı

DEVRE DIŞI

Herhangi bir iletim hızı denetimi yoktur. Bu varsayılan değerdir.

STATIC

Statik iletim hızı denetimini gerçekleştirir. Verici, TransRateLimit parametresi tarafından belirlenen oranı aşan bir hızda iletilemedi.

Devingen

Verici iletim hızını alıcılardan aldığı geri bildirimlere göre uyarlar. Bu durumda, iletim hızı sınırı, TransRateLimit parametresi tarafından belirtilen değerden daha fazla olamaz. Verici, optimum iletim hızına ulaşmaya çalışır.

TransRateSınırı

Kb/s cinsinden iletim hızı sınırı.

SocketTTL

SocketTTL değeri, çoklu yayın trafiğin bir yönlendirici üzerinden geçebileceğini ya da geçebileceği yönlendiricilerin sayısını belirler.

Toplu

İletilerin toplu mı, yoksa hemen mi gönderileceğini denetler: 2 olası değer vardır:

- *NO* (Hayır) İletiler toplu olarak gönderilmez, hemen gönderilir.
- *YES* (EVET) İleti vuruşlanmış.

Döngü

Çoklu yayın döngülerini etkinleştirmek için değeri 1 olarak ayarlayın. Çoklu yayın döngüsü, gönderilen verilerin anasisteme geri gönderilip geri gönderilmediğini tanımlar.

Arabirim

Çok hedefli trafik akışlarının üzerindeki arabirimin IP adresi. Daha fazla bilgi ve sorun giderme için bkz. [Çok noktaya yayın olmayan bir ağ üzerinde çoklu yayın uygulamalarının sınanması ve Çok noktaya gönderim trafiği için uygun ağı ayarlanması](#)

FeedbackMode

NACK

Olumsuz onaylarla geri bildirim. Bu varsayılan değerdir.

ACK

Olumlu kabul görerek geri bildirim.

WAIT1

Vericinin, alıcıların herhangi birinden yalnızca 1 adet ACK ' yi beklediği olumlu kabul görerek geri bildirim.

HeartbeatTimeout

Sağlıklı işletim bildirim zaman aşımı (milisaniye). 0 değeri, sağlıklı işletim bildirim olaylarının alıcı ya da konu alıcıları tarafından yükseltilmediğini belirtir. Varsayılan değer 20000 'dir.

HeartbeatInterval

Sağlıklı işletim bildirim aralığı (milisaniye). 0 değeri, herhangi bir kalp atışlarının gönderilmemesinin gönderildiğini gösterir. Sağlıklı işletim bildirim zaman aşımı olaylarını önlemek için sağlıklı işletim bildirim aralığı, **HeartbeatTimeout** değerinden büyük ölçüde daha küçük olmalıdır. Varsayılan değer 2000 'dir.

IBM MQ Low Latency Messaging ile çoklu yayın birlikte çalışabilirlik

IBM MQ Multicast ve IBM MQ Low Latency Messaging (LLM) ile birlikte çalışabilirliği anlamak için bu bilgileri kullanın.

LLM kullanan bir uygulama için temel bilgi yükü aktarımı, iletileri her iki yönde de değiş tokuş etmek için çoklu yayın kullanan başka bir uygulama ile mümkündür. Çoklu yayın, LLM teknolojisini kullansa da, LLM ürününün kendisi yerleşik değildir. Therefore it is possible to install both LLM and IBM MQ Multicast, and operate and service the two products separately.

Çoklu yayınlı iletişim kuran LM uygulamalarının ileti özelliklerini gönderip alması gerekebilir. IBM MQ ileti özellikleri ve MQMD alanları, aşağıdaki tabloda gösterildiği gibi, belirli LLM ileti özelliği kodlarına sahip LLM ileti özellikleri olarak iletilir:

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
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

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.ApplOriginData	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1022
MQPubOptions	RMM_MSG_PROP_INT32	LLM_PROP_KIND_int32	-1053

LLM hakkında daha fazla bilgi için, bkz. LLM ürün belgelerine bakın: [IBM MQ Low Latency Messaging](#).

IBM i Yönetme IBM MQ for IBM i

IBM üzerinde IBM MQ ' i yönetmek için kullanabileceğiniz yöntemleri tanıtır.

Yönetim görevleri; kümeleri, işlemleri ve IBM MQ nesnelere (kuyruk yöneticileri, kuyruklar, ad listeleri, süreç tanımlamaları, kanallar, istemci bağlantı kanalları, dinleyiciler, hizmetler ve kimlik doğrulama bilgileri nesnelere) yaratma, başlatma, değiştirme, görüntüleme, durdurma ve silme işlemleri içerir.

IBM MQ for IBM i ' i nasıl yönetebilmeye ilişkin ayrıntılar için aşağıdaki bağlantılara bakın:

- [“CL komutlarını kullanarak IBM MQ for IBM i ' in yönetilmesi” sayfa 332](#)
- [“IBM MQ for IBM i yönetimine ilişkin diğer yöntemler” sayfa 346](#)
- [“Work management for IBM i” sayfa 351](#)

İlgili kavramlar

[“Availability, backup, recovery, and restart on IBM i” sayfa 358](#)

IBM MQ for IBM i ' in yedekleme ve geri yükleme stratejisine yardımcı olması için IBM i günlük kaydı desteğini nasıl kullandığını anlamak için bu bilgileri kullanın.

[IBM MQ for IBM i kuyruk yöneticisi kitaplık adlarının anlaşılması](#)

İlgili görevler

[IBM üzerindeki yapılandırma bilgilerinin değiştirilmesi](#)

[IBM üzerinde güvenliğin ayarlanması](#)

İlgili başvurular

[“IBM MQ for IBM i susturulması” sayfa 401](#)

Bu bölümde, 'quiesce' un nasıl susturacağını açıklar. IBM MQ for IBM i.

[“IBM i ' ta ölü-harflı kuyruk işleyicisi” sayfa 144](#)

IBM i ' in ölme harflı kuyruk işleyicisi nedir ve bunu nasıl başlatıyorsunuz?

[Determining problems with IBM MQ for IBM i applications](#)

[IBM üzerindeki kurulabilir hizmetler ve bileşenler](#)

[Sistem ve varsayılan nesnelere](#)

IBM i CL komutlarını kullanarak IBM MQ for IBM i ' in yönetilmesi

IBM MQ IBM i komutlarını anlamak için bu bilgileri kullanın.

Kuyruk yöneticileri, kuyruklar, konular, kanallar, ad listeleri, süreç tanımları ve kimlik doğrulama bilgileri nesnelereyle ilişkili olanlar da dahil olmak üzere, çoğu IBM MQ komutu, ilgili **WRK*** komutu kullanılarak erişilebilir.

Kümedeki asıl komut şudur: **WRKMQM**. Bu komut, örneğin, sistemdeki tüm kuyruk yöneticilerinin bir listesini, durum bilgileriyle birlikte görüntülemek için size olanak sağlar. Diğer bir seçenek olarak, her bir girdiye karşı çeşitli seçenekler kullanarak, tüm kuyruk yöneticisine özgü komutları da işleyebilirsiniz.

WRKMQM komutundan, her kuyruk yöneticisinin belirli alanlarını seçebilirsiniz; örneğin, kanallarla, konularla ya da kuyruklarla çalışmak ve oradan tek tek nesnelere seçilebilir.

IBM MQ uygulama tanımlarını kaydetme

IBM MQ uygulamalarını yarattığınızda ya da uyarladığınızda, yaratılan tüm IBM MQ tanımlarının bir kaydını tutmak yararlı olur. Bu kayıt şu öge için kullanılabilir:

- Kurtarma amacı
- Bakım
- IBM MQ uygulamalarını kaydetme

IBM MQ uygulama tanımlarını 1/2 şekilde kaydedebilirsiniz:

1. Sunucu için IBM MQ tanımlarınızı oluşturmak üzere CL programları oluşturma.
2. Çapraz platform IBM MQ komut dilini kullanarak IBM MQ tanımlarınızı oluşturmak için SRC üyeleri olarak MQSC metin dosyaları oluşturuluyor.

Kuyruk nesnelerinin tanımlamaya ilişkin daha fazla ayrıntı için bkz. [“MQSC komutlarını kullanan yönetim” sayfa 11](#) ve [“IBM MQ Programları Komut Biçimlerinin Kullanılması” sayfa 24](#).

İlgili başvurular

[IBM MQ for IBM i CL komutları başvurusu](#)

Before you start using the IBM MQ for IBM i using CL commands

IBM MQ altsistemini başlatmak ve bir yerel kuyruk yöneticisi yaratmak için bu bilgileri kullanın.

Başlamadan önce

Ensure that the IBM MQ subsystem is running (using the command STRSBS QMQM/QMQM), and that the job queue associated with that subsystem is not held. Varsayılan olarak, IBM MQ altsistemi ve iş kuyruğunun her ikisi de QMQM kitaplığındaki QMQM adını taşır.

Bu görev hakkında

Kuyruk yöneticisi başlatmak için IBM i komut satırını kullanma

Yordam

1. Bir IBM i komut satırından CRTMQM komutunu kullanarak yerel bir kuyruk yöneticisi yaratın.
Bir kuyruk yöneticisi yarattığınızda, bu kuyruk yöneticisini varsayılan kuyruk yöneticisi yapma seçeneğiniz vardır. Kuyruk yöneticisi adı değiştirgesi (MQMNAME) atlanırsa, varsayılan kuyruk yöneticisi (yalnızca bir tane olabilir), bir CL komutunun uygulanacağı kuyruk yöneticidir.
2. Bir IBM i komut satırından STRMQM komutunu kullanarak yerel kuyruk yöneticisini başlatın.
If the queue manager startup takes more than a few seconds IBM MQ will show status messages intermittently detailing the startup progress. Bu iletilerle ilgili ek bilgi için [İletiler ve neden kodları](#) başlıklı konuya bakın.

Sonraki adım

You can stop a queue manager by issuing the ENDMQM command from the IBM i command line, and control a queue manager by issuing other IBM MQ commands from an IBM i command line.

Uzak kuyruk yöneticileri uzaktan başlatılamaz, ancak yerel işletmenler tarafından sistemlerinde yaratılıp başlatılmalıdır. Bu tür işlemleri etkinleştirmek için uzak işletim tesislerinin (IBM MQ for IBM dışında) olduğu bir kural dışı durum.

Yerel kuyruk yöneticisi uzak kuyruk yöneticisini durduramıyor.

Not: Bir IBM MQ sistemini susturmanın bir parçası olarak, etkin kuyruk yöneticilerini susturmamanız gerekir. Bu, [“IBM MQ for IBM isusturulması” sayfa 401](#) içinde açıklanmaktadır.

IBM için IBM MQ nesneleri oluşturmaya ilişkin yöntemleri anlamak için bu bilgileri kullanın.

Başlamadan önce

Aşağıdaki görevler, komut satırından IBM MQ for IBM i ' u kullanabilmeniz için çeşitli yollar önermektedir.

Bu görev hakkında

IBM MQ nesneleri yaratmak için iki çevrimiçi yöntem vardır; bunlar şunlardır:

Yordam

1. Create komutunu kullanarak, örneğin: **Create MQM Queue** komutu: **CRTMQMQ**
2. Using a Work with MQM object command, followed by F6, for example: The **Work with MQM Queues** command: **WRKMQMQ**

Sonraki adım

Tüm komutların bir listesi için bkz. [IBM MQ for IBM i CL komutları](#).

Not: Tüm MQM komutları Message Queue Manager Commands (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 panosu**' ta, Queue name alanında yaratmak istediğiniz kuyruğun adını yazın. Karışık bir vaka adı belirtmek için, adı tırnak içine alınız.
3. Queue type alanına *LCL yazın.
4. Varsayılan kuyruk yöneticisini kullanmıyorsanız, bir kuyruk yöneticisi adı belirtin ve Enter tuşuna basın. Yeni bir değer ile değerlerin herhangi birini geçersiz kılabilir. Diğer alanları görmek için ileriye kaydırın. Kümeler için kullanılan seçenekler, seçenekler listesinin sonunda yer alıyor.
5. Herhangi bir değeri değiştirdiğinizde, kuyruğu yaratmak için Enter tuşuna basın.

WRKMQMQ komutu kullanılarak yerel kuyruk yaratılması

Yordam

1. Komut satırına WRKMQMQ yazın.
2. Kuyruk yöneticisinin adını girin.
3. Bilgi istemi panelini görüntülemek istiyorsanız, F4 tuşuna basın. Bilgi istemi panosu, soysal bir kuyruk adı ya da kuyruk tipi belirterek, görüntülenen kuyrukların sayısını azaltmak için yararlıdır.
4. Enter tuşuna basın ve **MQM Kuyruklarıyla Çalışma panosu** görüntülenir. Değerlerin herhangi birini yeni bir değerle geçersiz kılabilir. Diğer alanları görmek için ileriye kaydırın. Kümeler için kullanılan seçenekler, seçenekler listesinin sonunda yer alıyor.
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 334 . Kuyruğu yarattığınız zaman, **MQM Kuyruklarıyla Çalışma panosu** yeniden görüntülenir. F5=Refresh tuşuna bastığınızda listeye yeni kuyruk eklenir.

Kuyruk yöneticisi özniteliklerinin değiştirilmesi

Bu görev hakkında

CHGMQM komutunda belirlenen kuyruk yöneticisinin özniteliklerini değiştirmek için, değiştirmek istediğiniz öznitelikleri ve değerleri belirtin. Örneğin, `jupiter.queue.manager` özniteliklerini değiştirmek için aşağıdaki seçenekleri kullanın:

Yordam

Komut satırına **CHGMQM** yazın ve F4 tuşuna basın.

Sonuçlar

Komut, kullanılmayan ileti kuyruğunu değiştirir ve olayların engellenmesini sağlar.

IBM i IBM üzerindeki yerel kuyruklarla çalışma

Bu kısım, yerel kuyrukları yönetmek için kullanabileceğiniz bazı komutlara ilişkin örnekleri içerir. All the commands shown are also available using options from the **WRKMQM komut panosu**.

Yerel kuyruk tanımlanması

Bir uygulama için, yerel kuyruk yöneticisi, uygulamanın bağlı olduğu kuyruk yöneticidir. Yerel kuyruk yöneticisi tarafından yönetilen kuyruklar, o kuyruk yöneticisinin yerel olduğu söylenmektedir.

Yerel bir kuyruğun tanımlanmasını yaratmak ve kuyruk adı verilen veri yapısını yaratmak için **CRTMQMQ QTYPE *LCL** komutunu kullanın. Ayrıca, kuyruk özelliklerini varsayılan yerel kuyruğunlardan da değiştirebilirsiniz.

Bu örnekte, `orange.local.queue` olarak tanımladığımız kuyruk, aşağıdaki özelliklere sahip olacak şekilde belirtilir:

- Bir ilk çıkış (FIFO) temelinde işlev görerek, bu olanak için geçersiz kılınır, devre dışı bırakılır ve çalışır.
- Bu bir *olağan* kuyruğdur; yani, bir başlatma kuyruğu ya da iletim kuyruğu değil ve tetikleyici iletileri oluşturmaz.
- Kuyruk derinliği üst sınırı 1000 iletilerdir; ileti uzunluğu üst sınırı 2000 byte 'tır.

Aşağıdaki komut varsayılan kuyruk yöneticisinde 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 kuyruğun bir iletim kuyruğu olmadığını belirtir.
2. Aynı kuyruk yöneticisinde `orange.local.queue` adlı bir yerel kuyruğunuz varsa, bu komut başarısız olur. Bir kuyruğun var olan tanımlanmasının üzerine yazmak istiyorsanız REPLACE *YES özniteliğini kullanın, ancak aynı zamanda bkz. [“Yerel kuyruk özniteliklerinin değiştirilmesi” sayfa 336.](#)

Ölü-mektup kuyruğu tanımlanması

Her kuyruk yöneticisinin, doğru hedeflerine teslim edilemeyen iletilerin daha sonra alınması için saklanabilmesi için, her kuyruk yöneticisinin bir ölü-mektup kuyruğu olarak kullanılmak üzere bir yerel kuyruğu olması gerekir. Kuyruk yöneticisine, ölü-mektup kuyruğu hakkında açık bir şekilde bilgi vermelisiniz. Bunu, **CRTMQM** komutunda bir ölü harf kuyruğu belirterek yapabilir ya da **CHGMQM** komutunu kullanarak daha sonra bir komut belirleyebilirsiniz. Ayrıca, kullanılmadan önce ölü-mektup kuyruğunu da tanımlamanız gerekir.

Ürünle birlikte SYSTEM . DEAD . LETTER . QUEUE adlı bir örnek ölü mektup kuyruğu ürünle birlikte sağlanır. Kuyruk yöneticisi yarattığınızda bu kuyruk otomatik olarak yaratılır. Gerekirse bu tanımlamayı değiştirebilirsiniz. Bunu yeniden adlandırmaya gerek yoktur, ancak beğendiyseniz de, bunu yeniden adlandırmanız gerekir.

Bir ölü-mektup kuyruğunda aşağıdakiler dışında özel bir gereksinim yoktur:

- Yerel bir kuyruk olmalıdır.
- MAXMSGL (ileti uzunluğu üst sınırı) özniteliği, kuyruk yöneticisinin "ölü-harflik" üstbilgisinin (MQDLH) büyüklüğünü işlemek için kuyruk yöneticisinin **arti** işlemesi gereken en büyük iletileri barındıracak şekilde kuyruğu etkinleştirmesi gerekir.

IBM MQ , bir ölü-mektup kuyruğunda bulunan iletilerin nasıl işleneceğini ya da kaldırılacağı hakkında bilgi vermenizi sağlayan, çıkmaz bir kuyruk işleyicisi sağlar. Daha fazla bilgi için bkz. ["IBM i' ta ölü-harfli kuyruk işleyicisi" sayfa 144.](#)

Varsayılan nesne özniteliklerinin görüntülenmesi

Bir IBM MQ nesnesi tanımladığınızda, bu nesne varsayılan nesneden belirtmediğiniz öznitelikleri alır. Örneğin, bir yerel kuyruk tanımladığınızda, kuyruk, tanımda çıkardığınız tüm öznitelikleri varsayılan yerel kuyruktan (SYSTEM . DEFAULT . LOCAL . QUEUE adı verilen) 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)
```

This command creates a queue with the same attributes as our original queue `orange.local.queue`, rather than those of the system default local queue.

Ayrıca, bir kuyruk tanımını kopyalamak için **CPYMQMQ** komutunu kullanabilir, ancak özgün özniteliklerde bir ya da daha fazla değişiklik yerine geçer. Örneğin:

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('third.queue') MQMNAME(MYQUEUEMANAGER)  
MAXMSGL(1024)
```

This command copies the attributes of the queue `orange.local.queue` to the queue `third.queue`, but specifies that the maximum message length on the new queue is to be 1024 bytes, rather than 2000.

Not: **CPYMQMQ** komutunu kullandığınızda, kuyruktaki iletileri değil, yalnızca kuyruk özniteliklerini kopyaladınız.

Yerel kuyruk özniteliklerinin değiştirilmesi

You can change queue attributes in two ways, using either the **CHGMQM** command or the **CPYMQMQ** command with the REPLACE *YES attribute. In ["Yerel kuyruk tanımlanması" sayfa 335](#), you defined the

queue orange.local.queue. Örneğin, bu kuyruktaki ileti uzunluğu üst sınırını 10.000 byte 'a artırmamız gerekiyorsa.

- **CHGMQM** komutunu kullanarak:

```
CHGMQM QNAME('orange.local.queue') MQMNAME(MYQUEUEMANAGER) MAXMSGLEN(10000)
```

Bu komut, ileti uzunluğu üst sınırı olan tek bir özneteliği değiştirir; diğer tüm öznetelikler de aynı kalır.

- Using the **CRTMQM** command with the REPLACE *YES option, for example:

```
CRTMQM QNAME('orange.local.queue') QTYPE(*LCL) MQMNAME(MYQUEUEMANAGER)  
MAXMSGLEN(10000) REPLACE(*YES)
```

Bu komut yalnızca ileti uzunluğu üst sınırı değil, diğer tüm özneteliklerin varsayılan değerlerini de değiştirir. Kuyruk artık etkin yerleştirildi, oysa daha önce engellendi. Put enabled is the default, as specified by the queue SYSTEM.DEFAULT.LOCAL.QUEUE, unless you have changed it.

Varolan bir kuyruktaki ileti uzunluğu üst sınırını **azalt**, var olan iletiler bundan etkilenmez. Ancak yeni mesajların yeni kriterlere uygun olması gerekir.

Yerel kuyruğun temizlenmesi

To delete all the messages from a local queue called magenta.queue, use the following command:

```
CLRMQM QNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Şu durumda bir kuyruğu temizleyemezsiniz:

- Eşitleme noktası altında kuyruğa yerleştirdiğiniz kesinleştirilmemiş iletiler var.
- Şu anda bir uygulama kuyruğu açık durumda.

Yerel Kuyruğun Silinmesi

Yerel bir kuyruğu silmek için **DLTMQM** komutunu kullanın.

Kuyruğun üzerinde kesinleştirilmemiş iletiler varsa ya da kullanımda olduğunda kuyruk silinemez.

Büyük kuyrukların etkinleştirilmesi

IBM MQ, 2 GB 'den büyük kuyrukları destekler. See your operating system documentation for information on how to enable IBM i to support large files.

IBM i ürün bilgileri, [IBM Documentation](#) içinde bulunabilir.

Bazı yardımcı programlar 2 GB 'den büyük dosyalarla başa çıkamayabilirler. Büyük dosya desteğini etkinleştirmeden önce, bu tür bir desteğe ilişkin sınırlamalar olup olmadığını görmek için işletim sistemi belgelerinize bakın.

IBM i IBM üzerindeki diğer ad kuyruklarıyla çalışma

Bu bölümde, diğer ad kuyruklarını yönetmek için kullanabileceğiniz bazı komutlara ilişkin örnekler yer alır. All the commands shown are also available using options from the **WRKMQM** komut panosu.

Diğer ad kuyruğu (bazen kuyruk diğer adı olarak da bilinir), MQI çağrılarını yeniden yönlendiren bir yöntem sağlar. Diğer ad kuyruğu, gerçek bir kuyruk değil, gerçek bir kuyruğa çözülen bir tanımlamadır. Diğer ad kuyruğu tanımlaması, TGTQNAME özneteliği tarafından belirtilen bir hedef kuyruk adı içerir.

Bir uygulama bir MQI çağrısında diğer ad kuyruğunu belirtiyorsa, kuyruk yöneticisi gerçek kuyruk adını yürütme sırasında çözer.

For example, an application has been developed to put messages on a queue called `my.alias.queue`. Bu kuyruğun bir **MQOPEN** isteğini ve dolaylı olarak bu kuyruğa bir ileti koymasını durumunda, bu kuyruğun adını belirtir. Uygulama, kuyruğun bir diğer ad kuyruğu olduğunu dikkate almıyor. Bu diğer adı kullanan her bir MQI çağrısı için, kuyruk yöneticisi gerçek kuyruk adını çözer; bu ad yerel bir kuyruk ya da bu kuyruk yöneticisinde tanımlanmış uzak bir kuyruk olabilir.

TGTQNAME özniteliğinin değerini değiştirerek, büyük olasılıkla başka bir kuyruk yöneticisine ilişkin MQI çağrılarını başka bir kuyruğa yeniden yönlendirebilirsiniz. Bu, bakım, geçiş ve yük dengelemesi için kullanışlıdır.

Diğer ad kuyruğunun tanımlanması

Aşağıdaki komut bir diğer ad kuyruğu yaratır:

```
CRTMQMQ QNAME('my.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
MQMNAME(MYQUEUEMANAGER)
```

Bu komut, `my.alias.queue` ögesini `specifykuyruğuna (yellow.queue)` belirten MQI çağrılarını yeniden yönlendirir. The command does not create the target queue; the MQI calls fail if the queue `yellow.queue` does not exist at run time.

Diğer ad tanımını değiştirirseniz, MQI çağrılarını başka bir kuyruğa yeniden yönlendirebilirsiniz. Örneğin:

```
CHGMQM QNAME('my.alias.queue') TGTQNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Bu komut, MQI çağrılarını başka bir kuyruğa (`magenta.queue`) yeniden yönlendirir.

Ayrıca, başka uygulamalar için farklı özniteliklere sahip olmak üzere tek bir kuyruk (hedef kuyruk) yapmak için diğer ad kuyruklarını da kullanabilirsiniz. Bunu, her uygulama için bir tane olmak üzere iki diğer ad tanımlayarak yapabilirsiniz. İki uygulama olduğunu varsayın:

- Uygulama ALPHA, `yellow.queue` a ileti yerleştirebilir, ancak bundan ileti almanıza izin verilmez.
- Uygulama BETA, `yellow.queue` den ileti alabilir, ancak bu iletiye ileti koymasına izin verilmez.

Bunu, aşağıdaki komutları kullanarak yapabilirsiniz:

```
/* 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(MYQUEUEMANAGER)

/* 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(MYQUEUEMANAGER)
```

ALPHA uses the queue name `alphas.alias.queue` in its MQI calls; BETA uses the queue name `betas.alias.queue`. Her ikisi de aynı kuyruğa erişir, ama farklı şekillerde.

Diğer ad kuyruklarını tanımladığınızda, bu öznitelikleri yerel kuyruklarla aynı şekilde tanımladığınızda **REPLACE *YES** özniteliğini kullanabilirsiniz.

Diğer ad kuyruklarıyla diğer komutların 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(MYQUEUEMANAGER)

/* ALTER the base queue name, to which the alias resolves. */
/* FORCE = Force the change even if the queue is open. */
```

```
CHQMOMQ QNAME('alphas.alias.queue') TGTQNAME('orange.local.queue') FORCE(*YES)
MQMNAME(MYQUEUEMANAGER)
```

IBM i IBM üzerinde model kuyruklarıyla çalışma

Bu kısım, model kuyruklarını yönetmek için kullanabileceğiniz bazı komutlara ilişkin örnekler içerir. All the commands shown are also available using options from the **WRKMOMQ komut panosu**.

Kuyruk yöneticisi, bir uygulama tarafından, model kuyruğu olarak tanımlanmış bir kuyruk adı belirleyen bir MQI çağırısı alırsa, dinamik bir kuyruk yaratır. Kuyruk yaratıldığında, kuyruk yöneticisi yeni dinamik kuyruğun adını içerir. Model kuyruğu, bu şablondan yaratılan dinamik kuyrukların özniteliklerini belirleyen bir şablondur.

Model kuyrukları, uygulamalar için gerekli olan kuyruklar yaratmaları için uygun bir yöntem sağlar.

Model kuyruğu tanımlanması

Bir öznitelikler kümesiyle birlikte, bir yerel kuyruğu tanımladığınız şekilde bir model kuyruğu tanımladınız. Model kuyrukları ve yerel kuyruklar, yaratılan dinamik kuyrukların geçici mi kalıcı mı, yoksa kalıcı mı olduğunu belirleyebileceğiniz model kuyruklarında aynı öznitelikler kümesine sahiptir. (Kalıcı kuyruklar kuyruk yöneticisi yeniden başlatıldığında sürdürür, geçici olanlar da yoktur). 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, otomatik olarak `SYSTEM.DEFAULT.MODEL.QUEUE` varsayılan kuyruğundan kopyalanır.

Model kuyruklarını tanımladığınızda, bunları yerel kuyruklarla aynı şekilde tanımladığınızda, `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. */
CHGMOMQ MQMNAME(MYQUEUEMANAGER) QNAME('blue.model.queue') PUTENBL(*YES)
```

IBM i IBM üzerinde tetikleme ile çalışma

Tetikleme ve süreç tanımlamalarını öğrenmek için bu bilgileri kullanın.

IBM MQ , kuyruklardaki belirli koşullar karşılandığında otomatik olarak bir uygulama başlatmak için bir olanak sağlar. Bir kuyruktaki ileti sayısı belirli bir sayıya ulaştığında koşulların bir örneği de vardır. Bu olanak, *tetikleme* olarak adlandırılır ve [Tetikleme kanalları](#) içinde ayrıntılı bir şekilde açıklanır.

Tetikleme nedir?

Kuyruk yöneticisi, tetikleyici olayları olduğu gibi bazı koşulları tanımlar. Bir kuyruk için tetikleme etkinleştirilirse ve bir tetikleme olayı ortaya çıkarsa, kuyruk yöneticisi bir tetikleme iletisini başlatma kuyruğu adı verilen bir kuyruğa gönderir. Başlatma kuyruğunda tetikleme iletisinin varlığı, bir tetikleme olayının ortaya çıktığını gösterir.

Kuyruk yöneticisi tarafından oluşturulan tetikleme iletileri kalıcı değil. Bu, günlüğe kaydetmeyi azaltmanın (bu şekilde başarımı yükselmesi) ve yeniden başlatma sırasında yinelemeleri en aza indirmenin etkisine sahiptir. Bu nedenle, yeniden başlatma süresini iyileştirin.

Tetikleme monitörü nedir?

Başlatma kuyruğunu işleyen program, tetikleme izleme uygulaması olarak adlandırılır ve tetikleme iletilerinde yer alan bilgilere dayalı olarak, tetikleme iletilerini okumanız ve uygun işlemi yapmak gerekir. Olağan durumda bu işlem, tetikleme iletilerinin oluşturulmasına neden olan kuyruğun işlenmesi için başka bir uygulama başlatmış olabilir. Kuyruk yöneticisinin bakış açısından, tetikleme izleme uygulamasıyla ilgili özel bir şey yoktur; bu, bir kuyruktan ileti okuyan başka bir uygulamadır (başlangıç kuyruğu).

Tetikleme izleyicisinin iş gönderme özniteliklerinin değiştirilmesi

The trigger monitor supplied as command **STRMQMTRM** submits a job for each trigger message using the system default job description, QDFTJOB. Sunulan işlerin her zaman QDFTJOB adı verilen ve kitaplık listesi (*SYSVAL) de içinde olmak üzere varsayılan iş tanımının özniteliklerine sahip olması nedeniyle bu, sınırlamalara sahiptir. IBM MQ, bu öznitelikleri geçersiz kılmak için bir yöntem sağlar. Örneğin, gönderilen işlerin daha anlamlı iş adlarına sahip olması için aşağıdaki gibi uyarlanması mümkündür:

1. İş tanımlamasında, istediğiniz tanımlama belirtin; örneğin, günlüğe kaydetme değerleri.
2. Tetikleme işleminde kullanılan süreç tanımlamasına ilişkin Ortam Verilerini belirtin:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)')
```

Tetikleyici İzleme Programı, belirtilen tanımlama kullanarak bir SBMJOB işlemi gerçekleştirir.

Süreç tanımlamasının Ortam Verilerinde uygun anahtar sözcük ve değeri belirterek, SBMJOB ' un diğer özniteliklerini geçersiz kılmak mümkündür. Bu öznitelik tetikleyici izleyicisi tarafından doldurulduğundan, bunun tek kural dışı durumu CMD anahtar sözcüğüdür. Hem iş adının, hem de tanımlamanın değiştirileceği süreç tanımlamasının ortam verilerini belirtmek için kullanılan komut örneği:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)
JOB(TRIGGER)')
```

Tetikleme için bir uygulama kuyruğu tanımlama

Uygulama kuyruğu, ileti alışverişi için uygulamalar tarafından, MQI aracılığıyla kullanılan bir yerel kuyruktur. Tetikleme, uygulama kuyruğunda tanımlanacak kuyruk özniteliklerinin sayısını gerektirir. Tetikleme, TRGENBL özniteliği tarafından etkinleştirilir.

In this example, a trigger event is to be generated when there are 100 messages of priority 5 or higher on the local queue motor.insurance.queue, as follows:

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.insurance.queue') QTYPE(*LCL)
PRCNAME('motor.insurance.quote.process') MAXMSGLN(2000)
DFTMSGPST(*YES) INITQNAME('motor.ins.init.queue')
TRGENBL(*YES) TRGTYP(*DEPTH) TRGDEPTH(100) TRGMSGPTY(5)
```

parametrelerin bulunduğu yer:

MQMNAME(MYQUEUEMANAGER)

Kuyruk yöneticisinin adı.

QNAME('motor.insurance.queue')

Tanımlanmakta olan uygulama kuyruğunun adı.

PRCNAME('motor.insurance.quote.process')

Tetikleme izleme programı tarafından başlatılacak uygulamanın adı.

MAXMSGLEN(2000)

Kuyruktaki ileti uzunluğu üst sınırı.

DFTMSGPST(*YES)

Bu kuyruktaki iletiler varsayılan olarak kalıcı olur.

INITQNAME('motor.ins.init.queue')

Kuyruk yöneticisinin tetikleme iletisini koyacağı başlangıç kuyruğunun adı.

TRGENBL(*YES)

Tetikleme özniteliği değeri.

TRGTYPE(*DEPTH)

Bir tetikleme olayı, gerekli önceliğin ileti sayısı (**TRGMSGPTY**) olduğunda oluşturulur **TRGDEPTH** içinde belirtilen sayıya ulaşır.

TRGDEPTH(100)

Bir tetikleme olayı oluşturmak için gereken ileti sayısı.

TRGMSGPTY(5)

Bir tetikleme olayı oluşturulup oluşturulmayacağına karar vermek için kuyruk yöneticisi tarafından sayılacak iletilerin önceliği. Yalnızca, 5 ya da üzeri önceliğe sahip iletiler sayılır.

Başlatma kuyruğu tanımlanması

Bir tetikleme olayı ortaya çıktığında, kuyruk yöneticisi bir tetikleme iletisini uygulama kuyruğu tanımlamasında belirtilen başlangıç kuyruğuna koyar. Başlatma kuyruklarının özel ayarları yoktur, ancak yol göstermesi için yerel kuyruk `motor.ins.init.queue` ' un aşağıdaki tanımlamasını 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 kuyruktan iletileri işlemek için kullanılan uygulamayla ilişkilendirir. This is done through the PRCNAME attribute on the application queue `motor.insurance.quote.queue`. The following command creates the required process, `motor.insurance.quote.process`, identified in this example:

```
CRTMQMPRC MQMNAME(MYQUEUEMANAGER) PRCNAME('motor.insurance.quote.process')
TEXT('Insurance request message processing')
APPTYPE(*OS400) APPID(MQTEST/TESTPROG)
USRDATA('open, close, 235')
```

parametrelerin bulunduğu yer:

MQMNAME(MYQUEUEMANAGER)

Kuyruk yöneticisinin adı.

PRCNAME('motor.insurance.quote.process')

Süreç tanımlamasının adı.

TEXT('Insurance request message processing')

Bu tanımlamanın ilişkilendirdiği uygulama programının açıklaması. Bu metin, **DSPMQMPRC** komutunu kullandığınızda görüntülenir. Bu, işlemin ne yaptığını tanımlamanıza yardımcı olabilir. Dizede boşluk kullanırsanız, dizgiyi tek tırnak işareti içine almalısınız.

APPTYPE(*OS400)

Başlatılacak uygulamanın tipi.

APPID(MQTEST/TESTPROG)

Tam olarak nitelenmiş bir dosya adı olarak belirtilen, yürütülebilir uygulama dosyasının adı.

USRDATA('open, close, 235')

Uygulama tarafından kullanılacak, kullanıcı tanımlı veriler.

Süreç tanımlamanızın görüntülenmesi

Tanımınızın sonuçlarını incelemek için **DSPMQMPCRC** komutunu kullanın. Örneğin:

```
MQMNAME(MYQUEUEMANAGER) DSPMQMPCRC('motor.insurance.quote.process')
```

Ayrıca, bir süreç tanımlamasını silmek için **CHGMQMPRC** komutunu ve var olan bir süreç tanımlamasını değiştirmek için **DLTMQMPRC** komutunu kullanabilirsiniz.

IBM i

IBM üzerinde iki IBM MQ sistemi arasında iletişim kurma

Bu kodlama örneği, birbirleriyle iletişim kurabilmek 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 bir kuyruk yöneticisi yaratmak için QMGRA1 adını tıklatın.

```
CRTMQM  MQMNAME(QMGRA1) TEXT('System A - Queue +  
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

2. Bu kuyruk yöneticisini başlatın.

```
STRMQM  MQMNAME(QMGRA1)
```

3. Define the IBM MQ objects on SYSTEMA that you need to send messages to a queue manager on SYSTEMB.

```
/* 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) +  
RMTMQNAME(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 bir kuyruk yöneticisi yaratmak için QMGRB1 adını tıklatın.

```
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 duyacak 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, kanal başlatılabilmesi için SYSTEMB üzerinde bir TCP/IP dinleyicisi başlatın. Bu örnek, 1414 varsayılan kapısını kullanır.

```
STRMQMLSR MQMNAME(QMGRB1)
```

Şimdi SYSTEMA ile SYSTEMB arasında test iletileri göndermeye hazırsınız. Sağlanan örneklerden birini kullanarak, SYSTEMA SYSTEMA ' da uzak kuyruğunuza bir dizi ileti koyun.

Start the channel on SYSTEMA, either by using the command **STRMQMCHL**, or by using the command **WRKMQMCHL** and entering a start request (Option 14) against the sender channel.

Kanal RUNNING durumuna geçmeli ve iletiler SYSTEMB' ta TARGETB kuyruğuna gönderilmelidir.

Şu komutu girerek iletilerinizi denetleyin:

```
WRKMQMSG 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') +
RMTMQMNAME(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/AMQOINQ4') /* 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/AMQOSET4') /* 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/AMQOECH4') /* COBOL */ +
/* APPID('QMOM/AMQ3ECH4') /* RPG - ILE */

/*****/
/*                               */
/* Normal return.                               */
/*                               */
/*****/

```

```
SNDPGMMSG MSG('AMQSAMP4 Completed creating sample +
objects for ' *CAT &QMGRNAME)
RETURN
ENDPGM
```

```
/*
/* END OF AMQSAMP4
/*
/*
/*
```

IBM i IBM MQ for IBM iyönetimine iliřkin dięer yöntemler

CL komutlarının kullanılması, IBM MQ for IBM idenetimi için tercih edilen yöntemdir. Ancak, MQSC komutları, PCF komutları ve uzak denetim de içinde olmak üzere çeşitli dięer yönetim yöntemlerini kullanabilirsiniz.

You normally use IBM i CL commands to administer IBM MQ for IBM i. Bu komutlara genel bir bakış için bkz. [“CL komutlarını kullanarak IBM MQ for IBM i ' in yönetilmesi” sayfa 332.](#)

Kuyruk yöneticilerinin çalışmasını izlemek için IBM MQ özel işlem den geçirme olaylarını kullanabilirsiniz. IBM MQ özel işlem den geçirme olayları ve bunların nasıl kullanılacağı hakkında bilgi için bkz. [Instrumentation olayları](#) .

IBM i CL komutlarını kullanmaya alternatif olarak aşağıdaki alt konularda açıklanan yönetim yöntemlerinden herhangi birini kullanabilirsiniz:

IBM i IBM üzerinde yerel ve uzak yönetim

IBM MQ for IBM i nesnelerini yerel olarak ya da uzaktan yönetmenizi sağlar.

Yerel denetim , yerel sisteminizde tanımladığınız kuyruk yöneticilerindeki denetim görevlerini gerçekleştirmenizi sağlar. In IBM MQ, you can consider this as local administration because no IBM MQ channels are involved, that is, the communication is managed by the operating system. Bu tip bir görevi gerçekleştirmek için, uzak sistemde oturum açmanız ve komutları buradan yayınlamanız ya da sizin için komutları yayınlayabilecek bir işlem yapmanız gerekir.

IBM MQ , *uzaktan yönetim* olarak bilinen bir noktadan yönetmeyi tek bir noktadan destekler. Uzak denetim, hedef kuyruk yöneticisine programlanabilir komut biçimi (PCF) denetim iletileri SYSTEM. ADMIN. COMMAND. QUEUE ' e gönderilmesinden oluşur.

PCF iletileri oluşturmanın çeşitli yolları vardır. Bu bilgiler şunlardır:

1. PCF iletileri kullanılarak bir program yazılıyor. Bkz. [“IBM üzerinde PCF komutları kullanan yönetim” sayfa 348.](#)
2. Bir program, PCF iletilerini gönderen MQAI kullanılarak yazılıyor. Bkz. [“PCF 'lerin kullanımını kolaylaştırmak için MQAI' nin kullanılması” sayfa 35.](#)
3. Using the IBM MQ Explorer, available with IBM MQ for Windows, which allows you to use a graphical user interface (GUI) and generates the correct PCF messages. Bkz. [“IBM MQ Explorer ile IBM MQ for IBM kullanılması” sayfa 348.](#)
4. Komutları dolaylı olarak bir uzak kuyruk yöneticisine göndermek için **STRMQMQSC** komutunu kullanın. Bkz. [“IBM üzerinde MQSC komutlarını kullanarak yönetim” sayfa 346.](#)

Örneğin, uzak bir kuyruk yöneticilikindeki bir kuyruk tanımlamasını değiştirmek için bir uzak komut yayınlatabilirsiniz.

Bazı komutlar bu şekilde yayınlanamaz; özellikle, kuyruk yöneticilerini yaratma ya da başlatma ve komut sunucuları başlatılıyor. Bu tip bir görevi gerçekleştirmek için, uzak sistemde oturum açmanız ve komutları buradan yayınlamanız ya da sizin için komutları yayınlayabilecek bir işlem yapmanız gerekir.

IBM i IBM üzerinde MQSC komutlarını kullanarak yönetim

Bu bilgileri, MQSC komutları ve IBM MQ for IBM i ' i yönetmek için nasıl kullanacağını öğrenmek için kullanın.

IBM MQ komut dosyası (MQSC) komutları, EBCDIC metninde, insan tarafından okunabilir biçimde yazılır. Kuyruk yöneticisi nesnelerini, kuyrukları, süreç tanımlamalarını, ad listelerini, kanalları, istemci bağlantı kanallarını, dinleyicileri, hizmetleri, konuları ve kimlik doğrulama bilgileri nesnelerini de içeren kuyruk yöneticisi nesnelerini yönetmek için MQSC komutlarını kullanıyorsunuz.

You issue MQSC commands to a queue manager using the **STRMQMQSC** IBM MQ CL command. Bu y "ntem, yalnızca, giriş y" ntemi, sunucu kitaplık sistemindeki bir kaynak fiziksel ktkten almanyaca g Bu kaynak fiziksel dosyanın varsayılan adı QMQSC' dir.



Uyarı: QTEMP kitaplığının kullanımı sınırlı olduğu için, QTEMP kitaplığını STRMQMQSC ' ye kaynak kitaplık olarak kullanmayın. Komuta giriş dosyası olarak başka bir kitaplık kullanmanız gerekir.

IBM MQ for IBM i , QMQSCadlı bir kaynak dosya sağlamıyor. MQSC komutlarını işlemek için, aşağıdaki komutu vererek, QMQSC kaynak dosyasını seçiminizin bir kitaplığında yaratmanız gerekir:

```
CRTSRCPF FILE(MYLIB/QMQSC) RCDLEN(240) TEXT('IBM MQ - MQSC Source')
```

MQSC kaynağı bu kaynak dosya içindeki üyelerde tutuluyor. Üyelerle çalışmak için şu komutu girin:

```
WRKMBRPDM MYLIB/QMQSC
```

Artık yeni üyeler ekleyebilir ve var olanları koruyabilirsiniz.

MQSC komutlarını etkileşimli olarak da girebilir, RUNMQSC komutunu ya da:

1. Typing in the queue manager name and pressing the Enter key to access the **WRKMQM** results panel.
2. Bu panodaki F23=More options ögesini seçin.
3. [Şekil 23 sayfa 347](#) içinde gösterilen panodaki etkin bir kuyruk yöneticisine karşı 26. seçeneği belirleniyor.

Böyle bir MQSC oturumunu sona erdirmek için endyazın.

[Şekil 23 sayfa 347](#) , öznitelikleriyle bir MQSC komutu (DEFINE QLOCAL) gösteren bir MQSC komut kütüğünden çıkarılır.

```
.  
. .  
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 23. MQSC komut kütüğünden, myprog.in

IBM MQ ortamları arasında taşınabilirlik için, MQSC komut dosyalarındaki satır uzunluğunu 72 karaktere sınırlayın. Artı işareti, komutun sonraki satırda devam ettiğini gösterir.

MQSC ' de belirtilen nesne öznitelikleri, büyük/küçük harf duyarlı olmasalar da, bu bölümde büyük harfli olarak (örneğin, RQMNAME gibi) gösterilir.

Not:

1. Bir MQSC dosyasının biçimi, dosya sistemindeki konumuna bağlı değildir.
2. MQSC öznitelik adları sekiz karakterle sınırlıdır.

3. MQSC komutları, z/OS içinde olmak üzere diğer platformlarda kullanılabilir.

Her bir MQSC komutuna ve sözdizimine ilişkin açıklamalar için [MQSC komutları](#) başlıklı konuya bakın.

IBM i

IBM üzerinde PCF komutları kullanan yönetim

The purpose of IBM MQ programmable command format (PCF) commands is to allow administration tasks to be programmed into an administration program. Bu şekilde, kuyruklar ve süreç tanımlamaları yaratılabilir ve bir programdan kuyruk yöneticilerini değiştirebilirsiniz.

PCF komutları, MQSC komutları tarafından sağlanan işlevlerin aynı aralığını kapsamaya devam eder. Ancak, MQSC komutlarının tersine, PCF komutları ve yanıtları okuyabileceğiniz bir metin biçiminde değildir.

Bir programı, tek bir düğümden ağdaki herhangi bir kuyruk yöneticisine PCF komutları yayınlatabileceğiniz şekilde yazabilirsiniz. Bu şekilde, yönetim görevlerini merkezileştirebilir ve otomatikleştirebilirsiniz.

Her PCF komutu, bir IBM MQ iletilerinin uygulama verileri bölümüne gömülü bir veri yapısıdır. Her komut, başka bir iletiyle aynı şekilde MQPUT adlı MQI işlevini kullanarak hedef kuyruk yöneticisine gönderilir. İleti alan kuyruk yöneticisindeki komut sunucusu, iletiyi bir komut iletileri olarak yorumlayıp komutu çalıştırır. Yanıtları almak için, uygulama bir MQGET çağrısını yayınlar ve yanıt verileri başka bir veri yapısında döndürülür. Uygulama daha sonra yanıtı işleyebilir ve uygun şekilde davranabilir.

Kısaca, uygulama programcısının bir PCF komut iletileri yaratmak için belirtmesi gereken bazı şeyler şunlardır:

İleti tanımlayıcısı

Bu, standart bir IBM MQ ileti tanımlayıcısının içinde yer alan bir açıklamadır:

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

Uygulama Verileri

PCF üstbilgisini de içeren PCF iletilerini içerir. Bu ileti aşağıdaki gibi:

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

Çıkış PCF ' leri, ileti metninde MQSC komutları içeren PCF komutlarıdır. Bir uzak kuyruk yöneticisine komut göndermek için PCF ' leri kullanabilirsiniz. Ek bilgi için "[PCF 'lerin kullanımını kolaylaştırmak için MQAI' nin kullanılması](#)" sayfa 35 ' e bakın.

PCF veri yapılarının tam açıklaması ve bunların nasıl gerçekleştirileceği için bkz. [Komutlar ve yanıtlar için yapılar](#).

IBM i

IBM MQ Explorer ile IBM MQ for IBM i kullanılması

Use this information to administer IBM MQ for IBM i using the IBM MQ Explorer.

IBM MQ for Windows (x86 platform), and IBM MQ for Linux (x86 and x86-64 platforms) provide an administration interface called the IBM MQ Explorer to perform administration tasks as an alternative to using CL, control, or MQSC commands.

The IBM MQ Explorer allows you to perform local or remote administration of your network from a computer running Windows (x86 platform), or Linux (x86 and x86-64 platforms), by pointing the IBM MQ Explorer at the queue managers and clusters you are interested in.

IBM MQ Explorer ile şunları yapabilirsiniz:

- Kuyruk yöneticisini başlatma ve durdurma (yalnızca yerel makineniz için).
- Kuyruklar, konular ve kanallar gibi IBM MQ nesnelilerinin tanımlarını tanımlayın, görüntüleyin ve değiştirin.
- Kuyruklardaki iletilere göz atın.
- Bir kanalı başlat ve durdur.

- Bir kanalla ilgili durum bilgilerini görüntüler.
- Bir kümedeki kuyruk yöneticilerini görüntüleyin.
- Hangi uygulamaların, kullanıcıların ya da kanalların belirli bir kuyruğu açık olduğunu görmek için işaretleyin.
- **Create New Cluster** (Yeni Küme Oluştur) sihirbazını kullanarak yeni bir kuyruk yöneticisi kümesi yaratın.
- **Add Queue Manager to Cluster** (Küme Yöneticisine Kuyruk Yöneticisi Ekle) sihirbazını kullanarak bir kümeye kuyruk yöneticisi ekleyin.
- TLS (Transport Layer Security; İletim Katmanı Güvenliği) kanal güvenliği ile birlikte kullanılan kimlik doğrulama bilgileri nesnesini yönetmenizi sağlar.

Çevrimiçi rehberliği kullanarak şunları yapabilirsiniz:

- Kuyruk yöneticileri, kuyruklar, kanallar, süreç tanımlamaları, istemci bağlantı kanalları, dinleyiciler, konular, hizmetler, ad listeleri ve kümeler dahil olmak üzere çeşitli kaynakları tanımlayın ve denetleyebilirsiniz.
- Bir kuyruk yöneticisini ve ilişkili süreçleri başlatır ya da durdurur.
- İş istasyonunuzdaki ya da diğer iş istasyonlarındaki kuyruk yöneticilerini ve ilişkili nesnelere görüntüleyebilirsiniz.
- Kuyruk yöneticilerinin, kümelerin ve kanalların durumunu denetleyin.

Bir sunucu makinesinde IBM MQ 'i yönetmek için IBM MQ Explorer 'i kullanma girişiminde bulunmadan önce aşağıdaki gereksinimleri karşıladığınızdan emin olun. Şuna bakın:

1. A command server is running for any queue manager being administered, started on the server by the CL command **STRMQMCSVR**.
2. Her uzak kuyruk yöneticisi için uygun bir TCP/IP dinleyicisi var. Bu, **STRMQMLSR** dinleyicisinin başlattığı IBM MQ dinleyicidir.
3. The server connection channel, called SYSTEM . ADMIN . SVRCONN, exists on every remote queue manager. Bu kanalı kendiniz oluşturmalısınız. Bu, denetlenmekte olan her uzak kuyruk yöneticisi için zorunludur. O olmadan, uzaktan yönetim mümkün değildir.
4. SYSTEM . MQEXPLORER . REPLY . MODEL kuyruğunun var olduğunu doğrulayın.

IBM i IBM üzerinde uzak denetim için komut sunucusunun yönetilmesi

Use this information to learn about the remote administration of IBM MQ for IBM i command server.

Her kuyruk yöneticisinin kendisiyle ilişkili bir komut sunucusu olabilir. Komut sunucusu, uzak kuyruk yöneticilerinden gelen komutları ya da uygulamalardan PCF komutlarından herhangi bir komutu işler. Bu komut, komutun kökenine bağlı olarak bir tamamlanma kodu ya da işletmen iletisini işlemek için kuyruk yöneticisine sunar ve bir tamamlanma kodu ya da işletmen iletisi döndürür.

Bir komut sunucusu, PCF 'leri, MQAI' yi ve ayrıca uzak denetim için de yer alan tüm yönetim için zorunludur.

Not: Uzaktan yönetim için, hedef kuyruk yöneticisinin çalışır durumda olduğundan emin olmalısınız. Ters durumda, komutlar içeren iletiler, gönderildikleri kuyruk yöneticisini bırakamaz. Bunun yerine, bu iletiler uzak kuyruk yöneticisine hizmet veren yerel iletim kuyruğunda kuyruğa alınır. Mümkünse bu durumdan kaçınin.

Komut sunucusunu başlatmak ve durdurmak için ayrı denetim komutları vardır. IBM MQ Gezginini 'ni kullanarak, aşağıdaki bölümlerde açıklanan işlemleri gerçekleştirebilirsiniz.

Komut sunucusunun başlatılması ve durdurulması

Komut sunucusunu başlatmak için bu CL komutunu kullanın:

```
STRMQMCSVR MQMNAME('saturn.queue.manager')
```

Burada `saturn.queue.manager` , komut sunucusunun başlatılmakta olduğu kuyruk yöneticidir.

Komut sunucusunu durdurmak için aşağıdaki CL komutlarından birini kullanın:

1.

```
ENDMQMSVR MQMNAME('saturn.queue.manager') OPTION(*CNTRLD)
```

controlled, komut sunucusunun durdurulduğu kuyruk yöneticidir. Burada `saturn.queue.manager` , denetimli bir durdurma işlemini gerçekleştirir. Bu varsayılan seçenektir; bu seçenek, `OPTION(*CNTRLD)` ' nin atlanabileceği anlamına gelir.

2.

```
ENDMQMSVR MQMNAME('saturn.queue.manager') OPTION(*IMMED)
```

to perform an immediate stop, where `saturn.queue.manager` is the queue manager for which the command server is being stopped.

Komut Sunucusunun Durumunun Görüntülenmesi

Uzak denetim için, hedef kuyruk yöneticindeki komut sunucusunun çalışmakta olduğundan emin olun. Çalıştırılmamışsa, uzak komutlar işlenemez. Any messages containing commands are queued in the target queue manager's command queue `SYSTEM.ADMIN.COMMAND.QUEUE`.

Bir kuyruk yöneticisine ilişkin komut sunucusunun durumunu görüntülemek için, burada `saturn.queue.manager` komutunu çağırılmış, CL komutu aşağıdaki gibi olur:

```
DSPMQMSVR MQMNAME('saturn.queue.manager')
```

Bu komutu hedef makinede yayınlayın. Komut sunucusu çalışıyorsa, [Şekil 24 sayfa 350](#) 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 24. MQM Komut Sunucusu panosunu görüntüle

IBM i Web konsolu komutlarının çalıştırılması

Web konsolu ile ilgili Qshell komutlarının IBM MQ for IBM i' ta doğru şekilde çalışması için ortamınızı aşağıdaki metinde açıkladığı şekilde yapılandırmanız gerekir.

Bu görev hakkında

Qshell başlatıldığında, işin CCSID 'sini temel alan komutların işlenmesi için iç çizelgeleri başlatır. Web konsolu ile ilgili Qshell komutlarının IBM i üzerinde doğru çalışması için ortamınızı yapılandırmanız gerekir.

Bir ülke değeri, LANG ortam değişkeni yerel bir nesneye giden yol adına ayarlanarak ayarlanır. Örneğin, ABD İngilizcesi için yerel ayarı belirlemek için LANG ortam değişkeni aşağıdaki gibi ayarlanır:

```
LANG=/QSYS.LIB/EN_US.LOCALE
```

Qshell 'de, tüm ortam değişkenlerini listelemek için, komutu çıkış komutuyla denetleyebilirsiniz. Genellikle, yürütme ortamı için ülke değerini etkileyebilecek LANG ' dir. LC_ALL kullanılabilir de olabilir.

Qshell komutlarını doğru olarak çalıştırmak için, yerel ayar ortam ayarının iş ayarınızla tutarlı olması gerekir.

Yordam

DSPJOB JOB komutunu kullan (JobNumber/USERProfile/JobName)

- İş tanımlaması özniteliklerini görüntülemek için seçenek 2 'yi belirleyin.
- Aşağıdaki öznitelikler LANG ya da LC_ALL ortam ayarlarıyla tutarlı olmalıdır.

- Dil kodu
- Ülke ya da bölge tanıtıcısı
- Kodlanmış karakter takımı tanıtıcısı

Örneğin,

```
LANG=/QSYS.LIB/FR_FR.LOCALE
```

İş öznitelikleriniz aşağıdaki gibi olmalıdır:

- Dil kodu..... fra
- Ülke ya da bölge tanıtıcısı..... Cu
- Kodlanmış karakter takımı tanıtıcısı.... 297

Sonraki adım

Ulusal dil desteğine ilişkin ek bilgi için [Ulusal dil desteği \(NLS\) ile ilgili dikkat edilmesi gereken noktalar](#) başlıklı IBM Documentation konusuna bakın.

IBM i Work management for IBM i

This information describes the way in which IBM MQ handles work requests, and details the options available for prioritizing and controlling the jobs associated with IBM MQ.

Uyarı

IBM i ve IBM MQ iş yönetimi kavramlarını tam olarak anlamadığınız sürece, IBM MQ iş yönetimi nesnelereyi değiştirmeyin.

Altsistemlere ve iş tanımlamalarına ilişkin ek bilgi için IBM i ürün belgelerindeki [Work Management \(İş Yönetimi\)](#) başlığı altında yer alan bilgilere bakın. [Başlangıç işleri](#) ve [Toplu işler](#) bölümlerindeki bölümlere özel önem veriniz.

IBM MQ for IBM i , IBM i UNIX ortamını ve IBM i iş parçacıklarını içerir. **değil** , Integrated File System (IFS) içindeki nesnelere herhangi bir değişiklik yapar.

Olağan işlemler sırasında, bir IBM MQ kuyruk yöneticisi farklı görevleri yerine getirmek için bir dizi toplu işi başlatır. Varsayılan olarak, bu toplu işler, IBM MQ kurulduğunda yaratılan QMQM altsisteminde çalışır.

İş yönetimi, sisteminizden en iyi başarıyı elde etmek ya da yönetimi daha basit hale getirmek için IBM MQ görevlerini terleme işlemine başvuruda bulunur.

Örneğin, şunları yapabilirsiniz:

- Bir kuyruk yöneticisini diğerinden daha duyarlı hale getirmek için işlerin çalıştırma önceliğini değiştirin.
- Bir iş sayısının çıkışını belirli bir çıkış kuyruğuna doğru yeniden yönlendir.

- Belirli bir tipteki tüm işleri belirli bir altsistemde çalışır duruma getirin.
- Bir altsistemle hataları yalıtın.

İş yönetimi, IBM MQ işleriyle ilişkili iş tanımlamaları yaratılarak ya da değiştirilerek yürütülür. Aşağıdakiler için iş yönetimini yapılandırabilirsiniz:

- Tüm IBM MQ kuruluşu.
- Tek tek kuyruk yöneticileri.
- Tek tek kuyruk yöneticileri için tek tek işler.

IBM i IBM için IBM MQ görevleri

Bu, IBM MQ for IBM i işlerinden oluşan bir tablodur ve her birinin kısa bir açıklamasıdır.

Bir kuyruk yöneticisi çalışırken, IBM MQ altsistemindeki QMQM kullanıcı tanıtımı altında çalışmakta olan aşağıdaki toplu işlerin bazılarını ya da tümünü görürsünüz. İşler, [Çizelge 21 sayfa 352](#)'de kısaca açıklanmıştır.

Kuyruk Yöneticisiyle Çalış (WRKMQM) panosunda seçenek 22 'yi kullanarak, kuyruk yöneticisine bağlı tüm işleri görüntüleyebilirsiniz. WRKMQLSR komutunu kullanarak dinleyicileri görüntüleyebilirsiniz.

Çizelge 21. IBM MQ Görevler.	
İş adı	İşlev
AMQZMUC0	Yardımcı program yöneticisi. Bu iş, kritik kuyruk yöneticisi yardımcı programlarını yürütür; örneğin, günlük zinciri yöneticisi.
AMQZXMA0	Kuyruk yöneticisi tarafından başlatılan ilk iş olan yürütme denetleyicisi. Bu, MQCONN isteklerini işler ve IBM MQ API çağrılarını işlemek için aracı işlemlerini başlatır.
AMQZFUMA	Nesne yetkisi yöneticisi (OAM).
AMQZLAA0	MQCNO_STANDARD_BINDING MQCNO_STANDARD_BINDING komutunu kullanarak 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, kritik kuyruk yöneticisi yardımcı programlarını yürütür; örneğin, günlük zinciri yöneticisi.
AMQFQPUB	Kuyruğa yollanmış yayınlama/abone olma yardımcı programı.
AMQFCXBA	Aracı işçi işi.
RUNMQBRK	Aracı denetim işi.
AMQRPMPPA	Kanal işleme işi.
AMQCRSTA	TCP/IP tarafından çağrılan kanal yanıtlayıcısı.
AMQCRS6B	LU62 alıcı kanalı ve istemci bağlantısı (nota bakın).
AMQRRMFA	Kümeler için havuz yöneticisi.
AMQCLMAA	İş parçacıklı olmayan TCP/IP dinleyicisi.
AMQPCSEA	PCF ve uzak denetim isteklerini işleyen PCF komut işlemcisi.
RUNMQTRM	Tetikleme monitörü.

Çizelge 21. IBM MQ Görevler. (devamı var)	
İş adı	İşlev
RUNMQDLQ	Ölü harf kuyruğu işleyicisi.
RUNMQCHI	Kanal başlatıcı.
MQCHCHL	Her bir gönderen kanalı için başlatılan gönderen kanal işi.
RUNMQLSR	Yivli TCP/IP dinleyici.
AMQRCMLA	Kanal MQSC ve PCF komut işlemcisi.

Not: LU62 günlük nesnesi işi, iletişim altsisteminde çalışır ve yürütüm özelliklerini, işi başlatmak için kullanılan yöneltme ve iletişim girişlerinden alır. Ek bilgi için bkz. [Initiated end \(Receiver\)](#).

IBM i IBM üzerinde iş yönetimi nesneleri

IBM MQ kurulduğunda, iş yönetimine yardımcı olması için QMQM kitaplığında çeşitli nesnelere sağlanır. Bu nesnelere, IBM MQ işlerinin kendi altsisteminde çalışması için gerekli olan nesnelere dir.

Örnek iş tanımları, IBM MQ toplu işlerinden iki tanesi için sağlanır. Bir IBM MQ işi için belirli bir iş tanımı sağlanmıyorsa, QMQMJOB D varsayılan iş tanımlaması ile çalışır.

IBM MQ kurulumu sırasında sağlanan iş yönetimi nesnelere, [Çizelge 22 sayfa 353](#) içinde listelenir ve bir kuyruk yöneticisi için oluşturulan nesnelere [Çizelge 23 sayfa 353](#)' ta listelenir.

Not: İş yönetimi nesnelere QMQM kitaplığında bulunabilir ve kuyruk yöneticisi nesnelere kuyruk yöneticisi kitaplığında bulunabilir.

Çizelge 22. İş yönetimi nesnelere		
Ad	Tip	Tanım
AMQZLAA0	*JOB D	IBM MQ aracı işlemlere tarafından kullanılan iş tanımı
AMQZLSA0	*JOB D	Yalıtılmış bağ tanımları kuyruk yöneticisi aracısı
AMQZXMA0	*JOB D	IBM MQ yürütme denetleyicilere tarafından kullanılan iş tanımı
QMQM	*SBS D	Tüm IBM MQ işlerinin çalıştırıldığı altsistem
QMQM	*JOB Q	Belirtilen altsisteme bağlı iş kuyruğu
QMQMJOB D	*JOB D	Bir iş için belirli bir iş tanımlaması yoksa, varsayılan IBM MQ iş tanımlaması kullanılır.
QMQMMSG	*MSG Q	IBM MQ işlere için varsayılan ileti kuyruğu.
QMQMRUN20	*CLS	Yüksek öncelikli IBM MQ işlere için bir sınıf açıklaması
QMQMRUN35	*CLS	Orta öncelikli IBM MQ işlere için bir sınıf tanımı
QMQMRUN50	*CLS	Düşük öncelikli IBM MQ işlere için bir sınıf açıklaması

Çizelge 23. Bir kuyruk yöneticisi için yaratılan iş yönetimi nesnelere		
Ad	Tip	Tanım
AMQA000000	*JRNRCV	Yerel günlük nesnesi
AMQAJRN	*JRN	Yerel günlük

Çizelge 23. Bir kuyruk yöneticisi için yaratılan iş yönetimi nesneleri (devamı var)

Ad	Tip	Tanım
AMQJRNINF	*USRSPC	Bir kuyruk yöneticisinin başlatılması ve ortam kurtarması için gereken en son günlük nesneleriyle güncellenen kullanıcı alanı. Bu kullanıcı alanı, hangi günlük nesnelerin arşivlenmesini gerektireceğini ve hangilerinin güvenli silinebileceğini belirlemek için bir uygulama tarafından sorgulanabilir.
AJMQAJRNMMSG	*MSGQ	Yerel günlük ileti kuyruğu
AMQCRC6B	*PGM	LU6.2 bağlantısını başlatmak için program
AMQRFOLD	*DOSYA	Geçirilen kuyruk yöneticisi kanal tanımlama dosyası
QMQMMSG	*MSGQ	Kuyruk yöneticisi ileti kuyruğu

IBM i

How IBM MQ uses work management objects on IBM i

Bu bilgilerde, IBM MQ ' in iş yönetimi nesneleri nasıl kullandığı ve yapılandırma örnekleri nasıl sağladığı açıklanır.



Uyarı: Altsistemde izin verilen iş sayısını öncelik olarak sınırlamak için QMQM altsistemindeki iş kuyruğu giriş ayarlarını değiştirmeyin. Bu işlemi gerçekleştirmeye çalışırsanız, teslim edildikten sonra gerekli IBM MQ işlerinin çalışmasını durdurabilir ve kuyruk yöneticisi başlatma işleminin başarısız olmasına neden olabilirsiniz.

İş yönetimini nasıl yapılandıracağını anlamak için, öncelikle IBM MQ ' in iş tanımlarını nasıl kullandığını anlamalısınız.

İşi başlatmak için kullanılan iş tanımlaması, işin birçok özneliğini denetler. Örneğin:

- İşin kuyruğa alındığı iş kuyruğu ve işin çalıştırıldığı altsistem.
- İşin yürütüm değiştirgelemleri için kullandığı işi ve sınıfı başlatmak için kullanılan yöneltme verileri.
- İşin yazdırma dosyaları için kullandığı çıkış kuyruğu.

Bir IBM MQ işini başlatma işlemi üç adımda göz önünde bulundurulabilir:

1. IBM MQ , bir iş tanımlaması seçer.

IBM MQ , toplu iş için hangi iş tanımının kullanılacağını belirlemek için aşağıdaki tekniği kullanır:

- a. İşle aynı adı taşıyan bir iş tanımlaması için kuyruk yöneticisi kitaplığına bakın. Kuyruk yöneticisi kitaplığına ilişkin ek bilgi için [IBM MQ for IBM i kuyruk yöneticisi kitaplık adlarının anlaşılması](#) başlıklı konuya bakın.
- b. QMQMJOBd varsayılan iş tanımlaması için kuyruk yöneticisi kitaplığına bakın.
- c. İşle aynı adı taşıyan bir iş tanımlaması için QMQM kitaplığına bakın.
- d. QMQM kitaplığında varsayılan iş tanımlamasını (QMQMJOBd) kullanın.

2. İş, iş kuyruğuna gönderilir.

IBM MQ ile sağlanan iş tanımlamaları, QMQM kitaplığındaki QMQM iş kuyruğuna iş yerleştirmek için varsayılan olarak ayarlanmıştır. QMQM iş kuyruğu, sağlanan QMQM altsistemine bağlanır; bu nedenle, QMQM altsisteminde çalışan varsayılan işler varsayılan olarak çalışır.

3. İş, altsisteme girer ve yöneltme adımlarından geçer.

İş, altsisteme girdiğinde, iş tanımında belirlenen yöneltme verileri, işe ilişkin yöneltme girişlerini bulmak için kullanılır.

Yöneltme verileri, QMQM altsisteminde tanımlı olan yöneltme girişlerinden biriyle eşleşmelidir ve iş tarafından sağlanan sınıfların (QMQMRUN20, QMQMRUN35ya da QMQMRUN50) hangilerinin kullanıldığını tanımlar.

Not: IBM MQ işleri başlatılamıyorsa, altsistemin çalışır durumda olduğundan ve iş kuyruğunun tutulmadığından emin olun.

IBM MQ iş yönetimi nesnelere değiştiriyorsanız, her şeyin doğru bir şekilde ilişkilendirildiğinden emin olun. Örneğin, iş tanımlamasında QMQM/QMQM dışında bir iş kuyruğu belirlerseniz, altsistem için bir ADDJOBQE işlemi gerçekleştirildiğinden emin olun, bu da QMQM.

Çizelge 21 sayfa 352 ' ta belgelenen her iş için, aşağıdaki çalışma sayfasını örnek olarak kullanarak bir iş tanımı yaratabilirsiniz:

```
What is the queue manager library name? _____
Does job description AMQZXA0 exist in the queue manager library? Yes No
Does job description QMQMJOB exist in the queue manager library? Yes No
Does job description AMQZXA0 exist in the QMQM library? Yes No
Does job description QMQMJOB exist in the QMQM library? Yes No
```

Bu soruların tümüne hayır yanıtını verdiyseniz, QMQM kitaplığında QMQMJOB genel iş tanımlaması yaratın.

IBM MQ ileti kuyruğu

Her kuyruk yöneticisi kitaplığında bir IBM MQ ileti kuyruğu (QMOMMSG) yaratılır. Kuyruk yöneticisi işleri sona erdirildiğinde ve IBM MQ iletileri kuyruğa gönderildiğinde işletim sistemi iletileri bu kuyruğa gönderilir. Örneğin, başlatma sırasında hangi günlük nesnelere gerekli olduğunu bildirmek için. Bu ileti kuyruğunda ileti sayısını, izlenebilmeyi kolaylaştırmak için yönetilebilir bir boyutta tutun.

IBM i IBM için varsayılan sistem örnekleri

Bu örnekler, standart işlerin bir kısmı kuyruk yöneticisi başlatma sırasında sunulduğunda, değiştirilmemiş bir IBM MQ kuruluşunun nasıl çalıştığını gösterir.

İlk olarak, AMQZXA0 yürütme denetleyicisi işi başlar.

1. Issue the **STRMQM** command for queue manager TESTQM.

2. IBM MQ , QMTESTQM adlı kuyruk yöneticisi kitaplığında, ilk olarak AMQZXA0 iş tanımlaması ve QMQMJOB iş tanımlaması için arama yapar.

Neither of these job descriptions exist, so IBM MQ looks for job description AMQZXA0 in the product library QMQM. Bu iş tanımlaması var, bu nedenle işi sunmak için kullanılır.

3. İş tanımlaması, IBM MQ varsayılan iş kuyruğunu kullanır; bu nedenle, iş QMQM/QMQM iş kuyruğuna gönderilir.

4. AMQZXA0 iş tanımındaki yönlendirme verileri QMQMRUN20 olduğundan, sistem, altsistem yönlendirme girişlerinde bu verilerle eşleşen bir veri araması yapar.

Varsayılan olarak, 9900 sıra numarasına sahip yönlendirme girdisinin QMQMRUN20 ile eşleşen karşılaştırma verileri vardır; bu nedenle iş, o yönlendirme girişinde tanımlanan sınıfla (QMQMRUN20 adı da verilir) başlatılır.

5. QMQM/QMQMRUN20 sınıfının çalıştırma önceliği 20 değerine ayarlandığı için, AMQZXA0 işi, sistemdeki en etkileşimli işlerle aynı önceliğe sahip olan QMQM altsisteminde çalışır.

IBM i IBM i' ta iş yönetimi örneklerini yapılandırma

IBM MQ işlerinin yürütme ortamı özelliklerini değiştirmek için IBM MQ iş tanımlarını nasıl değiştirebileceğimizi ve nasıl değiştirebileceğimizi öğrenmek için bu bilgileri kullanın.

IBM MQ iş yönetiminin esnekliğinin anahtarı, IBM MQ ' in iş tanımlarını aramak için iki katmanlı bir şekilde yatar:

- Bir kuyruk yöneticisi kitaplığında iş tanımları yaratır ya da değiştiriyorsanız, bu değişiklikler QMQM ' deki genel iş tanımlarını geçersiz kılar, ancak değişiklikler yereldir ve yalnızca o kuyruk yöneticisini tek başına etkiler.

- QMQM kitaplığında genel iş tanımları yaratır ya da varolanları değiştirirseniz, bu iş tanımları, 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 artırmaktadır.

Havuz yöneticisi ve kanal başlatıcı işleri, AMQRRMFA ve RUNMQCHI ' yi yapmak için, kuyruk yöneticisi TESTQM için mümkün olduğunca hızlı çalıştırın ve aşağıdaki adımları gerçekleştirin:

- a. Kuyruk yöneticisi kitaplığında denetlemek istediğiniz IBM MQ işlemlerinin adlarıyla QMQM/ QMQMJOBDB iş tanımının yerel yinlemelerini 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 tanımlamasındaki yöneltme verileri parametresini 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 tanımlarını kullan
 - İşler altsisteme girildiğinde QMQMRUN20 sınıfı kullanıldığından, öncelik 20 ile çalıştırın.
2. Aşağıdaki örnekte, QMQM altsistemine ilişkin yeni bir çalışma önceliği sınıfı tanımlanmaktadır.

- a. Diğer kuyruk yöneticilerinin sınıfa erişmelerine izin vermek için QMQM kitaplığında yinelenen bir sınıf yaratın ve aşağıdaki komutu verin:

```
CRTDUPOBJ OBJ(QMQMRUN20) FROMLIB(QMQM) OBJTYPE(*CLS) TOLIB(QMQM)
NEWOBJ(QMQMRUN10)
```

- b. Aşağıdaki komutu girerek sınıfı yeni çalışma önceliğine sahip olacak şekilde değiştirin:

```
CHGCLS CLS(QMQM/QMQMRUN10) RUNPTY(10)
```

- c. Aşağıdaki komutu girerek, yeni sınıf tanımlamasını altsisteme ekleyin:

```
ADDRTGE SBSDB(QMQM/QMQM) SEQNBR(8999) CMPVAL('QMQMRUN10') PGM(QSYS/QCMD)
CLS(QMQM/QMQMRUN10)
```

Not: Yöneltme sırası numarası için herhangi bir sayısal değer belirleyebilirsiniz, ancak değerlerin sıralı düzende olması gerekir. Bu sıra numarası, altsisteme, yöneltme verilerinin bir yöneltme verileri eşleşmesi için hangi yöneltme girişlerinin aranacağı sıralamayı bildirir.

- d. Aşağıdaki komutu girerek, yerel ya da genel iş tanımlamasını yeni öncelik sınıfını kullanacak şekilde değiştirin:

```
CHGJOBDB JOBDB(QMQMLibname/QMQMJOBDB) RTGDTA('QMQMRUN10')
```

Şimdi QMLibraryname ile ilişkili tüm kuyruk yöneticisi işleri, 10 'un çalışma önceliği kullanır.

3. Aşağıdaki örnekte, kendi altsisteminde bir kuyruk yöneticisi çalıştırılır

QBATCH altsistemindeki kuyruk yöneticisi TESTQM için tüm işleri yapmak üzere aşağıdaki adımları gerçekleştirin:

- a. Kuyruk yöneticisi kitaplığındaki QMQM/QMQMJOBDB iş tanımının yerel bir kopyasını şu komutla yaratın:

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) TOLIB(QMTESTQM)
```

- b. İşlerin QBATCH iş kuyruğunu kullandığından emin olmak için iş tanımındaki iş kuyruğu parametresini değiştirin.

```
CHGJOB JOB(QMTESTQM/QMQMJOB) JOBQ(*LIBL/QBATCH)
```

Not: İş kuyruğu altsistem tanımlamasıyla ilişkilendirilir. İşlerin iş kuyruğunda kaldığını bulursanız, iş kuyruğu tanımının SBSDB ' de tanımlandığını doğrulayın. Altsistem için DSPSBSDB komutunu kullanın ve 6seçeneğini, İş kuyruğu girişlerini kullanın.

Kuyruk yöneticisi TESTQM için şimdi tüm işler şimdi:

- Kuyruk yöneticisi kitaplığındaki yeni yerel varsayılan iş tanımlamasını kullan
- İş kuyruğu QBATCH iş kuyruğuna gönderildi.

İşlerin doğru şekilde yönlendirildiğinden ve önceliklendirildiğinden emin olmak için:

- QBATCH altsistemindeki IBM MQ işleri için yöneltme girişleri yaratın ya da
- Hangi yöneltme verilerinin kullanıldığına bakılmaksızın, QCMD ' yi çağırın, tüm yöneltme girişlerine güvenen bir hedef kiti.

Bu seçenek yalnızca, iş kuyruğu QBATCH için etkin iş sayısı üst sınırı olarak *NOMAX değeri belirlendiyse işlev görmektedir. Sistemin varsayılan değeri 1 'dir.

4. Aşağıdaki örnek, başka bir IBM MQ altsistemi oluşturur.

- a. Aşağıdaki komutu girerek QMQM kitaplığında yinelenen bir altsistem yaratın:

```
CRTDUPOBJ OBJ(QMQM) FROMLIB(QMQM) OBJTYPE(*SBSDB) TOLIB(QMQM) NEWOBJ(QMQM2)
```

- b. QMQM iş kuyruğunu kaldırmak için aşağıdaki komutu kullanın:

```
RMVJOBQE SBSDB(QMQM/QMQM2) JOBQ(QMQM/QMQM)
```

- c. Altsisteme ilişkin yeni bir iş kuyruğu yaratmak için aşağıdaki komutu kullanın:

```
CRTJOBQ JOBQ(QMQM/QMQM2) TEXT('Job queue for IBM MQ Queue Manager')
```

- d. Altsisteme bir iş kuyruğu girişi eklemek için aşağıdaki komutu girin:

```
ADDJOBQE SBSDB(QMQM/QMQM2) JOBQ(QMQM/QMQM2) MAXACT(*NOMAX)
```

- e. Aşağıdaki komutu girerek kuyruk yöneticisi kitaplığında yinelenen bir QMQMJOB yarattın:

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) TOLIB(QMlibraryname)
```

- f. Yeni iş kuyruğunu kullanmak için, aşağıdaki komutu girerek iş tanımlamasını değiştirin:

```
CHGJOB JOB(QMlibraryname/QMQMJOB) JOBQ(QMQM/QMQM2)
```

- g. Altsistemi başlatmak için aşağıdaki komutu kullanın:

```
STRSBS SBSDB(QMQM/QMQM2)
```

Not:

- a. Altsistemi herhangi bir kitaplıkta belirleyebilirsiniz. Ürünün yeniden kurulması için herhangi bir nedenle ya da QMQM kitaplığı değiştirilirse, yaptığınız değişiklikler kaldırılır.

b. QMLibraryname ile ilişkilendirilmiş tüm kuyruk yöneticisi işleri şimdi QMQM2alt sistemi altında çalışır.

IBM i Availability, backup, recovery, and restart on IBM i

IBM MQ for IBM i 'in yedekleme ve geri yükleme stratejisine yardımcı olması için IBM i günlük kaydı desteğini nasıl kullandığını anlamak için bu bilgileri kullanın.

You must be familiar with standard IBM i backup and recovery methods, and with the use of journals and their associated journal receivers on IBM i, before reading this section. Bu konulara ilişkin bilgi edinmek için [Yedekleme ve kurtarmabaşlıklı](#) konuya bakın.

Yedekleme ve kurtarma stratejisini anlamak için, öncelikle IBM MQ for IBM i 'un verilerini IBM i dosya sistemindeki ve tümleşik dosya sistemindeki (IFS) nasıl düzenlediğini anlamanız gerekir.

IBM MQ for IBM i , verilerini, her kuyruk yöneticisi yönetim ortamı için ayrı bir kitaplıkta ve IFS dosya sistemindeki akış dosyalarında tutar.

Kuyruk yöneticisi özel kitaplıkları, kuyruk yöneticisinin iş yönetimini denetlemek için gereken günlükler, günlük nesnelere ve nesnelere içerir. IFS dizinleri ve dosyaları, IBM MQ yapılandırma dosyalarını, IBM MQ nesnelere açıklamalarını ve içerdikleri verileri içerir.

Bu nesnelere, bir sistem hatasından kurtarılabilir olan her değişiklik, uygun nesneye uygulanmadan önce bir günlük olarak kaydedilir. Bu durum, günlüğe kaydedilen bilgilerin yeniden canlandırılarak bu tür değişikliklerin kurtarılabilirliğinin etkisine sahiptir.

Bir sunucu ya da kuyruk yöneticisi hatası durumunda, artan kuyruk yöneticisi kullanılabilirliği sağlamak ve kurtarma işlemini hızlandırmak için, IBM MQ for IBM i 'u farklı sunucularda birden çok kuyruk yöneticisi eşgörunümü kullanacak şekilde yapılandırabilirsiniz.

IBM i Queue manager journals on IBM i

IBM MQ for IBM i 'un, yerel nesnelere yapılan güncellemeleri denetlemek için kullandığı günlüklerin nasıl kullandığını anlamak için bu bilgileri kullanın.

Her kuyruk yöneticisi kitaplığı, o kuyruk yöneticisine ilişkin bir günlük içerir; günlüğün adı QM *GRLIB/AMQ A JRN*, burada QM *GRLIB* kuyruk yöneticisi kitaplığının adıdır; A ise, tek bir yönetim ortamı kuyruk yöneticisi durumunda, kuyruk yöneticisi yönetim ortamı için benzersiz olan bir harftir (A).

QM *GRLIB* takes the name QM, followed by the name of the queue manager in a unique form. Örneğin, TEST adlı bir kuyruk yöneticisinin QMTEST adlı bir kuyruk yöneticisi kitaplığı vardır. Kuyruk yöneticisi kitaplığı, **CRTMQM** komutu kullanılarak bir kuyruk yöneticisi yaratılırken belirtilebilir.

Günlüklerde, günlüğe kaydedilmekte olan bilgileri içeren günlük nesnelere ilişkilendirilir. Alıcılar, bilgilerin yalnızca eklenebileceği ve sonunda doldurabileceği nesnelere.

Günlük nesnelere, eski bilgileri içeren değerli disk alanı kullanır. Ancak, bu sorunu en aza indirmek için bilgileri kalıcı depolama alanına yerleştirebilirsiniz. Günlüğe bir günlük nesnesi belirli bir zamanda eklenir. Günlük nesnesi önceden belirlenmiş eşik büyüklüklerine ulaşırsa, bu değer ayırdır ve yeni bir günlük nesnesiyle değiştirilir. **CRTMQM** ve **THRESHOLD** parametresini kullanarak bir kuyruk yöneticisi yarattığınızda günlük nesnelere eşiklerini belirtebilirsiniz.

Yerel IBM MQ for IBM i günlük nesnelere ilişkili günlük nesnelere, her kuyruk yöneticisi kitaplığında bulunur ve aşağıdaki gibi bir adlandırma kuralını benimleyebilir:

```
AMQ Arnnnnn
```

burada:

A

A-Z bir harftir. Tek yönetim ortamı kuyruk yöneticileri için A . Bu, çok eşgörunümlü bir kuyruk yöneticisinin farklı eşgörunümlere göre değişir.

nnnnn

sırada bir sonraki günlük için 1 artırılan 00000 to 99999 ondalık sayısıdır.

cusu

Bir günlük nesnesi her geri yüklendiğinde 1 artırılarak 0 to 9 ondalık sayısıdır.

Günlüklerin sırası tarih temelinde. Ancak, sonraki günlüğün adlandırıldığı yer aşağıdaki kurallara dayalıdır:

1. AMQAınnnnn , AMQAı (nnnnn+1) 'a gider ve nnnnn 99999' a eriştiğindewraps' a gider. Örneğin, AMQA099999 , AMQA000000değerine, AMQA999999 ise AMQA900000' a gider.
2. Kural 1 tarafından üretilen bir adı taşıyan bir günlük zaten varsa, QSYSOPR ileti kuyruğuna CPI70E3 iletisi gönderilir ve otomatik alıcı geçişi durdurulur.

Sorunu incelemeniz ve el ile yeni bir günlük nesnesi ekleninceye kadar, şu anda bağlı olan günlük nesnesi kullanılmaya devam eder.

3. Sıra içinde yeni bir ad yoksa (yani, tüm olası günlük adları sistemde bulunur) aşağıdakilerden her ikisini de yapmanız gerekir:
 - a. Artık silmeye gerek kalmaması (bkz. "IBM üzerinde günlük yönetimi" sayfa 363).
 - b. Günlük değişikliklerini, (**RCDMQMIMG**) kullanarak en son günlük nesnesiyle kaydedin. daha sonra önceki adımı tekrarlayın. Bu, eski günlük nesnesi adlarının yeniden kullanılmasına olanak tanır.

AMQAJRN günlüğü, eşik değerine ulaşıldığında işletim sisteminin günlük nesnelerini otomatik olarak değiştirmesini sağlamak için MNGRCV (*SYSTEM) seçeneğini kullanır. Sistemin alıcıları nasıl yönettiği ile ilgili daha fazla bilgi için *IBM i Yedekleme ve Kurtarmabaşlıklı* konuya bakın.

Günlük nesnesinin varsayılan eşik değeri 100.000 KB 'dir. Kuyruk yöneticisini yaratırken bu değeri daha büyük bir değer olarak ayarlayabilirsiniz. LogReceiverSize özneliğinin başlangıç değeri, mqs.ini dosyasının LogDefaults kısmına yazılır.

Bir günlük nesnesi belirlenen eşik değerin ötesine geçtiğinde, günlük nesnesi ayrılır ve yeni bir günlük nesnesi yaratılır ve önceki günlük nesnesinden öznelikleri devralır. Sistem otomatik olarak yeni bir günlük nesnesi bağlandığında, kuyruk yöneticisi yaratıldıktan sonra LogReceiverSize ya da LogASP özneliklerinde yapılan değişiklikler dikkate alınmaz.

Sistemin yapılandırılmasına ilişkin ek bilgi için [IBM i üzerinde yapılandırma bilgilerinin değiştirilmesi](#) başlıklı konuya bakın.

If you need to change the size of journal receivers after the queue manager has been created, create a new journal receiver and set its owner to QMQM using the following commands:

```
CRTJRNRCV JRNRCV(QM GRLIB/AMQ Arnnnnn) THRESHOLD(xxxxxx) +  
TEXT('MQM LOCAL JOURNAL RECEIVER')  
CHGOBJOWN OBJ(QM GRLIB/AMQ Arnnnnn) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
```

burada:

QMGLIB

Kuyruk yöneticisi kitaplığınızın adı

A

Eşgörünüm tanıtıcısıdır (genellikle A).

rrnnnnn

Daha önce açıklanan adlandırma sırasında bir sonraki günlük nesnesi

xxxxxx

Yeni günlük nesnesi eşik değeri mi (KB cinsinden)

Not: Alıcı büyüklüğü üst sınırı işletim sistemi tarafından yönetilir. Bu değeri denetlemek için **CRTJRNRCV** komutundaki THRESHOLD anahtar sözcüğünde yer alan bir değer olup olmadığını denetleyin.

Şimdi, yeni alıcısını AMQAJRN günlüğüne şu komutla bağlayın:

```
CHGJRN JRN(QMGLIB/AMQ A JRN) JRNRCV(QMGLIB/AMQ Annnnnn)
```

Bu günlük nesnelere nasıl yönetileceği hakkında ayrıntılı bilgi için bkz. [“IBM üzerinde günlük yönetimi”](#) sayfa 363 .

IBM i Queue manager journal usage on IBM i

IBM MQ for IBM i ' un, yerel nesnelere yapılan güncellemeleri denetlemek için kullandığı günlüklerin nasıl kullandığını anlamak için bu bilgileri kullanın.

İleti kuyruklarına ilişkin kalıcı güncellemeler iki aşamada gerçekleşir. Güncellemeyi temsil eden kayıtlar önce günlüğe yazılır, sonra kuyruk dosyası güncellenir.

Bu nedenle günlük nesnelere, kuyruk dosyalarından daha güncel bir tarihe dönüşebilir. Yeniden başlatma işlemlerinin 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 noktası kendisi, kuyruk yöneticisini yeniden başlatmak için gereken günlük kaydı dizilerinden 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 kapatıldığında ve belirli sayıda işlem günlüğe kaydedildikten sonra alınır.

Bir kuyruk yöneticisini, kuyruk yöneticilerindeki tüm nesnelere yönelik RCDMQMIMG komutunu vererek ve sonuçları aşağıdaki gibi görüntüleyerek bir denetim noktası almak için zorlayabilirsiniz:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNTA(*YES)
```

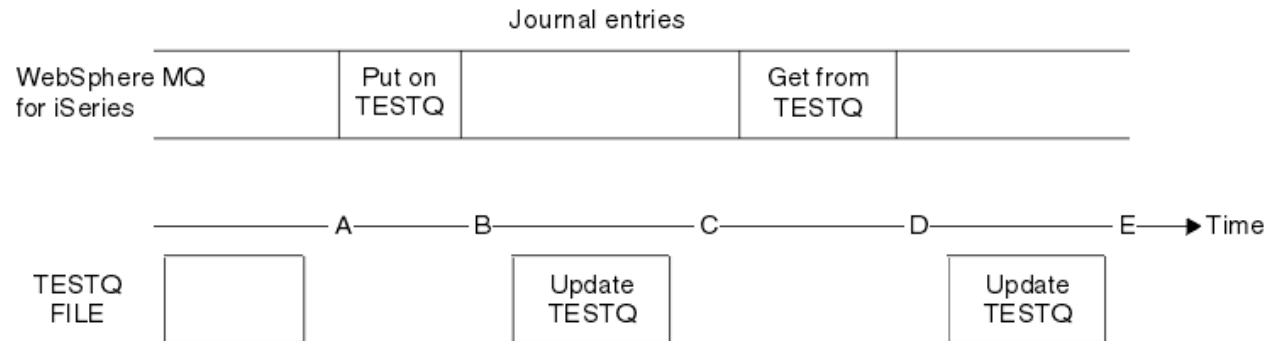
Kuyruklar daha fazla ileti işledikçe, denetim noktası kaydı kuyrukların yürürlükteki durumuyla tutarsız hale gelir.

IBM MQ yeniden başlatıldığında, günlükteki en son denetim noktası kaydını bulur. Bu bilgiler, her denetim noktasının sonunda güncellenen denetim noktası (checkpoint) dosyasında tutulur. Denetim noktası kaydı, günlük ile veri arasındaki en son tutarlılık noktasını temsil eder. Bu denetim noktasındaki veriler, kuyrukları denetim noktası zamanında var olduğu şekilde yeniden oluşturmak için kullanılır. Kuyruklar yeniden yaratıldığında, günlük kuyrukları, sistem arızası ya da kapanma işleminden önce girdikleri duruma geri getirmek için ileri doğru yürütülmektedir.

IBM MQ ' un günlüğü nasıl kullandığını anlamak için, TESTkuyruk yöneticisinde TESTQ adlı yerel bir kuyruğun vakasını göz önünde bulundurun. Bu, IFS dosyası tarafından temsil edilir:

```
/QIBM/UserData/mqm/qmgrs/TEST/queues
```

Belirtilen bir ileti bu kuyruğa konursa ve kuyruktan alındıysa, gerçekleşen işlemler Şekil Şekil 25 sayfa 360' da gösterilir.



Şekil 25. MQM nesnelere güncellerken olayların sırası

Çizgede gösterilen beş nokta (A-E-E), aşağıdaki durumları tanımlayan zaman noktalarını temsil eder:

A

Kuyruğun IFS dosyası gösterimi, günlüğün içerdiği bilgilerle tutarlı olur.

B

Kuyruktaki Put işlemini tanımlayan günlüğe bir günlük girişi yazılır.

C

Kuyruğa uygun güncelleme yapılır.

D

Kuyruktan bir Get işlemi tanımlamaya ilişkin günlüğe bir günlük girişi yazılır.

E

Kuyruğa uygun güncelleme yapılır.

The key to the recovery capabilities of IBM MQ for IBM i is that the user can save the IFS file representation of TESTQ as at time A, and subsequently recover the IFS file representation of TESTQ as at time E, by restoring the saved object and replaying the entries in the journal from time A onwards.

Bu strateji, sistem hatasından sonra kalıcı iletileri kurtarmak için IBM MQ for IBM i tarafından kullanılır. IBM MQ , günlük nesnelere belirli bir girişi anımsar ve başlangıçta bu noktadan itibaren günlüklerdeki girişleri yeniden oynatmasını sağlar. Bu başlatma girdisi düzenli olarak yeniden hesaplanır, böylece IBM MQ yalnızca sonraki başlatma sırasında gerekli olan en düşük yinelemeyi gerçekleştirmesi gerekir.

IBM MQ , nesnelere bireysel olarak kurtarılması sağlar. Bir nesnenle ilgili tüm kalıcı bilgiler yerel IBM MQ for IBM i günlüklerinde kaydedilir. Zarar görmüş ya da bozuk hale gelen herhangi bir IBM MQ nesnesi, günlük içinde tutulan bilgilerden tamamen yeniden oluşturulabilir.

Sistemin günlük nesnelere nasıl yönetmesine ilişkin ek bilgi için bkz. [“Availability, backup, recovery, and restart on IBM i” sayfa 358.](#)

IBM i **IBM üzerindeki ortam görüntüleri**

On IBM i, a media image is a complete copy of an IBM MQ object that is recorded in the journal. Bazı bozuk ya da zarar görmüş nesnelere ortam görüntülerinden otomatik olarak kurtarılabilir.

Uzun süreli bir IBM MQ nesnesi, yaratıldığı noktaya geri dönerek çok sayıda günlük girişi gösterebilir. Bunu önlemek için IBM MQ for IBM i , bir nesnenin ortam görüntüsü kavramını içerir.

Bu ortam görüntüsü, günlüğe kaydedilen IBM MQ nesnesinin tam bir kopyasıdır. Bir nesnenin görüntüsü alınır, nesne bu görüntüden başlayarak günlük girişleri yeniden oluşturularak yeniden oluşturulabilir. Her bir IBM MQ nesnesi için yeniden yürütme noktasını temsil eden günlükteki giriş, ortam kurtarma girdisi olarak anılır. IBM MQ aşağıdakilerin kaydını tutar:

- Her kuyruk yöneticisi nesnesi için ortam kurtarma girişi.
- Bu setin içindeki en eski girdi (ayrıntılar için [“IBM üzerinde günlük yönetimi” sayfa 363](#) içindeki AMQ7462 hata iletilerinde bakın).

*CTLG nesnesinin ve *MQM nesnesinin görüntüleri düzenli olarak alınır; çünkü bu nesnelere kuyruk yöneticisinin yeniden başlatılması için çok önemlidir.

Uygun olduğunda diğer nesnelere resimleri alınır. Varsayılan olarak, ENDCCTJOB (*YES) parametresiyle **ENDMQM** komutu kullanılarak bir kuyruk yöneticisi sona erdirildiğinde, tüm nesnelere görüntüleri alınır. Bu işlem, çok büyük kuyruk yöneticileri için önemli ölçüde zaman alabilir. Hızlı bir şekilde sona erdirmeniz gerekiyorsa, ENDCCTJOB (*YES) parametresiyle RCDMQMIMG (*NO) parametresini belirleyin. Bu tür durumlarda, aşağıdaki komutu kullanarak, kuyruk yöneticisi yeniden başlatıldıktan sonra günlüklerde tam bir ortam görüntüsü kaydetmenizi önerilir:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME)
```

IBM MQ , nesnenin görüntüsünü, günlüğün küçük bir girişi tarafından tam olarak tanımlanabileceği uygun bir nokta bulursa otomatik olarak bir nesnenin görüntüsünü kaydeder. Ancak, bazı nesnelere için bu durum hiçbir zaman gerçekleşmeyebilir; örneğin, sürekli olarak çok sayıda ileti içeren kuyruklar.

Uzun bir süre boyunca devam etmek üzere en eski ortam kurtarma girdisinin tarihine izin vermektense, seçilen nesnelerin görüntüsünü el ile gerçekleştirebilmenizi sağlayan IBM MQ komutunu RCDMQMIMGkomutunu kullanın.

Ortam görüntülerinden kurtarma

IBM MQ , bozuk ya da zarar görmüş olduğu saptandıysa, bazı nesneleri otomatik olarak ortam görüntülerinden kurtarır. Bu durum özellikle, olağan kuyruk yöneticisi başlatma bölümünün bir parçası olarak özel *MQM ve *CTLG nesneleri için geçerlidir. Kuyruk yöneticisinin son kapatma işlemi sırasında herhangi bir syncpoint işlemi tamamlanmadıysa, başlatma işlemini tamamlamak için, etkilenen herhangi bir kuyruk da otomatik olarak kurtarılır.

You must recover other objects manually, using the IBM MQ command RCRMQMOBJ. Bu komut, IBM MQ nesnesini yeniden yaratmak için günlükteki girişleri yeniden yürütür. Bir IBM MQ nesnesi zarar görse, tek geçerli işlemler bu nesneyi silmeniz ya da bu yöntemle yeniden yaratmak olmalıdır. Ancak, kalıcı olmayan iletilerin bu şekilde kurtarılmadığını göz önünde bulundurun.

IBM i **IBM MQ for IBM üzerindeki denetim noktaları**

Denetim noktaları, kurtarma işlemi için bilinen tutarlı bir başlangıç noktası sağlamak üzere çeşitli zamanlarda alınır.

The checkpoint thread within process AMQZMUC0 is responsible for taking the checkpoint at the following points:

- Kuyruk yöneticisi başlangıcı (STRMQM).
- Kuyruk yöneticisi kapatıldı (ENDMQM).
- Son denetim noktasından bu yana geçen bir süre geçtikten sonra (varsayılan dönem 30 dakikadır) ve önceki denetim noktasından bu yana en az sayıda günlük kaydı yazılıdır (varsayılan değer 100 'dür).
- Bir dizi günlük kaydı yazıldıktan sonra. Varsayılan değer 10 000 'dir.
- Günlük eşiği büyüklüğü aşıldıktan sonra yeni bir günlük nesnesi otomatik olarak yaratılmıştır.
- Tam bir ortam görüntüsü aşağıdaki gibi alındığında:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNDTA(*YES)
```

IBM i **IBM MQ for IBM i verilerinin yedekleri**

Her kuyruk yöneticisi için iki tip IBM MQ yedeğini anlamak için bu bilgileri kullanın.

Her kuyruk yöneticisi için dikkate alınacak iki tip IBM MQ yedeklemesi vardır:

- Veri ve günlük yedekleme.

Her iki veri kümesinin de tutarlı olmasını sağlamak için, kuyruk yöneticisini sona erdirdikten sonra bu işlemi gerçekleştirin.

- Günlük yedekleme.

Bunu, kuyruk yöneticisi etkinken yapabilirsiniz.

Her iki yöntem için de kuyruk yöneticisi IFS dizininin ve kuyruk yöneticisi kitaplığının adlarını bulmanız gerekir. Bunları IBM MQ yapılandırma dosyasında (mq5.ini) bulabilirsiniz. Ek bilgi için [The QueueManager stanzabaşlıklı konuyu](#) bakın.

Her iki yedekleme tipini de yapmak için aşağıdaki yordamları kullanın:

Belirli bir kuyruk yöneticisinin verileri ve günlük yedeklemesi

Not: Kuyruk yöneticisi çalışırken saklama sırasında etkin istek kullanmayın. Bekleyen değişikliklerle ilgili tüm kesinleştirme tanımları kesinleştirilmedikçe ya da geriye işlenmedikçe, böyle bir istek tamamlanamaz. Bu komut, kuyruk yöneticisi etkin olduğunda kullanılırsa, kanal bağlantıları olağan bir şekilde sona ermeyebilir. Her zaman aşağıdaki yordamı kullanın.

1. Şu komutu kullanarak boş bir günlük nesnesi yaratın:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. Tüm IBM MQ nesnelere ilişkin bir MQM görüntüsünü kaydetmek için **RCDMQMIMG** komutunu kullanın ve daha sonra, komutu kullanarak bir denetim noktasını zorlayın:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. Kanalları sona erdirin ve kuyruk yöneticisinin çalışmadığından emin olun. Kuyruk yöneticiniz çalışıyorsa, bunu **ENDMQM** komutuyla durdurun.
4. Aşağıdaki komutu girerek kuyruk yöneticisi kitaplığını yedekleyin:

```
SAVLIB LIB(QMTEST)
```

5. Kuyruk yöneticisi IFS dizinlerini yedekleyerek aşağıdaki komutu çalıştırın:

```
SAV DEV(...) OBJ(('QIBM/UserData/mqm/qmgrs/test'))
```

Belirli bir kuyruk yöneticisinin günlük yedeklemesi

Günlüklerde ilgili tüm bilgiler tutulduğu için, bir kerede tam bir saklama işlemi gerçekleştirdiğiniz sürece, günlük nesnelere saklanarak kısmi yedekler gerçekleştirilebilir. Bu kayıt, tam yedekleme zamanından bu yana yapılan tüm değişiklikleri kaydeder ve aşağıdaki komutların yayınlanarak gerçekleştirilmesini sağlar:

1. Şu komutu kullanarak boş bir günlük nesnesi yaratın:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. Tüm IBM MQ nesnelere ilişkin bir MQM görüntüsünü kaydetmek için **RCDMQMIMG** komutunu kullanın ve daha sonra, komutu kullanarak bir denetim noktasını zorlayın:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. Komutu kullanarak günlük nesnelere saklamak için aşağıdaki komutu kullanın:

```
SAVOBJ OBJ(AMQ*) LIB(QMTEST) OBJTYPE(*JRNRCV) .....
```

Basit bir yedekleme stratejisi, her hafta IBM MQ kitaplıklarının tam yedeğini gerçekleştirmekte ve günlük yedekleme işlemi gerçekleştirmektedir. Bu, tabii ki, işletmenize ilişkin yedekleme stratejinizi nasıl ayarlamaya bağlıdır.

IBM i **IBM üzerinde günlük yönetimi**

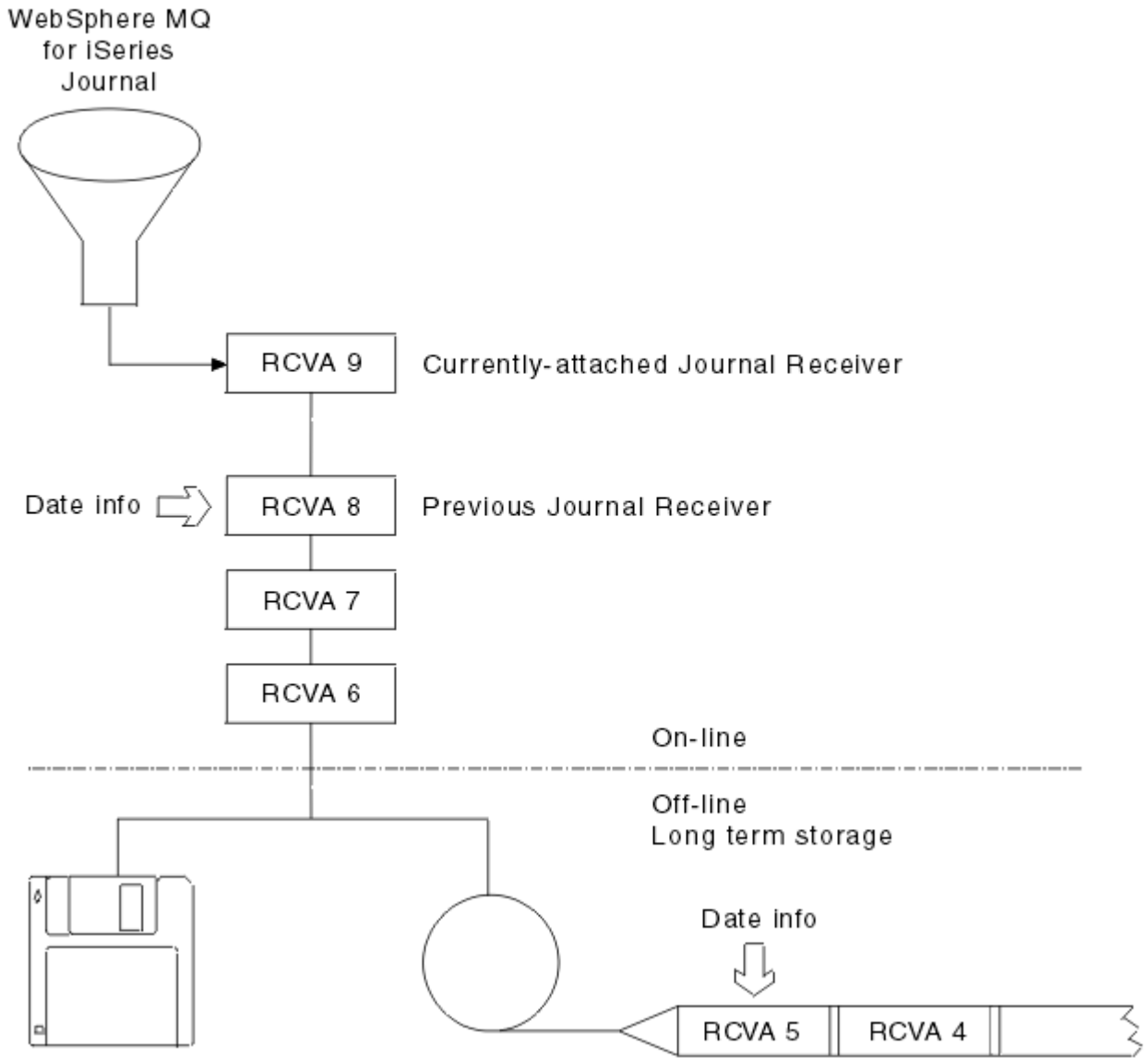
Yedekleme stratejinizin bir parçası olarak, günlük nesnelere göz kulak olun. Çeşitli nedenlerden dolayı, günlük nesnelere ilişkin IBM MQ kitaplıklarından kaldırılması yararlı olur:

- Alanı serbest bırakmak için; bu, tüm günlük nesnelere için geçerlidir.
- Başlatma sırasında performansı artırmak için (STRMQM)
- Nesnelere yeniden yaratma başarımını yükseltmek için (RCRMQMOBJ)

Bir günlük nesnesini silmeden önce, yedek kopyasına sahip olduğunuz ve artık günlük nesnesine gerek kalmadığına dikkat etmeniz gerekir.

Günlük nesnelere, bir kurtarma işlemi için gerekiyorsa, geri yükleme için kullanılabilir duruma getirilmeleri koşuluyla, günlüklerden ayrılıp saklandıktan sonra kuyruk yöneticisi kitaplığından kaldırılabilir.

Günlük yönetimi kavramı [Şekil 26 sayfa 364](#) içinde gösterilir.



Şekil 26. IBM itarihinde günlük kaydı

Yedeklenen bir günlük nesnesinin kuyruk yöneticisi kitaplığından ne zaman kaldırılabilceğini ve yedeklemenin ne zaman atılabileceğini saptamak için, IBM MQ günlüklerinde ne kadar geriye gidilmesi gerektiğini bilmek önemlidir.

IBM MQ , bu süreyi belirlemeye yardımcı olması için kuyruk yöneticisi ileti kuyruğuna (kuyruk yöneticisi kitaplığındaki QMQMSG) iki ileti gönderir. Bu iletiler, yerel bir günlük nesnesini değiştirdiğinde ve bir denetim noktasını zorlamak için RCDMQIMG kullandığında yayınlanır. İki ileti:

AMQ7460

Başlatma kurtarma noktası. This message defines the date and time of the startup entry from which IBM MQ replays the journal in the event of a startup recovery pass. Bu kaydı içeren günlük nesnesi IBM MQ kitaplıklarında bulunuyorsa, bu ileti aynı zamanda kaydı içeren günlük nesnesinin adını da içerir.

AMQ7462

En eski ortam kurtarma girişi. Bu ileti, bir nesneyi ortam görüntüünden yeniden yaratmak için kullanılacak en eski girişin tarihini ve saatini tanımlar.

Tanımlanan günlük nesnesi, gereken en eski günlük nesnesidir. Daha eski yaratma tarihlerine sahip başka herhangi bir IBM MQ günlük nesnesi artık gerekli değildir. Yalnızca yıldızlar görüntüleniyorsa, yedeklerin en eski günlük nesnesi olan saptanması için belirtilen tarihten geri yüklenmesi gerekir.

Bu iletiler günlüğe kaydedildiğinde, IBM MQ , yalnızca tek bir giriş içeren kuyruk yöneticisi kitaplığına bir kullanıcı alanı nesnesi de yazar: Sistemde alınması gereken en eski günlük nesnesinin adı. Bu kullanıcı alanına AMQJRNINF adı verilir ve veriler şu biçimde yazılır:

```
JJJJJJJJJLLLLLLLLLLLLLLYYMMDDHHMSSmmm
```

Burada:

JJJJJJJJJ

IBM MQ ' un hala gereksinim duyduğu en eski günlük nesnesidir.

LLLLLLLLLLL

Günlük nesnesi kitaplığı adıdır.

YYYY

IBM MQ ' un gereksinim duyduğu en eski günlük girdisidir.

MM

IBM MQ ' un gereksinim duyduğu en eski günlük girdisinin aysıdır.

DD

IBM MQ ' un gereksinim duyduğu en eski günlük girdisinin günüdür.

HH

IBM MQ ' un gereksinim duyduğu en eski günlük girdisinin saatidir.

SS

IBM MQ ' un gereksinim duyduğu en eski günlük girdisinin saniyeleri.

mmm

IBM MQ ' in gereksinim duyduğu en eski günlük girdisinin milisaniyesidir.

Sistemden en eski günlük nesnesi silindiğinde, bu kullanıcı alanı günlük nesnesi adı için yıldız işareti (*) içerir.

Not: RCDMQIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) ' un düzenli olarak gerçekleştirilmesi, IBM MQ için başlatma zamanını kaydedebilir ve kurtarma için kaydetmeniz ve geri yüklemeniz gereken yerel günlük nesnelerinin sayısını azaltabilir.

IBM MQ for IBM i , başlatma için ya da bir nesneyi yeniden yaratmak için kurtarma geçişi gerçekleştirmediği sürece günlük nesnelere başvuramaz. If it finds that a journal it requires is not present, it issues message AMQ7432 to the queue manager message queue (QMOMMSG), reporting the time and date of the journal entry it requires to complete the recovery pass.

Bu gerçekleşirse, yedekten bu tarihten sonra ayrılmış olan tüm günlük nesnelerini, kurtarma başarılı tarafından başarılı bir şekilde geçirmesine izin vermek için geri yükleyin.

Günlük nesnesini, başlangıç girişini içeren günlük nesnesini ve kuyruk yöneticisi kitaplığında bulunan sonraki günlük nesnelerini de içeren günlük nesnelerini saklayın.

Günlük nesnesini, en eski Media Recovery Entry günlük nesnelerini ve sonraki tüm günlük nesnelerini, kuyruk yöneticisi kitaplığında var olan ya da yedeklenen günlük nesnelerini saklayın.

Bir denetim noktasını zorlarken:

- AMQ7460 ' da adlandırılan günlük nesnesi gelişmiş değilse, bu, kesinleştirilmesinin ya da geri döndürülmesi gereken tamamlanmamış bir iş birimi olduğunu gösterir.
- AMQ7462 içinde belirtilen günlük nesnesi gelişmiş değilse, bir ya da daha çok zarar görmüş nesne olduğunu gösterir.

IBM üzerindeki tam bir kuyruk yöneticisini (veri ve günlükler) geri yükleme

Bir yedekten ya da uzak makineden bir ya da daha çok kuyruk yöneticisini geri yüklemek için bu bilgileri kullanın.

Bir yedekten bir ya da daha çok IBM MQ kuyruk yöneticisini kurtarmanız gerekiyorsa, aşağıdaki adımları gerçekleştirin.

1. IBM MQ kuyruk yöneticilerini susturma.
2. En son tam yedeğinizden oluşan ve daha sonra günlük nesnelere oluşan en son yedekleme kümelerinizi bulun.
3. IBM MQ veri kitaplıklarını tam yedeklemeden durumuna geri yüklemek için, tam yedekten bir RSTLIB işlemi gerçekleştirin ve aşağıdaki komutları yürütün:

```
RSTLIB LIB(QMQRLIB1) .....
RSTLIB LIB(QMQRLIB2) .....
```

Bir günlük nesnesi bir günlük yedeklemede kısmen saklandıysa ve sonraki yedeklemede tam olarak saklandıysa, yalnızca tam olarak saklanmış olan bir günlük nesnesi geri yüklenir. Günlükleri kronolojik sırada tek tek geri yükleyin.

4. IBM MQ IFS dizinlerini IFS kütük sistemine geri yüklemek için aşağıdaki komutu kullanarak RST işlemi gerçekleştirin:

```
RST DEV(...) OBJ('/QIBM/UserData/mqm/qmgrs/testqm') ...
```

5. İleti kuyruğu yöneticisini başlatın. Tüm yedekleme tarihinden bu yana yazılan tüm günlük kayıtlarını yeniden yürütür ve tüm IBM MQ nesnelere, günlük yedekleme sırasında tutarlı duruma geri yükler.

Eksiksiz bir kuyruk yöneticisini farklı bir makinede geri yüklemek istiyorsanız, kuyruk yöneticisi kitaplığından her şeyi geri yüklemek için aşağıdaki yordamı kullanın. (Örnek kuyruk yöneticisi adı olarak TEST ' i kullanınız.)

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 üzerinde belirli bir kuyruk yöneticiye ilişkin günlük nesnelere geri yüklenmesi**

Günlük nesnelere geri yüklenmesine ilişkin farklı yolları anlamak için bu bilgileri kullanın.

En yaygın işlem, kaldırılan bir günlük nesnesini kuyruk yöneticisi kitaplığına geri yüklemek olur; kaldırılan bir günlük nesnesi, sonraki bir kurtarma işlevi için yeniden gereklidir.

Bu basit bir görevdir ve günlük nesnelere standart IBM i RSTOBJ komutu kullanılarak geri yüklenmesini gerektirir:

```
RSTOBJ OBJ(QMQMDATA/AMQA000005) OBJTYPE(*JRNCV) .....
```

Günlük nesnelere, tek bir günlük nesne yerine geri yüklenmesi gerekebilir. Örneğin, AMQA000007 , IBM MQ kitaplıklarındaki en eski günlük nesnedir ve hem AMQA000005 hem de AMQA000006 ' in geri yüklenmesi gerekir.

Bu durumda, nesnelere tek tek ters kronolojik sırayla geri yükleyin. Bu her zaman gerekli değildir, ama iyi bir uygulamadır. Önemli durumlarda, geri yüklenen günlük nesnelere günlükle ilişkilendirmek için IBM i komutunu WRKJRNA komutunu kullanmanız gerekebilir.

Günlükleri geri yüklerken, sistem, günlük nesne sırasında yeni bir ad içeren bağlı bir günlük nesne yaratır. Ancak, yeni ad üretilen yeni ad, geri yüklemeniz gereken bir günlük nesnesiyle aynı olabilir. Bu sorunun üstesinden gelmesi için el ile araya girilmesi gerekir; yeni bir günlük nesne ve günlük nesne geri yüklenmeden önce yeni bir günlük nesne yaratmak için gereklidir.

Örneğin, saklanmış AMQAJRN günlüğüyle ve aşağıdaki günlük nesnelere ilgili sorunu göz önünde bulundurun:

- AMQA000000
- AMQA100000
- AMQA200000
- AMQA300000
- AMQA400000
- AMQA500000
- AMQA600000
- AMQA700000
- AMQA800000
- AMQA900000

AMQAJRN günlüğünü kuyruk yöneticisi kitaplığına geri yüklerken, sistem otomatik olarak AMQA000000 günlük nesnesini yaratır. Bu otomatik olarak oluşturulan günlük nesne, geri yükleyemediğiniz, geri yüklemek istediğiniz var olan günlük nesnelere (AMQA000000) çakışmayla çakışıyor.

Çözüm:

1. Sonraki günlük nesnesini el ile yaratın (bkz. [“Queue manager journals on IBM i” sayfa 358](#)):

```
CRTJRNRCV JRNRCV(QMGRLIB/AMQA900001) THRESHOLD(XXXXX)
```

2. Günlük nesnesiyle günlüğü el ile yaratın:

```
CRTJRN JRN(QMGRLIB/AMQAJRN) MNGRCV(*SYSTEM) +  
JRNRCV(QMGRLIB/AMQA900001) MSGQ(QMGRLIB/AMQAJRNMSG)
```

3. Restore the local journal receivers AMQA000000 to AMQA900000.

IBM i

IBM üzerinde çok eşgözümlü kuyruk yöneticileri

Çok eşgözümlü kuyruk yöneticileri, etkin sunucu başarısız olursa yedek sunucuya otomatik olarak geçiş yaparak kullanılabilirliği artırır. Etkin ve beklemedeki sunucular, aynı kuyruk yöneticisinin birden çok örneğidir; aynı kuyruk yöneticisi verilerini paylaşırlar. Etkin yönetim ortamı başarısız olursa, kuyruk yöneticisinin kuyruklarını yeniden oluşturabilmesi için, günlüğünü yedekte bekleyen yedek veritabanına aktarmanız gerekir.

Çok eşgözümlü kuyruk yöneticilerini çalıştırdığınız IBM i sistemlerini yapılandırın. Böylece, etkin kuyruk yöneticisi yönetim ortamı başarısız olursa, kullanmakta olduğu günlük, devralan yedek yönetim ortamı tarafından kullanılabilir. Kendi yapılandırma ve yönetim görevlerinizi, günlüğü, devralan eşgözümlüde bulunan etkin eşgözümlüden yapmak için tasarlayabilirsiniz. İletileri kaybetmek istemiyorsanız, tasarımınızın yedek günlüğün, hata noktasındaki etkin günlükle tutarlı olduğundan emin olmanız gerekir.

Tasarımınızı, tutarlılığı sürdürmekte olan sonraki konulardaki örneklerle tanımlanan iki yapılandırmadan uyarlayabilirsiniz.

1. Günlüğü, etkin kuyruk yöneticisi yönetim ortamını, beklemedeki yönetim ortamları çalıştıran sistemlere çalıştıran sistemden ikizle.
2. Günlüğü, etkin yönetim ortamını bir yedek yönetim ortamına çalıştıran sistem tarafından aktarılan bir Bağımsız Yardımcı Bellek Havuzu (IASP) içine yerleştirin.

İlk çözüm, temel ASP ' leri kullandığı için ek donanım ya da yazılım gerektirmez. İkinci çözüm, ayrı olarak fiyatlandırılmış bir IBM i Lisans Ürünü 5761-SS1 Seçenek 41 olarak kullanılabilen IBM i kümeleme desteğine gereksinim olan değişimli kullanılabilir IASP ' leri gerektirir.

IBM i **IBM üzerinde güvenilirlik ve kullanılabilirlik**

Çok eşgörünümlü kuyruk yöneticileri uygulamaların kullanılabilirliğini geliştirmeyi hedefler. Teknolojik ve fiziksel kısıtlar, olağanüstü durumdan kurtarma taleplerini karşılamak için farklı çözümlere gereksinim duyarsınız, kuyruk yöneticilerini yedeklemeniz ve sürekli işlem yapmak.

Güvenilirlik ve kullanılabilirlik için yapılandırılırken, çok sayıda faktörü pazarladığınızda, dört ayrı tasarım noktası elde edilmiştir:

Olağanüstü durumdan kurtarma

Tüm yerel varlıklarınızı yok eden büyük bir olağanüstü durumdan sonra kurtarma için optimize edilmiştir.

Disaster recovery on IBM i is often based on geographic mirroring of IASP.

Yedekle

Yerelleştirilmiş bir arızadan, genel olarak bir insan hatasından ya da bazı öngörülemeyen bir teknik sorun sonrası kurtarma için iyileştirilmiştir.

IBM MQ , kuyruk yöneticilerini düzenli olarak yedeklemek için yedek kuyruk yöneticileri sağlar. Ayrıca, yedeklemenin para birimini geliştirmek için kuyruk yöneticisi günlüklerinin zamanuyumsuz eşlemesini de kullanabilirsiniz.

Kullanılabilirlik

Sunucu ya da disk arızası gibi öngörülebilir teknik arızalar sonrasında, neredeyse kesintisiz bir hizmetin ortaya çıkmasını sağlayan işlemleri geri yüklemek üzere optimize edilmiştir.

Kurtarma işlemi genellikle dakika cinsinden ölçülür ve algılama bazen kurtarma işleminden daha uzun sürer. Çok eşgörünümlü bir kuyruk yöneticisi, *kullanılabilirlik* için yapılandırılırken size yardımcı olur.

Sürekli işlem

Kesintisiz bir hizmet sağlamak için optimize edilmiştir.

Sürekli çalışma çözümleri, algılama sorununu çözer ve neredeyse her zaman aynı işi birden fazla sisteme teslim eder ve ya ilk sonucu kullanırsa ya da doğruluk, en az iki sonucu karşılaştırarak önemli bir değerlendirmeye sahiptir.

Çok eşgörünümlü bir kuyruk yöneticisi, *kullanılabilirlik* için yapılandırılırken size yardımcı olur. Kuyruk yöneticisinin bir eşgörünümü bir kerede etkin olur. Bir standby eşgörünümünün değiştirilmesi, sistemin nasıl yapılandırıldığı, yüklendiği ve ayarlanana bağlı olarak on saniyeden on saniyeden fazla ya da daha fazla süreye kadar sürer.

A multi-instance queue manager can give the appearance of a nearly uninterrupted service if used with reconnectable IBM MQ MQI clients, which are able to continue processing without the application program necessarily being aware of a queue manager outage; see the topic [Otomatikleştirilmiş istemci yeniden bağlantısı](#).

IBM i **IBM üzerinde yüksek kullanılabilirlik çözümünün bileşenleri**

Kuyruk yöneticisi verileri için güçlü bir ağ depolama alanı, kuyruk yöneticisi günlükleri için günlük eşleme ya da güçlü IASP depolaması sağlayarak ve yeniden başlatılabilir kuyruk yöneticisi hizmetleri olarak yapılandırılmış uygulamaların yeniden bağlanabilir istemcileri için güçlü bir ağ oluşturma depolaması

sağlayarak çok eşgörünümlü kuyruk yöneticilerini kullanarak yüksek düzeyde kullanılabilirlik çözümü oluşturun.

Çok eşgörünümlü bir kuyruk yöneticisi, başka bir sunucudaki başka bir kuyruk yöneticisi yönetim ortamının başlatılışına devam ederek kuyruk yöneticisi hatasının saptanmasını sağlar. Başlatma işlemini tamamlamak için, yönetim ortamının ağ üzerinde çalışan depolama alanındaki paylaşılan kuyruk yöneticisi verilerine ve yerel kuyruk yöneticisi günlüğünün kopyasına erişmesi gerekir.

Yüksek düzeyde kullanılabilirlik çözümü yaratmak için kuyruk yöneticisi verilerinin kullanılabilirliğini, yerel kuyruk yöneticisi günlüğünün para birimini ve yeniden bağlanabilir istemci uygulamaları oluşturmanızı ya da kuyruk yöneticisi devam ettiğinde, uygulamalarınızı otomatik olarak yeniden başlatılacak kuyruk yöneticisi hizmetleri olarak konuşturmanız gerekir. Otomatik istemci yeniden bağlanması IBM MQ classes for Javatarafından desteklenmez.

Kuyruk yöneticisi verileri

Kuyruk yöneticisi verilerini paylaşılan, yüksek düzeyde kullanılabilir ve büyük olasılıkla RAID düzey 1 ya da daha büyük bir disk kullanarak paylaşılan, yüksek düzeyde kullanılabilir ve güvenilir bir depolama alanına yerleştirin. Dosya sisteminin, çok eşgörünümlü kuyruk yöneticileri için paylaşılan bir dosya sistemine ilişkin gereksinimleri karşılamaya gereksinimi vardır; paylaşılan dosya sistemlerine ilişkin gereksinimlerle ilgili daha fazla bilgi için [Paylaşılan dosya sistemlerine ilişkin gereksinimler](#) başlıklı konuya bakın. Ağ Dosya Sistemi 4 (NFS4), bu gereksinimleri karşılayan bir protokoldür.

Kuyruk yöneticisi günlükleri

Yedek yönetim ortamının kuyruk yöneticisi verilerini tutarlı bir duruma geri yükleyebilmesi için, kuyruk yöneticisi yönetim ortamlarının kullandığı IBM i günlüklerini de yapılandırmanız gerekir. Kesintisiz hizmet için bu, etkin eşgörünüm başarısız olduğunda günlüklerin durumlarına geri yüklenmesi gerektiği anlamına gelir. Yedekleme ya da olağanüstü durumdan kurtarma çözümlerinden farklı olarak, günlüklerin daha önceki bir denetim noktasına geri yüklenmesi yeterli değildir.

You cannot physically share journals between multiple IBM i systems on networked storage. Kuyruk yöneticisi günlüklerini başarısızlık noktasındaki tutarlı duruma geri yüklemek için, etkin kuyruk yöneticisi yönetim ortamına yerel olan fiziksel günlüğü, etkinleştirilen yeni yönetim ortamına ya da beklemedeki yedek yönetim ortamlarının çalıştırıldığı günlük ikizlemelerine aktarmanız gerekir. İkizlenmiş günlük, başarısız olan eşgörünümüne ait yerel günlükle tam olarak zamanuyumlu olarak saklanan bir uzak günlük eşlemesidir.

Üç yapılandırma, çok eşgörünümlü bir kuyruk yöneticisine ilişkin günlüklerin nasıl yönetileceğini tasarlamaya ilişkin noktaları başlatıyor.

1. Etkin eşgörünüm ASP 'deki synchronized journal replication (günlük ikizlemenin), beklemedeki eşgörünümler ASP' lerine kullanılması.
2. Yapılandırılan bir IASP ' nin, etkin yönetim ortamından kuyruk yöneticisi günlüğünü etkin yönetim ortamı olarak devralacak beklemedeki yönetim ortamına taşıması için aktarma işlemi.
3. Eşitlenmiş ikincil IASP aynaları kullanılıyor.

Kuyruk yöneticisi verilerini IBM MQ IBM i CRTMQM komutundaki bir iASP' ye yerleştirmeye ilişkin ek bilgi için [ASP seçeneklerine](#) bakın.

Ayrıca, IBM Documentation içindeki IBM i bilgilerinde [Yüksek kullanılabilirlik](#) başlıklı konuya bakın.

Uygulamalar

To build a client to automatically reconnect to the queue manager when the standby queue manager resumes, connect your application to the queue manager using MQCONN and specify MQCNO_RECONNECT_Q_MGR in the **MQCNO** Seçenekler field. Yeniden bağlanabilir istemcileri kullanan üç örnek program için [Yüksek kullanılabilirlik örnek programları](#) ve kurtarma için istemci uygulamalarının tasarlanmasına ilişkin bilgi için [Uygulama kurtarma](#) başlıklı konuya bakın.

IBM üzerinde NetServer kullanarak kuyruk yöneticisi verileri için ağ paylaşımı yaratılması Kuyruk yöneticisi verilerini depolamak için bir IBM i sunucusunda ağ paylaşımı yaratın. Ağ paylaşımına erişmek için kuyruk yöneticisi eşgörünümüne gidecek olan iki sunucudan bağlantı kurmasını sağlar.

Başlamadan önce

- Bu görev için üç adet IBM i sunucusu gereklidir. Ağ paylaşımı sunuculardan birinde tanımlıdır, GAMMA. Diğer iki sunucu, ALPHA ve BETA, GAMMA 'ya bağlanmakta.
- IBM MQ 'u üç sunucuya kurun.
- System i Navigator'ı kurun; bkz. [System i Navigator](#).

Bu görev hakkında

- GAMA 'da kuyruk yöneticisi dizinini oluşturun ve kullanıcı profilleri QMQM ve QMQMADM için doğru sahiplik ve izinleri ayarlayın. Dizin ve izin, IBM MQ 'un GAMMA' da kurularak kolayca yaratılmasına neden olur.
- GAMA 'daki kuyruk yöneticisi verileri dizinine bir paylaşım oluşturmak için System i Navigator 'ı kullanın.
- Paylaşımına işaret eden ALPHA ve BETA 'da dizinler oluşturun.

Yordam

1. GAMA 'da, kuyruk yöneticisi verilerini, sahip olarak QMQM kullanıcı tanıtımıyla ve birincil grup olarak QMQMADM olan kuyruk yöneticisi verilerini barınamak için oluşturun.

İpucu:

Dizini doğru izinlerle oluşturmanın hızlı ve güvenilir bir yolu, IBM MQ 'yi GAMMA' da kurmandır.

Later, if you do not want to run IBM MQ on GAMMA, uninstall IBM MQ. After uninstallation, the directory /QIBM/UserData/mqm/qmgrs remains on GAMMA with the owner QMQM user profile, and QMQMADM the primary group.

Görev, paylaşımına ilişkin GAMMA üzerindeki /QIBM/UserData/mqm/qmgrs dizinini kullanır.

2. System i Navigator **Bağlantı ekle** sihirbazını başlatın ve GAMMA sistemine bağlanın.
 - a) Windows masaüstünüzde **System i Navigator** simgesini çift tıklayın.
 - b) Bağlantı yaratmak için **Evet** düğmesini tıklayın.
 - c) Follow the instructions in the **Bağlantı Ekle** wizard and create a connection from the IBM i system to GAMMA.

GAMA bağlantısı, **Sistemlerim**' e eklenir.
3. GAMA 'da yeni bir dosya paylaşımı ekleyin.
 - a) **System i Navigator** penceresinde, My Connections/GAMMA/File Systems klasöründeki File Shares klasörünü tıklayın.
 - b) **Görevlerim** penceresinde **IBM i NetServer paylaşımlarını yönet** seçeneğini tıklayın.

Yeni bir pencere (**IBM i NetServer -GAMMA**), masaüstünüzde açılır ve paylaşılan nesnelere gösterir.
 - c) Shared Objects klasörünü sağ tıklayın > **Dosya** > **Yeni** > **Dosya seçeneklerini belirleyin**.

Yeni bir pencere (**IBM i NetServer Dosya Paylaşımı-GAMMA**) açılır.
 - d) Paylaşımına bir ad verin (örneğin, WMQ).
 - e) Erişim denetimini Read/Write olarak ayarlayın.
 - f) Daha önce oluşturduğunuz /QIBM/UserData/mqm/qmgrs dizinine göz atarak **Yol adı** 'yı seçin ve **Tamam** 'ı tıklayın.

IBM i NetServer Dosya Paylaşımı-GAMMA penceresi kapatılır ve WMQ paylaşılan nesnelere penceresinde listelenir.

4. Paylaşılan nesnelere penceresinde **WMQ** öğesini farenin sağ düğmesiyle tıklattığınızda **Dosya > İzinler** seçeneğini tıklattığınızda.

/QIBM/UserData/mqm/qmgrsnesnesi için bir pencere (**Qmgrs Permissions-GAMMA**) açılır.

- a) Check the following permissions for QMQM, if they are not already set:

Read
Write
Execute
Management
Existence
Alter
Reference

- b) Check the following permissions for QMQMADM, if they are not already set:

Read
Write
Execute
Reference

- c) /QIBM/UserData/mqm/qmgrs' e izin vermek istediğiniz diğer kullanıcı profillerini ekleyin.

For example, you might give the default user profile (Public) Read and Execute permissions to /QIBM/UserData/mqm/qmgrs.

5. GAMMA için /QIBM/UserData/mqm/qmgrs erişimi verilen tüm kullanıcı profillerinin, GAMMA erişimi olan sunucularda aynı parolaya sahip olduğunu doğrulayın.

Özellikle, paylaşımına erişecek olan diğer sunuculardaki QMQM kullanıcı profillerinin, GAMMA 'daki QMQM kullanıcı profiliyle aynı parolaya sahip olduğundan emin olun.

İpucu: Parolaları belirlemek için System i Navigator 'daki My Connections/GAMMA/Users and Groups klasörünü tıklattığınızda. Alternatif olarak, **CHFUSRPRF** ve **CHGPWD** komutlarını kullanın.

Sonuçlar

Share olanağını kullanarak diğer sunuculardaki GAMMA 'ya erişmenizi sağlar. Diğer görevleri gerçekleştiriyorsanız, /QNTC/GAMMA/WMQyolunu kullanarak ALPHA ve BETA 'dan GAMMA'ya erişebileceğiniz denetleyin. /QNTC/GAMMA dizini ALPHA ya da BETA 'da yoksa, dizini yaratmanız gerekir. NetServer etki alanına bağlı olarak, dizini yaratmadan önce IPL ALPHA ya da BETA 'ya sahip olabilirsiniz.

```
CRTDIR DIR('/QNTC/GAMMA')
```

When you have checked that you have access to /QNTC/GAMMA/WMQ from ALPHA or BETA, issuing the command, CRTMQM MQMNAME('QM1') MQMDIRP('/QNTC/GAMMA/WMQ') creates /QIBM/UserData/mqm/qmgrs/QM1 on GAMMA.

Sonraki adım

Create a multi-instance queue manager by following the steps in either of the tasks, [“IBM üzerinde günlük ikizleme ve NetServer kullanan çok eşgörümlü bir kuyruk yöneticisi yaratılması” sayfa 382](#) or [“IBM üzerinde NetServer ve günlük ikizlemesi kullanılarak, tek bir eşgörümlü kuyruk yöneticisinin çok eşgörümlü kuyruk yöneticisine dönüştürülmesi” sayfa 386](#).

IBM i **IBM üzerinde yedek sisteme geçiş performansı**

Bir kuyruk yöneticisi yönetim ortamını algılamak için gereken süre başarısız oldu ve daha sonra, bir bekleme sırasında işlenmeye devam etmek, yapılandırmaya bağlı olarak on saniyeden on beş dakika ya da daha fazla süre arasında değişebilir. Yüksek kullanılabilirlik çözümlerinin tasarlanması ve test edilmesi için performansın önemli bir değerlendirmeye sahip olması gerekir.

Çok eşgörünümli bir kuyruk yöneticisinin günlük eşlemesini kullanacak şekilde yapılandırılıp yapılandırılmayacağını ya da bir IASP 'nin mi kullanılacağını belirlemeye karar vermede avantaj ve dezavantajlar vardır. İkizleme, kuyruk yöneticisinin bir uzak günlüğe zamanuyumlu olarak yazmasını gerektirir. Bir donanım bakış açısından, bu gereksinme başarımı etkilemez; ancak, bir yazılım perspektifinden, uzak günlüğe yazma işlemi, yalnızca yerel bir günlüğe göre değil, daha büyük bir yol uzunluğu vardır ve bu durum, çalışmakta olan bir kuyruk yöneticisinin başarımını bir ölçüde azaltması beklenebilir. Ancak, beklemedeki kuyruk yöneticisi devraldığında, başarısız olmadan önce yerel günlüğü uzak günlüğün sağladığı uzak günlükten eşzamanlamaya ilişkin gecikme, genellikle IBM i 'un kuyruk yöneticisinin yedek yönetim ortamını çalıştıran sunucuya IASP' yi algılayıp aktarması için geçen süreye göre küçüktür. IASP aktarma süreleri, saniyeler içinde tamamlanabilmesi için on-on beş dakika kadar uzun olabilir. IASP aktarım süresi, IASP yedek sisteme aktarıldığında ve birleştirilmesi gereken erişim yollarının ya da dizinlerin büyüklüğünün *kullanılır kılındı* olması gereken nesne sayısına bağlıdır.

When the standby queue manager takes over, the delay in synchronizing its local journal from the remote journal maintained by the active instance before it failed, is typically small in comparison to the time it takes for IBM i to detect and transfer the independent ASP to the server running the standby instance of the queue manager. Bağımsız ASP aktarım zamanları, saniyeler içinde tamamlanabilmesi için on-on beş dakika kadar uzun olabilir. Bağımsız ASP aktarım zamanı, bağımsız ASP yedek sisteme aktarıldığında ve birleştirilmesi gereken erişim yollarının ya da dizinlerin büyüklüğünün *kullanılır kılındı* olması gereken nesne sayısına bağlıdır.

Ancak, günlük aktarımı, yedek yönetim ortamının tam olarak sürdürülmesi için gereken süreyi etkileyen tek faktör değildir. Ayrıca, ağ dosya sisteminin, başlatma işlemine devam etmek üzere yedek yönetim ortamına sinyal veren kuyruk yöneticisi verilerinin üzerindeki kilidi serbest bırakması için gereken süreyi de göz önünde bulundurmanız gerekir. Ayrıca, yönetim ortamının iletileri yeniden işlemeye başlayabilmesi için günlükten kuyrukları kurtarmak için gereken süreyi de göz önünde bulundurmanız gerekir. Bu diğer gecikme kaynakları, beklemedeki bir yönetim ortamı başlatmak için gereken süreyi içerir. Geçiş yapmak için gereken toplam süre aşağıdaki bileşenlerden oluşur:

Arıza algılama süresi

The time it takes for NFS to release the lock on the queue manager data, and the standby instance to continue its startup process.

Aktarma süresi

Bir HA kümesi durumunda, etkin eşgörünümü yedek yönetim ortamına barındıran sistemden IASP 'yi aktarmak için IBM i ' u (IASP ' yi) alır ve günlük kopyalaması durumunda, yerel günlüğü uzak eşlemedeki verilerle yedeklemek için gereken süreyi alır.

Yeniden başlatma zamanı

Yeni etkin kuyruk yöneticisi yönetim ortamının, geri yüklenen günlüğündeki en son denetim noktasından kuyruklarını yeniden oluşturması ve iletilerin işlenmesine devam etmesi için gereken süre.

Not:

Devralan yedek yönetim ortamı daha önce etkin olan eşgörünümle zamanuyumlu olarak eşlenecek şekilde yapılandırıldıysa, başlatma geciktirilebilir. Uzak günlük, daha önce etkin olan yönetim ortamına ev sahipliği yapan sunucudaysa ve sunucu başarısız olursa, yeni etkinleştirilen eşgörünüm uzak günlüğe kopyalanamıyor olabilir.

Zamanuyumlu yanıt beklenecek varsayılan saat bir dakikadır. Eşleme zamanından önce gecikme üst sınırını yapılandırabilirsiniz. Diğer bir seçenek olarak, yedek yönetim ortamlarını, başarısız olan etkin yönetim ortamına zamanuyumsuz kopyalamayı kullanmaya başlamak için yapılandırabilirsiniz. Daha sonra, başarısız olan eşgörünüm yeniden beklemede olduğunda zamanuyumlu kopyalamaya geçiş yapabilirsiniz. Aynı dikkat edilmesi, zamanuyumlu bağımsız ASP aynalarının kullanılması için de geçerlidir.

Bu bileşenler için, hata durumunda yedek sisteme geçiş süresini değerlendirmenize ve hangi yapılandırma yaklaşımının kullanılacağı kararınıza katlanmanıza yardımcı olacak ayrı temel çizgi ölçümleri yapabilirsiniz. En iyi yapılandırma kararını vermek için, aynı sunucudaki diğer uygulamaların hata durumunda yedek sisteme geçiş işlemi nasıl gerçekleştireceğini ve IASP ' yi kullanan yedekleme ya da olağanüstü durumdan kurtarma işlemleri olup olmadığını da göz önünde bulundurmanız gerekir.

IASP aktarım süreleri, küme yapılandırmanızı ayarlayarak kısaltılabilir:

1. Kümedeki sistemler arasında kullanıcı tanımlarının, kullanıcı tanımlarının UID ve GID 'leri değiştirmesi gereksinmesini ortadan kaldırmak için aynı GID ve UID' ye sahip olması gerekir.
2. Disk havuzu grubuna ilişkin çapraz başvuru çizelgesini yaratmak üzere birleştirilmesi gerektiğinde, sistem ve temel kullanıcı disk havuzlarındaki veritabanı nesnelere sayısını en aza indirin.
3. Daha fazla performans ipucu, IBM Redbook, *Implementing PowerHA for IBM i, SG24-7405* içinde bulunabilir.

Temel ASP ' leri, günlük ikizlemeyi ve küçük bir yapılandırmayı kullanan bir yapılandırma, onlarca saniye içinde geçiş işlemini sona ermelidir.

IBM i IBM i kümeleme yeteneklerini IBM MQ kümeleme ile birleştirmeye genel bakış

Running IBM MQ on IBM i, and exploiting the IBM i clustering capabilities can provide a more comprehensive High Availability solution, than using only IBM MQ clustering.

Bu yetenesahip olmak için, aşağıdaki bilgileri ayarlamanız gerekir:

1. IBM i makinenizdeki kümeler; bkz. [“IBM i Kümeler” sayfa 373](#)
2. Kuyruk yöneticisini taşıdığınız bağımsız yardımcı bellek havuzu (IASP); bkz. [“Bağımsız yardımcı bellek havuzları \(IASP ' ler\)” sayfa 373](#)
3. Bir küme kaynağı grubu (CRG); aşağıdakileri tanımladığınız [“Aygıt kümesi kaynak grupları” sayfa 374](#) a bakın:
 - Kurtarma etki alanı
 - IASP
 - Programdan çıkın; bkz. [“Aygıt CRG çıkış programı” sayfa 374](#)

IBM i Kümeler

Bir IBM i kümesi, mantıksal olarak birbirine bağlanmış olan, IBM i bilgisayarları ya da bölümleri olan eşgörünümlerin bir derlemidir.

Bu gruplamanın amacı, tek bir hata noktasını ortadan kaldırmak ve uygulama ve veri esnekliğini artırmak için her bir eşgörünümün yedeklenmesini sağlamak. Bir küme yaratılsa, çeşitli küme kaynağı grubu (CRG) tipleri, kümedeki uygulamaları, verileri ve aygıtları yönetmek için yapılandırılabilir.

Ek bilgi için [Küme oluşturma ve Küme Oluştur \(CRTCLU\)](#) komutu konusuna bakın.

Bağımsız yardımcı bellek havuzları (IASP ' ler)

IASP, tek düzeyli depolamanın bir uzantısı olarak hizmet veren bir kullanıcı ASP tipidir. Bu depolama parçası, sistem depolama alanından bağımsızlığı nedeniyle sisteme IPL ' ye gerek kalmadan kolayca işlenebilir.

Bir IASP, başka bir işletim sistemi eşgörünümlerine kolayca geçiş yapabilir ya da başka bir işletim sistemi eşgörünümündeki bir hedef IASP ' ye eşlenebilir. Eşgörünümler arasında bir IASP ' yi değiştirmek için iki yöntem kullanılabilir:

- İlk yöntem, kümedeki tüm bilgisayarların ve IASP ' yi içeren değişimli kullanılabilir disk kulesine, High Speed Link (HSL) döngüsü kullanılarak bağlanabilmelidir.
- İkinci yöntem, işletim sistemi yönetim ortamlarının, giriş/çıkış işlemcilerinin (IOP) bölümler arasında değişimli olarak değiştirebileceği aynı IBM i bilgisayarında bölümlemenizi gerektirir. Bir IASP ' yi eşleyebilmek için özel bir donanım gerekmez. Eşleme, ağ üzerinden TCP/IP kullanılarak gerçekleştirilir.

Ek bilgi için [Configure Device ASP \(CFGDEVASP\)](#) (Aygıt ASP ' yi Yapılandır) komutunu bakın.

Aygıt kümesi kaynak grupları

Birkaç tip küme kaynağı grubu (CRGs) vardır. Farklı CRG tipleri hakkında daha fazla bilgi için bkz. [Küme kaynağı grubu](#).

Bu konu, bir aygıt CRG 'ye yoğunlaştırılır. Aygıt CRG 'si:

- Bağımsız yardımcı bellek 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, birincil düğüm olarak hangi küme düğümünün dikkate alıncağını belirtir. Düğümlerin geri kalanı yedekler olarak kabul edilir. Yedekleme düğümleri, kurtarma etki alanında bulunan düğümlere bağlı olarak, ilk yedekleme, ikinci yedekleme ve benzeri düğümlerin hangi düğümle olduğunu belirterek, kurtarma etki alanında da sıralanır.

Birincil düğüm hatasında, çıkış programı kurtarma etki alanındaki tüm düğümlerde çalıştırılır. İlk yedeklemede çalışan çıkış programı, bu düğümü yeni birincil düğüm yapmak için gerekli kullanıma hazırlamaları yapabilir.

Ek bilgi için [Aygıt CRG 'leri Yaratılması ve Küme Kaynağı Grubu Yarat \(CRTCRG\)](#) komutu konusuna bakın.

Aygıt CRG çıkış programı

İşletim sistemi kümesi kaynak hizmeti, kurtarma etki alanının tanımladığı düğümlerden birinde bir olay ortaya çıktığında, bir aygıt CRG çıkış programını çağırır; örneğin, hata durumunda yedek sisteme geçiş ya da yedek sisteme geçiş olayı.

Hata durumunda yedek sisteme geçiş olayı, kümenin birincil düğümü başarısız olduğunda ve CRG 'ler yönettikleri tüm kaynaklarla değiştirilirse ve belirli bir CRG manüel olarak birincil düğümden yedek düğüme geçtiğinde bir geçiş olayı ortaya çıkar.

Her iki yöntemle de, çıkış programı, ilk yedek düğümü yeni birincil düğüme dönüştüren önceki birincil düğümden çalışan tüm programları başlatıp başlatmaktan sorumlu olur.

Örneğin, IBM MQ ile, çıkış programı IBM MQ altsistemini (QMOM) başlatmaktan ve kuyruk yöneticilerinden sorumlu olmalıdır. Kuyruk yöneticileri, tetikleme izleme programları gibi dinleyici ve hizmetleri otomatik olarak başlatmak için yapılandırılmalıdır.

A sample exit program, AMQSCR4, is available on IBM i from IBM MQ 9.1.

Değişimli kullanılabilir IASP konfigürasyonu

IBM MQ can be set up to take advantage of the clustering capabilities of IBM i. Bunu yapmak için:

1. Veri merkezi sistemleri arasında bir IBM i kümesi yaratın
2. Kuyruk yöneticisini IASP 'ye taşıyın.

[“Bağımsız bir yardımcı bellek havuzuna taşıma ya da kaldırma, kuyruk yöneticisi ya da bağımsız bir yardımcı bellek havuzu” sayfa 375](#) , 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 375](#) , bu işlemi gerçekleştirmenize yardımcı olacak bazı örnek kodları içerir.

İlgili kavramlar

[“Bağımsız ASP 'ler ve yüksek kullanılabilirlik” sayfa 395](#)

Bağımsız ASP 'ler, uygulamaların ve verilerin sunucular arasında taşınmasını sağlar. Bağımsız ASP 'lerin esnekliği, bazı IBM i yüksek kullanılabilirlik çözümlerinin temelini teşkil ettikleri anlamına gelir.

Kuyruk yöneticisi günlüğü için bir ASP 'yi ya da bağımsız ASP 'yi kullanıp kullanmayacağınızı göz önünde

bulundurarak, bağımsız ASP ' lere dayalı diğer yüksek kullanılabilirlik yapılandırmasını göz önünde bulundurmanız gerekir.

IBM i *Aygıt kümesi kaynak grubunun yapılandırılması*
Bir aygıt kümesi kaynak grubu (CRG) oluşturmak için bir örnek program.

Bu görev hakkında

Aşağıdaki örnekte aşağıdakine dikkat edin:

- [PRIMARY SITE NAME] ve [BACKUP SITE NAME], sekiz karakterlik ya da daha az karakterden oluşan iki ayrı dizgi olabilir.
- [PRIMARY IP] ve [BACKUP IP] ikizleme için kullanılacak IP adresleridir.

Yordam

1. Kümenin adını belirleyin.
2. CRG çıkış programı adını ve kitaplığını tanımlayın.
3. Bu CRG tarafından tanımlanacak birincil düğüm ve yedek düğümlerin adını belirleyin.
4. Bu CRG tarafından yönetilecek IASP ' yi tanımlayın ve birincil düğüm altında yaratıldığından emin olun.
5. Aşağıdaki komutu kullanarak yedek düğümlerde bir aygıt tanımlaması yaratın:

```
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 başlatmak için şu komutu kullanın:

```
STRTCPIFC INTNETADR(' [TAKEOVER IP]')
```

8. İsteğe bağlı: IASP ' niz 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 ' niz ikizlenmiş 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 *Bağımsız bir yardımcı bellek havuzuna taşıma ya da kaldırma, kuyruk yöneticisi ya da bağımsız bir yardımcı bellek havuzu*

Kuyruk yöneticisini bir IASP ' den kaldırmak için kuyruk yöneticisini bağımsız bir yardımcı bellek havuzuna (IASP) ve komutlarına taşımak için kullanılan örnek bir program.

Bu görev hakkında

Aşağıdaki örnekte aşağıdakine dikkat edin:

- [YÖNETİCİAD] kuyruk yöneticinizin adıdır.
- [IASP NAME], IASP ' nizin adıdır.
- [MANAGER LIBRARY], kuyruk yöneticisi kitaplığınızın adıdır.

- [MANAGER DIRECTORY], kuyruk yöneticisi dizininin adıdır.

Yordam

1. Birincil düğümünüzü ve yedek düğümlerinizi tanımlayın.
2. Birincil düğümünüzde aşağıdaki yordamı gerçekleştirin:
 - a) Kuyruk yöneticinizin sona erdirildiğinden emin olun.
 - b) Komutu kullanarak IASP 'nizin vary on olduğundan emin olun.

```
VRYCFG CFGOBJ([IASP NAME]) CFGTYPE(*DEV) STATUS(*ON)
```

- c) IASP altında kuyruk yöneticisi dizinini yaratın.
Kök altında IASP 'nizin adı altında bir dizin olacaktır:

```
QSH CMD('mkdir -p /[IASP_NAME]/QIBM/UserData/mqm/qmgrs/')
```

- d) Yöneticinizin IFS nesnelere, yalnızca aşağıdaki komutu kullanarak IASP altında oluşturduğunuz 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 bir geçici saklama dosyası yaratın:

```
CRTSAVF QGPL/MGRLIB
```

- f) Aşağıdaki komutu kullanarak kuyruk yöneticisi kitaplığını MGRLIB saklama kütüğüne 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) Kuyruk yöneticisi kitaplığını IASP 'ye geri yüklemek için aşağıdaki komutu kullanın:

```
RSTLIB SAVLIB([MANAGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)  
RSTASPDEV([IASP NAME])
```

- i) Geçici saklama dosyasını silmek için aşağıdaki komutu kullanın:

```
DLTF FILE(QGPL/MGRLIB)
```

- j) Aşağıdaki komutu kullanarak, IASP altındaki kuyruk yöneticisi IFS nesnelere simgesel 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 bağlantı ekleyin:

```
SETASPGRP [IASP NAME]
```

- l) Şu komutu kullanarak kuyruk yöneticinizi başlatın:

```
STRMQM [MANAGER NAME]
```

3. Yedekleme düğümünüzde ya da düğümlerinizde 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 simgesel 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) Add the following at the end of the file /QIBM/UserData/mqm/mqs.ini:

```
QueueManager:
Name=[MANAGER_NAME]
Prefix=/QIBM/UserData/mqm
Library=[MANAGER_LIBRARY]
Directory=[MANAGER_DIRECTORY]
```

4. Bir kuyruk yöneticisini IASP ' den kaldırmak için aşağıdaki komutları girin:

- VRYCFG CFGOBJ ([IASP ADI]) CFGTYPE (*DEV) DURUMU (*ON)
- SETASPGRP [IASP ADI]
- ENDMQM [YÖNETİCİAD
- DLTMQM [YÖNETİCİAD

IBM i IBM üzerinde ASP için ikizlenmiş günlük yapılandırması

İkizlenmiş günlükler arasında zamanuyumlu eşlemeyi kullanarak güçlü bir çok eşgörünümlü kuyruk yöneticisi yapılandırın.

İkizlenmiş bir kuyruk yöneticisi yapılandırması, temel ya da bağımsız yardımcı bellek havuzlarında (ASP) oluşturulan günlükler kullanır.

IBM işletim sisteminde, kuyruk yöneticisi verileri günlüklere ve bir dosya sistemine yazılır. Günlükler, kuyruk yöneticisi verilerinin ana kopyasını içerir. Günlükler, zamanuyumlu ya da zamanuyumsuz günlük eşlemesi kullanan sistemler arasında paylaşılır. Bir kuyruk yöneticisi yönetim ortamını yeniden başlatmak için yerel ve uzak günlüklerin bir karışımı gereklidir. Kuyruk yöneticisi yeniden başlatma işlemi, sunucudaki yerel ve uzak günlüklerin ve paylaşılan ağ dosya sistemindeki kuyruk yöneticisi verilerine ilişkin günlük kayıtlarını okur. Dosya sistemindeki veriler, kuyruk yöneticisini yeniden başlatma hızlarını hızlandırır. Denetim noktaları dosya sisteminde depolanır, dosya sistemi ile günlükler arasındaki eşzamanlama noktaları işaretlenir. Denetim noktasından önce saklanan günlük kayıtları, tipik kuyruk yöneticisi yeniden başlatma işlemleri için gerekli değildir. Ancak, dosya sistemindeki veriler güncel olmayabilir ve kuyruk yöneticisi yeniden başlatma işlemi tamamlamak için denetim noktası kullanıldıktan sonra günlük kayıtları kullanılır. Yönetim ortamına bağlı günlüklerdeki veriler, yeniden başlatma işleminin başarıyla tamamlanabilmesi için güncel tutulurlar.

Ancak, yedekleme sunucusundaki uzak günlük zamanuyumsuz bir şekilde eşlenmiş oluyorsa ve hata, uyumlulaştırılmadan önce hata oluştuysa, günlük kayıtları bile güncel olmayabilir. Eşitlenmeyen bir uzak günlüğü kullanarak bir kuyruk yöneticisini yeniden başlatmaya karar vermeniz durumunda, beklemedeki kuyruk yöneticisi yönetim ortamı, etkin eşgörünüm başarısız olmadan önce silinen iletileri yeniden işleyebilir ya da etkin eşgörünüm başarısız olmadan önce alınan iletileri işlenemez.

Başka bir olasılık da, dosya sisteminin en son denetim noktası kaydını içermesi ve yedek veritabanında eşzamanlanmamış bir uzak günlük içermemesi olabilir. Bu durumda kuyruk yöneticisi otomatik olarak yeniden başlatılmaz. Uzak günlük uyumlulaştırılıncaya ya da beklemedeki kuyruk yöneticisini dosya sisteminden başlatılıncaya kadar beklediğiniz bir seçiminiz vardır. Bu durumda, dosya sisteminde kuyruk yöneticisi verilerinin uzak günlükten daha yeni bir denetim noktası vardır; bu denetim noktası, etkin eşgörünüm başarısız olmadan önce işlenen tüm iletileri içermeyebilir. Günlüklerle eşzamanlamanın dışında bir soğuk yeniden başlatma işleminden sonra bazı iletiler yeniden işlenebilir ve bazıları işlenmez.

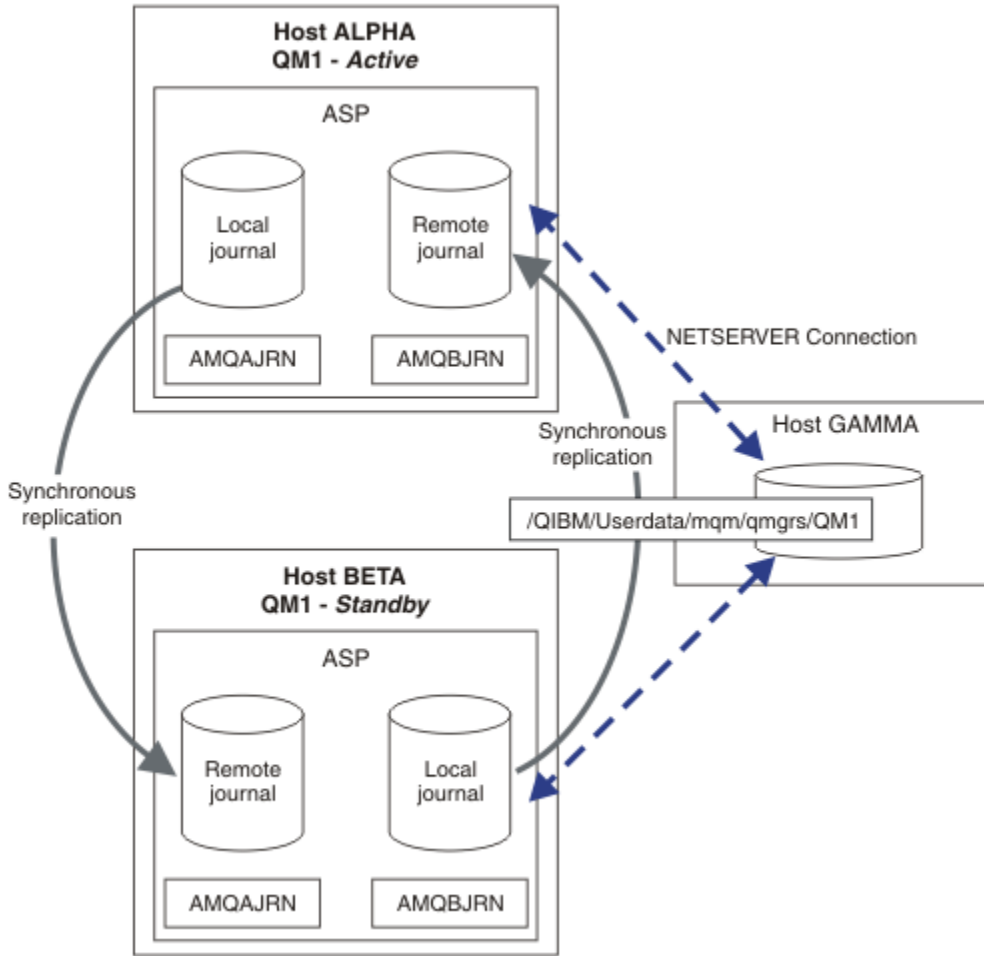
Çok eşgörünümlü bir kuyruk yöneticisiyle, bir kuyruk yöneticisinin hangi yönetim ortamının etkin olduğunu ve hangisinin yedek veritabanı olduğunu denetlemek için dosya sistemi de kullanılır. Etkin eşgörünüm, kuyruk yöneticisi verilerine bir kilit edinir. Bekleme kilidi, kilidi elde etmeyi bekler ve bunu yaptığında,

etkin yönetim ortamı olur. Kilit, olağan biçimde sona ererse, etkin eşgörünüm tarafından serbest bırakılır. Kütük sistemi etkin yönetim ortamını saptarsa ya da kütük sistemine erişemiyorsa, kilit kütük sistemi tarafından serbest bırakılır. Dosya sistemi, hata saptamaya 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 olarak yeniden başlatma olanağı sağlar. Kuyruk yöneticisi verilerinin saklandığı kütük sisteminin başarısız olması nedeniyle kuyruk yöneticisi verilerinin geri yüklenmesini de destekler.

In [Şekil 27 sayfa 378](#), if ALPHA fails, you can manually restart QM1 on BETA, using the mirrored journal. Çoklu yönetim ortamı kuyruk yöneticisi yeteneğini QM1 olarak ekleyerek, ALPHA 'daki etkin yönetim ortamı başarısız olursa, yedek QM1 yönetim ortamı BETA' ya otomatik olarak devam eder. QM1 can also resume automatically if it is the server ALPHA that fails, not just the active instance of QM1. BETA etkin kuyruk yöneticisi yönetim ortamının anasistemi olduğunda, beklemedeki yönetim ortamı ALPHA üzerinde başlatılabilir.

[Şekil 27 sayfa 378](#), kuyruk yöneticisi verilerini saklamak için NetServer kullanarak bir kuyruk yöneticisinin iki eşgörünümü arasındaki günlüklerin kopyalandığına ilişkin bir konfigürasyon gösterir. Örüntüleri daha fazla günlük içerecek şekilde genişletilebilir ve dolayısıyla daha fazla eşgörünüm ekleyebilirsiniz. Follow the journal naming rules explained in the topic, "[Queue manager journals on IBM i](#)" sayfa 358. Şu anda bir kuyruk yöneticisinin çalışmakta olan eşgörünümlerinin sayısı iki ile sınırlıdır, biri etkin, diğeri ise yedekte bekleme durumunda.



Şekil 27. Kuyruk yöneticisi günlüğünü ikizle

ALPHA anasistem üzerindeki QM1 yerel günlüğü, AMQAJRN (ya da daha tam olarak, QMQM1/AMQAJRN) ve BETA ' da QMQM1/AMQBJRNadlı günlük günlük olur. Her yerel günlük, kuyruk yöneticisinin diğer tüm

yönetim ortamlarındaki uzak günlüklere eşlenir. Kuyruk yöneticisi iki yönetim ortamı ile yapılandırıldıysa, yerel bir 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 27 sayfa 378 içindeki eşleme kipi *SYNC, *ASYNCdeğil. *ASYNC daha hızlıdır, ancak uzak günlük durumu *ASYNCPENDolduğunda bir hata oluşursa, yerel ve uzak günlük tutarlı değildir. Uzak günlüğün yerel günlükle birlikte olması gerekir. *SYNCseçeneğini belirlerseniz, tamamlanan yazma gerektiren bir çağrıdan dönmeye önce, yerel sistem uzak günlüğü bekler. Yerel ve uzak günlükler genellikle birbiri ile tutarlı kalır. Yalnızca *SYNC işleminin belirlenmiş bir süreden uzun sürerse¹ve uzak günlük kaydı devre dışı bırakıldığında, günlüklerin eşitlemeden kurtulması gerekir. Bir hata günlük ileti kuyruğuna ve QSYSOPR ' ye kaydedilir. Kuyruk yöneticisi bu iletiyi algılar, kuyruk yöneticisi hata günlüğüne bir hata yazar ve kuyruk yöneticisi günlüğün uzak eşlemesini devre dışı bırakır. Etkin kuyruk yöneticisi yönetim ortamı, bu günlüğe uzak günlük kaydı olmadan sürdürülür. Uzak sonucu yeniden kullanılabilir durumda olduğunda, zamanuyumlu uzak günlük kopyalamayı el ile yeniden etkinleştirmeniz gerekir. Günlükler daha sonra yeniden eşitlenir.

Şekil 27 sayfa 378 içindeki *SYNC / *SYNC yapılarındaki bir sorun, BETA ' daki yedek kuyruk yöneticisi yönetim ortamının denetime nasıl gireceğini içerir. BETA kuyruk yöneticisi yönetim ortamı ilk kalıcı iletisini yazarken, uzak günlüğü ALPHA ' da güncelleme girişiminde bulunur. ALPHA 'dan BETA' ya geçiş kontrol nedeni ALPHA 'nın başarısızlığı ve ALFA' nın hala kapalı olması durumunda, uzaktan günlük kaydı ALPHA ' ya hata ortaya çıktı. BETA, ALPHA ' nın yanıt vermesini bekler ve daha sonra, uzak günlüğe kaydetme işlemini devre dışı bırakır ve yalnızca yerel günlük kaydı ile iletileri işlemeye devam eder. BETA 'nın bir süre beklemeli, ALFA' nın kapalı olduğunu, bir hareketsizlik sürmesine neden olduğunu.

Uzak günlüğe kaydetme işlemini *SYNC ya da *ASYNC olarak ayarlamak arasında seçim yapılması, bir değiş tokuş olup olmadığını belirlemek için kullanılan bir terimdir. Çizelge 24 sayfa 379 , bir kuyruk yöneticisi çifti arasında *SYNC ve *ASYNC günlük kaydı kullanılarak yapılan değiş tokuşları özetler.

Çizelge 24. Uzak günlük kaydı seçenekleri			
Etkin	Yedek	*SYNC	*ASYNC
*SYNC		<ol style="list-style-type: none">1. Yedek yedek sisteme geçiş ve yedek sisteme geçiş2. Yedek yönetim ortamı, hata durumunda yedek sisteme geçiş işleminden hemen sonra devam etmez.3. Uzak günlük kaydı her zaman kullanılabilir olmalıdır4. Kuyruk yöneticisi başarıımı uzak günlüğe kaydetmeye bağlıdır	<ol style="list-style-type: none">1. Yedek yedek sisteme geçiş ve yedek sisteme geçiş2. Yedek sonucu kullanılabilir durumda olduğunda, uzak günlüğe kaydetme işlemi *SYNC olarak değiştirilmelidir3. Uzak günlüğe kaydetme işlemi yeniden başlatıldıktan sonra kullanılabilir durumda kalmalıdır4. Kuyruk yöneticisi başarıımı uzak günlüğe kaydetmeye bağlıdır

¹ Belirlenen süre, IBM i 5 ve IBM i 6.1 ' deki 1-3600 saniye aralığında belirlenen süre 60 saniyedir.

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 yedek sisteme geçiş işleminden sonra bazı iletiler kaybolabilir ya da yinelenabilir 2. Etkin yönetim ortamının gecikmeden devam etmesi için, beklemedeki yönetim ortamının kullanılabilir durumda olması gerekmez. 3. Başarım, uzak günlüğe kaydetmeye bağlı değildir
*ASYNC			

*SYNC / *SYNC

The active queue manager instance uses *SYNC journaling, and when the standby queue manager instance starts, it immediately tries to use *SYNC journaling.

1. Uzak günlük, etkin kuyruk yöneticisinin yerel günlüğüyle tutarlı olarak tutarlı. Kuyruk yöneticisi yedek yönetim ortamına geçtiyse, hemen devam edebilir. Yedekleme ortamı, iletilerin herhangi bir kaybı ya da yinelenmesi olmadan olağan bir şekilde sürdürülür. Son denetim noktasından bu yana uzak günlük kaydı başarısız olursa iletiler kaybedilir ya da yinelenir ve daha önce etkin olan kuyruk yöneticisi yeniden başlatılamaz.
2. Kuyruk yöneticisi yedek yönetim ortamı için başarısız olursa, hemen başlatılamayabilir. Beklemedeki kuyruk yöneticisi yönetim ortamı *SYNC günlük kaydı ile etkinleştirilir. Yedeği devreye sokma nedeni, uzak günlük kaydının yedek yönetim ortamını bulandıran sunucuya kaydedilmesini engelleyebilir. Kuyruk yöneticisi, kalıcı iletiler işlenmeden önce sorun algılanıncaya kadar bekler. Bir hata günlük ileti kuyruğuna ve QSYSOPR ' ye kaydedilir. Kuyruk yöneticisi bu iletiyi algılar, kuyruk yöneticisi hata günlüğüne bir hata yazar ve kuyruk yöneticisi günlüğün uzak eşlemesini devre dışı bırakır. Etkin kuyruk yöneticisi yönetim ortamı, bu günlüğe uzak günlük kaydı olmadan sürdürülür. Uzak sunucu yeniden kullanılabilir durumda olduğunda, zamanuyumlu uzak günlük kopyalamayı el ile yeniden etkinleştirmeniz gerekir. Günlükler daha sonra yeniden eşitlenir.
3. Uzak günlüğü saklamak için, uzak günlüğünün kopyalanacağı sunucu her zaman kullanılabilir olmalıdır. Uzak günlük tipik olarak, yedek kuyruk yöneticisini barındıran aynı sunucuya kopyalanır. Sunucu kullanılamaz duruma gelebilir. Bir hata günlük ileti kuyruğuna ve QSYSOPR ' ye kaydedilir. Kuyruk yöneticisi bu iletiyi algılar, kuyruk yöneticisi hata günlüğüne bir hata yazar ve kuyruk yöneticisi günlüğün uzak eşlemesini devre dışı bırakır. Etkin kuyruk yöneticisi yönetim ortamı, bu günlüğe uzak günlük kaydı olmadan sürdürülür. Uzak sunucu yeniden kullanılabilir durumda olduğunda, zamanuyumlu uzak günlük kopyalamayı el ile yeniden etkinleştirmeniz gerekir. Günlükler daha sonra yeniden eşitlenir.
4. Uzak günlük kaydı, yerel günlüğe kaydetme değerinden daha düşüktür ve sunucular büyük bir uzaklıktan ayrılmışsa, büyük ölçüde daha yavaş olur. Kuyruk yöneticisi, kuyruk yöneticisi başarımını düşüren uzak günlük kaydı için beklemelidir.

Bir sunucu çifti arasındaki *SYNC / *SYNC yapılandırışındaki hata durumunda yedek yönetim ortamını sürdürme gecikmesi de olumsuz olur. *SYNC / *ASYNC yapılandırışında bu sorun yoktur.

*SYNC / *SYNC does guarantee no message loss after switchover or failover, as long as a remote journal is available. Hata durumunda yedek sisteme geçiş ya da yedek sisteme geçiş işleminden sonra 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 beklemedeki kuyruk yöneticisi yönetim ortamı başlatıldığında, günlüğe kaydetme işlemini *ASYNC günlüğe kaydetme işlevini kullanır. Yeni yedek yönetim ortamını barındıran sunucu kullanılabilir duruma geldikten kısa

bir süre sonra, sistem işletmeninin etkin yönetim ortamındaki uzak günlüğü *SYNColarak değiştirmesi gerekir. İşletmen, uzak günlük kaydını *ASYNCL'den *SYNC değerine değiştirdiğinde, uzak günlüğün durumu *ASYNCPENDise etkin yönetim ortamı duraklatmasını sağlar. Etkin kuyruk yöneticisi yönetim ortamı, kalan günlük girişlerinin uzak günlüğe aktarılincaya kadar bekler. Uzak günlük yerel günlükle uyumlulaştırıldığında, yeni yedek veritabanının yeni etkin yönetim ortamıyla yeniden tutarlı bir şekilde tutarlı olması gerekir. Çok eşgörünümli kuyruk yöneticilerinin yönetimi perspektifinden, bir *SYNC / *ASYNCL yapılandırmasında, IBM i sistem işletmeninin ek bir görevi vardır. Başarısız kuyruk yöneticisi yönetim ortamının yeniden başlatılmasına ek olarak, işletmenin uzak günlük kaydını *SYNC olarak değiştirmesi gerekir.

1. Uzak günlük, etkin kuyruk yöneticisinin yerel günlüğüyle tutarlı olarak tutarlı. Etkin kuyruk yöneticisi yönetim ortamı yerine geçtiyse ya da yedek yönetim ortamı üzerinde başarısız olursa, beklemedeki yönetim ortamı hemen sürdürür. Yedekleme ortamı, iletilerin herhangi bir kaybı ya da yinelenmesi olmadan olağan bir şekilde sürdürülür. Son denetim noktasından bu yana uzak günlük kaydı başarısız olursa iletiler kaybedilir ya da yinelenir ve daha önce etkin olan kuyruk yöneticisi yeniden başlatılamaz.
2. Etkin yönetim ortamını barındıran sistem yeniden kullanılabilir duruma geldikten kısa bir süre sonra, sistem işletmeninin uzak günlüğü *ASYNCL ile *SYNC değerine değiştirmesi gerekir. Uzak günlük *SYNColarak değiştirilmeden önce, işletmen uzak günlüğünün yakalaması için beklenebilir. Diğer bir seçenek olarak, işletmen uzak yönetim ortamını hemen *SYNC değerine çevirebilir ve etkin yönetim ortamını, beklemedeki yönetim ortamı günlüğü yakalanıncaya kadar beklemesini zorunlu kılabilir. Uzak günlük kaydı *SYNCLdeğerine ayarlandığında, yedek yönetim ortamı genellikle etkin yönetim ortamıyla tutarlı olarak tutarlı bir şekilde tutarlı olur. Son denetim noktasından bu yana uzak günlük kaydı başarısız olursa iletiler kaybedilir ya da yinelenir ve daha önce etkin olan kuyruk yöneticisi yeniden başlatılamaz.
3. Bir yedek anahtardan ya da yedek sisteme geçişten yapılandırma geri yüklendiğinde, uzak günlüğün bulunduğu sunucu her zaman kullanılabilir durumda olmalıdır.

Yedek kuyruk yöneticisinin hata durumunda yedek sisteme geçiş işleminden sonra hızlı bir şekilde devam etmesi için *SYNC / *ASYNCL seçeneğini belirleyin. Yeni etkin yönetim ortamında uzak günlük ayarını *SYNC olarak el ile geri yüklemeniz gerekir. *SYNC / *ASYNCL yapıları, bir çift çok eşgörünümli kuyruk yöneticisinin denetlenmesine ilişkin olağan örüntüyle eşleşir. Bir yönetim ortamı başarısız olduktan sonra, etkin yönetim ortamının başarısız olamayacağı bir süre önce beklemedeki yönetim ortamının yeniden başlatılmasından önce bir süre vardır.

***ASYNCL / *ASYNCL**

Etkin ve beklemedeki kuyruk yöneticilerini barındıran sunucuların her ikisi de *ASYNCL günlük kaydını kullanacak şekilde yapılandırılıyor.

1. Yedek sisteme geçiş ya da hata durumunda yedek sisteme geçiş işlemi gerçekleştiğinde, kuyruk yöneticisi günlükle yeni sunucuda devam eder. Yedek sisteme geçiş ya da hata durumunda yedek sisteme geçiş gerçekleştiğinde günlük eşitlenmeyebilir. Sonuç olarak iletiler kaybolabilir ya da yinelenebilir.
2. Etkin yönetim ortamı, beklemedeki kuyruk yöneticisini bulunduran sunucu kullanılabilir durumda olsa bile çalışır. Yerel günlük, kullanılabilir olduğunda bekleme sunucusuyla zamanuyumsuz olarak eşlenir.
3. Yerel kuyruk yöneticisinin başarımı, uzak günlüğe kaydetme işlemi tarafından geçersiz kılarak etkilenmez.

Performans asıl gereksiniminiz ise *ASYNCL / *ASYNCL seçeneğini belirleyin ve hata durumunda yedek sisteme geçiş ya da yedek sisteme geçiş işleminden sonra bazı iletileri gevşetmeye ya da yinelenmeye hazırlıksınız.

***ASYNCL / *SYNCL**

Bu seçenek bileşimini kullanmak için herhangi bir neden yoktur.

Uzak günlükten kuyruk yöneticisi etkinleştirilmesi

Günlükler zamanuyumlu olarak ya da zamanuyumsuz olarak eşlenir. Uzak günlük etkin olmayabilir ya da yerel günlükle kapsanmış olabilir. Uzak günlük, zamanuyumlu olarak eşlenmiş olsa da, kısa süre önce

etkinleştirilmiş olabileceği için, kapma işlemi devam edebilir. Kuyruk yöneticisinin, başlatma sırasında kullandığı uzak günlüğün durumu için geçerli olduğu kurallar aşağıdaki gibidir.

1. Beklemedeki başlatma, yedekleme ve günlük durumu *FAILED ya da *INACTPENDise, yedek başlatma işlemi başarısız olur.
2. Yedek veritabanının etkinleştirilmesi başladığında, beklemedeki uzak günlük durumunun *ACTIVE ya da *INACTIVE olması gerekir. Durum *INACTIVE ise, tüm günlük verileri eşlenmezse, etkinleştirme işleminin başarısız olması mümkün olur.

Hata, ağ dosya sistemindeki kuyruk yöneticisi verilerinin, uzak günlükle ilgili daha yeni bir denetim noktası kaydına sahip olması durumunda ortaya çıkar. Uzak günlük, denetim noktaları arasındaki varsayılan 30 dakika üst sınırı içinde iyi bir şekilde etkinleştirildiği sürece, bu hatanın ortaya konması olası değildir. Beklemedeki kuyruk yöneticisi, dosya sisteminden daha yeni bir denetim noktası kaydı okuyorsa, bu kayıt başlatılmaz.

Bir seçeneğiniz vardır: Etkin sunucudaki yerel günlüğün geri yüklenebilmesi ya da bekleme bekleme kuyruğu yöneticisinin soğuması için beklenecek bir beklenti vardır. Soğuk başlatmayı seçerseniz, kuyruk yöneticisi günlük verileri olmadan başlar ve dosya sistemindeki kuyruk yöneticisi verilerinin tutarlılık ve eksiksizliğine dayanır.

Not: Bir kuyruk yöneticisini soğuk olarak başlatırsanız, son denetim noktasından sonra iletileri kaybetme ya da çoğaltma riskini çalıştırırsınız. İletiler hareketleri günlüğe yazıldı, ancak bazı işlemler, dosya sistemindeki kuyruk yöneticisi verilerine yazılmamış olabilir. Bir kuyruk yöneticisini soğuk olarak başlattığınızda, yeni bir günlük başlatılır ve dosya sistemindeki kuyruk yöneticisi verilerine yazılmamış işlemler kaybolur.

3. Beklemedeki kuyruk yöneticisi etkinleştirme işlemi, beklemedeki uzak günlük durumunun *ASYNCPEND ya da *SYNCPEND ' den *ASYNCPEND ya da *SYNCPEND değerine değişmesini bekler. İletiler yürütme denetleyicisinin iş günlüğüne düzenli aralıklarla yazılır.

Not: Bu durumda etkinleştirme, etkinleştirmekte olan beklemedeki kuyruk yöneticisinde yerel yerel yerel günlüğü bekliyor. Kuyruk yöneticisi, uzak günlük olmadan da devam etmeden önce bir süre bekler. Uzak günlüğüne (ya da günlüklere) zamanuyumlu olarak yazma girişiminde bulunduğu ve günlük kullanılabilir durumda değilse bekler.

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 üzerinde günlük ikizleme ve NetServer kullanan çok eşgörünümlü bir kuyruk yöneticisi yaratılması

İki IBM i sunucusunda çalışmak için çok eşgörünümlü bir kuyruk yöneticisi yaratın. Kuyruk yöneticisi verileri, NetServer' ı kullanarak üçüncü bir IBM i sunucusunda saklanır. Kuyruk yöneticisi günlüğü, uzak günlüğe kaydetme özelliğini kullanarak iki sunucu arasında ikizlenir. The **ADDQMQRN** command is used to simplify creating the remote journals.

Başlamadan önce

1. Görev için üç IBM i sunucusu gerekir. Install IBM MQ on two of them, ALPHA and BETA in the example. ürünün en az IBM WebSphere MQ 7.0.1 Fix Pack 1 konumunda olması gerekir.
2. Üçüncü sunucu, NetServer ile ALPHA ve BETA ' ya bağlı bir IBM i sunucudur. Kuyruk yöneticisi verilerini paylaşmak için kullanılır. Bir IBM MQ kuruluşuna sahip olmak zorunda değildir. It is useful to install IBM MQ on the server as a temporary step, to set up the queue manager directories and permissions.
3. QMQM kullanıcı tanıtımının tüm üç sunucularda aynı parolaya sahip olduğundan emin olun.
4. IBM i NetServer' ı kurun; bkz. [i5/OS NetServer](#).

Bu görev hakkında

Şekil 28 sayfa 385 içinde gösterilen yapılandırmayı oluşturmak için aşağıdaki adımları gerçekleştirin. Kuyruk yöneticisi verileri, IBM i NetServer kullanılarak bağlantılıdır.

- Kuyruk yöneticisi verilerini saklamak için ALFA ve BETA 'dan izin paylaşımı için ALPHA ve BETA' dan bağlantılar yaratın. Görev, gerekli izinleri, kullanıcı profillerini ve parolaları da ayarlar.
- Kuyruk yöneticisi yönetim ortamlarını çalıştıracığı IBM i sistemlerine İlişkisel Veri Tabanı Girişleri (RDBE) ekleyin. RDBE girişleri, uzak günlüğe kaydetme işlemi için kullanılan IBM i sistemlerine bağlanmak için kullanılır.
- Create the queue manager QM1 on the IBM i server, ALPHA.
- Diğer IBM i sunucusu BETA ' da QM1 için kuyruk yöneticisi denetim bilgilerini ekleyin.
- Hem kuyruk yöneticisi yönetim ortamları için hem de IBM i sunucularında uzak günlükler yaratın. Her kuyruk yöneticisi yerel günlüğe yazar. Yerel günlük, uzak günlüğe kopyalanır. **ADDQMJRN** , günlüklerin ve bağlantıların eklenmesini basitleştirir.
- Kuyruk yöneticisini başlatın ve bir yedek yönetim ortamı izin veriyor.

Yordam

1. Do the task, “IBM üzerinde NetServer kullanarak kuyruk yöneticisi verileri için ağ paylaşımı yaratılması” sayfa 370.

Sonuç olarak, ALPHA ve BETA 'nın GAMMA üzerinde /QIBM/UserData/mqm/qmgrs ' e işaret eden bir paylaşım (/QNTC/GAMMA/WMQ) vardır. QMQM ve QMQMADM kullanıcı profilleri gerekli izinlere sahiptir ve QMQM , tüm üç sistemde parolaları eşleştirir.

2. İlişkisel Veri Tabanı Girişleri 'ni (RDBE), anasistem kuyruk yöneticisi eşgörünümlerine giden IBM i sistemlerine ekleyin.

a) ALPHA 'da BETA' ya bağlantı oluşturun.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

b) BETA 'da ALPHA' ya olan bağlantıları oluşturun.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. Create the queue manager QM1 on ALPHA, saving the queue manager data on GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)  
MQMDIRP(' /QNTC/GAMMA/WMQ ')
```

Yol, kuyruk yöneticisi verilerini yaratmak için NetServer nesnesini kullanır.

4. ALPHA ' yı çalıştır. Bu komut, BETA ' ya uzak bir günlük ekler.

```
ADDQMJRN MQMNAME(QM1) RMTJRNRDB(BETA)
```

Etkin yönetim ortamı ALPHA 'da olduğu zaman, yerel günlüğünde ALPHA' da günlük girişleri yaratır. ALPHA 'daki yerel günlük, BETA' daki uzak günlüğe kopyalanıyor.

5. ALPHA için oluşturulan IBM MQ yapılandırma verilerini incelemek için komutu kullanın.

Sonraki adımda bilgi gereklidir.

Bu örnekte, aşağıdaki yapılandırma ALPHA ' da yaratılır;

```
Name=QM1  
Prefix=/QIBM/UserData/mqm
```

```
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

6. Komutu kullanarak BETA ' da QM1 kuyruk yöneticisi eşgörünümü yaratın. BETA 'daki kuyruk yöneticisi denetim bilgilerini değiştirmek için BETA' da aşağıdaki komutu çalıştırın.

```
ADDQMOMINF MQMNAME(QM1)
PREFIX('/QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMOM1)
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

İpucu: Yapılandırma bilgilerini kopyalayın ve yapıştırın. Kuyruk yöneticisi Stanza, ALPHA ve BETA ' da aynı.

7. BETA ' da kaç. Bu komut, BETA 'da yerel bir günlük ve ALPHA' da uzak bir günlük ekler.

```
ADDQMOMJRN MQMNAME(QM1) RMTJRNADB(ALPHA)
```

Etkin eşgörünümü BETA 'da olduğunda, günlük girişlerini, BETA' daki yerel günlüğünde yaratır. BETA 'daki yerel günlük ALLPHA' daki uzak günlüğe kopyalanıyor.

Not: Diğer bir seçenek olarak, zamanuyumsuz günlük kaydı kullanarak BETA 'dan ALPHA' ya uzak günlük kaydı ayarlamak isteyebilirsiniz.

“7” sayfa 384adımındaki komut yerine BETA 'dan ALPHA' ya zamanuyumsuz günlük kaydı ayarlamak için bu komutu kullanın.

```
ADDQMOMJRN MQMNAME (QM1) RMTJRNADB (ALPHA) RMTJRNADLV (*ASYNCR)
```

Sunucu ya da günlük kaydı ALPHA 'da hatanın kaynağı varsa, BETA yeni günlük girişlerinin ALPHA' ya kopyalanacağını beklemeden başlar.

Eşleme kipini *SYNCR olarak değiştirin, komutu kullanarak, ALPHA yeniden çevrimiçi olur.

Günlüklerin zamanuyumsuz olarak mı, yoksa zamanuyumsuz olarak mı, yoksa ikisinin karışımı mı olduğuna karar vermek için “[IBM üzerinde ASP için ikizlenmiş günlük yapılandırması](#)” sayfa 377 içindeki bilgileri kullanın. Varsayılan değer, uzak günlükten gelen bir yanıt için 60 saniyelik bekleme süresiyle zamanuyumlu olarak eşlenmektedir.

8. ALPHA ve BETA ' daki günlüklerin etkinleştirildiğini ve uzak günlük kopyalamasının durumunun olduğunu doğrulayın.

a) ALFA ' DA:

```
WRKMOMJRN MQMNAME(QM1)
```

b) -Evet.

```
WRKMOMJRN MQMNAME(QM1)
```

9. ALPHA ve BETA ' da kuyruk yöneticisi eşgörünümlerini başlatın.

a) ALPHA ' da ilk yönetim ortamını başlatın ve etkin yönetim ortamı haline bakın. Yedek bir yönetim ortamına geçmenin etkinleştirilmesi.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) BETA ' da ikinci eşgörünümü başlatın ve yedek yönetim ortamı haline getirsin.

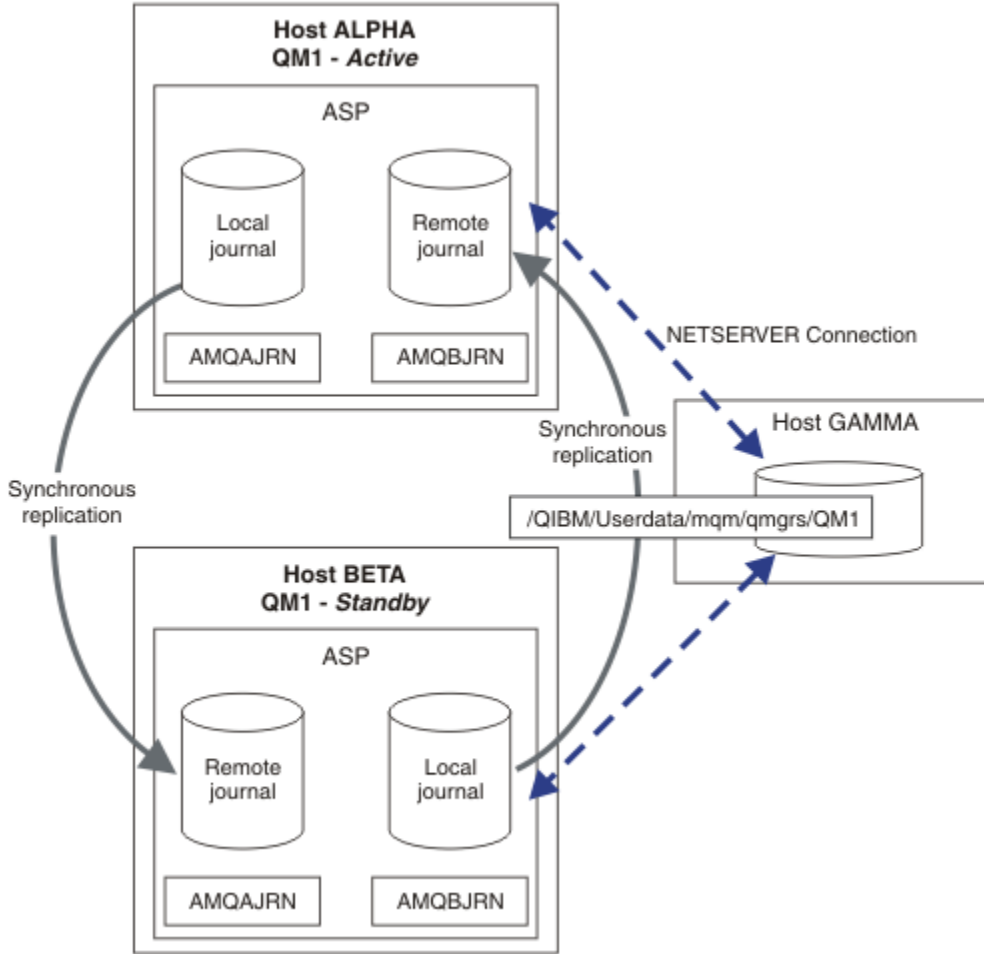
```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```


Sonuçlar

Kuyruk yöneticisi durumunu denetlemek için kullan:

1. ALPHA 'daki kuyruk yöneticisi örneğinin durumu olmalıdır.
2. BETA 'da kuyruk yöneticisi yönetim ortamının durumu olmalıdır.

Örnek



Şekil 28. İkizlenmiş günlük yapılandırması

Sonraki adım

- Etkin ve beklemedeki eşgörünümlerin otomatik olarak değişip geçmediğini denetleyin. Anahtarı test etmek için örnek yüksek kullanılabilirlik örnek programlarını çalıştırabilirsiniz; bkz. [Yüksek kullanılabilirlik örnek programları](#). Örnek programlar, 'C' istemcilerdir. Bunları bir Windows ya da Unix altyapısından çalıştırabilirsiniz.

1. Yüksek kullanılabilirlik örnek programlarını başlatın.
2. ALLPHA 'da, anahtar isteğinde bulunan kuyruk yöneticisini sona erdirin:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. BETA örneğinin etkin olup olmadığını denetleyin.

4. RESTART üzerinde yeniden başlat

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Diğer yüksek kullanılabilirlik yapılandırmalarına bakın:
 1. Kuyruk yöneticisi verilerini bir Windows sunucusuna yerleştirmek için NetServer ' ı kullanın.
 2. Kuyruk yöneticisi günlüğünü ikizlemek için uzak günlük kaydı işlevini kullanmak yerine, günlüğü bağımsız bir ASP ' ye saklayın. Bağımsız ASP 'yi ALPHA' dan BETA ' ya aktarmak için IBM i kümeleme olanağını kullanın.

IBM i IBM üzerinde NetServer ve günlük ikizlemesi kullanılarak, tek bir eşgörünüm kuyruk yöneticisinin çok eşgörünümlü 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 günlüğe kaydetme özelliğini kullanarak kuyruk yöneticisi günlüğünü ikinci bir IBM i sunucusuna ikizler.

Başlamadan önce

1. Görev için üç IBM i sunucusu gerekir. Örneğin, var olan IBM MQ kurulumunun en az IBM WebSphere MQ 7.0.1 Fix Pack 1 konumunda olması gerekir. ALFA, örnekte QM1 adlı bir kuyruk yöneticisi çalıştırıyor.
2. Install IBM MQ on the second IBM i server, BETA in the example.
3. Üçüncü sunucu, NetServer ile ALPHA ve BETA ' ya bağlı bir IBM i sunucudur. Kuyruk yöneticisi verilerini paylaşmak için kullanılır. Bir IBM MQ kuruluşuna sahip olmak zorunda değildir. It is useful to install IBM MQ on the server as a temporary step, to set up the queue manager directories and permissions.
4. QMQM kullanıcı tanıtımının tüm üç sunucularda aynı parolaya sahip olduğundan emin olun.
5. IBM i NetServer' ı kurun; bkz. [i5/OS NetServer](#).

Bu görev hakkında

Tek bir eşgörünüm kuyruk yöneticisini [Şekil 29 sayfa 390](#) 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 bağlanan ağ paylaşımında saklayarak yeniden yaratılır. This procedure is more reliable than moving the queue manager directories and files to the network share using the **CPY** command.

- Kuyruk yöneticisi verilerini saklamak için ALFA ve BETA 'dan izin paylaşımı için ALPHA ve BETA' dan bağlantılar yaratın. Görev, gerekli izinleri, kullanıcı profillerini ve parolaları da ayarlar.
- Kuyruk yöneticisi yönetim ortamlarını çalıştıracığı IBM i sistemlerine İlişkisel Veri Tabanı Girişleri (RDBE) ekleyin. RDBE girişleri, uzak günlüğe kaydetme işlemi için kullanılan IBM i sistemlerine bağlanmak için kullanılır.
- Kuyruk yöneticisi günlüklerini ve tanımlamalarını saklayın, kuyruk yöneticisini durdurun ve silin.
- Kuyruk yöneticisini yeniden yaratın, kuyruk yöneticisi verilerini GAMMA ' daki ağ paylaşımına göre saklayın.
- Kuyruk yöneticisinin ikinci örneğini diğer sunucuya ekleyin.
- Hem kuyruk yöneticisi yönetim ortamları için hem de IBM i sunucularında uzak günlükler yaratın. Her kuyruk yöneticisi yerel günlüğe yazar. Yerel günlük, uzak günlüğe kopyalanır. **ADDQMJRN** , günlüklerin ve bağlantıların eklenmesini basitleştirir.
- Kuyruk yöneticisini başlatın ve bir yedek yönetim ortamı izin veriyor.

Not:

Görevin "4" sayfa 387 . adımında, tek örnek kuyruk yöneticisini (QM1) sildiniz. Kuyruk yöneticisinin silinmesi, kuyruklardaki tüm kalıcı iletilerin silinmesine neden olur. Bu nedenle, kuyruk yöneticisini dönüştürmeden önce, kuyruk yöneticisi tarafından saklanan tüm iletilerin işlenmesini tamamlayın. If

processing all the messages is not possible, back up the queue manager library before step “4” sayfa 387. Restore the queue manager library after step “5” sayfa 387.

Not:

Görevin “5” sayfa 387 . adımımda QM1' u yeniden yaratın. Kuyruk yöneticisi aynı ada sahip olsa da, kuyruk yöneticisi tanıtıcısı farklı. Kuyruk yöneticisi kümelemesi, kuyruk yöneticisi tanıtıcısını kullanır. Bir kümede kuyruk yöneticisini silmek ve yeniden yaratmak için, önce kuyruk yöneticisini kümeden kaldırmalısınız; bkz. Bir kümeden kuyruk yöneticisinin kaldırılması: Alternatif yöntem ya da Kümeden kuyruk yöneticisini kaldırma. Kuyruk yöneticisini yeniden yarattığınız zaman, kümeyi kümeye ekleyin. Önceki adla aynı ada sahip olsa da, küme içindeki diğer kuyruk yöneticilerine yeni bir kuyruk yöneticisi gibi görünüyor.

Yordam

1. Do the task, “IBM üzerinde NetServer kullanarak kuyruk yöneticisi verileri için ağ paylaşımı yaratılması” sayfa 370.

Sonuç olarak, ALPHA ve BETA 'nın GAMMA üzerinde /QIBM/UserData/mqm/qmgrs ' e işaret eden bir paylaşım (/QNTC/GAMMA/WMQ) vardır. QMQM ve QMQMADM kullanıcı profilleri gerekli izinlere sahiptir ve QMQM , tüm üç sistemde parolaları eşleştirir.

2. İlişkisel Veri Tabanı Girişleri 'ni (RDBE), anasistem kuyruk yöneticisi eşgörünümlerine giden IBM i sistemlerine ekleyin.

a) ALPHA 'da BETA' ya bağlantı oluşturun.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

b) BETA 'da ALPHA' ya olan bağlantıları oluşturun.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. Kuyruk yöneticisi nesnelere yeniden yaratan komut dosyalarını yaratın.

```
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. Create the queue manager QM1 on ALPHA, saving the queue manager data on GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)  
MQMDIRP(' /QNTC/GAMMA/WMQ ')
```

Yol, kuyruk yöneticisi verilerini yaratmak için NetServer nesnesini kullanır.

6. Saklanmış tanımlamalardan QM1 için kuyruk yöneticisi nesnelere yeniden yaratın.

```
STRMQMQSC SRCMBR(QM1) SRCFILE(*CURLIB/QMQSC) MQMNAME(QM1)
```

7. Yetkileri saklanan bilgilerden uygulayın.

a) Saklanmış 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 için güvenlik bilgilerini yenileyin.

```
RFRMQMAUT MQMNAME(QM1)
```

8. ALPHA ' yı çalıştır. Bu komut, BETA ' ya uzak bir günlük ekler.

```
ADDMQMJRN MQMNAME(QM1) RMTJRNRDB(BETA)
```

Etkin yönetim ortamı ALPHA 'da olduğu zaman, yerel günlüğünde ALPHA' da günlük girişleri yaratır. ALPHA 'daki yerel günlük, BETA' daki uzak günlüğe kopyalanıyor.

9. ALPHA için oluşturulan IBM MQ yapılandırma verilerini incelemek için komutu kullanın.

Sonraki adımda bilgi gereklidir.

Bu örnekte, aşağıdaki yapılandırma ALPHA ' da yaratılır;

```
Name=QM1  
Prefix=/QIBM/UserData/mqm  
Library=QMQM1  
Directory=QM1  
DataPath= /QNTC/GAMMA/WMQ /QM1
```

10. Komutu kullanarak BETA ' da QM1 kuyruk yöneticisi eşgörünümü yaratın. BETA 'daki kuyruk yöneticisi denetim bilgilerini değiştirmek için BETA' da aşağıdaki komutu çalıştırın.

```
ADDMQMINF MQMNAME(QM1)  
PREFIX('/QIBM/UserData/mqm')  
MQMDIR(QM1)  
MQMLIB(QMQM1)  
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

İpucu: Yapılandırma bilgilerini kopyalayın ve yapıştırın. Kuyruk yöneticisi Stanza, ALPHA ve BETA ' da aynı.

11. BETA ' da kaç. Bu komut, BETA 'da yerel bir günlük ve ALPHA' da uzak bir günlük ekler.

```
ADDMQMJRN MQMNAME(QM1) RMTJRNRDB(ALPHA)
```

Etkin eşgörünümü BETA 'da olduğunda, günlük girişlerini, BETA' daki yerel günlüğünde yaratır. BETA 'daki yerel günlük ALLPHA' daki uzak günlüğe kopyalanıyor.

Not: Diğer bir seçenek olarak, zamanuyumsuz günlük kaydı kullanarak BETA 'dan ALPHA' ya uzak günlük kaydı ayarlamak isteyebilirsiniz.

“7” sayfa 384adımındaki komut yerine BETA 'dan ALPHA' ya zamanuyumsuz günlük kaydı ayarlamak için bu komutu kullanın.

```
ADDMQMJRN MQMNAME (QM1) RMTJRNRDB (ALPHA) RMTJRNDLV (*ASYNC)
```

Sunucu ya da günlük kaydı ALPHA 'da hatanın kaynağı varsa, BETA yeni günlük girişlerinin ALPHA' ya kopyalanacağını beklemeden başlar.

Eşleme kipini *SYNColarak değiştirin, komutu kullanarak, ALPHA yeniden çevrimiçi olur.

Günlüklerin zamanuyumsuz olarak mı, yoksa zamanuyumsuz olarak mı, yoksa ikisinin karışımı mı olduğuna karar vermek için [“IBM üzerinde ASP için ikizlenmiş günlük yapılandırması” sayfa 377](#) içindeki bilgileri kullanın. Varsayılan değer, uzak günlükten gelen bir yanıt için 60 saniyelik bekleme süresiyle zamanuyumlu olarak eşlenmektedir.

12. ALPHA ve BETA 'daki günlüklerin etkinleştirildiğini ve uzak günlük kopyalamanın durumunun olduğunu doğrulayın.

a) ALFA ' DA:

```
WRKMQMJRN MQMNAME(QM1)
```

b) -Evet.

```
WRKMQMJRN MQMNAME(QM1)
```

13. ALPHA ve BETA 'da kuyruk yöneticisi eşgörünümlerini başlatın.

a) ALPHA 'da ilk yönetim ortamını başlatın ve etkin yönetim ortamı haline bakın. Yedek bir yönetim ortamına geçmenin etkinleştirilmesi.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) BETA 'da ikinci eşgörünümü başlatın ve yedek yönetim ortamı haline getirsin.

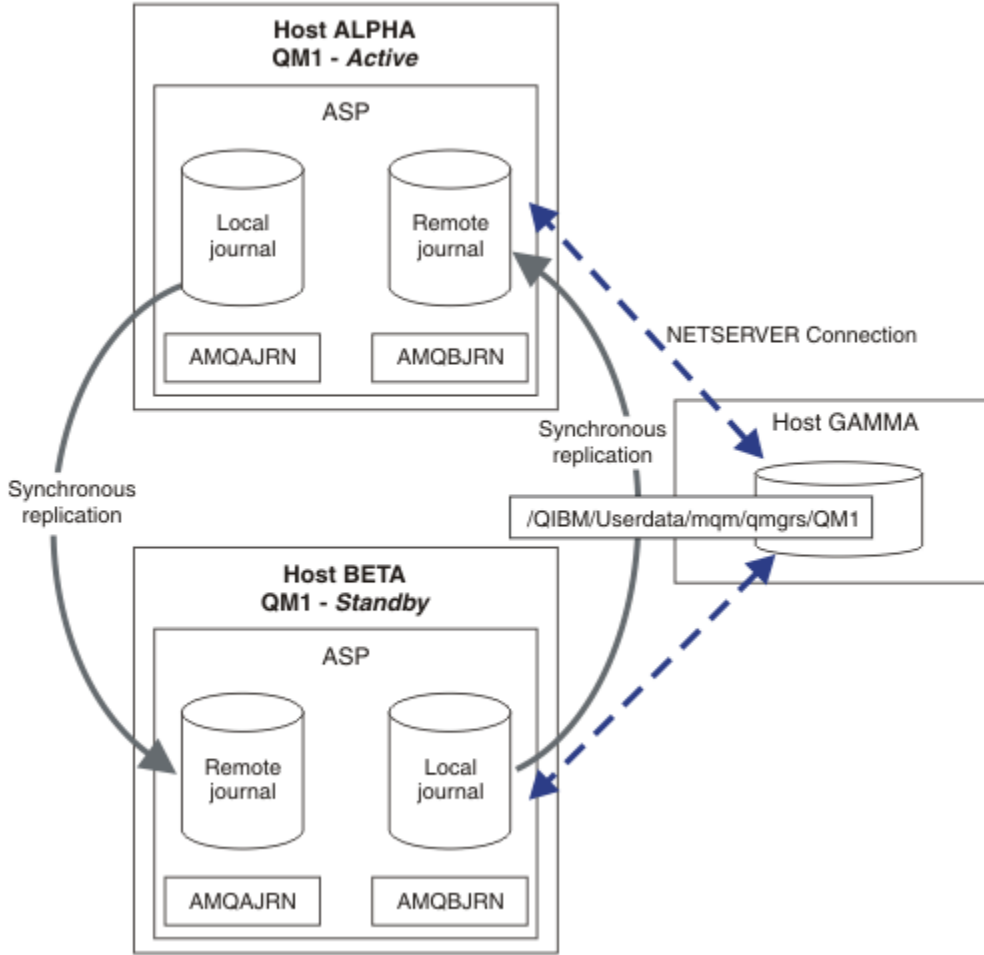
```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

Sonuçlar

Kuyruk yöneticisi durumunu denetlemek için kullan:

1. ALPHA 'daki kuyruk yöneticisi örneğinin durumu olmalıdır.
2. BETA 'da kuyruk yöneticisi yönetim ortamının durumu olmalıdır.

Örnek



Şekil 29. İkizlenmiş günlük yapılandırması

Sonraki adım

- Etkin ve beklemedeki eşgörünümlerin otomatik olarak değişip geçmediğini denetleyin. Anahtarı test etmek için örnek yüksek kullanılabilirlik örnek programlarını çalıştırabilirsiniz; bkz. [Yüksek kullanılabilirlik örnek programları](#). Örnek programlar, 'C' istemcilerdir. Bunları bir Windows ya da Unix altyapısından çalıştırabilirsiniz.

1. Yüksek kullanılabilirlik örnek programlarını başlatın.
2. ALLPHA ' da, anahtar isteğinde bulunan kuyruk yöneticisini sona erdirin:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. BETA örneğinin etkin olup olmadığını denetleyin.
4. RESTART üzerinde yeniden başlat

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Diğer yüksek kullanılabilirlik yapılandırmalarına bakın:

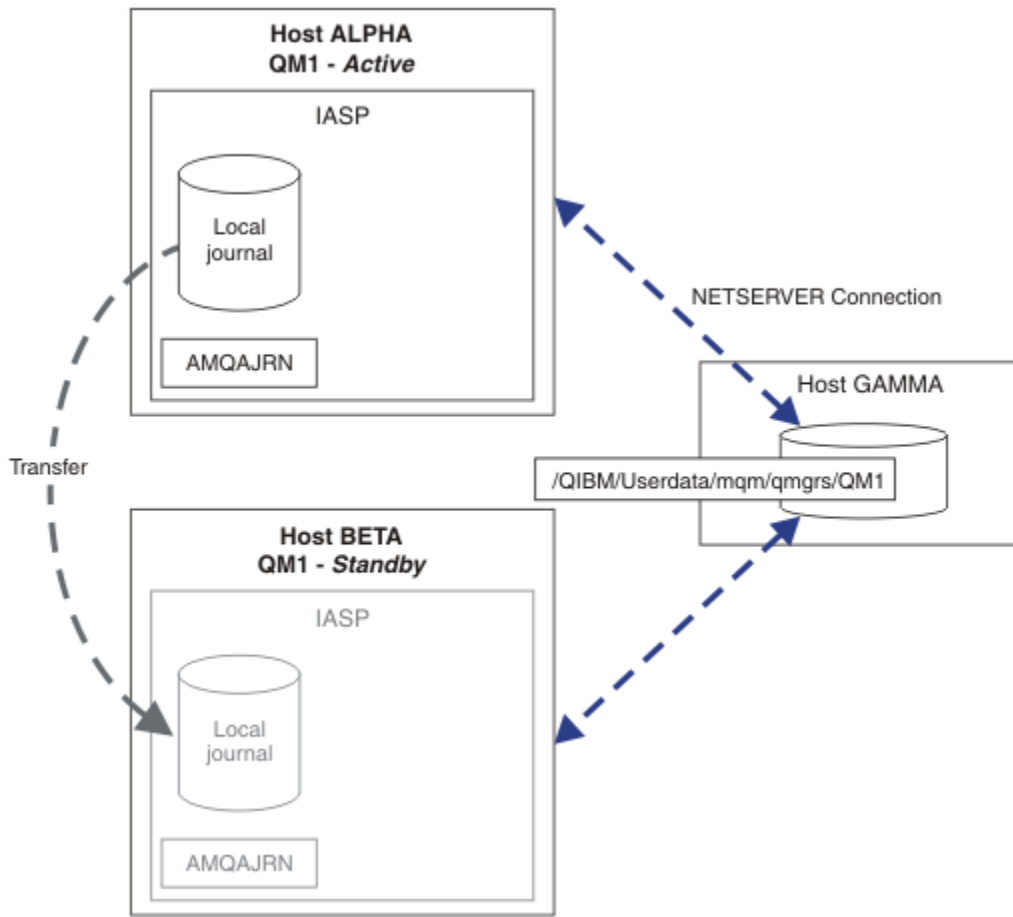
1. Kuyruk yöneticisi verilerini bir Windows sunucusuna yerleştirmek için NetServer ' ı kullanın.

2. Kuyruk yöneticisi günlüğünü ikizlemek için uzak günlük kaydı işlevini kullanmak yerine, günlüğü bağımsız bir ASP 'ye saklayın. Bağımsız ASP 'yi ALPHA' dan BETA 'ya aktarmak için IBM i kümeleme olanağını kullanın.

IBM i **IBM üzerinde anahtarlamalı bağımsız ASP günlük yapılandırması**

Çok eşgörünümlü bir kuyruk yöneticisi yapılanışı yaratmak için bağımsız bir ASP günlüğünü çoğaltmanıza gerek yoktur. Bağımsız ASP 'yi etkin kuyruk yöneticisinden beklemedeki kuyruk yöneticisine aktarmak için bir aracı otomatikleştirmeniz gerekir. Her biri çok eşgörünümlü bir kuyruk yöneticisinin kullanılmasını gerektirmeyen, bağımsız bir ASP 'yi kullanarak mümkün olan alternatif yüksek kullanılabilirlik çözümleri vardır.

Bağımsız bir ASP 'yi kullanırken, kuyruk yöneticisi günlüğünü ikizlemenize gerek yoktur. Küme yönetimi kuruluysa ve kuyruk yöneticisi eşgörünümünü barındıran sunucular aynı küme kaynağı grubunda yer alıyorsa, etkin yönetim ortamını çalıştıran anasistem başarısız olursa, kuyruk yöneticisi günlüğü etkin sunucunun kısa bir uzaklığı içinde başka bir sunucuya otomatik olarak aktarılabilir. Günlüğü, planlanan anahtarın bir parçası olarak el ile de aktarabilir ya da bağımsız ASP 'yi programlı olarak aktarmak için bir komut yordamı yazabilirsiniz.



Şekil 30. Bir kuyruk yöneticisi günlüğünü bağımsız bir ASP kullanarak aktarma

Çok eşgörünümlü kuyruk yöneticisi işlemi için, kuyruk yöneticisi verileri paylaşılan bir kütük sisteminde saklanmalıdır. Dosya sistemi çeşitli platformlarda barındırılabilir. Bir ASP 'de ya da bağımsız bir ASP' de çok eşgörünümlü kuyruk yöneticisi verilerini saklayamazsınız.

Paylaşılan kütük sistemi yapılanıştaki iki rolü gerçekleştirir: Aynı kuyruk yöneticisi verileri, kuyruk yöneticisinin tüm yönetim ortamlarını daha iyi paylaşır. Dosya sisteminde, kuyruk yöneticisinin yalnızca bir eşgörünümünün, başlatıldıktan sonra kuyruk yöneticisi verilerine erişimi olmasını güvenceye alan sağlam bir kilitleme protokolü olmalıdır. Kuyruk yöneticisi başarısız olursa ya da dosya sunucusu kesilirse, dosya

sistemi, artık dosya syste ile iletişimde olmayan etkin eşgörünüm tarafından tutulan kuyruk yöneticisi verilerine kilitlenme işlemini serbest bırakmalıdır. Bundan sonra, beklemedeki kuyruk yöneticisi yönetim ortamı kuyruk yöneticisi verilerine okuma/yazma erişimi kazanabilir. Dosya sistemi iletişim kuralı, çok eşgörünümlü kuyruk yöneticileriyle doğru bir şekilde çalışmak için bir kural kümesine uygun olmalıdır; bkz. [“IBM üzerinde yüksek kullanılabilirlik çözümünün bileşenleri” sayfa 368.](#)

Kilitleme mekanizması, kuyruk yöneticisini başlatma komutunu diziselleştirir ve kuyruk yöneticisinin hangi yönetim ortamının etkin olduğunu denetler. Kuyruk yöneticisi etkinleştikten sonra, kuyruklarını yerel günlükten ya da HA kümesinin yedek sunucuya aktardığı yerel günlükten yeniden oluşturur. Aynı kuyruk yöneticisi ile yeniden bağlantı için bekleyen yeniden bağlanabilir istemciler yeniden bağlanmaya devam eder ve tüm hafif işlemler geriletilir. Kuyruk yöneticisi hizmetleri olarak başlatılacak şekilde yapılandırılan uygulamalar başlatılır.

Bağımsız ASP 'deki hatalı etkin kuyruk yöneticisi eşgörünümünden yerel günlüğün, küme kaynak yöneticisini yapılandırarak ya da bağımsız ASP 'yi el ile aktararak, yeni etkinleştirilen beklemedeki kuyruk yöneticisi yönetim ortamını 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 kullanırsanız, uzak günlüklerin ve ikizlemenin konfigürasyonunun tanımlanmasını engeller.

Bağımsız bir yardımcı bellek havuzu (ASP) kullanmayı seçtiyseniz, göz önünde bulundurmanız gereken diğer yüksek düzeyde kullanılabilir yapılandırmalar vardır. Bu çözümlere ilişkin arka plan, [“Bağımsız ASP 'ler ve yüksek kullanılabilirlik” sayfa 395](#) içinde açıklanmıştır.

1. Çok eşgörünümlü kuyruk yöneticilerini kullanmak yerine, tek bir yönetim ortamı kuyruk yöneticisini tamamen bağımsız bir ASP 'ye kurun ve yapılandırın ve kuyruk yöneticisini sona ermek için IBM i yüksek kullanılabilirlik hizmetlerini kullanın. Kuyruk yöneticisinin sunucudan bağımsız olarak başarısız olup olmadığını saptamak için, bir kuyruk yöneticisi izleyicisiyle çözümü genişletmeniz gerekebilir. Bu, *Supportpac MC41: Configuring IBM MQ for iSeries for High Availability* da sağlanan çözümün temelini teşkil eder.
2. Bağımsız ASP 'yi yerel veriyolunda bağımsız ASP 'yi değiştirmek yerine bağımsız ASP 'yi ikizlemek için bağımsız ASP 'ler ve siteler arası ikizleme (XSM) kullanın. Bu, bağımsız ASP çözümünün coğrafi aralığını, uzun mesafeli bir olanak üzerinden günlük kayıtları yazmak için geçen zamana kadar genişletir.

IBM i *Creating a multi-instance queue manager using an independent ASP and NetServer on IBM i*
İki IBM i sunucusunda çalışmak için çok eşgörünümlü bir kuyruk yöneticisi yaratın. Kuyruk yöneticisi verileri, NetServer' ı kullanarak bir IBM i sunucusu depolanır. Kuyruk yöneticisi günlüğü bağımsız bir ASP 'de saklanır. Kuyruk yöneticisi günlüğünü içeren bağımsız ASP 'yi diğer IBM i Server sunucusuna aktarmak için IBM i kümeleme ya da el ile yordam kullanın.

Başlamadan önce

1. Görev için üç IBM i sunucusu gerekir. Install IBM MQ on two of them, ALPHA and BETA in the example. ürünün en az IBM WebSphere MQ 7.0.1 Fix Pack 1 konumunda olması gerekir.
2. Üçüncü sunucu, NetServer ile ALPHA ve BETA 'ya bağlı bir IBM i sunucudur. Kuyruk yöneticisi verilerini paylaşmak için kullanılır. Bir IBM MQ kuruluşuna sahip olmak zorunda değildir. It is useful to install IBM MQ on the server as a temporary step, to set up the queue manager directories and permissions.
3. QMQM kullanıcı tanıtımının tüm üç sunucularda aynı parolaya sahip olduğundan emin olun.
4. IBM i NetServer' ı kurun; bkz. [i5/OS NetServer](#).
5. Bağımsız ASP 'yi hatalı kuyruk yöneticisinden, devralacak yedek veritabanına aktarmak için gereken yordamları yaratın. Bağımsız ASP aktarım yordamlarınızı tasarlarken *SupportPac MC41: Configuring IBM MQ for iSeries for High Availability* yöntemlerinden bazılarını bulabilirsiniz.

Bu görev hakkında

[Şekil 31 sayfa 394](#) içinde gösterilen yapılandırmayı oluşturmak için aşağıdaki adımları gerçekleştirin. Kuyruk yöneticisi verileri, IBM i NetServer kullanılarak bağlantılıdır.

- Kuyruk yöneticisi verilerini saklamak için ALFA ve BETA 'dan izin paylaşımı için ALPHA ve BETA' dan bağlantılar yaratın. Görev, gerekli izinleri, kullanıcı profillerini ve parolaları da ayarlar.
- Create the queue manager QM1 on the IBM i server, ALPHA.
- Diğer IBM i sunucusu BETA ' da QM1 için kuyruk yöneticisi denetim bilgilerini ekleyin.
- Kuyruk yöneticisini başlatın ve bir yedek yönetim ortamı izin veriyor.

Yordam

1. Do the task, "IBM üzerinde NetServer kullanarak kuyruk yöneticisi verileri için ağ paylaşımı yaratılması" sayfa 370.

Sonuç olarak, ALPHA ve BETA 'nın GAMMA üzerinde /QIBM/UserData/mqm/qmgrs ' e işaret eden bir paylaşım (/QNTC/GAMMA/WMQ) vardır. QMQM ve QMQMADM kullanıcı profilleri gerekli izinlere sahiptir ve QMQM , tüm üç sistemde parolaları eşleştirir.

2. Create the queue manager QM1 on ALPHA, saving the queue manager data on GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIRP(' /QNTC/GAMMA/WMQ ')
```

Yol, kuyruk yöneticisi verilerini yaratmak için NetServer nesnesini kullanır.

3. ALPHA için oluşturulan IBM MQ yapılandırma verilerini incelemek için komutu kullanın.

Sonraki adımda bilgi gereklidir.

Bu örnekte, aşağıdaki yapılandırma ALPHA ' da yaratılır;

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

4. Komutu kullanarak BETA ' da QM1 kuyruk yöneticisi eşgörünümü yaratın. BETA 'daki kuyruk yöneticisi denetim bilgilerini değiştirmek için BETA' da aşağıdaki komutu çalıştırın.

```
ADDQMINF MQMNAME(QM1)
PREFIX(' /QIBM/UserData/mqm ')
MQMDIR(QM1)
MQMLIB(QMOM1)
DATAPATH(' /QNTC/GAMMA/WMQ /QM1 ')
```

İpucu: Yapılandırma bilgilerini kopyalayın ve yapıştırın. Kuyruk yöneticisi Stanza, ALPHA ve BETA ' da aynı.

5. ALPHA ve BETA ' da kuyruk yöneticisi eşgörünümlerini başlatın.

- a) ALPHA ' da ilk yönetim ortamını başlatın ve etkin yönetim ortamı haline bakın. Yedek bir yönetim ortamına geçmenin etkinleştirilmesi.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- b) BETA ' da ikinci eşgörünümü başlatın ve yedek yönetim ortamı haline getirsin.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

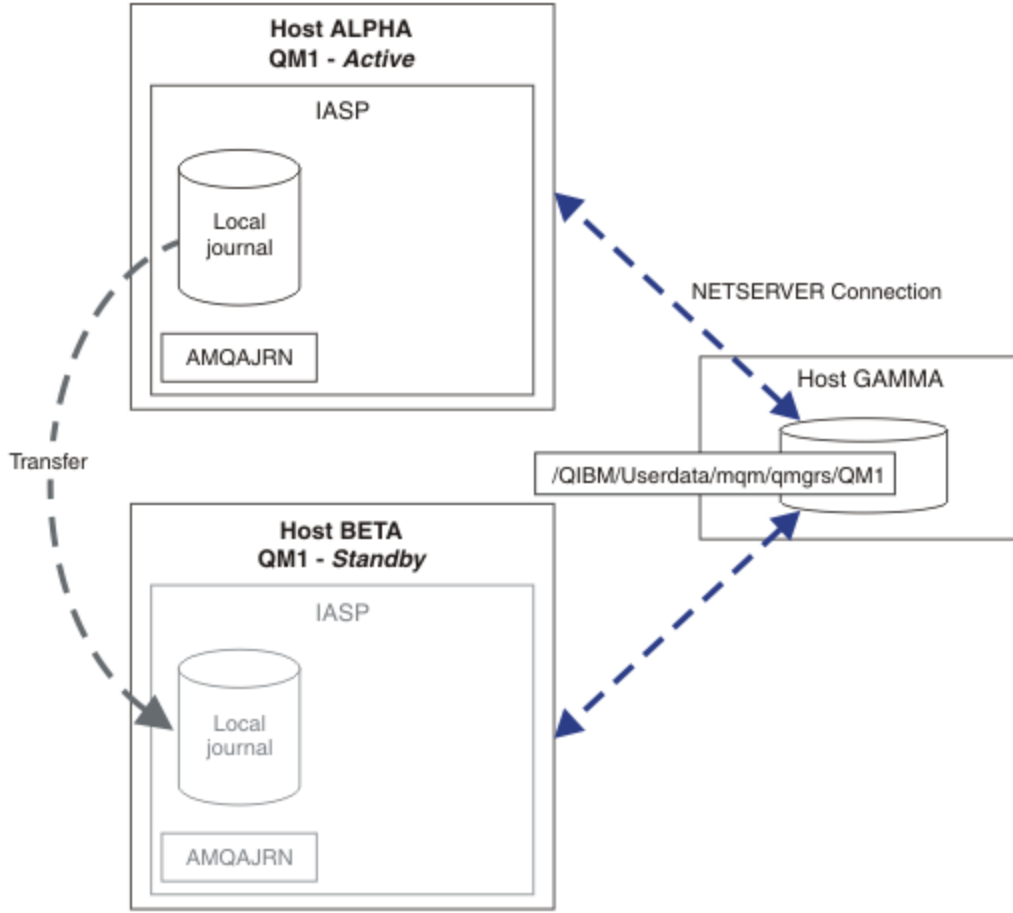
Sonuçlar

Kuyruk yöneticisi durumunu denetlemek için kullan:

1. ALPHA ' daki kuyruk yöneticisi örneğinin durumu olmalıdır.

2. BETA ' da kuyruk yöneticisi yönetim ortamının durumu olmalıdır.

Örnek



Şekil 31. Bir kuyruk yöneticisi günlüğünü bağımsız bir ASP kullanarak aktarma

Sonraki adım

- Etkin ve beklemedeki eşgörünümlerin otomatik olarak değişip geçmediğini denetleyin. Anahtarı test etmek için örnek yüksek kullanılabilirlik örnek programlarını çalıştırabilirsiniz; bkz. [Yüksek kullanılabilirlik örnek programları](#). Örnek programlar, 'C' istemcilerdir. Bunları bir Windows ya da Unix altyapısından çalıştırabilirsiniz.

1. Yüksek kullanılabilirlik örnek programlarını başlatın.
2. ALLPHA ' da, anahtar isteğinde bulunan kuyruk yöneticisini sona erdirin:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. BETA örneğinin etkin olup olmadığını denetleyin.
4. RESTART üzerinde yeniden başlat

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Diğer yüksek kullanılabilirlik yapılandırmalarına 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 günlüğe kaydetme özelliğini kullanın.

IBM i

Bağımsız ASP ' ler ve yüksek kullanılabilirlik

Bağımsız ASP ' ler, uygulamaların ve verilerin sunucular arasında taşınmasını sağlar. Bağımsız ASP ' lerin esnekliği, bazı IBM i yüksek kullanılabilirlik çözümlerinin temelini teşkil ettikleri anlamına gelir. Kuyruk yöneticisi günlüğü için bir ASP 'yi ya da bağımsız ASP' yi kullanıp kullanmayacağınızı göz önünde bulundurarak, bağımsız ASP ' lere dayalı diğer yüksek kullanılabilirlik yapılandırmasını göz önünde bulundurmanız gerekir.

Yardımcı bellek havuzları (ASP ' ler), IBM i mimarisinin bir oluşturma öbeğidir. Disk birimleri, tek bir ASP oluşturmak için birlikte gruplanır. Nesnelere farklı ASP 'lerde yerleştirilerek, bir ASP' deki verileri, başka bir ASP ' deki disk hatalarından etkilenmeden koruyabilirsiniz.

Her IBM i sunucusunun, sistem ASP 'si olarak bilinen en az bir *temel* ASP 'si vardır. Bu, ASP1olarak belirlenir ve bazen *SYSBAS olarak da bilinir. Aynı ad alanını paylaştıkları için, sistem ASP 'si tarafından uygulamanın bakış açısından ayırt edilemeyen en çok 31 adet temel *kullanıcı* ASP 'si yapılandırabilirsiniz. Birçok disk üzerinden uygulama dağıtmak için birden çok temel ASP ' yi kullanarak performansını artırabilir ve kurtarma süresini kısaltabilirsiniz. Birden çok temel ASP ' nin kullanılması, disk hatasına karşı bir derece yalıtım sağlayabilir, ancak genel olarak güvenilirliği artırmaz.

Bağımsız ASP ' ler özel bir ASP tipidir. Bunlar genellikle bağımsız disk havuzları olarak adlandırılırlar. Bağımsız disk havuzları, IBM i yüksek kullanılabilirliğinin temel bileşenleridir. Bağımsız disk depolama birimlerinde bağlı oldukları yürürlükteki sistemden bağımsız olarak, kendilerini bağımsız olarak dikkate alan verileri ve uygulamaları saklayabilirsiniz. Değişimli kullanılabilir ya da değişimli kullanılabilir olmayan bağımsız ASP ' leri yapılandırabilirsiniz. Kullanılabilirlik perspektifinden, genel olarak yalnızca değişimli kullanılabilir bağımsız ASP ' lerle ilgileniyorsanız, sunucudan sunucuya otomatik olarak aktarılabilir. Sonuç olarak, bağımsız ASP ' deki uygulamaları ve verileri sunucudan sunucuya taşıyabilirsiniz.

Temel kullanıcı ASP 'lerinin tersine, bağımsız ASP' ler sistem ASP ' niyle aynı ad alanını paylaşmaz. Kullanıcı ASP 'leriyle çalışan uygulamalar, bağımsız bir ASP' yle çalışmak için değişiklik gerektirir. Kullandığınız yazılımınızı ve kullandığınız üçüncü kişi yazılımların bağımsız bir ASP ortamında çalıştığını doğrulamanız gerekir.

Bağımsız ASP 'nin farklı bir sunucuya bağlandığında, bağımsız ASP' nin ad alanı, sistem ASP 'sinin ad alanı ile birleştirilmesi gerekir. This process is called *değişken-açık* the independent ASP. Bağımsız bir ASP 'yi sunucuyu IPL'lemeden farklı bir ASP' ye göre farklılık gösterebilirsiniz. Bağımsız ASP ' lerin otomatik olarak bir sunucudan başka bir sunucuya aktarılması için kümeleme desteği gerekir.

Bağımsız ASP ' lerle güvenilir çözümler oluşturma

Bir ASP 'ye günlük kaydı yapmak ve günlük kopyalamayı kullanmak yerine, bağımsız bir ASP' ye günlük kaydı, yedek kuyruk yöneticisini, başarısız olan kuyruk yöneticisi eşgörünümünden yerel günlüğün bir kopyasıyla sağlamak için alternatif bir yöntem sağlar. Bağımsız ASP ' yi otomatik olarak başka bir sunucuya aktarmak için, kümeleme desteğini kurmanız ve desteklemeniz gerekir. Bağımsız ASP ' ler için, küme desteğine ve düşük düzeyli disk ikizleme ile, çok eşgörünümlü kuyruk yöneticileriyle bir araya getirebileceğiniz ya da yerine konabileceğiniz düşük düzeyli disk ikizleme için bir dizi yüksek kullanılabilirlik çözümü vardır.

Aşağıdaki listede, bağımsız ASP ' lere dayalı güvenilir bir çözüm oluşturmak için gerekli bileşenler açıklanmaktadır.

Günlük kaydı

Kuyruk yöneticileri ve diğer uygulamalar, sunucu hatası nedeniyle bellekteki veri kaybına karşı korumak için güvenli verileri diske yazmak için yerel günlükler kullanır. Bu bazen zaman noktası tutarlılığı olarak adlandırılır. Bu, bir süre içinde gerçekleşen birden çok güncelliğin tutarlılığını garanti etmez.

Kesinleştirme denetimi

Genel hareketler kullanarak, günlüklere ve veritabanlarına ilişkin güncellemeleri koordine edebilirsiniz. Böylece, günlüğe yazılan veriler tutarlıdır. İki aşamalı kesinleştirme protokolünü kullanarak bir süre boyunca tutarlılığı sağlar.

Anahtarlamalı Disk

Anahtarlamalı diskler, bir HA kümesindeki aygıt kümesi kaynak grubu (CRG) tarafından yönetilir. CRG, bağımsız ASP ' leri planlanmamış bir hizmet kesilmesi durumunda otomatik olarak yeni bir sunucuya geçirir. CRG ' ler coğrafi olarak yerel GÇ veriyolunun kapsamından sınırlıdır.

Yerel günlüğünüzü, değişimli kullanılabilir bir bağımsız ASP ' de yapılandırarak, günlüğü farklı bir sunucuya aktarabilir ve ileti işleme işlemini sürdürebilirsiniz. Bağımsız ASP ' nin başarısız olması durumunda, syncpoint denetimi olmadan yapılan ya da syncpoint denetimiyle kesinleştirilmemiş, kalıcı iletilerde değişiklik yapılmaması gerekir.

Değişimli kullanılabilir bağımsız ASP ' lerde hem günlüğe kaydetme hem de kesinleştirme denetimi kullanıyorsanız, veritabanı günlüklerini ve kuyruk yöneticisi günlüklerini farklı bir sunucuya aktarabilir ve tutarsızlık ya da kesinleştirilmiş işlem kaybı olmadan işlem işlemlerini sürdürebilirsiniz.

Siteler arası ikizleme (XSM)

XSM, birincil bağımsız ASP 'yi bir TCP/IP ağı üzerinde coğrafi olarak uzak ikincil bağımsız ASP' ye kopyalar ve bir hata durumunda otomatik olarak denetim aktarır. Zamanuyumlu ya da zamanuyumsuz bir ikizleme konfigürasyonunu tanımlama seçeneğiniz vardır. Zamanuyumlu ikizleme, veri üretim sistemindeki yazma işlemlerinin tamamlanmadan önce ikizlendiği için kuyruk yöneticisinin performansını azaltır, ancak ikincil bağımsız ASP ' nin güncel olduğunu garanti eder. Ancak, zamanuyumsuz ikizleme kullanırsanız, ikincil bağımsız ASP ' nin güncel olduğunu garanti edemezsiniz. Zamanuyumsuz ikizleme, ikincil bağımsız ASP ' nin tutarlılığını korumaktadır.

Üç XSM teknolojisi var.

Coğrafi ikizleme

Coğrafi ikizleme, bağımsız ASP ' leri geniş bir alana geçmenize olanak sağlayan kümelemenin bir uzantısıdır. Hem zamanuyumlu hem de zamanuyumsuz kipleri vardır. Yüksek kullanılabilirlik durumunu yalnızca zamanuyumlu kipte garanti edebilirsiniz, ancak bağımsız ASP ' lerin ayrılması performansı çok fazla etkileyebilir. Yerel olarak yüksek düzeyde kullanılabilirlik sağlamak ve uzaktan olağanüstü durumdan kurtarma sağlamak için coğrafi ikizlemeyi anahtarlamalı diskle birleştirebilirsiniz.

Metro ikizleme

Metro ikizleme, yerel veriyolundan daha uzun mesafeler üzerinde hızlı yerel zamanuyumlu ikizleme sağlayan bir aygıt düzeyi hizmetidir. Bunu, kuyruk yöneticisinin yüksek düzeyde kullanılabilirliği ve bağımsız ASP ' nin iki kopyasını, kuyruk yöneticisi günlüğünü yüksek düzeyde kullanabilmeniz için, çok eşgörünümlü bir kuyruk yöneticisiyle birleştirebilirsiniz.

Genel ikizleme

Global ikizleme, zamanuyumsuz ikizleme sağlayan ve daha uzun mesafeler üzerinde yedekleme ve olağanüstü durumdan kurtarma için uygun olan, ancak yüksek kullanılabilirlik için normal bir seçim olmadığı için, para birimi yerine yalnızca zaman tutarlılığı sağladığı için, aygıt düzeyinde bir hizmettir.

Dikkate almanız gereken önemli karar noktaları şunlardır:

ASP mi, yoksa bağımsız ASP mi?

Çok eşgörünümlü kuyruk yöneticilerini kullanmak için bir IBM i HA kümesi çalıştırmamız gerekmez. Bağımsız ASP 'leri ya da bağımsız ASP' leri kullanıyorsanız ya da bağımsız ASP 'leri gerektiren diğer uygulamalar için uygunluk gereksiniminiz varsa, bağımsız ASP' leri seçebilirsiniz. Kuyruk yöneticisi izleme programının kuyruk yöneticisi hatasını algılamak için, çok eşgörünümlü kuyruk yöneticileriyle bağımsız ASP ' leri birleştirmeye değer bir yöntem olabilir.

Kullanılabilirlik mi?

Kurtarma süresi hedefi (RTO) nedir? Yakın kesintisiz bir davranışa gerek duyuyorsanız, hangi çözümün en hızlı kurtarma süreye sahip olduğu ortaya çıktı?

Günlük kullanılabilirliği?

Tek bir hata noktası olarak, günlüğü nasıl ortadan kaldırılıyorsa RAID 1 aygıtlarını ya da daha iyisini kullanarak bir donanım çözümü kullanılabilir ya da eşleme günlüklerini ya da disk ikizleme özelliğini kullanarak bir yazılım çözümü kullanabilirsiniz.

Mesafe?

Etkin ve beklemedeki kuyruk yöneticisi eşgörünümleri birbirinden ne kadar uzakta olabilir. Kullanıcılarınız, en çok 250 metre yükseklikten, eşzamanlı olarak çoğaltmanın performans düşüşlerine göz yumabilir mi?

Beceriler?

Çözümün düzenli olarak sürdürülmesi ve uygulanabilmesinde rol alan idari görevleri otomatikleştirmek için yapılacak çalışmalar vardır. Otomasyonu yapmak için gereken beceriler, ASP 'lere ve bağımsız ASP' lere dayalı çözümler için farklıdır.

IBM i **IBM üzerinde çok eşgörünümlü bir kuyruk yöneticisinin silinmesi**

Çok eşgörünümlü bir kuyruk yöneticisini silmeden önce, uzak günlüğe kaydetme işlemini durdurun ve kuyruk yöneticisi yönetim ortamlarını kaldırın.

Başlamadan önce

1. Bu örnekte, QM1 kuyruk yöneticisinin iki eşgörünümlü ALPHA ve BETA sunucularında tanımlanmıştır. Etkin yönetim ortamı, BETA yönetim ortamı yedek veritabanıdır. The queue manager data associated with the queue manager QM1 is stored on the IBM i server GAMMA, using NetServer. Bkz. "[IBM üzerinde günlük ikizleme ve NetServer kullanan çok eşgörünümlü bir kuyruk yöneticisi yaratılması](#)" sayfa 382.
2. Tanımlı olan uzak günlüklerin IBM MQ tarafından silinebilmesi için ALPHA ve BETA 'nın bağlı olması gerekir.
3. Verify that the /QNTC directory and server directory file share can be accessed, using the system commands **EDTF** or **WRKLNK**

Bu görev hakkında

Before you delete a multi-instance queue manager from a server using the **DLTMQM** command, remove any queue manager instances on other servers using the **RMVMQMINF** command.

RMVMQMINF komutunu kullanarak bir kuyruk yöneticisi yönetim ortamını kaldırdığınızda, önekli AMQ olan yerel ve uzak günlükler ve yönetim ortamıyla ilişkilendirildiğinde, silinir. Kuyruk yöneticisi yönetim ortamına ilişkin yapılandırma bilgileri (yerel sunucu için yerel olarak) da silinir.

Kuyruk yöneticisinin kalan yönetim ortamını tutan sunucuda **RMVMQMINF** komutunu çalıştırmayın. Bunu yapmak, **DLTMQM** 'in doğru şekilde çalıştığından emin olun.

Delete the queue manager using the **DLTMQM** command. Kuyruk yöneticisi verileri ağ paylaşımından kaldırılır. Local and remote journals prefixed with AMQ and associated with the instance are deleted. **DLTMQM** ayrıca, kuyruk yöneticisi örneğine ilişkin yapılandırma bilgilerini sunucuya yerel olarak siler.

Örnekte, yalnızca iki kuyruk yöneticisi örneği vardır. IBM MQ , bir etkin kuyruk yöneticisi yönetim ortamı ve bir yedek yönetim ortamı olan çalışmakta olan çok eşgörünümlü bir yapılandırma destekler. Çalışmakta olan yapılandırlarda kullanılacak ek kuyruk yöneticisi yönetim ortamları yarattıysa, geri kalan yönetim ortamını silmeden önce **RMVMQMINF** komutunu kullanarak bunları kaldırın.

Yordam

1. Kuyruk yöneticisi eşgörünümleri arasında etkin olmayan uzak günlük kaydı yapmak için, her sunucuda **CHGMQMJRN RMTJRNSTS** (*INACTIVE) komutunu çalıştırın.
 - a) ALFA ' DA:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('BETA') RMTJRNSTS(*INACTIVE)
```

b) -Evet.

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('ALPHA') RMTJRNSTS(*INACTIVE)
```

2. Her iki QM1eşgörünümünü 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 **RMVMQMINF** komutunu ALPHA komutunu çalıştırın.

```
RMVMQMINF MQMNAME(QM1)
```

RMVMQMINF , QM1 için kuyruk yöneticisi yapılanış bilgilerini ALPHA ' dan kaldırır. Günlük adının öneki AMQise, QM1 ile ilişkilendirilmiş yerel günlüğü ALPHA ' dan siler. Günlük adı önekli AMQ ise ve uzak bir günlük yaratıldıysa, uzak günlüğü BETA ' dan da kaldırır.

4. QM1'u silmek için BETA' da **DLTMQM** komutunu çalıştırın.

```
DLTMQM MQMNAME(QM1)
```

DLTMQM , GAMMA üzerindeki ağ paylaşımından kuyruk yöneticisi verilerini siler. QM1 'a ilişkin kuyruk yöneticisi yapılanış bilgilerini BETA' dan kaldırır. If the journal name is prefixed by AMQ, it deletes the local journal associated with QM1 from BETA. Günlük adı önekli AMQ ise ve uzak bir günlük yaratıldıysa, uzak günlüğü ALPHA ' dan da kaldırır.

Sonuçlar

DLTMQM ve **RMVMQMINF** , **CRTMQM** ve **ADDMQJRN** tarafından oluşturulan yerel ve uzak günlükleri siler. Komutlar, günlük nesnelerini de siler. The journals and journal receivers must follow the naming convention of having names starting with AMQ. **DLTMQM** and **RMVMQMINF** remove the queue manager objects, queue manager data, and the queue manager configuration information from mq.s.ini.

Sonraki adım

Alternatif bir yaklaşım, [“1” sayfa 397](#) adımında günlüğe kaydetme işlemi devre dışı bırakıldıktan sonra ve kuyruk yöneticisi yönetim ortamlarını sona erdirmeden önce aşağıdaki komutları yayınlamaya devam eder. Ya da, adlandırma kuralını izlemediyseniz, günlükleri ve günlük nesnelerini adlarına göre silmeniz gerekir.

1. ALFA ' DA:

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('BETA')
```

2. -Evet.

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('ALPHA')
```

Günlükleri sildikten sonra, diğer adımları da devam ettirin.

IBM üzerinde çok eşgörümlü bir kuyruk yöneticisinin yedeklenmesi

Bu yordam, yerel sunucudaki kuyruk yöneticisi nesnelere ve ağ dosya sunucusundaki kuyruk yöneticisi verilerini nasıl yedeklemenizi gösterir. Örneğin, diğer kuyruk yöneticilerine ilişkin verileri yedeklemek için uyarlayın.

Başlamadan önce

In this example, the queue manager data associated with the queue manager QM1 is stored on the IBM i server called GAMMA, using NetServer. Bkz. "IBM üzerinde günlük ikizleme ve NetServer kullanan çok eşgörümlü bir kuyruk yöneticisi yaratılması" sayfa 382. IBM MQ , ALPHA ve BETA sunucularında kurulu. Kuyruk yöneticisi (QM1), ALPHA ve BETA ' da yapılandırıldı.

Bu görev hakkında

IBM i , uzak bir dizinden veri saklamayı desteklemez. Uzak dosya sistemindeki kuyruk yöneticisi verilerini, yerel dosya sistemi sunucusu için yerel yedekleme yordamlarını kullanarak saklayın. Bu görevde, ağ dosya sistemi bir IBM i sunucusunda, GAMMA ' da yer alıyor. Kuyruk yöneticisi verileri, GAMMA üzerindeki bir saklama dosyasında yedeklenir.

Ağ dosya sistemi Windows ya da Linux işletim sisteminde ise, kuyruk yöneticisi verilerini sıkıştırılmış bir dosyada saklayabilir ve daha sonra saklayabilirsiniz. If you have a back-up system, such as Tivoli Storage Manager, use it to back up the queue manager data.

Yordam

1. QM1 ile ilişkili kuyruk yöneticisi kitaplığı için ALPHA üzerinde bir saklama dosyası yaratın.

Saklama dosyasını adlamak için kuyruk yöneticisi kitaplığı adını kullanın.

```
CRTSAVF FILE(QGPL/QMQM1)
```

2. Kuyruk yöneticisi kitaplığını ALPHA ' daki saklama dosyasında saklayın.

```
SAVLIB LIB(QMQM1) DEV(*SAVF) SAVF(QGPL/QMQM1)
```

3. GAMA ' da kuyruk yöneticisi verileri dizini için bir saklama dosyası yaratın.

Saklama dosyasını adlamak için kuyruk yöneticisi adını kullanın.

```
CRTSAVF FILE(QGPL/QMDQM1)
```

4. Kuyruk yöneticisi verilerinin kopyasını GAMA ' daki yerel dizinden kaydedin.

```
SAV DEV('/QSYS.LIB/QGPL.LIB/QMDQM1.FILE') OBJ('/QIBM/Userdata/mqm/qmgrs/QM1')
```

Çok eşgörümlü kuyruk yöneticilerini ayarlamak için komutlar

IBM MQ , günlük kopyalamayı yapılandırmanın basitleştirilmesi, yeni kuyruk yöneticisi eşgörümleri eklenmesi ve kuyruk yöneticilerinin bağımsız ASP ' yi kullanacak şekilde yapılandırılmasına ilişkin komutları içerir.

Yerel ve uzak günlükler yaratmak ve yönetmek için kullanılan günlük komutları şunlardır:

ADDMQMJRN

Bu komutla, bir kuyruk yöneticisi yönetim ortamı için yerel ve uzak günlükler yaratabilir ve eşlemenin zaman uyumlu mu, yoksa zaman uyumsuz mu olacağını, zaman uyumlu zamanaşımını mı, yoksa uzak günlüğün hemen etkinleştirilip etkinleştirilmeyeceğini yapılandırabilirsiniz.

CHGMQMJRN

Komut, eşleme günlüklerini etkileyen zamanaşımı, durum ve teslim parametrelerini değiştirir.

RMVMQMJRN

Bir kuyruk yöneticisi eşgörünümünden adlandırılmış *uzak* günlüklerini kaldırır.

WRKMQMJRN

Yerel kuyruk yöneticisi yönetim ortamı için yerel ve uzak günlüklerin durumunu listeler.

mqs.ini dosyasını değiştiren aşağıdaki komutları kullanarak ek kuyruk yöneticisi yönetim ortamları ekleyin ve bunları yönetin.

ADDMQMINF

Komut, farklı bir IBM i sunucusuna yeni bir kuyruk yöneticisi örneği eklemek için, DSPMQMINF komutuyla mqs.ini dosyasından açtığınız bilgileri kullanır.

RMVMQMINF

Bir kuyruk yöneticisi yönetim ortamını kaldırın. Bu komutu, varolan bir kuyruk yöneticisinin eşgörünümünü kaldırmak ya da farklı bir sunucudan silinmiş bir kuyruk yöneticisine ilişkin yapılan iş bilgilerini kaldırmak için kullanın.

Çok eşgörünümlü bir kuyruk yöneticisinin yapılandırılmasına yardımcı olmak için **CRTMQM** komutunun üç değiştirgesi vardır.

MQMDIRP (*DFT | *dizin-önek*)

Ağ üzerinde çalışan depolama üzerinde kuyruk yöneticisi verilerine eşlenmiş bir bağlama noktası seçmek için bu parametreyi kullanın.

ASP (*SYSTEM|*ASPDEV| *yardımcı-depolama-havuz-no*)

Kuyruk yöneticisi günlüğünü sisteme ya da temel bir kullanıcı ASP ' ye yerleştirmek için *SYSTEM ya da *yardımcı-saklama-havuzu-no* değerlerini belirtin. Kuyruk yöneticisi günlüğünü bağımsız bir ASP ' ye yerleştirmek için, *ASPDEV seçeneğini belirleyin ve **ASPDEV** parametresini kullanarak bir aygıt adı da ayarlayın.

ASPDEV (*ASP|*aygıt-adi*)

Birincil ya da ikincil bağımsız ASP aygıtı için bir *aygıt adı* belirleyin. *ASP seçeneğinin belirlenmesi, **ASP** (*SYSTEM) değerinin belirlenmesine neden olur.

IBM i

IBM üzerinde performans ve disk kaybına uyum sorunları

Performansı ve güvenilirliği artırmak için farklı yardımcı depolama havuzları kullanın.

Uygulamalarınızda çok sayıda kalıcı ileti ya da büyük ileti kullanıyorsanız, bu iletinin diske yazılması, sistemin başarımında önemli bir faktör haline gelir.

Bu olanakla baş etmek için yeterli disk etkinleştirmenizin olduğundan emin olun ya da kuyruk yöneticisi günlük alıcılarınızın tutulacağı ayrı bir Yardımcı Bellek Havuzu (ASP) düşünün.

You can specify which ASP your queue manager library and journals are stored on when you create your queue manager using the ASP parameter of **CRTMQM**. Varsayılan olarak, kuyruk yöneticisi kitaplığı ve günlükler ve IFS verileri sistem ASP ' inde saklanır.

ASP ' ler, belirli bir ya da daha çok disk birimindeki nesnelerin yalıtılmasına olanak tanır. Bu, bir disk ortamı hatası nedeniyle veri kaybını da azaltabilir. Çoğu durumda, yalnızca etkilenen ASP ' deki disk birimlerinde saklanan veriler kaybedilir.

Ayrı kullanıcı ASP ' lerindeki kuyruk yöneticisi kitaplığını ve günlük verilerini, hata durumunda yedek sisteme geçiş sağlamak ve disk çekişmesini azaltmak için kök IFS dosya sistemindeki ayrı ayrı kullanıcı ASP' lerinde saklamanız önerilir.

Ek bilgi için, IBM i belgelerinde [Backup and recovery](#) başlıklı konuya bakın.

IBM i

IBM i kitaplığındaki IBM MQ kitaplıklarını kaydetmek için SAVLIB ' in kullanılması

IBM MQ kitaplıklarını saklamak için SAVLIB LIB(*ALLUSR) kullanamazsınız; bu kitaplıkların adları Q ile başlayan adlara sahiptir.

Tüm kuyruk yöneticisi kitaplıklarını saklamak için SAVLIB LIB(QM*) komutunu kullanabilirsiniz; ancak, *SAVF dışında bir saklama aygıtı kullanıyorsanız, bu kitaplıkları kullanabilirsiniz. DEV (*SAVF) için, sisteminizdeki her kuyruk yöneticisi kitaplığı için bir SAVLIB komutu kullanmanız gerekir.

IBM i IBM MQ for IBM isusturulması

Bu bölümde, quiesce' un nasıl susturacağını açıklar. IBM MQ for IBM i.

To quiesce IBM MQ for IBM i:

1. Herhangi bir nesneye erişemediğinizi doğrulayan yeni bir etkileşimli IBM MQ for IBM i oturumunda oturum açın.
2. Aşağıdakileri sağladığınızdan emin olun:
 - *ALLOBJ yetkisi ya da QMQM kitaplığı için nesne yönetimi yetkisi
 - ENDSBS komutunu kullanmak için yeterli yetki
3. Tüm kullanıcılara IBM MQ for IBM i' u durduracağını bildirin.
4. Bundan sonra nasıl devam ederseniz, tek bir kuyruk yöneticisini (diğerlerinin var olabileceği) kapatmak (susturma) isteyip istememenize bağlıdır (bkz. [“Shutting down a single queue manager for IBM MQ for IBM i” sayfa 401](#)). ya da tüm kuyruk yöneticilerine bakın (bkz. [“IBM MQ for IBM için tüm kuyruk yöneticilerini kapatma” sayfa 403](#)).
5. qshell içinde aşağıdaki 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 ' ta ve önceki sürümlerle karşılaştırıldığında farklı şekilde çalışır.

Önceki sürümlerde, ENDCCTJOB (*YES) belirttiğinizde, MQ uygulamanızı sizin için zorla sonlandırmanızı sağlar.

IBM MQ for IBM i V6.0 ya da sonraki yayın düzeylerinde, ENDCCTJOB (*YES) belirttiğinizde, uygulamalarınız sonlandırılmaz, ancak kuyruk yöneticisinden bağlantınız kesilir.

ENDCCTJOB (*YES) değerini belirlerseniz ve bir kuyruk yöneticisinin sona erdirileceğini algılamak için yazılmamış uygulamalar varsa, yeni bir MQI çağrısı yayınlanırsa, çağrı bir MQRC_CONNECTION_BROKEN (2009) hatasıyla geri döner.

ENDCCTJOB (*YES) kullanmanın bir alternatifi olarak, ENDCCTJOB (*NO) parametresini kullanın ve WRKMQM seçenek 22 (İşlerle çalış) komutunu kullanarak, kuyruk yöneticisinin yeniden başlatılmasını önleyecek tüm uygulama işlerini el ile sona erdirin.

IBM i Shutting down a single queue manager for IBM MQ for IBM i

Üç tip kapanma tipini anlamak için bu bilgileri kullanın.

In the procedures that follow, we use a sample queue manager name of QMgr1 and a sample subsystem name of SUBX. Gerekiyorsa, bu adları kendi değerlerinizle değiştirin.

Planlı kapanma

IBM üzerinde bir kuyruk yöneticisinin sona erdirilmesinin planlandığı

1. Kapatmadan önce şunu yürütün:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNTA(*YES)
```

2. Kuyruk yöneticisini sona erdirmek için aşağıdaki komutu yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

QMgr1 sona ermezse, kanal ya da uygulamalar büyük olasılıkla meşgul olur.

3. QMgr1 ' u hemen kapatmanız gerekiyorsa, aşağıdakileri yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

Planlanmamış kapatma

1. Kuyruk yöneticisini sona erdirmek için aşağıdaki komutu yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

QMgr1 sona ermezse, kanal ya da uygulamalar büyük olasılıkla meşgul olur.

2. QMgr1 ' i hemen kapatmanız gerekiyorsa, aşağıdakileri yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

Olağan dışı koşullar altında sona erdirin

1. Kuyruk yöneticisini sona erdirmek için aşağıdaki komutu yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

QMgr1 sona ermezse, 3. adımla devam edin:

- QMgr1 kendi altsisteminde mi, yoksa
 - Aynı altsistemi QMgr1 ile paylaşan tüm kuyruk yöneticilerini sona erdirebilirsiniz. Bu tür tüm kuyruk yöneticileri için planlanmamış kapatma yordamını kullanın.
2. Altsistemi paylaşan tüm kuyruk yöneticilerine ilişkin yordamda tüm adımları attığınızda (örneklerimizde SUBX), aşağıdaki işlemleri yürütün:

```
ENDSBS SUBX *IMMED
```

Bu komut tamamlanamazsa, planlanmamış kapatma yordamını kullanarak tüm kuyruk yöneticilerini sona erdirin ve makineniz üzerinde IPL işlemi gerçekleştirin.

Uyarı: Makinenizde hemen sonra bir IPL işlemi gerçekleştirmeye hazır değilseniz, ENDJOB ya da ENDSBS'in sonucu olarak bitmeyen IBM MQ işleri için ENDJOBABN ' yi kullanmayın.

3. Altsistemi başlatmak için aşağıdaki komutu yürüterek:

```
STRSBS SUBX
```

4. Aşağıdaki komutu yürüterek, kuyruk yöneticisini hemen sona erdirin:

```
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:

- IPL işlemini gerçekleştirerek makinenizi yeniden başlatmış bulunmanızı ya da
- Yalnızca tek bir kuyruk yöneticisi var

Tidy up IBM MQ shared memory by executing:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

5. adımdan önce.

If the queue manager restart takes more than a few seconds, IBM MQ adds status messages intermittently to the job log detailing the startup progress.

Kuyruk yöneticinizi yeniden başlattığınız sorunlar devam ederse, IBM destek birimine başvurun. Any further action you might take could damage the queue manager, leaving IBM MQ unable to recover.

IBM i

IBM MQ for IBM için tüm kuyruk yöneticilerini kapatma

Üç tip kapanma tipini anlamak için bu bilgileri kullanın.

Yordamlar, tek bir kuyruk yöneticisi için neredeyse aynıdır, ancak mümkün olan yerlerde kuyruk yöneticisi adı yerine *ALL ' yi kullanarak ve tersi durumda, her kuyruk yöneticisi adını sırayla kullanarak bir komutu yeniden kullanarak kullanabilirsiniz. Throughout the procedures, we use a sample queue manager name of QMgr1 and a sample subsystem name of SUBX. Bunları kendi ellerinizle değiştirin.

Planlı kapanma

1. Kapanmadan bir saat önce:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNTA(*YES)
```

Sona erdirmek istediğiniz her kuyruk yöneticisi için bunu yineleyin.

2. Kuyruk yöneticisini sona erdirmek için aşağıdaki komutu yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

Sona erdirmek istediğiniz her kuyruk yöneticisi için bunu yineleyin; ayrı komutlar koşturularak çalışabilir.

Herhangi bir kuyruk yöneticisi makul bir süre içinde sona ermezse (örneğin, 10 dakika), 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 sona erdirmek için aşağıdaki komutu yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Sona erdirmek istediğiniz her kuyruk yöneticisi için bunu yineleyin; ayrı komutlar koşturularak çalışabilir.

Kuyruk yöneticileri sona ermezse, kanal ya da uygulamalar büyük olasılıkla meşgul olur.

2. Kuyruk yöneticilerini hemen kapatmanız gerekiyorsa, aşağıdakileri yürütün:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

Olağan dışı koşullar altında sona erdirin

1. Kuyruk yöneticilerini sona erdirmek için aşağıdaki komutu yürütün:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Sona erdirmek istediğiniz her kuyruk yöneticisi için bunu yineleyin; ayrı komutlar koşturularak çalışabilir.

2. Aşağıdaki altsistemleri (örneğinizde SUBX) sona erdirin:

```
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 işlemi gerçekleştirmeye hazır değilseniz, ENDJOB ya da ENDSBS' in sonucu olarak sona erdirilmeyen işler için ENDJOBABN kullanmayın.

3. Altsistemleri yürüterek başlat:

```
STRSBS SUBX
```

Başlatmak istediğiniz her altsistem için bunu yineleyin.

4. Aşağıdaki komutu yürüterek, kuyruk yöneticilerini hemen aşağı kapatın:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

5. Şu komutu yürüterek kuyruk yöneticilerini yeniden başlatın:

```
STRMQM MQMNAME(QMgr1)
```

Başlatmak istediğiniz her kuyruk yöneticisi için bunu yineleyin.

If any queue manager restart takes more than a few seconds IBM MQ will show status messages intermittently detailing the startup progress.

Herhangi bir kuyruk yöneticisini yeniden başlatarak sorun devam ederse, IBM destek birimine başvurun. Any further action you might take could damage the queue managers, leaving MQSeries or IBM MQ unable to recover.

z/OS

YönetmeIBM MQ for z/OS

Kuyruk yöneticilerinin ve ilişkili kaynakların yönetilmesi, bu kaynakları etkinleştirmek ve yönetmek için sık sık gerçekleştirdiğiniz görevleri içerir. Kuyruk yöneticilerinizi ve ilişkili kaynakları yönetmeyi tercih ettiğiniz yöntemi seçin.

IBM MQ for z/OS , ürünle birlikte sağlanan bir dizi yardımcı program ve program tarafından denetlenebilir ve yönetilebilir. IBM MQ for z/OSkomutunu yönetmek için IBM MQ Script (MQSC) komutlarını ya da Programları Komut Biçimleri 'ni (PCF ' ler) kullanabilirsiniz. IBM MQ for z/OSkomutlarının kullanılmasına ilişkin bilgi almak için bkz. "[IBM MQ for z/OSkomutlarına komut verilmesi](#)" sayfa 405.

IBM MQ for z/OS , sistem denetimi konusunda size yardımcı olacak bir dizi yardımcı program da sağlar. Farklı yardımcı programlar programları ve bunların nasıl kullanılacağı hakkında bilgi için bkz. "[IBM MQ for z/OS yardımcı programları](#)" sayfa 413.

IBM MQ for z/OS ' in nasıl yönetileceği ve üstlenmeniz gereken farklı yönetim görevlerine ilişkin ayrıntılar için aşağıdaki bağlantılara bakın:

İlgili kavramlar

[IBM MQ for z/OS kavramlar](#)

İlgili görevler

[“Yerel IBM MQ nesneleriyle çalışma” sayfa 119](#)

Yerel IBM MQ nesnelerini, Message Queue Interface (MQI) olanağını kullanan uygulama programlarını desteklemek için denetleyebilirsiniz.

[“Uzak IBM MQ nesneleriyle çalışma” sayfa 180](#)

Uzak kuyruk yöneticilerindeki IBM MQ nesnelerini, MQSC komutlarını, PCF komutlarını ya da administrative REST APIkomutunu kullanarak yönetebilirsiniz. Bu yöntemlerin herhangi birini kullanmadan önce, uzak kuyruk yöneticisine komutların gönderilebilmesi ve yerel kuyruk yöneticisi tarafından alınan yanıtların gönderilebilmesi için yerel kuyruk yöneticisi ile uzak kuyruk yöneticisi arasındaki iletim kuyruklarını ve kanalları tanımlamanız gerekir. Diğer bir seçenek olarak, bir kuyruk yöneticisi kümesi yapılandırılabilir ve daha sonra aynı uzak denetim yöntemlerini kullanabilirsiniz.

[“YönetmeIBM MQ” sayfa 7](#)

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

[Planlama](#)

[z/OSüzerinde IBM MQ ortamınızın planlanması](#)

[Yapılandırılıyor](#)

[yapılandırmaz/OS](#)

[IBM MQ for z/OS yardımcı programlarının kullanılması](#)

İlgili başvurular

[“z/OSüzerinde MQSC komutları yayınlatabileceğiniz kaynaklar” sayfa 20](#)

MQSC komutları, komutta bağlı olarak çeşitli kaynaklardan yayınlanabilir.

[Programlanabilir komut biçimleri başvurusu](#)

z/OS

IBM MQ for z/OSkomutlarına komut verilmesi

Bir kuyruk yöneticisini denetlemek için toplu kipte ya da etkileşimli kipte IBM MQ komut dosyası komutlarını (MQSC) kullanabilirsiniz.

IBM MQ for z/OS , aşağıdaki kaynaklardan yayınlanabilen MQSC komutlarını destekler:

- z/OS konsolu ya da eşdeğeri (SDSF/TSO gibi).

When using the z/OS console, you need to add /cpf to the start of a command, where cpf is the command prefix for the queue manager subsystem.

- Kullanıma hazırlama giriş veri kümeleri.
- Sıralı bir veri kümesindeki komutların bir listesini işlemek için sağlanan toplu iş yardımcı programı CSQUTIL.

Kullanıma hazırlama giriş veri kümelerini ya da sağlanan toplu iş yardımcı programını kullanırken, komut başlangıcına /cpf eklemeniz gerekmez.

- Komut giriş kuyruğuna ileti olarak bir komut göndererek uygun bir şekilde yetkilendirilmiş bir uygulama. Uygulama aşağıdakilerden biri olabilir:
 - Bir toplu bölge programı
 - Bir CICS uygulaması
 - Bir IMS uygulaması
 - TSO uygulaması
 - Başka bir IBM MQ sisteminde bir uygulama programı ya da yardımcı programı

[Çizelge 26 sayfa 409](#) , MQSC komutlarını ve bunların yayınlanabilecekleri kaynakları özetler.

Bu komutların işlevlerinin çoğu IBM MQ for z/OS işlem ve denetim panolarından uygun bir şekilde kullanılabilir.

Bir kuyruk yöneticisinin kaynak tanımlamalarında, komutları (doğrudan ya da dolaylı olarak) kullanarak yapılan değişiklikler, IBM MQ altsisteminin yeniden başlatılması için korunur.

IBM MQ for z/OS ayrıca, Programmable Command Format (PCF) komutlarını da destekler. Bunlar, IBM MQ denetimi için uygulama yaratılmasını kolaylaştırır. MQSC komutları, insan tarafından okunabilen metin formunda, PCF ise metin dizelerini ayrıştırmak zorunda kalmadan, uygulamaların istekleri oluşturmasına ve yanıtları okumasına olanak sağlar. MQSC komutları gibi, uygulamalar komut giriş kuyruğuna ileti olarak gönderilerek PCF komutları yayınlanır. PCF komutlarının kullanılmasıyla ilgili ek bilgi ve komutların ayrıntıları için [Programlanabilir komut biçimleri başvurusu](#) belgelerine bakın.

z/OS IBM MQ for z/OS üzerindeki özel ve genel tanımlar

IBM MQ for z/OS üzerinde bir nesne tanımladığınızda, bu tanımlı diğer kuyruk yöneticileriyle (bir *genel* tanım) paylaşmak isteyip istemediğinizi ya da nesne tanımlamasının yalnızca bir kuyruk yöneticisi tarafından mı kullanılacağını (*özel* bir tanımlama) seçebilirsiniz. Buna nesne *yok etme* adı verilir.

Genel tanım

Kuyruk yöneticiniz bir kuyruk paylaşım grubuna aitse, yaptığınız tüm nesne tanımlamalarını grubun diğer üyeleriyle paylaşmayı seçebilirsiniz. Başka bir deyişle, bir nesneyi yalnızca bir kez tanımlamanız ve tüm sistem için gerekli olan tanımların toplam sayısını azaltmanız gerekir.

Genel nesne tanımlamaları bir *paylaşılan havuzda* (Db2 paylaşılan veritabanında) tutulur ve kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine kullanılabilir. Bu nesnelerin bir GROUP (GROUP) yok etme düzeni vardır.

Özel tanım

Yalnızca bir kuyruk yöneticisi için gerekli olan bir nesne tanımlaması yaratmak ya da kuyruk yöneticinizin kuyruk paylaşım grubunun bir üyesi değilse, bir kuyruk paylaşım grubunun diğer üyeleriyle paylaşılmayan nesne tanımları yaratabilirsiniz.

Özel nesne tanımlamaları, tanımlayan kuyruk yöneticisinin sıfır olan sayfa kümesi için tutulur. Bu nesnelerin QMGR yok etme durumu vardır.

CF yapıları (yani, kanallar, ad listeleri, süreç tanımlamaları, kuyruklar, kuyruk yöneticileri, depolama sınıfı tanımları ve kimlik doğrulama bilgileri nesnelere) dışında tüm IBM MQ nesnelere için özel tanımlamalar ve kuyruk yöneticileri dışında tüm nesne tipleri için genel tanımlar yaratabilirsiniz.

IBM MQ , bir grup nesnesinin tanımını, onu kullanan her bir kuyruk yöneticisinin sıfır sıfır değerine otomatik olarak kopyalar. Tanımın kopyasını geçici olarak isterseniz değiştirebilir ve IBM MQ , gerekirse havuz kopyasından sayfa kümesi kopyalarını yenilemişmenizi sağlar.

IBM MQ her zaman, başlatma sırasında havuz kopyasından sayfa kümesi kopyalarını yenilemeyi dener (kanal komutları için, bu işlem kanal başlatıcı yeniden başlatıldığında yapılır) ya da grup nesnesi değiştirilirse.

Not: Tanımın kopyası grubun tanımlamasından yenilenir; ancak, tanımın kopyasını yarattıktan sonra grubun tanımı değiştiyse.

Bu, sayfa kümesi kopyalarının, kuyruk yöneticisi etkin değilken yapılan değişiklikler de dahil olmak üzere havuzdaki sürümü yansımaları sağlar. Kopyalar DEFINE REPLACE komutları üretilerek yenilenir; bu nedenle, yenileme işleminin gerçekleştirilmediği durumlar vardır; örneğin:

- Kuyruğun bir kopyası açıksa, kuyruğun kullanımını değiştiren bir yenileme işlemi başarısız olur.
- Kuyruğun bir kopyası üzerinde ileti varsa, o kuyruğun silinmesine neden olan yenileme işlemi başarısız olur.
- Bir kuyruğun kopyasının değiştirilmesi için ALTER ile birlikte ALTER (ALTER) işlemi gereklidir.

Bu durumda, yenileme o kopyada gerçekleştirilmez; ancak, diğer tüm kuyruk yöneticilerindeki kopyalarda gerçekleştirilir.

Kuyruk yöneticisi kapanırsa ve bağımsız olarak yeniden başlatılırsa, nesnelerin herhangi bir yerel kopyası silinir; örneğin, kuyruğun ilişkili iletileri vardır.

Yalnızca yerel kuyruklar için geçerli olan üçüncü bir nesne yok etme nesnesi vardır. Bu, paylaşılan kuyruklar oluşturmanızı sağlar. Paylaşılan bir kuyruk için tanım paylaşılan havuzda tutulur ve kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine kullanılabilir. Ayrıca, paylaşılan bir kuyrukta bulunan iletiler, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine de kullanılabilir. Bu, Paylaşılan kuyruklar ve kuyruk paylaşım grupları içinde açıklanmaktadır. Paylaşılan kuyruklar, SHARED nesnesine ilişkin bir nesneye sahiptir.

Aşağıdaki çizelge, kuyruk yöneticilerine ilişkin nesne yok etme seçeneklerinin etkisini yalnızca bağımsız olarak ve bir kuyruk paylaşım grubunun bir üyesi olarak özetlemektedir.

Süre	Bağımsız kuyruk yöneticisi	Kuyruk paylaşım grubunun üyesi
MMGR	Sayfa kümesi sıfır üzerinde tutulan nesne tanımlaması.	Sayfa kümesi sıfır üzerinde tutulan nesne tanımlaması.
GRUP	izin verilmez.	Paylaşılan havuzda tutulan nesne tanımlaması. Grup içindeki her bir kuyruk yöneticisinin sıfır sayfa kümesinde tutulan yerel kopya.
Paylaşılan	izin verilmez.	Paylaşılan havuzda kuyruk tanımlaması tutuldu. İletilerin, gruptaki herhangi bir kuyruk yöneticisinde kullanılabilir.

Genel tanımların işlenmesi

Paylaşılan havuzda tutulan bir nesnenin tanımını değiştirmek isterseniz, havuzdaki sürümü değiştirmek isteyip istemediğinizi ya da sayfa kümesi sıfır üzerindeki yerel kopyanın mı değiştirileceğini belirtmeniz gerekir. Bu işlemi yapmak için komutun bir parçası olarak nesne yok etme olanağını kullanın.

Directing commands to different queue managers on z/OS

Komutun çalıştırıldığı kuyruk yöneticisini denetlemek için *komut kapsamı* 'ı kullanabilirsiniz.

Bir komutu girildiği kuyruk yöneticisinde ya da kuyruk paylaşım grubundaki farklı bir kuyruk yöneticisinde yürütmeyi seçebilirsiniz. Ayrıca, belirli bir komutu, bir kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine koşturarak yayınlamayı da seçebilirsiniz. Bu, hem MQSC komutları hem de PCF komutları için olanaklıdır.

Bu, *komut kapsamı* tarafından belirlenir. Komut kapsamı, üzerinde çalışmak istediğiniz nesnenin hangi sürümünü saptamak için nesne yok etme ile birlikte kullanılır.

Örneğin, bir nesnenin özniteliklerinden bazılarını, bu tanımın paylaşılan havuzda tutulmasını isteyebilirsiniz.

- Yalnızca bir kuyruk yöneticisinde sürümü değiştirmek ve havuzdaki sürümde değişiklik yapmak ya da diğer kuyruk yöneticileri tarafından kullanılanlarla ilgili değişiklikler yapmak isteyebilirsiniz.
- Gelecekteki kullanıcılar için paylaşılan havuzdaki sürümü değiştirmek isteyebilirsiniz, ancak var olan kopyaları değişmeden bırakır.
- Paylaşılan havuzdaki sürümü değiştirmek isteyebilirsiniz, ancak değişikliklerinizin, nesnenin sayfa kümesi sıfır üzerindeki bir kopyasını tutan kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine hemen yansımaları da isteyebilirsiniz.

Komutun bu kuyruk yöneticisine, başka bir kuyruk yöneticisine ya da tüm kuyruk yöneticilerine yürütülüp yürütülmediğini belirlemek için komut kapsamını kullanın. Yönetmekte olduğunuz nesnenin paylaşılan havuzda mı (bir grup nesnesi) olduğunu ya da sayfa kümesinin sıfır (bir kuyruk yöneticisi nesnesi) üzerinde yerel bir kopya olup olmadığını belirtmek için nesne yok etmeyi kullanın.

Kuyruk paylaşım grubundaki her kuyruk yöneticisi paylaşılan kuyruğu tek bir kuyruk olarak işlediğinden, bir paylaşılan kuyrukla çalışmak için komut kapsamını ve nesne yok etmeyi belirtmeniz gerekmez.

z/OS IBM MQ for z/OS için komut özeti

Bu konuyu ana MQSC ve PCF komutlarının bir başvurusu olarak kullanın.

Çizelge 25 sayfa 408 , IBM MQ nesnelere değiştirme, tanımlama, silme ve görüntüleme için IBM MQ for z/OS üzerinde kullanılabilir olan MQSC ve PCF komutlarını özetler.

MQSC komutu	ALTER	Tanımla	GÖRÜNTÜLE	DELETE
PCF komutu	Değiştir	Oluştur/Kopyala	Sor	Sil
YAZAR BİLGİLERİ	X	X	X	X
CFSTATUS			X	
CFUSTRİ	X	X	X	X
Kanal	X	X	X	X
SOHBET DURUMU			X	
NAMELIST	X	X	X	X
Süreç	X	X	X	X
QALIAS	M	M	M	M
QKüME			M	
QLOCAL	M	M	M	M
MMGR	X		X	
QMODEL	M	M	M	M
QREMOTE	M	M	M	M
kuyruk	P	P	X	P
QSTATUS			X	
STGCLASS	X	X	X	X

Çizelge simgelerinin anahtarı:

- M = Yalnızca MQSC
- Yalnızca p = PCF
- X = her ikisi

Diğer IBM MQ kaynaklarını yönetmenize ve Çizelge 25 sayfa 408' ta özetlenenlere ek olarak diğer işlemleri de gerçekleştirmenize olanak sağlayan birçok başka MQSC ve PCF komutu vardır.

Çizelge 26 sayfa 409 , her MQSC komutunu ve her komutun aşağıdaki komutlardan yayınlanabileceğini gösterir:

- CSQINP1 kullanıma hazırlama giriş verileri kümesi
- CSQINP2 kullanıma hazırlama giriş verileri kümesi
- z/OS konsolu (ya da eşdeğeri)
- SYSTEM.COMMAND.INPUT kuyruğu ve komut sunucusu (uygulamalardan, CCOSKENE 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
ALTER AUTHINFO		X	X	X
ARABELLEĞİ DEĞİŞTİR		X	X	X
ALTER CFSTRUCT		X	X	X
KANALI ALTER		X	X	X
ALTER NAMELIST		X	X	X
ALTER PSID			X	X
ALTER PROCESS		X	X	X
QALIAS ALTER		X	X	X
ALTER QLOCAL		X	X	X
ALTER QMGR		X	X	X
ALTER QMODEL		X	X	X
ALTER QREMOTE		X	X	X
ALTER SECURITY	X	X	X	X
ALTER STGCLASS		X	X	X
ALTER SUB		X	X	X
KONUUYU DEĞİŞTİR		X	X	X
ALTER TRACE	X	X	X	X
Günlüğü	X	X	X	X
BACKUP CFSTRUCT			X	X
QLOCAL ' I TEMİZLE		X	X	X
DEFINE YAZAR		X	X	X
ARABELLEK HAVUZU TANIMLA	X	X		
CFSTRUCT DEFINE		X	X	X
KANAL TANIMLA		X	X	X
OTURUM KAPAT			X	X
AD LISTESINI TANı		X	X	X
Süreç TANIMLA		X	X	X
DEĞERLERİ	X		X	X
QALIAS ' YI		X	X	X
QLOCAL ' I TANIMLA		X	X	X
QMODEL ' I TANIMLA		X	X	X
QREMOTE TANIMLA		X	X	X
STGCLASS TANIMLA		X	X	X

Çizelge 26. MQSC komutlarının çalıştırılacağı kaynaklar (devamı var)

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
ALT			X	X
KONUUYU TANIMLA		X	X	X
YAZAR BİLGİLERİNİ SIL		X	X	X
ARABELLEK HAVUZUNU SIL			X	X
CFSTRUCT SIL		X	X	X
KANAL SILME			X	X
ADı SIL		X	X	X
Süreci Sil		X	X	X
PSID SIL			X	X
QALIAS SIL		X	X	X
QLOCAL SIL		X	X	X
QMODEMI SIL		X	X	X
QREMOTE SIL		X	X	X
STGCLASı SIL		X	X	X
SUB SIL		X	X	X
KONUUYU SIL		X	X	X
ARŞİV GÖRÜNTÜLE	X	X	X	X
AUTHENTICAFO GÖRÜNTÜLE		X	X	X
CFSTATUS GÖRÜNTÜLE			X	X
CFSTRUCT GÖRÜNTÜLE		X	X	X
KANAL GÖRÜNTÜLE		X	X	X
DURUMU GÖRÜNTÜLE			X	X
CLUSQMGR GÖRÜNTÜLE			X	X
CMDSERV GÖRÜNTÜLE	X	X	X	X
GÖRÜNEN EKРАН		X	X	X
ÇİNCE GÖRÜNTÜLE		X	X	X
GRUBU GÖRÜNTÜLE		X	X	X
GÜNCELLE	X	X	X	X
GÖRÜNTÜLEME		X	X	X
İŞLEM SÜRÜ		X	X	X
QALIAS (QALI		X	X	X
QKüME GÖRÜN		X	X	X
QLOCAL ' I GÖRÜNTÜLE		X	X	X

Çizelge 26. MQSC komutlarının çalıştırılacağı kaynaklar (devamı var)

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
QMGR GÖRÜNTÜLE		X	X	X
QMODEL ' I GÖRÜNTÜLE		X	X	X
QREMOTE DISPLAY		X	X	X
QSTATUS GÖRÜNTÜLE		X	X	X
GÖRÜNTÜLE		X	X	X
GÜVENLİK			X	X
STGCLASS GÖRÜNTÜLE		X	X	X
GÖRÜNTÜLE		X	X	X
KONUYU GÖRÜNTÜLE		X	X	X
SİSTEMİ GÖRÜNTÜLE	X	X	X	X
GÖRÜNTÜLE		X	X	X
İZLEME İZLEME	X	X	X	X
KULLANIM KULLANI		X	X	X
QLOCAL ' I TAŞI		X	X	X
PING KANALI			X	X
BSSS ' LERİ KURTAR	X	X	X	X
CFSTRUCT ' U KURTAR			X	X
KÜME YENİLE		X	X	X
QMGR ' YI YENİLE		X	X	X
Güvenliği yenileme		X	X	X
KANALI			X	X
KÜMEYİ Sı		X	X	X
QSTATS ' İ Sı		X	X	X
TPIPE ' YI			X	X
KANALIN			X	X
BELIRSİZ KALICI		X	X	X
QMGR ' YI Sı			X	X
GÜVENLİĞİ		X	X	X
ARŞİV	X	X	X	X
OTURUM AÇMA	X	X	X	X
SİSTEM AYARLA	X	X	X	X
KANAL BAŞLAT			X	X
CHINIT BAŞLAT		X	X	X

Çizelge 26. MQSC komutlarının çalıştırılacağı kaynaklar (devamı var)

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
CMDSERV BAŞLAT	X	X	X	
DINLEYICI BAŞLAT			X	X
QMGR ' YI			X	
İZLEMEYI	X	X	X	X
KANAL DURDUR			X	X
CHINIT DURDURUN			X	X
CMDSERV ' I DURDUR	X	X	X	
DINLEYICIYI DURDUR			X	X
DURDUR QMGR			X	X
İZLEME DURDUR	X	X	X	X
QMGR ' YI AS			X	X

MQSC komutları içinde, her komut tanımı, komutun çalıştırılabileceği kaynakları tanımlar.

z/OS IBM MQ for z/OS için başlatma komutları

Kuyruk yöneticisi başlatma işlemi denetlemek için kullanıma hazırlama komutları kullanılabilir.

Commands in the initialization input data sets are processed when IBM MQ is initialized on queue manager startup. Kullanıma hazırlama giriş veri kümelerinden üç tip komut yayınlanabilir:

- Başka bir yerde tanımlanamayacak IBM MQ varlıkları tanımlamaya ilişkin komutlar; örneğin, DEFINE BUFFPOOL.

These commands must reside in the data set identified by the DD name CSQINP1. Bunlar, kullanıma hazırlama aşamasından önce işlenirler. Bunlar, konsol, işlemler ve denetim panoları ya da bir uygulama programı aracılığıyla yayınlanamaz. Bu komutlara verilen yanıtlar, başlatılan görev yordamında CSQOUT1 deyiminde başvuruda bulunan sıralı veri kümesine yazılır.

- Yeniden başlatmadan sonra kurtarılabilir IBM MQ nesnelere tanımlamak için kullanılan komutlar. These definitions must be specified in the data set identified by the DD name CSQINP2. Bunlar sayfa kümesi sıfır olarak depolanır. CSQINP2 , kullanıma hazırlama aşaması yeniden başlatma aşamasından sonra işlenir. Bu komutlara verilen yanıtlar, başlatılan görev yordamında CSQOUT2 deyiminde başvuruda bulunan sıralı veri kümesine yazılır.
- IBM MQ nesnelere işlemek için kullanılan komutlar. These commands must also be specified in the data set identified by the DD name CSQINP2. Örneğin, IBM MQ tarafından sağlanan bir örnek, altsistem için bir çıkış kuyruğu kuyruğu belirlemek üzere bir ALTER QMGR komutu içerir. Bu komutlara verilen yanıt, CSQOUT2 çıkış veri kümesine yazılır.

Not: If IBM MQ objects are defined in CSQINP2, IBM MQ attempts to redefine them each time the queue manager is started. Nesnelere önceden varsa, bunları tanımlama girişimi başarısız olur. Nesnenizi CSQINP2' de tanımlamanız gerekiyorsa, DEFE komutlarının REPLACE değiştirgesini kullanarak bu sorunu önleyebilirsiniz; ancak, bu durum kuyruk yöneticisinin önceki çalışması sırasında yapılan değişiklikleri geçersiz kılar.

Örnek kullanıma hazırlama veri kümesi üyeleri IBM MQ for z/OS ile birlikte sağlanır. Bunlar [Sample definitions supplied with IBM MQ](#) içinde anlatılır.

Dağıtılmış kuyruğa alma komutları için kullanıma hazırlama komutları

START CHINIT komutu için CSQINP2 başlatma verileri kümesini de kullanabilirsiniz. Dağıtılmış kuyruğa alma ortamınızı (örneğin, başlangıç dinleyicilerine) tanımlamak için bir dizi diğer komutlara gereksinim duyarsanız, IBM MQ , kanal başlatıcı tarafından başlatılan görev yordamınızın bir parçası olarak işlenen CSQINPX adlı bir üçüncü kullanıma hazırlama giriş veri kümesi sağlar.

Veri kümesinde bulunan MQSC komutları, kanal başlatıcı kullanıma hazırlamanın sonunda yürütülür ve çıkış, CSQOUTX DD deyimiyle belirlenen veri kümesine yazılır. Örneğin, dinleyicilere örnek olarak başlatmak için CSQINPX başlatma verilerini kullanabilirsiniz.

A sample channel initiator initialization data set member is supplied with IBM MQ for z/OS. Bu, [Sample definitions supplied with IBM MQ](#) içinde açıklanmaktadır.

Yayınlama/Abone Olma için kullanıma hazırlama komutları

Yayınlama/abone olma ortamınızı tanımlamak için bir dizi komutlara gereksinim duyarsanız (örneğin, abonelikleri tanımlarken), IBM MQ , CSQINPT olarak adlandırılan dördüncü bir kullanıma hazırlama giriş veri kümesi sağlar.

Veri kümesinde yer alan MQSC komutları yayınlama/abone olma başlatma sonunda yürütülür ve çıkış, CSQOUTT DD bildiriminde belirlenen veri kümesine yazılır. Örneğin, örneğin abonelikleri tanımlamak için CSQINPT başlatma verileri kümesini kullanabilirsiniz.

Örnek bir yayınlama/abone olma kullanıma hazırlama veri kümesi üyesi IBM MQ for z/OS ile birlikte sağlanır. Bu, [Sample definitions supplied with IBM MQ](#) içinde açıklanmaktadır.

z/OS

IBM MQ for z/OS yardımcı programları

IBM MQ for z/OS , sistem denetimi için kullanabileceğiniz bir dizi yardımcı program sağlar.

IBM MQ for z/OS , aşağıdakiler de içinde olmak üzere çeşitli yönetim görevlerini gerçekleştirmenize yardımcı olmak üzere bir dizi yardımcı program sağlar:

- İleti güvenliği ilkelerini yönetmenizi sağlar.
- Yedekleme, geri yükleme ve yeniden düzenleme görevlerini gerçekleştirir.
- Komutları verin ve nesne tanımlamalarını işler.
- Veri dönüştürme çıktıları oluşturur.
- Önyükleme veri kümesini değiştirir.
- Günlüklere ilişkin bilgileri listeler.
- Günlükleri yazdırır.
- Db2 tablolarını ve diğer Db2 yardımcı programlarını ayarlar.
- İleti kuyruğunda ileti işleme.

İleti güvenliği ilkesi yardımcı programı

İleti güvenliği ilkesi yardımcı programı (CSQOUTIL), ileti güvenliği ilkelerini yönetmek için bağımsız bir yardımcı program olarak çalışır. Ek bilgi için [İleti güvenliği ilkesi yardımcı programı \(CSQOUTIL\)](#) konusuna bakın.

CSQUTIL yardımcı programı

Bu, görevleri yedeklemenize, geri yüklemenize ve yeniden düzenlemenize yardımcı olmak için sağlanan bir yardımcı program programıdır. Ek bilgi için [CSQUTIL yardımcı programı](#) başlıklı konuya bakın.

Veri dönüştürme çıkış yardımcı programı

IBM MQ for z/OS veri dönüştürme çıkış yardımcı programı (**CSQUCVX**) veri dönüştürme çıkış yordamları oluşturmak için bağımsız bir yardımcı program olarak çalışır.

Değişiklik günlüğü döküm 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 aşağıdaki işlevleri gerçekleştirmek için kullanabilirsiniz:

- Etkin ya da arşiv günlüğü veri kümelerini ekleyin ya da silin.
- Arşiv günlükleri için parolalar sağlayın.

Yazdırma günlüğü eşlemi yardımcı programı

IBM MQ for z/OS yazdırma günlüğü eşlemi yardımcı programı (**CSQJU004**) Aşağıdaki bilgileri listelemek için bağımsız bir yardımcı program olarak çalışır:

- Tüm etkin ve arşiv günlüğü veri kümelerinin her iki kopyası için veri kümesi adı ve günlük RBA ilişkilendirmesini günlüğe kaydet. İkili günlük kaydı etkin değilse, veri kümelerinin yalnızca bir kopyası vardır.
- Etkin günlük verileri kümeleri, yeni günlük verileri için kullanılabilir.
- Önyükleme veri kümesindeki (BSDS) denetim noktası kayıtları kuyruğunun içeriği.
- Arşiv günlüğü komut geçmişi kaydının içeriği.
- Sistem ve yardımcı program zaman damgaları.

Günlük yazdırma yardımcı programı

Günlük yazdırma yardımcı programı (**CSQ1LOGP**) bağımsız bir yardımcı program olarak çalıştırılır. Aşağıdakileri belirterek yardımcı programı çalıştırabilirsiniz:

- Bir önyükleme veri kümesi (BSDS)
- Etkin günlükler (BSDS olmadan)
- Arşiv günlükleri (BSDS olmadan)

Kuyruk paylaşım grubu yardımcı programı

Kuyruk paylaşım grubu yardımcı programı (**CSQ5PQSG**) Db2 tablolarını ayarlamak ve kuyruk paylaşım grupları için gereken diğer Db2 görevlerini gerçekleştirmek için bağımsız bir yardımcı program olarak çalışır.

Etkin günlük ön biçim yardımcı programı

Etkin günlük önformat yardımcı programı (**CSQJUFMT**) bir kuyruk yöneticisi tarafından kullanılmadan önce etkin günlük veri kümelerini biçimlendirir. Etkin günlük veri kümeleri yardımcı program tarafından önceden biçimlendirildiyse, günlük yazma başarımı, kuyruk yöneticisinin etkin günlüklerden ilk geçişindeki ilk geçişte iyileştirilir.

Ölü-harfli kuyruk işleyicisi yardımcı programı

The dead-letter queue handler yardımcı programı (**CSQUDLQH**) bağımsız bir yardımcı program olarak çalışır. Ölü-mektup kuyruğunda bulunan iletileri denetler ve bu iletileri, yardımcı programa sağladığınız bir kural kümesine göre işler.

qload yardımcı programı

IBM MQ 8.0'tan, IBM MQ Supportpac MO03' ta verilen **qload** yardımcı programı **dmpmqmsg** yardımcı programı olarak IBM MQ ' a tümleştirilmiştir.

z/OSüzerinde, yardımcı program yürütülebilir bir modül olarak (SCSQLOAD kitaplığındaki CSQUDMSG), uyumluluk için QLOAD diğer adıyla kullanılabilir. Örnek JCL, SCSQPROC ' de CSQ4QLOD üyesi olarak da sağlanmıştır.

z/OS CCOSTil yardımcı programı for IBM MQ for z/OS

CSQUTIL yardımcı programı, yedekleme, geri yükleme ve yeniden düzenleme görevlerini gerçekleştirmenize ve komutları ve süreç nesnesi tanımlamalarını gerçekleştirmenize yardımcı olmak için IBM MQ for z/OS ile birlikte sağlar.

CSQUTIL yardımcı programı hakkında daha fazla bilgi için bkz. [IBM MQ yardımcı programı \(CQUOtil\)](#). Bu yardımcı programı kullanarak aşağıdaki işlevleri çağırabilirsiniz:

Komut

MQSC komutlarını vermek, nesne tanımlamalarını kaydetmek ve istemci-kanal tanımlama dosyalarını yapmak için.

Kopyala

Adlandırılmış bir IBM MQ for z/OS ileti kuyruğunun içeriğini ya da adlandırılmış bir sayfa kümesinin tüm kuyruklarının içeriğini okumak ve bunları sıralı bir dosyaya koymak ve özgün kuyruğu alıkoymak için.

KOPYALA

Tüm sayfa kümelerini daha büyük sayfa kümelerine kopyalamak için.

EMPTY

Adlandırılmış bir IBM MQ for z/OS ileti kuyruğunun içeriğini ya da adlandırılmış bir sayfa kümesinin tüm kuyruklarının içeriğini silmek için, kuyrukların tanımlamalarını korumasını sağlar.

Biçim

IBM MQ for z/OS sayfa kümelerini biçimlendirmek için.

LOAD

Adlandırılmış bir IBM MQ for z/OS ileti kuyruğunun içeriğini ya da bir adlandırılmış sayfanın tüm kuyruklarının içeriğini, COPY işleviyle yaratılan sıralı bir dosyadan geri yüklemek için.

PAGEINFO

Sayfa kümesi bilgilerini bir ya da daha çok sayfa kümesinden ayıklamak için.

YENİDEN DÜZENLE

Tüm sayfa kümelerini diğer sayfa kümesi veri kümelerine kopyalamak için, kopyadaki günlük bilgilerini ilk durumuna getirin.

SKOPI

Kuyruk yöneticisi çevrimdışıyken bir kuyruğun içeriğini bir veri kümesine kopyalamak için.

SDEFS

Kuyruk yöneticisi çevrimdışıyken nesnelere için bir tanımlama komutları kümesi üretmek için.

YÜKLE

Önceki COPY YA DA SKOPI işleminin hedef veri kümesinden gelen iletileri geri yüklemek için. SLOAD işlemi tek bir kuyruğu işler.

Diğerine Geç

Küme gönderen kanallarıyla ilişkili iletim kuyruğunu değiştirmek ya da sorgulamak için.

XPARM

Bir kanal başlatıcı parametre yükleme modülünü kuyruk yöneticisi özniteliklerine dönüştürmek için (geçiş amacıyla).

z/OS Çalışırken IBM MQ for z/OS

IBM MQ for z/OS' i çalıştırmak için bu temel yordamları kullanın.

You can also perform the operations described in this section using the IBM MQ Explorer, which is distributed with IBM MQ for Windows and IBM MQ for Linux (x86 and x86-64 platforms). Fix Central' tan karşıdan yükleme için bağımsız bir IBM MQ Explorer kullanılabilir. Daha fazla bilgi için bkz [“IBM MQ Explorer komutunu kullanarak yönetim” sayfa 110](#).

z/OS z/OS üzerinde kuyruk yöneticisi komutları yayınlıyor

IBM MQ denetim komutlarını bir z/OS konsolundan ya da yardımcı program CSQUTIL yardımcı programıyla yayınlatabilirsiniz. Komutlar, komutu hangi IBM MQ altsistemine işleyeceğini belirtmek için komut öneki dizgisini (CPF) kullanabilir.

You can control most of the operational environment of IBM MQ using the IBM MQ commands. IBM MQ for z/OS , bu komutların hem MQSC ' yi hem de PCF tiplerini destekler. Bu konuda, MQSC komutlarını kullanarak özniteliklerin nasıl belirtileceği anlatılır ve bu komut, PCF adları yerine MQSC komut adlarını kullanarak bu komutları ve öznitelikleri belirtir. MQSC komutlarının sözdizimine ilişkin ayrıntılar için bkz. MQSC komutları. PCF komutlarının sözdizimine ilişkin ayrıntılar için bkz. [“IBM MQ Programları Komut Biçimlerinin Kullanılması” sayfa 24](#). Uygun bir yetkili bir kullanıcısanız, aşağıdaki komutlardan IBM MQ komutlarını yayınlatabilirsiniz:

- Kullanıma hazırlama giriş veri kümeleri ([“IBM MQ for z/OS için başlatma komutları” sayfa 412](#) içinde açıklanmıştır).
- Bir z/OS konsolu ya da eşdeğeri; örneğin, SDSF
- z/OS master get komutu rutini, MGCRE (SVC 34)
- IBM MQ yardımcı programı, CSQUTIL ([IBM MQ yardımcı programı](#) içinde açıklanmıştır.)
- Aşağıdaki olabilen bir kullanıcı uygulaması:
 - Bir CICS programı
 - TSO programı
 - Bir z/OS toplu iş programı
 - Bir IMS programı

Bununla ilgili bilgi için bkz. [“Writing programs to administer IBM MQ for z/OS” sayfa 436](#) .

Bu komutların işlevlerinin çoğu, operasyon ve denetim panelleri tarafından, TSO ve ISPF ' den erişilebilen ve [“IBM MQ for z/OS için işlemler ve denetim panelleri” sayfa 422](#) içinde açıklandığı şekilde uygun bir şekilde sağlanır.

Ek bilgi için bkz.

- [“Issuing commands from a z/OS console or its equivalent” sayfa 416](#)
 - [Komut öneki dizgileri](#)
 - [Komut vermek için z/OS konsolunun kullanılması](#)
 - [Komut yanıtları](#)
- [CSQUTIL yardımcı programdan komut verilmesi](#)

Issuing commands from a z/OS console or its equivalent

Tüm IBM MQ komutlarını bir z/OS konsolundan ya da eşdeğer bir konsoldan yayınlayabilirsiniz. You can also issue IBM MQ commands from anywhere where you can issue z/OS commands, such as SDSF or by a program using the MGCRC macro.

Konsolda yazılan bir komutun sonucu olarak görüntülenebilecek veri miktarı üst sınırı 32 KB 'dir.

Not:

1. Bir IMS uçbiriminden IMS/SSR komut biçimini kullanarak IBM MQ komutlarını yayınlayamazsınız. Bu işlev IMS bağdaştırıcısı tarafından desteklenmez.
2. SDSF tarafından sağlanan giriş alanı, bazı komutlar için, özellikle de kanallara ilişkin komutlar için yeterli uzunlukta olmayabilir.

Komut öneki dizgileri

Her IBM MQ komutunun öneki, [Şekil 32 sayfa 417](#) içinde gösterildiği gibi bir komut öneki dizgisiyle (CPF) önek olarak eklenmelidir.

Because more than one IBM MQ subsystem can run under z/OS, the CPF is used to indicate which IBM MQ subsystem processes the command. For example, to start the queue manager for a subsystem called CSQ1, where CPF is ' +CSQ1 ', you issue the command +CSQ1 START QMGR from the operator console. Bu CPF, altsistem adı çizelgesinde (CSQ1altsistemi için) tanımlanmalıdır. Bu, [Komut öneki dizgilerinin tanımlanması \(CPF ' ler\)](#) başlıklı konuda açıklanmaktadır. Örneklerde, dizgi ' +CSQ1 ' komut öneki olarak kullanılır.

Komutları vermek için z/OS konsolunu kullanma

You can type simple commands from the z/OS console, for example, the DISPLAY command in [Şekil 32 sayfa 417](#). Ancak, karmaşık komutlar için ya da sık sık kullandığınız komut kümeleri için, komut verme yöntemlerinin diğer yöntemleri daha iyidir.

```
+CSQ1 DISPLAY QUEUE(TRANSMIT.QUEUE.PROD) TYPE(QLLOCAL)
```

Şekil 32. z/OS konsolundan bir DISPLAY komutu verilmesi

Komut yanıtları

Komutlara doğrudan yanıtlar, komutu veren konsola gönderilir. IBM MQ , z/OS' ta kullanılabilir olan *Extended Console Support* (EMCS) işlevini destekler ve bu nedenle 4 baytlık tanıtıcılar içeren konsollar kullanılabilir. Ayrıca, QMGR ve STOP QMGR dışındaki tüm komutlar, MGCRC makrosunu kullanan bir program tarafından verildiğinde, Komut ve Yanıt Belirteçleri (CARTs) kullanımını destekler.

CSQUTIL yardımcı programından komut verilmesi

CSQUTIL yardımcı programının COMMAND işlevini kullanarak sıralı bir veri kümesinden komut yayınlayabilirsiniz. Bu yardımcı program, komutlar olarak, *sistem komutu giriş kuyruğuna* ilişkin komutları aktarır ve SYPRINE ' deki özgün komutlarla birlikte yazdırılan yanıtın gelmesini bekler. Bu konuya ilişkin ayrıntılı bilgi için [IBM MQ Utility programı](#) başlıklı konuya bakın.

z/OS üzerinde kuyruk yöneticisini başlatma ve durdurma

Kuyruk yöneticisini durdurmak ve başlatmak için bu konuyu bir giriş olarak kullanın.

Bu kısımda, bir kuyruk yöneticisinin nasıl başlatılacağı ve durdurulacağı ele alınmıştır. Aşağıdaki konularla ilgili bilgileri içerir:

- “Başlamadan ÖnceIBM MQ” sayfa 418
- “Kuyruk Yöneticisinin Başlatılması” sayfa 418
- “Kuyruk Yöneticisinin Durdurulması” sayfa 420

Kuyruk yöneticisini başlatma ve durdurma görel olarak basittir. Kuyruk yöneticisi olağan koşullarda durduğunda, son işlemi sonlandırma denetim noktasını almandır. Bu denetim noktası ve günlükler, kuyruk yöneticisine yeniden başlatılması için gereken bilgileri verir.

Bu kısım, START ve STOP komutlarıyla ilgili bilgileri içerir ve olağandışı bir sona erdirme gerçekleşikten sonra başlatma sırasında kısa bir genel bakış içerir.

Başlamadan ÖnceIBM MQ

After you have installed IBM MQ, it is defined as a formal z/OS subsystem. Bu ileti, z/OS' un ilk program yüklemesi (IPL) sırasında görüntülenir:

```
CSQ3110I +CSQ1 CSQ3UR00 - SUBSYSTEM ssnm INITIALIZATION COMPLETE
```

Burada *ssnm* , IBM MQ altsistem adıdır.

Bundan sonra, o altsistem için kuyruk yöneticisini (*Sistem denetimi komutlarını verme yetkisi bulunan herhangi bir z/OS konsolundan*), bir z/OS SYS komut grubuna başlatabilirsiniz. Yetkili konsoldan START komutunu vermelisiniz; bu komutu JES ya da TSO üzerinden yayınlamayabilirsiniz.

Kuyruk paylaşım gruplarını kullanıyorsanız, kuyruk yöneticisini başlatmadan önce önce RRS 'yi önce RRS' yi ve sonra Db2' u başlatmanız gerekir.

Kuyruk Yöneticisinin Başlatılması

Bir kuyruk yöneticisi başlatarak QMGR START komutunu çalıştırın. Ancak, uygun yetkiniz yoksa, START komutunu başarıyla kullanamazsınız. IBM MQ güvenliğine ilişkin bilgi için [z/OS üzerinde güvenliğin ayarlanması](#) başlıklı konuya bakın. [Şekil 33 sayfa 418](#) , START komutuna ilişkin örnekleri gösterir. (Komut öneki dizgisi (CPF) olan bir IBM MQ komutuna önek atmanız gerektiğini unutmayın.)

```
+CSQ1 START QMGR
+CSQ1 START QMGR PARM(NEWLOG)
```

Şekil 33. Kuyruk yöneticisinin bir z/OS konsolundan başlatılması

START QMGR komutunun sözdizimiyle ilgili bilgi için [START QMGR](#) başlıklı konuya bakın.

You cannot run the queue manager as a batch job or start it using a z/OS command START. Bu yöntemler, IBM MQ için bir adres alanı başlatıp olağandışı bir şekilde sona erme olasılığının yüksek olması durumunda. Ne de CSQUTIL yardımcı programından ya da benzer bir kullanıcı uygulamasından kuyruk yöneticisi başlamanıza neden olmaz.

You can, however, start a queue manager from an APF-authorized program by passing a START QMGR command to the z/OS MGCRE (SVC 34) service.

Kuyruk paylaşım grupları kullanıyorsanız, kuyruk yöneticisini başlatırken ilişkili Db2 sistemleri ve RRS etkin olmalıdır.

Başlangıç seçenekleri

Kuyruk yöneticisi başlattığınızda, bir sistem değiştirgesi birimi yüklenir. Sistem değiştirge modülünün adını aşağıdaki iki yoldan biriyle belirtebilirsiniz:

- /cpf START QMGR komutunun PARM parametresiyle birlikte, örneğin

```
/cpf START QMGR PARM(CSQ1ZPRM)
```

- Örneğin, başlatma yordamında bir değiştirgeyle, JCL EXEC deyiminin kodu

```
//MQM EXEC PGM=CSQYASCP,PARM='ZPARAM(CSQ1ZPRM)'
```

Bir sistem parametre modülü, kuyruk yöneticisi uyarlandığında belirtilen bilgileri sağlar.

IBM MQ 9.1.0 seçeneğinden, kuyruk yöneticisi kullanımının kaydedileceği ürünü belirtmek için **QMGRPROD** seçeneğini ve kullanılırsa, AMS ' nin eşdeğerini belirtmek için **AMSPROD** seçeneğini kullanabilirsiniz. İzin verilen değerlerin ayrıntıları için MQSC START QMGR komutuna bakın.

Örnek bir JCL EXEC deyimi aşağıdaki gibidir:

```
//MQM EXEC PGM=CSQYASCP,PARM='QMGRPROD(MQ)'
```

Ürün kullanımı kaydına ilişkin daha fazla bilgi için bkz. [z/OS MVS Ürün Yönetimi](#) .

Kuyruk yöneticisine ilişkin JCL yordamında bir ya da daha çok değiştirenin yerini almak için ENVPARM seçeneğini de kullanabilirsiniz.

Örneğin, kuyruk yöneticisi başlatma yordamınızı güncelleyebilirsiniz; böylece DDname CSQINP2 bir değiştendir. Bu, başlatma yordamını değiştirmeden CSQINP2 DDadını değiştirebileceğiniz anlamına gelir. Bu, işlemler için geriletme ve kuyruk yöneticisi işlemleri sağlayan değişiklikleri uygulamak için kullanışlıdır.

Kuyruk yöneticisi CSQ1 için başlatma yordamınızın [Şekil 34 sayfa 419](#) gibi görüldüğünü varsayın.

```
//CSQ1MSTR PROC INP2=NORM
//MQMESA EXEC PGM=CSQYASCP
//STEPLIB DD DISP=SHR,DSN=th1qual.SCSQANLE
// DD DISP=SHR,DSN=th1qual.SCSQAUTH
// DD DISP=SHR,DSN=db2qual.SDSNLOAD
//BDS1 DD DISP=SHR,DSN=myqual.BSDS01
//BDS2 DD DISP=SHR,DSN=myqual.BSDS02
//CSQP0000 DD DISP=SHR,DSN=myqual.PSID00
//CSQP0001 DD DISP=SHR,DSN=myqual.PSID01
//CSQP0002 DD DISP=SHR,DSN=myqual.PSID02
//CSQP0003 DD DISP=SHR,DSN=myqual.PSID03
//CSQINP1 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1INP1)
//CSQINP2 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1&INP2.)
//CSQOUT1 DD SYSOUT=*
//CSQOUT2 DD SYSOUT=*
```

Şekil 34. Örnek başlatma yordamı

Daha sonra, kuyruk yöneticinizi şu komutla başlarsınız:

```
+CSQ1 START QMGR
```

Kullanılan CSQINP2 , CSQ1NORM adlı bir üyedir.

Ancak, üretim için yeni bir program paketi yerleştirdiğinizi varsayın; böylece, kuyruk yöneticisi CSQ1' u bir sonraki başlatışığınızda CSQINP2 tanımlarının CSQ1NEWüyesinden alınması gerekir. Bunu yapmak için, şu komutla kuyruk yöneticisini başlatasınız:

```
+CSQ1 START QMGR ENVPARM(' INP2=NEW')
```

CSQ1NORMyerine CSQ1NEW kullanılır. Not: z/OS , simgesel parametrelere (INP2=NEW' de olduğu gibi) ilişkin KEYWORD=va1ue belirtilerini 255 karakter ile sınırlar.

Olağandışı sona erdirme işleminden sonra başlatma

IBM MQ , yeniden başlatmanın olağan bir kapanma mı, yoksa olağandışı sonlandırma mı olduğunu otomatik olarak saptar

Bir kuyruk yöneticisi olağan dışı sona erdikten sonra, STOP QMGR komutu verildikten sonra olağan dışı bir şekilde başlatıldıktan sonra başlatılmış olur. STOP QMGR ' yi durdurduktan sonra, sistem çalışmasını düzenli bir şekilde bitirir ve durdurmadan önce bir sonlandırma denetim noktası alır. Kuyruk yöneticisini yeniden başlattığınızda, sistem durumunu sona erdirme sırasında belirlemek için sistem denetim noktasından ve kurtarma gününden gelen bilgileri kullanır.

Ancak, kuyruk yöneticisi olağandışı sona ererse, işini bitirmeden ya da sonlandırma denetim noktasını ele geçirmeden sonlandırılır. Bir olağandışı sondan sonra kuyruk yöneticisini yeniden başlattığınızda, günlüğün sonundaki bilgileri kullanarak sonlandırma sırasında durumunu yeniler ve çeşitli görevlerin durumunu size bildirir. Olağan durumda, yeniden başlatma işlemi tüm tutarsız durumları çözer. Ancak, bazı durumlarda tutarsızlıkları çözmek için belirli adımları atmanız gerekir.

Başlatma sırasında kullanıcı iletileri

Bir kuyruk yöneticisini başarıyla başlattığınızda, kuyruk yöneticisi bir başlatma iletileri kümesi üretir.

Kuyruk Yöneticisinin Durdurulması

Bir kuyruk yöneticisini durdurmadan önce, IBM MQile ilgili tüm yazma işlemi yanıtama (WTOR) iletilerinin yanıt almalıdır; örneğin, günlük istekleri alınıyor. [Şekil 35 sayfa 420](#) içindeki her komut, çalışan bir kuyruk yöneticisini sonlandırmış olur.

```
+CSQ1 STOP QMGR
+CSQ1 STOP QMGR MODE(QUIESCE)
+CSQ1 STOP QMGR MODE(FORCE)
+CSQ1 STOP QMGR MODE(RESTART)
```

Şekil 35. Kuyruk Yöneticisinin Durdurulması

STOP QMGR komutu, QMGR kipinin (QUIESCE) DURDURULACAK varsayılanlarını durdurur.

QUIESCE kipindeki IBM MQ , yeni bağlantı iş parçacıklarının yaratılmasına izin vermez, ancak varolan iş parçacıklarının devam etmesine izin verir; bu, yalnızca tüm iş parçacıkları sona erdiğinde sona erer. Uygulamalar, kuyruk yöneticisinin susturulması durumunda bildirim almak için istekte bulunabilir. Bu nedenle, istekte bulunulan uygulamaların, bağlantı kesme fırsatına sahip olması için QUIESCE kipinin kullanılmasını olanaklı bir şekilde kullanın. Ayrıntılar için bkz. [Sonlandırma sırasında olanlar](#) .

Kuyruk yöneticisi STOP QMGR MODE (QUIESCE) komutuna yanıt olarak makul bir süre içinde sona ermezse, bağlantı iş parçacıklarının var olup olmadığını saptamak için DISPLAY CONN komutunu kullanın

ve ilişkili uygulamaları sonlandırmak için gereken adımları gerçekleştirin. İş parçacığı yoksa, STOCE QMGR MODE (FORCE) komutunu verin.

STOP QMR MODE (QUIESCE) ve STOP QMR MODE (FORCE) komutları, MVS Automatic Restart Manager (ARM) olanağından IBM MQ kaydını siler ve ARM 'in kuyruk yöneticisini otomatik olarak yeniden başlatmasını önler. STOP QMGR MODE (RESTART) komutu, IBM MQ 'un ARM' den silinmemesi dışında, 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 kaydedilmediyse, STOP QMR 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 \(ARM\) olanağını kullanma](#)" sayfa 496.

STOP QMR MODE (FORCE) işlemi kuyruk yöneticisini sonlandırmazsa, yalnızca kuyruk yöneticisi adres alanını iptal edin.

Bir kuyruk yöneticisi, adres alanını iptal ederek ya da STOP QMGR MODE (FORCE) komutunu kullanarak durdurulursa, tutarlılık, bağlı CICS ya da IMS sistemleriyle sağlanır. Bir kuyruk yöneticisi yeniden başlatıldığında ve CICS ya da IMS sistemiyle bağlantı kurulduğunda tamamlandığında, kaynakların yeniden eşzamanlanması başlatılır.

Not: When you stop your queue manager, you might find message IEF352I is issued. z/OS , adres alanını kullanılamaz olarak işaretlemeyen bir hata saptarsa, bu ileti bir bütünlük açığa çıkmasına neden olur. Bu iletiyi yoksayabilirsiniz.

Durdurma iletileri

Bir STOP QMGR komutu yayınladıktan sonra, CSQY009I ve CSQY002I iletilerine örnek olarak bakın:

```
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 belirtilenlere bağlıdır.

STOP komutu başarıyla tamamlandığı zaman, z/OS konsolunda aşağıdaki iletiler görüntülenir:

```
CSQ9022I +CSQ1 CSQYASCP ' STOP QMGR' NORMAL COMPLETION  
CSQ3104I +CSQ1 CSQ3EC0X - TERMINATION COMPLETE
```

ARM kullanıyorsanız ve MODE (RESTART) belirtmediyseniz, 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
```

IBM MQ for z/OS için işlemler ve denetim panoları

You can use the IBM MQ operations and control panels to perform administration tasks on IBM MQ objects. Komutlara ve denetim panolarına giriş olarak bu konuyu kullanın.

Bu panoları, IBM MQ nesnelere tanımlamak, görüntülemek, değiştirmek ya da silmek için kullanıyorsunuz. Günlük denetim için ve nesnelere küçük değişiklikler yapmak için panoları kullanın. Birçok nesneyi ayarlıyorsanız ya da değiştiriyorsanız, CSQUTIL yardımcı programı COMMAND işlevini kullanın.

İşlem ve denetim panoları, kanal başlatıcı için denetimler destekler (örneğin, bir kanalı ya da TCP/IP dinleyicisini başlatmak için), kümeleme için ve güvenlik için. Ayrıca, iş parçacıklarına ve sayfa kümesi kullanımına ilişkin bilgileri görüntülemenizi de sağlar.

Panolar, MQSC tipi IBM MQ komutlarını bir kuyruk yöneticisine sistem komut giriş kuyruğu yoluyla göndererek çalışır.

Not:

1. IBM MQ for z/OS işlemleri ve denetimleri panoları (CSQOREXX), sürüm 7 'den başlayarak eklenen tüm yeni işlevi ve değişiklikleri desteklemeyebilir. Örneğin, konu nesnelere ya da aboneliklerine doğrudan işlenmesine ilişkin herhangi bir pano yoktur.

Aşağıdaki desteklenen mekanizmalardan birini kullanarak, diğer panolardan doğrudan olmayan yayınlama/abone olma tanımlarını ve diğer sistem denetimlerini yönetmenize olanak sağlar:

- a. IBM MQ Gezgin
- b. z/OS Konsol
- c. Programlanabilir Komut Biçimi (PCF) iletileri
- d. CSQUTIL ' in COMMAND işlevi
- e. IBM MQ Web Konsolu

CSQOREXX panolarındaki genel **Command** işleminin, SMDS ile ilgili komutlar da içinde olmak üzere geçerli bir MQSC komutu vermenize olanak tanıdığı unutmayın. CSQUTIL sorunlarıyla ilgili COMMAND işlevinin tüm komutlarını kullanabilirsiniz.

2. You cannot issue the IBM MQ commands directly from the command line in the panels.
3. İşlemleri ve denetim panolarını kullanmak için, doğru güvenlik yetkilendirmesine sahip olmanız gerekir; bu, [Komut güvenliği ve komut kaynağı güvenliği için kullanıcı kimlikleri](#) içinde açıklanmıştır.
4. CSQUTIL ya da CSQOREXX panolarını kullanarak bir kullanıcı kimliği ve parola sağlayamazsınız. Bunun yerine, kullanıcı kimliğiniz MQCONN ' da BATCH profili için UPDATE yetkisi varsa, **CHCKLOCL(REQUIREND)** ayarını atlayabilirsiniz. Ek bilgi için [Yerel olarak bağlı uygulamalarda CHCKLOCL ' in kullanılması](#) başlıklı konuya bakın.

Operasyon ve denetim panolarına ilişkin başlatma ve kurallar

You can control IBM MQ and issue control commands through the ISPF panels.

IBM MQ işlemlerine ve denetim panolarına erişme

If the ISPF/PDF primary options menu has been updated for IBM MQ, you can access the IBM MQ operations and control panels from that menu. Menü'nün güncellenmesiyle ilgili ayrıntılar için [Görev 20: İşlemleri ve denetim panolarını ayarlamabaşlıklı konuya bakın.](#)

IBM MQ işlemlerine ve denetim panolarına TSO komut işlemcisi panosundan erişebilirsiniz (genellikle, ISPF/PDF birincil seçenekler menüsünde 6. seçenek). Bu işlemi yapmak için çalıştırdığınız EXEC adı CSQOREXX. Bunun iki parametresi vardır; th1qua1 kullanılacak IBM MQ kitaplıklarının üst düzey niteleyicidir ve lang1ettter kullanılacak ulusal dil kitaplıklarını tanımlayan harftir (örneğin, U.Siçin E). İngilizce). IBM MQ kitaplıkları ISPF kurulumunuza kalıcı olarak kurulduysa, parametreler atlanabilir. Diğer bir seçenek olarak, TSO komut satırından CSQOREXX komutunu da yayınlatabilirsiniz.

Bu panolar, en az resmi eğitime sahip işletmenler ve yöneticiler tarafından kullanılmak üzere tasarlanmıştır. Bu yönergeleri, çalışan panolarla okuyun ve önerilen farklı görevleri deneyin.

Not: While using the panels, temporary dynamic queues with names of the form SYSTEM.CS\$QOREXX.* yaratılır.

İşlemlere ve denetim panolarına ilişkin kurallar

IBM MQ karakter dizgileri ve adlarına ilişkin genel kurallarla ilgili olarak [IBM MQ nesnelерinin adlandırılmasına ilişkin kurallar](#) konusuna bakın. Ancak, yalnızca işlemler ve denetim panolarında geçerli olan bazı kurallar vardır:

- Dizgileri, örneğin açıklamaları tek ya da çift tırnak içine almayın.
- Bir metin alanına bir kesme işareti ya da tırnak işareti eklerseniz, bunu yinelemek ya da bir kaçış karakteri eklemeniz gerekmez. Karakterler tam olarak sizin yazdığınız şekilde kaydedilir; örneğin:

```
This is Maria's queue
```

Pano işlemcisi, bunları IBM MQ' a iletmeniz için iki katına çıkarır. Ancak, bunu yapmak için verilerinizi kesmeniz gerekiyorsa, bunu yapar.

- Çoğu alanda büyük ya da küçük harfli karakterler kullanabilirsiniz ve Enter tuşuna bastığınızda büyük harf karakterlerine katlanabilirsiniz. Kural dışı durumlar şunlardır:
 - Depolama sınıfı adları ve bağlaşım olanağı yapısı adları; A-Z arasındaki büyük harflerle başlayıp Z-Z ya da sayısal karakterler arasında büyük harfle takip edilmesi gereken adlar.
 - Çevrilmemiş belirli alanlar. Bu üyeler şunlardır:
 - Uygulama tanıtıcısı
 - Tanım
 - Ortam verileri
 - Nesne adları (ancak küçük harfli bir nesne adı kullanırsanız, bu adı bir z/OS konsoluna giremeyebilirsiniz)
 - Uzak sistem adı
 - Verileri tetikle
 - Kullanıcı verileri
- Adlarda, baştaki boşluklar ve önde gelen altçizgi yoksayılr. Bu nedenle, nesne adlarına boşluk ya da altçizgi karakterleriyle başlayan bir ad giremezsiniz.
- Alt çizgiler, boş alanların kapsamını göstermek için kullanılır. Enter tuşuna bastığınızda, sondaki altçizgi boşluk karakteriyle değiştirilir.
- Birçok açıklama ve metin alanı, her bir parçanın IBM MQ tarafından bağımsız olarak ele alınmakta olan birden çok parçada gösterilir. Bu, sondaki boşlukların alıkonacağı ve metnin bitişik olmadığı anlamına gelir.

Boş alanlar

Bir IBM MQ nesnesi için **Tanımla** işlemi belirttiğinizde, tanım panelindeki her alan bir değer içerir. IBM MQ ' in değerleri aldığı yere ilişkin bilgi için görüntü panolarına ilişkin genel yardım (genişletilmiş yardım) başlıklı konuya bakın. Boşluk içeren bir alanın üzerine yazarsanız ve boşluklara izin verilmiyorsa, IBM MQ kuruluş varsayılan değerini alana koyar ya da gereken değeri girmenizi ister.

Bir IBM MQ nesnesi için **Alter** işlemi belirttiğinizde, değişiklik panosundaki her alan o alanın geçerli değerini içerir. Bir alanı boşluk karakteriyle yazarsanız ve boşluklara izin verilmiyorsa, alanın değeri değişmez.

z/OS üzerindeki nesnelere ve işlemler

İşlem ve denetim panoları size birçok farklı nesne tipi ve üzerinde gerçekleştirebileceğiniz eylemlerin sayısını sunar.

İşlemler, ilk panoda listelenir ve nesnelere üzerinde işlem yapmak ve bu nesnelere ilgili bilgileri görüntülemek için bu işlemleri geçerli kılar. Bu nesnelere, tüm IBM MQ nesnelere, bazı ekstrelerle birlikte içerir. Nesnelere aşağıdaki kategorilere ayrılır.

- [Kuyruklar, işlemler, kimlik doğrulama bilgileri nesnelere, ad listeleri, depolama sınıfları ve CF yapıları](#)
- [Kanallar](#)
- [Küme nesnelere](#)
- [Kuyruk yöneticisi ve güvenlik](#)
- [Bağlantılar](#)
- [Sistem](#)

Refer to [Eylemler](#) for a cross-reference table of the actions which can be taken with the IBM MQ objects.

Kuyruklar, işlemler, kimlik doğrulama bilgileri nesnelere, ad listeleri, depolama sınıfları ve CF yapıları

Bunlar temel IBM MQ nesnelereidir. Her tipten bir çok kişi olabilir. Bunlar, LIST ya da DISABLE, LIST WITH FILTER, LIKE, MANAGE ve ALTER eylemlerini kullanarak, süzgeç, tanımlanmış ve silinmiş ve görüntülenebilecek ve değiştirilebilen özniteliklere sahip listelenmiş olabilir. (Nesnelere, MANAGE işlemleri kullanılarak silinir.)

Bu kategori aşağıdaki nesnelere oluşur:

QLOCAL	Yerel kuyruk
QREMOTE	Uzak kuyruk
QALIAS	Bir kuyruğa dolaylı başvuru için diğer ad kuyruğu
QMODEL	Kuyrukları devingen olarak tanımlamak için kullanılan model kuyruğu
kuyruk	Herhangi bir kuyruk tipi
QSTATUS	Yerel kuyruğun durumu
Süreç	Bir tetikleme olayı ortaya çıktığında başlatılacak uygulamaya ilişkin bilgiler
YAZAR BİLGİLERİ	Kimlik doğrulama bilgileri: LDAP sunucularını kullanarak Sertifika İptal Listesi (CRL) denetimini gerçekleştirmek için gereken tanımlamalar
NAMELIST	Ad listesi (kuyruklar ya da kümeler gibi)
STGCLASS	Depolama sınıfı
CFUSTRİ	bağlaşım tesisi (CF) yapısı
CFSTATUS	CF yapısının durumu

Kanallar

Dağıtılmış kuyruklama için kanallar kullanılır. Her tipte çok sayıda olabilir ve bunlar süzgeç, tanımlı, silinmiş, görüntülenen ve değiştirilmiş olarak listelenmiş, listelenmiş olabilir. Ayrıca, START, STOP ve PERFORM eylemlerini kullanan diğer işlevleri de vardır. PERFORM, kanal işlevlerini ilk durumuna getirme, ping isteği ve çözme işlemleri sağlar.

Bu kategori aşağıdaki nesnelere oluşur:

Kanal	Herhangi bir kanal tipi
Gönderen	Gönderen kanalı
SERVER	Sunucu kanalı

Alıcı	Alıcı kanalı
İsteyen	İstekte bulunanın kanalı
CLUSRCVR	Küme-alıcı kanalı
CLUSDR	Küme-gönderen kanalı
SVRCONN	Sunucu bağlantısı kanalı
NTNTCONN	İstemci bağlantı kanalı
SOHBET DURUMU	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 ad olabilir ve adlar yinelenir. Bunlar, süzgeç ile listelenmiş ve görüntülenebilirler. PERFORM, START ve STOP, LIST işlemleri aracılığıyla da kullanılabilir.

Bu kategori aşağıdaki nesnelere oluşur:

CLUSQ	Bir kümeye ait olan bir kuyruk için oluşturulan küme kuyruğu
CLUSCHL	Bir kümeye ait olan bir kanal için oluşturulan küme kanalı
CLUSQMgr	Küme kuyruğu yöneticisi, bir küme kanalıyla aynı, ancak kuyruk yöneticisi adına göre tanıtılıyor

Küme kanalları ve küme kuyruğu yöneticilerinin PERFORM, START ve STOP işlemleri vardır, ancak yalnızca dolaylı olarak DISABLE işlemi aracılığıyla yapılır.

Kuyruk yöneticisi ve güvenlik

Kuyruk yöneticisi ve güvenlik nesneleri tek bir yönetim ortamı içerir. Bunlar listelenebilirler ve görüntülenebilir ve değiştirilebilir öznelikler (LIST ya da DISABLE ve ALTER işlemleri kullanılarak) ve PERFORM işlemi kullanılarak kullanılacak başka işlemlere sahip olabilir.

Bu kategori aşağıdaki nesnelere oluşur:

Yönetici	Kuyruk yöneticisi: PERFORM işlemi, askıya alma ve sürdürme kümesi işlevlerini sağlar
Güvenlik	Güvenlik işlevleri: PERFORM işlemi yenileme ve yeniden doğrulama işlevlerini sağlar

Bağlantı

Bağlantılar, süzgeç uygulayıp, görüntülenerek listelenmiş olabilir.

Bu kategori yalnızca bağlantı nesnesinden, CONNECT ' den oluşur.

Sistem

Diğer işlevlerden oluşan bir derlem. Bu kategori aşağıdaki nesnelere oluşur:

SYSTEM	Sistem işlevleri
CONTROL	SYSTEM eşanlamlısı

Kullanılabilir işlevler şunlardır:

LIST YA DA DISPLAY	Kuyruk paylaşım grubunu, dağıtılmış kuyruklama, sayfa kümesi ya da veri kümesi kullanım bilgilerini görüntüler.
KİŞİ	Kümelemeyi yenileme ya da sınırlama

START

Kanal başlatıcıyı ya da dinleyicilerini başlatır

DUR

Kanal başlatıcıyı ya da dinleyicilerini durdurun

İşlemler

Her nesne tipi için gerçekleştirebileceğiniz eylemler, aşağıdaki tabloda gösterilir:

<i>Çizelge 27. IBM MQ nesneleri için geçerli işlemler ve denetim panosu işlemleri</i>								
Nesne	Değiştir	Benzer tanımla	Yönet (1)	Liste ya da Görüntü	Süzgeçle Listele	Gerçekleştir	Başlangıç	Durdur
YAZAR BİLGİLERİ	X	X	X	X	X			
CFSTATUS				X				
CFUSTRİ	X	X	X	X	X			
Kanal	X	X	X	X	X	X	X	X
SOHBET DURUMU				X	X			
NTNTCONN	X	X	X	X	X			
CLUSCHL				X	X	X (2)	X (2)	X (2)
CLUSQ				X	X			
CLUSQMGR				X	X	X (2)	X (2)	X (2)
CLUSRCVR	X	X	X	X	X	X	X	X
CLUSDR	X	X	X	X	X	X	X	X
CONNECT				X	X			
CONTROL				X		X	X	X
Yönetici	X			X		X		
NAMELIST	X	X	X	X	X			
Süreç	X	X	X	X	X			
QALIAS	X	X	X	X	X			
QLOCAL	X	X	X	X	X			
QMODEL	X	X	X	X	X			
QREMOTE	X	X	X	X	X			
QSTATUS				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
Güvenlik	X			X		X		
Gönderen	X	X	X	X	X	X	X	X
SERVER	X	X	X	X	X	X	X	X

Çizelge 27. IBM MQ nesneleri için geçerli işlemler ve denetim panosu işlemleri (devamı var)

Nesne	Değiştir	Benzer tanımla	Yönet (1)	Liste ya da Görüntü	Süzgeçle Liste	Gerçekleştir	Başlangıç	Durdur
SVRCONN	X	X	X	X	X		X	X
STGCLASS	X	X	X	X	X			
SYSTEM				X		X	X	X

Not:

1. Sil ve diğer işlevleri sağlar.
2. Liste ya da Görüntüle işlemini kullanma

z/OS z/OSüzerindeki nesne atımları

Çalışmanız gereken nesnenin *yok etme* değerini belirtebilirsiniz. Yok etme, **tanım** nesnesinin nerede tutulduğu ve nesnenin nasıl davrandığı anlamına gelir.

Yok etme, yalnızca aşağıdaki nesne tiplerinden herhangi biriyle çalışıyorsanız önem atmaktadır:

- Kuyruklar
- kanallar
- işlemler
- ad listeleri
- depolama sınıfları
- kimlik doğrulama bilgileri nesneleri

Diğer nesne tipleriyle çalışıyorsanız, yok etme (yok etme) dikkate alınmaz.

İzin verilen değerler şunlardır:

Q

QMGR. Nesne tanımlamaları kuyruk yöneticisinin sayfa kümesinde yer alıyor ve yalnızca kuyruk yöneticisi tarafından erişilebilir.

C

KOPYALA. Nesne tanımlamaları kuyruk yöneticisinin sayfa kümesinde yer alıyor ve yalnızca kuyruk yöneticisi tarafından erişilebilir. Bunlar, GROUP ' un yok edilmesi olarak tanımlanan nesnelere yerel kopyalarıdır.

P

ÖZEL. Nesne tanımlamaları kuyruk yöneticisinin sayfa kümesinde yer alıyor ve yalnızca kuyruk yöneticisi tarafından erişilebilir. Nesnelere, QMGR ya da COPY ' ye ilişkin bir yok etme olarak tanımlanmıştır.

G

GRUP. Nesne tanımlamaları paylaşılan havuzda yer alıyor ve kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından erişilebilir.

S

-SHARON. Bu atma yalnızca yerel kuyruklar için geçerlidir. Kuyruk tanımlamaları paylaşılan havuzda yer alıyor ve kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından erişilebilir.

A

Tüm. İşlem kuyruğu yöneticisi hedef kuyruk yöneticisiyse ya da *, **Tümü** işlemlerinin nesnelere dahil edilir; tersi durumda, QMGR ve COPY atımları yalnızca nesnelere dahil edilir. Bu varsayılandır.

z/OS z/OSüzerinde ISPF denetim masasını kullanarak kuyruk yöneticisi, varsayılan değerler ve düzeyler seçilmesi

Kuyruk yöneticilerinizi denetlemek için ISPF ' de CSQOREXX exec olanağını kullanabilirsiniz.

İlk panoyu görüntülerken, herhangi bir kuyruk yöneticisine bağlanmadığınızı da göz atmanızı sağlar. Ancak, Enter tuşuna bastığınız anda, kuyruk yöneticisine ya da **Bağlantı adı** alanında belirtilen kuyruk paylaşım grubundaki bir kuyruk yöneticisine bağlandığınız gibi. Bu alanı boş bırakabilirsiniz; bu, toplu iş uygulamaları için varsayılan kuyruk yöneticisini kullandığınızı gösterir. Bu, CSQBDEFV ' de tanımlıdır (bu konuda bilgi için bkz. [Görev 19: Toplu İş Kümesi, TSO ve RRS bağdaştırıcıları](#)).

İstediğiniz işlemlerin gerçekleştirilmesini istediğiniz kuyruk yöneticisini belirtmek için **Hedef kuyruk yöneticisi** alanını kullanın. Bu alanı boş bırakırsanız, varsayılan değer olarak **Connect name** (Bağlantı adı) alanında belirlenen kuyruk yöneticisi kullanılır. Bağlanmadığınız bir hedef kuyruk yöneticisi belirtebilirsiniz. Bu durumda, olağan durumda bir kuyruk yöneticisi diğer adı tanımlaması sağlayan bir uzak kuyruk yöneticisi nesnesinin adını belirtirdiniz (komut giriş kuyruğunu açarken ad *ObjectQMgradı* olarak kullanılır). Bunu yapmak için, uzak kuyruk yöneticisine erişmek için uygun kuyrukların ve kanalların ayarlanmış olması gerekir.

The **İşlem kuyruğu yöneticisi** field allows you to specify a queue manager that is in the same queue sharing group as the queue manager specified in the **Hedef kuyruk yöneticisi** field to be the queue manager where the actions you request are to be performed. Bu alanda * belirtirseniz, istekte bulunduğunuz işlemler kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinde gerçekleştirilir. Bu alanı boş bırakırsanız, varsayılan değer olarak **Hedef kuyruk yöneticisi** alanında belirtilen değer kullanılır. **İşlem kuyruğu yöneticisi** alanı, MQSC komutlarında açıklanan CMDSCOPE komut değiştiricisini kullanmaya 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 isteklerde bulunduktan sonra ilk panoya döndüğünüzde, gerçek adlarla tamamlanan alanları bulursunuz.

Kuyruk yöneticisi düzeyleri

The Operations and Control panels work satisfactorily only with queue managers running on z/OS at IBM WebSphere MQ 7.10 or later.

Bu koşullar karşılanmazsa, eylemlerin yalnızca kısmen, yanlış ya da hiç olmadığı ve kuyruk yöneticisinden gelen yanıtların tanınmadığından büyük olasılıkla bu koşullar geçerli olur.

İşlem kuyruğu yöneticisi IBM MQ 8.0.0ya da üstünde değilse, bazı alanlar görüntülenmez ve bazı değerler girilemez. Birkaç nesne ve eylemde izin verilmez. Bu tür durumlarda, devam etmek istediğinizi doğrulamanızı isteyen ikincil bir pencere açılır.

z/OS Using the function keys and command line with the ISPF control panels on z/OS

Panoları kullanmak için, işlev tuşlarını kullanmanız ya da ISPF denetim panosu komut alanında eşdeğeri komutların girilmesi gerekir.

- [İşlev tuşları](#)
 - [İşlemlerinizin işlenmesi](#)
 - [“IBM MQ kullanıcı iletilerinin görüntülenmesi” sayfa 429](#)
 - [İşlemlerinizin iptal edilmesi](#)
 - [Yardım alma](#)
- [Komut satırının kullanılması](#)

İşlev tuşları

The function keys have special settings for IBM MQ. (This means that you cannot use the ISPF default values for the function keys; if you have previously used the KEYLIST OFF ISPF command anywhere, you must type KEYLIST ON in the command area of any operations and control panel and then press Enter to enable the IBM MQ settings.)

Bu işlev tuşu ayarları panolarda, Şekil 36 sayfa 430 içinde gösterildiği şekilde görüntülenebilir. Ayarlar gösterilmiyorsa, herhangi bir işlem ve denetim masasının komut alanına PFSHOW yazın ve **Enter** tuşuna basın. Ayarların görüntüsünü kaldırmak için PFSHOW OFF komutunu kullanın.

İşlem ve denetim panolarındaki işlev tuşu ayarları, CUA standartlarına uygun olmalıdır. Tuş ayarını olağan ISPF yordamlarıyla (**KEYLIST** yardımcı programı gibi) değiştirebilmeniz de, bunu yapmanız önerilmez.

Not: PFSHOW ve **KEYLIST** komutlarının kullanılması, sahip olduğunuz diğer tüm mantıksal ISPF ekranlarını etkiler ve bu ekranların ayarları, işlemleri ve denetim panolarını bıraktığınızda kalır.

İşlemlerinizin işlenmesi

Bir panelde istenen işlemi gerçekleştirmek için **Enter** tuşuna basın. Panodan alınan bilgiler kuyruk yöneticisine işlenmek üzere gönderilir.

Panolara her **Enter** tuşuna bastığınızda, IBM MQ bir ya da daha çok işletmen iletisi oluşturur. İşlem başarılı olursa, CSQ9022 Ionay iletisini alabilirsiniz, tersi durumda bazı hata iletileri alabilirsiniz.

IBM MQ kullanıcı iletilerinin görüntülenmesi

IBM MQ kullanıcı iletilerini görmek için herhangi bir panodaki F10 işlev tuşuna basın.

İşlemlerinizin iptal edilmesi

İlk panoda hem F3 , hem de F12 , işlemler ve denetim panolarından çıkın ve sizi ISPF ' ye geri döndürmenizi sağlar. Kuyruk yöneticisine bilgi gönderilmez.

On any other panel, press function keys F3 or F12 to leave the current panel **en son Enter tuşuna bastığından bu yana yazdığınız verileri yoksayma**. Yine, kuyruk yöneticisine bilgi gönderilmez.

- F3 düğmesi, sizi doğrudan ilk panoya götürür.
- F12 , sizi önceki panoya geri götürür.

Yardım alma

Her panelde, kendisiyle ilişkili yardım panoları bulunur. Yardım panoları ISPF protokollerini kullanır:

- Görevle ilgili genel yardımı (genişletilmiş yardım) görmek için herhangi bir panodaki F1 tuş işlevi tuşuna basın.
- Bu alana ilişkin belirli yardımı görmek için, imlecin bulunduğu tüm alanlardaki F1 tuşuna basın.
- Genel yardımı almak için herhangi bir alan yardımı panosundan F5 tuş işlevi tuşuna basın.
- Press function key F3 to return to the base panel, that is, the panel from which you pressed function key F1.
- İşlev tuşlarıyla ilgili yardım almak için herhangi bir yardım panosundan F6 işlev tuşuna basın.

Yardım bilgileri ikinci ya da sonraki sayfalarda devam ederse, panonun sağ üst bölümünde bir **Diğer** göstergesi görüntülenir. Yardım sayfaları arasında gezinmek için bu işlev tuşlarını kullanın:

- Sonraki yardım sayfasına (varsa) gitmek için F11 tuşuna basın.
- F10 , önceki yardım sayfasına (varsa) geri dönmek için.

Komut satırını kullanma

İşlev tuşlarından kullanılabilir olduğu için, işlemlerin ve denetim panolarının kullandığı komutları vermek için hiçbir zaman komut satırını kullanmanız gerekmez. Komut satırı, normal ISPF komutlarını (**PFSHOW** gibi) girmenize olanak tanıyacak şekilde sağlanır.

ISPF komutu PANELID ON , yürürlükteki CSQOREXX panosunun adını görüntüler.

Komut satırı başlangıçta, sahip olduğunuz ISPF ayarlarından bağımsız olarak, panoların alt kısmında varsayılan konumda görüntülenir. Komut satırının konumunu değiştirmek için, herhangi bir işlem ve denetim panosundan SETTINGS ISPF komutunu kullanabilirsiniz. Ayarlar, işlemler ve denetim panolarıyla sonraki oturumlar için hatırlanır.

z/OS z/OS üzerindeki işlemleri ve denetim panolarını kullanma

CSQOREXX 'ten görüntülenen ilk denetim panosunu araştırmak için bu konuyu kullanın.

Şekil 36 sayfa 430 , bir pano oturumunu başlattığınızda görüntülenen panoyu gösterir.

```
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 . . . . . MQ1C - local queue manager or group
Target queue manager . . . MQ1C
                        - connected or remote queue manager for command input
Action queue manager . . . MQ1C - 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
```

Şekil 36. IBM MQ işlemleri ve denetim ilk panosu

Bu panodan aşağıdaki 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 komut verilmesini 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 nesnelerin (belirtilen tipte) listesini görüntülemek için **Enter** tuşuna basın. Daha sonra sırayla çalışmak için bir ya da daha çok nesne seçebilirsiniz. Listedeki tüm işlemler kullanılabilir.

Not: Görüntülenmekte olan nesneler listesiyle sonuçlanan seçimleri yapmanız ve daha sonra bu listeden çalışmanız önerilir. Tüm nesne tipleri için izin verildiğinden **Görüntüle** işlemini kullanın.

z/OS z/OS üzerinde Command Facility 'nin kullanılması

Kuyruk yöneticisine geçirecek MQSC komutlarını girmek ya da sonda etmek için düzenleyiciyi kullanın.

From the primary panel, CSQOPRIA, select option **8 Komut**, to start the Command Facility.

Sıralı bir dosyanın (*örnek*.CSQUTIL.COMMANDS) bir düzenleme oturumuna sunulur; CSQUTIL COMMAND işlevine giriş olarak kullanılır; bkz. [IBM MQ komutlarına komut verilmesi](#).

Komut öneki dizgisi (CPF) ile komut örneklerine gerek yoktur.

You can continue MQSC commands on subsequent lines by terminating the current line with the continuation characters + or -. Alternatively, use line edit mode to provide long MQSC commands or the values of long attribute values within the command.

satır düzenleme

Satır düzenlemesini kullanmak için, imleci düzenleme panosunda uygun çizeye taşıyın ve kaydırılabilir bir panoda tek bir satır görüntülemek için **F4** seçeneğini kullanın. Tek bir hat en çok 32 760 bayt veri olabilir.

Satır düzenlemesini bırakmak için:

- **F3 çıkışı** , hat ve çıkışlarda yapılan değişiklikleri saklar
- **F12 cancel** (İptal) düğmesi, satırda yapılan değişiklikleri düzenleme panosuna döner.

Düzenleme oturumunda yapılan değişiklikleri atmak için, dosyanın içeriğini değiştirmeden bırakmak üzere düzenleme oturumunu sonlandırmak için **F12 iptal** seçeneğini kullanın. Komutlar yürütülmez.

Komutların yürütülmesi

MQSC komutlarını girmeyi tamamladığınızda, dosyanın içeriğini saklamak için **F3 çıkışı** ile düzenleme oturumunu sona erdirin ve komutları kuyruk yöneticisine aktarmak için CSQUTIL komutunu çağırın. Komut işlemindeki çıkış, *prefix*kütüğünde tutulur.CSQUTIL.OUTPUT. Yanıtları görüntüleyebileceğiniz bu dosyada otomatik olarak bir düzenleme oturumu açılır. Bu oturumdan çıkmak ve ana menüye dönmek için **F3 exit** düğmesine basın.

z/OS üzerindeki IBM MQ nesneleriyle çalışma

Bu belgede açıklanan görevlerin çoğu IBM MQ nesnelere işlem gerektirmektedir. Nesne tipleri kuyruk yöneticileridir, kuyruklar, süreç tanımlamaları, ad listeleri, kanallar, istemci bağlantı kanalları, dinleyiciler, hizmetler ve kimlik doğrulama bilgileri nesnelere.

- [Basit kuyruk nesnelere tanımlanması](#)
- [Diğer nesne tiplerinin tanımlanması](#)
- [Nesne tanımlamalarıyla çalışılması](#)
- [Ad listeleriyle çalışma](#)

Yalın kuyruk nesnelere tanımlanması

Yeni bir nesne tanımlamak için, temel olarak var olan bir tanımlamayı kullanın. Bunu şu üç yöntemden birinde yapabilirsiniz:

- İlk panoda seçilen seçeneklerin sonucu olarak görüntülenen bir listenin üyesi olan bir nesne seçilerek. Daha sonra işlem tipi 2 girin (**Beğen**) (Seçilen nesnenin yanındaki işlem alanında). Yeni nesneniz, yok etme dışında, seçilen nesnenin özniteliklerine sahiptir. Daha sonra, yeni nesnenizdeki tüm öznitelikleri istediğiniz gibi değiştirebilirsiniz.
- İlk panoda, **Benzer tanımla** işlem tipini seçin, **Nesne tipi** alanında tanımlamakta olduğunuz nesnenin tipini girin ve **Ad** alanına, var olan bir nesnenin adını girin. Yeni nesneniz, yok etme dışında, **Ad** alanında adlandırdığınız nesnenlerle aynı özniteliklere sahiptir. Bundan sonra, yeni nesne tanımınızdaki tüm öznitelikleri istediğiniz gibi değiştirebilirsiniz.
- **Bu gibi tanımla** işlem tipini seçerek, bir nesne tipi belirleyip **Ad** alanını boş bırakılarak bu alanı boş bırakılabilir. Daha sonra yeni nesnenizi tanımlayabilir ve kuruluşunuz için tanımlanmış varsayılan özniteliklere sahip olur. Bundan sonra, yeni nesne tanımınızdaki tüm öznitelikleri istediğiniz gibi değiştirebilirsiniz.

Not: İlk panoda tanımlamadığınız nesnenin adını girmez, ancak size sunulan **Tanımla** panelinde gösterilir.

Aşağıdaki örnek, şablon olarak var olan bir kuyruğu kullanarak yerel bir kuyruğun nasıl tanımlanmasını gösterir.

Yerel kuyruk tanımlanması

İşlemler ve denetim panolarından bir yerel kuyruk nesnesi tanımlamak için, yeni tanımlamanız için temel olarak var olan bir kuyruk tanımlamasını kullanın. Tamamlanacak birkaç pano var. Tüm

panoları tamamladığınızda, özniteliklerin doğru olduğundan emin olun, tanımınızı kuyruk yöneticisine göndermek için Enter tuşuna basın ve böylece gerçek kuyruğu oluşturur.

İlk panoda ya da ilk panoda seçilen seçeneklerin sonucu olarak görüntülenen bir listedeki bir nesne girdisine karşı **Like like** (Tanımlama) işlemini kullanın.

Örneğin, ilk panodan başlayarak şu alanları doldurun:

İşlem	2 (Tanımla)
Nesne tipi	QLOCAL
NAME	QUEUE.YOU.LIKE. Bu ad, yeni kuyruğunuza ilişkin öznitelikleri sağlayan kuyruğun adıdır.

Define a Local Queue (Yerel Kuyruk Tanımla) panosunu görüntülemek için Enter tuşuna basın Yeni kuyruğun adını sağlayabilmeniz için kuyruk adı alanı boş olur. Tanım, bu yeni tanımlamayı temel alan kuyruğun üzerinde yer alan bir kuyruktan. Bu alanda, yeni kuyruğa ilişkin kendi açıklamanızı içeren bir alan belirleyin.

Diğer alanlardaki değerler, yok etme dışında, bu yeni kuyruğun üzerine basmakta olduğunuz kuyruktan bu değerlerdir. Bu alanları istediğiniz gibi yazabilirsiniz. Örneğin, uygun olarak yetkili uygulamalar bu kuyruğa ileti koyabiliyorsa, **Etkin bırak** alanına Y yazın (önceden Y değilse).

Alan yardımı almak için imleci bir alana getirin ve işlev tuşuna F1 tuşuna basın. Alan yardımı, her öznitelik için kullanılacak değerler hakkında bilgi sağlar.

İlk panoyu tamamladığınızda, ikinci panoyu görüntülemek için işlev tuşuna F8 basın.

İpuçları:

1. Bu aşamada Enter tuşuna basmayın; tersi durumda, kalan alanları tamamlama şansınız bulunmadan önce kuyruk yaratılacaktır. (Önceden zamanından önce Enter tuşuna basarsanız, endişelenmeyin; tanımınızı her zaman üzerinde değiştirebilirsiniz.)
2. F3 ya da F12 işlev tuşlarına basmayın; ya da yazdığınız veriler kaybedilir.

Tetikleme tanımlaması, olay denetimi ve geri alma raporlama panoları da içinde olmak üzere, geri kalan panoları görmek ve tamamlamak için F8 tuşuna basın.

Yerel kuyruk tanımlamanızın tamamında

Tanımınız tamamlandığında, bilgileri işlemek üzere kuyruk yöneticisine göndermek için Enter tuşuna basın. Kuyruk yöneticisi, belirttiğiniz tanımlamaya göre kuyruğu yaratır. Kuyruğun yaratılmasını istemiyorsanız, çıkmak için F3 işlev tuşuna basın ve tanımlamayı iptal edin.

Diğer nesne tiplerinin tanımlanması

Diğer nesne tiplerini tanımlamak için, yeni tanımlamanız için temel olarak var olan bir tanımı kullanarak Yerel kuyruk tanımlanması' ta açıklandığı gibi kullanın.

İlk panoda ya da ilk panoda seçilen seçeneklerin sonucu olarak görüntülenen bir listedeki bir nesne girdisine karşı **Like like** (Tanımlama) işlemini kullanın.

Örneğin, ilk panodan başlayarak şu alanları doldurun:

İşlem	2 (Tanımla)
Nesne tipi	QALIAS, AD LISTESI, PROCESS, CHANNEL ve diğer kaynak nesnelere.
NAME	Boş bırakın ya da aynı tipte var olan bir nesnenin adını girin.

Karşılık gelen DEFE panolarını görüntülemek için Enter tuşuna basın. Alanları zorunlu olarak tamamlayın ve bilgileri kuyruk yöneticisine göndermek için yeniden Enter tuşuna basın.

Yerel bir kuyruk tanımlamak gibi, genel olarak başka bir nesne tipinin tanımlanması genellikle birkaç panelin tamamlanmasını gerektirir. Bir ad listesinin tanımlanması, "Ad listeleriyle çalışma" sayfa 433 içinde açıklandığı gibi ek bir çalışma gerektirir.

Nesne tanımlamalarıyla çalışılması

Bir nesne tanımlandığında, **İşlem** alanında bir işlem belirleyebilir, bunu değiştirebilir, görüntüleyebilir ya da yönetebilirsiniz.

Her durumda, aşağıdakilerden birini yapabilirsiniz:

- İlk panoda seçilen seçeneklerin sonucu olarak görüntülenen bir listeden çalışmak istediğiniz nesneyi seçin. Örneğin, nesneleri görüntülemek için **Eylem** alanına 1 , **Nesne tipi** alanında Queue ve **Ad** alanında * olarak girilirken, sistemde tanımlı olan tüm kuyrukların bir listesi görüntülenir. Bundan sonra, bu listeden, üzerinde çalışmak istediğiniz kuyruğu seçebilirsiniz.
- **Nesne tipi** ve **Ad** alanlarını doldurarak, çalışmakta olduğunuz nesneyi belirttiğiniz ilk panodan başlayın.

Nesne Tanımlamasının Değiştirilmesi

Bir nesne tanımlamasını değiştirmek için, işlem 3 değerini belirleyin ve ALTER (ALTER) panolarını görmek için Enter tuşuna basın. Bu paneller DEFE panellerine çok benzer. İsteddiğiniz değerleri değiştirebilirsiniz. Yaptığınız değişiklikler tamamlanınca, bilgileri kuyruk yöneticisine göndermek için Enter tuşuna basın.

Nesne Tanımlamasının Görüntülenmesi

Bir nesnenin ayrıntılarını değiştirmeden görmek istiyorsanız, işlem 1 'i belirleyin ve DISPLAY panolarını görmek için Enter tuşuna basın. Yine, bu panolar, alanların hiçbirini değiştiremeyimiz dışında, DEFE panolarına benzer. Başka bir nesnenin ayrıntılarını görüntülemek için nesne adını değiştirin.

Nesne silinmesi

Bir nesneyi silmek için, işlem 4 (Yönet) seçeneğini belirleyin ve **Sil** işlemi, sonuçtaki menüde sunulan eylemlerden biridir. **Sil** işlemi seçin.

İsteğinizi onaylamanız isteniyor. İşlev tuşuna F3 ya da F12 tuşuna basmak istiyorsanız, istek iptal edilir. Enter tuşuna bassanız, istek onaylanır ve kuyruk yöneticisine iletilir. Daha sonra, belirttiğiniz nesne silinir.

Not: Kanal başlatıcısı başlatılmadıkça, en çok kanal nesnesini silemezsiniz.

Ad listeleriyle çalışma

Ad listeleriyle çalışırken, diğer nesnelere için istediğiniz gibi devam edin.

DEFE LIKE ya da ALTER işlemleri için, listeye ad eklemek ya da listedeki adları değiştirmek için F11 işlev tuşuna basın. Bu, ISPF düzenleyicisiyle çalışmayı içerir ve tüm olağan ISPF düzenleme komutlarıyla kullanılabilir. Ad listesindeki her adı ayrı bir satıra girin.

ISPF düzenleyicisini bu şekilde kullandığınızda, işlev tuşu ayarları olağan ISPF ayarlarıdır ve diğer işlemler ve denetim panoları tarafından kullanılan **not** .

Listede küçük harfli adlar belirtmeniz gerekiyorsa, düzenleyici panosu komut satırında CAPS (OFF) belirleyin. Bunu yaparken, ileride düzenlediğiniz tüm ad listeleri CAPS (ON) belirtinceye kadar küçük harfle gösterilir.

Ad listesini düzenlemeyi tamamlayınca, ISPF düzenleme oturumunu sonlamak için F3 işlev tuşuna basın. Değişiklikleri kuyruk yöneticisine göndermek için Enter tuşuna basın.

Dikkat: Bu aşamada Enter tuşuna basmadıysanız, ancak F3 tuşuna basın; bunun yerine, yazdığınız güncellemelerden birini kaybedersiniz.

z/OS Birden çok küme iletim kuyruğu kullanılarak sistemin uygulanması

Kanal tek bir kümede mi, yoksa çakışan bir kümeden mi kullanılırsa, bu bir fark etmez. Kanal seçilip başlatıldığı zaman, kanal, tanımlara bağlı olarak iletim kuyruğunu seçer.

Yordam

- If you are using the DEFCLXQ option, see [“Kuyruklar ve anahtarlama otomatik tanımlamanın kullanılması”](#) sayfa 434.
- Aşamalı bir yaklaşım kullanıyorsanız, bkz. [“Aşamalı bir yaklaşımı kullanarak kümeli gönderici kanallarınızı değiştirme”](#) sayfa 434.

z/OS Kuyruklar ve anahtarlama otomatik tanımlamanın kullanılması

DEFCLXQ seçeneğini kullanmayı planlıyorsanız bu seçeneği kullanın. Her kanal için bir kuyruk yaratılacak ve her yeni kanal olacak.

Yordam

1. SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE (KUYRUK) ve gerekiyorsa öznitelikleri değiştirin.
Bu kuyruk SCSQPROC (csq4insx) üyesinde tanımlıdır.
2. SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE Model kuyruğu.
3. Bu model kuyruğu için güvenlik ilkeleri ve SYSTEM.CLUSTER.TRANSMIT. * * kuyruklar.
For z/OS the channel initiator started task user ID needs:

- For CLASS (MQADMIN) ile ilgili denetim erişimi

```
ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channelName
```

- Bu öge için CLASSE (MQQUEUE) erişimi güncelleme

```
ssid.SYSTEM.CLUSTER.TRANSMIT.channelName
```

z/OS Aşamalı bir yaklaşımı kullanarak kümeli gönderici kanallarınızı değiştirme

Aşamalı bir yaklaşımı kullanmayı planlıyorsanız bu seçeneği kullanın. Bu işlem, kuruluşunuzun gereksinimlerine uyacak şekilde, yeni küme gönderici kanallarına çeşitli zamanlarda taşınmanızı sağlar.

Başlamadan önce

- İş uygulamalarınızı ve hangi kanalların kullanıldığını belirleyin.
- Kullandığınız kuyruklar için, içinde oldukları kümeleri görüntüleyin.
- Bağlantı adlarını, uzak kuyruk yöneticilerinin adlarını ve kanal desteklerini gösteren kanalları görüntülemek için kanalları görüntüleyin.

Bu görev hakkında

- Bir iletim kuyruğu yaratın. z/OS üzerinde, kuyruk için hangi sayfa kümesini kullanabileceğinizi göz önünde bulunmayı isteyebilirsiniz.
- Kuyruk için güvenlik ilkesi ayarlayın.
- Kuyruk izlemesini bu kuyruk adını içerecek şekilde değiştirin.
- Bu iletim kuyruğunu hangi kanalların kullanacağına karar verin. Kanalların benzer bir adı olmalı, bu nedenle genel karakterler ' * ' CLCHNAME içinde kanalı tanıtır.
- Yeni işlevi kullanmaya hazır olduğunda, iletim kuyruğunu, bu iletim kuyruğunu kullanacak kanalların adını belirtebilmek için değiştirin. Örneğin, CLUSTER1.TOPARISya da CLUSTER1.* ya da *.TOPARIS

- Kanalları başlat

Yordam

1. Kümede tanımlı olan küme gönderen kanallarını görüntülemek için DIS CLUSQMGR(XXXX) XMITQ komutunu kullanın; burada XXXX , uzak kuyruk yöneticisinin adıdır.
2. İletim kuyruğu için güvenlik profilini ayarlayın ve kanal başlatıcısına kuyruk erişimi 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
```

Anahtar (SWITCH) komutunu kullanarak kullanıcı kimliği aşağıdaki erişime sahip olmalıdır:

```
alter cl(MQADMIN) ssid.QUEUE.queueName
```

4. Kanalları durdurup yeniden başlatın.

Kanal değişikliği, kanal bir MQSC komutu kullanmaya başladığında ya da CSQUTIL işlevini kullandığında gerçekleşir. CSQUTIL SWITCH CHANNEL (*) STATUS komutunu kullanarak hangi kanalların yeniden başlatılması gerektiğini belirleyebilirsiniz.

Kanal başlatıldığında sorunlarla karşı çıkarsanız, kanalı durdurun, sorunları çözer ve kanalı yeniden başlatın.

CLCHNAME özniteliğini istediğiniz sıklıkta değiştirebileceğiniz için dikkat edin.

Kullanılan CLCHNAME değeri, kanal başlatıldığında, CLCHNAME tanımını değiştirdiğinizde, kanal tanımlamaları başlatıldığı zamandan itibaren kullanmaya devam ederken, CLCHNAME tanımını değiştirebilirsiniz. Kanal yeni tanımlamayı yeniden başlatıldığında kullanır.

z/OS z/OS üzerinde yapılan değişikliği geri al

Sonuçlar beklediğiniz gibi değilse, bir değişikliği geri almak için bir süreçten geçmeniz gerekir.

Ne yanlış gidebilir ki?

Yeni iletim kuyruğu beklediğiniz gibi değilse:

1. CLCHNAME ' in beklediğiniz gibi olup olmadığını denetleyin
2. Anahtar işleminin tamamlıp bitirmediğini denetlemek için iş günlüğünü gözden geçirin. Değilse, bekleyin ve kanala ilişkin yeni iletim kuyruğunu daha sonra denetleyin.

Birden çok küme iletim kuyruğu kullanıyorsanız, iletim kuyrukları tanımlarını belirttik olarak tasarlama ve karmaşık çakışan yapılandırmalardan kaçınmanız önemlidir. Bu şekilde, sorunlar varsa, özgün kuyruklara ve yapılandırmaya geri dönebileceğinizden emin olabilirsiniz.

Farklı bir iletim kuyruğunu kullanmaya devam etme sırasında sorunlarla karşılaşırsanız, değişikliğe devam edebilmemiz için önce sorunları çözmeniz gerekir.

Yeni bir değişiklik isteğinin yapılabilmesi için önce var olan bir değişiklik isteği tamamlanmalıdır. Örneğin:

1. Derinlik üst sınırı olan yeni bir iletim kuyruğu tanımlayın ve gönderilmek üzere bekleyen 10 ileti vardır.
2. CLCHNAME parametresindeki kanal adını belirtmek için iletim kuyruğunu değiştirin.
3. Kanalı durdurun ve yeniden başlatın. İletilerin taşınma girişimi başarısız olur ve sorunları bildirir.
4. İletim kuyruğunda CLCHNAME parametresini boş olacak şekilde değiştirin.
5. Kanalı durdurun ve yeniden başlatın. Kanal, orijinal isteği tamamlamaya devam eder ve kanal, yeni iletim kuyruğunu kullanmaya devam eder.

6. İletilerin taşınması başarılı bir şekilde tamamlanırsa, sorunları çözmeniz ve kanalı yeniden başlatmanız gerekir.

Kanal yeniden başlatıldığında, herhangi bir değişiklik alır. Bu nedenle, CLCHNAME ' i boşluklara ayarladıysanız, kanal belirtilen iletim kuyruğunu kullanmaz.

Bu örnekte, iletim kuyruğunda CLCHNAME 'ın boşluklara değiştirilmesi, kanalın SYSTEM.CLUSTER.TRANSMIT kuyruğu, CLCHNAME parametresindeki kanal adıyla eşleşen başka iletim kuyrukları olabilir. Örneğin, bir soysal ad ya da DEFCLXQ kuyruk yöneticisi özniteliği kanal olarak ayarlanabileceği için, kanal SYSTEM.CLUSTER.TRANSMIT kuyruğu.

z/OS

Writing programs to administer IBM MQ for z/OS

Bir kuyruk yöneticisini yönetmek için kendi uygulama programlarınızı yazabilirsiniz. Kendi yönetim programlarınızı yazmaya ilişkin gereksinimleri anlamak için bu konuyu kullanın.

Genel kullanıma hazır programlama arabirimi bilgileri

Bu konular kümesi, bir IBM MQ uygulama programından IBM MQ komutlarını çıkarmanıza olanak sağlamak için ipuçları ve kılavuz bilgiler içerir.

Not: Bu konuda, MQI çağruları C dili gösterimi kullanılarak tanımlanır. COBOL, PL/I ve çevirici dillerindeki çağrılara ilişkin tipik çağrılar için [İşlev çağruları](#) el kitabına bakın.

Her şeyin nasıl işlediğini anlamak

Anahatlarda, bir uygulama programından komut verme yordamı şu şekildedir:

1. Build an IBM MQ command into a type of IBM MQ message called a *istek iletisi*. Komut MQSC ya da PCF biçiminde olabilir.
2. Gönder (MQPUT kullan) bu ileti, sistem komut giriş kuyruğu adı verilen özel bir kuyruğa iletilsin. The IBM MQ command processor runs the command.
3. Al (MQGET kullanın) Yanıtlama kuyruğunda *yanıt iletileri* olarak belirtilen komutun sonuçları. Bu iletiler, komutunuzun başarılı olup olmadığını ve olup olmadığını, sonuçların ne olduğunu belirlemek için gereken kullanıcı iletilerini içerir.

Daha sonra, sonuçları işlemek için uygulama programınıza bağlı olur.

Bu konu kümesi şunları içerir:

z/OS

Denetim programları için kuyruklar hazırlanıyor

Denetim programları, sistem komut girişi ve alma yanıtları için önceden tanımlanmış kuyruklar gerektirir.

Bu bölüm, MQSC biçimindeki komutlar için geçerlidir. PCF ' ye eşdeğer bir değer için bkz. ["IBM MQ Programları Komut Biçimlerinin Kullanılması"](#) sayfa 24.

Herhangi bir MQPUT ya da MQGET çağrısını yayınlamadan önce, önce kullanacağınız kuyrukları tanımlamanız ve daha sonra açmanız gerekir.

Sistemin tanımlanması-komut giriş kuyruğu

Sistem komutu giriş kuyruğu, SYSTEM.COMMAND.INPUT. Sağlanan CSQINP2 başlatma verileri kümesi, thlqual.SCSQPROC(CSQ4INSG), sistem komutu giriş kuyruğu için varsayılan bir tanımlama içerir. Diğer altyapılarda IBM MQ ile uyumluluk için, bu kuyruğun diğer adı (SYSTEM.ADMIN.COMMAND.QUEUE (Kuyruk) da sağlanır. Ek bilgi için [Sample definitions supplied with IBM MQ](#) başlıklı konuya bakın.

Yanıtlama Kuyruğu Tanımlama

IBM MQ komut işlemcisinden yanıt iletilerini almak için bir yanıtlama kuyruğu tanımlamanız gerekir. Bu, yanıtlama iletilerinin üzerine yerleştirilmesine izin veren özniteliklerle herhangi bir kuyruk olabilir. Ancak, olağan işletim için bu öznitelikleri belirtin:

- KULLANIM (NORMAL)
- NOTRIGGER (uygulamanız tetikleme kullanmadığı sürece)

Komutlar için kalıcı iletiler kullanmaktan kaçının, ancak bunu yapmayı seçerseniz, yanıtlama kuyruğu geçici bir dinamik kuyruk olmamalıdır.

Sağlanan CSQINP2 başlatma verileri kümesi thlqual.SCSQPROC(CSQ4INSG), SYSTEM.COMMAND.REPLY.MODEL. Dinamik bir yanıtlama kuyruğu oluşturmak için bu modeli kullanabilirsiniz.

Not: Komut işlemcisi tarafından oluşturulan yanıtlar en çok 15 000 bayt uzunluğunda olabilir.

Yanıtlama kuyruğu olarak kalıcı bir dinamik kuyruk kullanırsanız, uygulamanızın kuyruğu silme girişiminde bulunmadan önce, tüm PUT ve GET işlemlerinin tamamlanmasına izin vermesi gerekir; tersi durumda, MQRC2055 (MQRC_Q_NOT_EMPTY) geri döndürülebilir. Bu durumda, birkaç saniye sonra kuyruk silme işlemini yeniden deneyin.

Sistem komutu giriş kuyruğunun açılması

Sistem komutu giriş kuyruğunu açabilmeniz için, uygulama programınızın kuyruk yöneticinize bağlı olması gerekir. Bunu yapmak için MQI çağrısını MQCONN ya da MQCONNX kullanın.

Daha sonra, sistem komutu giriş kuyruğunu açmak için MQI çağrısını MQOPEN kullanın. Bu aramayı kullanmak için:

1. **Options** parametresini MQOO_OUTPUT olarak ayarlayın
2. MQOD nesne tanımlayıcı alanlarını aşağıdaki gibi ayarlayın:

ObjectType

MQOT_Q (nesne bir kuyruktur)

ObjectName

SYSTEM.COMMAND.INPUT

ObjectQMgrName

İstek iletilerinizi yerel kuyruk yöneticinize göndermek istiyorsanız, bu alanı boş bırakın. Bu, komutlarınızın yerel olarak işlendiği anlamına gelir.

IBM MQ komutlarınızın uzak bir kuyruk yöneticisiyle işlenmesini istiyorsanız, bu komutların adını buraya yazın. Ayrıca, Dağıtılmış kuyruklama ve kümelerde açıklandığı gibi, doğru kuyruklar ve bağlantılar kurmalısınız.

Yanıtlama Kuyruğu Açma

Yanıtları bir IBM MQ komutundan almak için, bir yanıt kuyruğu açmanız gerekir. Bunu yapmanın bir yolu, yanıt kuyruğu olarak kalıcı bir dinamik kuyruk oluşturmak için MQOPEN çağrısındaki SYSTEM.COMMAND.REPLY.MODEL model kuyruğunu belirtmeyi sağlar. Bu aramayı kullanmak için:

1. **Options** parametresini MQOO_INPUT_SHARED olarak ayarlayın
2. MQOD nesne tanımlayıcı alanlarını aşağıdaki gibi ayarlayın:

ObjectType

MQOT_Q (nesne bir kuyruktur)

ObjectName

Yanıtın gönderileceği kuyruğun adı. Belirttiğiniz kuyruk adı bir model kuyruğu nesnesinin adıysa, kuyruk yöneticisi dinamik bir kuyruk yaratır.

ObjectQMgrName

Yerel kuyruk yöneticinizin yanıtlarını almak için bu alanı boş bırakın.

Dynami.cQName

Yaratılacak dinamik kuyruğun adını belirleyin.

Komut sunucusunun kullanılması

Komut sunucusu, komut işlemcisi bileşeniyle çalışan bir IBM MQ bileşenidir. İletileri yorumlayan, denetim isteklerini çalıştıran ve yönetim uygulamanıza yanıt gönderen komut sunucusuna biçimlendirilmiş iletiler gönderebilirsiniz.

Komut sunucusu, sistem komut giriş kuyruğundan istek iletilerini okur, bunları doğrular ve geçerli olanları komut işlemcisine komut olarak geçirir. Komut işlemcisi komutları işler ve tüm yanıtları, belirttiğiniz yanıtlama kuyruğuna yanıt iletileri olarak yerleştirir. İlk yanıt iletisi, CSQN205I kullanıcı iletisini içerir. Ek bilgi için [“Yanıt iletilerinin komut sunucusundan yorumlanması” sayfa 441](#) başlıklı konuya bakın. Komut sunucusu ayrıca, kanal başlatıcı ve kuyruk paylaşım grubu komutlarını, bunların yayınlandıkları yerde işler.

Komutlarınızı işleyen kuyruk yöneticisinin tanımlanması

Bir denetim programından çıkardığınız komutları işleyen kuyruk yöneticisi, iletinin yerleştiği sistem komutu giriş kuyruğunun sahibi olan kuyruk yöneticidir.

Komut sunucusu başlatılıyor

Genellikle, kuyruk yöneticisi başlatıldığında komut sunucusu otomatik olarak başlatılır. It becomes available as soon as the message CSQ9022I 'START QMGR' NORMAL COMPLETION is returned from the START QMGR command. Sistem sonlandırma aşaması sırasında bağlı tüm görevler kesildiğinde, komut sunucusu durdurulur.

START CMDSERV komutunu kullanarak komut sunucusunu kendiniz denetleyebilir ve CMDSERV komutlarının STOP CMDSERV komutlarını durdurabilirsiniz. IBM MQ yeniden başlatıldığında komut sunucusunun otomatik olarak başlatılmasını önlemek için, CSQINP1 ya da CSQINP2 kullanıma hazırlama veri kümelerinize STOP CMDSERV komutunu ekleyebilirsiniz. Ancak, kanal başlatıcı ya da kuyruk paylaşım grubu komutlarının işlenmesini önlediği için bu önerilmez.

STOP CMDSERV komutu, yürürlükteki iletiyi işlemeyi bitirir bitirmez komut sunucusunu durdurur ya da herhangi bir ileti işlenmiyorsa hemen sona erir.

Komut sunucusu programda STOP CMDSERV komutu tarafından durdurulduysa, programdan başka komut işlenemez. Komut sunucusunu yeniden başlatmak için, z/OS konsolundan bir START CMDSERV komutu vermelisiniz.

Kuyruk yöneticisi çalışırken komut sunucusunu durdurup yeniden başlatırsanız, komut sunucusu durduğunda komut sunucusu giriş kuyruğunda bulunan tüm iletiler, komut sunucusu yeniden başlatıldığında işlenir. Ancak, komut sunucusu durdurulduktan sonra kuyruk yöneticisini durdurup yeniden başlatırsanız, komut sunucusu yeniden başlatıldığında yalnızca, sistem komutu giriş kuyruğunda bulunan kalıcı iletiler işlenir. Sistem komutu giriş kuyruğunda kalıcı olmayan tüm iletiler kaybedilir.

Komut Sunucusuna Komutlar Gönderilmesi

Her komut için, komutu içeren bir ileti oluşturur ve bu iletiyi sistem komut giriş kuyruğuna yerleştirdiniz.

IBM MQ komutlarını içeren bir ileti oluşturma

You can incorporate IBM MQ commands in an application program by building request messages that include the required commands. Bu tür komutların her biri için:

1. Komutu temsil eden bir karakter dizilimini içeren bir arabellek yaratın.
2. Çağrıya ilişkin **buffer** parametresindeki arabellek adını belirterek bir MQPUT çağrısı yayınlayın.

C 'de bunu yapmanın en basit yolu, 'char' kullanarak bir arabellek tanımlamanız. Örneğin:

```
char message_buffer[ ] = "ALTER QLOCAL(SALES) PUT(ENABLED)";
```

Bir komut oluştururken, boş karakterle biten bir karakter dizgisi kullanın. Bu şekilde tanımlanmış bir komutun başında bir komut öneki dizgisi (CPF) belirtmeyin. Bu, başka bir kuyruk yöneticisinde çalıştırmak isterseniz komut komut dosyalarınızı değiştirmek zorunda kalmayacağz anlamına gelir. Ancak, bir CPF ' nin yanıt kuyruğuna yerleştirilecek yanıt iletilerine eklendiğini dikkate almalısınız.

Komut sunucusu, tırnak işareti içine alınmadıkça, tüm küçük harfleri büyük harfe çevirir.

Komutlar en çok 32 762 karakter uzunluğunda olabilir.

Sistem komutu giriş kuyruğuna ileti konması

Sistem komutu giriş kuyruğuna komut içeren istek iletilerini koymak için MQPUT çağrısını kullanın. Bu çağrıda, önceden açtığınız yanıtlayıcı kuyruğunun adını belirtmenizi sağlar.

MQPUT çağrısını kullanmak için:

1. Bu MQPUT parametrelerini ayarlayın:

Hconn

MQCONN ya da MQCONNX çağrısının döndürdüğü bağlantı tanıtıcısı.

Hobj

Sistem komutu giriş kuyruğu için MQOPEN çağrısının döndürdüğü nesne tanıtıcısı.

BufferLength

Biçimlendirilmiş komutun uzunluğu.

Buffer

Komutu içeren arabelleğin adı.

2. Bu MQMD alanlarını ayarla:

MsgType

MQMT_REQUEST

Format

MQFMT_STRING ya da MQFMT_NONE

If you are not using the same code page as the queue manager, set *CodedCharSetId* as appropriate and set MQFMT_STRING, so that the command server can convert the message. Komutunuzun PCF olarak yorumlanmasına neden olduğu için MQFMT_ADMIN ' i ayarlamayın.

ReplyToQ

Yanıtlayıcı kuyruğunun adı.

ReplyToQMGr

Yanıtların yerel kuyruk yöneticinize gönderilmesini istiyorsanız bu alanı boş bırakın. IBM MQ komutlarınızın uzak bir kuyruk yöneticisine gönderilmesini istiyorsanız, bu komutların adını buraya yazın. Ayrıca, Dağıtılmış kuyruklama ve kümelerde açıklandığı gibi, doğru kuyruklar ve bağlantılar kurmalısınız.

3. Diğer herhangi bir MQMD alanını gerektiği şekilde ayarlayın. Normalde komutlar için kalıcı olmayan iletiler kullanmalısınız.
4. Herhangi bir *PutMsgOpts* seçeneğini gerektiği şekilde ayarlayın.

MQPMO_SYNCPOINT (varsayılan değer) değerini belirlerseniz, bir eşitleme noktası çağrısıyla MQPUT çağrısını izlemeniz gerekir.

MQPUT1 ve sistem komutu giriş kuyruğunu kullanma

Sistem komut girişi kuyruğuna yalnızca bir ileti koymak istiyorsanız, **MQPUT1** çağrısını kullanabilirsiniz. This call combines the functions of an **MQOPEN**, followed by an **MQPUT** of one message, followed by an

MQCLOSE, all in one call. Bu çağrıyı kullanırsanız, parametreleri uygun şekilde değiştirin. Ayrıntılar için bkz. [MQPUT1 çağrısını kullanarak bir ileti kuyruğa konması](#) .

Komutlarınıza yanıt alma

Komut sunucusu, aldığı her istek ileti için bir yanıt kuyruğuna yanıt gönderir. Herhangi bir denetim uygulamasının yanıt iletilerini alması ve yanıtlaması gerekir.

Komut işlemcisi komutlarınızı işlediğinde, MQPUT çağrısında belirtilen yanıtlama kuyruğuna herhangi bir yanıt ileti konmaya başlanır. Komut sunucusu yanıt iletilerini, aldığı komut iletiyle aynı kalıcılık ile gönderir.

Yanıt bekleniyor

İstek iletinizden bir yanıt almak için MQGET çağrısını kullanın. Bir istek ileti birkaç yanıt ileti üretebilir. Ayrıntılar için bkz. [“Yanıt iletilerinin komut sunucusundan yorumlanması” sayfa 441](#).

Bir MQGET çağrısının, yanıt ileti oluşturulmasını bekleyeceği bir zaman aralığı belirtebilirsiniz. Bir yanıt almezseniz, [“Yanıt alamazsanız” sayfa 442](#) başlıklı konuda başlayan denetim listesini kullanın.

MQGET çağrısını kullanmak için:

1. Bu değiştireleri ayarlayın:

Hconn

MQCONN ya da MQCONNX çağrısının döndürdüğü bağlantı tanıtıcısı.

Hobj

Yanıtlama kuyruğu için MQOPEN çağrısının döndürdüğü nesne tanıtıcısı.

Buffer

Yanıtın gönderileceği alanın adı.

BufferLength

Yanıtın gönderileceği arabellek uzunluğu. Bu, en az 80 bayt olmalıdır.

2. Verdiğiniz komuttan yalnızca verdiğiniz yanıtları aldığınızdan emin olmak için, uygun *MsgId* ve *CorrelId* alanlarını belirtmeniz gerekir. Bu, MQPUT çağrısında belirttiğiniz rapor seçeneklerine, MQMD_REPORT ' a bağlıdır:

MQRO_NONE

İkili sıfır, '00 ... 00' (24 boş değer).

MQRO_NEW_MSG_ID

İkili sıfır, '00 ... 00' (24 boş değer).

Bu seçenek, bu seçeneklerin hiçbiri belirtilmediyse varsayılan değerdir.

MQRO_PASS_MSG_ID

The *MsgId* from the MQPUT.

MQRO_NONE

MQPUT çağrısından *MsgId* .

MQRO_COPY_MSG_ID_TO_CORREL_ID

MQPUT çağrısından *MsgId* .

Bu seçenek, bu seçeneklerin hiçbiri belirtilmediyse varsayılan değerdir.

MQRO_PASS_COREL_ID

MQPUT çağrısından *CorrelId* .

Rapor seçenekleriyle ilgili daha fazla ayrıntı için bkz. [Rapor seçenekleri ve ileti işaretleri](#).

3. Aşağıdaki *GetMsgOpts* alanlarını ayarlayın:

Options

MQGMO_BEKLE

If you are not using the same code page as the queue manager, set `MQGMO_CONVERT`, and set `CodedCharSetId` as appropriate in the MQMD.

WaitInterval

Yerel kuyruk yöneticisinden gelen yanıtlar için 5 saniye deneyin. Milisaniye olarak kodlanmış, bu 5 000 olur. Uzak kuyruk yöneticisinden ve kanal denetimi ve durum komutlarından gelen yanıtlar için 30 saniye deneyin. Milisaniye olarak kodlanmış, bu 30 000 olur.

Atılan iletiler

Komut sunucusu bir istek iletisinin geçerli olmadığını bulursa, bu iletiyi atar ve `CSQN205I` iletisini adlandırılmış yanıtlama kuyruğuna yazar. Bir yanıtlama kuyruğu yoksa, `CSQN205I` iletisi ölü-mektup kuyruğuna konabilir. Bu iletteki dönüş kodu, özgün istek iletisinin neden geçerli olmadığını gösterir:

00D5020F	Bu tip MQMT_REQUEST tipinde değil.
00D50210	Uzunluğu sıfır.
00D50212	32 762 bayttan daha uzun.
00D50211	Tüm boşlukları içerir.
00D5483E	Dönüştürme 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_YOK
<i>Encoding</i>	MQENC_NATIVE
<i>Priority</i>	Yayınladığınız iletteki MQMD ' ye ilişkin olarak.
<i>Persistence</i>	Yayınladığınız iletteki MQMD ' ye ilişkin olarak.
<i>CorrelId</i>	MQPUT rapor seçeneklerine bağlıdır.
<i>ReplyToQ</i>	Yok.

Komut sunucusu, MQPMO yapısının *Options* alanını MQPMO_NO_SYNCPOINT olarak ayarlar. Bu, yanıtları bir sonraki eşitleme noktasında bir grup olarak değil, yaratıldığı için alabileceğiniz anlamına gelir.

z/OS Yanıt iletilerinin komut sunucusundan yorumlanması

IBM MQ tarafından doğru bir şekilde işlenen her istek iletisi en az iki yanıt iletisi üretir. Her yanıt iletisi tek bir IBM MQ kullanıcı iletisi içerir.

Yanıtın uzunluğu, çıkılan komuta bağlıdır. Alabileceğiniz en uzun yanıt bir DISPLAY NAMELIST ' ten (DISPLAY NAMELIST) ve bu da 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). Sayım, bu ilk iletiyi içerir.
- Komut ön işlemcisinden dönüş kodu.
- Komut işlemcisinden neden kodu olan bir neden kodu.

Bu ileti CPF içermiyor.

Örneğin:

```
CSQN205I    COUNT=    4, RETURN=0000000C, REASON=00000008
```

COUNT alanı 8 bayt uzunluğunda ve sağa yaslanmış. Her zaman, COUNT=konumundan hemen sonra 18 konumunda başlar. RETURN (RETURN) alanı, karakter onaltılık alanında 8 bayt uzunluğunda ve 35. konumdaki RETURN= ' dan hemen sonra. NESANEY alanı, karakter onaltılı değeri uzunluğundaki 8 byte 'tır ve 52 konumunda REASON= ' dan hemen sonra olur.

RETURN= değeri 00000000 ise ve REASON= değeri 00000004 ise, yanıt iletileri kümesi tamamlanmamış olur. CSQN205I iletisi tarafından belirtilen yanıtları aldıktan sonra, başka bir yanıt kümesi beklemek için daha fazla bir MQGET çağrısı yayınlayın. Sonraki yanıt kümesinde ilk ileti yeniden CSQN205I olur, bu da kaç yanıt olduğunu ve daha fazla yanıt olup olmadığını gösterir.

Tek tek iletilere ilişkin ek bilgi için [IBM MQ for z/OS iletileri, tamamlama, ve neden kodları belgelerine](#) bakın.

İngilizce dışındaki bir dil özelliği kullanıyorsanız, yanıtların metni ve yerleşim düzeni burada gösterilenlerden farklıdır. However, the size and position of the count and return codes in message CSQN205I are the same.

Yanıt alamazsanız

Komut sunucusuna bir yanıt almak için yanıt almezseniz, atabileceğiniz bir dizi adım vardır.

İstek iletinize bir yanıt almezseniz, bu denetim listesi aracılığıyla çalışın:

- Komut sunucusu çalışıyor mu?
- *WaitInterval* yeterince uzun mu?
- Sistem komut girişi ve yanıtlama kuyrukları doğru olarak tanımlıyor mu?
- Bu kuyruklara MQOPEN çağrıları başarılı oldu mu?
- Hem sistem komut girişi, hem de yanıt kuyrukları, MQPUT ve MQGET çağrıları için etkinleştirilmiştir.
- Kuyruklarınızın MAXDEPTH ve MAXMSGL özniteliklerinin artırımı olduğunu düşündünüz mü?
- *CorrelId* ve *MsgId* alanlarını doğru mu kullanıyorsunuz?
- Kuyruk yöneticisi hala çalışıyor mu?
- Komut doğru yapıldı mı?
- Tüm uzak bağlarınız doğru tanımlanmış ve çalışıyor mu?
- MQPUT çağrıları doğru olarak tanımlandı mı?
- Yanıtlama kuyruğu kalıcı dinamik kuyruk yerine geçici bir dinamik kuyruk olarak tanımlandı mı? (İstek iletisi kalıcıysa, yanıt için kalıcı bir dinamik kuyruk kullanmanız gerekir.)

Komut sunucusu yanıt oluşturduğunda, ancak bunları belirttiğiniz yanıtlama kuyruğuna yazamıyorsa, yanıtlar, bunları ölü harf kuyruğuna yazar.

MGCRE kullanarak komut geçirme

Uygun yetkilendirmeye, bir uygulama programı bir z/OS hizmet yordamını kullanarak birden çok kuyruk yöneticisine istekte bulunabilir.

Doğru yetkiniz varsa, programınızdan IBM MQ komutlarını MGCRE (SVC 34) z/OS hizmeti tarafından birden çok kuyruk yöneticisine iletebilirsiniz. CPF değeri, komutun yönlendirdiği belirli kuyruk yöneticisini tanıtır. CPF ' ler hakkında bilgi için bkz. [Komut güvenliği ve komut kaynağı güvenliği için kullanıcı kimlikleri ve "z/OS üzerinde kuyruk yöneticisi komutları yayınlanıyor" sayfa 416.](#)

MGCRE kullanırsanız, komutta doğrudan yanıtları almak için bir Komut ve Yanıt Simgesi (CART) kullanabilirsiniz.

z/OS Komutlara ve yanıtlarına ilişkin örnekler

Bu konuyu, komut sunucusuna ve komut sunucusundan gelen yanıtlara ilişkin bir dizi örnek olarak kullanın.

Burada, IBM MQ iletilerine oluşturulabilen bazı komutlar ve yanıtlar olan kullanıcı iletileri yer alıyor. Aksi belirtilmediği sürece, yanıtın her satırı ayrı bir iletidir.

- DEFINE KOMUTUNDAN GELEN İLETİLER
- DELETE komutundan iletiler
- DISABLE komutlarındaki iletiler
- CMDSCOPE ile komutlardan ileti sayısı
- CMDSCOPE ile komut oluşturan komutlardan ileti sayısı

DEFE komutundan gelen iletiler

Komut:

```
DEFINE QLOCAL(Q1)
```

bu iletileri üretir:

```
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000  
CSQ9022I +CSQ1 CSQMMSGP ' DEFINE QLOCAL ' NORMAL COMPLETION
```

Bu yanıt iletileri olağan bir şekilde tamamlanır.

DELETE komutundan gelen iletiler

Komut:

```
DELETE QLOCAL(Q2)
```

bu iletileri üretir:

```
CSQN205I COUNT= 4, RETURN=00000000, 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.

DISABLE komutlarından gelen iletiler

Aşağıdaki örneklerde, bazı DISPLAY komutlarının yanıtları gösterilmektedir.

Ölü-mektup kuyruğunun adını bulma

Bir kuyruk yöneticisi için ölü harf kuyruğunun adını öğrenmek istiyorsanız, bu komutu bir uygulama programından çalıştırın:

```
DISPLAY QMGR DEADQ
```

Gerekli adı çıkarabileceğiniz şu 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 komuttaki sonuçların, o komutta belirtilen özniteliklere nasıl bağımlı olduğu gösterilmektedir.

Örnek 1

Şu komutu kullanarak yerel bir kuyruk tanımlanınız:

```
DEFINE QLOCAL(Q1) DESCR('A sample queue') GET(ENABLED) SHARE
```

Bir uygulama programından aşağıdaki komutu yayınlayın:

```
DISPLAY QUEUE(Q1) SHARE GET DESCR
```

bu üç kullanıcı iletisi döndürülür:

```
CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000  
CSQM401I +CSQ1 QUEUE(Q1  
QLOCAL ) QSGDISP(QMGR )  
DESCR(A sample queue  
) SHARE GET(ENABLED )  
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION
```

Not: İkinci ileti olan CSQM401I, burada dört satırı kaplayan bir ileti olarak gösterilir.

Örnek 2

İki kuyrukta A harfiyle başlayan adlar vardır:

- A1 , PUT özniteliği DISABLE değerine ayarlanmış bir yerel kuyruktır.
- A2 , PUT özniteliği ENABLED (ETKİN) değerine ayarlanmış bir uzak kuyruktır.

Bir uygulama programından aşağıdaki komutu yayınlayın:

```
DISPLAY QUEUE(A*) PUT
```

bu dört kullanıcı iletisi 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 satır kaplayan iletiler gösterilir.

DISPLAY NAMELIST komutundan gelen iletiler

Aşağıdaki komutu kullanarak bir ad listesi tanımlırsınız:

```
DEFINE NAMELIST(N1) NAMES(Q1,SAMPLE_QUEUE)
```

Bir uygulama programından aşağıdaki komutu yayınlayın:

```
DISPLAY NAMELIST(N1) NAMES NAMCOUNT
```

Aşağıdaki üç kullanıcı ileti döndürülür:

```
CSQN205I  COUNT= 3, RETURN=00000000, REASON=00000000
CSQM407I +CSQ1 NAMELIST(N1 ) QS
GDISP(QMGR ) NAMCOUNT( 2) NAMES(Q1
,SAMPLE_QUEUE )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY NAMELIST' NORMAL COMPLETION
```

Not: İkinci ileti olan CSQM407I, burada üç satırı kaplayan bir ileti olarak gösterilir.

CMDScope komutuna sahip komutlardan gelen iletiler

Aşağıdaki örneklerde, CMDScope öznitelikle girilen komutlardan gelen yanıtlar gösterilir.

ALTER PROCESS komutundan gelen iletiler

Komut:

```
ALT PRO(V4) CMDScope(*)
```

aşağıdaki iletileri 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=0000000C, REASON=00000008
CSQN123E !MQ25 'ALT PRO' command for CMDSCOPE(*) abnormal completion
```

Bu iletiler, komutun kuyruk yöneticisi MQ25 ' e girilmiş ve iki kuyruk yöneticisine (MQ25 ve MQ26) gönderildiğini söyler. MQ25 üzerinde komut başarılı oldu, ancak süreç tanımlaması MQ26 üzerinde yok, bu nedenle komut o kuyruk yöneticisinde başarısız oldu.

DISPLAY PROCESS komutundan gelen iletiler

Komut:

```
DIS PRO(V*) CMDSCOPE(*)
```

aşağıdaki iletileri 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 kuyruk yöneticisi MQ25 ' e girilmiş ve iki kuyruk yöneticisine (MQ25 ve MQ26) gönderildiğini söyler. Her bir kuyruk yöneticisindeki tüm süreçlerle ilgili bilgiler, V. harfiyle başlayan adlarla ilgili olarak görüntülenir.

DISPLAY CHSTATUS komutundan gelen 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 kuyruk yöneticisi MQ25 ' e girilmiş ve iki kuyruk yöneticisine (MQ25 ve MQ26) gönderildiğini söyler. Her kuyruk yöneticisinde kanal durumu hakkında bilgi görüntülenir.

STOP CHANNEL komutundan gelen iletiler

Komut:

```
STOP CHL(VT) CMDSCOPE(*)
```

bu 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
CSQN205I 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 kuyruk yöneticisi MQ25 ' e girilmiş ve iki kuyruk yöneticisine (MQ25 ve MQ26) gönderildiğini söyler. Kanal VT her kuyruk yöneticisinde durduruldu.

CMDSCOPE ile komut oluşturan komutlardan gelen iletiler

Komut:

```
DEF PRO(V2) QSGDISP(GROUP)
```

bu 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 kuyruk yöneticisine (MQ25) girilmiş olduğunu söyler. Nesne paylaşılan havuzda yaratıldığında, başka bir komut üretildi ve kuyruk paylaşım grubundaki tüm etkin kuyruk yöneticilerine (MQ25 ve MQ26) gönderilir.

z/OS

z/OS üzerindeki IBM MQ kaynaklarını yönetme

IBM MQ for z/OS tarafından kullanılan kaynakların nasıl yönetileceğini öğrenmek için bu konudaki bağlantıları kullanın; örneğin, günlük dosyalarını, veri kümelerini, sayfa kümelerini, arabellek havuzlarını ve bağlaşım tesisi yapılarını yönetme.

IBM MQ for z/OS kullanırken tamamlamanız gereken farklı yönetim görevlerine ilişkin ayrıntılar için aşağıdaki bağlantıları kullanın:

- [“Günlüklerin yönetilmesi” sayfa 449](#)
- [“Önyükleme veri kümesinin yönetilmesi \(BSDS\)” sayfa 457](#)
- [“Sayfa kümelerini yönetme” sayfa 465](#)
- [“Sayfa kümelerini nasıl yedeklemeniz ve kurtarmanız” sayfa 471](#)
- [“CSQUTIL kullanarak kuyrukları yedekleme ve geri yükleme” sayfa 475](#)
- [“Arabellek havuzlarının yönetilmesi” sayfa 475](#)
- [“Managing queue sharing groups and shared queues on z/OS” sayfa 477](#)

İlgili kavramlar

IBM MQ for z/OS kavramlar

[“Yönetme IBM MQ for z/OS” sayfa 404](#)

Kuyruk yöneticilerinin ve ilişkili kaynakların yönetilmesi, bu kaynakları etkinleştirmek ve yönetmek için sık sık gerçekleştirdiğiniz görevleri içerir. Kuyruk yöneticilerinizi ve ilişkili kaynakları yönetmeyi tercih ettiğiniz yöntemi seçin.

[“IBM MQ for z/OS komutlarına komut verilmesi” sayfa 405](#)

Bir kuyruk yöneticisini denetlemek için toplu kipte ya da etkileşimli kipte IBM MQ komut dosyası komutlarını (MQSC) kullanabilirsiniz.

[“Recovery and restart on z/OS” sayfa 486](#)

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 yöneticilerinin yapılandırılması](#)

[IBM MQ for z/OS yardımcı programlarının kullanılması](#)

İlgili başvurular

[“z/OS üzerinde MQSC komutları yayınlatabileceğiniz kaynaklar” sayfa 20](#)

MQSC komutları, komutta bağlı olarak çeşitli kaynaklardan yayınlanabilir.

[“IBM MQ for z/OS yardımcı programları” sayfa 413](#)

IBM MQ for z/OS , sistem denetimi için kullanabilmeniz için kullanabileceğiniz bir dizi yardımcı program sağlar.

[Programlanabilir komut biçimleri başvurusu](#)

z/OS Günlüklerin yönetilmesi

Günlük arşivleme işlemi de içinde olmak üzere, günlük kaydı sıkıştırması, günlük kaydı kurtarma ve yazdırma günlüğü kayıtlarını kullanarak IBM MQ günlük kütüklerinizin nasıl yönetileceğini anlamak için bu konuyu kullanın.

Bu konuda, IBM MQ günlüklerinin yönetilmesinde yer alan görevler açıklanmaktadır. Bu kısımları içerir:

z/OS Günlüklerin **ARCHIVE LOG** komutuyla arşivlenmesi

Yetkili bir işletmen, **ARCHIVE LOG** komutu kullanılarak gerektiğinde yürürlükteki IBM MQ etkin günlük veri kümelerini arşivleyebilir.

ARCHIVE LOG komutunu verdiğinizde, IBM MQ yürürlükteki etkin günlük veri kümelerini keser, zamanuyumsuz bir boşaltma işlemi çalıştırır ve BSDS ' yi yükleme işleminin bir kaydı ile günceller.

ARCHIVE LOG komutunda bir **MODE (QUIESCE)** seçeneği vardır. Bu seçenekle, IBM MQ işleri ve kullanıcıları bir kesinleştirme noktasından sonra susturulmuş olur ve sonuçta elde edilen tutarlılık noktası yüklenmeden önce geçerli etkin günlüğe kaydedilir.

Site kurtarmayı iptal etmek için bir yedekleme stratejisi planlarken **MODE (QUIESCE)** seçeneğini kullanmayı düşünün. Kurtarma sırasında en güncel yedek sayfa kümesi kopyası ile arşiv günlüğü kullanıldığında, veri tutarsızlıklarının sayısını en aza indiren, sistem çapında bir tutarlılık noktası oluşturur. Örneğin:

```
ARCHIVE LOG MODE(QUIESCE)
```

ARCHIVE LOG komutunu bir **TIME** parametresi belirtmeden yayıyorsanız, susturma süresi varsayılan değeri olarak CSQ6ARVP makrosunun **QUIESCE** değıştirgesinin değerine ayarlanır. **ARCHIVE LOG MODE (QUIESCE)** ' in tamamlanması için gereken süre belirtilen süreden daha azsa, komut başarıyla tamamlanır; tersi durumda, zaman süresi dolduğunda komut başarısız olur. You can specify the time period explicitly by using the **TIME** option, for example:

```
ARCHIVE LOG MODE(QUIESCE) TIME(60)
```

Bu komut, **ARCHIVE LOG** işlemi gerçekleşmeden önce 60 saniyeye kadar susturulmuş bir süreyi belirtir.

Dikkat: Using the **TIME** option when time is critical can significantly disrupt IBM MQ availability for all jobs and users that use IBM MQ resources.

Komut varsayılan deşer olarak, komut g " nderme zamanından zamanuyumsuz olarak iYlenilir. (Komutu diğer IBM MQ komutlarıyla zamanuyumlu olarak işlemek için, **QUIESCE** ile **WAIT (YES)** seçeneğini kullanın, ancak z/OS konsolunun tüm **QUIESCE** dönemi için IBM MQ komut girişinden kilitlenmiş olduğunu unutmayın.)

Susturma dönemi sırasında:

- Kuyruk yöneticisindeki işlerin ve kullanıcıların kesinleştirme işleme yoluyla geçmelerine izin verilir, ancak kesinleştirme işleminden sonra herhangi bir IBM MQ kaynağını güncellemeye çalışırlarsa askıya alınır.
- Geçici olarak askıya alınan işler ya da kullanıcılar tarafından tutulan kilitleri bekliyor olabileceğinden, yalnızca okuma verileri olan işler ve kullanıcılar etkilenebilir.
- Yeni görevler başlayabilir, ancak verileri güncelleyemezler.

The output from the **DISPLAY LOG** command uses the message CSQV400I to indicate that a quiesce is in effect.

```

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
WRTHRSRSH 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

```

For example, for Long Term Support and Continuous Delivery before IBM MQ 9.1.2:

```

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
WRTHRSRSH 20
DEALLCT    0
COMPLLOG   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 quiesce geçmiş kaydı, etkin günlük veri kümelerinin kısaltıldığı tarih ve saat ile güncellenir ve yürürlükteki etkin günlük veri kümelerindeki son yazılan RBA ile güncellenir. IBM MQ , geçerli etkin günlük veri kümelerini, kullanılabilir sonraki etkin günlük veri kümelerine geçer ve boşaltma işleminin başlatıldığını belirten CSQJ311I iletisi verir.

Susturma süresi sona ermeden güncellemeler susturulamazsa, IBM MQ iletisi CSQJ317I ve **ARCHIVE LOG** işlemleri sonlandırılır. Yürürlükteki etkin günlük veri kümeleri kesildi ya da var olan sonraki günlük veri kümelerine geçmiyor ve yük boşaltma işlemi başlatılmadı.

Whether the quiesce was successful or not, all suspended users and jobs are then resumed, and IBM MQ issues message CSQJ312I, stating that the quiesce is ended and update activity is resumed.

Yürürlükteki etkin günlük kullanılabilir son etkin günlük veri kümesi olduğunda **ARCHIVE LOG** verilirse, komut işlenmez ve IBM MQ aşağıdaki iletiyi yayınlar:

```
CSQJ319I -csect-name CURRENT ACTIVE LOG DATA SET IS THE LAST
KULLANILABİLİR ETKİN OTURUM GÜNCELLEME
SONA ERDİRİLECEKTİR
```

ARCHIVE LOG komutu başka bir **ARCHIVE LOG** komutu zaten devam ederken yayınlandıysa, yeni komut işlenmez ve IBM MQ aşağıdaki iletiyi yayınlar:

```
CSQJ318I -ARŞIV GÜNCELLEŞTİRME KOMUTU DEVAM EDİYOR
```

Arşivleme sırasında yayınlanan iletilerle ilgili bilgi için bkz. [IBM MQ for z/OS iletileri](#).

Başarısızlığın ardından günlük arşivi işleminin yeniden başlatılması

Günlük arşivleme işlemi sırasında bir sorun varsa (örneğin, ayırma ya da manyetik bant bağlamalarıyla ilgili bir sorun), etkin günlüğün arşivlenmesi askıya alınmış olabilir. Arşiv işlemini iptal edebilir ve aşağıdaki komutu kullanarak yeniden başlatabilirsiniz:

```
ARCHIVE LOG CANCEL OFFLOAD
```

Bu komut, devam etmekte olan tüm yük boşaltma işlemlerini iptal eder ve arşiv işlemini yeniden başlatır. Bu, arşivlenmemiş en eski günlük veri kümesiyle başlar ve boşaltma ihtiyacı olan tüm etkin günlük veri kümeleriyle devam eder. Askıya alınmış günlük arşivi işlemleri yeniden başlatılır.

Bu komutu, yürürlükteki günlük arşivi görevinin çalışmadığından eminseniz ya da başarısız olan önceki bir girişimi yeniden başlatmak istiyorsanız bu komutu kullanın. Bunun nedeni, komutun, yük boşaltma görevinin olağandışı olarak sona erdirilmesine neden olabileceği için, bu da bir döküm ile sonuçlanabilir.

Arşivleme ve günlüğe kaydetme denetimi

Sıkıştırma, yazdırma, arşivleme, kurtarma ve günlüğe kaydetme işlemini, CSQ6LOGP, CSQ6ARVP ve CSQ6SYSP makrolarını kullanarak denetleyebilirsiniz. Özel nesnelere yapılan değişikliklerin yalnızca IBM MQ günlüklerinde günlüğe kaydedildiğini unutmayın. Grup nesnelere (paylaşılan gelen kanallar gibi) yapılan değişiklikler de günlüğe kaydedilir, çünkü tanımlar grup etrafında yayılır ve yerel olarak tutulur.

Arşivlemenin ve günlüğe kaydetmenin birçok yönü, kuyruk yöneticisi uyarlandığında, sistem parametre modülünün CSQ6LOGP, CSQ6ARVP ve CSQ6SYSP makroları kullanılarak ayarlanan parametreler tarafından denetlenir. Bu makroların ayrıntıları için bkz. [Sistem parametre biriminizi uyarlayın](#).

Bir kuyruk yöneticisi IBM MQ MQSC SET LOG, SET SYSTEM ve SET ARCHIVE komutları kullanılarak çalışırken bu parametrelerden bazıları değiştirilebilir. Bunlar [Çizelge 28 sayfa 451](#) içinde gösterilir:

<i>Çizelge 28. Kuyruk yöneticisi çalışırken değiştirilebilen parametreleri arşivleme ve günlüğe kaydetme</i>	
SET komutu	Parametreler
LOG	WRTHRSR, MAXARCH, DEALCT, MAXRTU, COMPLOG
arşiv	Tümü
SYSTEM	OTURUM Aç

MQSC [DISPLAY LOG](#), [DISPLAY ARCHIVE](#) ve [DISPLAY SYSTEM](#) komutlarını kullanarak tüm parametrelerin ayarlarını görüntüleyebilirsiniz. Bu komutlar, arşivleme ve günlüğe kaydetme ile ilgili durum bilgilerini de gösterir.

Günlük sıkıştırması denetleniyor

Herhangi birini kullanarak günlük kayıtlarının sıkıştırmasını etkinleştirebilir ve devre dışı bırakabilirsiniz.

- MQSC ' de SET ve DISPLAY LOG komutları; bkz. [MQSC komutları](#)
- PCF arabirimi çağrılıyor. Bakınız [“IBM MQ Programlanabilir Komut Biçimlerine Giriş” sayfa 23](#)

- Sistem parametre modülünde CSQ6LOGP makrosunu kullanma; bkz. [CSQ6LOGPkomutunu kullanma](#)

Günlük kayıtlarının yazdırılması

CSQ1LOGP yardımcı programını kullanarak günlük kayıtlarını ayıklayabilir ve yazdırabilirsiniz. Yönergeler için [Günlük yazdırma yardımcı programı](#) başlıklı konuya bakın.

Günlükleri kurtarma

Olağan durumda, özellikle ikili günlük kaydı kullanıyorsanız, IBM MQ günlüklerini yedekleme ve geri yükleme gereksiniminiz yoktur. Ancak, bir günlükteki G/Ç hatası gibi nadir durumlarda günlükleri kurtarmanız gerekebilir. Veri kümesini silmek ve yeniden tanımlamak için Access Method Services olanağını kullanın ve daha sonra, ilgili ikili günlüğü bu veri kümesini içine kopyalayın.

Arşiv günlüğü veri kümeleri atılıyor

Arşiv günlüğü veri kümelerinizi atabilir ve günlükleri otomatik olarak ya da el ile atmak için seçebilirsiniz.

Bir sayfa kümesi kaybolursa, sayfa kümesi ortam kurtarması ya da CF yapısı kaybolursa, bir sayfa kümesi kaybolursa, sayfa kümesi ortamı kurtarma işlemini gerçekleştirebilmek için yeterli günlük verileri saklamanız gerekir. Kurtarma işlemi için gerekli olabilecek arşiv günlüğü veri kümelerini atmayın; bu arşiv günlüğü veri kümelerini atsanız, gerekli kurtarma işlemlerini gerçekleştiremeyebilirsiniz.

Arşiv günlüğü veri kümelerinizin atılabileceğini doğruladıysanız, aşağıdaki yöntemlerden birini kullanarak bunu yapabilirsiniz:

- [Otomatik arşiv günlüğü veri kümesi silme](#)
- [Arşiv günlük veri kümelerini el ile silme](#)

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 kümeleri için alıkoyma süresi, CSQ6ARVP kuruluş makrosu için saklama süresi alanı ARCRETN tarafından belirlenir (ek bilgi için [UsingCSQ6ARVPUsing'](#) ye bakın).

Alıkoyma süresi için varsayılan değer, arşiv günlüklerinin 9999 gün boyunca alıkonacağı (üst sınır) olduğunu belirtir.

Önemli: Alıkoyma süresini değiştirebilirsiniz, ancak planladığınız yedekleme döngülerinin sayısını karşılayabildiğinizden emin olmanız gerekir.

IBM MQ , arşiv günlüğü veri kümeleri oluşturulduğunda, JCL parametresinin RETPD parametresine ilişkin değer olarak alıkoyma süresi değerini kullanır.

MVS/DFP depolama yönetimi altsistemi (SMS) tarafından ayarlanan alıkoyma süresi bu IBM MQ parametresiyle geçersiz kılınabilir. Alıkoyma süresi tipik olarak, IBM MQ ya da SMS tarafından belirlenen daha küçük değere ayarlanır. Depolama yöneticisi ve IBM MQ yöneticisi, IBM MQ için uygun bir alıkoyma süresi değeri üzerinde anlaşmaya varmalıdır.

Not: Bazı manyetik bant yönetim sistemleri, alıkoyma sürelerinin dış el ile geçersiz kılınmasını sağladığından, IBM MQ , BSDS ' den arşiv günlüğü veri kümeleriyle ilgili bilgileri silmek için otomatik bir yöntemle sahip değildir. Bu nedenle, veri kümesi saklama süresi sona erdikten sonra ve veri kümesi manyetik bant yönetim sistemi tarafından çizildikten sonra, bir arşiv günlüğü veri kümesi hakkındaki bilgiler BSDS ' de hala olabilir. Bunun tersine, arşiv günlüğü veri kümesi sayısı üst sınırı aşılmış olabilir ve veri kümesi son kullanma tarihine ulaşmadan önce BSDS ' deki veriler atılmış olabilir.

Arşiv günlüğü veri kümeleri otomatik olarak silinirse, işlemin BSDS ' deki arşiv günlüklerinin listesini güncellemediğini unutmayın. Değişiklik günlüğü döküm yardımcı programıyla BSDS ' yi ["BSDS ' nin](#)

Değiştirilmesi” sayfa 459’inde açıklandığı şekilde güncelleyebilirsiniz. Güncelleme şart değil. Eski arşiv günlükleri BSDS ' de boş yer yapıştırır, ancak başka bir zarar vermez.

Arşiv günlüğü veri kümelerinin el ile silinmesi

You must keep all the log records as far back as the lowest RBA identified in messages CSQI024I and CSQI025I. Bu RBA, [Method 1: Full backup](#)(Yöntem 1: Tam yedekleme) seçeneğini kullanarak kurtarma noktası yaratırken çıkardığınız DISPLAY USAGE komutu kullanılarak elde edilir.

Herhangi bir günlük atılmadan önce Paylaşılmayan kaynaklar için kurtarma noktası oluşturma başlıklı konuya bakın.

Arşiv günlüğü veri kümelerini bul ve at

Kurtarma için gereken en düşük günlük RBA ' yı oluşturduysanız, yalnızca daha önceki günlük kayıtlarını içeren arşiv günlüğü veri kümelerini aşağıdaki yordamı izleyerek bulabilirsiniz:

1. BSDS ' nin içeriğini yazdırmak için yazdırma günlüğü eşleme yardımcı programını kullanın. Çıkışa bir örnek için bakınız: [The print log map utility](#).
2. Find the sections of the output titled ARCHIVE LOG COPY n DATA SETS. İkili günlük kaydı kullanıyorsanız, iki bölüm vardır. STARTRBA ve ENDRBA etiketli kolonlar her birimdeki RBA ' ların aralığını gösterir. Find the volumes with ranges that include the minimum RBA you found with messages CSQI024I and CSQI025I. Bunlar, tutmanız gereken en eski birimlerdir. Çift günlük kaydı kullanıyorsanız, bu tip iki birim vardır.

Herhangi bir birim uygun bir aralığa sahip değilse, aşağıdaki durumlardan biri geçerli olur:

- RBA alt sınırı henüz arşivlenmedi ve tüm arşiv günlüğü birimlerini atabilirsiniz.
- The list of archive log volumes in the BSDS wrapped around when the number of volumes exceeded the number allowed by the MAXARCH parameter of the CSQ6LOGP macro. BSDS bir arşiv günlüğü birimini kaydetmezse, bu birim kurtarma işlemi için kullanılamaz. Bu nedenle, var olan birimlerle ilgili bilgileri BSDS ' ye eklemeyi düşünün. Yönergeler için, bkz. [“Arşiv günlüklerinde değişiklik” sayfa 461](#).

Ayrıca MAXARCH değerini artırmayı düşünün. Bilgi için bkz. [CSQ6LOGPkomutunu kullanma](#).

3. Tutmak istediğiniz en erken birimin STARTRBA değerinden daha küçük bir ENDRBA değeri olan arşiv günlüğü veri kümesini ya da birimini silin. İkili günlük kaydı kullanıyorsanız, bu tür kopyaları da silin.

BSDS girdileri toparlandığından, BSDS arşiv günlüğü bölümündeki ilk birkaç girdi, sondaki girdilerden daha yeni olabilir. Tarih ve saatin birleşimine bakın ve yaşlarını karşılaştır. En az LOGRBA alt sınırını içeren arşiv günlüğü girdisinden önce tüm girdileri atayabileceğini varsaymayın.

Veri kümelerini silin. Eğer arşivler teypteysse, kasetleri silin. DASD üzeriyse, her veri kümesini silmek için bir z/OS yardımcı programı çalıştırın. Daha sonra, BSDS ' nin yalnızca var olan arşiv birimlerini listelemesini istiyorsanız, atılan birimlere ilişkin girdileri silmek için günlük döküm yardımcı programını (CSQJU003) kullanın. Örneğin, [“Arşiv günlüklerinde değişiklik” sayfa 461](#) başlıklı konuya bakın.

Günlüğün yıldırıcı etkisi

Uzun süren işlemler, günlük veri kümelerine yayılan iş günlüğü kayıtlarına neden olabilir. IBM MQ bu senaryoya, günlük verilerinin alınacağı günlük kayıtlarını ve kuyruk yöneticisini yeniden başlatma süresini en iyi duruma getirmek için günlük kayıtlarını taşıyan bir teknik kullanarak bu senaryoya işler.

Bir iş birimi uzun olduğu düşünüldüğünde, her bir günlük kaydının bir temsili daha sonra günlüğe yazılır. Bu, *göz korkutucu olarak günlüğe kaydetme* olarak bilinir. [Günlük dosyaları](#)' ta daha tam olarak açıklanmıştır.

Kuyruk yöneticisi, bir hata oluştuğundan sonra, iş bütünlüğü birimini sağlamak için, bu şunurlu günlük kayıtlarını orijinalleri yerine kullanır. Bunun iki avantajı vardır:

- İş birimi eşgüdümü birimi için alıkonması gereken günlük verilerinin miktarı azaltılır
- daha az günlük verileri kuyruk yöneticisi yeniden başlatma sırasında traversed olmalıdır, bu nedenle 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 eşgüdümü birimi. Bir CF yapısını ya da sayfa kümesini etkileyen bir ortam hatası ortaya çıkarsa, kuyruk yöneticisi önceki bir kopyayı geri yükleyerek ve günlükte bulunan verileri kullanarak bu ortamı güncelleyerek ortamı kurtarabilir noktasına kadar kurtarabilir. Bir iş biriminde gerçekleştirilen kalıcı etkinlik, günlüğe kaydedildiğinde, hata durumunda yedeklenebilecek ya da kilitlerin değiştirilen kaynaklarda kurtarılabilmesi için kaydedilir. Kuyruk yöneticisi kurtarmasını etkinleştirmek için alıkoymanız gereken günlük verileri miktarı bu iki öğeden etkilenir.

Ortam kurtarma işlemi için, en azından en yeni ortam kopyasından ortam kurtarma işlemi gerçekleştirilebilmek için yeterli günlük verilerini saklamanız ve geri dönebilmeniz gerekir. (Siteniz, eski yedeklerden kurtarma yeteneğini öngörüyor olabilir.) İş bütünlüğü birimi için, en eski uçuş ya da belirsiz iş biriminiz için günlük verilerini korumanız gerekir.

To assist you with managing the system, the queue manager detects old units of work at each log archive and reports them in messages CSQJ160 and CSQJ161. İç görev, bu eski iş birimlerine ilişkin iş günlüğü bilgilerini okur ve günlükteki yürürlükteki konuma daha kısa bir biçimde yeniden yazar. CSQR026 iletisi ne zaman oluştuğunu gösterir. MQSC komutu DISPLAY USAGE TYPE (DATASET), günlük verilerinin alıkonmasını yönetmenize de yardımcı olabilir. Komut, aşağıdaki üç kurtarma bilgilerini bildirir:

1. İş birimi kurtarma birimi için günlüğün ne kadarının korunması gerekir.
2. Sayfa kümelerinin ortam kurtarma işlemi için günlüğün ne kadarının korunması gerekir.
3. Kuyruk paylaşım grubundaki bir kuyruk yöneticisi için, CF yapılarının ortam kurtarma işlemi için günlüğün ne kadarının korunması gerektiğini belirleyin.

Bu bilgi parçalarının her biri için, veri kümesi için gerekli olan en eski günlük verilerini eşleştirmek için bir girişimde bulunmanız gerekir. Yeni iş birimleri başlangıç ve durdurma birimi olarak (1) günlük içinde daha yeni bir konuma taşınması beklenir. Hareket etmiyorsa, uzun süredir çalışan UOW iletileri, bir sorun olduğu konusunda sizi uyarır. (2) kuyruk yöneticisi şimdi sona erdirilecekse ve yeniden başlatılacaksa, sayfa kümesi ortam kurtarması ile ilgilidir. Sayfa kümelerinizi en son yedeklediğinizde ya da bir sayfa kümesi hatası varsa, kullanmak zorunda kalabileceğiniz yedeklemeyi bilmiyorsa, bu değer bilmiyor. Genellikle, arabellek havuzlarında tutulan değişiklikler sayfa kümelerine yazıldığı için, denetim noktası işleme sırasında günlük içinde daha yeni bir konuma geçer. (3) içinde, kuyruk yöneticisi, bu kuyruk yöneticisinde ya da kuyruk paylaşım grubundaki diğer kuyruk yöneticilerindeki CF yapısı yedeklemelerini biliyor. Ancak, CF yapısı kurtarma işlemi, son yedeklemeden bu yana CF yapısıyla etkileşim içinde olan kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinden günlük verilerinin birleştirilmesini gerektirir. Bu, günlük verilerinin bir günlük kaydı sıra numarasıyla (ya da LRSN), zaman damgası tabanlı ve bu nedenle, kuyruk paylaşım grubundaki farklı kuyruk yöneticilerinde farklı olacak bir RBA değil, tüm kuyruk paylaşım grubu boyunca geçerli olacak şekilde tanımlandığı anlamına gelir. Olağan durumda, bu ya da kuyruk paylaşım grubundaki diğer kuyruk yöneticilerinde BACKUP CFSTRUCT komutlarıyla daha yeni bir konuma geçilir.

Kuyruk yöneticisi günlüğünün ilk durumuna getirilmesi

Kuyruk yöneticisinin günlüğünü nasıl sıfırlatacağını anlamak için bu konuyu kullanın.

Kuyruk yöneticisi günlüğü RBA 'nın günlük RBA' nın sonundan 0 'a kaydırılmasına izin vermemeniz gerekir; bu durum bir kuyruk yöneticisi kesintisine yol açar ve tüm kalıcı veriler kurtarılamaz duruma gelir. Günlük RBA 'nın sonu ya bir FFFFFFFFFF (6 baytlık RBU' lar kullanımda) ya da FFFFFFFFFFFFFFFF (8 baytlık RBA ' lar kullanılsa).

Kuyruk yöneticisi iletileri [CSQI045I](#), [CSQI046E](#), [CSQI047E](#), [CSQJ031D](#) ve [CSQJ032E](#) ile yayınlar. Bu durumda, planlanmamış bir kesintiyi önlemek için işlem yapmayı planlamanız gerektiğini belirtir.

RBA değeri FFF800000000 değerine ulaştığında kuyruk yöneticisi [00D10257](#) (6 baytlık günlük RBA 'lar kullanımdaysa) ya da FFFFFFFC00000000 (8 baytlık günlük RBA' ları kullanımdaysa) neden kodu ile sona erer.

6 baytlık günlük RBA 'ları kullanımdaysa, kuyruk yöneticisinin “[Daha büyük günlük Görelı Bayt Adresını uygulama](#)” sayfa 455içinde açıklanan işlemleri izleyerek, kuyruk yöneticisinin günlüğünü sıfırlamak yerine 8 baytlık günlük RBU' ları kullanacak şekilde dönüştürmeyi düşünün. Kuyruk yöneticisinin 8 baytlık günlük RBA ' yı kullanmak üzere dönüştürülmesi, günlüğün ilk durumuna getirilmesinden daha kısa bir kesinti yapılmasını gerektirir ve günlüğü ilk durumuna getirmeniz gerekmeden önce zaman aralığını artırır.

Message CSQJ034I, issued during queue manager initialization, indicates the end of the log RBA range for the queue manager as configured, and can be used to determine whether 6-byte or 8-byte log RBAs are in use.

Kuyruk yöneticisinin günlüğünü ilk durumuna getirmek için izlenmesi gereken yordam aşağıdaki gibidir:

1. Çözülmemiş iş birimlerini çözün. Çözülmemiş iş birimlerinin sayısı, INDOUBT sayısı olarak CSQR005I iletilerinde kuyruk yöneticisi başlangıcında görüntülenir. Her bir denetim noktasında ve kuyruk yöneticisi kapatılırken, kuyruk yöneticisi komutu otomatik olarak yayınlar.

DISPLAY CONN(*) TYPE(CONN) ALL WHERE(UOWSTATE EQ UNRESOLVED) çözülmemiş iş birimleri hakkında bilgi sağlamak için.

Kurtarma birimlerinin çözülmesine ilişkin bilgi için [Nasıl belirsiz kurtarma birimlerinde çözümlenir](#) konusuna bakın. Nihai geri alma işlemi, belirsiz kurtarma birimlerini el ile çözmek için **RESOLVE INDOUBT MQSC** komutunu kullanmaktadır.

2. Kuyruk yöneticisini temizleme işlemini sona erdirin.

Her iki komut da, değiştirilen sayfaları arabellek havuzlarından sayfa kümelerine boşaltmak için **STOP QMGR** ya da **STOP QMGR MODE(FORCE)** ' ı kullanabilirsiniz.

3. Kuyruk yöneticisi bir kuyruk paylaşım grubunun bir parçasıysa, kuyruk paylaşım grubundaki tüm yapılar için CFSTRUCT yedeklemelerini diğer kuyruk yöneticilerine alın. Bu, en son yedeklerin bu kuyruk yöneticisinin günlüğünde yer almamasını ve CFSTRUCT kurtarma işlemi için bu kuyruk yöneticisinin günlüğünün gerekli olmadığını doğrular.
4. Yeni günlükleri ve BSDS ' yi CSQJU003 kullanarak tanımlayın (değişiklik günlüğü döküm yardımcı programının kullanımıyla ilgili ek bilgi için [Değişiklik günlüğü döküm yardımcı programı](#) başlıklı konuya bakın).
5. Bu kuyruk yöneticisine ilişkin tüm sayfa kümelerine karşı **CSQUTIL RESETPAGE** komutunu çalıştırın (bu işlemin kullanılmasıyla ilgili ek bilgi için [Sayfa kopyalanması ve günlüğünün ilk durumuna getirilmesi](#) başlıklı konuya bakın). Sayfa kümesi RBU ' ların bağımsız olarak ilk duruma getirilebileceğini göz önünde bulundurun. Bu nedenle, bu adım için geçen süreyi azaltmak için birden çok eşzamanlı iş (örneğin, bir sayfa başına bir küme) gönderilebilir.
6. Kuyruk yöneticisini yeniden başlat

İlgili kavramlar


“[Daha büyük günlük Görelı Bayt Adresını uygulama](#)” sayfa 455

IBM MQ for z/OS 8.0öncesinde, IBM MQ for z/OS , günlük içindeki verilerin yerini belirlemek için 6 baytlık bir RBA günlüğünü kullandı. IBM MQ for z/OS 8.0' tan, günlük RBA 8 bayt uzunluğunda olabilir ve günlüğü sıfırlamak zorunda kalmaktan önceki süreyi uzatır.

Daha büyük günlük Görelı Bayt Adresını uygulama

IBM MQ for z/OS 8.0öncesinde, IBM MQ for z/OS , günlük içindeki verilerin yerini belirlemek için 6 baytlık bir RBA günlüğünü kullandı. IBM MQ for z/OS 8.0' tan, günlük RBA 8 bayt uzunluğunda olabilir ve günlüğü sıfırlamak zorunda kalmaktan önceki süreyi uzatır.

IBM MQ 9.2.0 Uzun Süreli Destek kuyruğu yöneticileri ve IBM MQ 9.2.4' dan önce oluşturulan Sürekli Teslimat kuyruğu yöneticileri için bu özelliğin açık bir şekilde etkinleştirilmesi gerekir.

 **V 9.2.5** IBM MQ 9.2.5 ve sonraki düzeylerde oluşturulan kuyruk yöneticileri için bu özellik önceden etkinleştirilmiştir.

8 baytlık günlük RBA ' yı etkinleştirmeyi planlarken dikkat edilmesi gereken noktalar için bkz. [Planning to arge the maximum addressable log range](#) .

Önemli: **V9.2.0** Kuyruk yöneticiniz bir kuyruk paylaşım grubunun bir parçası değilse ve daha sonra, 8 baytlık günlük RBA 'yı etkinleştirmiş olan IBM MQ for z/OS 9.0.0' a geri dönmeniz durumunda, bu yayınlarda **OPMODE=NEWFUNC, 900** programını kullandığınızdan emin olun, tersi durumda kuyruk yöneticinizin başlatılamaması gerekir.

Tek bir IBM MQ for z/OS kuyruk yöneticisinde 8 baytlık günlük RBA 'yı etkinleştirmek için bu yönergeleri gösterildiği sırayla gerçekleştirin:

1. **V9.2.0** If your queue manager is in a queue sharing group ensure that all queue managers in the queue sharing group that are at IBM MQ for z/OS 9.0.0 are running with **OPMODE=(NEWFUNC,900)** before moving to step "2" sayfa 456.

Bunu yapmak için toplam kuyruk paylaşım grubu kesintisi sürmenize gerek yoktur. Sırayla IBM MQ for z/OS 9.0.0 konumunda olan her bir kuyruk yöneticisini durdurabilirsiniz, **OPMODE=(NEWFUNC,900)** olarak değiştirin ve yeniden başlatın.

Kuyruk paylaşım grubundaki tüm IBM MQ for z/OS 9.0.0 kuyruk yöneticileri **OPMODE=(NEWFUNC,900)** ile çalıştırıldıktan sonra, kuyruk paylaşım grubundaki her kuyruk yöneticisi için, tüm kuyruk yöneticileri yeni BSDS ile çalıştırılacağı kadar aşağıdaki adımları gerçekleştirin.

2. Benzer özniteliklere sahip yeni BSDS veri kümelerini geçerli BSDS ' lere ayırın. You can tailor sample CSQ4BSDS and delete any irrelevant statement, or you can use your existing JCL, but change the BSDS name to something like ++HLQ++ . NEW . BSDS01.

Notlar:

- a. Yeni BSDS 'yi ayırmak için işi göndermeden önce yeni BSDS' nizin özniteliklerini denetleyin. Değiştirebilecek tek öznitelik, BSDS ' nin büyüklüğüdür.
 - b. Yeni BSDS, geçerli BSDS ' nin içerdiği daha fazla veri içerir, bu nedenle yeni veri kümelerinin yeterli kullanılabilir alan ile ayrıldığından emin olmanız gerekir. thlqual.SCSQPROC(CSQ4BSDS) içindeki örnek JCL, yeni bir BSDS tanımlarken önerilen değerleri içerir.
3. Kuyruk yöneticisini temizleme işlemini sona erdirin.
 4. Var olan BSDS ' leri yeni BSDS veri kümelerine dönüştürmek için BSDS conversion utility (CSQJUCNV) komutunu çalıştırın. Bu işlem genellikle birkaç saniye sürer.
Var olan BSDS ' niz bu işlem sırasında değiştirilmeyecek ve başarısız bir dönüştürme durumunda kuyruk yöneticisinin kullanıma hazırlanması için bunu kullanabilirsiniz.
 5. Yürürlükteki BSDS 'leri eski BSDS olacak şekilde yeniden adlandırın ve yeni BSDS ' yi yürürlükteki BSDS olacak şekilde yeniden adlandırın; böylece kuyruk yöneticisini sonraki yeniden başlatıldığınızda 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')
```

Ayrıca, VSAM kümesinin hem veri hem de dizin kısımlarını yeniden adlandırmak için komut verdiğinizden emin olun.

6. Kuyruk yöneticisini yeniden başlatın. Bu, 6 baytlık günlük RBA kullanılırken gerçekleştirileceği süre içinde başlamalıdır.

Dönüştürülen BSDS ' ye erişilememesi nedeniyle kuyruk yöneticisi başarıyla yeniden başlatılamazsa, başarısızlığın nedenini saptama girişiminde bulunmayı deneyin, sorunu çözün ve işlemi yeniden deneyin. Gerekirse, yardım almak için IBM destek merkezimize başvurun.

Gerekirse, bu noktada değişiklik şu noktada yedeklenebilir:

- a. Yürürlükteki BSDS ' ler yeni BSDS olacak şekilde yeniden adlandırılıyor.
- b. Eski BSDS ' ler yürürlükteki BSDS olacak şekilde yeniden adlandırılıyor.
- c. Kuyruk yöneticisi yeniden başlatılıyor.

Kuyruk yöneticisi dönüştürülen BSDS ile başarıyla yeniden başlatıldıktan sonra, eski BSDS ' leri kullanarak kuyruk yöneticisini başlatma girişiminde bulunmayın.

7. Kuyruk yöneticisi kullanıma hazırlanırken kuyruk yöneticisi için, yapılandırılan kuyruk yöneticisi için günlük RBA ' nın sonunu göstermek üzere CSQJ034I iletisi yayınlandı. Görüntülenen günlük RBA aralığının bitmesinin FFFFFFFFFFFFFFFF olduğunu doğrulayın. Bu, 8 baytlık günlük RBA ' nın kullanımda olduğunu gösterir.

Not: Yeni bir kuyruk yöneticisinde 8 baytlık günlük RBA ' yı etkinleştirmek için, ilk başlatılmadan önce, boş bir sürüm 1 biçimi BSDS yaratmalı ve bir sürüm 2 biçimi BSDS üretmek için BSDS dönüştürme yardımcı programına giriş olarak kullanmanız gerekir. Bu süreci nasıl gerçekleştirdiğiniz hakkında bilgi için [Önyükleme ve günlük veri kümelerinin oluşturulması](#) başlıklı konuya bakın.

İlgili kavramlar

[Daha büyük günlük Görelî Bayt Adresi](#)

İlgili görevler

[Adreslenebilir günlük aralığı üst sınırını artırmayı planlama](#)

İlgili başvurular

[BSDS dönüştürme yardımcı programı \(CSQJUCNV\)](#)

z/OS Önyükleme veri kümesinin yönetilmesi (BSDS)

Önyükleme veri kümesi (BSDS), günlük veri kümelerine ve günlük kayıtlarına gönderme yapmak için kullanılır. BSDS ' yi nasıl inceleyebileceğinizin, değiştirebileceğinizin ve kurtarmanın nasıl gerçekleştirileceğini anlamak için bu konuyu kullanın.

Ek bilgi için [Önyükleme veri kümesibaşlıklı konuya](#) bakın.

Bu konuda, önyükleme veri kümesinin yönetilmesinde yer alan görevler açıklanmaktadır. Bu kısımları içerir:

- [“BSDS ' nin içerdiği bilgileri bulma” sayfa 457](#)
- [“BSDS ' nin Değiştirilmesi” sayfa 459](#)
- [“BSDS ' nin Kurtarılıyor” sayfa 463](#)

z/OS BSDS ' nin içerdiği bilgileri bulma

BSDS ' nin içeriğini incelemek için yazdırma günlüğü eşleme yardımcı programını (CSQJU004) kullanabilirsiniz.

Yazdırma günlüğü eşleme yardımcı programı (CSQJU004), BSDS ' de saklanan bilgileri listeleyen bir toplu iş yardımcı programıdır. Çalıştırılmasına ilişkin yönergeler için [Yazdırma günlüğü eşleme yardımcı programı](#) başlıklı konuya bakın.

BSDS şunları içerir:

- [Zaman damgaları](#)
- [Etkin günlük veri kümesi durumu](#)

BSDS ' deki zaman damgaları

Yazdırma günlüğü eşleme yardımcı programının çıkışı, BSDS ' de saklanan çeşitli sistem olaylarının tarih ve saatini kaydetmek için kullanılan zaman damgalarını gösterir.

Aşağıdaki zaman damgaları raporun üstbilgi bölümünde yer alır:

SİSTEM DAMGASI

BSDS ' nin en son güncellendiği tarih ve saati yansıtır. BSDS zaman damgası aşağıdaki durumlarda güncellenebilir:

- Kuyruk yöneticisi başlatılır.

- Günlük yazma etkinlikleri sırasında yazma eşiğine ulaşılır. Belirlediğiniz çıkış arabelleği sayısına ve sistem etkinlik hızına bağlı olarak, BSDS saniyeler içinde birkaç kez güncellenebilir ya da birkaç saniye, dakika ya da saat için güncellenmeyebilir. Yazma eşiğinin ayrıntıları için, [CSQ6LOGPkomutunu kullanma](#)indeki CSQ6LOGP makrosu WRTHRS parametresine bakın.
- IBM MQ , bir hata nedeniyle normal ikili BSDS kipinden tek bir BSDS kipine düşmektedir. Bu durum, bir BSDS kaydı alma, ekleme, işaret etme, güncelleme ya da silme isteği başarısız olduğunda ortaya çıkabilir. Bu hata ortaya çıktığında, IBM MQ , devre dışı bırakılan BSDS ile zaman damgası uyumsuzluğunu zorlamak için kalan BSDS ' deki zaman damgasını günceller.

YARDIMLA

BSDS ' nin içeriğinin değişiklik günlüğü stok yardımcı programı tarafından değiştirildiği tarih ve saat (CSQJU003).

Aşağıdaki zaman damgaları raporun etkin ve arşiv günlüğü veri kümeleri bölümünde yer alır:

Etkin günlük tarihi

Etkin günlük girişinin BSDS ' de yaratıldığı tarih; CSQJU003 NEWLOG değeri yapıldığında.

Etkin günlük süresi

Etkin günlük girişinin BSDS ' de yaratıldığı saat (CSQJU003 NEWLOG) gerçekleştirildiği zaman.

Arşiv günlüğü tarihi

Arşiv günlüğü girdisinin BSDS ' de yaratıldığı tarih; yani, CSQJU003 NEWLOG işlemi yapıldığında ya da arşivin kendisi yapılsa.

Arşiv günlüğü süresi

Arşiv günlüğü girişinin BSDS ' de yaratıldığı saat (CSQJU003 NEWLOG değeri yapıldığında ya da arşivin kendisi yapılsa).

Etkin günlük veri kümesi durumu

BSDS, etkin bir 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 kullanılandan önce bir noktaya kesildi. Her iki durumda da, veri kümesi başlangıç ve bitiş RBA değerleri sıfır olarak sıfırlanır.

Yeniden Kullanılabilir

Veri kümesi tanımlandı, ancak IBM MQ tarafından hiçbir zaman kullanılmadı ya da veri kümesi boşaltıldı. Yazdırma günlüğü eşlemi çıkışında, son YENIDEN kullanılabilir veri kümesine ilişkin RBA başlangıç RBA değeri, son arşiv günlüğü veri kümesinin başlangıç RBA değerine eşit.

YENIDEN KULLANILMA

Veri kümesi, boşaltılmamış kayıtlar içeriyor.

Durduruldu

Ofsayt işlemcisi bir kaydı okurken bir hata saptadı 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 ile başlayan ve kesilen etkin günlük veri kümesindeki en son geçerli kayıt kesimine kadar devam eden officrid (boşaltma) olur. Son geçerli kayıt kesiminin RBA 'si, etkin günlük veri kümesinin bitiş RBA ' sına göre daha düşük. Günlüğe kaydetme, kullanılabilir bir sonraki etkin günlük veri kümesine geçilir ve kesintisiz olarak devam eder.

ya da

- Etkin günlüğü kesilen bir ARCHIVE LOG işlevi çağrıldı.

Durum, yazdırma günlüğü eşlemi yardımcı programının çıktısında görüntülenir.

z/OS BSDS ' nin Deęiřtirilmesi

IBM MQ bunu otomatik olarak yaptıęı için, BSDS ' nin gnlęe kaydetme olaylarının kayıtlarıyla gncellenmesini korumak için özel adımlar atmanıza gerek yoktur.

Ancak, ařaęıdakilerden herhangi birini yapmak isterseniz, BSDS ' yi deęiřtirmek isteyebilirsiniz:

- Daha fazla etkin gnlk veri kmesi ekleyin.
- Etkin gnlk veri kmelerini yeni ayrılan veri kmelerine (rneęin, daha byk etkin gnlk ayrımları saęlarken) kopyalayın.
- Gnlk verileri kmelerini dięer aygıtlara tařıyın.
- Hasar grmř bir BSDS ' yi kurtarın.
- Eski arřiv gnlę veri kmelerini atın.

Deęiřiklik gnlę dkm yardımcı programını (CSQJU003) alıřtırarak BSDS ' yi deęiřtirebilirsiniz. Bu yardımcı programı yalnızca, kuyruk yneticisi etkin deęilse alıřtırın ya da tutarsız sonular alabilirsiniz. Yardımcı programın iřlemi, SYSIN veri kmesindeki deyimler tarafından denetlenir. Bu blmde eřitli rnekler gsterilir. Eksiksiz ynergeler için [Deęiřiklik gnlę dkm yardımcı programı](#)bařlıklı konuya bakın.

Etkin gnlk verileri kmesini, IBM MQ etkin gnlk veri kmelerini kuyruk yneticisi bařlatma sırasında dıřlayıcı (DISP = OLD) olarak ayırdıęı için, yalnızca kuyruk yneticisi etkin deęil olduęunda kopyalayabilirsiniz.

z/OS Etkin gnlklere iliřkin deęiřiklikler

BSDS ' yi kullanarak etkin gnlklerin nasıl deęiřtirileceęini anlamak için bu konuyu kullanın.

Deęiřiklik gnlę yardımcı programını kullanarak etkin gnlklere iliřkin BSDS ' deki giriřleri ekleyebilir, bunları silebilir ve kaydedebilir. Burada yalnızca rnekler gsterilir; kullanmak istediklerinize gsterilen veri kmesi adlarını deęiřtirin. Yardımcı programın ayrıntıları için [Deęiřiklik gnlę dkm yardımcı programı](#)bařlıklı konuya bakın.

Daha fazla bilgi için ařaęıdaki blmlere bakın:

- [BSDS ' ye kayıt giriři eklenmesi](#)
- [BSDS ' den etkin gnlk verileri kmesiyle ilgili bilgileri silme](#)
- [BSDS ' de ayarlanan gnlk verileri hakkında bilgi kaydedilmesi](#)
- [Etkin gnlk byklęnn artırılması](#)
- [CSQJUFMT kullanımı](#)

BSDS ' ye kayıt girdileri ekleme

Etkin bir gnlk "durduruldu" olarak iřaretlendiyse, gnlęe kaydetme için yeniden kullanılmaz; ancak okuma için kullanılmaya devam eder. Yeni etkin gnlk veri kmelerini tanımlamak için eriřim yntemi hizmetlerini kullanın ve daha sonra, yeni veri kmelerini BSDS ' ye kaydetmek için gnlk dkm alma yardımcı programını kullanın. rneęin, řunları kullanın:

```
NEWLOG DSN=MQM111.LOGCOPY1.DS10,COPY1
NEWLOG DSN=MQM111.LOGCOPY2.DS10,COPY2
```

Eski bir etkin gnlk verilerinin ierięini yenisine kopyalıyorsanız, NEWLOG iřlevindeki RBA aralıęını ve bařlangı ve bitiř zaman damgalarını da verebilirsiniz.

BSDS ' den etkin gnlk verileri kmesiyle ilgili bilgileri silme

BSDS ' den etkin bir gnlk veri kmesine iliřkin bilgileri silmek için ařaęıdaki zellikleri kullanabilirsiniz:

```
DELETE DSNNAME=MQM111.LOGCOPY1.DS99
DELETE DSNNAME=MQM111.LOGCOPY2.DS99
```

BSDS ' de ayarlanan günlük verileri hakkında bilgileri kaydetme

BSDS ' de var olan bir etkin günlük verileri kümesine ilişkin bilgileri kaydetmek için aşağıdaki adresi kullanın:

```
NEWLOG DSNNAME=MQM111.LOGCOPY1.DS10,COPY2,STARTIME=19930212205198,
ENDTIME=19930412205200,STARTRBA=6400,ENDRBA=94FF
```

Aşağıdaki gibi, BSDS ' de bu tip bilgileri içeren bir kayıt eklemeniz gerekebilir:

- Veri kümesine ilişkin giriş silindi, ancak yeniden gerekiyor.
- Bir etkin günlük verisi kümesinin içeriğini başka bir veri kümesine kopyalıyorsunuz.
- BSDS ' yi yedek kopyadan kurtarıyorsunuz.

Etkin günlük büyüklüğünün artırılması

Bu işlemi gerçekleştirmenin iki yöntemi vardır.

1. Kuyruk yöneticisi etkin olduğunda:

- a. JCL kullanarak yeni daha büyük günlük veri kümeleri tanımlayın.
- b. MQSC DEFine LOG komutunu kullanarak, yeni günlük verileri kümelerini etkin kuyruk yöneticisine ekleyin.
- c. Yürürlükteki etkin günlüğü taşımak için MQSC ARCHIVE LOG komutunu kullanın ve yeni daha büyük bir günlük kullanın.
- d. Daha küçük etkin günlük veri kümesinin arşivlenmesinin tamamlanmasını bekleyin.
- e. Eski küçük etkin günlükleri kaldırmak için CSQJU003 yardımcı programını kullanarak kuyruk yöneticisini kapatın.
- f. Kuyruk yöneticisini yeniden başlatın.

2. Kuyruk yöneticisi etkinlik dışı olduğunda:

- a. Kuyruk yöneticisini durdurun. Bu adım, IBM MQ etkin olduğunda özel kullanım için tüm etkin günlük veri kümelerini ayırdığı için bu adım zorunludur.
- b. Etkin günlük veri kümelerinizi yeniden adlandırmak için, NEWNAME seçeneğiyle birlikte ALTER 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, BSDSs içinde yeni adlar oluşturmak için değişiklik günlüğü döküm yardımcı programını çalıştırmanız gerekmez. Eski veri kümesi adları ve doğru RBA aralıkları zaten BSDSS ' de bulunur.

- d. Eski (yeniden adlandırılan) veri kümelerini uygun yeni veri kümelerine kopyalamak için Erişim Yöntemi Hizmetleri REPRO ' u kullanın.

Not: Bu adım uzun sürebilir, bu nedenle kuruluşunuzun bu dönem için işlem dışı olması gerekir.

- e. Kuyruk yöneticisini başlatın.

Tüm günlük veri kümeleriniz aynı büyükse, sisteminiz işlevsel olarak daha tutarlı ve verimli olur. Günlük veri kümeleri aynı boyutta değilse, sisteminizin günlüklerinin izlenmesi daha zordur ve bu nedenle boşluk kaybedilebilir.

CSQJUFMT kullanımı

Etkin bir günlüğün büyüklüğünü artırırken CSQJUFMT biçimi çalıştırmayın.

CSQJUFMT komutunu çalıştırırsanız (kuyruk yöneticisinin yeni etkin günlüğüne ilk kez yazması için bir 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
```

Buna ek olarak, Erişim Yöntemi Hizmetleri REPRO ' yı kullanıyorsanız, yeni bir boş günlük tanımladığınızdan emin olun.

Eski (yeniden adlandırılan) veri kümesini kendi yeni veri kümesine kopyalamak için REPRO kullanırsanız, varsayılan değer NOREPLACE 'tır.

Bu, REPRO ' un önceden belirlenmiş veri kümesinde bulunan bir kaydı değiştirmede anlamına gelir. Veri kümesinde biçimlendirme işlemi gerçekleştirildiğinde, RBA değeri sıfırlanır. Net sonuç, biçimlendirmeden sonra boş olmayan bir veri kümesinden oluşur.

z/OS Arşiv günlüklerinde değişiklik

Arşiv günlüklerinin nasıl değiştirileceğini anlamak için bu konuyu kullanın.

Arşiv günlükleri için BSDS ' deki girişlere ekleyebilir, parolayı silebilir ve bu girişlerin parolasını değiştirebilirsiniz. Burada yalnızca örnekler gösterilir; kullanmak istediklerinizle gösterilen veri kümesi adlarını değiştirin. Yardımcı programın ayrıntıları için [Değişiklik günlüğü döküm yardımcı programı](#) başlıklı konuya bakın.

- [Arşiv günlüğü eklenmesi](#)
- [Arşiv günlüğünün silinmesi](#)
- [Arşiv günlüğünün parolasının değiştirilmesi](#)

Arşiv günlüğü eklenmesi

Bir nesnenin kurtarılması, var olan bir arşiv günlüğü veri kümesinin okunmaya bağlı olduğu durumlarda, BSDS ' nin bu veri kümesiyle ilgili bilgileri içermesi gerekir; böylece IBM MQ bunu bulabiliyor. BSDS ' de var olan bir arşiv günlüğü veri kümesiyle ilgili bilgileri kaydetmek için aşağıdaki adresi 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 çok birimdeki bir arşiv günlüğü veri kümesini silmek için şunu 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. Yazdırma günlüğü işlemi yardımcı programını kullanarak BSDS ' yi listeleyin.

2. Delete the entry for the archive log data set with the changed password, using the DELETE function of the CSQJU003 utility (see topic [Değişiklik günlüğü döküm yardımcı programı](#)).
3. Yeni bir arşiv günlüğü veri kümesi için veri kümesinin adını yazın. Use the NEWLOG function of the CSQJU003 utility (see topic [Değişiklik günlüğü döküm yardımcı programı](#)), and give the new password, the starting and ending RBAs, and the volume serial numbers (which can be found in the print log map utility output, see [Yazdırma günlüğü eşlemi yardımcı programı](#)).

Yeni arşiv günlüğü veri kümelerinin parolasını değiştirmek için şunu kullanın:

```
ARCHIVE PASSWORD= password
```

Yeni arşiv günlüğü veri kümelerinin parolalarını yerleştirmeyi durdurmak için aşağıdaki özellikleri kullanın:

```
ARCHIVE NOPASSWD
```

Not: Dış güvenlik yöneticiniz yoksa, yalnızca ARCHIVE yardımcı programı işlevini kullanın.



Günlüklere ve BSDS 'ye ilişkin üst düzey niteleyiciyi (HLQ) değiştirme

Üst düzey niteleyiciyi (HLQ) değiştirmek için gereken yordamı anlamak için bu konuyu kullanın.

Başlamadan önce

Günlüklerin ya da veri kümelerinin herhangi birini yeni veri kümelerine kopyalamadan önce, kuyruk yöneticisini olağan bir şekilde sona erdirmeniz gerekir. Bu, verilerin tutarlı olmasını ve yeniden başlatma işlemi sırasında herhangi bir kurtarma gerekmediğinden emin olun.

Bu görev hakkında

Bu görev, günlükler ve BSDS 'ler için HLQ' un nasıl değiştirileceği hakkında bilgi sağlar. Bunu yapmak için aşağıdaki adımları izleyin:

Yordam

1. Run the log print utility CSQJU004 to record the log data set information. Bu bilgilere daha sonra ihtiyaç duyulmaktadır.
2. Aşağıdakilerden birini yapabilirsiniz:
 - a) Yeniden adlandırılacak ve yeniden adlandırılacak olan günlük ve BSDS veri kümelerinde DSS yedekleme ve geri yükleme işlemi çalıştırma ya da
 - b) HLQ veri kümelerini yaratmak ve eski veri kümelerinden verileri kopyalamak için AMS DEFINE ve REPRO ' ları kullanın.
3. MSTR ve CHIN yordamlarını, yeni veri kümelerini işaret edecek şekilde değiştirin.
4. BSDS ' nin yeni kopyasındaki eski günlük bilgilerini CSQJU003kullanarak silin.
5. Yeni günlük verileri kümelerini, CSQJU003NEWLOG işlevini kullanarak yeni BSDS ' ye tanımlayın. Her bir günlüğe ilgili tüm bilgileri, HLQ ' dan ayrı olarak tutun.
6. Yeni BSDS, eski BSDS ' deki eski günlükler için kaydedilmiş bilgileri yansıtmalıdır. HLQ, değişen tek şey olmalı.

Sonraki adım

Kuyruk yöneticisini başlatmadan önce, eski ve yeni BSDS 'ler için CSQJU004 çıkışını karşılaştırın (HLQ' lar hariç), bunların TAMAMEN aynı görünmelerini sağlayın.

Not: Bu işlemleri gerçekleştirirken dikkatli olun. Yanlış eylemler kurtarılamaz durumlara yol açabilir. PRINT LOG MAP UTILITY çıkışını denetleyin ve kurtarma ya da yeniden başlatma için gereken tüm bilgilerin içerildiğinden emin olun.

z/OS **BSDS 'nin Kurtarılıyor**

IBM MQ ikili BSDS kipinde çalışıyorsa ve bir BSDS zarar görürse, IBM MQ 'yi tek bir BSDS kipine sokarsa, IBM MQ bir sorun olmadan çalışmaya devam eder (sonraki yeniden başlatma işlemi sürünceye kadar).

Ortamı ikili BSDS kipine döndürmek için:

1. Zarar görmüş BSDS 'leri yeniden adlandırmak ya da silmek ve zarar görmüş BSDS ile aynı adı taşıyan yeni bir BSDS tanımlamak için Access Method Services olanağını kullanın. Örneğin, thlqual.SCSQPROC içindeki CSQ4BREC işinde örnek denetim deyimleri bulunabilir.
2. Yeni ayrılan veri kümesindeki geçerli BSDS 'nin bir kopyasını oluşturmak ve ikili BSDS kipini yeniden yürürlüğe almak için IBM MQ komutunu RECOVER BSDS yayınlayın.

IBM MQ tekli BSDS kipinde çalışıyorsa ve BSDS zarar görmüşse ya da IBM MQ ikili BSDS kipinde çalışıyorsa ve her iki BSDS de zarar görmüşse, kuyruk yöneticisi durur ve BSDS veri kümeleri onarılmaya kadar yeniden başlatmaz. Bu durumda:

1. En son arşiv günlüğü veri kümesiyle ilişkilendirilmiş BSDS 'leri bulun. The data set name of the most recent archive log appears on the job log in the last occurrence of message CSQJ003I, which indicates that offload processing has been completed successfully. Bu yordamın geri kalanına hazırlık olarak, bu iletide belirtilen başarılı tüm arşivlerin bir günlüğünü tutmak iyi bir uygulamadır:
 - Arşiv günlükleri DASD ise, BSDS kullanılabilir herhangi bir DASD 'de ayrılır. BSDS adı, karşılık gelen arşiv günlüğü veri kümesi adına benzer; bu örnekte olduğu gibi, A 'dan B' ye kadar olan son niteleyicinin yalnızca ilk harfini değiştirin:

Arşiv günlüğü adı

CSQ.ARCHLOG1. **A** 0000001

BSDS kopyası adı

CSQ.ARCHLOG1. **E** 0000001

- Arşiv günlükleri manyetik bantsa, BSDS ilk arşiv günlüğü biriminin ilk veri kümesidir. BSDS daha sonraki birimlerde yinelenmez.
2. En son arşiv günlüğü veri kümesinde BSDS 'nin kopyası yoksa (örneğin, yükleme sırasında bir hata oluştuğundan), daha önceki boşaltma işlemlerinden BSDS' nin önceki bir kopyasını bulun.
 3. *hasarlı* BSDSs komutunu NEWNAME seçeneğiyle Access Method Services ALTER komutunu kullanarak yeniden adlandırın. Zarar görmüş bir BSDS 'yi silmek istiyorsanız, Erişim Yöntemi Hizmetleri DELETE komutunu kullanın. Her bir zarar görmüş BSDS için, yeni bir veri kümesi olarak yeni bir BSDS tanımlamak için Access Method Services (Erişim Yöntemi Hizmetleri) olanağını kullanın. thlqual.SCSQPROC içindeki CSQ4BREC işi, yeni bir BSDS tanımlamak için Access Method Services denetim deyimlerini içerir.
 4. Use the Access Method Services REPRO command to copy the BSDS from the archive log to one of the replacement BSDSs you defined in step “3” sayfa 463. Do not copy any data to the second replacement BSDS, you do that in step “5” sayfa 465.

- a. Yeni BSDS 'nin içeriğini yazdırın.

Yeni BSDS 'nin içeriğini yazdırmak için yazdırma günlüğü eşlemi yardımcı programını (CSQJU004) kullanın. Bu, kurtarma çalışmanıza devam etmeden önce yeni BSDS 'nin içeriğini gözden geçirmenizi sağlar.

- b. Yeni BSDS 'de arşiv günlüğü veri kümesi dökümünün güncellenmesi.

Yazdırma günlüğü eşlemi yardımcı programının çıkışını inceleyin ve yeni BSDS 'nin, BSDS' nin kopyalandığı arşiv günlüğünün kaydını içermemesini denetleyin. Yeni BSDS eski bir kopyaysa, dökümü son zamanlarda daha fazla yaratılmış tüm arşiv günlüğü veri kümelerini içermeyebilir. Geçerli altsistem dökümünü yansıtmak için arşiv günlüğü veri kümelerinin BSDS dökümünün güncellenmesi gerekir.

Yedek BSDS 'yi güncellemek için, değişiklik günlüğü döküm yardımcı programını (CSQJU003) NEWLOG deyimini kullanarak, BSDS' nin kopyalandığı arşiv günlüğünün kaydını ekleyin. Arşiv günlüğü veri kümesi parola korumalıysa, NEWLOG işlevinin PASSWORD (PASSWORD) seçeneğini kullanın. Ayrıca, arşiv günlüğü veri kümesi kataloğa alındıysa, NEWLOG işlevinin CATALOG seçeneğinin CATALOG=YES olarak ayarlandığından emin olun. BSDS kopyasından sonra yaratılmış ek arşiv günlüğü veri kümelerini eklemek için NEWLOG deyimini kullanın.

c. Yedek BSDS ' deki parolaları güncelleyin.

BSDS, arşiv günlüğü veri kümelerine ve etkin günlük veri kümelerine ilişkin parolalar içerir. Yeni BSDS ' deki parolaların, kuruluşunuz tarafından kullanılan geçerli parolaları yansıttığından emin olmak için, PASSWORD seçeneğini kullanarak değişiklik günlüğü döküm ARCHVE yardımcı programı işlevini kullanın.

d. Yedek BSDS ' de etkin günlük veri kümesi dökümünü güncelleyin.

Olağan dışı durumlarda, BSDS kopyalandığı için kuruluşunuz etkin günlük veri kümelerini eklemiş, silmiş ya da yeniden adlandırmış olabilir. Bu durumda, yeni BSDS, kuruluşunuzun şu anda kullanılmakta olan etkin günlük verileri kümesinin gerçek sayısını ya da adlarını yansıtmaz.

Yeni BSDS günlük dökümünden etkin bir günlük veri kümesini silmeniz gerekiyorsa, değişiklik günlüğü döküm 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 belirtilip belirtilmediğinden emin olun. Etkin günlük veri kümesi parola korumalıysa, PASSWORD seçeneğini kullanın.

Yeni BSDS günlük dökümünde belirlenen etkin bir günlük verilerini yeniden adlandırmanız gerekiyorsa, REWLOG işlevini kullanarak, günlük döküm alma yardımcı programı DELETE işlevini kullanın. RBA aralığının NEWLOG işlevinde doğru belirtilip belirtilmediğinden emin olun. Etkin günlük veri kümesi parola korumalıysa, PASSWORD seçeneğini kullanın.

e. Yedek BSDS ' deki etkin günlük RBA aralıklarını güncelleyin.

Daha sonra, kuyruk yöneticisi yeniden başlatıldığında, BSDS 'de listelenen etkin günlük veri kümelerinin RBU' ları gerçek etkin günlük veri kümelerinde bulunan RBA ' larla karşılaştırır. RBA ' nın kabul edilmemesi durumunda, kuyruk yöneticisi yeniden başlatılmaz. Bu sorun, BSDS ' nin eski bir kopyası kullanıldığında büyütüldüğünde sorun ortaya çıktı. Bu sorunu çözmek için, gerçek etkin günlük veri kümelerindeki RBA ' ları kullanarak BSDS' de bulunan RBA ' yı ayarlamak için değişiklik günlüğü döküm yardımcı programını (CSQJU003) kullanın. Bunu aşağıdaki gibi yapın:

- Etkin günlük verileri kümesinin özet raporunu yazdırmak için yazdırma günlüğü kayıtları yardımcı programını (CSQ1LOGP) kullanın. Bu, RBA ' yı başlatma ve sona erdirmeye olanağını gösterir.
- Tüm etkin günlük veri kümelerinin RBA ' ları bilindiğinde, RBA aralıklarıyla gerçek RBA aralıklarını az önce yazdığınız RBA aralıklarıyla karşılaştırılır.

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 aralığı ayarına sahip olması gereken her bir etkin günlük veri kümesi için, değiştirme günlüğü döküm yardımcı programı DELETE işlevini kullanarak, yeni BSDS ' deki dökümden etkin günlük verilerini silin. Daha sonra, etkin günlük verilerini BSDS ' ye yeniden tanımlamak için NEWLOG işlevini kullanın. Etkin günlük veri kümeleri parola korumalıysa, NEWLOG işlevinin PASSWORD (PASSWORD) seçeneğini kullanın.

f. If only two active log data sets are specified for each copy of the active log, IBM MQ can have difficulty during queue manager restart. Etkin günlük veri kümelerinden biri dolunca ve doldurulmadığında sorun ortaya çıkabilir, ikinci etkin günlük veri kümesi dolmaya yakın olduğunda. Bu durumda, etkin günlüğün her bir kopyası için yeni bir etkin günlük verileri kümesi ekleyin ve yeni BSDS günlük dökümünde ayarlanan her 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 günlük döküm yardımcı programı NEWLOG işlevini kullanın. NEWLOG deyimindeki RBA aralıklarını belirtmenize gerek yoktur. Ancak, etkin günlük veri kümeleri parola korumalıysa, NEWLOG işlevinin PASSWORD (PASSWORD) seçeneğini kullanın. Example control statements to accomplish this task can be found in job CSQ4LREC in thlqual.SCSQPROC.

5. Güncellenen BSDS ' leri ikinci yeni BSDS veri kümesine kopyalayın. BSDSler artık aynı.

Bu noktada ikinci yedek BSDS ' nin içeriğini yazdırmak için yazdırma günlüğü eşlemi yardımcı programını (CSQJU004) kullanın.

6. Yürürlükteki etkin günlük veri kümenizi kaybetmeniz durumunda ne yapacağınıza ilişkin bilgi için [Etkin günlük sorunları](#) başlıklı konuya bakın.
7. Yeni oluşturulan BSDS ' yi kullanarak kuyruk yöneticisini yeniden başlatın. IBM MQ , yürürlükteki 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 465](#)
- [“Kuyruk yöneticisine sayfa kümesi eklenmesi” sayfa 466](#)
- [“Sayfa kümelerinden biri dolduğunda ne yapmanız gerekir?” sayfa 466](#)
- [“Sayfa kümelerindeki yükleri dengeleme” sayfa 466](#)
- [Sayfa kümesinin büyüklüğünü artırma](#)
- [“Sayfa kümesinin küçültülmesi” sayfa 470](#)
- [“Bir sayfa kümesinin yeniden tanıtılması” sayfa 471](#)
- [“Sayfa kümelerini nasıl yedeklemeniz ve kurtarmanız” sayfa 471](#)
- [“Sayfa kümelerinin nasıl silineceği” sayfa 475](#)
- [“CSQUTIL kullanarak kuyrukları yedekleme ve geri yükleme” sayfa 475](#)

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 kullanılır 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 468](#).
- Sayfa kümesini, 4 GB 'den 64 GB' ye kadar genişletecek 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. Taşınmakta 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 467. adımda yaratılan geçici kuyruğa dayalıdır.

8. Aşağıdaki komutu kullanarak iletileri yeni kuyruğa geri taşıyın:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

9. "3" sayfa 467 . 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 yerisini koyun.

Şekil 37 sayfa 468 , bir yük dengeleme işinden alınan bir çıkarmayı gösterir.

```
//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 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 37. 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 özniteliktir.

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=RADIQ (genişletilmiş biçim gereklidir) ya da DSNTYPE=EXTPREF (genişletilmiş biçim tercih edilir) kodlaması.
- DD deyimindeki LIKE= parametresinin, var olan bir genişletilmiş 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 genişletme \(COPYPAGE\)](#) .
5. Kuyruk yöneticisini yeniden başlatın.
6. Sayfa kümesini, sistem genişletmesini kullanacak şekilde değiştirin ve yürürlükteki ayırmanın ötesine büyümeye devam etmesini sağlayın.

Aşağıdaki JCL, örnek Erişim 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 genişletmeye izin verecek şekilde değiştirilmesi

ALTER PSID komutunu EXPAND (USER) ya da EXPAND (SYSTEM) seçenekleriyle birlikte kullanın. Sayfa kümelerinin genişletilmesiyle ilgili genel bilgi için [ALTER PSID ve Sayfa kümesinin genişletilmesi \(COPYPAGE\)](#) başlıklı konuya bakın.

İletilerin yeni, daha büyük bir sayfa kümesine taşınması

Bu teknik, kuyruk yöneticisinin durdurulmasını ve yeniden başlatılmasını içerir. Bu, yeniden başlatma sırasında paylaşılan kuyruklarda olmayan kalıcı olmayan iletileri siler. Silinmesini istemediğiniz kalıcı olmayan iletilerinizi varsa, bunun yerine yük dengelemeyi kullanın. Daha fazla ayrıntı için bkz. "[Sayfa kümelerindeki yükleri dengeleme](#)" sayfa 466. Bu tanımda, genişletmek istediğiniz sayfa kümesine *kaynak* sayfa kümesi denir; yeni, daha büyük sayfa kümesine *hedef* sayfa kümesi denir.

Aşağıdaki adımları izleyin:

1. Kuyruk yöneticisini durdurun.
2. Hedef sayfa kümesini, daha büyük bir ikincil kapsam değeriyle kaynak sayfa kümesinden daha büyük olduğundan 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 genişletme \(COPYPAGE\)](#) .
5. Aşağıdakilerden birini yaparak, hedef sayfa kümesini kullanarak kuyruk yöneticisini yeniden başlatın:
 - Kuyruk yöneticisi tarafından başlatılan görev yordamını, hedef sayfa kümesine gönderme yapacak şekilde değiştirin.
 - Kaynak sayfa kümesini silmek için Erişim 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 oluşturmuş olduğunuzdan 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 yöneticinizi yeniden başlatmaktır (bkz. "[Kuyruk yöneticisi yeniden kullanıma hazırlanıyor](#)" sayfa 494). 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 470. adımda yarattığınız sayfa kümesinde CSQUTIL ' in FORMAT TYPE (NEW) işlevini çalıştırın.
5. "[3](#)" sayfa 470. adımda yaratılan sayfa kümesini kullanarak kuyruk yöneticisini yeniden başlatın.
6. "[2](#)" sayfa 470. 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 işlemi 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.

z/OS Sayfa kümelerini nasıl yedeklemeniz ve kurtarmanız

Yedeklemenin ve iyileşmenin kullanabileceği farklı mekanizmalar var. Bu mekanizmaları anlamak için bu konuyu kullanın.

Bu bölümde aşağıdaki konular anlatılır:

- [“Paylaşılmayan kaynaklar için kurtarma noktası yaratılması” sayfa 472](#)
- [“Sayfa kümeleri yedekleniyor” sayfa 473](#)

- “Sayfa kümeleri kurtarılıyor” sayfa 474
- Sayfa kümelerinin nasıl silineceği

Paylaşılan kaynaklar için kurtarma noktasının nasıl oluşturulacağı hakkında bilgi için bkz. “Paylaşılan kuyrukların kurtarılıyor” sayfa 480.

Paylaşılmayan kaynaklar için kurtarma noktası yaratılması

IBM MQ , nesnelere ve paylaşılmayan kalıcı iletileri her ikisi de varsa, yürürlükteki durumuna kurtarabilir:

1. Daha önceki bir noktadan sayfa kümelerinin kopyaları var.
2. Bu noktadan kurtarma işlemini gerçekleştirmek için tüm IBM MQ günlükleri kullanılabilir.

Bunlar, paylaşılmayan kaynaklar için bir kurtarma noktasını temsil eder.

Her iki nesne ve ileti, sayfa kümelerinde tutulur. Aynı sayfa kümesinde, farklı kuyruklardan birden çok nesne ve ileti var olabilir. Kurtarma amacıyla, nesnelere ve iletiler yalıtılmış olarak yedeklenemez; bu nedenle, verilerin doğru şekilde kurtarılmasını sağlamak için bir sayfa kümesinin bir bütün olarak yedeklenmeleri 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 yükleyerek ve kuyruk yöneticisini yeniden başlatarak sayfa kümesini kurtarabilirsiniz. IBM MQ , günlük değişikliklerini, yedek kopyanın noktasından sayfa kümesine uygular.

Kurtarma noktası oluşturmanın iki yolu vardır:

Tam Olarak Yedekle

Tüm güncellemeleri sayfa kümeleri üzerinde zorlayan kuyruk yöneticisini durdurun.

Bu, yalnızca yedeklenen sayfa kümesi veri kümelerini ve bu noktadan sonra günlükleri kullanarak, kurtarma noktasından yeniden başlamanıza olanak tanır.

Yaklaşık yedekleme

Kuyruk yöneticisini durdurmadan, sayfa kümelerinin *bulanık* yedek kopyalarını alın.

Bu yöntemi kullanırsanız ve ilişkili günlükleriniz daha sonra zarar görürse ya da kaybolursa, yaklaşık sayfa kümesi yedek kopyalarını kurtarmak için kullanamazsınız. Bunun nedeni, yaklaşık sayfa kümesi yedek kopyalarının, kuyruk yöneticisinin durumunun tutarsız bir görünümünü içerdiğinden ve kullanılabilir olan günlüklere bağımlı olması olabilir. Günlükler kullanılamıyorsa, altsistem etkin değilken alınan son yedek sayfa kümesi kopyalarına geri dönmeniz gerekir (Yöntem 1) ve veri kaybını o zaman kabul edin.

Yöntem 1: Tam yedekleme

Bu yöntem, kuyruk yöneticisinin kapatılmasını içerir. Bu işlem, sayfa kümelerinin tüm güncelleştirmelerini, sayfa kümelerinin tutarlı bir durumda olması için ayarlar.

1. Kuyruk yöneticisini kullanan tüm IBM MQ uygulamalarını durdurun (önce tamamlanmasına izin verir). Bu, örneğin erişim güvenliği ya da kuyruk ayarları değiştirilerek yapılabilir.
2. Tüm etkinlik tamamlandığında, belirsiz kurtarma birimlerini görüntüler ve çözün. (DISPLAY CONN ve RESOLVE INDOUBT başlıklı konu anlatıldığı gibi, DISPLAY CONN ve RESOLVE INDOUBT komutlarını kullanın.)

Bu, sayfa kümelerini tutarlı bir duruma getirir; bunu yapmazsanız, sayfa kümeleriniz tutarlı olmayabilir ve yaklaşık bir yedekleme işlemi gerçekleştirmekte etkili bir durumunuz vardır.

3. En son günlük verilerinin günlük verileri kümelerine yazıldığından emin olmak için ARCHIVE LOG komutunu verin.
4. STOP QMGR MODE (QUIESCE) komutunu verin. En düşük RBA değerini CSQI024I ya da CSQI025I iletilerinde kaydedin (daha fazla bilgi için bkz. CSQI024I ve CSQI025I). RBA değeri ile belirtilen günlük veri kümelerini geçerli günlük veri kümesine kadar olan günlük veri kümelerini alıkoymak gerekir.

5. Tüm kuyruk yöneticisi sayfa kümelerinin yedek kopyalarını alın (bkz. [“Sayfa kümeleri yedekleniyor” sayfa 473](#)).

Yöntem 2: Fuzzy yedekleme

Bu yöntem, kuyruk yöneticisini sona erdirmeyi içermez. Bu nedenle, yedekleme işlemi sırasında sanal saklama arabelleklerinde güncellemeler olabilir. Bu, sayfa kümelerinin tutarlı bir durumda olmadığı ve yalnızca günlüklerle kurtarma için kullanılabilmesi anlamına gelir.

1. Issue the DISPLAY USAGE TYPE(ALL) command, and record the RBA value in the CSQI024I or CSQI025I messages (see [CSQI024I](#) and [CSQI025I](#) for more information).
2. Sayfa kümelerinin yedek kopyalarını alın (bkz. [“Sayfa kümeleri yedekleniyor” sayfa 473](#)).
3. En son günlük verilerinin günlük verileri kümelerine yazıldığından emin olmak için ARCHIVE LOG komutunu verin. Kurtarma noktasından yeniden başlatmak için, günlük verileri kümelerini RBA değeri ile gösterilen günlük verileri kümesinden başlayarak geçerli günlük veri kümesine kadar alıkoymak gerekir.

Sayfa kümeleri yedekleniyor

Bir sayfa kümesini kurtarmak için, IBM MQ ' un günlüğe ne kadar geri döneceği hakkında bilgi sahibi olması gerekir. IBM MQ maintains a log RBA number in page zero of each page set, called the *kurtarma günlüğü sıra numarası* (LSN). This number is the starting RBA in the log from which IBM MQ can recover the page set. Bir sayfa kümesini yedeklediğinizde, bu numara da kopyalanır.

Kopya daha sonra sayfa kümesini kurtarmak için kullanıldıysa, IBM MQ bu RBA değerinden geçerli RBA ' ya kadar tüm günlük kayıtlarına erişime sahip olmalıdır. That means you must keep enough of the log records to enable IBM MQ to recover from the oldest backup copy of a page set you intend to keep.

Sayfa kümelerini kopyalamak için ADRDSSU COPY işlevini kullanın.

Ek bilgi için [COPY DATASET Command Syntax for Logical Data Set](#) belgesine 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 kopyaladığınızda, önce sayfa sıfırının sayfa sıfırını kopyalayan bir kopyalama yardımcı programı kullanmanız gerekir. Bunu yapmazsanız, sayfa küğünüzdeki verileri bozabilirsiniz.

Bir sayfa kümesinin dinamik olarak genişletilmesi işlemi kesintiye uğratılırsa, örneğin kaybedilen sisteme güç vererek, bir sayfa kümesinin yedeğini almak için ADRDSSU ' yu kullanmaya devam edebilirsiniz.

Bir Access Method Services IDCAMS LISTCAT ENT('page set data set name') ALLOCgerçekleştirdiyseniz, HI-ALLOC-RBA 'in HI-USED-RBA' den daha yüksek olduğunu görürsünüz.

Bu sayfa bir sonraki doldurulmuyorsa, mümkünse yine uzatılır ve yüksek kullanılan RBA ile en yüksek ayrılmış RBA arasındaki sayfalar başka bir yeni kapsam ile birlikte kullanılır.

Nesne tanımlarının yedeklenmesi

Nesne tanımlarınızın kopyalarını da yedeklemeniz gerekir. Bunu yapmak için, CSQUTIL COMMAND işlevinin MAKEDEF özelliğini kullanın (Verilen komutlarda IBM MQ (COMMAND) olarak açıklanmıştır).

Kuyruk yöneticinizin bir yedek kopyasını alsanız ve en güncel sürümü alıkoymak için nesne tanımlarınızı yedekleyin.

Sayfa kümeleri kurtarılıyor

Bir hata nedeniyle kuyruk yöneticisi sona erdirildiyse, kuyruk yöneticisi olağan durumda yeniden başlatma işlemi sırasında tüm kurtarma işlemleri yeniden başlatılabilir. Ancak, bu tür bir kurtarma işlemi, sayfa kümelerine ya da günlük veri kümelerine uygun değilse, bu tür bir kurtarma işlemi mümkün değildir. Artık kurtarabileceğiniz kapsam, sayfa kümelerinin yedek kopyalarının ve günlük veri kümelerinin kullanılabilirliğine bağlıdır.

Bir kurtarma noktasından yeniden başlatmak için aşağıdakilerin sahip olması gerekir:

- Kurtarılacak sayfa kümesinin yedek kopyası.
- “Yöntem 2: Fuzzy yedekleme” sayfa 473’ünde açıklanan "bulanık" yedekleme işlemi, kaydedilen RBA değerini içeren günlük verileri kümesini, ARCHIVE LOG komutuyla yapılan günlük verileri kümesini ve bunların arasındaki tüm günlük veri kümelerini de kullandıysanız.
- Tam yedekleme kullandıysanız, ancak ARCHIVE LOG komutunun ardından günlük veri kümeleriniz yoksa, CSQUTIL yardımcı programının FORMAT TYPE (REPLACE) işlevini tüm sayfa kümelerine karşı **çalıştırmanıza** gerek yoktur.

Bir sayfayı yürürlükteki durumuna geri yüklemek için, ARCHIVE LOG komutundan sonra tüm günlük veri kümelerinin ve kayıtlarının da olması gerekir.

Bir sayfa kümesinin kurtarılmasına ilişkin iki yöntem vardır. Her iki yöntemi de kullanmak için kuyruk yöneticisi durdurulmalıdır.

Basit kurtarma

Bu yöntem daha basit bir yöntemdir ve çoğu kurtarma durumu için uygundur.

1. Yedekten geri yüklemek istediğiniz sayfa kümesini silin.
2. Yedek kopyadan sayfayı kurtarmak için ADRDSSU COPY işlevini kullanın.

Diğer bir seçenek olarak, yedek kopyanın adını özgün adla yeniden adlandırabilir ya da kuyruk yöneticisi yordamınızdaki CSQP00xx DD deyimini, yedek sayfa kümenizi işaret edecek şekilde değiştirebilirsiniz. Ancak, sayfa kümesini kaybederseniz ya da bozarsanız, geri yükleme işlemi için yedek bir kopyaya sahip olmanız gerekir.

3. Kuyruk yöneticisini yeniden başlatın.
4. Kuyruk yöneticisi başarıyla yeniden başlatıldığında, uygulamalarınızı yeniden başlatabilirsiniz.
5. Geri yüklenen sayfaya ilişkin olağan yedekleme yordamlarınızı yeniden yürürlüğe girin.

Gelişmiş kurtarma

Bu yöntem, kurtarılacak büyük bir sayfa kümeniz varsa ya da son yedek kopyanın alınmasından bu yana sayfa kümesinde çok fazla etkinlik varsa, performans avantajları sağlar. Ancak, basit yöntemle göre daha fazla el ile müdahale gerektirir; bu da hata riskini ve kurtarma işlemi gerçekleştirme için gereken süreyi artırabilir.

1. Yedekten geri yüklemek istediğiniz sayfa kümesini silin ve yeniden tanımlayın.
2. Sayfanın yedek kopyasını yeni sayfa kümesine kopyalamak için ADRDSSU ' yu kullanın. Dinamik olarak genişletilebilmesi için yeni sayfa kügenizi ikincil kapsam değeriyle tanımlayın.

Diğer bir seçenek olarak, yedek kopyanın adını özgün adla yeniden adlandırabilir ya da kuyruk yöneticisi yordamınızdaki CSQP00xx DD deyimini, yedek sayfa kümenizi işaret edecek şekilde değiştirebilirsiniz. Ancak, sayfa kümesini kaybederseniz ya da bozarsanız, geri yükleme işlemi için yedek bir kopyaya sahip olmanız gerekir.

3. Kuyruk yöneticinizin CSQINP1 tanımlarını değiştirerek, sayfa kümesiyle ilişkilendirilmiş arabellek havuzunu olabildiğince büyük hale getirmek üzere değiştirin. Arabellek havuzunu büyüktürerek, tüm değiştirilen sayfaları arabellek havuzunda saklayabilir ve sayfa kümesinin G/Ç miktarını azaltabilir.
4. Kuyruk yöneticisini yeniden başlatın.
5. Kuyruk yöneticisi başarıyla yeniden başlatıldığında, o sayfa kümesi için olağan arabellek havuzu tanımlamasını kullanarak, kuyruk yöneticisi susturulmuş durumdan (quiesce kullanılarak) durdurup yeniden başlatın. Bu ikinci yeniden başlatma başarıyla tamamlandıktan sonra, uygulamalarınızı yeniden başlatabilirsiniz.
6. Geri yüklenen sayfaya ilişkin olağan yedekleme yordamlarınızı yeniden yürürlüğe girin.

Kuyruk yöneticisi yeniden başlatıldığında ne olur

Kuyruk yöneticisi yeniden başlatıldığında, sayfa kümesinin yeniden başlatma noktasından başlayarak, günlükte kayıtlı olan sayfa kümesinde yapılan tüm değişiklikleri uygular. IBM MQ , birden çok sayfa kümesini bu şekilde kurtarabilir. Ortam kurtarma işlemi sırasında, sayfa kümesi dinamik olarak genişletiliyor.

Yeniden başlatma işlemi sırasında, IBM MQ , RBA günlüğünü aşağıdaki değerden en düşük değeri olarak başlayacak şekilde saptar:

- Her sayfa kümesi için denetim noktası günlük kaydından kurtarma LSN ' si.
- Her bir sayfa kümesindeki sayfa sıfırından kurtarma LSN ' yi ayarlayın.
- Yedeklemenin alındığı sırada sistemdeki en eski tamamlanmamış kurtarma biriminin RBA 'sını.

Tüm nesne tanımlamaları sayfa kümesi sıfır olarak saklanır. İletiler, kullanılabilir herhangi bir sayfa kümesinde saklanabilir.

Not: Sayfa kümesi sıfır kullanılamıyorsa, kuyruk yöneticisi yeniden başlayamaz.

Sayfa kümelerinin nasıl silineceği

DELETE PSID komutunu kullanarak bir sayfa kümesini silersiniz; bu komuta ilişkin ayrıntılar için [DELETE PSID](#) başlıklı konuya bakın.

Herhangi bir depolama sınıfı tarafından hala gönderme yapılan bir sayfa kümesini silemezsiniz. Bir sayfa kümesine hangi depolama sınıflarının başvuruda olduğunu öğrenmek için DISPLAY STGCLASS seçeneğini kullanın.

Veri kümesi IBM MQ ' den ayrıdır, ancak silinmez. Bu, ileride kullanılmak üzere kullanılabilir durumda kalır ya da z/OS olanağı kullanılarak silinebilir.

Kuyruk yöneticinize ilişkin başlatılan görev yordamından sayfa kümesini kaldırın.

Remove the definition of the page set from your CSQINP1 initialization data set.

z/OS CSQUTIL kullanarak kuyrukları yedekleme ve geri yükleme

CSQUTIL kullanarak yedekleme ve geri yükleme hakkında daha fazla bilgi almak için bu konuyu başvuru olarak kullanın.

CSQUTIL yardımcı program işlevlerini kullanarak kuyrukları yedeklemek ve geri yüklemek için kullanabilirsiniz. Kuyruğu yedeklemek için, iletileri bir kuyruktan veri kümesine kopyalamak için COPY YA DA SKOPI işlevini kullanın. Kuyruğu geri yüklemek için, LOAD ya da SLOAD tamamlayıcı işlevini kullanın. Daha fazla bilgi için bkz. [IBM MQ yardımcı programı](#).

z/OS Arabellek havuzlarının yönetilmesi

Arabellek havuzlarınızı değiştirmek ya da silmek için bu konuyu kullanın.

Bu konuda, arabellek havuzlarının nasıl değiştirileceği ve silineceği ele alınmıştır. Bu kısımları içerir:

- “Arabellek havuzundaki arabelleklerin sayısını deęiřtirme” sayfa 476
- “Arabellek Havuzunun Silinmesi” sayfa 476

Buffer pools are defined during queue manager initialization, using [ARABELLEK HAVUZU TANIMLA](#) commands issued from the initialization input data set CSQINP1. Kuyruk yöneticisi çalışırken, bu konuda ayrıntılı işlemler kullanılarak, bunların öznitelikleri iş gereksinimlerine yanıt olarak deęiřtirilebilir. Kuyruk yöneticisi, denetim noktası günlük kayıtlarındaki yürürlükteki arabellek havuzu özniteliklerini kaydeder. CSQINP1 içindeki arabellek havuzu tanımlaması REPLACE öznitelięini içermiyorsa, bunlar sonraki kuyruk yöneticisinin yeniden başlatılmasındaki otomatik olarak geri yüklenir.

Yürürlükteki arabellek özniteliklerini görüntülemek için [DISPLAY USAGE](#) komutunu kullanın.

Ayrıca, arabellek havuzlarını DSN seçeneęiyle [DEFE PSID](#) komutunu kullanarak devingen olarak tanımlayabilirsiniz.

Arabellek havuzlarını dinamik olarak deęiřtirirseniz, kullanıma hazırlama verileri kümesinde (CSQINP1) tanımlarını da güncellemeniz gerekir.

Sayfa kümelerinin, depolama sınıflarının, arabelleklerin ve arabellek havuzlarının bir açıklaması ve geçerli başařım açısından dikkate alınması gereken bazı noktalar için bkz. [Planlama on z/OS](#) .

Not: Arabellek havuzları, önemli depolamayı kullanır. Bir arabellek havuzunun büyüklüğünü artırdığınızda ya da yeni bir arabellek havuzu tanımladığınızda, yeterli saklama alanının kullanılabilir olduğundan emin olun. Daha fazla bilgi için bakınız: [Address space storage](#).

Arabellek havuzundaki arabelleklerin sayısını deęiřtirme

Arabellek havuzu çok küçükse, koşul konsolda [CSQP020E](#) iletisine neden olabilir; ALTER BUFPOOL komutunu kullanarak bu durumda daha fazla arabellek ayırabilirsiniz:

1. Günlükteki [CSQY220I](#) iletilerine bakarak yeni arabellekler için ne kadar alanın kullanılabilir olduğunu belirleyin. Kullanılabilir alan MB olarak raporlanır. Arabellek büyüklüğü 4 KB olduğu için, kullanılabilir alanın her bir MB ' si 256 arabellek ayırmanızı sağlar. Dięer görevler için gerekli olduğu için, tüm boş alanı arabelleklere ayırmayın.

Arabellek havuzu sabit 4 KB sayfaları kullanıyorsa, bu, PAGECLAS öznitelięinin FIXED4KB' dir, LPAR üzerinde yeterli miktarda gerçek saklama alanı olduğundan emin olun.

2. Bildirilen boş alan yetersizse, komutu kullanarak başka bir arabellek havuzundan bazı arabellekleri serbest bırakın.

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

Burada *buf-havuz-tnt* , alanı geri almak istediğiniz arabellek havuzudur ve *tamsayı* , bu arabellek havuzuna ayrılacak yeni arabellek sayısıdır; bu havuz, kendisine ayrılan başlangıçtaki arabelleklerin sayısından küçük olmalıdır.

3. Arabellekleri, komutu kullanarak genişletmek istediğiniz arabellek havuzuna ekleyin

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

Burada *buf-havuz-tnt* , genişletilecek arabellek havuzu ve *tamsayı* , bu arabellek havuzuna ayrılacak yeni arabellek sayısıdır; bu havuz, kendisine ayrılan başlangıçtaki arabelleklerin sayısından büyük olmalıdır.

Arabellek Havuzunun Silinmesi

Bir arabellek havuzu artık herhangi bir sayfa kümesi tarafından kullanılmadığında, bu havuz için ayrılan sanal saklama alanını serbest bırakmak için bu havuzu silin.

[ARABELLEK HAVUZUNU SIL](#) komutunu kullanarak bir arabellek havuzunu silersiniz. Bu arabellek havuzunu kullanan herhangi bir sayfa kümesi varsa komut başarısız olur.

Sayfa kümelerinin nasıl silineceği hakkında bilgi için bkz. [“Sayfa kümelerinin nasıl silineceği” sayfa 475](#) .

z/OS Managing queue sharing groups and shared queues on z/OS

IBM MQ , paylaşılan kaynak tiplerini, örneğin kuyruk paylaşım grupları, paylaşılan kuyruklar ve bağlaşım olanağını kullanabilir. Bu paylaşılan kaynakları yönetmek için gereken yordamları gözden geçirmek için bu konuyu kullanın.

Bu bölümde aşağıdaki konulara ilişkin bilgiler yer almaktadır:

- [“Kuyruk paylaşım gruplarını yönetme” sayfa 477](#)
- [“Paylaşılan kuyrukların yönetilmesi” sayfa 480](#)
- [“Grup nesnelere yönetilmesi” sayfa 485](#)
- [“Bağlaşım tesisinin yönetilmesi” sayfa 485](#)

z/OS 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 477](#)
- [“Kuyruk paylaşım grubuna kuyruk yöneticisi eklenmesi” sayfa 478](#)
- [“Kuyruk yöneticisini kuyruk paylaşım grubundan kaldırma” sayfa 479](#)
- [“Db2 çizelgelerinden bir kuyruk paylaşım grubunun kaldırılması” sayfa 479](#)
- [“Db2 tanımlamalarının tutarlılığının doğrulanması” sayfa 480](#)

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 478](#) içindeki adımları izleyerek kuyruk paylaşım grubuna bir kuyruk yöneticisi ekleme
5. [“Bağlaşım olanağı yapısı eklenmesi” sayfa 485](#) 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 bağlaşım olanağı (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 kurtarılabilir 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 uyarlanmasına 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 479 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 kuyrukları nasıl kurtaracağını, taşımayı ve yeni düzeye geçirmeyi anlamak için bu konuyu kullanın.

Bu bölümde aşağıdaki görevler anlatılır:

- "Paylaşılan kuyrukların kurtarılıyor" sayfa 480
- "Paylaşılan kuyrukların taşınması" sayfa 481
- "Paylaşılmayan kuyrukların paylaşılan kuyruklara geçirilmesi" sayfa 483
- Db2 bağlantısının askıya yapılması

Paylaşılan kuyrukların kurtarılıyor

IBM MQ paylaşılan kuyruklardaki kalıcı iletileri aşağıdaki durumlarda kurtarabilir:

- İletilerin bulunduğu CF yapılarının yedekleri gerçekleştirildi.
- Kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine ilişkin tüm günlükler, yedeklerin alındığı noktadan kurtarma işlemlerini gerçekleştirmek için kullanılabilir.
- Db2 kullanılabilir ve yapı yedekleme tablosu, en son CF yapısı yedeklediğinden daha yeni bir şekilde yedeklenmektedir.

Paylaşılan kuyrukte bulunan iletiler bir bağlaşım olanağı (CF) yapısında saklanır. Kalıcı iletiler, paylaşılan kuyruklara ve paylaşılmayan kuyruklara ilişkin kalıcı iletiler gibi, kuyruk yöneticisi günlüğüne kopyalanabilir. MQSC BACKUP CFSTRUCT ve RECOVER CFSTRUCT komutlarının, bir bağlaşım olanağı hatasının olası olmayan olayında bir CF yapısının kurtarılmasına olanak sağlanması için sağlanmıştır. Bu durumda, etkilenen yapıdaki kalıcı olmayan iletiler kaybedilir, ancak kalıcı iletiler kurtarılabilir. Yapı kurtarıncaya kadar, yapıyı kullanan başka bir uygulama etkinliği engellenir.

Kurtarma işlemini etkinleştirmek için, MQSC BACKUP CFSTRUCT komutunu kullanarak, bağlaşım olanağı listesi yapılarını sık sık yedeklemeniz gerekir. CF yapısındaki iletiler, yedeklemeyi yapan kuyruk yöneticisinin etkin günlük verileri kümesine yazılır. It writes a record of the backup to Db2: the name of the CF structure being backed up, the name of the queue manager doing the backup, the RBA range for this backup on that queue manager log, and the backup time. Paylaşılan kuyrukları etkin olarak kullanmamış olsanız bile, CF liste yapılarını yedeklemeniz, örneğin, gelecekte kullanmak üzere bir kuyruk paylaşım grubu kurmanız durumunda.

Kurtarma işlemini gerçekleştirebilen kuyruk yöneticisine bir MQSC RECOVERY CFSTRUCT komutu vererek, bir CF yapısını kurtarabilirsiniz; kuyruk paylaşım grubunda kuyruk yöneticisini kullanabilirsiniz. Kurtarılacak tek bir CF yapısını belirtebilir ya da birden çok CF yapısını aynı anda kurtarabilirsiniz.

Daha önce de belirtildiği gibi, CF listesi yapılarınızı sık sık yedeklemenizi, aksi takdirde bir CF yapısının kurtarılamaz uzun sürebileceğini ifade eder. Ayrıca, kurtarma işlemi iptal edilemez.

Paylaşılan bir kuyruğun tanımı bir Db2 veritabanında tutulur ve bu nedenle, standart Db2 veritabanı yordamları kullanılarak gerekiyorsa kurtarılabilir. Ek bilgi için [Paylaşılan kuyruklar ve kuyruk paylaşım grupları](#) başlıklı konuya bakın.

Paylaşılan kuyrukların taşınması

Bu bölümde, paylaşılan bir kuyruğu bir bağlaşım olanağı yapısından diğerine taşıyarak yük dengelemenin nasıl gerçekleştirileceği açıklanmaktadır. Ayrıca, paylaşılmayan bir kuyruğun paylaşılan bir kuyruğa nasıl taşınacağını ve paylaşılan bir kuyruğun paylaşılmayan bir kuyruğun nasıl taşınacağını da açıklar.

Bir kuyruğu taşıdığınızda, yordamın bir parçası olarak geçici bir kuyruk tanımlamanız gerekir. Bunun nedeni, her kuyruğun benzersiz bir ada sahip olması gerektiğinden, kuyrukların farklı kuyruk atımları olsa da aynı adı iki kez kuyruklamayabilmelidir. IBM MQ , aynı adı taşıyan ("2" sayfa 481 adımı olduğu gibi) iki kuyruk olmasına tahammül eder, ancak kuyrukları kullanamazsınız.

- Bir kuyruğun bir bağlaşım olanağı yapısından başka bir yapıya taşınması
- Paylaşılmayan kuyruğun paylaşılan bir kuyruğa taşınması
- Paylaşılan kuyruğun paylaşılmayan bir kuyruğa taşınması

Bir kuyruğun bir bağlaşım olanağı yapısından başka bir yapıya taşınması

Kuyrukları ve bunların iletilerini bir CF yapısından başka bir yapıya taşımak için, MQSC MOVE QLOCAL komutunu kullanın. Yeni bir CF yapısına taşımak istediğiniz kuyruğu ya da kuyrukları tanımladığınızda, her kuyruğu taşımak için aşağıdaki yordamı kullanın:

1. Taşımak istediğiniz kuyruğun herhangi bir uygulama tarafından kullanılmadığından emin olun; yani, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinde, IPPROCS ve OPPOCS kuyruk özniteliklerinin sıfır olması gerekir.
2. Prevent applications from putting messages on the queue being moved by altering the queue definition to disable MQPUT s. Kuyruk tanımlamasını PUT (DEVRE Dışı) olarak değiştirin.
3. Aşağıdaki komut kullanılarak 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 çalışmadan varsa, tanımlama yapmadan önce bu kuyruğu 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. Taşınmakta olan, CFSTRUCT özniteliğini değiştirerek, aşağıdaki komutu kullanarak kuyruğu yeniden tanımlayın:

```
DEFINE QL(Queue_To_Move) LIKE(Temp_Queue) CFSTRUCT(NEW) QSGDISP(SHARED)
```

Kuyruk yeniden tanımlandığında, “3” sayfa 481adımında yaratılan geçici kuyruğa dayalı olur.

7. Aşağıdaki komutu kullanarak iletileri yeni kuyruğa geri taşıyın:

```
MOVE QLOCAL(Temp) TOQLOCAL(Queue_To_Move)
```

8. “3” sayfa 481 adımında yaratılan kuyruk artık gerekli değil. Silmek için aşağıdaki komutu kullanın:

```
DELETE QL(Temp_Queue)
```

9. If the queue being moved was defined in the CSQINP2 data sets, change the CFSTRUCT attribute of the appropriate DEFINE QLOCAL command in the CSQINP2 data sets. Var olan kuyruk tanımlamasının yerine yenisi konması için REPLACE anahtar sözcüğünü ekleyin.

Şekil 38 sayfa 482 , bir kuyruğun bir CF yapısından diğerine taşınması için örnek bir iş gösterir.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
// DD DSN=thlqual.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(Queue_To_Move) 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 38. Bir kuyruğun CF yapısından başka bir yapıya taşınması için örnek iş

Paylaşılmayan kuyruğun paylaşılan bir kuyruğa taşınması

Paylaşılmayan bir kuyruğu paylaşılan bir kuyruğa taşıma yordamı, bir kuyruğun bir CF yapısından başka bir kuyruğa taşınma yordamı gibidir (bkz. “Bir kuyruğun bir bağlaşım olanağı yapısından başka bir yapıya taşınması” sayfa 481). Şekil 39 sayfa 483 , bunu yapmak için bir örnek iş verir.

Not: Paylaşılan kuyruklardaki iletilerin, ileti büyüklüğü üst sınırı, ileti kalıcılığı ve kuyruk izin tipi ile ilgili belirli kısıtlamalara tabi olduğunu unutmayın; böylece, 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 39. Paylaşılmayan kuyruğun paylaşılan bir kuyruğa taşınması için örnek iş

Paylaşılan kuyruğun paylaşılmayan bir kuyruğa taşınması

Paylaşılan bir kuyruğu paylaşılmayan bir kuyruğa taşıma yordamı, bir kuyruğu bir CF yapısından başka bir kuyruğa taşıma yordamı gibi bir yordama benzer (bkz. [“Bir kuyruğun bir bağlaşım olanağı yapısından başka bir yapıya taşınması” sayfa 481](#)).

[Şekil 40 sayfa 483](#), bunu yapmak için bir örnek 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 40. Paylaşılan kuyruğun paylaşılmayan bir kuyruğa taşınması için örnek iş

Paylaşılmayan kuyrukların paylaşılan kuyruklara geçirilmesi

Paylaşılmayan kuyrukları paylaşılan kuyruklara geçirmenin iki aşaması vardır:

- Kuyruk paylaşım grubundaki ilk (ya da yalnızca) kuyruk yöneticisinin yeni düzeye geçirilmesi
- Kuyruk paylaşım grubundaki diğer kuyruk yöneticilerinin yeni düzeye geçirilmesi

Kuyruk paylaşım grubundaki ilk (ya da yalnızca) kuyruk yöneticisinin yeni düzeye geçirilmesi

[Şekil 39 sayfa 483](#), paylaşılmayan bir kuyruğu paylaşılan bir kuyruğa taşımaya ilişkin bir örnek iş gösterir. Yeni düzeye geçirilmesi gereken her kuyruk için bunu yapın.

Not:

1. Paylaşılan kuyruklardaki iletiler, ileti büyüklüğü üst sınırı, ileti kalıcılığı ve kuyruk dizini tipi ile ilgili bazı kısıtlamalara tabi olur; bu nedenle, paylaşılmayan bazı kuyrukları paylaşılan bir kuyruğa taşıyamayabilirsiniz.
2. Paylaşılan kuyruklar için doğru dizini tipini kullanmanız gerekir. Bir iletim kuyruğunu paylaşılan bir kuyruk olması için yeni düzeye geçiriyorsanız, dizini tipi MSGID olmalıdır.

Kuyruk boşsa ya da üzerindeki iletileri alkoymak zorunda kalmayacaksa, kuyruğu yeni düzeye geçirmek daha kolaydır. [Şekil 41 sayfa 484](#) , bu koşullarda kullanılmak üzere bir örnek iş gösterir.

```
//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 41. Paylaşılmayan bir kuyruğu, paylaşılan bir kuyruğa iletmeksizin taşımaya ilişkin örnek iş

Kuyruk paylaşım grubundaki diğer kuyruk yöneticilerinin yeni düzeye geçirilmesi

1. Var olan bir paylaşılan kuyrukta aynı adı taşımayan her kuyruk için, kuyruğu [Şekil 39 sayfa 483](#) ya da [Şekil 41 sayfa 484](#) içinde açıklandığı gibi taşıyın.
2. For queues that have the same name as an existing shared queue, move the messages to the shared queue using the commands shown in [Şekil 42 sayfa 484](#).

```
MOVE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR) TOQLOCAL(Queue_TO_MOVE)
DELETE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR)
```

Şekil 42. İletileri paylaşılmayan bir kuyruktan var olan bir paylaşılan kuyruktan taşıma

Db2 bağlantısının askıya alınması

If you want to apply maintenance or service to the Db2 tables or package related to shared queues without stopping your queue manager, you must temporarily disconnect queue managers in the data sharing group (DSG) from Db2.

Bunu yapmak için:

1. MQSC komutunu `QMGR 'YI AS TESIS (Db2)` kullanın.
2. Bağlamaları yapın.
3. MQSC komutu `QMGR 'YI SÜ TESIS (Db2)` olanağını kullanarak Db2 ' e yeniden bağlanın.

Bu komutların kullanılmasında kısıtlamalar olduğunu unutmayın.



Uyarı: Db2 bağlantısı askıya alındığında, aşağıdaki işlemler kullanılamayacaktır. Bu nedenle, işletmeniz en az meşgul olduğu bir zamanda bu işi yapmanız gerekir.

- Yönetim için paylaşılan kuyruk nesnelere erişim (tanımlama, silme, değiştirme)
- Paylaşılan kanallar başlatılıyor
- İletileri Db2 içinde saklama

- CFSTRUCT yedekleme ya da kurtarma

z/OS Grup nesnelere ilişkin yönetilmesi

Grup nesnelere ilişkin nasıl çalışacağını anlamak için bu konuyu kullanın.

IBM MQ , bir grup nesnesinin tanımını, onu kullanan her bir kuyruk yöneticisinin sıfır sıfır değerine otomatik olarak kopyalar. Tanımın kopyasını geçici olarak değiştirebilirsiniz ve IBM MQ , havuz kopyasından sayfa kümesi kopyalarını yenilemişmenizi sağlar. IBM MQ her zaman, başlatma sırasında havuz kopyasından sayfa kümesi kopyalarını yenilemeyi dener (kanal nesnelere ilişkin bu işlem, kanal başlatıcısı yeniden başlatıldığında yapılır). Bu, sayfa kümesi kopyalarının, kuyruk yöneticisi etkin değilken yapılan değişiklikler de dahil olmak üzere havuzdaki sürümü yansımalarını sağlar.

Yenileme işleminin gerçekleştirilmediği durumlar vardır; örneğin:

- Kuyruğun bir kopyası açıksa, kuyruğun kullanımını değiştirecek bir yenileme işlemi başarısız olur.
- Kuyruğun bir kopyası üzerinde ileti varsa, o kuyruğun silinmesine neden olacak bir yenileme işlemi başarısız olur.

Bu durumda, yenileme o kopyada gerçekleştirilmez; ancak, diğer tüm kuyruk yöneticilerindeki kopyalarda gerçekleştirilir. Bir grup nesnesini ekledikten, değiştirdikten ya da sildikten sonra ve kuyruk yöneticisinde ya da kanal başlatıcısı yeniden başlatıldıktan sonra nesne kopyalama nesnelere ilişkin ilgili sorunları denetleyin ve düzeltin.

z/OS Bağlaşım tesisinin yönetilmesi

Bağlantı olanağı (CF) yapılarını nasıl ekleyeceğini ya da kaldırmanın nasıl yapılacağını anlamak için bu konuyu kullanın.

Bu bölümde aşağıdaki görevler anlatılır:

- “Bağlaşım olanağı yapısı eklenmesi” sayfa 485
- “Bağlaşım olanağı yapısının çıkarılması” sayfa 485

Bağlaşım olanağı yapısı eklenmesi

Bağlaşım olanağı yapısı eklemek için aşağıdaki yordamı kullanın:

1. CFRM ilke verileri kümesinde CF yapısını tanımlayın. Bağlaşım olanağını bağlaşım olanağını ayarla alanında ayarlamaya ilişkin bilgiler, bağlaşım tesisi yapılarını adlandırma kurallarını ve CFRM ilke verileri kümesindeki yapıların nasıl tanımlanmasını açıklar.
2. Bu yapıyı, ileti verilerini SMDS ' ye yüklemek için yapılandırmak istiyorsanız, veri kümelerini ayırın ve önbiçimden ayırın. Ayrıntılar için bkz. paylaşılan ileti verileri kümesi yaratılması .
3. Define the structure to IBM MQ using the CFSTRUCT DEFINE command.

Bağlaşım olanağı yapısının çıkarılması

Bağlaşım olanağı yapısını kaldırmak için aşağıdaki yordamı kullanın:

1. Silmek istediğiniz bağlaşım olanağı yapısını kullanarak tüm kuyrukların bir listesini almak için aşağıdaki komutu kullanın:

```
DISPLAY QUEUE(*) QSGDISP(SHARED) CFSTRUCT(structure-name)
```

2. Yapıyı kullanan tüm kuyrukları silin.
3. Delete the CF structure from IBM MQ using the CFSTRUCT SIL command.

4. Yapı, ileti verilerini SMDS 'ye boşaltmak üzere yapılandırıldıysa, SMDS' yi silin.
5. CFRM ilke veri kümesinden yapı tanımlamasını kaldırın ve IXCMIAPU yardımcı programını çalıştırın. (This is the reverse of the customization task set up the coupling facility, described in [Bağlaşım olanağını ayarla.](#))

z/OS **Bağlaşım olanağı listesi izlenmesini ayarlama**

Bağlaşım olanağı listesi izlemeyi anlamak için bu konuyu kullanın.

Coupling facility (CF) list monitoring is used to monitor the state of list structures containing IBM MQ shared queues. Bir ileti paylaşılan bir kuyruğa eklendiğinde ve kuyruğun derinlik geçişleri sıfırdan sıfır değerine geçtiğinde, CF, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine bunu bildirir. Bildirim gönderilirken, kuyruk yöneticileri tetikleme (FIRST) kullanan tetikleyici izleyicileri ya da alma işlemi gerçekleştiren uygulamaları bilgilendirmek de içinde olmak üzere bir dizi işlem gerçekleştirebilirler.

Varsayılan olarak, CF tüm kuyruk yöneticilerine aynı anda kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine bildirir. Bazı yapılanışlarda aşağıdakiler gibi sorunlar ortaya çıkartabilir:

- Çok sayıda iletinin kuyruk paylaşım grubundaki belirli bir kuyruk yöneticisine gitmesinin, genellikle en hızlı LPAR üzerinde çalışan kuyruk yöneticisi ya da CF ' ye en yakın olan kuyruk yöneticisi olduğu iş yükü dağılımı.
- Çok sayıda başarısız oldu ve CPU süresinin boşa harcanması sonucu ortaya çıktı.

z/OS V2R3 , **KEYRNOTIFYDELAY**adlı yeni bir bağlaşım olanağı kaynak yöneticisi (cFRM) özneliği sunar; bu özellik, paylaşılan kuyruklar içeren (uygulama yapıları, uygulama yapıları, yönetici yapısı değil) liste yapılarıyla kullanılabilir ve belirli iş yükleri için iş yükü aksaması ve boş MQGET çağrılarının etkilerini en aza indirebilir ya da boş MQGET çağrılarını azaltır.

KEYRNOTIFYDELAY yalnızca, CFLEVEL 22 ya da sonraki bir sürümü çalıştıran bir CF ' deki yapılar üzerinde ayarlanabilir.

Değer, 0 ile 1.000.000 mikrosaniye aralığında, en çok yedi ondalık basamak olmalıdır. Sıfır dışında bir değere ve bir kuyruk geçişlerinin derinliğinden sıfır dışında bir değere ayarlanırsa, CF kuyruk paylaşım grubundan tek bir kuyruk yöneticisi seçer ve bu, gruptaki tüm diğer kuyruk yöneticilerinden önce bu kuyruk yöneticisini bildirir.

Kuyruk yöneticisi çevrimsel bir şekilde seçildi. If the selected queue manager does not process the message inside the time interval described by **KEYRNOTIFYDELAY** all the other queue managers in the queue sharing group will also be notified.

KEYRNOTIFYDELAY ' a ilişkin ek bilgi için şu adresi kullanabilirsiniz: [Understanding Keyrange Monitoring Notification Delay.](#)

LISTNOTIFYDELAY ve **SUBNOTIFYDELAY**adında iki benzer CFRM özneliği olduğunu unutmayın. Bunların hiçbirisi IBM MQ iş yükü üzerinde ölçülebilir bir etkiye sahip değil.

z/OS **Recovery and restart on z/OS**

IBM MQtarafından kullanılan kurtarma ve yeniden başlatma mekanizmalarını anlamak için bu konuyu kullanın.

z/OS **yeniden başlatmaIBM MQ**

Kuyruk yöneticisi sona erdirildikten sonra, kuyruk yöneticisinin nasıl sonlandırılabiline bağlı olarak, farklı yeniden başlatma yordamları gerekir. Kullanabileceğiniz farklı yeniden başlatma yordamlarını anlamak için bu konuyu kullanın.

Bu konuda, aşağıdaki durumlarda kuyruk yöneticinizin yeniden başlatılmasına ilişkin bilgiler yer alır:

- [“Olağan kapanmadan sonra yeniden başlatma” sayfa 487](#)
- [“Olağandışı sona erdirme işleminden sonra yeniden başlatma” sayfa 487](#)
- [“Sayfa kümelerinizi kaybetmediyseniz yeniden başlatma” sayfa 487](#)

- [“Günlük veri kümelerinizi kaybetmediyseniz yeniden başlatma” sayfa 487](#)
- [CF yapılarınızı kaybetmediyseniz yeniden başlatma](#)

Olağan kapanmadan sonra yeniden başlatma

Kuyruk yöneticisi STOP QMGR komutuyla durdurulursa, sistem çalışmasını düzenli bir şekilde bitirir ve durdurmadan önce bir sonlandırma denetim noktası alır. Kuyruk yöneticisini yeniden başlattığınızda, sistem durumunu sona erdirmeye sırasında belirlemek için sistem denetim noktasından ve kurtarma gününden gelen bilgileri kullanır.

Kuyruk yöneticisini yeniden başlatmak için, START QMGR komutunu [“z/OSüzerinde kuyruk yöneticisini başlatma ve durdurma” sayfa 417](#) içinde açıkladığı biçimde çalıştırın.

Olağandışı sona erdirmeye işleminden sonra yeniden başlatma

IBM MQ , yeniden başlatmanın olağan bir kapanma mı, yoksa olağandışı sonlandırma mı olduğunu otomatik olarak saptar

Kuyruk yöneticisi olağan dışı sona erdirildikten sonra, STOP QMGR komutu verildikten sonra olağan dışı bir şekilde başlatıldıktan sonra başlatılır. Kuyruk yöneticisi olağan dışı bir şekilde sonlanırsa, işini bitirmeden ya da sonlandırma denetim noktasını ele geçirmeden sonlandırılır.

Kuyruk yöneticisini yeniden başlatmak için, START QMGR komutunu [“z/OSüzerinde kuyruk yöneticisini başlatma ve durdurma” sayfa 417](#) içinde açıkladığı biçimde çalıştırın. Bir kuyruk yöneticisini olağandışı sona erdirmeye işleminden sonra yeniden başlattığınızda, bu durum, günlüğün bilgileri kullanılarak sonlandırma sırasında durumunu bilgilerini yeniler ve çeşitli görevlerin durumunu size bildirir.

Olağan durumda, yeniden başlatma işlemi tüm tutarsız durumları çözer. Ancak, bazı durumlarda tutarsızlıkları çözmek için belirli adımları atmanız gerekir. Bu, [“İş birimlerinin el ile kurtarılıyor” sayfa 500](#) içinde açıklanmaktadır.

Sayfa kümelerinizi kaybetmediyseniz yeniden başlatma

Sayfa kümelerinizi kaybetmediyseniz, kuyruk yöneticisini yeniden başlatabilmeniz için önce bunları yedek kopyalarınızdan geri yüklemeniz gerekir. Bu, [“Sayfa kümelerini nasıl yedeklemeniz ve kurtarmanız” sayfa 471](#) içinde açıklanmaktadır.

Ortam kurtarma işlemi için gereken süre uzunluğundan dolayı, kuyruk yöneticisi bu koşullar altında yeniden başlatma işlemi uzun sürebilir.

Günlük veri kümelerinizi kaybetmediyseniz yeniden başlatma

Bir kuyruk yöneticisini durdurduktan sonra (STOP QMGR komutunu kullanarak), günlüğün her iki kopyası da kaybolursa ya da zarar görmüşse, kuyruk yöneticisini, tutarlı sayfa kümeleri (Method 1: Full backup (Yöntem 1: Tam yedekleme) kullanarak üretilir) sağlayan kuyruk yöneticisini yeniden başlatabilirsiniz.

Bu yordamı izleyin:

1. Kuyruk yöneticinizde var olan her bir sayfa kümesine karşılık gelecek yeni sayfa kümelerini tanımlayın. Sayfa kümesi tanımlamasıyla ilgili bilgi için [Görev 15: Sayfa kümelerinizi tanımlama](#) başlıklı konuya bakın.

Her yeni sayfa kümesinin, karşılık gelen kaynak sayfa kümesinden daha büyük olduğundan emin olun.

2. Hedef sayfa kümesini biçimlendirmek için CSQUTIL FORMAT işlevini kullanın. Daha ayrıntılı bilgi için [Sayfa kümelerini biçimlendirme](#) başlıklı konuya bakın.

3. Var olan sayfa kümelerini kopyalamak ya da yerine getirmek için CSQUTIL 'in RESETPAGE işlevini kullanın ve her bir sayfada RBA günlük RBA' yı sıfırlayın. Bu işleyle ilgili ek bilgi için [Sayfa kümesinin kopyalanması ve günlük durumuna getirilmesi](#) başlıklı konuya bakın.
4. Kuyruk yöneticisi günlük veri kümelerine ve BSDS ' yi CSQJU003 kullanarak yeniden tanımlayın (bkz. [Değişiklik günlüğü döküm yardımcı programı](#)).
5. Yeni sayfa kümelerini kullanarak kuyruk yöneticisini yeniden başlatın. Bunu yapmak için, aşağıdakilerden birini yapın:
 - Kuyruk yöneticisini değiştirme işlemi, yeni sayfa kümelerine gönderme yapmak için görev yordamına başladı. Ek bilgi için [Görev 6: IBM MQ kuyruk yöneticisi yordamlarıyla ilgili yordam yaratılması](#) başlıklı konuya bakın.
 - Eski sayfa kümelerini silmek ve yeni sayfa kümelerini yeniden adlandırmak için, eski sayfa kümeleriyle aynı adları vermek için Erişim Yöntemi Hizmetleri olanağını kullanın.

Dikkat: Herhangi bir IBM MQ sayfa kümesini silmeden önce, gerekli yedek kopyalarını yaptığınızdan emin olun.

Kuyruk yöneticisi, kuyruk paylaşım grubunun bir üyesiye, GROUP ve SHARED nesne tanımlamaları olağan durumda kaybedilen ya da zarar görmüş günlüklerden etkilenmez. Ancak, paylaşılan kuyruk iletileri, kaybedilen ya da zarar görmüş günlüklerin kapsadığı bir iş biriminde yer aldıysa, bu tür kesinleştirilmemiş iletiler üzerindeki etkisi tahmin edilemez.

Not: Günlükler zarar görmüşse ve kuyruk yöneticisi bir kuyruk paylaşım grubunun üyesiye, paylaşılan kalıcı iletileri kurtarma yeteneği kaybolabilir. RECOVER (YES) özniteliğe sahip tüm CF yapılarına ilişkin kuyruk paylaşım grubundaki başka bir etkin kuyruk yöneticisinde bir BACKUP CFSTRUCT komutunu hemen verin.

CF yapılarınızı kaybetmediyseniz yeniden başlatma

Kuyruk yöneticisi sonlandırmadığı için CF yapılarınızı kaybedersen yeniden başlatmanız gerekmez.

z/OSüzerinde alternatif site kurtarma

Tek bir kuyruk yöneticisini ya da kuyruk paylaşım grubunu kurtarabilir ya da disk ikizlemeyi düşünebilirsiniz.

Daha fazla ayrıntı için aşağıdaki bölümlere bakın:

- [Diğer bir sitede tek bir kuyruk yöneticisinin kurtarılması](#)
- [Bir kuyruk paylaşım grubunun kurtarılması](#).
 - [CF yapısı ortamı kurtarma](#)
 - [Başyerde kuyruk paylaşım grubu yedekleniyor](#)
 - [Diğer yerde bir kuyruk paylaşım grubunun kurtarılması](#)
- [Disk ikizlemenin kullanılması](#)

Diğer bir sitede tek bir kuyruk yöneticisinin kurtarılıyor

Bir IBM MQ bilgi işlem merkezinin toplam kaybı oluşursa, kurtarma yerinde başka bir kuyruk yöneticisi ya da kuyruk paylaşım grubu üzerinde kurtarma işlemi yapabilirsiniz. (Bir kuyruk paylaşım grubuna ilişkin diğer site kurtarma yordamı için bkz. [“Diğer yerde bir kuyruk paylaşım grubunun kurtarılması” sayfa 492](#) .)

Kurtarma yerinde başka bir kuyruk yöneticisini kurtarmak için, sayfa kümelerini ve günlükleri düzenli olarak yedeklemeniz gerekir. Tüm veri kurtarma işlemlerinde olduğu gibi, olağanüstü durumdan kurtarma işleminin amaçları, mümkün olduğunca az veri, iş yükü işleme (güncelleme) ve zaman olarak kaybetmeye devam eder.

Kurtarma yerinde:

- The recovery queue managers **gerekir** have the same names as the lost queue managers.

- Her kurtarma kuyruğu yöneticinde kullanılan sistem parametre modülü (örneğin, CSQZPARM), ilgili kayıp kuyruk yöneticisiyle aynı parametreleri içermelidir.

Bunu gerçekleştirdiğinizde, aşağıdaki yordamda açıklandığı gibi tüm kuyruk yöneticilerinizi yeniden kurun. Bu, tek bir kuyruk yöneticisi için kurtarma yerinde olağanüstü durum kurtarma işlemi gerçekleştirmek için kullanılabilir. Bu ürün, mevcut olan her şeyin geçerli olduğunu varsayar:

- Arşiv günlüklerinin ve BSDS ' lerin birincil yerde olağan çalışması tarafından yaratılan kopyaları (etkin günlükler, birincil yerdeki kuyruk yöneticisiyle birlikte kaybedilir).
- Sayfa kümelerinin, kullanılabilir en son arşiv günlük kopyalarından daha eski ya da daha eski olan birincil yerdeki kuyruk yöneticisinden 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 kümelerini tanımlayın ve bunları birincil yerden sayfa kümelerine ilişkin kopyalardaki verilerle yükleyin.
2. Yeni etkin günlük veri kümelerini tanımlayın.
3. Yeni bir BSDS veri kümesi tanımlayın ve en son arşivlenmiş BSDS ' leri bu dizine kopyalamak için Access Method Services REPRO (Erişim Yöntemi Hizmetleri REPRO) olanağını kullanın.
4. Bu en son BSDS ' den bilgileri yazdırmak için CSQJU004 yazdırma günlüğü eşleme yardımcı programını kullanın. Bu BSDS arşivlendiği sırada, sahip olduğunuz en son arşivlenen günlük etkin bir günlük olarak kesildi ve arşivlenen bir günlük olarak görünmüyor. Bu günlüğün STARTRBA ve ENDRBA ' yı kaydedin.
5. Use the change log inventory utility, CSQJU003, to register this latest archive log data set in the BSDS that you have just restored, using the STARTRBA and ENDRBA recorded in Step “4” sayfa 489.
6. Use the DELETE option of CSQJU003 to remove all active log information from the BSDS.
7. BSDS ' ye etkin günlükler eklemek için CSQJU003 NEWLOG seçeneğini kullanın, STARTRBA ya da ENDRBA belirlemeyin.
8. BSDS 'ye yeniden başlatma denetimi kaydı eklemek için CSQJU003 ' i kullanın. CRESTART CREATE, ENDRBA=highrba değerini belirtin; burada highrba , kullanılabilir en son arşiv günlüğünün yüksek RBA (Adım “4” sayfa 489 içinde bulunur) ve artı 1 'dir.

BSDS şimdi tüm etkin günlükleri boş olarak tanımlıyor, kullanılabilir durumda olduğunuz arşivlenmiş tüm günlükler ve günlüklerinizin sonundaki kontrol noktaları yok.

9. START QMGR komutunu kullanarak kuyruk yöneticisini yeniden başlatın. Kullanıma hazırlama sırasında, aşağıdaki gibi bir işletmen yanıt iletisi gönderilir:

```
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 verileri kurtarır.

CSQJU003 ve CSQJU004 kullanımlarına ilişkin bilgi için [IBM MQ yardımcı programlarının kullanılması](#) başlıklı konuya bakın.

The following example shows sample input statements for CSQJU003 for steps 6, 7, and 8:

```
* 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 DSNAMES=MQM2 . LOGCOPY1 . DS03 , COPY1
NEWLOG DSNAMES=MQM2 . LOGCOPY1 . DS04 , COPY1
NEWLOG DSNAMES=MQM2 . LOGCOPY2 . DS01 , COPY2
NEWLOG DSNAMES=MQM2 . LOGCOPY2 . DS02 , COPY2
NEWLOG DSNAMES=MQM2 . LOGCOPY2 . DS03 , COPY2
NEWLOG DSNAMES=MQM2 . LOGCOPY2 . DS04 , COPY2
```

```
* Step 8
CRESTART CREATE , ENDRBA=063000
```

Kurtarma yerinde kanal başlatıcıyı yeniden başlatmak için göz önünde bulundurmanız gereken şeyler, kanal başlatıcısını farklı bir z/OS görüntüsünde yeniden başlatmak için ARM kullanırken karşılaştılar gibi olur. Ek bilgi için [“IBM MQ ağında ARM ' nin kullanılması”](#) sayfa 498 başlıklı konuya bakın. Your recovery strategy should also cover recovery of the IBM MQ product libraries and the application programming environments that use IBM MQ (CICS , for example).

Değişiklik günlüğü döküm yardımcı programının diğer işlevleri (CSQJU003) olağanüstü durumdan kurtarma senaryolarında da kullanılabilir. HIGHRBA işlevi, önyükleme veri kümesi içinde en yüksek RBA yazılan ve en yüksek RBA boşaltma değerlerine sahip değerlerin güncellenmesine olanak sağlar. CHECKPT işlevi, yeni denetim noktası kuyruğu kayıtlarının eklenmesine ya da BSDS ' de varolan denetim noktası kuyruğu kayıtlarının silinmesine olanak sağlar.

Dikkat: Bu işlevler IBM MQ verilerinizin bütünlüğünü etkileyebilir. Only use them in disaster recovery scenarios under the guidance of IBM service personnel.

Hızlı kopyalama teknikleri

Kuyruk yöneticisi dondurulmuşken, tüm sayfa kümeleri ve günlüklerin kopyaları yapılırsa, kopyalar başka bir yerde kuyruk yöneticisini yeniden başlatmak için kullanılabilir tutarlı bir küme olur. Genellikle, kuyruk yöneticisinin çok daha hızlı bir şekilde yeniden başlatılmasını sağlar; bu işlem, küçük ortam kurtarma işleminin gerçekleştirilmesine neden olur.

Kuyruk yöneticisini dondurmak için SUSPEND QMGR LOG komutunu kullanın. Bu komut, arabellek havuzlarını sayfa kümelerine temizler, bir denetim noktasını alır ve daha fazla günlük yazma etkinliğini durdurur. Günlük yazma etkinliği askıya alındığında, siz RESUME QMGR LOG komutu verinceye kadar kuyruk yöneticisi etkin bir şekilde dondurulmuştur. Kuyruk yöneticisi dondurulsa da, sayfa kümeleri ve günlükler kopyalanabilir.

Sayfa kümelerini ve günlükleri hızla kopyalamak için FLASHCOPY ya da SNAPSHOT gibi araçları kopyalayarak, kuyruk yöneticisinin dondurulmuş olduğu süre en az bir alt sınır değerine düşürülebilmektedir.

Ancak, bir kuyruk paylaşım grubu içinde, SUSPEND QMGR LOG komutu bu kadar iyi bir çözüm olmayabilir. Etkili olmak için, günlüklerin kopyalarının tümü kurtarma için aynı noktayı içermelidir; bu da, aynı anda kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinde SUSPEND QMGR LOG komutunun verilmesi ve bu nedenle tüm kuyruk paylaşım grubunun bir süre dondurulmasını sağlamak zorundadır.

Kuyruk paylaşım grubunun kurtarılması

Bir ana site faciası durumunda, uzak bir yerdeki bir kuyruk paylaşım grubunu, ana siteden yedek veri kümelerini kullanarak yeniden başlatabilirsiniz. To recover a queue sharing group you need to coordinate the recovery across all the queue managers in the queue sharing group, and coordinate with other resources, primarily Db2. Bu bölüm, bu görevleri ayrıntılı olarak açıklar.

- [CF yapısı ortamı kurtarma](#)
- [Başyerede kuyruk paylaşım grubu yedekleniyor](#)
- [Diğer yerde bir kuyruk paylaşım grubunun kurtarılması](#)

CF yapısının ortam kurtarması

Paylaşılan kuyruklardaki kalıcı iletileri tutmak için kullanılan bir CF yapısının ortam kurtarma işlemi, günlüğe kaydedilen güncellemelerin uygulanması tarafından kurtarılabilecek ortamın yedeğine sahip olmaya dayanır. MQSC BACKUP CFSTRUCT komutunu kullanarak, CF yapılarınızın yedeklerini düzenli olarak ele alın. Paylaşılan kuyruklara ilişkin tüm güncellemeler (MQGET'ler ve MQPUT'ler), güncelleme gerçekleştirildiği kuyruk yöneticisinin günlüğüne yazılır. Bir CF yapısının ortam kurtarmasını gerçekleştirmek için, günlüğe kaydedilen güncellemeleri, o CF yapısını kullanan tüm kuyruk yöneticilerinin günlüklerinden bu yedeğe uygulamanız gerekir. MQSC RECOVER CFSTRUCT komutunu kullandığınızda, IBM MQ , günlükleri ilgili kuyruk yöneticilerinden otomatik olarak birleştirir ve güncellemeleri en son yedeklemenize uygular.

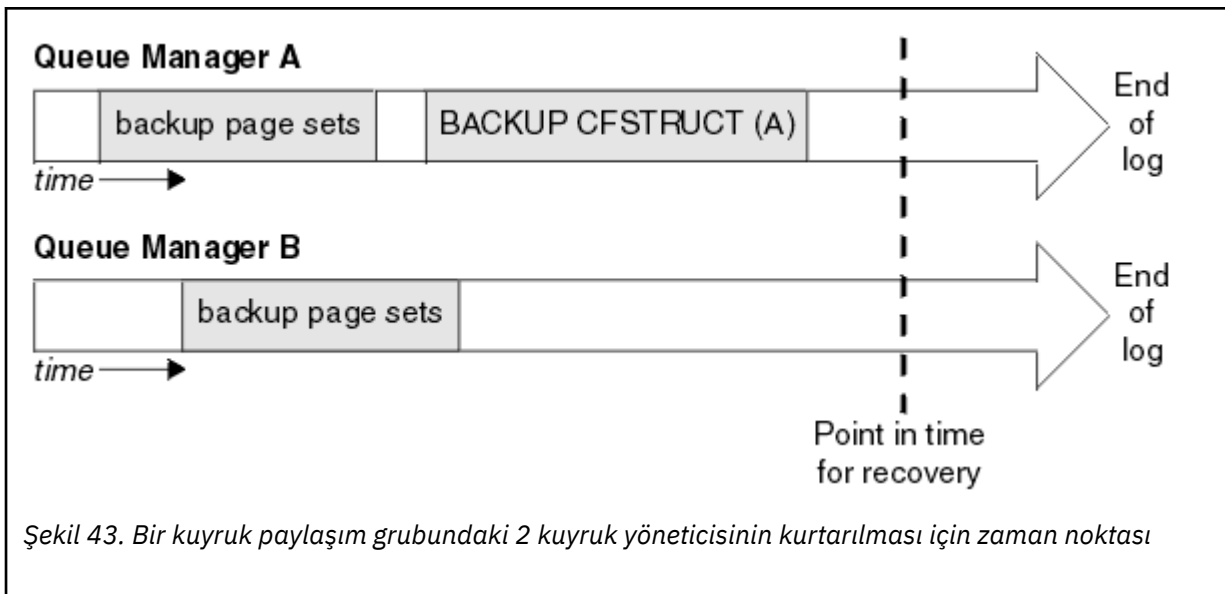
CF yapısı yedeklemesi, BACKUP CFSTRUCT komutunu işleyen kuyruk yöneticisinin günlüğüne yazılır; bu nedenle, toplanacak ve diğer siteye taşınmak üzere başka veri kümesi yok.

Birincil yerde kuyruk paylaşım grubu yedekleniyor

Ana sitede, düzenli olarak tutarlı bir yedekleme kümesi oluşturmanız gerekir. Bu, alternatif bir sitede kuyruk paylaşım grubunu yeniden oluşturmak için bir olağanüstü durum durumunda kullanılabilir. Tek bir kuyruk yöneticisi için, kurtarma işlemi zaman içinde isteğe bağlı bir zaman noktasına, genellikle uzak yerde bulunan günlüklerin sonuna kadar olabilir. Ancak, kalıcı iletilerin paylaşılan bir kuyrukta saklandığı durumlarda, kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisi, kuyrukta güncellemeleri (MQPUT ya da MQGET 'ler) gerçekleştirmiş olabileceğinden, paylaşılan kuyrukları kurtarmak için, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinin günlükleri birleştirilmelidir.

Bir kuyruk paylaşım grubunun kurtarılması için, tüm kuyruk yöneticilerine ilişkin günlük verilerinin günlük aralığı içinde olan bir zaman noktası oluşturmanız gerekir. Ancak, ortamı yalnızca günlükte **ileri** kurtardığınızda, bu zaman noktası, BACKUP CFSTRUCT komutunun yayınlanmasından sonra ve herhangi bir sayfa kümesi yedeklemesi gerçekleştirildikten sonra olmalıdır. (Genellikle, kurtarma işlemi için zaman noktası bir iş günü ya da haftanın sonuna karşılık gelebilir.)

Aşağıdaki çizge, bir kuyruk paylaşım grubundaki iki kuyruk yöneticisine ilişkin zaman çizgilerini göstermektedir. Her kuyruk yöneticisi için, sayfa kümelerinin yaklaşık yedekleri alınır (bkz. [Method 2: Fuzzy backup](#)). Kuyruk yöneticisi A ' da bir BACKUP CFSTRUCT komutu verilir. Daha sonra, etkin günlüğü kesmek için her kuyruk yöneticisinden bir ARCHIVE LOG komutu verilir ve bu komutu, diğer yere taşınabilen kuyruk yöneticisinden çevrimdışı ortama kopyalayabilirsiniz. Günlüğün sonu, ARCHIVE LOG komutunun yayınlandığı zamanı tanımlar ve bu nedenle, genellikle diğer yerde bulunan günlük verilerinin kaplamı belirtilir. Kurtarma için zaman noktası, herhangi bir sayfa kümesinin ya da CF yapısı yedeklerinin sonu ile alternatif sitede bulunan en eski günlük sonunun arasında olmalıdır.



IBM MQ records information associated with the CF structure backups in a table in Db2. Depending on your requirements, you might want to coordinate the point in time for recovery of IBM MQ with that for

Db2, or it might be sufficient to take a copy of the IBM MQ CSQ.ADMIN_B_STRBACKUP table after the BACKUP CFSTRUCT commands have finished.

Bir kurtarma işlemi için hazırlık yapmak için:

1. Kuyruk paylaşım grubundaki her kuyruk yöneticisi için sayfa kümesi yedeklemeleri yaratın.
2. RECOVER (YES) özneliğe sahip her CF yapısı için bir BACKUP CFSTRUCT komutu yayınlayın. Bu komutları tek bir kuyruk yöneticisinden ya da iş yükünü dengelemek için kuyruk paylaşım grubu içindeki farklı kuyruk yöneticilerinden yayınlatabilirsiniz.
3. Tüm yedekleme işlemleri 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 yedeklemelerini, arşivlenen günlükleri, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine ilişkin arşivlenmiş BSDS 'leri ve seçtiğiniz Db2 yedekleme bilgilerinizi, off-site üzerinden iletin.

Diğer yerde bir kuyruk paylaşım grubunun kurtarılması

Kuyruk paylaşım grubunu kurtarabilmek için, ortamı hazırlamanız gerekir:

1. Kuyruk paylaşım grubunu kurduğunuzda, bağlaşım tesisinizde çalışan yeni kurulan tesislerde eski bilginiz varsa, önce bunu temizlemeniz gerekir:

Not: Bağlaşım tesisinde eski bilginiz yoksa, 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 adı ile başlayan tüm yapılar için, bu yapılardan bağlantıyı zorlamak için z/OS komut SETXCF FORCE CONNECTION ' ' ı kullanın:

```
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 paylaşma gruplarını geri yükleyin.
3. CSQ.ADMIN_B_STRBACKUP çizelgesi, en son ana sitede alınan en son yapı yedekleriyle ilgili bilgileri içerir.

Not: STRBACKUP çizelgesinin, en son yapı yedekleme bilgilerini içermesi önemlidir. Daha eski yapı yedekleme bilgileri, en son bir DISPLAY USAGE TYPE (DATASET) komutu tarafından verilen bilgilerin sonucu olarak attığınız veri kümelerini gerektirebilir. Bu, kurtarılan CF yapılarınızın doğru bilgileri içermeyeceğini gösterir.

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 olağan olur:

```
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 kümelerini tanımlayın ve bunları birincil yerden sayfa kümelerine ilişkin kopyalardaki verilerle yükleyin.
2. Yeni etkin günlük veri kümelerini tanımlayın.
3. Yeni bir BSDS veri kümesi tanımlayın ve en son arşivlenmiş BSDS 'leri bu dizine kopyalamak için Access Method Services REPRO (Erişim Yöntemi Hizmetleri REPRO) olanağını kullanın.
4. Bu en son BSDS 'den bilgileri yazdırmak için CSQJU004 yazdırma günlüğü eşleme yardımcı programını kullanın. Bu BSDS arşivlendiği sırada, sahip olduğunuz en son arşivlenen günlük etkin bir günlük olarak kesildi ve arşivlenen bir günlük olarak görünmüyor. Bu günlüğün STARTRBA, STARTLRSN, ENDRBA ve ENDLRSN değerlerini kaydedin.
5. Use the change log inventory utility, CSQJU003, to register this latest archive log data set in the BSDS that you have just restored, using the values recorded in Step “4” sayfa 493.
6. Use the DELETE option of CSQJU003 to remove all active log information from the BSDS.
7. BSDS 'ye etkin günlükler eklemek için CSQJU003 NEWLOG seçeneğini kullanın, STARTRBA ya da ENDRBA belirlemeyin.
8. Kuyruk paylaşım grubu için *recoverylrsn* değerini hesaplayın. *recoverylrsn* , kuyruk paylaşım grubundaki tüm kuyruk yöneticilerindeki (“4” sayfa 493 . adımda kaydedildiği şekilde), eksi 1 'den en düşük ENDLRSNS 'lerin en alt düzey yöneticidir. Örneğin, kuyruk paylaşım grubunda iki kuyruk yöneticisi varsa ve bunlardan biri için ENDLRSN B713 3C72 22C5'dir ve diğeri B713 3D45 2123 ise, *recoverylrsn* B713 3C72 22C4' dir.
9. BSDS 'ye yeniden başlatma denetimi kaydı eklemek için CSQJU003 ' i kullanın. Şunu belirtin:

```
CRESTART CREATE,ENDLRSN= recoverylrsn
```

where *recoverylrsn* is the value you recorded in Step “8” sayfa 493.

BSDS şimdi tüm etkin günlükleri boş olarak tanımlıyor, kullanılabilir durumda olduğunuz arşivlenmiş tüm günlükler ve günlüklerinizin sonundaki kontrol noktaları yok.

CRESTART kaydını, kuyruk paylaşım grubu içindeki her kuyruk yöneticisi için BSDS 'ye eklemelisiniz.

10. Kuyruk paylaşımı grubundaki her kuyruk yöneticisini START QMGR komutunu kullanarak yeniden başlatın. Kullanıma hazırlama sırasında, aşağıdaki gibi bir işletmen yanıt iletisi gönderilir:

```
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 verileri kurtarır.

IBM WebSphere MQ 7.0.1 ve daha sonraki bir yayın düzeyinde, ilk kuyruk yöneticisi, kuyruk paylaşım grubunun diğer üyeleri için denetimci yapısı bölümlerini ve kendi kendine ait bölümleri yeniden oluşturabiliyor ve artık bu aşamada kuyruk paylaşım grubundaki her kuyruk yöneticisini yeniden başlatmak zorunda kalmamaktadır.

11. Tüm kuyruk yöneticilerine ilişkin denetim yapısı verileri yeniden oluşturulduğunda, her CF uygulama yapısı için RECOVER CFSTRUCT komutunu verin.

Tek bir kuyruk yöneticisindeki tüm yapılar için RECOVER CFSTRUCT komutunu verdiyseniz, günlük birleştirme işlemi yalnızca bir kez gerçekleştirilir; bu nedenle, her bir kuyruk yöneticisinin günlük birleştirme adımını gerçekleştirmek için sahip olduğu her CF yapısı için farklı bir kuyruk yöneticisinden komut yayınlanmaktan daha hızlı olur.

Bir kuyruk paylaşım grubunda koşullu yeniden başlatma işlemi kullanıldığında, IBM WebSphere MQ 7.0.1 ve daha sonraki kuyruk yöneticilerinde, eşdüzey yönetici yeniden oluşturma işlemi gerçekleştirilirken, BSDS 'nin kendi kendilerine aynı CRESTART LRSN' yi içerip içermemelerini denetleyin. Bu, yeniden oluşturulan yönetici yapısının bütünlüğünü sağlamaktır. Bu nedenle, QSG '

deki diğer eşleri yeniden başlatmak önemlidir, böylece grubun herhangi bir üyesinin sonraki koşulsuz yeniden başlatmasından önce kendi KRESTART bilgilerini işleyebilir.

Disk ikizlemenin kullanılması

Birçok kuruluş artık, veri kümelerinin eşzamanlı kopyalarını alternatif bir sitede yapmak için IBM Metro Mirror (önceki adıyla PPRC) gibi disk ikizleme teknolojilerini kullanır. In such situations, many of the steps detailed become unnecessary as the IBM MQ page sets and logs at the alternative site are effectively identical to those at the prime site. Bu teknolojilerin kullanıldığı durumlarda, alternatif bir sitede bir kuyruk paylaşım grubunu yeniden başlatma adımları şu şekilde özetlenebilir:

- Diğer yerdeki IBM MQ CF yapılarını temizleyin. (Bu genellikle, herhangi bir önceki olağanüstü durumdan kurtarma alıştırmasından kalan bilgileri içerir).
- Restore Db2 systems and all tables in the database used by the IBM MQ queue sharing group.
- Kuyruk yöneticilerini yeniden başlatın. IBM WebSphere MQ 7.0.1 öncesinde, kuyruk paylaşımı grubunda tanımlı olan her kuyruk yöneticisini yeniden başlatmak gerekir. Her kuyruk yöneticisi, kuyruk yöneticisi yeniden başlatma işlemi sırasında denetimci yapısının kendi bölümünü yeniden kurtarır. Her bir kuyruk yöneticisi yeniden başlatıldıktan sonra, ana LPAR ' ları üzerinde olmayan bu kişiler yeniden kapatılabilir. IBM WebSphere MQ 7.0.1 ve üstü için, ilk kuyruk yöneticisi, kuyruk paylaşım grubunun diğer üyeleri için denetimci yapısı bölümlerini yeniden oluşturmada ve kuyruk paylaşım grubundaki her kuyruk yöneticisini yeniden başlatmak artık gerekmeyecek şekilde yeniden oluşturmalıdır.
- Yönetici yapısı yeniden oluşturulduktan sonra uygulama yapılarını kurtarın.

V 9.2.0 IBM MQ 9.1.2 ve daha sonra, Metro Mirror ile ikizlenmiş etkin günlüklere yazarken zHyperWrite kullanımını destekler. zHyperWrite, Metro Mirror' un kullanılmasına ilişkin performans etkisini azaltmanıza yardımcı olabilir; daha fazla bilgi için bkz. [Metro Mirror ile IBM MQ komutunu kullanma](#) .

z/OS *Kuyruk yöneticisi yeniden kullanıma hazırlanıyor*

Kuyruk yöneticisi olağandışı sona erdirildiyse, bu programı yeniden başlatamayabilirsiniz. Bunun nedeni, sayfa kümelerinizin ya da günlüklerinizin kaybolduğu, kesildiği ya da bozulduğu için olabilir. Bu gerçekleşirse, kuyruk yöneticisini yeniden kullanıma hazırlamanız gerekebilir (soğuk başlatma işlemi gerçekleştirin).

Dikkat

Kuyruk yöneticisini başka bir şekilde yeniden başlatamıyorsanız, yalnızca soğuk başlatma işlemi gerçekleştirin. Soğuk başlatma işlemi gerçekleştirmek, kuyruk yöneticinizi ve nesne tanımlarınızı kurtarmanıza olanak sağlar; **değil** , ileti verilerinizi kurtarabilecektir. Bunu yapmadan önce, bu konuda açıklanan diğer yeniden başlatma senaryolarının hiçbirinin sizin için çalışmadığından emin olun.

Yeniden başlattığınızda, tüm IBM MQ nesneniz tanımlı ve kullanılabilir durumda, ancak ileti verisi yok.

Not: Bir kuyruk yöneticisini bir kümenin parçasıyken yeniden kullanıma hazırlamayın. Önce kuyruk yöneticisini kümeden kaldırmalı (kümedeki diğer kuyruk yöneticilerindeki RESET CLUSTER komutlarını kullanarak), yeniden başlatma ve son olarak yeni kuyruk yöneticisi olarak kümeyle yeniden tanıtılmalıdır.

Bunun nedeni, yeniden başlatma işlemi sırasında kuyruk yöneticisi tanıtıcısının (QMID) değiştirildiğinden, eski kuyruk yöneticisi tanıtıcısına sahip herhangi bir küme nesnesinin kümeden kaldırılması gerekir.

Ek bilgi için aşağıdaki kısımlara bakın:

- [Kuyruk paylaşımı grubunda olmayan bir kuyruk yöneticisini yeniden başlatma](#)
- [Kuyruk yöneticilerinin kuyruk paylaşım grubunda yeniden başlatılması](#)

Kuyruk paylaşım grubunda olmayan bir kuyruk yöneticisini yeniden kullanıma hazırlama

Bir kuyruk yöneticisini yeniden başlatmak için aşağıdaki yordamı izleyin:

1. Kuyruk yöneticisini yeniden başlattığınızda kullanılacak nesne tanımlaması deyimlerini hazırlayın. Bunu yapmak için aşağıdakilerden birini yapın:
 - Sayfa kümesi sıfır, CSQUTIL SDEFS işlevini kullanın (bkz. IBM MQ tanımlama komutları listesini [oluşturma](#)). Tüm nesne tipleri (kimlik doğrulama bilgileri nesnelere, CF yapıları, kanallar, ad listeleri, işlemler, kuyruklar ve depolama sınıfları) için tanımlamalar edinmeniz gerekir.
 - Sayfa kümesi sıfır kullanılamıyorsa, nesne tanımlarınızı en son yedeklediğinizde tanımları kullanın.
2. Kuyruk yöneticisi veri kümelerinizi yeniden tanımlayın ("1" sayfa 495 adımı tamamlayıncaya kadar bunu gerçekleştirin).

Ek bilgi için [önyükleme ve günlük veri kümelerinin yaratılması ve sayfa kümelerinizin tanımlanması](#) başlıklı konuya bakın.
3. Yeni tanımlanan ve ilk kullanıma hazırlanmış günlük veri kümelerini, BSDS ' leri ve sayfa kümelerini kullanarak kuyruk yöneticisini yeniden başlatın. "1" sayfa 495 adımıyla yarattığınız nesne tanımlaması giriş deyimlerini, CSQINP2 kullanıma hazırlama giriş verileri kümesinde giriş olarak kullanın.

Kuyruk paylaşım grubundaki kuyruk yöneticileri yeniden kullanıma hazırlanıyor

Kuyruk paylaşım grubunda bir kuyruk yöneticisini yeniden başlatma işlemi daha karmaşık olur. Sayfa kümesi ya da günlük sorunları nedeniyle bir ya da daha çok kuyruk yöneticisini yeniden kullanıma hazırlamak gerekebilir; ancak, Db2 ile ya da bağlantı kurmak için bağlaşım tesisinde de sorun olabilir. Bu yüzden, çeşitli alternatifler vardır:

Soğuk başlangıç

Tüm kuyruk paylaşım grubunun yeniden başlatılması, tüm bağlaşım tesisleri yapılarının zorlanması, kuyruk paylaşım grubuna ilişkin tüm nesne tanımlamalarının Db2'den temizlenmesi, günlüklerin ve BSDS' lerin silinmesi ya da yeniden tanımlanması ve kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine ilişkin sayfa kümelerinin biçimlendirilmesini içerir.

Paylaşılan tanımlamalar alıkonda

Kuyruk paylaşım grubundaki tüm kuyruk yöneticileri için, günlükleri ve BSDS ' leri silin ya da yeniden tanımlayın ve tüm bağlaşım tesisleri yapılarını zorlayın. Yeniden başlatma sırasında, tüm iletiler silinecektir. Kuyruk yöneticileri, Db2 veritabanında hala var olan GROUP nesnelere karşılık gelen COPY nesnelere yeniden oluşturur. Paylaşılan kuyruklar var olmaya devam eder ve kullanılabilir.

Tek kuyruk yöneticisi yeniden kullanıma hazırlandı

Tek kuyruk yöneticisi için günlükleri ve BSDS ' leri ve biçim sayfa kümelerini silin ya da yeniden tanımlayın (bu işlem tüm özel nesnelere ve iletileri siler). Yeniden başlatma sırasında, kuyruk yöneticisi, Db2 veritabanında hala var olan GROUP nesnelere karşılık gelen COPY nesnelere yeniden yaratır. Paylaşılan kuyruklar var olmaya devam eder, bu iletiler üzerlerindeki iletiler için de kullanılabilir ve kullanılabilir.

Kuyruk paylaşım grubunun zaman zaman kurtarılmasını göster

Bu, alternatif site olağanüstü durumdan kurtarma senaryosudur.

Paylaşılan nesnelere, Db2 kurtarma (A Db2 sistemi başarısız olduğunda) tarafından elde edilen zaman noktasına geri kazanılır. Her kuyruk yöneticisi, diğer yerde bulunan yedek kopyalardan elde edilebilir bir zaman noktasına kurtarılabilir.

Kalıcı iletiler kuyruk paylaşım gruplarında kullanılabilir ve MQSC RECOVER CFSTRUCT komutu kullanılarak kurtarılabilir. Bu komutun, hata zamanına geri çekildiğini unutmayın. Ancak, kalıcı olarak paylaşılan kuyruk iletilerinin kurtarılması yoktur; CSQUTIL yardımcı programının COPY işlevini kullanarak yedek kopya üretmediğiniz sürece bunlar kaybedilir.

Farklı kuyruk yöneticilerindeki (gerçekte kurtarılmakta olan) yerel nesnelere arasında karşılıklı bağımlılık olmadığından ve yeniden başlatma sırasında Db2 ile kuyruk yöneticisi yeniden eşzamanlama işlemi, kuyruk yöneticisi temelinde kuyruk yöneticisinde gereken şekilde COPY nesnelere yaratması ya da silmesi gerektiğinden, her kuyruk yöneticisini aynı zaman noktasına geri yükleme girişiminde bulunmanız gerekmez.

Kuyruk yöneticilerinizi otomatik olarak yeniden başlatmak için ARM ' yi nasıl kullanabileceğinizin anlaşılması için bu konuyu kullanın.

Bu bölümde aşağıdaki konulara ilişkin bilgiler yer almaktadır:

- [“ARM nedir?” sayfa 496](#)
- [“ARM ilkeleri” sayfa 496](#)
- [“IBM MQ ağında ARM ' nin kullanılması” sayfa 498](#)

ARM nedir?

z/OS Automatic Restart Manager (ARM), kuyruk yöneticilerinizin kullanılabilirliğini artırabilen bir z/OS kurtarma işlecidir. Bir iş ya da görev başarısız olduğunda ya da çalışmakta olduğu sistem başarısız olduğunda ARM, işi ya da görevi işletmen müdahalesini gerçekleştirilmeden yeniden başlatabilir.

Bir kuyruk yöneticisi ya da kanal başlatıcısı başarısız olursa, ARM aynı z/OS görüntüsünde yeniden başlatır. z/OS ve dolayısıyla ilgili altsistemler ve uygulamalar başarısız olursa ARM, sistem birleşimi içindeki başka bir z/OS görüntüsünde, önceden tanımlanmış bir sırayla tüm hatalı sistemleri otomatik olarak yeniden başlatabilir. Buna *sistemler arası yeniden başlatma* adı verilir.

Kanal başlatıcısı ARM tarafından yalnızca istisnai durumlarda yeniden başlatın. Kuyruk yöneticisi ARM tarafından yeniden başlatılırsa, kanal başlatıcısı CSQINP2 kullanıma hazırlama veri kümesinden yeniden başlatın (bkz. [“IBM MQ ağında ARM ' nin kullanılması” sayfa 498](#)).

z/OS hatası durumunda, sistem birleşimi içindeki farklı bir z/OS görüntüsünde kuyruk yöneticisini yeniden başlatmak için ARM ' yi kullanabilirsiniz. The network implications of IBM MQ ARM restart on a different z/OS image are described in [“IBM MQ ağında ARM ' nin kullanılması” sayfa 498](#).

Otomatik yeniden başlatmayı etkinleştirmek için:

- Bir ARM çifti veri kümesi ayarlayın.
- z/OS ' in bir *ARM ilkesinde* gerçekleştirilmesini istediğiniz otomatik yeniden başlatma işlemlerini tanımlayın.
- ARM ilkesini başlat.

Ayrıca, IBM MQ başlatma sırasında ARM ile kaydolmalıdır (bu durum otomatik olarak gerçekleşir).

Not: Kuyruk yöneticilerini farklı z/OS görüntülerinde otomatik olarak yeniden başlatmak istiyorsanız, her kuyruk yöneticisini bir sistem olarak tanımlamanız gereken her z/OS görüntüsünde, kuyruk yöneticisinin yeniden başlatılabileceği, sistem birleşimi (sysplex) geniş bir benzersiz dört karakterlik altsistem adı ile tanımlamalısınız.

ARM çift veri kümeleri

ARM için gerekli olan iki veri kümesini tanımladığınızdan ve ARM desteği istediğiniz herhangi bir kuyruk yöneticisini başlatmadan önce çevrimiçi ve etkin durumda olduklarını doğrulayın. IBM MQ automatic ARM registration fails if the couple data sets are not available at queue manager startup. Bu durumda IBM MQ , çift veri kümesinin bulunmasının ARM desteği istemediğiniz anlamına geldiğini varsayar ve başlatma işlemi devam eder.

ARM çifti veri kümeleriyle ilgili bilgi için [z/OS MVS Setting up a Sysplex](#) başlıklı konuya bakın.

Automatic Restart Manager ilkeleri, kuyruk yöneticisinin herhangi bir yeniden başlatılmasını denetleyebilecek ARM işlevlerini denetleyen kullanıcı tanımlı kurallardır.

ARM işlevleri, kullanıcı tanımlı bir *ARM ilkesi* tarafından denetlenir. ARM tarafından yeniden başlatılacak bir kuyruk yöneticisi yönetim ortamını çalıştıran her z/OS görüntüsü, etkin ARM ilkesiyle bir ARM çiftine bağlanmalıdır.

IBM, varsayılan bir ARM ilkesi sağlar. You can define new policies, or override the policy defaults by using the *yönetim verileri yardımcı programı* (IXCMIAPU) provided with z/OS. *z/OS MVS Setting up a Sysplex*, bu yardımcı programı açıklar ve bir ARM ilkesinin nasıl tanımlamaya ilişkin tüm ayrıntıları içerir.

Şekil 44 sayfa 497, ARM ilkesine bir örnek gösterir. Bu örnek ilke, bir sistem birleşimi (sysplex) içindeki kuyruk yöneticisini yeniden başlatır (kuyruk yöneticisi başarısız olursa ya da bir sistem başarısız olursa).

```
//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(SYSQMGRM*) /* These jobs to be restarted by ARM */
/*
```

Şekil 44. Örnek ARM ilkesi

Daha fazla bilgi için bakınız:

- [ARM ilkesi tanımlanması](#)
- [ARM ilkesinin etkinleştirilmesi](#)
- [ARM ile kayda geçiriliyor](#)

ARM ilkesi tanımlanması

ARM ilkenizi aşağıdaki gibi ayarlayın:

- Her kuyruk yöneticisi yönetim ortamı için, o kuyruk yöneticisi yönetim ortamına bağlanan herhangi bir CICS ya da IMS altsistemi için RESTART_GROUPS tanımlayın. Bir altsistem adlandırma kuralı kullanıyorsanız, '?' seçeneğini kullanabilirsiniz. Alt sınır tanımlama çalışması ile RESTART_GROUPS tanımlamak için öge adlarınızda '*' genel arama karakteriyle joker karakter karakterlerini tanımlayabilirsiniz.
- Kanal başlatıcılarının yalnızca kanal başlatıcı başarısız olduğunda ve z/OS görüntüsü başarısız olduğunda yeniden başlatılacağını belirtmek için TERMTYPE (ELECTYPE) değerini belirtin.
- Kuyruk yöneticilerinizde kuyruk yöneticisi başarısız olursa ya da z/OS görüntüsü başarısız olursa, bunların yeniden başlatılacağını belirtmek için kuyruk yöneticilerinize ilişkin TERMTYPE (ALLTERM) değerini belirtin.
- Hem kuyruk yöneticileri hem de kanal başlatıcıları için RESTART_METHOD (HER IKISI, KALICI) BELIRTIN. Bu, ARM 'nin son başlatma sırasında saklanan JCL' yi (sistem simgeleriyle çözüldükten sonra) kullanarak yeniden başlatmasını bildirir. Bu, ARM ' ye tek tek ögenin başarısız olup olmadığını ya da z/OS görüntüsünün başarısız olup olmadığına ilişkin olarak bunu yapmasının gerektiğini bildirir.
- Diğer tüm ARM ilkesi seçenekleri için varsayılan değerleri kabul edin.

ARM ilkesini etkinleştirme

Otomatik yeniden başlatma yönetimi ilkenizi başlatmak için şu z/OS komutunu verin:

```
SETXCF START,POLICY,TYPE=ARM,POLNAME= mypol
```

İlke başlatıldığında, ARM çiftine bağlı tüm sistemler aynı etkin ilkeyi kullanır.

Otomatik yeniden başlatma işlemlerini geçersiz kılmak için SETXCF STOP komutunu kullanın.

ARM ile kaydolma

Kuyruk yöneticisi başlatma sırasında (ARM kullanılabilirliğine tabi),IBM MQ otomatik olarak bir *ARM ögesi* olarak kayıt yaptır. Kapanma aşamasında, talep etmemek istenmediği sürece kayıt dışı.

Başlatma sırasında, kuyruk yöneticisi ARM ' nin kullanılabilir olup olmadığını belirler. Bu durumda, IBM MQ , SYSMQGR *ssid* adını kullanarak kaydolar; burada *ssid* , dört karakter kuyruğu yöneticisi adıdır ve SYSMQGR öge tipidir.

STOP QMGR MODE (QUIESCE) ve STOP QMGR MODE (FORCE) komutları, kuyruk yöneticisini ARM 'den (başlatma sırasında ARM' ye kaydedtirildiyse) kayıttan kaldırır. Bu, ARM ' nin bu kuyruk yöneticisini yeniden başlatmasını önler. STOP QMGR MODE (RESTART) komutu, kuyruk yöneticisini ARM ' den kayıttan kaldırmaz; bu nedenle, otomatik yeniden başlatma işlemi için uygun olur.

Her kanal başlatıcı adres alanı, ARM ' nin kullanılabilir olup olmadığını belirler ve SYSMQCH *ssid* öge adıyla kaydedilirse, burada *ssid* kuyruk yöneticisi adı ve SYSMQCH ise öge tipidir.

Kanal başlatıcı, olağan durumda durduğunda her zaman ARM ' den kayıttan kaldırılır ve yalnızca olağandışı sona ererse kayıtlı kalır. Kuyruk yöneticisi başarısız olursa, kanal başlatıcı her zaman kayıttan kaldırılır.

IBM MQ ağında ARM ' nin kullanılması

Kuyruk yöneticisi yeniden başlatıldığında, kanal başlatıcılarının ve ilişkili dinleyicilerin kuyruk yöneticisi yeniden başlatıldığında otomatik olarak başlatılabilmesini istediğiniz kuyruk yöneticisini ayarlayabilirsiniz.

Hem LU 6.2 hem de TCP/IP iletişim protokolleri için aynı z/OS görüntüsünde tam olarak otomatik kuyruk yöneticisi yeniden başlatıldığından emin olmak için:

- CSQINPX veri kümesine uygun START LISTENER komutunu ekleyerek dinleyicilerinizi otomatik olarak başlatın.
- Start your channel initiator automatically by adding the appropriate START CHINIT command to the CSQINP2 data set.

TCP/IP ya da LU6.2 ile kuyruk yöneticisini yeniden başlatmak için bkz.

- [“TCP/IP ' yle birlikte farklı bir z/OS görüntüsünde yeniden başlatma” sayfa 498](#)
- [“LU 6.2 ile farklı bir z/OS görüntülerinde yeniden başlatma” sayfa 500](#)

CSQINP2 ve CSQINPX veri kümeleriyle ilgili bilgi için [Görev 13: Kullanıma hazırlama giriş veri kümelerinin uyarlanması](#) başlıklı konuya bakın.

TCP/IP ' yle birlikte farklı bir z/OS görüntüsünde yeniden başlatma

İletişim protokolünüz olarak TCP/IP kullanıyorsanız ve sanal IP adresleri kullanıyorsanız, diğer z/OS görüntülerinde kurtarılacak şekilde bunları yapılandırabilir ve bu kuyruk yöneticisine bağlanan kanalların herhangi bir değişiklik olmadan yeniden bağlanmasını sağlayacak şekilde yapılandırabilirsiniz. Ters durumda, bir kuyruk yöneticisini yalnızca kümeleri kullanıyorsanız ya da WLM dinamik Etki Alanı Ad Sistemi (DNS) mantıksal grup adı kullanarak bir kuyruk paylaşım grubuna bağlıyorsanız, bir kuyruk yöneticisini farklı bir z/OS görüntülerine taşıdıktan sonra yeniden bir TCP/IP adresi yeniden yerleştirebilirsiniz.

- [Kümeleme kullanılırken](#)
- [Bir kuyruk paylaşım grubuna bağlanılırken](#)

Kümeleme kullanılırken

z/OS ARM, aynı sistem birleşiminde farklı bir z/OS görüntüsünde kuyruk yöneticisini yeniden başlatarak, sistem hatasına yanıt verir; bu sistemin özgün z/OS görüntüsü için farklı bir TCP/IP adresi vardır. Aşağıda, bir kuyruk yöneticisinin TCP/IP adresini ARM tarafından farklı bir z/OS görüntüsünde taşıdıktan sonra yeniden atamak için IBM MQ kümelerini nasıl kullanabileceğiniz açıklanmıştır.

Bir istemci kuyruk yöneticisi, kuyruk yöneticisi hatasını saptadığında (kanal hatası olarak), küme iletim kuyruğunda uygun iletilerin hedef küme kuyruğunda farklı bir yönetim ortamını barındıran farklı bir sunucu kuyruk yöneticisine yeniden yerleştirilerek yanıt verir. Ancak, benzerlikli kısıtlar tarafından özgün sunucuya bağlı iletileri ya da sunucu kuyruk yöneticisi, toplu iş sonu işlemi sırasında başarısız olduğu için, belirsiz iletiler yeniden yerleştirilemiyor. Bu iletileri işlemek için aşağıdaki işlemleri yapın:

1. Her bir z/OS kuyruk yöneticisine farklı bir TCP/IP kapısı ve farklı bir küme alıcılı kanal adı ayırın. Her kuyruk yöneticisinin farklı bir kapiya gereksinimi vardır; böylece, iki sistem z/OS görüntüsünde tek bir TCP/IP yığınını paylaşır. One of these is the queue manager originally running on that z/OS image, and the other is the queue manager that ARM will restart on that z/OS image following a system failure. Her bir z/OS görüntüsünde her bir kapiyu yapılandırın; böylece, ARM herhangi bir z/OS görüntüsünde kuyruk yöneticisini yeniden başlatabilirler.
2. Kanal başlatıcısı başlatma sırasında gönderme yapılacak her kuyruk yöneticisi ve z/OS görüntü birleşimi için farklı bir kanal başlatıcı komut giriş dosyası (CSQINPX) yaratın.

Her CSQINPX dosyası, o kuyruk yöneticisine özgü bir START LISTENER PORT (kapi) komutu ve kuyruk yöneticisi ve z/OS görüntü birleşimine özgü bir küme alıcı kanalı için bir ALTER KANAL komutu içermelidir. ALTER CHANNEL komutunun bağlantı adını, yeniden başlatıldığı z/OS görüntüsünün TCP/IP adına ayarlamaya gerek vardır. Bağlantı adının bir parçası olarak, yeniden başlatılan kuyruk yöneticisine özgü kapi numarasını içermelidir.

Her kuyruk yöneticisinin başlangıç JCL ' si bu CSQINPX dosyası için sabit veri kümesi adına sahip olabilir ve her bir z/OS görüntüsü, paylaşılmayan bir DASD birimindeki her CSQINPX dosyasının farklı bir sürümüne sahip olmalıdır.

Bir ARM yeniden başlatma işlemi gerçekleşirse, IBM MQ değiştirilen kanal tanımlamasını küme havuzuna tanıtır ve bu da bunu sunucu kuyruk yöneticisine ilgi gösteren tüm istemci kuyruğu yöneticilerine yayınlar.

İstemci kuyruk yöneticisi, sunucu kuyruk yöneticisi hatasını bir kanal hatası olarak değerlendirir ve başarısız olan kanalı yeniden başlatmaya çalışır. İstemci kuyruk yöneticisi yeni sunucu bağlantısı adını öğrendiğinde, kanal yeniden başlatıldığında istemci kuyruk yöneticisi yeniden başlatılan sunucu kuyruğu yöneticisine yeniden bağlanır. İstemci kuyruk yöneticisi iletileri yeniden uyumlulaştırabilir, istemci kuyruk yöneticisinin iletim kuyruğunda belirsiz olan iletileri çözümlenebilir ve olağan işlemler devam edebilir.

Kuyruk paylaşım grubuna bağlanılırken

Bir TCP/IP dinamik Etki Alanı Ad Sistemi (DNS) mantıksal grup adı aracılığıyla bir kuyruk paylaşım grubuna bağlanırken, kanal tanımınızdaki bağlantı adı, fiziksel makinenin anasistem adı ya da IP adresi değil, kuyruk paylaşım grubunuzun mantıksal grup adını belirtir. Bu kanal başlatıldığında, dinamik DNS ' ye bağlanır ve daha sonra kuyruk paylaşım grubundaki kuyruk yöneticilerinden birine bağlanır. Bu işlem Kuyruk paylaşım gruplarını kullanarak IBM MQ for z/OS için iletişimi ayarlama ' ta açıklanmaktadır.

Bir görüntü hatası olasılığının düşük olması durumunda, aşağıdakilerden biri gerçekleşir:

- Başarısız olan görüntüdeki kuyruk yöneticileri, sistem şebekeniz üzerinde çalışan dinamik DNS ' den kayıt yaptır. Kanal, RETRING (yeniden deneme) durumuna girerek bağlantı hatasına yanıt verir ve ardından sistem şebekesinde çalışan dinamik DNS ' ye bağlanır. Dinamik DNS, gelen isteği, kalan görüntülerde hala çalışmakta olan kuyruk paylaşım grubunun kalan üyelerinden birine ayırır.
- Kuyruk paylaşım grubunda başka bir kuyruk yöneticisi etkin değilse ve ARM kuyruk yöneticisini ve kanal başlatıcısını farklı bir görüntüyle yeniden başlattıysa, grup dinleyicisi bu yeni görüntüden dinamik DNS ' e kaydolar. Bu, mantıksal grup adının (kanalın bağlantı adı alanındaki) dinamik DNS ' e kaydolar.

ye bağlandığı ve daha sonra aynı kuyruk yöneticisine bağlı olduğu, şimdi farklı bir görüntü üzerinde çalıştırıldığı anlamına gelir. Kanal tanımlaması için değişiklik yapılması gerekmedi.

Bu tür bir kurtarma işlemi gerçekleştirilmek üzere aşağıdaki noktalara dikkat edilmelidir:

- z/OS üzerinde, dinamik DNS, sistem birleşimi (sysplex) içindeki z/OS görüntülerinden birinde çalışır. Bu görüntü başarısız olursa, dinamik DNS konfigürasyonunun, birincil ad sunucusuna alternatif olarak işlev gören sistem birleşimi (sysplex) içinde etkin bir ikincil ad sunucusu olması için yapılandırılması gerekir. Birincil ve ikincil dinamik DNS sunucularına ilişkin bilgiler [OS/390 SecureWay CS IP Configuration](#) (CS IP Yapılandırması) kılavuzunda bulunabilir.
- TCP/IP grup dinleyicisi, bu z/OS görüntüsünde bulunmayabilir, belirli bir IP adresinde başlatılmış olabilir. Böyle bir durumda, dinleyiciye yeni görüntüdeki farklı bir IP adresi üzerinde başlatılması gerekebilir. Sanal IP adresleri kullanıyorsanız, bu bilgileri diğer z/OS görüntülerinde kurtarılacak şekilde yapılandırabilirsiniz; böylece, START LISTENER komutu için herhangi bir değişiklik yapılması gerekmez.

LU 6.2 ile farklı bir z/OS görüntülerinde yeniden başlatma

Yalnızca LU 6.2 iletişim protokollerini kullanıyorsanız, sistem birleşimi içindeki farklı bir z/OS görüntüsünde kuyruk yöneticisinin otomatik olarak yeniden başlatılmasından sonra ağ yeniden bağlanmasını etkinleştirmek için aşağıdaki yordamı kullanın:

- Sistem birleşimi içindeki her kuyruk yöneticisini benzersiz bir altsistem adıyla tanımlayın.
- Her kanal başlatıcısını, benzersiz bir LUNAME ile sysplex içinde tanımlayın. Bu değer, hem kuyruk yöneticisi özniteliklerinde hem de START LISTENER komutunun içinde belirtilir.

Not: LUNAME, APPC yan çizelgesindeki bir girişle eşleşir ve bu da gerçek LUNAME ' ye eşlenir.

- Sysplex içindeki her z/OS görüntüsü tarafından gönderme yapılan bir paylaşılan APPC yan çizelgesi ayarlayın. Bu, her kanal başlatıcısı LUNAME için bir giriş içermelidir. Bu konuda bilgi almak için [z/OS MVS Planning: APPC/MVS Management](#) başlıklı konuya bakın.
- Set up an APPCPM xx member of SYS1.PARMLIB for each channel initiator within the sysplex to contain an LUADD to activate the APPC side table entry for that channel initiator. Bu üyeler, her bir z/OS görüntüsü tarafından paylaşılmalıdır. The appropriate SYS1.PARMLIB member is activated by a z/OS command SET APPC= xx, which is issued automatically during ARM restart of the queue manager (and its channel initiator) on a different z/OS image, as described in the following text.
- Her kanal başlatıcısı için bu SYS1.PARMLIB üyesinin xx sonekini belirtmek için LU62ARM kuyruk yöneticisi öznitelğini kullanın. Bu, kanal başlatıcının, LUNAME ' yi etkinleştirmek için gerekli z/OS komut SET APPC= xx komutunu yayınlamasına neden olur.

Kanal başlatıcısını yeniden başlatabilmesi için, ARM ilkesini, yalnızca z/OS görüntüsü yukarı ken başarısız olursa; XCFAS adres alanına ilişkin kullanıcı kimliğinin IBM MQ komutunu START CHINIT komutu verme yetkisi olması gerekir. Kanal başlatıcısını, kanal başlatıcı ve dinleyicilerini başlatmak için CSQINP2 ve CSQINPX veri kümelerinde kullanmak yerine, z/OS görüntüsü de başarısız olsa otomatik olarak yeniden başlatmayın.

► z/OS İş birimlerinin el ile kurtarılıyor

You can manually recover units of work CICS, IMS, RRS, or other queue managers in a queue sharing group. Kuyruk yöneticisiyle her bağlantıyla ilişkili iş birimlerinin durumunu görüntülemek için kuyruk yöneticisi komutlarını kullanabilirsiniz.

Bu konu, aşağıdaki konularla ilgili bilgileri içerir:

- [“Bağlantıların ve iş parçacıklarının görüntülenmesi” sayfa 501](#)
- [“Recovering CICS units of recovery manually” sayfa 501](#)
- [“Recovering IMS units of recovery manually” sayfa 505](#)
- [“RRS kurtarma birimlerinin el ile kurtarılması” sayfa 506](#)

- “Kuyruk paylaşım grubundaki başka bir kuyruk yöneticisinde kurtarma birimlerinin kurtarılması” sayfa 507

Bağlantıların ve iş parçacıklarının görüntülenmesi

Kuyruk yöneticilerine ve ilişkili iş birimlerine bağlantılarla ilgili bilgi almak için `DISPLAY CONN` komutunu kullanabilirsiniz. Şu anda neler olduğunu görmek için ya da kuyruk yöneticisinin kapanmasına izin vermek için neyin sonlandırılması gerektiğini görmek için etkin iş birimlerini görüntüleyebilirsiniz ve çözümlenmemiş iş birimlerinin kurtarılmasına yardımcı olacak şekilde görüntüleyebilirsiniz.

Etkin iş birimleri

Yalnızca etkin iş birimlerini görüntülemek için, şu bilgileri kullanın:

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ ACTIVE)
```

Çözülmeyen iş birimleri

"belirsiz iş parçacığı" olarak da bilinen çözülmemiş bir iş birimi, iki aşamalı kesinleştirme işleminin ikinci geçişindeki bir birimdir. Kaynaklar, kendi adına IBM MQ içinde tutulur. Çözülmemiş iş birimlerinin görüntülenmesi için,

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

Çözülmemiş iş birimlerinin durumunu çözmek için dış müdahaleye gerek vardır. Bu, yalnızca kurtarma koordinatörünün (CICS, IMSya da RRS) başlatılması ya da aşağıdaki bölümlerde açıklandığı gibi daha fazla içerilebileceği olabilir.

Recovering CICS units of recovery manually

CICS bağdaştırıcısı yeniden başlatıldığında ne olacağını anlamak için bu konuyu kullanın ve daha sonra, ortaya çıkan çözülmemiş tüm kurtarma birimleriyle nasıl başa çıkacağı açıklanır.

CICS bağdaştırıcısı yeniden başlatıldığında ne olur

Bir bağlantı kesildiğinde, bağdaştırıcının *yeniden bağlanma süreci* işlemi sırasında *yeniden başlatma aşaması* üzerinden geçmesi gerekir. Yeniden başlatma aşaması kaynakları yeniden eşzamanlıyor. CICS ile IBM MQ arasında yeniden eşzamanlama, belirsiz iş birimlerinin tanımlanmasına ve çözülmesine olanak sağlar.

Yeniden eşzamanlamanın nedeni şunlar olabilir:

- Dağıtılmış kuyruklama bileşeninden belirtik bir istek
- IBM MQ' a bir bağlantı yapıldığında örtük bir istek

Yeniden eşzamanlama IBM MQ' e bağlanarak kaynaklandıysa, olayların sırası şöyledir:

1. The connection process retrieves a list of in-doubt units of work (UOW) IDs from IBM MQ.
2. UOW tanıtıcıları, konsolda CSQC313I iletilerinde görüntülenir.
3. UOW tanıtıcıları CICS' a iletilir.
4. CICS , her bir belirsiz UOW kimliği için bir yeniden eşzamanlama görevi (CRSY) başlatır.
5. Konsolda her bir belirsiz UOW görevi için görevin sonucu görüntülenir.

Bağlanma işlemi sırasında görüntülenen iletileri denetlemeniz gerekir:

CSQC313I

Bir UOW ' un kuşku içinde olduğunu gösterir.

CSQC400I

UOW 'yi tanımlar ve bu iletilerden biri tarafından takip edilir:

- CSQC402I ya da CSQC403I , UOW ' nin başarıyla çözüldüğünü (kesinleştirildiğini ya da yedeklendiğini) gösterir.
- CSQC404E, CSQC405E, CSQC406E ya da CSQC407E , UOW ' un çözümlenmediğini gösterir.

CSQC409I

Tüm UOWs ' lerin başarıyla çözümlendiğini gösterir.

CSQC408I

Tüm UOWs ' lerin başarıyla çözümlenmediğini gösterir.

CSQC314I

* ile vurgulanan UOW tanıtıcılarının otomatik olarak çözümediği konusunda uyarır. Bu UOWs 'ler, yeniden başlatıldığında dağıtımlı kuyruğa alma bileşeniyle açık bir şekilde çözümlenmelidir.

Şekil 45 sayfa 502 , z/OS konsolunda görüntülenen yeniden başlatma iletileri örneğini gösterir.

```
CSQ9022I +CSQ1 CSQYASCP ' START QMGR' NORMAL COMPLETION
+CSQC323I VICIC1 CSQCQCON 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 45. Yeniden başlatma iletileri örneği

Toplam CSQC313I iletileri sayısı, toplam CSQC402I artı CSQC403I ileti sayısına eşit olmalıdır. Toplamlar eşit değilse, bağlantı sürecinin çözümlenememesi gereken UOWs 'ler vardır. Çözümlemeyen UOWs 'ler, CICS (örneğin, soğuk başlatma) ya da IBM MQ ile ya da kuyruklama dağıtımlarıyla ortaya çıkan sorunlardan kaynaklanır. Bu sorunlar giderildiğinde, bağlantıyı keserek ve daha sonra yeniden bağlanarak başka bir yeniden eşzamanlama başlatabilirsiniz.

Diğer bir seçenek olarak, CSQC400I iletilerinde gösterilen UOW tanıtıcısını ve RESOLSE INDOUBT komutunu kullanarak her bir üstün UOW 'yi kendiniz çözebilirsiniz. CICS'ta *kurtarma tanımlayıcıları birimi* 'nin temizlenmesi için bir bağlantı kesme ve bir bağlantı başlatma işlemi başlatmanız gerekir. UOW 'ları manuel olarak çözmek için UOW 'un doğru sonucunu bilmeniz gerekir.

Çözümlemeyen UOWs ile ilişkili tüm iletiler IBM MQ tarafından kilitlenir ve Toplu İş, TSO ya da CICS görevi bunlara erişemez.

CICS başarısız olursa ve acil bir yeniden başlatma gerekiyorsa, CICS sisteminin GENERIC APPLID ' yi *değiştirmeyin* . Daha sonra IBM MQ' e yeniden bağlanıyorsanız, IBM MQ ile veri bütünlüğü güvenceye alınamaz. This is because IBM MQ treats the new instance of CICS as a different CICS (because the APPLID is different). Belirsiz çözümlenme, daha sonra yanlış CICS günlüğüne dayalıdır.

How to resolve CICS units of recovery manually

Bağdaştırıcının olağan dışı bir şekilde sona ermesi durumunda, CICS ve IBM MQ oluşturma işlemi, olağandışı bir şekilde ya da yeniden başlatma sırasında, hangi altsistemin olağandışı bitmesine bağlı olarak değişir.

Not: İş birimlerini göstermek için DFH\$INDB örnek programını kullanırsanız, IBM MQ UOWs ' yi her zaman doğru göstermediğini de bulabilirsiniz.

CICS IBM MQ' a bağlandığında, çözülmemiş bir ya da daha çok kurtarma birimi olabilir.

Konsola aşağıdaki iletilerden biri gönderilir:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E
- CSQC408I

Bu iletilerin ne anlama geliveriyle ilgili ayrıntılar için [CICS bağdaştırıcısı ve Köprü iletileri](#) iletilerine bakın.

CICS , bağlantı başlatma sırasında çözülmeyen kurtarma birimlerinin ayrıntılarını saklar. Bir girdi, IBM MQtarafından sunulan listede artık görünmediğinde temizlenir.

CICS ' in çözemeyeceği herhangi bir kurtarma birimi, IBM MQ komutları kullanılarak el ile çözümlenmelidir. Bu el ile yordam, yalnızca işletim hatalarının ya da yazılım sorunlarının otomatik olarak çözülmesini önlediği için, kuruluş içinde nadiren kullanılır. *Kuşku içinde çözülme sırasında bulunan tutarsızlıklar araştırılmalıdır.*

Kurtarma birimlerini çözmek için:

1. Aşağıdaki komutu kullanarak IBM MQ ' dan kurtarma birimlerinin bir listesini edinin:

```
+CSQ1 DISPLAY CONN( * ) WHERE(UOWSTATE EQ UNRESOLVED)
```

Aşağıdaki iletiyi alırsınız:

```
CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN (BC85772CBE3E0001)
EXTCONN (C3E2D8C3C7D9F0F9404040404040)
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, tutarlılık noktası günlük girişlerinin yazıldığı sırada CICS tarafından sağlanan CICS applid numarasını ve benzersiz bir numaradan oluşur. This unique number is stored in records written to both the CICS system log and the IBM MQ log at syncpoint processing time. Bu değer, CICS içinde *kurtarma simgesi* olarak adlandırılır.

2. Belirli bir kurtarma birimiyle ilgili girişler için CICS günlüğünü tarayın.

Kurtarma simgesi alanının (JCSRMTKN) ağ tanıtıcısından elde edilen değere eşit olduğu, göreve ilişkin kuruluş için bir PREPARE kaydı olup olmadığını arayın. Ağ tanıtıcısı, DISPLAY CONN komutu çıkışında IBM MQ tarafından sağlanır.

Kurtarma birimleri için CICS günlüğünde PREPARE kaydı, CICS görev numarasını sağlar. Bu CICS görevine ilişkin günlükte bulunan diğer tüm girişler bu sayı kullanılarak bulunabilir.

Günlüğü taradığınızda CICS günlük yazdırma yardımcı programı DFHJUP programını kullanabilirsiniz. Bu programın kullanılmasına ilişkin ayrıntılar için *CICS Operations and Utilities Guide* adlı yayına bakın.

3. Scan the IBM MQ log for records with the NID related to a particular unit of recovery. Daha sonra bu kurtarma birimine ilişkin günlük kayıtlarının geri kalanını elde etmek için bu kayıttan URID ' yi kullanın.

IBM MQ günlüğünü tararken, IBM MQ başlatma iletisinin CSQJ001I tarafından bu oturum için RBA başlatma işlemini sağladığını göz önünde bulundurun.

Yazdırma günlüğü kayıtları programı (CSQ1LOGP) bu amaçla kullanılabilir.

4. Gereksinim duyarsanız, IBM MQ' ta belirsiz bir şekilde çözümler.

IBM MQ , bir IBM MQ BELIRSIZ KALICI komutu kullanılarak kurtarma birimi için kurtarma işlemini yapmak üzere yönlendirilebilir.

Belirli bir *bağlantı-adi* ile ilişkili tüm iş parçacıklarını kurtarmak için NID (*) seçeneğini kullanın.

Komut, iş parçacığın kesinleştirildiğini ya da yedeklenip yedeklenmediğini gösteren aşağıdaki iletilerden birini üretir:

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id ABORT SCHEDULED
```

Belirsiz durumda çözümlerken, CICS ve bağdaştırıcı, yalnızca IBM MQ kaynakları etkilendiği için kurtarma birimlerini kesinleştirmek ya da yedeklemek için IBM MQ ' e komutlardan haberdar değildir. However,

CICS keeps details about the in-doubt threads that could not be resolved by IBM MQ. Bu bilgiler, sunulan liste boş olduğunda ya da liste, ayrıntılara sahip CICS ' in bir kurtarma birimini içermiyorsa temizlenir.

z/OS Recovering IMS units of recovery manually

IMS bağdaştırıcısı yeniden başlatıldığında ne olacağını anlamak için bu konuyu kullanın ve daha sonra, ortaya çıkan çözümlenmemiş tüm kurtarma birimleriyle nasıl başa çıkacağı açıklanır.

IMS bağdaştırıcısı yeniden başlatıldığında ne olur

IBM MQ ile bağlantı ne zaman yeniden başlatıldığında, bir kuyruk yöneticisi yeniden başlatıldığı ya da bir IMS /START SUBSYS komutu sonrasında IMS , aşağıdaki yeniden eşzamanlama işlemini başlatır:

1. IMS , Commit (Kesinleştirme) ya da Backout (Geri) çözümlüğünü içeren bir parametreyle IBM MQ IMS bağdaştırıcısına kuşku içinde olduğuna inandığı iş birimi (UOW) tanıtıcılarının listesini gösterir.
2. IMS bağdaştırıcısı, çözümlenme isteğini IBM MQ 'e iletir ve sonucu IMS'a geri gönderir.
3. Having processed all the IMS resolution requests, the IMS adapter gets from IBM MQ a list of all UOWs that IBM MQ still holds in doubt that were initiated by the IMS system. Bu bilgiler, CSQQ008Iiletisinde IMS ana uçbirimine bildirilir.

Not: Bir UOW şüphesi olduğunda, ilişkili IBM MQ iletisi IBM MQ tarafından kilitlenir ve herhangi bir uygulama tarafından kullanılamaz.

How to resolve IMS units of recovery manually

IMS IBM MQ'e bağlandığında, IBM MQ ' un bir ya da daha fazla belirsiz kurtarma birimi çözümlenmemiş olabilir.

IBM MQ 'de IMS ' in çözmediği belirsiz kurtarma birimleri varsa, IMS ana uçbiriminde aşağıdaki ileti yayınlanır:

```
CSQQ008I nn units of recovery are still in doubt in queue manager qmgr-name
```

Bu ileti yayınlanırsa, IMS ya soğuk başlatıldı ya da tamamlanmamış bir günlük manyetik bantı ile başlatıldı. Bu ileti, IBM MQ ya da IMS bir yazılım hatası ya da diğer altsistem hatası nedeniyle olağan dışı olarak sonlandırılırsa da yayınlanabilir.

CSQQ008I iletisi alındıktan sonra:

- Bağlantı etkin durumda kalır.
- IMS uygulamaları, IBM MQ kaynaklarına erişmeye devam edebilir.
- Bazı IBM MQ kaynakları kilitli kalmaya devam eder.

Belirsiz iş parçacığı çözülmezse, IMS ileti kuyrukları oluşturulabiliyor. IMS kuyrukları kapasiteyi doldurursa, IMS sona erer. Bu olası zorluktan haberdar olmalısınız ve belirsiz kurtarma birimleri tam olarak çözümleninceye kadar IMS ' u izlemeniz gerekir.

Kurtarma yordamı

IMS iş birimlerinin kurtarılması için aşağıdaki yordamı kullanın:

1. Force the IMS log closed, using /SWI OLDS, and then archive the IMS log. Use the utility, DFSERA10, to print the records from the previous IMS log tape. Tip X ' 3730 ' günlük kayıtları phase-2 kesinleştirme isteğini belirtir ve X ' 38 ' günlük kayıtları bir iptal isteğini belirtir. Her bağımlı bölgede son işlem için istenen işlemi kaydedin.
2. Kesinleştirme noktasına ulaşmamış olan her PSB ' yi yedeklemek için DL/I toplu işi çalıştırın. İşlemler hala işlenmekte olduğu için işlem biraz zaman alabilir. Ayrıca, işlemin geri kalanını ve ileti kuyruklarının geri kalanını etkileyebilecek bir dizi kaydı da kilitleyebilir.

3. Produce a list of the in-doubt units of recovery from IBM MQ using the following command:

```
+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)
```

For IMS, the NID consists of the IMS connection name and a unique number provided by IMS. Değer, IMS içinde *kurtarma simgesi* olarak adlandırılır. Daha fazla bilgi için *IMS Customization Guide* adlı belgeye bakın.

4. Compare the NIDs (IMSID plus OASN in hexadecimal) displayed in the DISPLAY THREAD messages with the OASNs (4 bytes decimal) shown in the DFSERA10 output. Kesinleştirmeye mi, yoksa geri mi verileceğine karar verin.
5. Perform in-doubt resolution in IBM MQ with the [BELIRSİZ KALICI](#) command, as follows:

```
RESOLVE INDOUBT( connection-name )
ACTION(COMMIT|BACKOUT)
NID( network-id )
```

bağlantı-adı bağlantısıyla ilişkili tüm iş parçacıklarını kurtarmak için NID (*) seçeneğini kullanın. Komut, iş parçacığın kesinleştirildiğini ya da yedeklenip yedeklenmediğini belirtmek için aşağıdaki iletilerden birindeki sonuç sonuçlarını gösterir:

```
CSQV414I THREAD network-id COMMIT SCHEDULED
CSQV415I THREAD network-id BACKOUT SCHEDULED
```

Belirsiz durumda çözülürken, IMS ve bağdaştırıcı, yalnızca IBM MQ kaynakları etkilendiği için, IBM MQ 'e verilen komutların farkında değildir ve kesinleştirilmemiş kurtarma birimleri geri çekilir.

RRS kurtarma birimlerinin el ile kurtarılması

Bu konuyu, belirsiz RS kurtarma birimleri olup olmadığını nasıl belirleyeceğini ve kurtarma birimlerinin nasıl el ile çözümleneceğini anlamak için bu konuyu kullanın.

RRS IBM MQ'a bağlandığında, IBM MQ 'un bir ya da daha fazla kuşku içinde kurtarma işlemi çözülmemiş olabilir. IBM MQ 'de RRS' nin çözmediği belirsiz kurtarma birimleri varsa, z/OS konsolunda aşağıdaki iletilerden biri yayınlanır:

- CSQ3011I
- CSQ3013I

- CSQ3014I
- CSQ3016I

Hem IBM MQ hem de RRS, belirsiz kurtarma birimleriyle ilgili bilgileri ve el ile çözümlmek için teknikler hakkında bilgi görüntülemek için araçlar sağlar.

IBM MQ' ta, belirsiz IBM MQ iş parçacıklarına ilişkin bilgileri görüntülemek için DISPLAY CONN komutunu kullanın. Komuttan elde edilen çıkış, bir eşgüdümçü olarak RRS ' ye sahip IBM MQ iş parçacıklarına ilişkin RRS kurtarma tanıtıcılarını içerir. Bu, kurtarma biriminin sonucunu belirlemek için kullanılabilir.

Use the RESOLVE INDOUBT command to resolve the IBM MQ in-doubt thread manually. Bu komut, doğru kararın ne olduğunu belirledikten sonra kurtarma birimini kesinleştirmek ya da yedeklemek için kullanılabilir.

Kuyruk paylaşım grubundaki başka bir kuyruk yöneticisinde kurtarma birimlerinin kurtarılması

Bir kuyruk paylaşım grubundaki diğer kuyruk yöneticilerindeki kurtarma birimlerini tanımlamak ve el ile kurtarma işlemini yapmak için bu konuyu kullanın.

Kuyruk paylaşım grubunun üyesi olan bir kuyruk yöneticisi başarısız olursa ve yeniden başlatılamazsa, gruptaki diğer kuyruk yöneticileri eşdüzey kurtarma işlemini gerçekleştirebilir ve bu işlemi devralabilirler. Ancak, bu kurtarma biriminin son atma işlemi yalnızca başarısız olan kuyruk yöneticisinde bilindiğinden, kuyruk yöneticisinin eşdüzey kurtarma ile çözülemeyecek belirsiz durumda kurtarma birimleri olabilir. Bu kurtarma birimleri, kuyruk yöneticisi sonunda yeniden başlatıldığında çözümlenir, ancak o zamana kadar kuşku kalır.

Başka bir deyişle, bazı kaynaklar (örneğin, iletiler) kilitlemiş olabilir ve gruptaki diğer kuyruk yöneticilerine kullanılamaz duruma getirilebilir. Bu durumda, etkin olmayan kuyruk yöneticisinde bu iş birimlerini görüntülemek için DISPLAY THREAD komutunu kullanabilirsiniz. Bu kurtarma birimlerinin, gruptaki diğer kuyruk yöneticilerinin kullanımına sunulmasını sağlamak için el ile çözümlmek 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 yayınlayın:

```
+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 yayınlar:

```
CSQV435I CANNOT USE QMNAME KEYWORD, QM01 IS ACTIVE
```

Belirsiz durumda olan iş parçacıklarını el ile çözmek için IBM MQ komutunu RESOLVE INDOUBT (belirsiz) komutunu kullanın. Komutta etkin olmayan kuyruk yöneticisinin adını belirtmek için QMNAME anahtar sözcüğünü kullanın.

Bu komut, kurtarma birimini kesinleřtirmek ya da yedeklemek için kullanılabilir. Komut, yalnızca kurtarma biriminin paylaşılan kısmını çözer; yerel iletiler etkilenmez ve kuyruk yöneticisi yeniden başlatılıncaya kadar kilitli kalır ya da CICS, IMSya da RRS toplu işine yeniden bağlanır.

z/OS IBM MQ ve IMS

IBM MQ , IMS, IBM MQ - IMS bağdařtırıcısı ve IBM MQ - IMS köprüsüyle arabirim için iki bileşen sağlar. Bu bileşenler yaygın olarak IMS bağdařtırıcısı ve IMS köprüsü olarak adlandırılır.

z/OS IMS bağdařtırıcısının çalıştırılması

IBM MQ bağdařtırıcısını IMS sistemlerine bağlayan IMS bağdařtırıcısının nasıl çalıştırılacağını anlamak için bu konuyu kullanın.

Not: IMS bağdařtırıcısı, herhangi bir işlem ve denetim panosunu kapsamaz.

Bu konu aşağıdaki bölümleri içerir:

- [“Controlling IMS connections” sayfa 508](#)
- [“IMS denetim bölgesinden bağlanma” sayfa 508](#)
- [“Belirsiz kurtarma birimlerinde görüntüleme” sayfa 510](#)
- [“IMS bağımlı bölge bağlantılarını denetleme” sayfa 512](#)
- [“IMSbağlantısı kesiliyor” sayfa 515](#)
- [“IMS tetikleme izleyicisini denetleme” sayfa 515](#)

z/OS Controlling IMS connections

IBM MQ' e bağlantıyı denetleyen ve izleyen IMS işleciyle ilgili komutları anlamak için bu konuyu kullanın.

IMS , IBM MQile bağlantıyı denetlemek ve izlemek için aşağıdaki işletmen komutlarını sağlar:

DEĞERLERİ

Deletes an in-doubt unit of recovery from IMS.

/GÖRÜNTÜLEME OASN ALT

Bekleyen kurtarma öğelerini görüntüler.

/SUBSYS ' I GÖRÜNTÜLE

Bağlantı durumunu ve iş parçacığı etkinliğini görüntüler.

/BAŞLATMA

IMS denetim bölgesini kuyruk yöneticisine bağlar.

/ALT SYS ' I

IMS bağlantısını bir kuyruk yöneticisinden bağlantıyı keser.

/İZLEME

IMS izlemesini denetler.

Bu komutlarla ilgili ek bilgi için, kullanmakta olduğunuz IMS düzeyi için *IMS/ESA Operator's Reference* adlı elkitabına bakın.

IMS komut yanıtları, komutun yayınlandığı uçbirime gönderilir. Authorization to issue IMS commands is based on IMS security.

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 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 MQiçin 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 çalıştığında ve *kuyruk yöneticisi etkin olmadığı*nda, IMS ana uçbirimine bir uygulama MQI çağrısı yaptığında 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 iletileri ve ilişkili CSQQ001I iletileri, 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) yaptığında iş parçacığı bağlantısı kurulur. Bu iş parçacığı, bölgenin süresi boyunca ya da bağlantı durduruluncaya kadar korunur.

İletileyle 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 iletilisine ekler.

Belirsiz kurtarma birimlerinde görüntüleme

Kurtarma birimlerinden kuşku görüntüleyebilir ve bunları kurtarmayı deneyebilirsiniz.

Bu konudaki kurtarma adımları listelemek ve kurtarmak için kullanılan işlem adımları, yalnızca göreceli olarak basit durumlar içindir. If the queue manager ends abnormally while connected to IMS, IMS might commit or back out work without IBM MQ being aware of it. Kuyruk yöneticisi yeniden başlatıldığında, bu iş *belirsiz* olarak tertiplenmektedir. İşin durumuyla ilgili bir karar verilmelidir.

Belirsiz durumdan kurtarma birimlerinin bir listesini görüntülemek için şu komutu verin:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

IBM MQ , aşağıdakine benzer bir iletiyle yanıt verir:

```
CSQM201I +CSQ1 CSQMDRTC DIS CONN DETAILS
CONN(BC0F6125F5A30001)
EXTCONN(C3E2D8C3C3E2D8F14040404040404040)
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(000000000000001F000000007472616E5F6964547565204E6F762020...)
QMURID(0000000026232)
URTYPE(XA)
USERID( )
APPLTAG(CSQ1CHIN)
ASID(0000)
APPLTYPE(CHINIT)
CHANNEL( )
CONNNAME( )
END CONN DETAILS
```

Bu iletteki özniteliklere ilişkin açıklamalar için [DISPLAY CONN](#) komutunun açıklamasına bakın.

Kurtarma sırasında kurtarılan kurtarma birimleri

Belirsiz kurtarma birimlerini kurtarmak için bu komutu çalıştırın:

```
+CSQ1 RESOLVE INDOUBT( connection-name ) ACTION(COMMIT|BACKOUT)
NID( net-node.number )
```

Burada:

connection-name

IMS sistem tanıtıcısı.

İŞLEM

Bu kurtarma biriminin kesinleştirilip kesinleştirilmeyeceğini (COMMIT) ya da geriye doğru (BACKOUT) geri verileceğini belirtir.

net-node.number

İlişkili net-node.number.

RESOLVE INDOUBT komutunu verdiğinizde, aşağıdaki iletilerden biri görüntülenir:

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id BACKOUT SCHEDULED
```

Yeniden elde edilen kurtarma girişlerinin çözümleniyor

IMS , belirli zamanlarda, yeniden kurtarma girişlerinin (RREs) bir listesini oluşturur. RREs are units of recovery about which IBM MQ might be in doubt. Bazı durumlarda ortaya çıkar:

- Kuyruk yöneticisi etkin değilse, IMS , kuyruk yöneticisi etkin oluncaya kadar çözülemeyen RREs ' lere sahiptir. Bu RRES ' ler bir sorun değil.
- If the queue manager is active and connected to IMS, and if IMS backs out the work that IBM MQ has committed, the IMS adapter issues message CSQQ010E. İki sistemdeki veriler tutarlı olması gerekiyorsa, bir sorun vardır. Bu sorunu çözmeye ilgili bilgi için bkz. [“Recovering IMS units of recovery manually” sayfa 505.](#)
- Kuyruk yöneticisi etkinse ve IMS' a bağlıysa, bu sorunu size bildirmiş hiçbir ileti olmasa da yine RREs olabilir. IMS ile IBM MQ bağlantısı kurulduktan sonra, bir sorun olup olmadığını öğrenmek için aşağıdaki IMS komutunu yayınlayabilirsiniz:

```
/DISPLAY OASN SUBSYS sysid
```

RRE ' yi temizlemek için, aşağıdaki IMS komutlarından birini yayınlayın:

```
/CHANGE SUBSYS sysid RESET
/CHANGE SUBSYS sysid RESET OASN nnnn
```

Burada *nnnn* , +CSQ1 DISPLAY komutuna yanıt olarak listelenen kaynak uygulama sırası numarasıdır. Bu, son IMS soğuk başlangıcından bu yana bu programa ilişkin çağrılar sırasında yer vererek, program örneğinin zamanlama numarasıdır. IMS , aynı zamanlama numarasına sahip iki adet belirsiz kurtarma kurtarma birimine sahip olamaz.

Bu komutlar, IMS ' in durumunu ilk durumuna getirir; IBM MQ ile yapılan hiçbir iletişimde sonuç vermez.

IMS bağımlı bölge bağlantılarını denetleme

You can control, monitor, and, when necessary, terminate connections between IMS and IBM MQ.

IMS bağımlı bölge bağlantılarının denetlenmesi aşağıdaki etkinlikleri içerir:

- [Bağımlı bölgelerden bağlanma](#)
- [Bölge hata seçenekleri](#)
- [Bağlantılardaki etkinliği izleme](#)
- [Bağımlı bölgelerden bağlantı kesiliyor](#)

Bağımlı bölgelerden bağlanma

Denetim bölgesinde kullanılan IMS bağıdaştırıcısı da bağımlı bölgelere yüklenir. Bağımlı bölgelerden IBM MQ' e bir bağılantı yapılır. Bu bağılantı, IBM MQ ve IMS çalışmalarının taahhüdlerini koordine etmek için kullanılır. Bağılantı başlatmak ve bağılantı yapmak için IMS aşağıdakileri gerçekleştirir:

1. Reads the subsystem member (SSM) from IMS.PROCLIB.

Bağımlı bölge EXEC parametresinde bir altsistem üyesi belirtilebilir. Bu değer belirlenmezse, denetim bölgesi SSM kullanılır. Bölge hiçbir zaman IBM MQ' a bağlanmadıysa, bağıdaştırıcının yüklenmesini önlemek için, girdi içermeyen bir üye belirtin.

2. IBM MQ bağıdaştırıcısını yükler.

Bir toplu ileti programı için, uygulama ilk ileti alışverişi komutunu yayınlayıncaya kadar yükleme işlemi gerçekleştirilmez. Bu sırada IMS , bağılantıyı yapmayı dener.

Bir ileti işleme programı bölgesi ya da IMS hızlı yolu bölgesi için, bölge kullanıma hazırlandığında girişimde bulunmaya çalışılır.

Bölge hata seçenekleri

Kuyruk yöneticisi etkin değilse ya da ilk ileti alışverişi komutu uygulama programlarından gönderildiğinde kaynaklar kullanılamıyorsa, işlem, SSM girişinde belirtilen hata seçeneğine bağılıdır. Seçenekler şunlardır:

R

Uygulamaya uygun dönüş kodu gönderilir.

Q

Uygulama, olağandışı bitiş kodu U3051ile olağandışı sona eriyor. Giriş iletisi yeniden kuyruğa alındı.

A

Uygulama, olağandışı bitiş kodu U3047ile olağandışı sona eriyor. Giriş iletisi atılır.

Bağılantılardaki etkinliği izleme

Bir uygulama ilk başarılı IBM MQ isteğini yerine getirdiğinde, bağımlı bir bölgeden bir iş parçacığı oluşturulur. You can display information about connections and the applications currently using them by issuing the following command from IBM MQ:

```
+CSQ1 DISPLAY CONN(*) ALL
```

Komut aşağıdakine benzer 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* özel değer DENETIMIDIR. Bağımlı bölgeler için, PSBname ile bitleştirilmiş PSTid 'dir. *auth-id*, iş kartından kullanıcı alanı ya da z/OS başlatma yordamları tablosundan tanıtıcıdır.

Görüntülenen listeye ilişkin açıklamalar için, [IBM MQ for z/OS iletileri](#), [tamamlama](#), ve [neden kodları](#) belgelerinde CSQV402I iletilerinin tanımına bakın.

IMS, IBM MQ ile bağlantıyı izlemek için bir görüntü birimi komutu sağlar. Bu program, her bir bağımlı bölge bağlantısında, LTERM kullanıcı adı ve denetim bölgesi bağlantı durumuyla ilgili hangi programın etkin olduğunu 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
```

Bağımlı bölgelerin her birinden iş parçacığı durumu aşağıdakilerden biridir:

```
CONN
CONN, ACTIVE (includes LTERM of user)
```

Bağımlı bölgelerden bağlantı kesiliyor

To change values in the SSM member of IMS.PROCLIB, you disconnect a dependent region. Bunu yapmak için aşağıdakileri yapmanız gerekir:

1. IMS komutunu verin:

```
/STOP REGION
```

2. SSM üyesini güncelleyin.

3. IMS komutunu verin:

```
/START REGION
```

z/OS **IMS bağlantısı kesiliyor**

IMS ya da kuyruk yöneticisi sonlandırıldığında bağlantı sona erdirilir. Diğer bir seçenek olarak, IMS ana uçbirim işletmeni bağlantıyı açık bir şekilde bozabilir.

IMS ile IBM MQarasındaki bağlantıyı sonlandırmak için aşağıdaki IMS komutunu kullanın:

```
/STOP SUBSYS sysid
```

Komut, genellikle ana uçbirim işletmeni (MTO) komutu veren uçbirime aşağıdaki iletiyi gönderir:

```
DFS058I STOP COMMAND IN PROGRESS
```

IMS komutu:

```
/START SUBSYS sysid
```

Bağlantıyı yeniden kurmak için gereklidir.

Not: The IMS command /STOP SUBSYS is not completed if an IMS trigger monitor is running.

z/OS **IMS tetikleme izleyicisini denetleme**

Durdurmak için CSQQTRMN işlemini kullanabilir ve IMS tetikleme izleyicisini başlatabilirsiniz.

IMS tetikleme izleme programı (CSQQTRMN hareketi), IMS tetikleme izleyicisini ayarlama içinde açıklanmıştır.

IMS tetikleme izleyicisini denetlemek için aşağıdaki başlara bakın:

- [CSQQTRMN başlatılıyor](#)
- [CSQQTRMN 'nin Durdurulması](#)

CSQQTRMN başlatılıyor

1. İzlemek istediğiniz her bir başlatma kuyruğu için CSQQTRMN programını çalıştıran, toplu iş odaklı BMP 'yi başlatın.
2. Toplu JCL ' nizi değiştirerek, aşağıdaki bilgileri içeren bir veri kümesine işaret eden CSQQUT1 DDname değerini 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 boşsa, CSQQDEFV ' de aday gösterilen varsayılan değer kabul edilir)
init_q_name	İzlenecek başlangıç kuyruğunun adı
lterm	Hata iletilerinin hedefi için IMS LTERM adı (bu boşsa, varsayılan değer MASTERDIR).
CONSOLEMESSAGES= YES	nominated'ye aday gösterilen IMS LTERM' ye gönderilen iletilerin da z/OS konsoluna gönderilmesi istenmektedir. Bu parametre atlanırsa ya da yanlış yazılmışsa, varsayılan değer konsola ileti göndermez.

3. Add a DD name of CSQQUT2 if you want a printed report of the processing of CSQQUT1 input.

Not:

1. The data set CSQQUT1 is defined with LRECL=80. Diğer DCB bilgileri veri kümesinden alınır. The DCB for data set CSQQUT2 is RECFM=VBA and LRECL=125.
2. Her kayıt için tek bir anahtar sözcük koyabilirsiniz. Anahtar sözcük değeri, anahtar sözcüğü izleyen ilk boşlukla sınırlanır; bu değer, açıklamaları içerebileceğiniz anlamına gelir. 1. kolonda yıldız işareti, tüm giriş kaydının bir açıklama olduğu anlamına gelir.
3. QMGRNAME ya da LTERM anahtar sözcüklerinden birini yanlış yazıyorsanız, CSQQTRMN bu anahtar sözcük için varsayılan değeri kullanır.
4. Tetikleme izleme programı BMP işini göndermeden önce altsistemin IMS (/START SUBSYS komutuyla) içinde başlatıldığından emin olun. Başlatılmamışsa, tetikleme izleme işiniz olağandışı bitiş kodu U3042 ile sonlandırılır.

CSQQTRMN durduruluyor

Başlatıldıktan sonra, CSQQTRMN, aşağıdaki olaylardan biri nedeniyle IBM MQ ile IMS arasındaki bağlantı kesilinceye kadar çalışır:

- Kuyruk yöneticisi sona erdiriliyor
- IMS Son

ya da bir z/OS STOP **jobname** komutu girilir.

z/OS IMS köprüsünü denetleme

IMS köprüsünü denetleyebileceğiniz IMS komutlarını anlamak için bu konuyu kullanın.

IBM MQ-IMS köprüsünü denetlemek için IBM MQ komutu yoktur. Ancak, IMS ' e teslim edilen iletileri aşağıdaki şekillerde durdurabilirsiniz:

- Paylaşılmayan kuyruklar için, tüm köprü kuyrukları için ALTER QLOCAL (xxx) GET (DISABLE) komutunu kullanın.
- Kümelenmiş kuyruklar için, SUSPEND QMGR CLUSTER (xxx) komutunu kullanın. Bu, yalnızca başka bir kuyruk yöneticisi kümelenmiş köprü kuyruğunu barındırırken etkili olur.
- Kümelenmiş kuyruklar için, SUSPEND QMGR FACILITY (IMSBRIDGE) komutunu kullanın. IMS'e başka ileti gönderilmez, ancak bekleyen tüm işlemlere ilişkin yanıtlar IMS' den alınır.

To start sending messages to IMS again, issue the RESUME QMGR FACILITY(IMSBRIDGE) command. Ayrıca, köprünün askıya alıp askıya alınmadığını görüntülemek için MQSC komutu DISPLAY SYSTEM de kullanabilirsiniz.

Bu komutlara ilişkin ayrıntılar için [MQSC komutları](#) başlıklı konuya bakın.

Ek bilgi için bkz:

- [“IMS köprüsünün başlatılması ve durdurulması” sayfa 517](#)
- [“Controlling IMS connections” sayfa 517](#)
- [Köprü kuyruklarının denetlenmesi](#)
- [“IMS köprüsünün yeniden eşzamanlanması” sayfa 518](#)
- [Tpipe adlarıyla çalışma](#)
- [IMS' dan iletilerin silinmesi](#)
- [Veri bağlantılarının silinmesi](#)
- [“IMS İşlem Süre Bitimi” sayfa 520](#)

IMS köprüsünün başlatılması ve durdurulması

OTMA ' yı başlatarak IBM MQ köprüsünü başlatın. IMS komutunu kullanın:

```
/START OTMA
```

ya da IMS sistem parametrelerinde OTMA=YES belirtilerek otomatik olarak başlatın. OTMA başlatılmışsa, 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 verildiğinde, bir IBM MQ olay iletisi üretilir.

Controlling IMS connections

IMS , IBM MQ ile bağlantıyı denetlemek ve izlemek için bu işletmen komutlarını sağlar:

/DEQUEUE TMEMBER *tmember* TPIPE *tpipe*

Bir Tpipe 'den iletileri kaldırır. Tüm iletileri kaldırmak için PURGE seçeneğini belirleyin ya da yalnızca ilk iletiyi kaldırmak için PURGE1 girin.

/GÖRÜNTÜLEME

OTMA sunucusuna ve istemcilerine ve istemci durumuna ilişkin özet bilgileri görüntüler.

/DISPLAY TMEMBER *ad*

Bir OTMA istemcisine ilişkin bilgileri görüntüler.

/DISPLAY TRACE TMEMBER *ad*

İzlenmekte olan bilgilerle ilgili bilgileri görüntüler.

/GÜVENLİ OTMA

Güvenlik seçeneklerini belirler.

/BAŞLAT OTMA

OTMA aracılığıyla iletişim sağlar.

/START TMEMBER *tuye* TPIPE *tpipe*

Tpipe adını başlatır.

/STOP OTMA

OTMA aracılığıyla iletişimi durdurur.

/STOP TMEMBER *tüye* TPIPE *tpipe*

Tpipe adını durdurur.

/İZLEME

IMS izlemesini denetler.

Bu komutlarla ilgili ek bilgi için, kullanmakta olduğunuz IMS düzeyi için *IMS/ESA Operator Reference* adlı elkitabına bakın.

IMS komut yanıtları, komutun yayınlandığı uçbirime gönderilir. Authorization to issue IMS commands is based on IMS security.

Köprü kuyruklarını denetleme

To stop communicating with the queue manager with XCF member name *üye* through the bridge, issue the following IMS command:

```
/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 ile iletişimi durdurmak için şu IMS komutunu verin:

```
/STOP TMEMBER tmember TPIPE tpipe
```

Her bir 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
```

Alternatively, you can alter the attributes of the IBM MQ queue to make it get inhibited.

IMS köprüsünün yeniden eşzamanlanması

The IMS bridge is automatically restarted whenever the queue manager, IMS, or OTMA are restarted.

IMS köprüsünün üstlendiği ilk görev, IMS ile yeniden eşzamanlanmasıdır. Bu, her uyumlulaştırılmış TPipe 'de IBM MQ ve IMS denetim sırası numaralarını içerir. A synchronized Tpipe is used when persistent messages are sent to IMS from an IBM MQ - IMS bridge queue using commit mode zero (commit-then-send).

Köprü IMS ile yeniden eşitlenemezse, IMS algılama kodu CSQ2023E iletilisinde döndürülür ve OTMA ile bağlantı durdurulur. If the bridge cannot resynchronize with an individual IMS Tpipe, the IMS sense code is returned in message CSQ2025E and the Tpipe is stopped. Bir Tpipe soğuk olarak başlatıldıysa, kurtarılabilir sıra numaraları otomatik olarak 1 'e ayarlanır.

Köprü bir Tpipe ile yeniden eşzamanlı olarak eşleştiginde eşleşmeyen sıra numaralarını keşfederse, CSQ2020E iletisi yayınlanır. IMS Tpipe ile yeniden eşzamanlamayı başlatmak için IBM MQ komutunu RESET TPIPE komutunu kullanın. XCF grubunu ve üye adını ve Tpipe adını sağlamanız gerekir; bu bilgiler ileti tarafından sağlanır.

Aşağıdakileri de belirtebilirsiniz:

- A new recoverable sequence number to be set in the Tpipe for messages sent by IBM MQ, and to be set as the partner's receive sequence number. Bunu belirtmezseniz, iş ortağının alma sıra numarası, geçerli IBM MQ gönderme sıra numarası olarak ayarlanır.
- A new recoverable sequence number to be set in the Tpipe for messages received by IBM MQ, and to be set as the partner's send sequence number. Bunu belirtmezseniz, iş ortağının gönderme sıra numarası, geçerli IBM MQ alma sıra numarası olarak ayarlanır.

Tpipe ile ilişkili çözülmemiş bir kurtarma birimi varsa, bu durum iletiyle de bilgilendirilir. Kurtarma biriminin kesinleştirilip kesinleştirilmeyeceğini belirtmek için IBM MQ komutunu RESET TPIPE komutunu kullanın, ya da geri alma işlemini geri almak için kullanın. Kurtarma birimini kesinleştirmeniz durumunda, ileti grubu önceden IMS' a gönderilmiştir ve köprü kuyruğundan silinir. Kurtarma birimini dışarıda yedeklediyseniz, iletiler daha sonra IMS' a gönderilmek üzere köprü kuyruğuna geri döndürülür.

Commit mode 1 (send-then-commit) Tpipes, uyumlulaştırılmaz.

Kesinleştirme kipi 1 işlemleri için dikkat edilecek noktalar

IMS' ta, kesinleştirme kipi 1 (CM1) hareketleri çıkış yanıtlarını eşitleme noktasından önce gönderir.

CM1 hareketi yanıt gönderemeyebilir; örneğin:

- Yanıtın gönderileceği Tpipe durdurulur
- OTMA durduruldu
- OTMA istemcisi (yani, kuyruk yöneticisi) ortadan kayboldu
- Yanıtın gönderileceği kuyruk ve kuyruk-iletiler kuyruğu kullanılamıyor

For these reasons, the IMS application sending the message pseudo-abends with code U0119. IMS işlemi ve programı bu durumda durdurulmaz.

These reasons often prevent messages being sent into IMS, as well as replies being delivered from IMS. Aşağıdaki durumlarda bir U0119 olağandışı sonu oluşabilir:

- İleti IMS' de olduğunda, Tpipe, OTMA ya da kuyruk yöneticisi durdurulur.
- IMS gelen iletiyle farklı bir Tpipe ile ilgili yanıt verir ve Tpipe durdurulur
- IMS farklı bir OTMA istemcisine yanıt verir ve bu istemci kullanılamaz.

Bir U0119 olağandışı sonu geldiğinde, IMS 'a gelen ileti ve yanıt iletileri IBM MQ ' e kaybedilir. If the output of a CMO transaction cannot be delivered for any of these reasons, it is queued on the Tpipe within IMS.

Veri bağlantısı adlarıyla çalışma

IBM MQ - IMS köprüsünü denetlemek için kullanılan komutların çoğu *tpipe* adını gerektirir. Bu konuyu kullanarak, tpipe adının ayrıntılarını nasıl bulabileceğinizi öğrenin.

IBM MQ - IMS köprüsünü denetleyen komutların birçoğu için *tpipe* adlarına gereksinim duyarsınız. Tpipe adlarını DISPLAY QUEUE komutundan alabilir ve aşağıdaki noktalara dikkat çekebilirsiniz:

- Veri bağlantısı adları, yerel bir kuyruk tanımlandığında atanır
- bir yerel kuyruğa iki tpipe adı verilir, biri senkronize olmak üzere, non-sync için bir
- tpipe names will not be known to IMS until after some communication between IMS and IBM MQ specific to that particular local queue takes place
- IBM MQ - IMS köprüsü tarafından kullanılacak bir tpipe için, ilişkili kuyruğu doğru XCF grubu ve üye adı alanlarına sahip olan bir Depolama Sınıfına atanmalıdır.

IMS' dan iletilerin silinmesi

A message that is destined for IBM MQ through the IMS bridge can be deleted if the Tmember/Tpipe is stopped. To delete one message for the queue manager with XCF member name *üye*, issue the following IMS command:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE1
```

Tpipe üzerindeki tüm iletileri silmek için aşağıdaki IMS komutunu çalıştırın:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE
```

Veri bağlantılarının silinmesi

IMS tborularını kendiniz silemezsiniz. Bunlar IMS tarafından aşağıdaki saatlerde silinir:

- IMS soğuk başlatıldığında, eşzamanlanır tpipes silinir.
- IMS yeniden başlatıldığında, uyumlulaştırılmamış veri bağlantıları silinir.

IMS İşlem Süre Bitimi

Süre bitim süresi bir işlemle ilişkilidir; herhangi bir IBM MQ iletiyle ilişkilendirilmiş bir süre bitim saati olabilir. The expiration interval is passed from the application, to IBM MQ, using the MQMD.Expiry field. Süre, bir iletinin süresi dolmadan önce, saniyenin onda biri olan bir değer olarak ifade edilir. Bir iletinin MQGET işlemini gerçekleştirme girişimi, süresi dolduğundan daha sonra, kuyruktan kaldırılmakta olan iletiyle ve gerçekleştirilen süre bitimi işlenirken sonuçlanıyor. Bir IBM MQ ağındaki kuyruk yöneticileri arasında ileti akışı olarak süre bitimi azalır. When an IMS message is passed across the IMS bridge to OTMA, the remaining message expiry time is passed to OTMA as a transaction expiration time.

Bir işlemin süre bitim süresi belirtilirse, OTMA giriş işlemlerini IMSiçinde üç farklı yerde sona erir:

- XCF ' den gelen giriş ileti
- giriş ileti kuyruğa alma süresi
- uygulama GU saati

GU zamanından sonra süre bitimi gerçekleştirilmez.

EXPTIME işlemi aşağıdaki işlemleri yaparak sağlanabilir:

- IMS hareket tanımlaması
- IMS OTMA ileti üstbilgisi
- IMS DFSINSXO kullanıcı çıkışı
- IMS CREATE ya da UPDATE TRAN komutları

IMS , 0243 ile bir hareketi olağandışı sonlandırarak bir işlemin süresinin dolduğunu ve bir ileti yayınlamak süresinin dolduğunu belirtir. Yayınlanan ileti, paylaşılmayan kuyruklar ortamında DFS555I ya da paylaşılan kuyruklar ortamında DFS2224I ' dir.

z/OS

Operating Advanced Message Security on z/OS

Advanced Message Security adres alanı, z/OS MODIFY komutunu kullanarak komutları kabul eder.

To enter commands for the Advanced Message Security (AMS) address space, use the z/OS MODIFY command.

Örneğin,

```
F qmgrAMSM, cmd
```

Burada *qmgr* , başlatılan görev adının önekiidir.

Çizelge 29 sayfa 521 , kabul edilen MODIFY komutlarını açıklar:

Çizelge 29. Advanced Message Security address space MODIFY komutları		
Komut	Seçenek	Tanım
GÖRÜNTÜLE		Sürüm bilgilerini görüntüle
YENİLE	ANAHTARLIK ilke TÜMÜ	Anahtarlık sertifikalarını, güvenlik ilkelerini ya da her ikisini yenileyin.
SSMFAUDIT	BAŞARILI ARIZA TÜMÜ	AMS iletileri başarılı bir şekilde koruduğunda ya da korumadığında, AMS iletileri ya da her ikisini de korumayı ya da korumayı başaramadığında, SMF denetiminin gerekli olup olmadığını belirleyin.
SMFTYPE	0 - 255	Set the SMF record type to be generated when AMS protects or unprotects messages. SMF denetimini geçersiz kılmak için kayıt tipi olarak 0 belirtin.

Not: Bir seçeneği belirtmek için, bir virgül ile ayrılmalıdır. Örneğin:

```
F qmgrAMSM, REFRESH KEYRING  
F qmgrAMSM, SSMFAUDIT ALL  
F qmgrAMSM, SMFTYPE 180
```

YENİLE komutu

REFRESH komutu, **REFRESH** komutu tamamlandıktan sonra MQOL komutu veren uygulamalar için geçerli kılınarak geçerli kılınan değişikliklerdir. Kuyruğu açık olan var olan uygulamalar, uygulamanın kuyruğu açıldığında seçenekleri kullanmaya devam eder. Yeni değerleri kullanmak için, uygulamanın kapatılması ve kuyruğun yeniden açılması gerekir.

AMSbaşlatılıyor ve durduruluyor

Advanced Message Security adres alanını başlatmak ya da durdurmak için bir komut girmenize gerek yoktur. The AMS address space is started automatically when the queue manager is started if AMS has been enabled with the **SPLCAP** parameter of CSQ6SYSP, and is stopped when the queue manager is stopped.

YönetmeIBM MQ Internet Pass-Thru

Bu bölümde, IBM MQ Internet Pass-Thru (MQIPT) olanağının nasıl yönetileceği açıklanmaktadır.

Configure MQIPT by making changes to the `mqipt.conf` configuration file as described in [IBM MQ Internet Pass-Thru yapılandırılıyor](#). To administer MQIPT, including refreshing MQIPT to bring configuration changes into effect without restarting MQIPT, use the **mqiptAdmin** command. MQIPT komutunu **mqiptAdmin** komutunu kullanarak yönetme hakkında bilgi için bkz. “Komut satırını kullanarak MQIPT uygulamasını yönetme” sayfa 524.

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 \(start MQIPT\)](#).

V9.2.0 **V9.2.0** Başlatılmakta olan MQIPT yönetim ortamına verilecek bir ad belirtmek için **mqipt** komutunu kullanabilirsiniz. MQIPT yönetim ortamının adı, komut kapısı kullanması gerekmeyen **mqiptAdmin** komutuyla MQIPT yerel yönetim ortamlarını denetlemek için kullanılır. Bu değıştirge 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 **mqiptService** komutunu kullanın.

- **Windows** Windows sistemlerinde **mqiPTService** komutu, MQIPT ürününü Windows hizmeti olarak kurar.
- **Linux** **AIX** AIX and Linux sistemlerinde **mqiPTService** 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 systemdgibi 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.

mqiPTService komutuyla ilgili daha fazla bilgi için bkz. [mqiPTService \(MQIPT hizmetini yönetme\)](#).

durdurmaMQIPT

mqiPTAdmin komutunu **-stop** parametresiyle kullanarak MQIPT ' i durdurabilirsiniz.

V 9.2.0 **V 9.2.0** Örneğin, aşağıdaki komut, yerel olarak **mqiPTAdmin** komutuyla aynı kullanıcı kimliği altında çalışan mqiPT1 adlı bir MQIPT yönetim ortamını durdurur:

```
mqiPTAdmin -stop -n ipt1
```

mqiPTAdmin komutu, aşağıdaki yöntemlerden birini kullanarak yönetmek için etkin MQIPT yönetim ortamına bağlanır:

- **V 9.2.0** **V 9.2.0** komut kapısını kullanmadan yerel bir MQIPT yönetim ortamına bağlanarak.
- bir komut kapısıyla ağ bağlantısı kurarak.

mqiPTAdmin komutunun bir komut kapısına komut gönderilerek MQIPT komutunu durdurmak için kullanılabilmesi için **RemoteShutDown** özelliği **true** olarak ayarlanarak uzaktan kapatma etkinleştirilmelidir.

MQIPT komutunu **mqiPTAdmin** komutunu kullanarak yönetme hakkında daha fazla bilgi için bkz. [“Komut satırını kullanarak MQIPT uygulamasını yönetme” sayfa 524.](#)

V 9.2.0 Parola şifreleme anahtarının belirtilmesi

IBM MQ 9.1.5' tan, MQIPT yapılandırması, varsayılan anahtar dışında bir şifreleme anahtarı kullanılarak şifrelenen parolalar içeriyorsa, parola şifreleme anahtarını MQIPT başlatıldığında okuyabileceği bir dosyada sağlamanız gerekir.

Parola şifreleme anahtarı dosyası

MQIPT tarafından saklanacak ve kullanılacak şifrelenen parolalar, sağladığınız bir şifreleme anahtarı kullanılarak şifrelenebilir. Bir şifreleme anahtarı sağlamadıysanız, varsayılan şifreleme anahtarı kullanılır. Parola şifreleme anahtarı belirtmenize gerek yoktur, ancak bunu yapmak daha güvenlidir. Kendi şifreleme anahtarınızı belirtmezseniz, varsayılan şifreleme anahtarı kullanılır.

Bir parola şifreleme anahtarı sağlıyorsa, bu anahtar, parolaları şifrelemek için kullanılan **mqiPTPW** komutu tarafından erişilebilen bir dosyada saklanmalıdır ve MQIPT. Dosyanın içeriğiyle ilgili tek sınırlama, en az bir karakter ve yalnızca bir satır metin içermeli.

Not: Yetkisiz kullanıcıların şifreleme anahtarını okumasını önlemek için, parola şifreleme anahtar dosyasında uygun dosya izinlerinin ayarlandığından emin olmanız gerekir. Yalnızca **mqiPTPW** komutunu çalıştıran ve MQIPT çalıştıran kullanıcının parola şifreleme anahtarını okumak için gerekli yetkiye sahip olması gerekir.

Aynı parola şifreleme anahtarı, bir MQIPTörneğine ilişkin tüm depolanan parolaları şifrelemek ve şifrelerini çözmek için kullanılır. Bu nedenle, her MQIPT kurulumu için yalnızca tek bir parola şifreleme anahtarı dosyasına gereksinim duyarsınız.

MQIPT kuruluşu için parola şifreleme anahtarı değiştirilirse, şifrelenen tüm parolaların yeni şifreleme anahtarı kullanılarak yeniden şifrelenmesi gerekir.

BaşlatılıyorMQIPT

Parola şifreleme anahtarı dosyasının varsayılan adı `MQIPT_HOME_DIR/mqipt_cred.key` dizinidir; burada `MQIPT_HOME_DIR`, `mqipt.conf` yapılandırma dosyasının saklandığı dizindir. MQIPT 'u otomatik olarak başlatılan bir hizmet olarak çalıştırmaya planlıyorsanız, parola şifreleme anahtarı dosyasını varsayılan adla oluşturmanız gerekir.

Parola şifreleme anahtarı dosyası varsayılan ad dışında bir adla yaratıldıysa, dosyanın adı, başlatıldığında MQIPT 'a sağlanmalıdır. Parola şifreleme anahtarı dosyasının adı, aşağıdaki yöntemlerden herhangi biri kullanılarak belirlenebilir:

1. the **-sf** parameter on the **mqipt** command used to start MQIPT.
2. `MQS_MQIPTCRED_KEYFILE` ortam değişkeni.
3. the `com.ibm.mq.ipc.cred.keyfile` Java property.

Parola şifreleme anahtarı dosyası sağlanmıyorsa, dosya varsa, varsayılan dosya adı kullanılır. Varsayılan parola şifreleme anahtarı dosyası yoksa, varsayılan parola şifreleme anahtarı kullanılır.

Komut satırını kullanarak MQIPT uygulamasını yönetme


MQIPTkomutunu yönetmek için komut satırındaki **mqiptAdmin** komutunu kullanabilirsiniz.

Aşağıdaki denetim işlevlerini gerçekleştirmek için **mqiptAdmin** komutunu kullanabilirsiniz:

- List active local instances of MQIPT.
- Yapılandırma dosyasında değişiklik yaptıktan sonra, MQIPT örneğini yenileyin.
- MQIPTyönetim ortamını durdurun.

`mqiptAdmin` komutu, MQIPT kuruluş dizininin bin alt dizininde bulunur.

The **mqiptAdmin** command connects to the active instance of MQIPT to administer using one of the following methods:

- bir komut kapısına ağ bağlantısı yaparak.
-  MQIPT yerel yönetim ortamına komut kapısı kullanılmadan bağlanarak.

The **mqiptAdmin** command is compatible with previous versions of MQIPT, but you cannot use the command to administer versions of MQIPT that are a higher version than the version of the **mqiptAdmin** command. In an environment that includes different versions of MQIPT, you must use the latest version of the **mqiptAdmin** command.

mqiptAdmin komutunun sözdizimiyle ilgili daha fazla bilgi için bkz. [mqiptAdmin \(MQIPT' u yönetin\)](#).

Komut kapısı olmayan yerel yönetim



IBM MQ 9.2.0' tan, MQIPT yerel yönetim ortamları bir komut kapısı kullanılmadan yönetilebilir. Local administration allows you to administer MQIPT by using the **mqiptAdmin** command only when it is run on the same system as the MQIPT instance that you want to administer.

mqiptAdmin için, komut kapısı kullanılmadan yerel bir MQIPT yönetim ortamını yönetmeye yetkili olması için, MQIPT yönetim ortamının aynı sistemde ve **mqiptAdmin** ile aynı kullanıcı kimliği altında çalışıyor olması gerekir. Diğer bir seçenek olarak, AIX and Linuxüzerinde **mqiptAdmin**, `root` olarak çalıştırılabilir.

Yerel yönetim varsayılan olarak etkindir. Yerel denetimi devre dışı bırakmak için **LocalAdmin** yapılandırma özelliğini kullanın. **LocalAdmin** özelliği hakkında daha fazla bilgi için bkz. [LocalAdmin](#).

MQIPT' un yerel yönetim ortamlarını denetlemek için, her yönetim ortamı için bir ad vermelisiniz. You can assign a name to an instance of MQIPT by using the **-n** parameter when starting MQIPT with the **mqipt**

command. MQIPT' u başlatırken bir ad belirtmezseniz, ana dizinin adı MQIPT yönetim ortamının adı olarak kullanılır. Örneğin, aşağıdaki komut MQIPT başlatılır ve yönetim ortamına ipt1 adını atar:

```
mcipt /opt/mcipt1 -n ipt1
```

Once the instance has a name, you can administer that instance by specifying the name in the **mciptAdmin** command with the **-n** parameter. Örneğin, aşağıdaki komut, ipt1adlı MQIPT yerel yönetim ortamını durdurur:

```
mciptAdmin -stop -n ipt1
```

You can list all local active instance of MQIPT that the **mciptAdmin** command is authorized to administer without using a command port by using the **mciptAdmin** command with the **-list** parameter. For example, the following command lists all local active instances of MQIPT that the user that started the **mciptAdmin** command is authorized to administer:

```
mciptAdmin -list
```

Komut kapısı kullanan yönetim



IBM MQ 9.2.0olanağından, güvenli olmayan bir komut kapısı ve TLS ile güvenli kılınan bir komut kapısı olan MQIPT ' u yapılandırabilirsiniz. Yönetmek istediğiniz MQIPT yönetim ortasıyla aynı sistemde bulunan herhangi bir kullanıcı olarak MQIPT ' u yönetmek için bu komut kapılarını kullanarak ya da uzak bir sistemden kullanabilirsiniz.

MQIPT ' in önceki sürümleri, yalnızca güvenli olmayan komut kapısına verilen yönetim komutlarını kabul etti.

Not: Güvenli olmayan komut kapısı bağlantıları şifrelenmemiş olduğundan, ağ üzerinden MQIPT erişim parolası da dahil olmak üzere güvenli olmayan komut kapısına gönderilen veriler, ağ üzerindeki diğer kullanıcılar tarafından görülebilir.

In order for MQIPT to listen on a command port for commands issued by the **mciptAdmin** command, a value must specified for either the **CommandPort** or **SSLCommandPort** properties in the global section of the `mcipt.conf` configuration file.

MQIPT komut kapılarından birini etkinleştirmeden önce Diğer güvenlik konuları başlıklı konulardaki güvenlik noktalarına ilişkin bilgileri gözden geçirin. Komut kapıları tarafından alınan komutlar için kimlik doğrulamayı etkinleştirmeyi düşünün. Komut kapısı kimlik doğrulamasıyla ilgili daha fazla bilgi için bkz. “Komut kapısı kimlik doğrulaması” sayfa 529.

To administer an instance of MQIPT using a command port, specify the network address of the host where MQIPT is running, and the command port number, as parameters to the **mciptAdmin** command. Örneğin, `mcipt.example.com` üzerinde çalışan MQIPT yönetim ortamını yenilemek için ve 1890 kapısını dinlemek üzere yapılandırılmış güvenli olmayan komut kapısına sahip olmak için aşağıdaki komutu verin:

```
mciptAdmin -refresh -r mcipt.example.com:1890
```

Ana makine adı ve kapı numarasını belirtmezseniz, **mciptAdmin** , localhostkapısına, 1881 kapısına bağlanmayı dener.

TLS komut kapısını kullanarak MQIPT ' i yönetme hakkında daha fazla bilgi için bkz. “TLS komut kapısı kullanılarak MQIPT ' in yönetilmesi” sayfa 525.

IBM MQ 9.2.0 TLS komut kapısı kullanılarak MQIPT ' in yönetilmesi

From IBM MQ 9.2.0, MQIPT can be configured to use a TLS command port to listen for administrative commands issued by the **mciptAdmin** command. Using the TLS command port protects sensitive data such as the MQIPT access password on the network between **mciptAdmin** and MQIPT. TLS komut kapısını yapılandırmak ve TLS komut kapısını kullanarak MQIPT ' yi yönetmek için bu yordamı kullanın.

Bu görev hakkında

TLS komut kapısı, PKCS #12 anahtar halkasında ya da PKCS #11 Cryptographic Token Interface 'i destekleyen şifreleme donanımında depolanan bir sunucu sertifikasıyla yapılandırılmalıdır. Komut kapısı sunucusu sertifikası, TLS anlaşması sırasında **mqiptAdmin** komutuna gönderilir. Bu görev, güvenilir bir Sertifika Yetkilisi 'nden (CA) yeni bir sunucu sertifikası istemenizi ve sertifikanın bir dosyada size geri gönderildiğini varsayar. **mqiptAdmin** komutu, sunucu sertifikasına imza atan CA ' nın CA sertifikasını kullanarak komut kapısı sertifikasını doğrular. CA sertifikası, **mqiptAdmin** komutu tarafından erişilebilen bir PKCS #12 anahtar halkası içinde 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 529.

This procedure describes how to manage the key rings and digital certificates that are required to use the TLS command port by using the **mqiptKeycmd** (iKeyman) command-line interface (CLI). CLI ' yı **mqiptKeycmd** komutunu kullanarak kullanabilirsiniz. Anahtar halkaları ve dijital sertifikalarını yönetmek için kullanılabilecek diğer komutlarla ilgili daha fazla bilgi için bkz. [mqiptKeyman](#) ve [mqiptKeycmd](#).

Yordam

1. Follow these steps to configure the TLS command port for the instance of MQIPT.

- a) TLS komut kapısı tarafından kullanılacak bir PKCS #12 anahtar halkası dosyası oluşturun. Anahtar halkası TLS komut kapısı sunucusu 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 dosyasının adıdır ve *password* , anahtarlık parolasıdır.

- b) CA tarafından imzalanmış TLS komut kapısı sunucusu sertifikası için bir sertifika isteği oluşturun.

To create a certificate request using the iKeyman CLI, enter the following command:

```
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 büyüklüğünü belirler.

-sig_alg algoritma

Girişin anahtar çiftinin oluşturulması için kullanılan asimetrik imza algoritmasını belirtir.

-dn ayırt edici ayırt edici ad

Çift tırnak işareti içine alınmış X.500 ayırt edici adını belirtir.

-file certreq_filename

Sertifika isteğine ilişkin dosya adını belirler.

- c) [“1.b” sayfa 526](#) adımıyla oluşturulan sertifika isteği dosyasını, imzalanacak sertifika kuruluşunuza gönderin.

- d) CA, imzalanmış sertifikayı size gönderdikten sonra, imzalanmış sertifikayı anahtarlık dosyasına alın. İmzalanmış sertifikayı CLI kullanarak anahtarlık imzasına almak için şu komutu girin:

```
mciptKeycmd -cert -receive -file cert_filename -db filename
-pw password -type pkcs12
```

Burada *cert_filename* , sertifikayı içeren dosyanın adı, *kütükadı* anahtar halkası dosyasının adıdır ve *password* , anahtarlık parolasıdır.

- e) Encrypt the key ring password using the **mciptPW** command.

Aşağıdaki komutu girin:

```
mciptPW -sf encryption_key_file
```

Burada *encryption_key_file* , MQIPT kurulumunuz için parola şifreleme anahtarını içeren bir dosyanın adıdır. MQIPT kurulumunuz varsayılan parola şifreleme anahtarını kullanıyorsa, **-sf** parametresini belirtmeniz gerekmez. İstendiğinde şifrelemek için anahtarlık parolasını yazın.

mciptPW komutuna ilişkin ek bilgi için [Anahtar halkası parolasının şifrenmesibaşlıklı konuya](#) bakın.

- f) Edit the *mcipt.conf* configuration file and specify the following properties to configure the TLS command port:

i) Set the value of the **SSLCommandPort** property to the TLS command port number.

ii) **SSLCommandPortKeyRing** özelliğinin değerini, [“1.a” sayfa 526](#)adımında yaratılan anahtar halkasının dosya adına ayarlayın.

iii) Set the value of the **SSLCommandPortKeyRingPW** to the string output by the **mciptPW** command in step [“1.e” sayfa 527](#).

iv) Set the value of the **SSLCommandPortSiteLabel** property to the label name of the TLS command port certificate, specified when creating the certificate request in step [“1.b” sayfa 526](#).

v) If you want to restrict inbound connections to the TLS command port to those from a particular network interface, set the value of the **SSLCommandPortListenerAddress** property to a network address belonging to one of the network interfaces on the system where MQIPT is running. Örneğin, gelen bağlantıları TLS komut kapısına yalnızca yerel makineden sınırlamak için, **SSLCommandPortListenerAddress** özelliğinin değerini localhostolarak ayarlayın.

- g) Start or refresh MQIPT to enable the TLS command port.

MQIPT issues console messages such as the following to display the TLS command port configuration that is in effect:

```
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. MQIPTkomutunu yönetmek için **mciptAdmin** komutunun kullanıldığı sistemde, **mciptAdmin** ' in TLS komut kapısına bağlanmasını sağlamak için bu adımları izleyin.

- a) **mciptAdmin** komutu tarafından güvenilirlik deposu olarak kullanılmak üzere bir PKCS #12 anahtar halkası oluşturun.

CLI kullanarak bir anahtarlık dosyası oluşturmak için aşağıdaki komutu girin:

```
mciptKeycmd -keydb -create -db filename -pw password -type pkcs12
```

Burada *kütükadı* , yaratılacak anahtarlık dosyasının adıdır ve *password* , anahtarlık parolasıdır.

- b) TLS komut kapısı sertifikasını, [“2.a” sayfa 527](#)adımında oluşturulan anahtar halkasına imzalayan CA sertifikasını içe aktarın.

CA sertifikasını iKeyman CLI ' yı 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.

certlabel

CA sertifikasına verilecek etiketi belirtir.

cert_dosyaadı

CA sertifikasını içeren dosyanın adını belirtir.

- c) Encrypt the key ring password using the **mqiPTPW** command.

Aşağıdaki komutu girin:

```
mqiPTPW -sf encryption_key_file
```

Burada *encryption_key_file* , parola şifreleme anahtarını içeren dosyanın adıdır. The password encryption key file can be different to the one used to encrypt passwords in the MQIPT configuration. The default password encryption key is used if you do not specify an encryption key file with the **-sf** parameter. İstendiğinde şifrelemek için anahtarlık parolasını yazın.

mqiPTPW komutuna ilişkin ek bilgi için [Anahtar halkası parolasının şifrelenmesibaşlıklı konuya](#) bakın.

- d) **mqiPTAdmin** komutu tarafından kullanılacak bir özellikler dosyası oluşturun 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 527adımında yaratılan anahtarlık kütüğünün adıdır.

anahtar_ring_parolası

is the encrypted password output by the **mqiPTPW** command in step “2.c” sayfa 528.

şifreleme_anahtar_dosyası

Parola şifreleme anahtarını içeren dosyanın adı. You need to specify the

PasswordProtectionKeyFile property only if an encryption key file was used to encrypt the key ring password in step “2.c” sayfa 528.

- 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 528.

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 doğrulamak için aşağıdaki gibi bir ileti yayınlar:

```
MQCAI109 The connection to MQIPT is secured with TLSv1.2.
```

Sonraki adım

TLS komut kapısı tarafından alınan komutlara ilişkin kimlik doğrulamasını etkinleştirmek için, “[Komut kapısı kimlik doğrulaması](#)” sayfa 529içindeki adımları izleyin.

MQIPT , güvenli olmayan komut kapısı ve bir parola kullanarak TLS komut kapısı tarafından alınan komutların kimliğini doğrulamak üzere yapılandırılabilir. Komut kapısı kimlik doğrulamasını etkinleştirmek için bu yordamı kullanın.

Bu görev hakkında

mciptAdmin komutu, komut kapısı kimlik doğrulaması etkinleştirilmiş bir MQIPT yönetim ortamının komut kapısına bağlandığında, kullanıcıların parola girmeleri için bilgi isteminde bulunur. MQIPT , **mciptAdmin** komutunda girilen parolayı MQIPT yapılandırılmasında belirtilen erişim parolasına göre doğrular.

Komut kapısı kimlik doğrulaması için ayarladığınız özellikler hem TLS komut kapısı hem de güvenli olmayan komut kapısı için geçerlidir.

Yordam

1. **mciptPW** komutunu kullanarak MQIPT erişim parolasını şifreleyin.

Aşağıdaki komutu girin:

```
mciptPW -sf encryption_key_file
```

Burada *encryption_key_file* , MQIPT kurulumunuz için parola şifreleme anahtarını içeren dosyanın adıdır. MQIPT kurulumunuz varsayılan parola şifreleme anahtarını kullanıyorsa, **-sf** parametresini belirtmeniz gerekmez. İstendiğinde şifrelemek için erişim parolasını yazın.

For more information about encrypting passwords in the MQIPT configuration, see [Depolanan parolaların şifrelenmesi](#).

2. **mcipt.conf** yapılandırma dosyasını düzenleyin ve aşağıdaki özellikleri belirtin:

```
AccessPW=encrypted_password  
RemoteCommandAuthentication=auth_setting
```

Burada:

encrypted_password

is the encrypted password output by the **mciptPW** command in step “1” sayfa 529.

yetki_ayarı

kimlik doğrulama gereksinmesi. Bu özellik aşağıdaki değerlerden birine ayarlanmışsa, komut kapısı kimlik doğrulaması etkinleştirilir:

isteğe bağlı

Parola gerekli değildir, ancak bir parola sağlanırsa, parola geçerli olmalıdır. Bu seçenek, geçiş sırasında yararlı olabilir. Örneğin,

zorunlu

Komut kapısı tarafından alınan her bir komutla geçerli bir parola sağlanmalıdır.

Bu özelliklerle ilgili daha fazla bilgi için [MQIPT genel özellikler](#) başlıklı konuya bakın.

3. Start or refresh MQIPT to bring the changes into effect.

MQIPT , komut kapısı kimlik doğrulamasının etkinleştirilip etkinleştirilmediğini belirten bir ileti yayınlar. Örneğin, MQIPT komutu, **mciptAdmin** komutu her çalıştırıldığında geçerli bir parola girilmesini gerektirecek şekilde yapılandırıldıysa, aşağıdaki ileti yayınlanır:

```
MQCPI021 Password checking has been enabled on the command port
```

Yedeklemeler yapma

Düzenli yedekleme yordamlarınızın bir parçası olarak yedeklemeniz gereken bir dizi MQIPT dosyası vardır.

Aşağıdaki dosyaları düzenli aralıklarla yedekle:

- Yapılandırma dosyası, mqipt.conf
- mqipt.conf içinde aşağıdaki özellikler tarafından belirtilen SSL/TLS anahtar halkası dosyaları:
 - **SSLClientKeyRing**
 - **SSLClientCAKeyRing**
 - **SSLServerKeyRing**
 - **SSLServerCAKeyRing**
 - **V9.2.0 V9.2.0 SSLCommandPortKeyRing**
- mqipt.conf içinde aşağıdaki özellikler tarafından belirtilen SSL/TLS anahtar halkası parola dosyaları:
 - **SSLClientKeyRingPW**
 - **SSLClientCAKeyRingPW**
 - **SSLServerKeyRingPW**
 - **SSLServerCAKeyRingPW**
- **V9.2.0** MQIPT yapılandırması, varsayılan anahtar dışında bir şifreleme anahtarıyla şifrelenen parolalar içeriyorsa, parola şifreleme anahtarı dosyası.
- Bu özellik ayarlandıysa, **SecurityManagerPolicy** tarafından belirtilen ilke dosyası.
- The security exit files and certificate exit files specified by the following properties in mqipt.conf:
 - **SecurityExitName**
 - **SSLExitName**
- Denetleme amacıyla gerekliyse, MQIPT ana dizininin log alt dizinindeki bağlantı günlüğü dosyaları.

Performans ayarlaması

Her bir MQIPT rotasının göreceli performansını, bir iş parçacığı havuzunun bir birleşimini ve boşta durma zamanaşımı belirtimini kullanarak ayarlayabilirsiniz.

Bağlantı iş parçacıkları

Her MQIPT rotasına, gelen iletişim isteklerini işleyen, koşut zamanlı olarak çalışan iş parçacıkları için bir çalışma havuzu atanır. Kullanıma hazırlama sırasında, iş parçacıklarının havuzu yaratılır (rotanın MinConnectionThreads özneliğine belirtilen büyüklükte) ve ilk gelen isteği işlemek için bir iş parçacığı atanır. Bu istek geldiğinde, başka bir iş parçacığı atanır, sonraki gelen isteğe hazır olur. İş için tüm iş parçacıkları atandığında, yeni bir iş parçacığı oluşturulur, çalışma havuzuna eklenir ve iş için atanır.

Bu şekilde, havuz, iş parçacığı sayısı üst sınırına (**MaxConnectionThreads** içinde belirtilir) ulaşıncaya kadar büyür. İş parçacıkları, bir etkileşim sona erdiğinde ya da belirtilen boşta durma zamanaşımı süresi dolduğunda havuza geri bırakılır. Çalışan iş parçacığı sayısı üst sınırına ulaşıldığında, bir sonraki gelen istek bir iş parçacığının çalışma havuzuna geri verilinceye kadar bekler.

Kullanılabilir iş parçacıklarının sayısını artırarak isteklerin beklemesi gereken süreyi azaltabilirsiniz. Ancak, bu artış, kullanılabilir sistem kaynaklarıyla dengelemeniz gerekir.

Boşta durma zamanaşımı

Etkinlik dışı durum nedeniyle, varsayılan olarak çalışan iş parçacıkları sonlandırılmaz. Bir sohbete bir iş parçacığı atandığında, olağan bir şekilde kapatılıncaya kadar bu etkileşim için atanmış olmaya devam eder. Rota devre dışı bırakılır ya da MQIPT kapatılır. İsteğe bağlı olarak, **IdleTimeout** özelliğinde boşta durma zamanaşımı aralığı (dakika cinsinden) belirtebilirsiniz; böylece, belirtilen süre içinde etkin olmayan iş parçacıkları yeniden çevrime sokulmalıdır. İş parçacıkları, bunları çalışma havuzuna yerleştirerek kullanılmak üzere geri dönüştürülmüş olur.

IBM MQ etkinliđi ara sıra oluşuyorsa, sağlıklı işletim bildirim aralıđını MQIPT zamanaşımından daha az bir değere ayarlayın; böylece iş parçacıkları sürekli olarak geri dönüştürülmez.

Özel notlar

Bu belge, ABD'de kullanıma sunulan ürünler ve hizmetler için hazırlanmıştır.

IBM, bu belgede sözü edilen ürün, hizmet ya da özellikleri diğer ülkelerde kullanıma sunmayabilir. Bulduğunuz yerde kullanıma sunulan ürün ve hizmetleri yerel IBM müşteri temsilcisinden ya da çözüm ortağınızdan öğrenebilirsiniz. Bir IBM ürün, program ya da hizmetine gönderme yapılması, açık ya da örtük olarak yalnızca o IBM ürünü, programı ya da hizmetinin kullanılabilirliğini göstermez. Aynı işlevi gören ve IBM'in fikri mülkiyet haklarına zarar vermeyen herhangi bir ürün, program ya da hizmet de kullanılabilir. Ancak, IBM dışı ürün, program ya da hizmetlerle gerçekleştirilen işlemlerin değerlendirilmesi ve doğrulanması kullanıcının sorumluluğundadır.

IBM'in, bu belgedeki konularla ilgili patentleri ya da patent başvuruları olabilir. Bu belgenin size verilmiş olması, patentlerin izinsiz kullanım hakkının da verildiği anlamına gelmez. Lisansla ilgili sorularınızı aşağıdaki adrese yazabilirsiniz:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Çift byte (DBCS) bilgilerle ilgili lisans soruları için, ülkenizdeki IBM'in Fikri Haklar (Intellectual Property) bölümüyle bağlantı kurun ya da sorularınızı aşağıda adrese yazın:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japonya

Aşağıdaki paragraf, İngiltere ya da bu tür hükümlerin yerel yasalarla uyuşmadığı diğer ülkelerde geçerli değildir: INTERNATIONAL BUSINESS MACHINES CORPORATION BU YAYINI, HAK İHLALİ YAPILMAYACAĞINA DAİR GARANTİLERLE TİCARİLİK VEYA BELİRLİ BİR AMACA UYGUNLUK İÇİN ZİMNİ GARANTİLER DE DAHİL OLMAK VE FAKS BUNLARLA SINIRLI OLMAMAK ÜZERE AÇIK YA DA ZİMNİ HİÇBİR GARANTİ VERMEKSİZİN "OLDUĞU GİBİ" ESASIYLA SAĞLAMAKTADIR. Bazı ülkeler bazı işlemlerde garantinin açık ya da örtük olarak reddedilmesine izin vermez; dolayısıyla, bu bildirim sizin için geçerli olmayabilir.

Bu yayın teknik yanlışlar ya da yazım hataları içerebilir. Buradaki bilgiler üzerinde düzenli olarak değişiklik yapılmaktadır; söz konusu değişiklikler sonraki basımlara yansıtılacaktır. IBM, önceden bildirimde bulunmaksızın, bu yayında açıklanan ürünler ve/ya da programlar üzerinde iyileştirmeler ve/ya da değişiklikler yapabilir.

Bu belgede IBM dışı Web sitelerine yapılan göndermeler kullanıcıya kolaylık sağlamak içindir ve bu Web sitelerinin onaylanması anlamına gelmez. Bu Web sitelerinin içerdiği malzeme, bu IBM ürününe ilişkin malzemenin bir parçası değildir ve bu tür Web sitelerinin kullanılmasının sorumluluğu size aittir.

IBM'e bilgi ilettiğinizde, IBM bu bilgileri size karşı hiçbir yükümlülük almaksızın uygun gördüğü yöntemlerle kullanabilir ya da dağıtabilir.

(i) Bağımsız olarak yaratılan programlarla, bu program da içinde olmak üzere diğer programlar arasında bilgi değiş tokuşuna ve (ii) değiş tokuş edilen bilginin karşılıklı kullanımına olanak sağlamak amacıyla bu program hakkında bilgi sahibi olmak isteyen lisans sahipleri şu adrese yazabilirler:

IBM Corporation
Yazılım Birlikte Çalışabilirlik Koordinatörü, Bölüm 49XA
3605 Highway 52 N

Rochester, MN 55901
U.S.A.

Bu tür bilgiler, ilgili kayıt ve koşullar altında ve bazı durumlarda bedelli olarak edinilebilir.

Bu belgede açıklanan lisanslı program ve bu programla birlikte kullanılacak tüm lisanslı malzeme, IBM tarafından, IBM Müşteri Sözleşmesi, IBM Uluslararası Program Lisansı Sözleşmesi ya da eşdeğer herhangi bir sözleşmenin kayıt ve koşulları altında sağlanır.

Burada belirtilen performans verileri denetimli bir ortamda elde edilmiştir. Bu nedenle, başka işletim ortamlarında çok farklı sonuçlar alınabilir. Bazı ölçümler geliştirilme düzeyindeki sistemlerde yapılmıştır ve bu ölçümlerin genel kullanıma sunulan sistemlerde de aynı olacağı garanti edilemez. Ayrıca, bazı sonuçlar öngörü yöntemiyle elde edilmiş olabilir. Dolayısıyla, gerçek sonuçlar farklı olabilir. Bu belgenin kullanıcıları, kendi ortamları için geçerli verileri kendileri doğrulamalıdır.

IBM dışı ürünlerle ilgili bilgiler, bu ürünleri sağlayan firmalardan, bu firmaların yayın ve belgelerinden ve genel kullanıma açık diğer kaynaklardan alınmıştır. IBM bu ürünleri sınınamamıştır ve IBM dışı ürünlerle ilgili performans doğruluğu, uyumluluk gibi iddiaları doğrulayamaz. IBM dışı ürünlerin yeteneklerine ilişkin sorular, bu ürünleri sağlayan firmalara yöneltilmelidir.

IBM'in gelecekteki yönelim ve kararlarına ilişkin tüm bildirimler değişebilir ve herhangi bir duyuruda bulunulmadan bunlardan vazgeçilebilir; bu yönelim ve kararlar yalnızca amaç ve hedefleri gösterir.

Bu belge, günlük iş ortamında kullanılan veri ve raporlara ilişkin örnekler içerir. Örneklerin olabildiğince açıklayıcı olması amacıyla kişi, şirket, marka ve ürün adları belirtilmiş olabilir. Bu adların tümü gerçek dışıdır ve gerçek iş ortamında kullanılan ad ve adreslerle olabilecek herhangi bir benzerlik tümüyle rastlantıdır.

YAYIN HAKKI LİSANSI:

Bu belge, çeşitli işletim platformlarında programlama tekniklerini gösteren, kaynak dilde yazılmış örnek uygulama programları içerir. Bu örnek programları, IBM'e herhangi bir ödemede bulunmadan, örnek programların yazıldığı işletim altyapısına ilişkin uygulama programlama arabirimiyle uyumlu uygulama programlarının geliştirilmesi, kullanılması, pazarlanması ya da dağıtılması amacıyla herhangi bir biçimde kopyalayabilir, değiştirebilir ve dağıtabilirsiniz. Bu örnekler her koşul altında tüm ayrıntılarıyla sınınamamıştır. Dolayısıyla, IBM bu programların güvenilirliği, bakım yapılabilirliği ya da işlevleri konusunda açık ya da örtük güvence veremez.

Bu bilgileri elektronik kopya olarak görüntülediyseniz, fotoğraflar ve renkli resimler görünmeyebilir.

Programlama arabirimi bilgileri

Programlama arabirimi bilgileri (sağlandıysa), bu programla birlikte kullanılmak üzere uygulama yazılımları yaratmanıza yardımcı olmak üzere hazırlanmıştır.

Bu kitap, müşterinin WebSphere MQ hizmetlerini edinmek üzere program yazmasına olanak tanıyan, amaçlanan programlama arabirimlerine ilişkin bilgiler içerir.

Ancak, bu bilgiler tanılama, değiştirme ve ayarlama bilgilerini de içerebilir. Tanılama, değiştirme ve ayarlama bilgileri, uygulama yazılımlarınızda hata ayıklamanıza yardımcı olur.

Önemli: Bu tanılama, değiştirme ve ayarlama bilgilerini bir programlama arabirimi olarak kullanmayın; bu, değişiklik söz konusu olduğunda kullanılır.

Ticari Markalar

IBM, IBM logosu, ibm.com, IBM Corporation 'ın dünya çapında birçok farklı hukuk düzeninde kayıtlı bulunan ticari markalarıdır. IBM ticari markalarının güncel bir listesini Web üzerinde "Telif hakkı ve ticari marka bilgileri" www.ibm.com/legal/copytrade.shtml adresinde bulabilirsiniz. Diğer ürün ve hizmet adları IBM'in veya diğer şirketlerin ticari markaları olabilir.

Microsoft ve Windows, Microsoft Corporation'ın ABD ve/veya diğer ülkelerdeki ticari markalarıdır.

UNIX, The Open Group şirketinin ABD ve diğer ülkelerdeki tescilli ticari markasıdır.

Linux, Linus Torvalds'ın ABD ve/ya da diđer ÷lkelerdeki tescilli ticari markasıdır.

Bu ÷r÷n, Eclipse Project (<https://www.eclipse.org/>) tarafından geliřtirilen yazılımları ierir.

Java ve Java tabanlı t÷m markalar ve logolar, Oracle firmasının ve/ya da iřtiraklerinin markaları ya da tescilli markalarıdır.



Parça numarası:

(1P) P/N: