

9.0

IBM Molanađının ynetilmesi

IBM

Not

Bu bilgileri ve desteklediđi ürünü kullanmadan önce, [“Özel notlar” sayfa 463](#) bölümündeki bilgileri okuyun.

Bu basım, yeni basımlarında tersi belirtilmediđi sürece, IBM® MQ sürüm 9 yayın düzeyi 0 ve sonraki tüm yayın düzeyleri ve deđişiklikler için geçerlidir.

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

© **Copyright International Business Machines Corporation 2015, 2023.**

İçindekiler

Yönetme.....	5
Yerel ve Uzak Denetim.....	8
Denetim komutlarını kullanarak IBM MQ uygulamasını yönetme.....	8
MQSC komutlarını kullanarak MQ uygulamasını yönetme.....	10
Komut dosyası (MQSC) komutları.....	10
MQSC komutlarındakiIBM MQ nesne adları.....	12
Standart giriş ve çıkış.....	13
MQSC komutlarının etkileşimli olarak kullanılması.....	13
MQSC komutlarını metin dosyalarından çalıştırma.....	15
Toplu iş dosyalarından MQSC komutları çalıştırılıyor.....	18
MQSC komutlarıyla ilgili sorunların çözülmesi.....	18
PCF komutlarını kullanarak IBM MQ yönetimini otomatikleştirme.....	19
IBM MQ Programlanabilir Komut Biçimlerine Giriş.....	20
PCF 'lerin kullanımını kolaylaştırmak için MQAI' nin kullanılması.....	32
REST APIkomutunu kullanarak yönetim.....	68
administrative REST APIile çalışmaya başlama.....	68
administrative REST APIkomutunu kullanma.....	74
REST APIkomutunu kullanarak uzaktan yönetim.....	80
REST API zaman damgaları.....	84
REST API Hata işleme.....	84
REST API Keşif.....	86
REST API ulusal dil desteği.....	88
IBM MQ Consolekomutunu kullanarak yönetim.....	89
IBM MQ Consoleile çalışmaya başlama.....	90
Yerel kuyruk yöneticileriyle çalışılması.....	93
IBM MQ nesneleriyle çalışma.....	95
Yetki kayıtlarıyla çalışma.....	110
Sistem kaynağı kullanımını izleme.....	112
Gösterge panosu düzenlerinin yapılandırılması.....	123
Gösterge panosu denetimleri.....	124
Klavye kısayolları.....	125
IBM MQ Explorerkomutunu kullanarak yönetim.....	125
IBM MQ Explorerile yapabildiğiniz.....	126
IBM MQ Explorer' in ayarlanması.....	127
IBM MQ Taskbar uygulamasını kullanma (yalnızcaWindows).....	133
IBM MQ uyarı izleyicisi uygulaması (yalnızca Windows).....	133
Yerel IBM MQ nesnelerini yönetme.....	133
Kuyruk yöneticisinin başlatılması ve durdurulması.....	134
MQI kanalları durduruluyor.....	137
Kuyruk yöneticileriyle çalışılması.....	138
Yerel kuyruklarla çalışma.....	140
Diğer ad kuyruklarıyla çalışma.....	145
Ölü-harfli kuyruklarla çalışma.....	147
Model kuyruklarıyla çalışma.....	165
Yönetimle ilgili konularla çalışma.....	166
Aboneliklerle çalışma.....	168
Hizmetlerle çalışma.....	172
Tetikleme için nesnelere yönetme.....	179
İki sistem arasında dmpmqmsg yardımcı programını kullanma.....	181
Uzak IBM MQ nesnelerini yönetme.....	184
Kanallar ve uzak kuyruğa alma.....	185
Yerel kuyruk yöneticisinden uzak denetim.....	186

Uzak kuyruğun yerel tanımlaması yaratılması.....	192
Dağıtılmış ağlara ilişkin zamanuysuz komutların sona erdiği denetleniyor.....	194
Uzak kuyruk tanımlamalarının diğer adlar olarak kullanılması.....	197
Veri dönüştürme.....	197
YönetmeMQ Telemetry.....	202
Linux ve AIXüzerinde telemetri için kuyruk yöneticisi yapılandırılması.....	202
Windowsüzerinde telemetri kuyruk yöneticisi yapılandırılıyor.....	204
MQTT istemcilerine ileti göndermek için dağıtımli kuyruğa alma yapılandırılması.....	206
MQTT istemci tanıtıcısı, yetkilendirme ve kimlik doğrulaması.....	208
TLS ' yi kullanarak telemetri kanalı kimlik doğrulaması.....	214
telemetri kanallarında yayın gizliliği.....	216
MQTT Java istemcilerinin ve telemetri kanallarının TLS yapılandırması.....	217
Telemetri kanalı JAAS yapılandırması.....	221
YönetmeIBM MQ Light.....	223
Viewing IBM MQ objects in use by MQ Light clients.....	223
MQ Light istemci tanıtıcısı, yetkilendirme ve kimlik doğrulaması.....	225
Kanallarda yayın gizliliği.....	227
Configuring MQ Light clients with TLS.....	228
MQ Light istemcilerinin kuyruk yöneticisinden çıkarılması.....	228
Çoklu yayını yönetme.....	229
Çok noktaya yayın ile çalışmaya başlama.....	229
IBM MQ Multicast konu topolojisi.....	230
Çok hedefli iletilerin boyutunu denetleme.....	231
Multicast ileti sistemi için veri dönüştürmenin etkinleştirilmesi.....	233
Çok hedefli uygulama izleme.....	233
Çok hedefli ileti güvenilirliği.....	234
Gelişmiş çok noktaya gönderim görevleri.....	235
YönetmeIBM MQ for IBM i.....	238
CL komutlarını kullanarak IBM MQ for IBM i ' in yönetilmesi.....	238
IBM MQ for IBM i yönetimine ilişkin diğer yöntemler.....	252
Work management for IBM i.....	256
Availability, backup, recovery, and restart on IBM i.....	264
IBM MQ for IBM i susturulması.....	308
YönetmeIBM MQ for z/OS.....	311
IBM MQ for z/OSkomutlarına komut verilmesi.....	312
IBM MQ for z/OS yardımcı programları.....	320
ÇalışırkenIBM MQ for z/OS.....	322
Writing programs to administer IBM MQ for z/OS.....	342
z/OSüzerindeki IBM MQ kaynaklarını yönetme.....	354
Recovery and restart on z/OS.....	392
IBM MQ ve IMS.....	414
Operating Advanced Message Security on z/OS.....	426
IBM MQ for z/OS Service Provider for z/OS Connect.....	427
Özel notlar.....	463
Programlama arabirimi bilgileri.....	464
Ticari Markalar.....	464

YönetmeIBM MQ

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

IBM MQ nesnelerini yerel olarak ya da uzaktan yönetebilirsiniz; bkz. [“Yerel ve Uzak Denetim” sayfa 8.](#)

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 5](#)
- [“IBM MQ Script \(MQSC\) komutları” sayfa 5](#)
- [“Programlanabilir Komut Biçimleri \(PCF ' ler\)” sayfa 6](#)
- [V 9.0.1 administrative REST API](#)
- [IBM i “IBM i Denetim Dili \(CL\)” sayfa 7](#)

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

- [Linux Windows “The IBM MQ Explorer” sayfa 7](#)
- [V 9.0.1 “The IBM MQ Console” sayfa 7](#)
- [Windows “Windows Varsayılan Yapılanış uygulaması” sayfa 7](#)
- [Windows “Microsoft Küme Hizmeti \(MSCS\)” sayfa 8](#)

[z/OS IBM MQ for z/OS](#) üzerindeki denetim arabirimleri ve seçeneklerle ilgili bilgi için bkz. [“YönetmeIBM MQ for z/OS” sayfa 311.](#)

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 19.](#)

IBM MQ denetim komutları



Denetim komutlarını kullanarak kuyruk yöneticilerindeki yönetim görevlerini kendileri gerçekleştirebilirsiniz.

IBM MQ for Windows, UNIX and Linux® 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.

Komut üzerindeki işaretlere bağlı olarak, runmqsc komutunu üç modda 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 yöneticisine çalıştırılır

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

ULW UNIX, 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 [Building command scripts](#) başlıklı konuya bakın.

IBM i IBM i sunucusunda komutları vermek için, bir Komut Dosyası dosyasında komut listesi oluşturun 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 i üzerinde, bir komut dosyasından yayınlanan komutlara verilen MQSC yanıtları, kuyruğa yollanmış bir dosyada döndürülür.

“Komut dosyası (MQSC) komutları” sayfa 10 , MQSC komutlarının ve sözdiziminin bir açıklamasını içerir.

MQSC komutlarını kullanmaya ilişkin daha fazla bilgi için bkz. [“MQSC komutlarını kullanarak MQ uygulamasını yönetme”](#) sayfa 10.

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 20.

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

The administrative REST API

V 9.0.1

administrative REST API , IBM MQ' u denetlemek için kullanabileceğiniz bir RESTful arabirimi sağlar. administrative REST API' i 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:

V 9.0.4 IBM MQ 9.0.4 ve sonraki yayın:

<https://localhost:9443/ibmmq/rest/v1/admin/installation>

IBM MQ 9.0.3 ve önceki düzeyler:

<https://localhost:9443/ibmmq/rest/v1/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)

IBM i

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

Linux

Windows

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 istasyonlarındaki 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 Explorer komutunu kullanarak yönetim” sayfa 125.](#)

You can use IBM MQ Explorer to administer remote queue managers on other platforms including z/OS, for details and to download the SupportPac MS0T, see <https://www.ibm.com/support/docview.wss?uid=swg24021041>.

The IBM MQ Console

V 9.0.1

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

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

Windows Varsayılan Yapılanış uygulaması

Windows

You can use the Windows Default Configuration program to create a *başlangıç* (or default) set of IBM MQ objects. Yaratılan varsayılan nesnelere bir özet, [Tablo 1: Windows varsayılan yapılandırma uygulaması tarafından yaratılan nesnelere](#) içinde listelenir.

Microsoft Küme Hizmeti (MSCS)

Windows

Microsoft Cluster Service (MSCS), sunucuları bir kümeiçine bağlamanızı, 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* işlemi gerçekleştirir, uygulamaların durum verilerini bozuk bilgisayardan kümedeki başka bir bilgisayara aktarır ve bu işlem burada yeniden başlatılır.

Microsoft Cluster Service (MSCS) 'nin desteklenmesi, IBM MQ for Windows sisteminizin MSCS kullanacak biçimde nasıl yapılandırılacağı hakkında ayrıntılı bilgi sağlar.

İlgili bilgiler

[Komut kümeleri karşılaştırması](#)

[IBM MQ teknik genel bakış](#)

[Planlama](#)

[Yapılandırılıyor](#)

Yerel ve Uzak Denetim

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.

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.

“Uzak IBM MQ nesnelere yönetme” sayfa 184 , uzak yönetmenin konusunu daha ayrıntılı olarak açıklar.

Denetim komutlarını kullanarak IBM MQ uygulamasını yönetme

Denetim komutları, UNIX, Linux, and Windows üzerinde bir dizi IBM MQ denetim görevi gerçekleştirmeniz için bir yolu sağlar.

Denetim komutlarını vermek istiyorsanız, kullanıcı kimliğinizin çoğu denetim komutu için mqm grubunun bir üyesi olması gerekir. Bu konuda daha fazla bilgi için bkz. [UNIX, Linux, and Windows üzerinde 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, bağlanılamaması 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 /U olarak girilebilir.
- SYSTEM.DEAD.LETTER.QUEUE ve jupiter.queue.manager, tam olarak gösterildiği şekilde girilmelidir.

UNIX ve Linux sistemlerinde denetim komutlarının kullanılması

Linux

UNIX

IBM MQ for UNIX ve Linux sistemlerinde, denetim komutlarını bir kabuk penceresinde girmenizi sağlar.

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

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

- Komut adı crtmqmolmalı, 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 bilgiler

[IBM MQ Control komutları başvurusu](#)

MQSC komutlarını kullanarak MQ uygulamasını yönetme

Genel görevleri tamamlamak için MQSC komutları nasıl kullanılır.

MQSC commands are available on all platforms, IBM i ve z/OS dahil.

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. Bu kısım, kuyruk yöneticileri, kuyruklar ve süreç tanımlamalarıyla ilgilidir; kanal, istemci bağlantı kanalı ve dinleyici nesnelere genel bakış için Nesnelerkonusuna bakın. Kuyruk yöneticisi nesnelere ilişkin tüm MQSC komutlarına ilişkin bilgi için bkz. [“Komut dosyası \(MQSC\) komutları” sayfa 10.](#)

You issue MQSC commands to a queue manager using the **runmqsc** command. (Bu komutla ilgili ayrıntılar için bkz. [runmqsc](#).) Bunu etkileşimli olarak yapabilir, komutları bir klavyeden yayınlatabilir ya da bir ASCII metin dosyasından komut dizisi çalıştırmak için standart giriş aygıtını (stdIn) yeniden yönlendirebilirsiniz. Her iki durumda da, komutların biçimi aynı olur. (Komutları bir metin dosyasından çalıştırılmasıyla ilgili bilgi için bkz. [“MQSC komutlarını metin dosyalarından çalıştırma” sayfa 15.](#))

You can run the **runmqsc** command in three ways, depending on the flags set on the command:

- Bir komutu çalıştırmadan, MQSC komutlarının yerel bir kuyruk yöneticisinde doğrulandığı, ancak çalıştırılmadığından emin olun.
- Yerel kuyruk yöneticisinde bir komutu çalıştırın; burada MQSC komutları yerel bir kuyruk yöneticisinde çalıştırılır.
- Uzak kuyruk yöneticisinde bir komut çalıştırın; burada MQSC komutları uzak bir kuyruk yöneticisinde çalıştırılır.

Sözdizimini görüntülemek için bir soru işaretinin ardından da komutu çalıştırabilirsiniz.

MQSC komutlarında belirtilen nesne öznitelikleri bu bölümde büyük/küçük harf ayrımı olmayan (QMNAME gibi) büyük harfli olarak gösterilir. MQSC komut özniteliği adları sekiz karakterle sınırlıdır.

V 9.0.1 Continuous Delivery için, IBM MQ 9.0.1 olanağından, **MQPROMPT** ortam değişkenini kullanarak istediğiniz bir bilgi istemini ayarlayabilirsiniz. In addition to plain text, the **MQPROMPT** variable also allows environment variables to be inserted, by using +VARNAME+ notation, in the same manner as IBM MQ service object definitions (see [“Hizmet Nesnesi Tanımlama” sayfa 173](#)). Örneğin:

```
sh> export MQPROMPT="+USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
sh> runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2023.  
Starting MQSC for queue manager MY.QMGR.  
username @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

V 9.0.0.1 IBM MQ 9.0.0 Fix Pack 1 olanağından, **MQPROMPT** ortam değişkeni de Long Term Support yayınında kullanılabilir.

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

Linux **Windows** Windows ya da Linux (x86 ve x86-64 altyapılarında), bu bölümde açıklanan işlemleri IBM MQ Explorer' i (.) kullanarak da gerçekleştirebilirsiniz. Daha fazla bilgi için, bkz. [“IBM MQ Explorer komutunu kullanarak yönetim” sayfa 125.](#)

İlgili bilgiler

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

[MQSC başvurusu](#)

Komut dosyası (MQSC) komutları

MQSC komutları, IBM MQ altyapılarında insan tarafından okunabilen komutların yayınına ilişkin tek bir yöntem sağlar.

Komutların genel biçimi [MQSC komutları](#) içinde açıklanmıştır.

MQSC komutlarını kullanırken aşağıdaki kuralları dikkate almalısınız:

- Her komut bir birincil deęiştirgeyle (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 deęeri varsa, deęer, ilgili olduęu parametrenin hemen ardından gerçekteşmelidir.

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

- Anahtar sözcükler, ayraçlar ve deęerler, herhangi bir sayıda boşluk ve virgöl ile ayrılabilir. Sözdizimi şemalarında gösterilen bir virgöl her zaman bir ya da daha çok boşlukla deęiştirilebilir. There must be at least one blank immediately preceding each parameter (after the primary parameter) z/OS'da .
- Komutun başında ya da sonunda ve parametreler, noktalama işaretleri ve deęerler 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, bunlar tırnak işaretleri içindeki dizgilerde deęilse).
- Yinelenen parametrelere izin verilmez. REPLACE NOREPLACE gibi "NO" sürümüne sahip bir deęiştirgenin yinelenmesine de izin verilmez.
- Boşluk, küçük harf karakterleri ya da özel karakterler içeren dizimler:

- Nokta (.)
- Eğik Çizgi (/)
- Alt çizgi (_)
- Yüzde işareti (%)

şu ana kadar tek tırnak işareti içine alınmalıdır:

- z/OS Issued from the IBM MQ for z/OS operations and control panels
- Yıldız işaretiyle biten genel deęerler (IBM i üzerinde bunlar tek tırnak işareti içine alınmalıdır)
- Tek bir yıldız imi; örneğin, TRACE (*) (IBM i üzerinde bunlar tek tırnak işareti içine alınmalıdır)
- İki nokta imi içeren bir aralık belirtimi; örneğin, 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 Çoklu platformlar 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. Bu kural dışı durum, kullanılmakta olan özniteliğin aşağıdakilerden biri olması durumunda olur:

- TOPICSTR
- SUB
- USERDATA
- SELECTOR

daha sonra boşluk içermeyen iki tek tırnak işareti sıfır uzunluklu bir dizgi olarak yorumlanır.

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.

- IBM WebSphere MQ 7.0'ta, SELECTOR, alt kullanıcı verileri gibi MQCHARV tiplerini temel alan bu dizgi özniteliklerindeki sondaki boşluklar, ' abc ' anlamına gelen anlamlı bir deęer olarak deęerlendirilir. 'abc' ile eşit deęildir.

- Sol ayraç ve ardından sağ parantez, arasında önemli bir bilgi yok; örneğin,

NAME ()

özellikle belirtilenler dışında geçerli değildir.

- 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; DEFİ, DEFE için geçerli bir eşanlamlı değil.

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

IBM MQ denetimi için MQSC komutlarının kullanılmasına ilişkin bir genel bakış için bkz. [“MQSC komutlarını kullanarak MQ uygulamasını yönetme” sayfa 10.](#)

MQSC komutları, bazı özel karakterleri belirli anlamlara sahip olacak şekilde kullanır. Bu özel karakterlerle ve bunların nasıl kullanılacağı hakkında daha fazla bilgi için [Özel anlamlar içeren karakterler başlıklı](#) konuya bakın.

MQSC komutlarını kullanarak komut dosyalarını nasıl oluşturabileceğiniz hakkında bilgi almak için [Building command scripts](#) başlıklı konuya bakın.

MQSC komutlarının tam listesi için bkz. [MQSC komutları.](#)

You can issue this command from sources 2CR. Kaynak simgelerle ilgili açıklamalar için [z/OS üzerindeki komutların kullanılması](#) konusuna bakın.

İlgili kavramlar

[“IBM MQ Programlanabilir Komut Biçimlerine Giriş” sayfa 20](#)

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 ağlar, boyut ve karmaşıklık içinde büyüdükçe.

İlgili bilgiler

[Komut komut dosyaları oluşturuluyor](#)

MQSC komutlarındaki IBM MQ nesne adları

MQSC komutlarında nesne adlarının kullanılması.

Örneklerde, nesnelere için bazı uzun adlar kullanıyoruz. Bu, uğraştığınız nesne tipini tanımlamanıza yardımcı olur.

MQSC komutlarını verdiğinizde, yalnızca kuyruğun yerel adını belirtmeniz gerekir. Örneklerimizde, aşağıdaki kuyruk adlarını kullanırız:

```
ORANGE . LOCAL . QUEUE
```

Adın LOCAL . QUEUE kısmı, bu kuyruğun yerel bir kuyruk olduğunu göstermekten başka bir neden değildir. Genel olarak yerel kuyrukların adları için **zorunlu değildir** .

Ayrıca, kuyruk yöneticisi adı olarak `saturn.queue.manager` adını da kullanırız. Adın `queue.manager` kısmı, bu nesnenin bir kuyruk yöneticisi olduğunu göstermek için gösterilir. Genel olarak kuyruk yöneticilerinin adları için bu gerekli değildir.

MQSC komutlarında büyük ve küçük harfe duyarlılık

MQSC komutları, öznitelikleri de içinde olmak üzere büyük harfli ya da küçük harfli olarak yazılabilir. Adları tek tırnak içine alınmadıkça, MQSC komutlarındaki nesne adları büyük harfe çevrilir (yani, QUEUE

ve kuyruk ayırt edilmiyor). Tırnak işaretleri kullanılmazsa, nesne büyük harfle bir ad ile işlenir. Ek bilgi için [Özel anlamları olan karakterler](#) başlıklı konuya bakın.

Tüm IBM MQ denetim komutlarıyla ortak olarak `runmqsc` komut çağırısı, bazı IBM MQ ortamlarında büyük/küçük harfe duyarlıdır. Ek bilgi için [“Denetim komutlarını kullanarak IBM MQ uygulamasını yönetme”](#) sayfa 8 başlıklı konuya bakın.

Standart giriş ve çıkış

`stdin` olarak da adlandırılan *standart giriş aygıtı*, sisteme girişin alındığı aygıttır. Genellikle bu klavye olur, ancak girişin bir dizesel kapıdan ya da bir disk dosyasından geldiğini belirtebilirsiniz. `stdout` olarak da bilinen *standart çıkış aygıtı*, sistemin çıktısının gönderildiği aygıttır. Genellikle bu bir görüntüdür, ancak çıkışı bir dizesel kapıya ya da bir dosyaya yeniden yönlendirebilirsiniz.

İşletim sistemi komutlarında ve IBM MQ denetim komutlarında `<` işleç girişi yeniden yönlendirir. Bu işleç bir dosya adıyla takip edildiyse, girişten giriş alınır. Benzer şekilde, `>` işleci çıkışı yeniden yönlendirir; bu işleç bir dosya adıyla takip edildiyse, çıkış o dosyaya yönlendirilir.

MQSC komutlarının etkileşimli olarak kullanılması

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

MQSC komutlarını etkileşimli olarak kullanmak için bir komut penceresi ya da kabuk açın ve şunu girin:

```
runmqsc
```

Bu komutta, kuyruk yöneticisi adı belirtilmemiştir, bu nedenle MQSC komutları varsayılan kuyruk yöneticisi tarafından işlenir. Farklı bir kuyruk yöneticisi kullanmak istiyorsanız, **runmqsc** komutunda kuyruk yöneticisi adını belirtin. For example, to run MQSC commands on queue manager `jupiter.queue.manager`, use the command:

```
runmqsc jupiter.queue.manager
```

Bundan sonra, yazdığınız tüm MQSC komutları bu kuyruk yöneticisi tarafından işlenir; bunun aynı düğümde olduğunu ve zaten çalışmakta olduğunu varsayarak.

Şimdi gerektiği şekilde, herhangi bir MQSC komutuna yazabilirsiniz. Örneğin, şunu deneyin:

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

Bir satıra sığması gereken çok fazla sayıda parametre içeren komutlar için, aşağıdaki satırda bir komutun devam ettiğini belirtmek için devam karakterleri kullanın:

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

Komut girişi, devam karakteri olmayan boş olmayan bir satırın son karakteriyle sonlandırılır. Ayrıca, komut girişini bir noktalı virgül (;) girerek belirtik olarak sonlandırabilirsiniz. (Bu, özellikle komut girişinin son satırının sonunda yanlışlıkla bir devama karakteri girdiğinizde yararlı olur.)

MQSC komutlarından geribildirim

MQSC komutlarını verdiğinizde, kuyruk yöneticisi işlemlerinizi doğrulayan işletmen iletilerini döndürür ya da size yaptığınız hatalardan söz eder. Örneğin:

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

Bu ileti, bir kuyruğun yaratıldığını doğrular.

```
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 ileti, bir sözdizimi hatası yaptığınızı gösterir.

Bu iletiler standart çıkış aygıtına gönderilir. Komutu doğru olarak girmediyseniz, doğru sözdizimi için [MQSC komutları](#) konusuna bakın.

MQSC komutlarının etkileşimli girişi sona erdiriliyor

MQSC komutlarıyla çalışmayı durdurmak için END komutunu girin.

Diğer bir seçenek olarak, işletim sisteminiz için EOF karakterini de kullanabilirsiniz.

İlgili kavramlar

[“MQSC komutlarını metin dosyalarından çalıştırma” sayfa 15](#)

MQSC komutlarının etkileşimli olarak çalıştırılması hızlı sınamalar için uygundur; ancak, çok uzun komutlara sahipseniz ya da belirli bir komut dizisini sürekli olarak kullanıyorsanız, stdin dosyasını bir metin dosyasından yeniden yönlendirmeyi deneyin.

İlgili bilgiler

[runmqsc](#)

UNIX ve Linux üzerinde runmqsc komutu

UNIX ve Linux üzerindeki **runmqsc** komut satırı, komut geri çağırma, komut tamamlama ve Emacs komut anahtarlarını destekler.

Aşağıdaki komut satırı düzenleyicisi işlevleri kullanılabilir:

- Daha önce girilen komutları geri almak için yukarı ok tuşunu ve aşağı ok tuşunu kullanın.
- Sekme tuşunu ve boşluk çubuğunu kullanarak bir komutun sonraki anahtar sözcüğü için otomatik tamamlama
- Emacs komut anahtarları ya da benzeri komut anahtarı işlevleri

Bu işlevleri kullanmak için, curses kitaplığının kurulu olması gerekir. Sistem kitaplığı sisteminizde kurulu değilse, **runmqsc** komut satırı düzenleyicisi işlevlerine sahip değildir ve **runmqsc** komut satırı başlatıldığında bir ileti görüntülenir. Kurulacak lansses kitaplığının adı UNIX platformuna bağlıdır:

- **AIX** AIX üzerinde, curses' yi kurun.
- **HP-UX** HP-UX üzerinde, Xcurses' yi kurun.
- Diğer tüm UNIX platformlarında ve Linux üzerinde, ncurses' i kurun.

Emacs anahtar bađ tanımlarının uyarlanması

Komutlara bađlı olan anahtarları uyarlayabilirsiniz. Örneđin, varsayılan Emacs tuş bađ tanımları yerine, anahtarları vi bađ tanımlarına bađlayabilirsiniz.

Anahtarlar, ana dizinde saklanan `.editrc` dosyası düzenlenerek uyarlanır. Ek bilgi için FreeBSD bilgi işlem sayfalarındaki [editrc](#) başlıklı konuya bakın.

Komut geri çağırma, komut tamamlama ve Emacs komut anahtarlarını devre dışı bırakma

Bir ortam deđiřkeni ayarlayarak, komut geri çağırma, komut tamamlama ve Emacs komut tuřlarını geersiz kılabilirsiniz. Set the environment variable `MQ_OVERRIDE_LIBEDIT_LOAD` to TRUE.

Bu ortam deđiřkeni, `runmqsc` ařađıdaki bilgilendirici iletiyi gsterdiđinde geici bir zm olarak kullanılabilir:

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

MQSC komutlarını metin dosyalarından alıřtırma

MQSC komutlarının etkileřimli olarak alıřtırılması hızlı sınamalar iin uygundur; ancak, ok uzun komutlara sahipseniz ya da belirli bir komut dizisini srekli olarak kullanıyorsanız, `stdin` dosyasını bir metin dosyasından yeniden ynlendirmeyi deneyin.

`stdin` 'u bir metin dosyasından yeniden ynlendirmek iin, nce olađan metin dzenleyicinizi kullanarak MQSC komutlarını ieren bir metin dosyası oluřturun ve sonra `runmqsc` komutunu alıřtırın.

Not: If you run the `runmqsc` command in client mode by redirecting `stdin` from a text file, IBM MQ expects the first line of the input file to be a password.

`runmqsc` komutunu kullandıđınızda yeniden ynlendirme iřlelerini kullanın. Örneđin, ařađıdaki komut `myprog.in` metin dosyasında bulunan bir komut dizisini alıřtırır:

```
runmqsc < myprog.in
```

Benzer řekilde, ıktıyı bir dosyaya da ynlendirebilirsiniz. Giriř iin MQSC komutlarını ieren bir dosya, bir MQSC komut dosyası adı verilir. Kuyruk yneticisinden yanıt ieren ıkıř dosyasına ıkıř dosyası adı verilir.

`stdin` ve `stdout` komutlarını `runmqsc` komutuna yeniden ynlendirmek iin komutun bu biimini kullanın:

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

Bu komut, `myprog.in`. MQSC komut dosyasında bulunan MQSC komutlarını ađırır; bir kuyruk yneticisi adı belirtmediđimizden, MQSC komutları varsayılan kuyruk yneticisine karřı alıřtırılıyor. ıkıř, `myprog.out` metin ktđüne gnderilir. [řekil 1 sayfa 16](#) shows an extract from the MQSC command file `myprog.in` and [řekil 2 sayfa 17](#) shows the corresponding extract of the output in `myprog.out`.

To redirect `stdin` and `stdout` on the `runmqsc` command, for a queue manager (`saturn.queue.manager`) that is not the default, use this form of the command:

```
runmqsc saturn.queue.manager < myprog.in > myprog.out
```

MQSC komut dosyaları

MQSC komutları, ASCII metin biiminde, insan tarafından okunabilen bir biimde yazılır. [řekil 1 sayfa 16](#) , znelikleriyle birlikte bir MQSC komutunu (**DEFINE QLOCAL**) gsteren bir MQSC komut ktđünden ıkarılır. [MQSC komutları](#) , her MQSC komutunun ve szdiziminin bir aıklamasını ierir.

```
.  
. .  
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 kütüğünden veri al

IBM MQ ortamları arasında taşınabilirlik için, MQSC komut dosyalarındaki satır uzunluğunu 72 karaktere sınırlayın. Artı işareti, komutun sonraki satırda devam ettiğini gösterir.

MQSC komut raporları

runmqsc komutu, stdout' a gönderilen bir rapor döndürür. Rapor şunları içerir:

- Raporun kaynağı olarak MQSC komutlarını tanıtan bir üstbilgi:

```
Starting MQSC for queue manager jupiter.queue.manager.
```

Burada `jupiter.queue.manager`, kuyruk yöneticisinin adıdır.

- Yayınlanan MQSC komutlarının numaralandırılmış isteğe bağlı listesi. Varsayılan değer olarak, girişin metni çıkışa yansıtılır. Bu çıkış içinde, her komutun önünde bir sıra numarası bulunur (Şekil 2 sayfa 17' da gösterildiği gibi). Ancak, çıktıyı gizlemek için **runmqsc** komutundaki **-e** işaretini kullanabilirsiniz.
- Hatalı olduğu belirtilen komutlara ilişkin bir sözdizimi hatası iletisi.
- Her komutun çalıştırılabilmesinin sonucunu gösteren bir işletmen iletisi. Örneğin, bir **DEFINE QLOCAL** komutunun başarıyla tamamlanmasına ilişkin işletmen iletisi şöyledir:

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

- Komut dosyasını çalıştırırken genel hatalardan kaynaklanan diğer iletiler.
- Okunan komutların sayısını, sözdizimi hataları olan komut sayısını ve işlenmeyen komut sayısını gösteren kısa bir istatistik özeti.

Not: Kuyruk yöneticisi yalnızca sözdizimi hataları olmayan komutları işlemeyi dener.


```
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. Bir MQSC komut raporu dosyasından çıkarma

Sağlanan MQSC komut dosyalarının çalıştırılması

Aşağıdaki MQSC komut dosyaları IBM MQ ile birlikte sağlanır:

amqscos0.tst

Örnek programlar tarafından kullanılan nesnelere ilişkin tanımlar.

amqscic0.tst

CICS işlemlerine ilişkin kuyrukların tanımlamaları.

Windows IBM MQ for Windows' ta bu dosyalar, `MQ_INSTALLATION_PATH\tools\mqsc\samplesdizininde` bulunur. `MQ_INSTALLATION_PATH`, IBM MQ ' in kurulu olduğu üst düzey dizini temsil eder.

Linux **UNIX** UNIX and Linux sistemlerinde bu dosyalar `MQ_INSTALLATION_PATH/sampdizininde` bulunur. `MQ_INSTALLATION_PATH`, IBM MQ ' in kurulu olduğu üst düzey dizini temsil eder.

Şu komutu çalıştıran komut:

```
runmqsc < amqscos0.tst >test.out
```

Komutları doğrulamak için runmqsc komutunu kullanma

Bir yerel kuyruk yöneticisine ilişkin MQSC komutlarını gerçekten çalıştırmadan doğrulamak için **runmqsc** komutunu kullanabilirsiniz. Bunu yapmak için, **runmqsc** komutuna **-v** işaretini ayarlayın; örneğin:

```
runmqsc -v < myprog.in > myprog.out
```

runmqsc ' u bir MQSC komut dosyasına karşı çağırdığınızda, kuyruk yöneticisi her komutu doğrular ve MQSC komutlarının çalıştırılmadan bir rapor döndürür. Bu, komut dosyanızdaki komutların sözdizimini denetlemenizi sağlar. Bu özellikle aşağıdaki durumlarda önemlidir:

- Bir komut kütüğünden çok sayıda komutu çalıştırma.
- Birçok kez bir MQSC komut dosyası kullanılıyor.

Döndürülen rapor, Şekil 2 sayfa 17 içinde gösterilen raporla benzerdir.

MQSC komutlarını uzaktan doğrulamak için bu yöntemi kullanamazsınız. Örneğin, bu komutu denerseniz:

```
runmqsc -w 30 -v jupiter.queue.manager < myprog.in > myprog.out
```

Kuyruk yöneticisinin uzak olduğunu göstermek için kullandığınız **-w** işareti yoksayılr ve komut, doğrulama kipinde yerel olarak çalıştırılır. 30 is the number of seconds that IBM MQ waits for replies from the remote queue manager.

İlgili kavramlar

“Standart giriş ve çıkış” sayfa 13

stdınolarak da adlandırılan *standart giriş aygıtı*, sisteme girişin alındığı aygıttır. Genellikle bu klavye olur, ancak girişin bir dizele kapıdan ya da bir disk dosyasından geldiğini belirtebilirsiniz. stdoutolarak da bilinen *standart çıkış aygıtı*, sistemin çıktısının gönderildiği aygıttır. Genellikle bu bir görüntüdür, ancak çıkışı bir dizele kapıya ya da bir dosyaya yeniden yönlendirebilirsiniz.

“MQSC komutlarının etkileşimli olarak kullanılması” sayfa 13

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

İlgili bilgiler

[runmqsc](#)

Toplu iş dosyalarından MQSC komutları çalıştırılıyor

Çok uzun komutlarınız varsa ya da belirli bir komut dizisini sürekli olarak kullanıyorsanız, stdın komutunu bir toplu iş kütüğünden yeniden yönlendirmeyi deneyin.

Bir toplu iş kütüğünden stdın ögesini yeniden yönlendirmek için, önce olağan metin düzenleyicinizi kullanarak MQSC komutlarını içeren bir toplu iş kütüğü yaratın. runmqsc komutunu kullandığınızda yeniden yönlendirme işleçlerini kullanın. Aşağıdaki örnek:

1. Test kuyruğu yöneticisi (TESTQM) yaratır
2. TCP/IP kapısı 1600 'i kullanmak için eşleşen bir CLNTCONN ve dinleyici kümesi yaratır
3. Sınama kuyruğu yaratır, TESTQ
4. amqspuc örnek programını kullanarak, kuyruğa ileti koyar.

```
export MYTEMPQM=TESTQM
export MYPOR=1600
export MQCHLLIB=/var/mqm/qmgrs/$MQTEMPQM/@ipcc

crtmqm $MYTEMPQM
strmqm $MYTEMPQM
runmqsc -m $MYTEMPQM -t TCP -p $MYPOR &

runmqsc $MYTEMPQM << EOF
DEFINE CHANNEL(NLTM) CHLTYPE(SVRCONN) TRPTYPE(TCP)
DEFINE CHANNEL(NLTM) CHLTYPE(CLNTCONN) QMNAME('$MYTEMPQM') CONNAME('hostname($MYPOR)')
ALTER CHANNEL(NLTM) CHLTYPE(CLNTCONN)
DEFINE QLOCAL(TESTQ)
EOF

amqspuc TESTQ $MYTEMPQM << EOF
hello world
EOF

endmqm -i $MYTEMPQM
```

Şekil 3. Bir toplu iş kütüğünden MQSC komutlarını çalıştırmak için kullanılan örnek komut dosyası

MQSC komutlarıyla ilgili sorunların çözülmesi

Çalıştırmak üzere MQSC komutları alamıyorsanız, bu ortak sorunlardan birinin size uygulanırsa da bu konudaki bilgileri kullanın. Bir komutun ürettiği hatayı okuduğunuzda sorunun ne olduğu her zaman açık değildir.

runmqsc komutunu kullandığınızda, aşağıdaki bilgileri anımsatın:

- Bir dosyadan girişi yeniden yönlendirmek için < işlecini kullanın. Bu işleci çıkarırsanız, kuyruk yöneticisi dosya adını kuyruk yöneticisi adı olarak yorumlar ve aşağıdaki hata iletisini yayınlar:

```
AMQ8118E: IBM MQ queue manager does not exist.
```

- Çıkışı bir dosyaya yeniden yönlendiriyorsanız, > yeniden yönlendirme işlecini kullanın. Varsayılan olarak, dosya, runmqsc zamanındaki geçerli çalışma dizinine konmaktadır. Çıktınızı belirli bir dosya ve dizine göndermek için tam olarak nitelenmiş bir dosya adı belirtin.
- Tüm kuyruk yöneticilerini görüntülemek için aşağıdaki komutu kullanarak, komutları çalıştırabilmek için kuyruk yöneticisini yarattığınız doğrulayın.

```
dspmq
```

- Kuyruk yöneticisi çalışıyor olmalıdır. Doğru değilse başlatın; (bkz. [Kuyruk Yöneticisi Başlatılması](#)). Çalışmakta olan bir kuyruk yöneticisini başlatmaya çalışırsanız, bir hata iletisi alabilirsiniz.
- Varsayılan bir kuyruk yöneticisi tanımlamadıysanız ya da bu hatayı aldıysanız, runmqsc komutunda bir kuyruk yöneticisi adı belirtin:

```
AMQ8146E: IBM MQ queue manager not available.
```

- runmqsc komutunun parametresi olarak bir MQSC komutu belirleyemezsiniz. Örneğin, bu geçerli değildir:

```
runmqsc DEFINE QLOCAL(FRED)
```

- **runmqsc** komutunu vermeden önce MQSC komutlarını giremezsiniz.
- You cannot run control commands from **runmqsc**. Örneğin, MQSC komutlarını etkileşimli olarak çalıştırırken bir kuyruk yöneticisi başlatmak için stirmqm komutunu yayınlamazsınız. Bunu yapmazsanız, aşağıdakine benzer hata iletileri alırsınız:

```
runmqsc
.
.
Starting MQSC for queue manager jupiter.queue.manager.

1 : stirmqm saturn.queue.manager
AMQ8405: Syntax error detected at or near end of cmd segment below:-s

AMQ8426: Valid MQSC commands are:
ALTER
CLEAR
DEFINE
DELETE
DISPLAY
END
PING
REFRESH
RESET
RESOLVE
RESUME
START
STOP
SUSPEND
2 : end
```

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 ağıdaki 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 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 (*MqType*) 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 20.

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 ağı içindeki herhangi bir kuyruk yöneticisi (PCF ' leri destekleyen) arasında değiş tokuş edilebilen komut ve yanıt iletilerini tanımlar. PCF ' ler kuyruk yöneticisi yönetimini ve diğer ağı yönetimini basitleştirir. Bunlar, dağıtık ağların karmaşık yönetimi sorununu çözmek için kullanılabilir, özellikle de ağlar, boyut ve karmaşıklık içinde büyüdükçe.

Programlanabilir Komut Biçimleri aşağıdaki tarafından desteklenmektedir:

-  IBM MQ for AIX
-  IBM MQ for HP-UX

- **IBM i** IBM MQ for IBM i
- **Linux** IBM MQ for Linux
- **Solaris** IBM MQ for Solaris
- **Windows** IBM MQ for Windows
- **z/OS** 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 kuyukların 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 22](#)
- [“IBM MQ' da PCF yanıtları” sayfa 24](#)

- **z/OS** “Genişletilmiş yanıtlar” sayfa 26
- [IBM MQ nesnelerinin adlandırılmasına ilişkin kurallar](#)
- [“IBM MQ' ta PCF komutları için yetki denetimi” sayfa 28](#)

PCF komut iletileri

PCF komut iletileri, bir PCF üstbilgisinden, o üstbilgide tanımlanan deęiřtirgelerden ve kullanıcı tanımlı ileti verilerinden oluşur. İletiler, İleti Kuyruęu arabirimi çağrılarını kullanılarak 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 dizge deęiřtirgesi](#). PCF üstbilgisi, komutu ve aynı iletilide 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 Çoklu platformlar' ta, PCF komutlarının gönderileceęi kuyruk her zaman SYSTEM.ADMIN.COMMAND.QUEUE.

z/OS z/OS'ta komutlar SYSTEM.COMMAND.INPUT' e gönderilir, ancak SYSTEM.ADMIN.COMMAND.QUEUE bu öęe için bir dięer ad olabilir. Bu kuyruęa hizmet veren komut sunucusu, yanıtları, komut iletilisinin 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 doğrulayın.

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

Bir PCF komutuna ilişkin 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 geçerli deęer, gerektięi gibi.

MsgType

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

Son kullanma tarihi

Herhangi bir geçerli deęer, gerektięi gibi.

Geribildirim

MQFB_NONE deęerine ayarla

Kodlama

Ařaęıdaki sistemlerden birine gönderiyorsanız, bu alanı ileti verileri için kullanılan kodlamaya ayarlayın; gerekirse dönüřtürme işlemi gerçekleştirilir:

- **IBM i** IBM i
- **Linux** Linux

-  UNIX
-  Windows

CodedCharSetId

Aşağıdaki sistemlerden birine gönderiyorsanız, bu alanı ileti verileri için kullanılan kodlanmış karakter takımı tanıtcısı olarak ayarlayın; gerekirse dönüştürme gerçekleştirilir:

-  IBM i
-  Linux
-  UNIX
-  Windows

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 tanıtcısı oluşturmasını istemek için MQMI_NONE belirtilebilir.

CorrelId

Gönderme uygulaması herhangi bir değer belirtebilir ya da ilinti tanıtcısı 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 kuyrukta gönderilmesi ve alınması

PCF iletileri belirtilen bir kuyruğa gönderiliyor

Belirli 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: Torba bir denetim iletisi içeriyorsa ve torbaya değer eklemek için mqAddSorgu çağrısı kullanıldıysa, MQIASY_COMMAND veri ögesinin değeri, MQAI tarafından tanınan bir QUARY komutu olmalıdır.

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

Belirlenen kuyruktan PCF iletileri alınıyor

Belirli bir 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 Explorer on Windows 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 (Programlanır Komut Biçimlerinin Tanımlamaları) *her zaman geri döndürülür*. Bu nedenle, bu yanıtlar için, verilerin hangi nesneye uygulanacağı saptanması için yanıtta nesnelere 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 izlenebilir.

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 MÖCC_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 MÖCFIN yapısı ve aşağıdakilerden birini içeren bir *Parameter* alanı ile birlikte:

- MÖIACF_PARAMETER_ID

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

- MÖIACF_ERROR_ID

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

- MÖIACF_SELECTOR

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

- MÖIACF_ERROR_OFFSET

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

- MÖIA_CODED_CHAR_SET_ID

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

Son (ya da yalnızca) hata yanıtı iletisi bir özet yanıtıdır; *CompCode* alanı MÖCC_FAILED ve bir *Reason* alanı MÖRCFC_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, MÖCFT_XR_MSG tipi ile
- MÖCFT_XR_ITEM tipi ile öge yanıtı
- MÖCFT_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 MÖCFC_SON değeri var. Kümedeki diğer tüm yanıtlar için bu değer MÖCFC_NOT_SON ' dir.

Herhangi bir yanıt, *Parameter* alanının MÖBACF_RESPONSE_SET değerine ayarladığı isteğe bağlı MÖCFBS 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 MÖCFBS 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 belirleyebilseniz 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 iletilisi, 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 iletilisi 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 iletilisi yanıtları 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 iletilen 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



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 1 sayfa 29.](#)

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 QMQADM 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



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 1 sayfa 29.](#)

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 Deęiřtir
- Kanalı Kopyala
- Kanal Yarat
- Kanalı Sil
- Ping Kanalı
- Kanalı Sıfırla
- Bařlangıç Kanalı
- Kanalı Durdur
- Kanal Bařlatıcısını Bařlat
- Kanal Dinleyicisi Bařlat
- Kanalı Çözümle
- Kümeyi Sıfırla
- Kümeyi Yenile
- Kuyruk Yöneticisini Askıya Al
- Kuyruk Yöneticisini Sürdür

Çoklu Platformlar için IBM MQ Nesnesi Yetkilileri

Multi

Çizelge 1. Nesne yetkileri		
Komut	IBM MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
Kimlik Doğrulama Bilgilerini Deęiřtir	dsp ve chg	yok
Kanalı Deęiřtir	dsp ve chg	yok
Kanal Dinleyiciyi Deęiřtir	dsp ve chg	yok
İstemci Baęlantı Kanalını Deęiřtir	dsp ve chg	yok
Ad listesini deęiřtir	dsp ve chg	yok
Süreci Deęiřtir	dsp ve chg	yok
Kuyruęu Deęiřtir	dsp ve chg	yok
Kuyruk Yöneticisini Deęiřtir	chg <i>bkz. Not 3 ve Not 5</i>	yok
Hizmeti Deęiřtir	dsp ve chg	yok
Kuyruęu Temizle	klr	yok
Kimlik Doğrulama Bilgilerini Kopyala	dsp	crt
Kimlik Doğrulama Bilgilerini Kopyala (Deęiřtir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
Kanalı Kopyala	dsp	crt
Kanal Kopyala (Deęiřtir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
Kanal Dinleyicisi Kopyala	dsp	crt

<i>Çizelge 1. Nesne yetkileri (devamı var)</i>		
Komut	IBM MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
Channel Listener 'ı Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
İstemci Bağlantı Kanalını Kopyala	dsp	crt
İstemci Bağlantı Kanalını Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
Ad listesini kopyala	dsp	crt
Ad Listesi Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: dsp ve chg</i>	crt
İşlemi Kopyala	dsp	crt
Kopyalama İşlemi (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
Kuyruğu Kopyala	dsp	crt
Kopyalama Kuyruğu (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: dsp ve chg</i>	crt
Kimlik Doğrulama Bilgileri Oluştur	<i>(sistem varsayılan kimlik doğrulama bilgileri) dsp</i>	crt
Kimlik Doğrulama Bilgileri Oluştur (Değiştir) <i>bkz. Not 1</i>	<i>(sistem varsayılan kimlik doğrulama bilgileri) dsp to: chg</i>	crt
Kanal Yarat	<i>(sistem varsayılan kanalı) dsp</i>	crt
Kanal Yarat (Değiştir) <i>bkz. Not 1</i>	<i>(sistem varsayılan kanalı) dsp to: chg</i>	crt
Kanal Dinleyicisi Yarat	<i>(sistem varsayılan dinleyicisi) dsp</i>	crt
Kanal Dinleyici Yarat (Değiştir) <i>bkz. Not 1</i>	<i>(sistem varsayılan dinleyicisi) dsp to: chg</i>	crt
İstemci Bağlantı Kanalı Oluştur	<i>(sistem varsayılan kanalı) dsp</i>	crt
İstemci Bağlantı Kanalı Oluştur (Değiştir) <i>bkz. Not 1</i>	<i>(sistem varsayılan kanalı) dsp to: chg</i>	crt
Ad Listesi Yarat	<i>(sistem varsayılan ad listesi) dsp</i>	crt
Ad Listesi Yarat (Değiştir) <i>bkz. Not 1</i>	<i>(sistem varsayılan ad listesi) dsp to: dsp ve chg</i>	crt
Süreç Yarat	<i>(sistem varsayılan işlemi) dsp</i>	crt
Süreç Yarat (Değiştir) <i>bkz. Not 1</i>	<i>(sistem varsayılan süreci) dsp to: chg</i>	crt
Kuyruk Yarat	<i>(sistem varsayılan kuyruğu) dsp</i>	crt
Kuyruk Yarat (Değiştir) <i>bkz. Not 1</i>	<i>(sistem varsayılan kuyruğu) dsp to: dsp ve chg</i>	crt
Hizmet oluştur	<i>(sistem varsayılan kuyruğu) dsp</i>	crt
Hizmet Oluştur (Değiştir) <i>bkz. Not 1</i>	<i>(sistem varsayılan kuyruğu) dsp to: chg</i>	crt
Kimlik Doğrulama Bilgilerini Sil	dsp ve dlt	yok

Çizelge 1. Nesne yetkileri (devamı var)

Komut	IBM MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
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
Kimlik Doğrulama Bilgilerini Sorgula	dsp	yok
Yetki Kayıtlarını Sorgula	bkz. Not 4	bkz. Not 4
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	not 3 'e bakın	yok
Sorgu Kuyruğu Durumu	dsp	yok
Hizmeti Sorgula	dsp	yok
Ping Kanalı	Ctrl	yok
Ping Kuyruğu Yöneticisi	not 3 'e bakın	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	(kuyruk yöneticisi nesnesi) chg bkz. Not 4	bkz. Not 4
Başlangıç Kanalı	Ctrl	yok

Çizelge 1. Nesne yetkileri (devamı var)

Komut	IBM MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
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:

- Değiştirilecek nesne varsa, bu komut geçerli olur; tersi durumda, yetki denetimi Create ya da Copy withod Replace gibi olur.
- Gereken yetki, çıkış metninin tanımladığı MQSC komutu tarafından belirlenir ve önceki komutlardan birine eşdeğerdir.
- Herhangi bir PCF komutunu işleyebilmek için, kullanıcı kimliğinin hedef sistemdeki kuyruk yöneticisi nesnesine ilişkin dsp yetkisi olmalıdır.
- Bu PCF komutu, komut sunucusu -a değiştirgesiyle başlatılmadıkça yetkilendirilir. Varsayılan olarak, komut sunucusu kuyruk yöneticisi başlatıldığında ve -a değiştirgesi olmadan başlar. Ek bilgi için [Programlanır komut biçimleri başvurusu](#) başlıklı konuya bakın.
- 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 görüntülenmesi](#) başlıklı 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, HP-UX, IBM i, Linux, Solaris, ve Windows. Nesneleri (ya da değiştirgeleri) Programlanır 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, nesnelerin özelliklerini (ya da parametrelerini), PCF ' leri kullanmaktan daha kolay bir şekilde işlemenize 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 dizginin 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 değiştirgeleri kullanılarak yazılan programlar uygun veri torbasına yazılır ve her yapı için tek bir deyim gerektirir. 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 işlemlerini kolaylaştırır.

Uygulamalar arasında veri alışverişi

Uygulama verileri, PCF biçiminde gönderilir ve MQAI tarafından paketlenmez ve paketlenmez. İletim verileriniz tamsayılar ve karakter dizgilerinden oluşuyorsa, PCF verileri için IBM MQ yerleşik veri dönüştürmesinden yararlanmak için MQAI ' yi kullanabilirsiniz. Bu, veri dönüştürme çıktıları yazma gereksinmesini önler.

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

MQAI ' nin kullanılmasına ilişkin örnekler

Gösterilen liste, MQAI kullanımını gösteren bazı örnek programlar verir. Örnekler aşağıdaki görevleri gerçekleştirir:

1. Yerel bir kuyruk yaratır. [“Yerel kuyruk yaratmak için Örnek C programı \(amqsaicq.c\)” sayfa 38](#)
2. Ekrandaki olayları basit bir olay izleyicisini kullanarak görüntüler. [“Olay izleyiciyi kullanarak olayları görüntülemek için Örnek C programı \(amqsaiem.c\)” sayfa 42](#)
3. Tüm yerel kuyrukların ve bunların yürürlükteki derinliklerinin listesini yazdırır. [“Kuyruklar ve yazdırma bilgileri \(amqsailq.c\) hakkında araştırma yapmak için C programı örneği” sayfa 54](#)
4. Tüm kanalların ve tiplerinin bir listesini yazdırır. [“Kanal nesnelerinin araştırılmasına ilişkin örnek C programı \(amqsaicl.c\)” sayfa 49](#)

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 iletimlerini kullanır. [Tips for configuring IBM MQ using the MQAI can be found in “Hints and tips for using MQAI to configure IBM MQ” sayfa 33.](#)

İlgili bilgiler

[IBM MQ Administration Interface başvurusu](#)

Hints and tips for using MQAI to configure IBM MQ

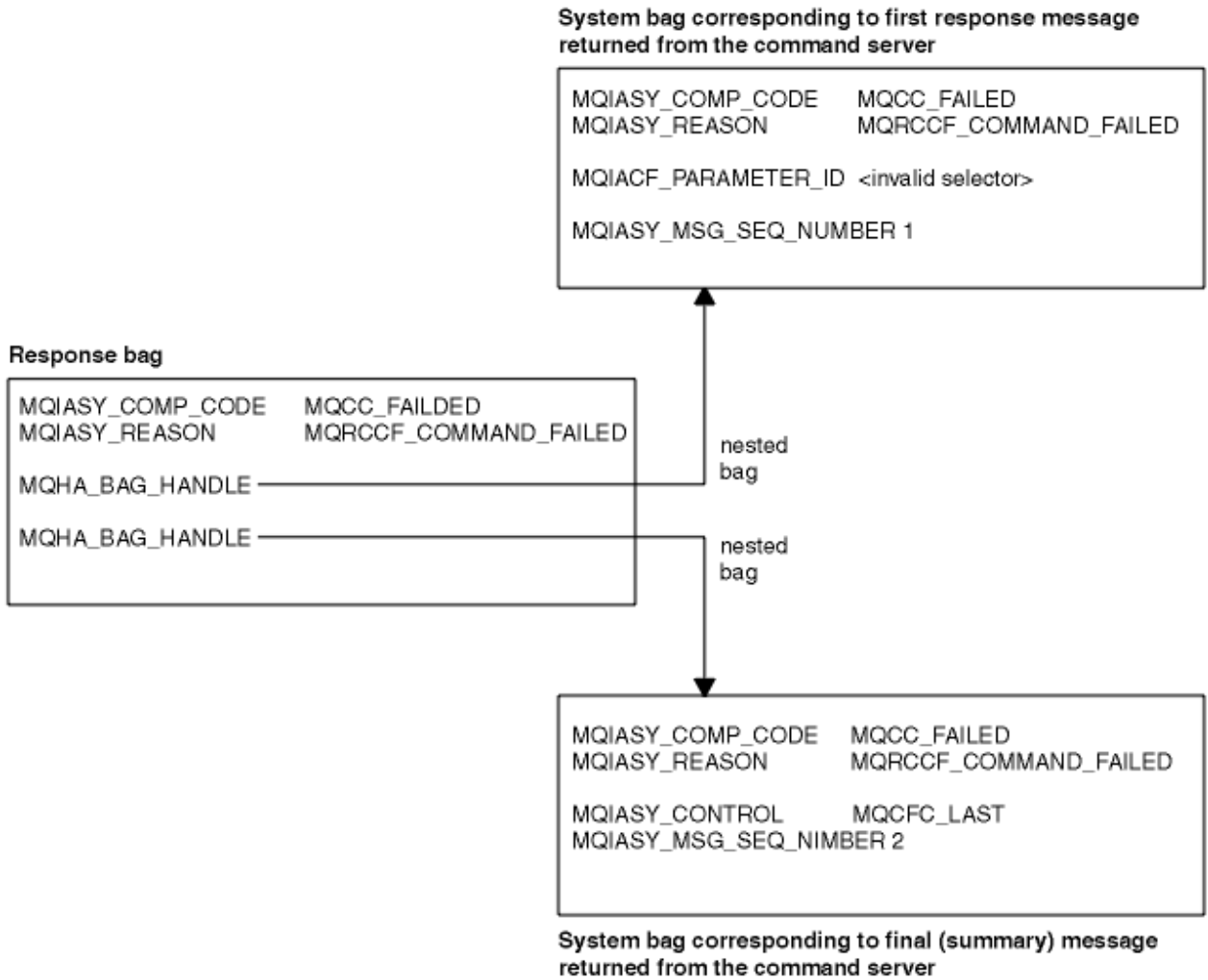
IBM MQ Denetim Arabirimi (MQAI), komut sunucusunun kendisiyle doğrudan ilgilenmek yerine, denetim komutlarını komut sunucusuna göndermek için PCF iletimlerini kullanır. Burada, MQAI olanağını kullanarak IBM MQ ' un yapılandırılmasına ilişkin bazı ipuçları bulunur.

- IBM MQ içindeki karakter dizgileri, sabit uzunluğa kadar doldurulmuyorlar. C kullanarak, boş karakterle 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 iç içe 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 iletisi tipi için geçerli olmadığından ve bu bilgi ayrıntısının, bir torba tanıtıcısı tarafından erişilebilen iç içe 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 67](#)

Aşağıdaki çizge bu senaryoyu göstermektedir:



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. Dizinlemeye ilişkin tüm ayrıntılar [“MQAI içinde dizin oluşturma” sayfa 35](#) içinde bulunabilir.

- Veri dönüştürme

Bir MQAI veri torbasında bulunan dizgiler çeşitli kodlanmış karakter kümelerinde olabilir ve bunlar mqSetTamsayı çağrısı kullanılarak dönüştürülebilmektedir. Veri dönüştürmeye ilişkin tüm ayrıntılar [“MQAI ' de veri dönüştürme işlemi” sayfa 36](#) içinde bulunabilir.

- İleti tanımlayıcısının kullanımı

MQAI, veri torbası yaratıldığında bir başlangıç değeri olarak ayarlanmış bir ileti tanımlayıcısı oluşturur. İleti tanımlayıcısının kullanımına ilişkin tüm ayrıntılar “MQAI 'daki ileti tanımlayıcısının kullanılması” sayfa 37’inde bulunabilir.

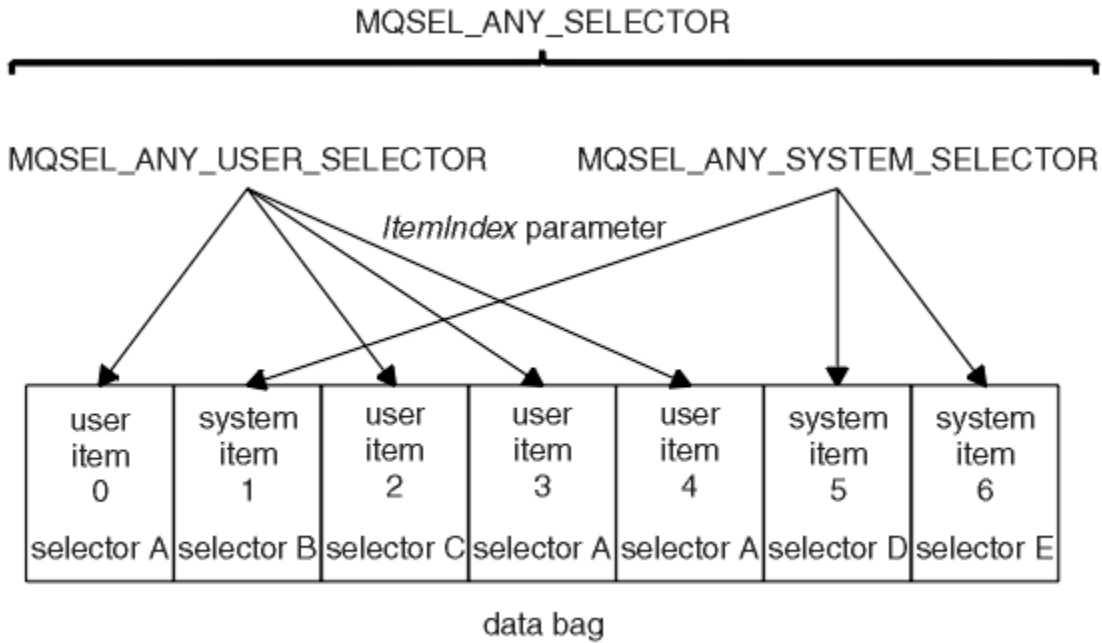
MQAI içinde dizin oluşturma

Dizinler, var olan veri öğelerini bir çantadan değiştirirken ya da bir çantadan çıkarırken kullanılır. Üç tip dizin oluşturma vardır; bu, 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 görelî dizin.
- Öğenin ait olduğu seçici (kullanıcı ya da sistem) kategorisine göre dizin.
- Çantadaki (kullanıcı ve sistem) tüm veri öğelerle görelî dizin.

Bu, kullanıcı seçicileri, sistem seçicileri ya da her ikisi tarafından Şekil 4 sayfa 35’ ta gösterildiği şekilde dizinlemeye olanak sağlar.



Şekil 4. Dizin oluşturma

Şekil Şekil 4 sayfa 35’ de, kullanıcı öğesi 3 (seçici A) aşağıdaki dizin çiftleri tarafından başvurulabilir:

Selector	ItemIndex
seçici A	1
MQSEL_ANY_USER_SELECTOR	2
MQSEL_ANY_SELECTOR	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. Bunun örnekleri için [Poşet içindeki bilgilerin değiştirilmesi](#) ve [Veri öğelerinin silinmesi](#) başlıklı konuya bakın.

Üç tip dizin oluşturma, veri öğelerinin kolayca alınmasına olanak sağlar. Örneğin, bir çantada belirli bir seçicinin üç eşgörünümü varsa, mqCountÖge çağırısı 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 elde edebilen öznitelikler için kullanışlıdır.

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. This is known as the *çanta CCSID 'si* and can be changed using the mqSetInteger call.

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

Çizelge 2 sayfa 36 , veri torbalarını iletilere dönüştürürken uygulanan kuralları gösterir (tam tersi de geçerlidir):

<i>Çizelge 2. CCSID işleme</i>			
MQAI çağrısı	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
mqBufferToBag	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	İletinin 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

Çizelge 2. CCSID işleme (devamı var)			
MQAI çağırısı	CCSID	Aramak için giriş	Çağrılacak çıktı
mqExecute	İstek-çanta CCSID 'si	İstek iletilisinin MQMD 'si için kullanılır (4)	Değiştirilmedi
mqExecute	Yanıt-çanta CCSID 'si	Yanıt iletilisinin veri dönüştürme için kullanılır (4)	Döndürülen veri CCSID değerine ayarla (3)
mqExecute	İstek paketindeki CCSID CCSID ' leri	İstek iletilisi için kullanılır	Değiştirilmedi
mqExecute	Yanıt paketindeki CCSID CCSID ' leri	Burada geçerli değil	Yanıt iletilisinde 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 bilgiler

Veri dönüştürme

ccsid_part2.tbl dosyası

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 ögesinden alınır. Veri çantanızı oluşturduğunuzda, bu ögenin başlangıç değeri, oluşturduğunuz torba tipine bağlı olarak ayarlanır:

Çizelge 3. PCF komut tipi	
Çanta tipi	MQIASY_TYPE ögesinin 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 3 sayfa 37](#).

Çizelge 4. MQMD ' nin biçimi ve MsgType deęiřtirgeleri		
PCF komut tipi	Biçim	MsgType
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

Çizelge 4. MQMD ' nin biçimi ve MsgType değıştirgeleri (devamı var)		
PCF komut tipi	Biçim	MsgType
MQCFT_*	MQFMT_PCF	MQMT_DATAGRAM

Çizelge 4 sayfa 37 , 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 diğer parametreler, Çizelge 5 sayfa 38 içinde gösterilen değerleri alır.

Çizelge 5. İleti tanımlayıcı değerleri	
Değiştirge	Değer
<i>StrucId</i>	MQMD_STRUC_ID
<i>Version</i>	MQMD_VERSION_1
<i>Report</i>	MQRO_NONE
<i>MsgType</i>	Bkz. Çizelge 4 sayfa 37
<i>Expiry</i>	30 saniye (not “1” sayfa 38)
<i>Feedback</i>	MQFB_YOK
<i>Encoding</i>	MQENC_NATIVE
<i>CodedCharSetId</i>	Torba CCSID değerine bağlıdır (not “2” sayfa 38)
<i>Format</i>	Bkz. Çizelge 4 sayfa 37
<i>Priority</i>	MQPRI_PRIORITY_AS_Q_DEF
<i>Persistence</i>	MQPER_NOT_PERSISTENT
<i>MsgId</i>	MQMI_NONE
<i>CorrelId</i>	MQCI_NONE
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	bkz. not “3” sayfa 38
<i>ReplyToQMGr</i>	blank (boşluk)

Notlar:

1. Bu değer, **OptionsBag** parametresi kullanılarak mqExecute çağrısında geçersiz kılınabilir. Bu konuda bilgi için bkz. mqExecute.
2. Bkz. “MQAI ' de veri dönüştürme işlemi” sayfa 36.
3. MQMT_REQUEST tipinde iletiler için, kullanıcı tarafından belirtilen yanıt kuyruğunun ya da MQAI tarafından oluşturulan geçici dinamik kuyruğun adı. Aksi takdirde boştur.

Yerel kuyruk yaratmak için Örnek C programı (amqsaicq.c)

The sample C program amqsaicq.c creates a local queue using the MQAI.

```

/*****
/*
/* 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, 2023.        */
/*                                                      */
/*****
/* 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", 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);
}
}

```

Olay izleyiciyi kullanarak olayları görüntülemek için Örnek C programı (amqsaie.m.c)

Örnek C programı amqsaie.m.c , MQAI olanağını kullanarak bir temel 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, 2023. 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)
        strncpy(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 */
    /* embedded bag display */

    /*****
    /* 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         */
                                                /* indent display                */

/*****
/* Count the number of items in the bag
*****/
mqCountItems(dataBag, MQSEL_ALL_SELECTORS, &itemCount, &compCode, &reason);

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

/*****
/* If no errors found, display each item in the bag
*****/
if (!errors)
{
    for (i = 0; i < itemCount; i++)
    {
        /*****
        /* First inquire the type of the item for each item in the bag
        *****/
        mqInquireItemInfo(dataBag, /* Bag handle
                               MQSEL_ANY_SELECTOR, /* Item can have any selector
                               i, /* Index position in the bag
                               &selector, /* Actual value of selector
                               /* returned by call
                               &itemType, /* Actual type of item
                               /* returned by call
                               &compCode, /* Completion code
                               &reason); /* Reason Code

        if (compCode != MQCC_OK)
            errors++;

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

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

        case MQITEM_INTEGER64:
            /*****
            /* Item is a 64-bit integer. Find its value and display its
            /* index, selector and value.
            *****/
            mqInquireInteger64(dataBag, /* Bag handle

```

```

                                MQSEL_ANY_SELECTOR, /* Allow any selector */
                                i, /* Index position in the bag */
                                &i64Value, /* Returned integer value */
                                &compCode, /* Completion code */
                                &reason); /* Reason Code */

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

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

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

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

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

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

```



```

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

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

```

Kanal nesnelerinin araştırılmasına ilişkin örnek C programı (amqsaicl.c)

Örnek C programı amqsaicl.c, MQAI olanağını kullanarak kanal nesnelerini sorgular.

```

/******
/*
/* 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, 2023. 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 */
    "clusdr   "  /* MQCHT_CLUSSDR */
};

```

```

"clussdr" /* MQCHT_CLUSSDR */
};
#endif

/*****
/* Macros
*****/
#if (MQAT_DEFAULT == MQAT_OS400)
#define OUTFILE "QTEMP/AMQSAICL(AMQSAICL)"
#define OPENOUTFILE(hdl, fname) \
    (hdl) = _Ropen((fname), "wr", rtrncode=Y");
#define CLOSEOUTFILE(hdl) \
    _Rclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    _Rwrite((hdl), (buf), (buflen));
#elif (MQAT_DEFAULT == MQAT_UNIX)
#define OUTFILE "/tmp/amqsaicl.txt"
#define OPENOUTFILE(hdl, fname) \
    (hdl) = fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
    fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));
#else
#define OUTFILE "amqsaicl.txt"
#define OPENOUTFILE(fname) \
    fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
    fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));
#endif

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

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

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

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

    /*****

```

```

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

if(outfp == NULL)
{
    printf("Could not open output file.\n");
    goto MOD_EXIT;
}
/*****
/* Create an admin bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &adminBag;, &compCode;, &reason;);
CheckCallResult("Create admin bag", compCode, reason);

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

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

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

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

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

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

/*****
/* Check the result from mqExecute call. If successful find the channel */
/* types for all the channels. If failed find the error. */
/*****
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
    /*****
    /* Count the number of system bags embedded in the response bag from the */
    /* mqExecute call. The attributes for each channel are in separate bags. */

```

```

/*****
mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags,
              &compCode, &reason);
CheckCallResult("Count number of bag handles", compCode, reason);

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

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

/*****
/* Get the channel type out of the channel attributes bag */
/*****
mqInquireInteger(cAttrsBag, MQIACH_CHANNEL_TYPE, MQIND_NONE, &chlType,
                 &compCode, &reason);
CheckCallResult("Get type", compCode, reason);

/*****
/* Use mqTrim to prepare the channel name for printing. */
/* Print the result. */
/*****
mqTrim(MQ_CHANNEL_NAME_LENGTH, chlName, chlName, &compCode, &reason);
sprintf(OutputBuffer, "%-20s%-9s", chlName, ChlType2String(chlType));
WRITEOUTFILE(outfp, OutputBuffer, 29)
}
}

else /* Failed mqExecute */
{
printf("Call to get channel attributes failed: Cc = %ld : Rc = %ld\n",
       compCode, reason);
/*****
/* If the command fails get the system bag handle out of the mqexecute */
/* response bag. This bag contains the reason from the command server */
/* why the command failed. */
/*****
if (reason == MQRCCF_COMMAND_FAILED)
{
mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag,
             &compCode, &reason);
CheckCallResult("Get the result bag handle", compCode, reason);

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

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

```

```

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

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

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

return 0;
}

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

```

Kuyruklar ve yazdırma bilgileri (amqsailq.c) hakkında araştırma yapmak için C programı örneği

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

```

/*****
/*
/* 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, 2023. */
/*
/*****
/*
/* 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(MQCBO_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(qAttrsBag, MQIA_CURRENT_Q_DEPTH, MQIND_NONE, &qDepth,
                    &compCode, &reason);
    CheckCallResult("Get depth", compCode, reason);

    /******
    /* Use mqTrim to prepare the queue name for printing. */
    /* Print the result. */
    /******
    mqTrim(MQ_Q_NAME_LENGTH, qName, qName, &compCode, &reason)
    printf("%4d %-48s\n", qDepth, qName);
  }
}

else                                                     /* Failed mqExecute */
{
  printf("Call to get queue attributes failed: Completion Code = %d :
        Reason = %d\n", compCode, reason);

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

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

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

```

```

    CheckCallResult("Delete the admin bag", compCode, reason);
}

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

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

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

```

Veri torbaları ve MQAI

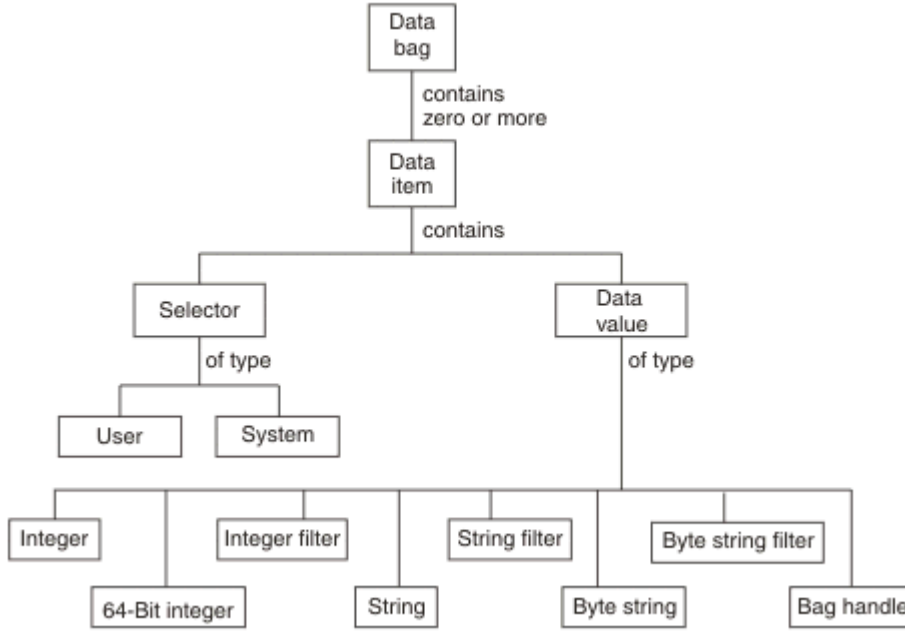
Veri torbası, IBM MQ Denetim Arabirimi 'ni (MQAI) kullanarak nesnelerin özelliklerini ya da değiştiremelerini işlemek için kullanılan bir araçtır.

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 61’indeki ayrıntılarda açıklanır.

İki tip seçici vardır; *kullanıcı seçicileri* ve *sistem seçicileri*. Bunlar [MQAI Selectors](#) içinde 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 35’inde açıklanmıştır.

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



Şekil 5. 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 60](#) içinde açıklandığı 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 60](#).

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 iletisi 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:

- Veri torbaları [“Veri torbalarının yaratılması ve silinmesi” sayfa 60](#) yaratabilir ve silebilirsiniz.
- You can send data between applications using data bags [“MQAI kullanılarak veri torbalarının alınması ve alınması” sayfa 60](#).

- Veri çantalarına [“Paketlere MQAI ile veri öğeleri eklenmesi”](#) sayfa 61 veri öğeleri ekleyebilirsiniz.
- [“Bir çantaya sorgu komutu eklenmesi”](#) sayfa 62 veri torbası içine bir sorgu komutu ekleyebilirsiniz.
- You can inquire within data bags [“Veri torbaları araştırılıyor”](#) sayfa 63.
- Veri öğelerini bir veri torbası [“Veri öğelerini sayma”](#) sayfa 65 içinde sayabilirsiniz.
- Bir veri torbası [“Bir çantadaki bilgilerin değiştirilmesi”](#) sayfa 63 içindeki bilgileri değiştirebilirsiniz.
- Bir veri torbasını temizleyebilirsiniz [“Clearing a bag using the mqClearBag call”](#) sayfa 64.
- [“Truncating a bag using the mqTruncateBag call”](#) sayfa 65 veri torbasını kesebilirsiniz.
- Torbaları ve arabellekleri ([“Çantalar ve arabellekler dönüştürülüyor”](#) sayfa 65) dönüştürebilirsiniz.

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 kullanılacak veri öğesi tipleri”](#) sayfa 61.
- 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 67. Kullanıcı kodu bir sistem torbasını silemiyor.

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

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 23 başlıklı konuya bakın.

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

MQAI içinde kullanılacak 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 öznitelikleri 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 61.](#)

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 kullanılacak 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 65.](#)
- [“Veri öğelerinin silinmesi” sayfa 66.](#)
- [“Paketlere MQAI ile veri öğeleri eklenmesi” sayfa 61.](#)
- [“Veri öğelerini süzme ve sorgulama” sayfa 62.](#)

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.

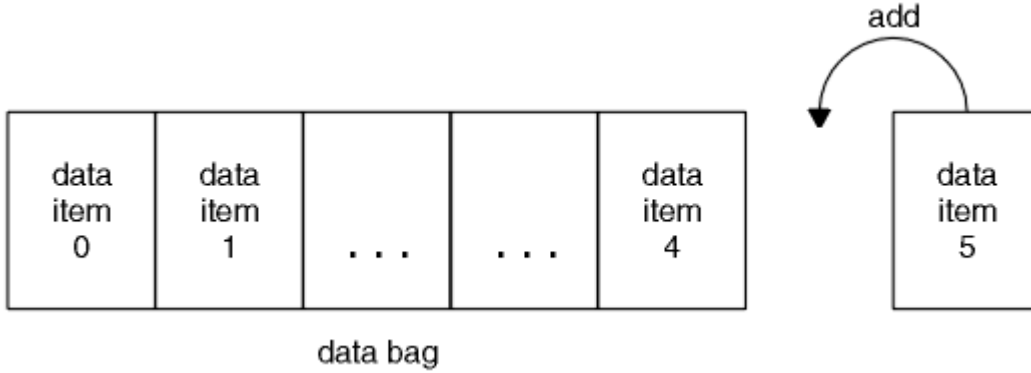
Çoğu sistem öğesi değiştirilebilir (bkz. [“Bir çantadaki bilgilerin değiştirilmesi” sayfa 63](#)), ancak çanta-yaratma seçenekleri kullanıcı tarafından değiştirilemez. Sistem öğelerini silemezsiniz. (Bkz. [“Veri öğelerinin silinmesi” sayfa 66.](#))

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 kullanılabilir veri öğesi tipleri” sayfa 61.

MQAI, çok sayıda tamsayı öğesi, 64 bit tamsayı öğeleri, tamsayı süzgeç öğeleri, karakter dizilimi öğeleri, dizgi süzgeci, bayt dizgi öğeleri ve byte dizgi süzgeç öğelerini paketlere eklemenizi sağlar ve Şekil 6 sayfa 62’inde gösterilir. Öğeler bir seçiciyle tanıtılır. 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 6. 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](#)’ında 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 61.

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](#).

Veri öğelerini süzme ve sorgulama

MQAI olanağını kullanırken, IBM MQ nesnelерinin özniteliklerini 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 kuyrukta) öznitelikleriyle eşleşmesi gerektiğini belirtir.

Süzülebilecek diğer öznitelikler, "[IBM MQ Programlanabilir Komut Biçimlerine Giriş](#)" sayfa 20 içinde bulunabilecek PCF sorgulamaya * komutlarına karşılık gelir. Örneğin, bir kanalın özniteliklerini 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 özniteliklerine sahip olabilirsiniz. Bu, ilgilendiğiniz seçiciyi belirtir. Seçiciyi belirtmezseniz, nesnenin tüm öznitelikleri döndürülür.

Aşağıda, bir kuyruğun özniteliklerini 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, ...)
```

Veri torbaları araştırılıyor

Aşağıdaki bilgileri araştırabilirsiniz:

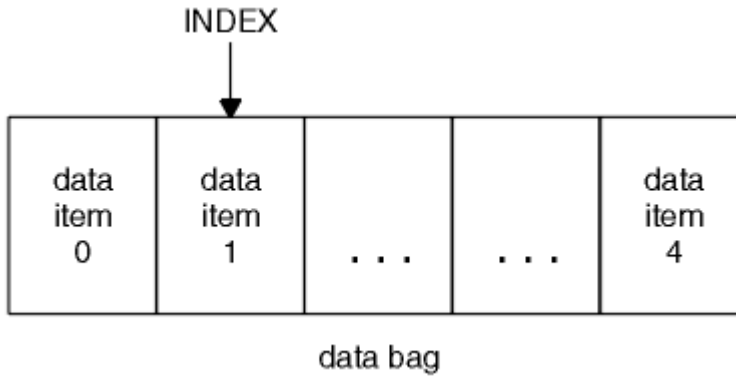
- The value of an integer item using the mqInquireInteger call. Bkz. [mqInquireInteger](#).
- mqInquireInteger64 çağrısını kullanan 64 bitlik bir tamsayı ögesinin değeri. Bkz. [mqInquireInteger64](#).
- mqInquireIntegerFilter çağrısını kullanan bir tamsayı süzgeci ögesinin değeri. Bkz. [mqInquireIntegerFilter](#).
- mqInquireDizgi çağrısını kullanan bir karakter dizilimi ögesinin değeri. Bkz. [mqInquireDizgisi](#).
- 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](#).

Bir çantadaki bilgilerin değiştirilmesi

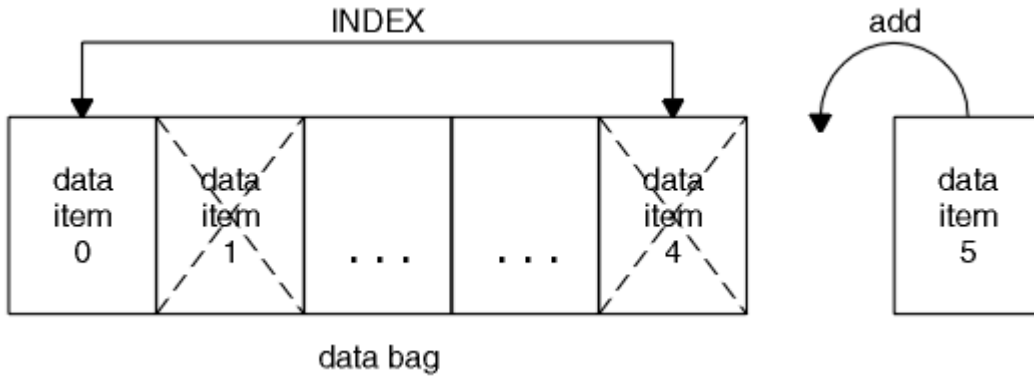
MQAI, mqSet* çağrılarını kullanarak bir çanta içindeki bilgileri değiştirmenize olanak sağlar. Yapabilecekleriniz:

1. Bir çantadaki veri ögelerini değiştirin. Dizin, değiştirilecek ögenin oluşumunun belirlenerek bir parametrenin tek tek bir eşgörünümünün değiştirilmesine olanak sağlar (bkz. [Şekil 7 sayfa 64](#)).



Şekil 7. Tek bir veri ögesinin değiştirilmesi

2. Belirtilen seçicinin var olan tüm oluşumlarını silin ve çantanın sonuna yeni bir geçiş ekleyin. (Bkz. Şekil 8 sayfa 64.) Özel bir dizin değeri, bir değiştirgenin **Tümü** yönetim ortamlarının değiştirilmesine olanak sağlar.



Şekil 8. Tüm veri ögelerini değiştirme

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

The `mqSetInteger` call lets you modify integer items within a bag. `mqSetInteger64` çağrısı, 64 bitlik tamsayı ögelerini değiştirmenize olanak sağlar. `mqSetIntegerFilter` çağrısı, tamsayı süzgeci ögelerini değiştirmenize olanak sağlar. `mqSetDizgi` çağrısı, karakter dizilimi olan ögeleri değiştirmenize olanak sağlar. `mqSetStringFilter` çağrısı, dizgi süzgeci ögelerini değiştirmenize olanak sağlar. `mqSetByteString` çağrısı, bayt dizgi ögelerini değiştirmenize olanak sağlar. `mqSetByteStringFilter` çağrısı, bayt dizgi süzgeci ögelerini değiştirmenize olanak sağlar. Diğer bir seçenek olarak, bu çağrılar, belirtilen seçicinin varolan tüm oluşumlarını silmek ve çantanın sonuna yeni bir geçiş eklemek için kullanabilirsiniz. Veri ögesi bir kullanıcı ögesi ya da bir sistem ögesi olabilir.

Bu aramaların tam açıklaması için bkz.

- [mqSetTamsayı](#)
- [mqSetInteger64](#)
- [mqSetIntegerFilter](#)
- [mqSetDizgisi](#)
- [mqSetStringFilter](#)
- [mqSetByteString](#)
- [mqSetByteStringSüzgeci](#)

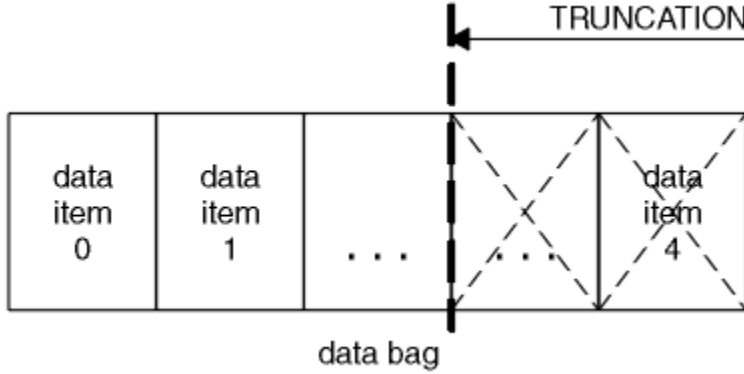
Clearing a bag using the `mqClearBag` call

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

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

Truncating a bag using the mqTruncateBag call

mqTruncateBag çağrısı, bir kullanıcı torbasındaki kullanıcı öğelerinin sayısını, en son eklenen parçadan başlayarak, torba sonundan itibaren silerek kullanıcı torbalarındaki sayısını azaltır. Örneğin, birden fazla ileti oluşturmak için aynı üstbilgi bilgileri kullanıldığında bu kullanılabilir.

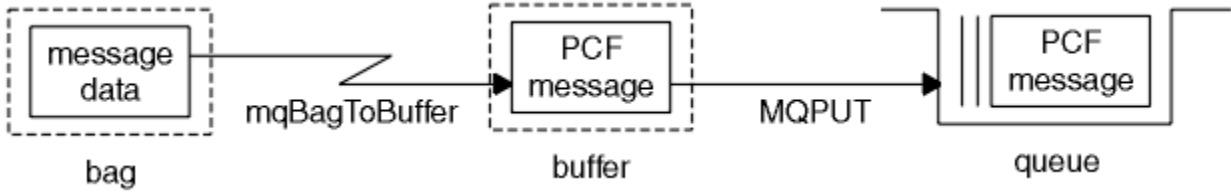


Şekil 9. Çantanın kesilmesi

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

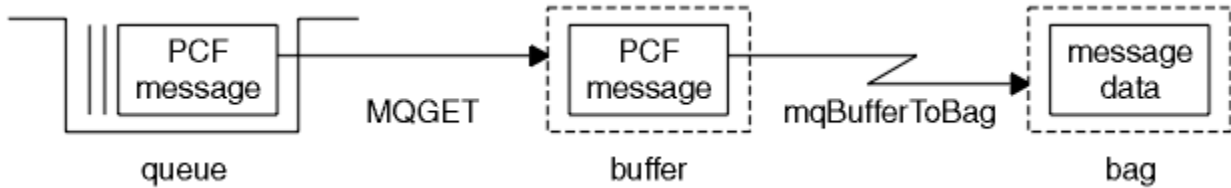
Çantalar ve arabellekler dönüştürülüyor

Uygulamalar arasında veri göndermek için, ilk olarak ileti verileri bir torbaya yerleştirilir. Daha sonra, çantadaki veriler mqBagToBuffer çağrısını kullanarak bir PCF iletisine dönüştürülür. PCF iletisi, MQPUT çağrısını kullanarak gerekli kuyruğa gönderilir. Bu, Şekil Şekil 10 sayfa 65 içinde gösterilir. mqBagToBuffer çağrısının tam açıklaması için bkz. [mqBagToBuffer](#).



Şekil 10. Çantaların PCF iletilerine dönüştürülmesi

Verileri almak için, ileti MQGET çağrısını kullanarak bir arabelleğe alınır. Arabelleğindeki veriler, mqBufferToBag çağrısını kullanarak bir torbaya dönüştürüle; arabelleğin geçerli bir PCF iletisi içermesi gerekir. Bu, Şekil Şekil 11 sayfa 65 içinde gösterilir. mqBufferToBag çağrısının tam açıklaması için bkz. [mqBufferToBag](#).



Şekil 11. PCF iletileri torba forma dönüştürülüyor

Veri öğelerini sayma

mqCountÖge 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 ayrıncı 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.

mqCountÖğeleri çağrısına ilişkin tam açıklama için bkz. [mqCountÖğeleri](#).

Veri öğelerinin silinmesi

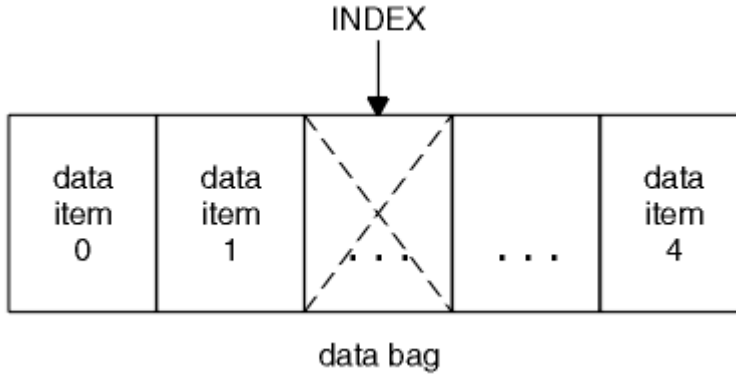
Çantalardaki öğeleri çeşitli şekillerde silebilirsiniz. Yapabilecekleriniz:

- Bir çantadan bir ya da daha fazla kullanıcı öğesini kaldırın. Ayrıntılı bilgi için bkz. [“mqDeleteÖğe çağrısını kullanarak bir çantadan veri öğelerinin silinmesi”](#) sayfa 66.
- **Tümü** kullanıcı öğelerini bir çantadan, yani *temizle* bir torbadan silin. Ayrıntılı bilgi için bkz. [“Clearing a bag using the mqClearBag call”](#) sayfa 64.
- Bir çantanın sonundaki kullanıcı öğelerini silin; bu, bir torbayı *kestir* . Ayrıntılı bilgi için bkz. [“Truncating a bag using the mqTruncateBag call”](#) sayfa 65.

mqDeleteÖğe çağrısını kullanarak bir çantadan veri öğelerinin silinmesi

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

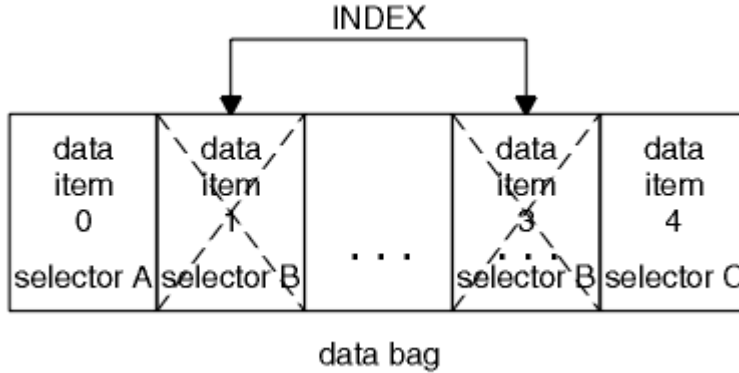
1. Belirtilen seçiciye ilişkin tek bir oluşum. (Bkz. [Şekil 12 sayfa 66.](#))



Şekil 12. Tek bir veri öğesinin silinmesi

ya da

2. Belirtilen seçicinin tüm oluşumları. (Bkz. [Şekil 13 sayfa 66.](#))



Şekil 13. Tüm veri öğeleri siliniyor

Not: Dizin, çantanın içinde ekleme sırasını korur, ancak diğer veri öğelerinin dizinlerini etkileyebilir. For example, the mqDeleteItem call does not preserve the index values of the data items that follow the deleted item because the indices are reorganized to fill the gap that remains from the deleted item.

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

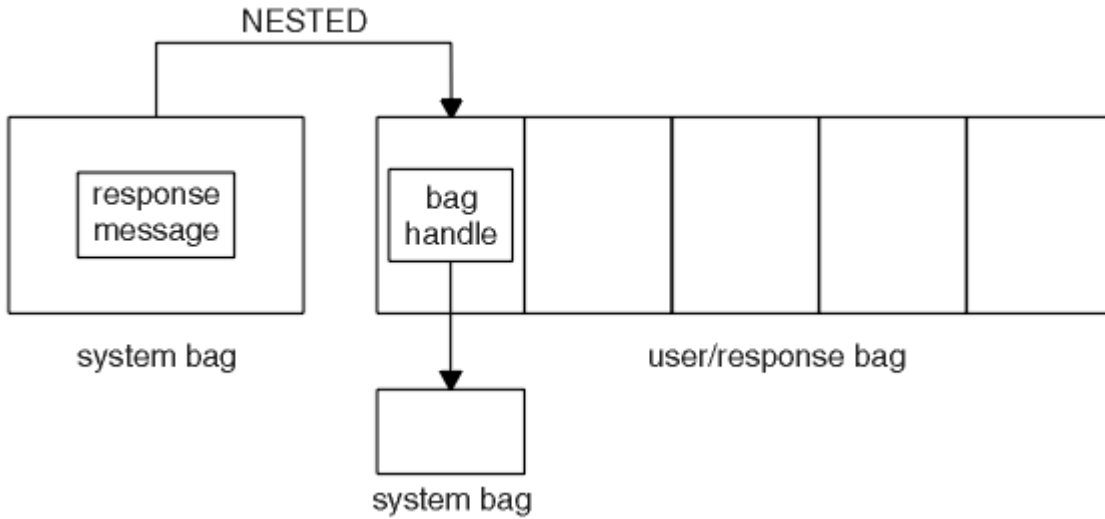
Sending administration commands to the qm command server using the mqExecute call

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 14 sayfa 67’inde *iç içe yerleştirme* kavramı gösterilir.



Şekil 14. 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 kodubaşlıklı](#) konuya bakın.

V 9.0.1 REST APIkomutunu kullanarak yönetim

Kuyruk yöneticileri ve kuyruklar [V 9.0.5](#) ve Managed File Transfer GET REST hizmetlerigibi IBM MQ nesnelere yönetmek için administrative REST API ' i kullanabilirsiniz. Bilgi, JSON biçimindeki 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](#).

Not:

IBM MQ 9.0.1' ta administrative REST API , IBM MQ güvenliği ile bütünleştirilmedi. Bu nedenle, administrative REST API varsayılan değer olarak geçersiz kılınmaktadır. You must manually enable the administrative REST API before you can use it. administrative REST API' in etkinleştirilmesiyle ilgili daha fazla bilgi için bkz. [“administrative REST APIile çalışmaya başlama”](#) sayfa 68.

[V 9.0.2](#) IBM MQ 9.0.2' tan administrative REST API , IBM MQ güvenliği ile bütünleştirilmiştir. administrative REST API varsayılan olarak etkindir. Ancak, administrative REST APIolanağını kullanabilmek için güvenliği yapılandırmanız gerekir. Daha fazla bilgi için [IBM MQ Console ve REST API security](#) başlıklı konuya bakın.

Yordam

- [“administrative REST APIile çalışmaya başlama”](#) sayfa 68
- [“administrative REST APIkomutunu kullanma”](#) sayfa 74
- [“REST APIkomutunu kullanarak uzaktan yönetim”](#) sayfa 80
- [“REST API zaman damgaları”](#) sayfa 84
- [“REST API Hata işleme”](#) sayfa 84
- [“REST API Keşif”](#) sayfa 86
- [“REST API ulusal dil desteği”](#) sayfa 88

V 9.0.1 administrative REST APIile çalışmaya başlama

administrative REST API 'u başlatmadan önce doğru bileşenleri kurmalı, REST API' i etkinleştirmeli, güvenliği yapılandır seçeneğini etkinleştirmeli ve mqweb sunucusunu başlatmalısınız.

Başlamadan önce

[IBM i](#) IBM üzerinde, komutların QSHELL içinde çalıştırılması gerekir.

Bu görev hakkında

Bu göreve ilişkin yordam, administrative REST APIile hızlı bir şekilde başlatılmaya odaklanır. Güvenlik anahattını yapılandırma adımları, temel kullanıcı kayıt dosyasını nasıl ayarlayacağını, ancak kullanıcıların ve rollerin yapılandırılmasına ilişkin diğer seçenekleri içerir. administrative REST APIiçin güvenlik yapılandırması hakkında daha fazla bilgi için bkz. [IBM MQ Console ve REST API security](#).

Not: mqwebuser.xml dosyasına erişmek için bir [ayrıcalıklı kullanıcı](#) olmanız gerekir.

Yordam

1. IBM MQ Console ve REST API bileşenini kurun:

- **V 9.0.4** **AIX** AIX' ta `mqm.web.rte` kütük kümesini kurun.
- **Linux** Linux' ta MQSeriesWeb bileşenini kurun. For more information about installing components and features on Linux, see [Linux kuruluş görevleri](#).
- **Windows** Windows' ta Web Administration özelliğini kurun. For more information about installing components and features on Windows, see [Windows kuruluş görevleri](#).
- **z/OS** z/OS' ta IBM MQ for z/OS Unix System Services Web Components özelliğini kurun. For more information about installing components and features on z/OS, see [z/OS kuruluş görevleri](#).

2. İsteğe bağlı: IBM MQ 9.0.2' dan, administrative REST API varsayılan olarak etkindir. Ancak, IBM MQ 9.0.1' ta administrative REST API varsayılan olarak devre dışıdır. If you are using IBM MQ 9.0.1, enable the REST API by adding the following XML to the `mqwebuser.xml` file, within the `<server>` tags:

```
<variable name="mqRestAutostart" value="true"/>
```

`mqwebuser.xml` , aşağıdaki dizinlerden birinde bulunabilir:

- **ULW** `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
- **z/OS** `WLP_user_directory/servers/mqweb`

where `WLP_user_directory` is the directory that was specified when the `crtmqweb.sh` script ran to create the mqweb server definition.

3. **V 9.0.2**

From IBM MQ 9.0.2, security is enabled for the administrative REST API. administrative REST API programını kullanabilmeniz için önce kullanıcıları ve rolleri yapılandırmanız gerekir:

- a) `basic_registry.xml` dosyasını `MQ_INSTALLATION_PATH/web/mq/samp/configuration` dizininden kopyalayın.
- b) Örnek XML dosyasını uygun dizine yerleştirin:

- **ULW** UNIX, Linux, and Windows üzerinde: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
- **z/OS** z/OS üzerinde: `WLP_user_directory/servers/mqweb`

where `WLP_kullanici_dizini` is the directory that was specified when the `crtmqweb.sh` script ran to create the mqweb server definition.

- c) Örnek XML dosyasını `mqwebuser.xml` olarak yeniden adlandırın.

Not: Yeniden adlandırılan bu dosya, IBM MQ Console için de kullanılan var olan bir dosyanın yerini alır. Therefore, if you changed the `mqwebuser.xml` file for the IBM MQ Console, copy your changes to the new XML file before you rename it.

- d) İsteğe bağlı: Kullanıcılar ve gruplar eklemek için `mqwebuser.xml` dosyasını düzenleyin. Assign those users and groups appropriate roles to be authorized to use the administrative REST API. Ayrıca, varsayılan olarak tanımlanan kullanıcıların parolalarını değiştirebilir ve yeni parolaların kodlanabileceğini de yapabilirsiniz. Daha fazla bilgi için [Kullanıcıları ve rolleri yapılandırma başlıklı](#) konuya bakın.

4. `mqweb` sunucusu ile uzak bağlantıları etkinleştir:

- From version 9.0.4, use the `setmqweb` command:
`setmqweb properties -k httpHost -v hostname`
- From version 9.0.1, add the following XML to the `mqwebuser.xml` file, within the `<server>` tags:

```
<variable name="httpHost" value="hostname" />
```

Burada *anasistem* , etki alanı adı sonekine sahip IP adresini, etki alanı ad sunucusunu (DNS) anasistem adını ya da IBM MQ ' un kurulu olduğu sunucunun DNS ana makine adını belirtir. Kullanılabilir tüm ağ arabirimlerini belirlemek için yıldız imi (*) kullanın.



Uyarı: z/OS V 9.0.4

z/OS' ta **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, değişken noktalarını mqweb sunucusu yapılandırmanıza gösteren WLP_USER_DIR ortam değişkenini ayarlamamız gerekir.

Bunu yapmak için şu komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

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

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

Ek bilgi için [Liberty sunucusu tanımlamasının yaratılması](#) başlıklı konuya bakın.

5. REST API' u destekleyen mqweb sunucusunu başlatın:

- **Linux** **Windows** Windows ve Linux üzerinde, [ayrıcılık kullanıcı](#) olarak komut satırına aşağıdaki komutu girin:
startmqweb
- **z/OS** z/OS' ta, [Görev 29: IBM WLP sunucusu için bir yordam oluşturma](#) içinde oluşturduğunuz yordamı başlatın.

Sonraki adım

1. **V 9.0.2** administrative REST API kullanıcısının mqweb sunucusu ile nasıl kimlik doğrulaması gerçekleştireceğini seçin. Aynı yöntemi tüm kullanıcılar için kullanmanıza gerek yoktur. Seçenekler şunlardır:
 - HTTP temel kimlik doğrulamasını kullanarak kullanıcıların kimliklerini doğrulamasına izin verin. Bu durumda, bir kullanıcı adı ve parola şifrelenmez, ancak şifrelenmez ve her bir REST API isteği ile kullanıcıya bu istek için kimlik doğrulama ve yetki verme isteği gönderilir. Bu kimlik doğrulamasının güvenli olması için güvenli bir bağlantı kullanmanız gerekir. Yani, HTTPS kullanmanız gerekir. Daha fazla bilgi için [REST API ile HTTP temel kimlik doğrulamasının kullanılması](#) başlıklı konuya bakın.
 - Belirteç kimlik doğrulamasını kullanarak kullanıcıların kimliklerini doğrulamasına izin verin. Bu durumda, kullanıcı, HTTP POST yöntemiyle REST API login kaynağına bir kullanıcı kimliği ve parola sağlar. Kullanıcının oturum açmış durumda kalmasını ve belirli bir süre için yetkilendirilmesini sağlayan bir LTPA belirteci oluşturulur. Bu kimlik doğrulamasının güvenli olması için güvenli bir bağlantı kullanmanız gerekir. Yani, HTTPS kullanmanız gerekir. Daha fazla bilgi için bkz. [Using token based authentication with the REST API](#).
 - İstemci sertifikalarını kullanarak kullanıcıların kimliklerini doğrulamasına izin verin. Bu durumda, kullanıcı administrative REST API' ta oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak istemci sertifikasını kullanır. Daha fazla bilgi için [REST API ile istemci sertifikası kimlik doğrulaması kullanma](#) başlıklı konuya bakın.
2. **V 9.0.1** HTTP bağlantılarının etkinleştirilmesi ve kapı numarasının değiştirilmesi de içinde olmak üzere REST API ayarlarını yapılandırın. Daha fazla bilgi için [IBM MQ Console ve REST API' in yapılandırılması](#) başlıklı konuya bakın.
3. **V 9.0.5** Optionally, configure the REST API for MFT. Daha fazla bilgi için, bkz. [“MFT REST API için gerekli yapılandırma”](#) sayfa 71.
4. **V 9.0.2** İsteğe bağlı olarak, REST API için Cross Origin Resource Sharing özelliğini yapılandırabilirsiniz. Varsayılan olarak, REST API ile aynı etki alanında barınmayan web kaynaklarından

REST API' a erişemezsiniz. Yani, çapraz başlangıç istekleri etkinleştirilmez. Belirtilen URL 'lerden gelen çapraz kaynak isteklerine izin vermek için Çapraz Kaynak Paylaşımı Paylaşımını (CORS) yapılandırabilirsiniz. Daha fazla bilgi için bakınız: [Configuring CORS for the REST API](#).

5. REST API' yi kullanın. Daha fazla bilgi için bkz. “[administrative REST API komutunu kullanma](#)” sayfa 74, “[REST API komutunu kullanarak uzaktan yönetim](#)” sayfa 80, ve [Yönetimsel REST API başvurusu](#).

Not: `endmqweb` komutunu kullanarak, mqweb sunucusunu istediğiniz zaman durdurabilirsiniz. Ancak, mqweb sunucusu çalışmıyorsa, REST API ya da IBM MQ Console' ı kullanamazsınız.

V 9.0.5 MFT REST API için gerekli yapılandırma

REST API'ı kullanmak üzere Managed File Transfer ' ı yapılandırmak için yapmanız gereken adımlar

mqwebuser.xml dosyasında şunu ayarlayın:

- `mqRestMftEnabled` özelliği *true* değerine sahip olmalıdır

Not: Bu özelliğin değerini değiştirdiğinizde, mqweb sunucusunu yeniden başlatmanız gerekir.

- `mqRestMftCoordinationQmgr` özelliği, mqweb sunucusunun çalışmakta olduğu makinede yerel olarak çalışan uygun bir MFT koordinasyon kuyruğu yöneticisidir.

Kuyruk yöneticisinin REST API için MFT eşgüdümü kuyruk yöneticisi olarak işlev görmeleri için aşağıdaki bilgileri çalıştırmanız gerekir:

- `fteSetupCoordination` command on the same local queue manager that was set for the `mqRestMftCoordinationQmgr` property in the mqwebuser.xml file.

Bu komut, IBM MQ nesnelere ilişkin tanımlamaları içeren bir MQSC dosyası oluşturur. Komutun, MFT kurulu herhangi bir makinede çalıştırılabildiği unutulmadır.

- Önceki adımda oluşturulan MQSC dosyası, gerekli IBM MQ nesnelere yaratmak için koordinasyon kuyruk yöneticisine karşı çalıştırılmalıdır.

İlgili bilgiler

[MFT REST API güvenliğinin yapılandırılması](#)

[MFT REST API' in yönetilmesi](#)

[Aktarımların GET-listesi](#)

[GET-aktarma durumu](#)

V 9.0.1 REST API URL 'sinin belirlenmesi

The default URL to access the administrative REST API at IBM MQ 9.0.4 and later is: <https://localhost:9443/ibmmq/rest/v1/admin>. IBM MQ 9.0.3 ve önceki sürümlerde varsayılan URL: <https://localhost:9443/ibmmq/rest/v1.messaging>. REST API ' a erişmek için kullanılan varsayılan URL: <https://localhost:9443/ibmmq/rest/v1/messaging>. If the host or port is changed from the default, or if HTTP is enabled, you can determine the URL by using the `dspmweb` command.

Bu görev hakkında

V 9.0.4 From IBM MQ 9.0.4, you can use the `dspmweb status` command to determine the REST API URL on Windows, Linux/ve z/OS.

For IBM MQ 9.0.3, and earlier, on Windows and Linux, you can use the `dspmweb` command. z/OS üzerinde, URL ' yi belirlemek için `messages.log` dosyasını aramanız gerekir.

Yordam

- V 9.0.4 URL adresini belirlemek için aşağıdaki yöntemlerden birini kullanın:

- IBM MQ 9.0.4' tan, **dspmweb status** komutunu ayrıcılık kullanıcı olarak kullanın:
 1. Komut satırına **strmqweb** komutunu girerek, mqweb sunucusunun çalıştığından emin olun.
 2. Komut satırındaki dspmweb status komutunu girerek URL ' yi görüntüleyin.
- IBM MQ 9.0.3 ve öncesi için, aşağıdaki yöntemlerden birini kullanın:
 - **Linux** **Windows** Windows ya da Linux üzerinde, **dspmweb** komutunu ayrıcılık kullanıcı olarak kullanın:
 1. Komut satırına **strmqweb** komutunu girerek, mqweb sunucusunun çalıştığından emin olun.
 2. Komut satırındaki dspmweb komutunu girerek URL ' yi görüntüleyin.
 - **z/OS** z/OS'ta, URL' yi bulmak için messages .log dosyasını kullanın:
 1. messages .log dosyasını açın.

The messages .log file is on the *WLP_user_directory/servers/mqweb/logs* path, where *WLP_user_directory* is the directory that was specified when the **crmqweb.sh** script ran to create the mqweb server definition.
 2. `ibmmq/rest/v1` ile biten en son CWWKT0016I ileti kodunu arayın. URL bu iletiye dahil edilir.

Örnek

The following example shows the message code CWWKT0016I with the URL as the default URL:

A CWWKT0016I: Web application available (default_host): https://localhost:9443/ibmmq/rest/v1

z/OS **V 9.0.2** mqweb sunucusunun z/OS sunucusunda doğru şekilde yapılandırıldığını kontrol etme

Steps you need to take to check that the mqweb server is correctly configured on z/OS, and to fix common configuration problems.

Yordam

1. mqweb sunucusunun başlattığı görevi denetleyin.

Aşağıdakine benzer iş günlüğü iletilerini görmemiz gerekir:

```
+CWWKE0001I: The server mqweb has been launched.
+CWWKZ0001I: Application com.ibm.mq.rest started in 1.493 seconds.
+CWWKZ0001I: Application com.ibm.mq.console started in 0.885 seconds.
+CWWKF0011I: The server mqweb is ready to run a smarter planet.
```

STDERR içinde ileti olmamalıdır

STDOUT ' ta aşağıdakine benzer iletiler olmalıdır:

```
Launching mqweb (WebSphere Application Server 17.0.0.2, WAS FOR Z/OS 17.0.0.2,
MQM MVS/ESA V9 R0.4/wlp-1.0.17.c1170220170523-1818) on IBM J9 VM,
version pmz6480sr4fp7-20170627_02 (SR4 FP7) (en_US)
[AUDIT ] CWWKE0001I: The server mqweb has been
launched.
[AUDIT ] CWWKG0028A: Processing included configuration resource: /mqm/V9R0M4/web/mq/etc/
mqweb.xml
[AUDIT ] CWWKG0028A: Processing included configuration resource:
var/mqm/mqweb904/servers/mqweb/mqwebuser.xml
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/api/docs/
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/api/explorer/
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/ibm/api/
[AUDIT ] CWWKT0016I: Web application available (default_host):
```



```

yourhost:yourport/ibm/api/explorer/
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/ibm/api/docs/subscription/websocket/
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport:9080/ibmmq/rest/v1/
[AUDIT ] CWWKZ0001I: Application com.ibm.mq.rest started in 1.493 seconds.
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/ibmmq/console/
[AUDIT ] CWWKT0016I: Web application available (default_host):
yourhost:yourport/ibmmq/console/internal/
[AUDIT ] CWWKZ0001I: Application com.ibm.mq.console started in 1.459 seconds.
[AUDIT ] CWWKF0012I: The server installed the following features:
[jsp-2.2, servlet-3.1, ssl-1.0, jndi-1.0, basicAuthenticationMQ-1.0,
apiDiscovery-1.0, localConnector-1.0, appSecurity-2.0, jaxrs-1.1,
concurrent-1.0, json-1.0, websocket-1.0, distributedMap-1.0,
applicationMonitorMQ-1.0].
[AUDIT ] CWWKF0011I: The server mqweb is ready to run a smarter planet.
[AUDIT ] MQWB2019I: MQ Console level: 9.0.4 - V904-GA904-L171016
[AUDIT ] MQWB0023I: MQ REST API level: 9.0.4 - V904-GA904-L171016

```

Notlar:

- Yordam başlatılamazsa, sorunları çözer.
- İletiler aşağıdakilere benziyorsa:

```

Web application available (default_host):
http://localhost:portnumber/ibmmq/console/

```

you have not set the **httpHost** property to allow remote connections to the mqweb server.

```
<variable name="httpHost" value="*" />
```

- httpHost** özelliğini değiştirdiyse, aşağıdakine benzer bir ileti elde edeceğiniz:

```

Web application available (default_host):
yourhost:portnumber/ibmmq/console/

```

TCP/IP yapınızıza bağlı olarak, URL olarak görüntülenen adresi kullanamayabilirsiniz.

- XML sunucusu yapılandırma dosyalarında yapılan değişiklikler birkaç saniye sonra yakalanır. mqweb sunucusunu yeniden başlatmaya gerek yoktur.
- IBM MQ Console ile bağlantı kurun.

IBM MQ Console başlığı ile görüntülenen bir pencere edinmeniz gerekir.

Notlar:

- This site can't be reached ya da Context Root Not Found ile bir pencere görürseniz, IBM MQ Console etkin değildir; beklemeniz gerekir.
- IBM MQ Console başlığı ile bir pencere görüntülenmezse, WLP_USER_DIRECTORY/servers/mqweb/logs/messages.log uygulamasında ek tanımlama bilgileri olabilir; burada WLP_USER_DIRECTORY, mqweb sunucu tanımlamasını yaratmak için **crtmqweb.sh** komut dosyasını çalıştırdığınızda belirttiğiniz dizindir.

Bu dosyanın ASCII 'de olduğunu ve dosyayı görüntülemek için **oedit** ' u USS komut satırında ya da ISPF seçeneğini 3.17 kullanarak ya da **va** (ASCII ' yi görüntüle) satır komutunu kullanabileceğini göz önünde bulundurun.

- STDOUT ' da aşağıdakilere benzer iletiler alabilirsiniz:

```

[WARNING ] SRVE0190E: File not found: /nls/en_GB/labels.json
[WARNING ] SRVE0190E: File not found: /nls/en_GB/errors.json
[WARNING ] SRVE0190E: File not found: /nls/en_GB/strings.json
[WARNING ] SRVE0190E: File not found: /nls/en_GB/pcf.json

```

Bu iletileri yoksayabilirsiniz.

d. mqweb sunucusunun çalışmakta olduğu sistemde tanımlanmış kuyruk yöneticileriniz varsa ve mqweb sunucusuyla aynı düzeyde çalışıyorsa, bu kuyruk yöneticileri MQ Console' deki **Yerel Kuyruk Yöneticileri** gerinde görüntülenir.

Hiçbir kuyruk yöneticisi görüntülenmiyorsa, son IPL ' den bu yana başlatılmış olan mqweb Server ile aynı düzeyde kuyruk yöneticilerine sahip değilsiniz.

3. Bir ileti aldıysanız:

Lost communication with the server Could not establish communication with the server.

a) mqweb sunucusunu başlatmak için kullanılan yordamda:

i) STEPLIB kitaplıklarının doğru düzeyde olduğunu ve APF ' nin yetkili olduğunu denetleyin.

ii) PATH ve LIBPATH noktasını doğru yolu denetleyin. The PATH and LIBPATH are defined in your web server started task procedure, that you based on sample CSQ4WEBS.

b) USS ' de `ls -Eltr PathPrefix/web/bin/dspmqqkomutunu` kullanın; burada PathPrefix , IBM MQ Unix System Services Components kuruluş yoludur.

Bu, `-rwxr-xr-t a-s- ... /mqm/V9R0M4/web/bin/dspmqqile` benzer bir çıktı görüntülemelidir.

T ve işaretlerin hazır olduğunu kontrol et.

Şu komutu kullanın:

- Yapışan biti (t) ayarlamak için `chmod +t PathPrefix/web/bin/dspmqq`
- `extattr +a PathPrefix/web/bin/dspmqq` to set the APF-authorized attribute (a)

V 9.0.1 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:

- **V 9.0.2** 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 kapıyı 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 gerekip gerekmediğini tanımlar mı, yoksa sorgu değiştirgelerinin 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) Şu örnek URL adresiyle başla:

V 9.0.4 IBM MQ 9.0.4 ve sonraki yayın:

```
https://host:port/ibmmq/rest/v1/admin
```

IBM MQ 9.0.3 ve önceki düzeyler:

```
https://host:port/ibmmq/rest/v1
```

ana makine

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

Varsayılan değer localhost'tur.

kapı

administrative REST API ' un kullandığı HTTPS kapı numarasını belirtir.

Varsayılan değer 9443'tur.

V 9.0.1 HTTP bağlantılarını etkinleştirdiyse, HTTPS yerine HTTP ' yi kullanabilirsiniz. HTTP ' nin etkinleştirilmesiyle ilgili daha fazla bilgi için [HTTP ve HTTPS kapılarının yapılandırılması](#) başlıklı konuya bakın.

URL örnek URL 'sinin nasıl belirleneceği hakkında daha fazla bilgi için bkz. [“REST API URL 'sinin belirlenmesi” sayfa 71](#).

b) Kaynağı URL yoluna ekleyin.

Aşağıdaki kaynaklar kullanılabilir:

- [kurulum](#)
- [qmgr](#)
- **V 9.0.2** [kuyruk](#)
- **V 9.0.4** [abonelik](#)
- **V 9.0.4** [channel](#)

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

V 9.0.4 IBM MQ 9.0.4 ve sonraki yayın:

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

IBM MQ 9.0.3 ve önceki düzeyler:

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

V 9.0.4 IBM MQ 9.0.4 ve sonraki yayın:

```
https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1
```

IBM MQ 9.0.3 ve önceki düzeyler:

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

V 9.0.4 IBM MQ 9.0.4 ve sonraki yayın:

```
https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1?attributes=*
```

IBM MQ 9.0.3 ve önceki düzeyler:

```
https://localhost:9443/ibmmq/rest/v1/qmgr/QM1?attributes=*
```

e) URL ' ye daha fazla isteğe bağlı sorgu parametreleri ekleyin.

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

2. URL ' de ilgili HTTP yöntemini çağırın. İsteğe bağlı JSON bilgi yükünü belirtin ve doğrulamak 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.

V 9.0.5 Aktarma ya da aktarma durumunun MFT listesini alma

How you obtain the Managed File Transfer list of transfers, using the REST API

Daha fazla bilgi için bkz. [REST API komutunu kullanarak yönetim](#).

Aktarımların bir listesini elde etmek için:

1. MFT ve MFT web uygulamasını barındıran sunucuya bir GET isteği gönderin. Bir isteği gönderirken URL ' nin biçimini göz önünde bulundurmanız gerekir.

Ek bilgi için [admin/mft/transfer](#) başlıklı konuya bakın.

2. Bir istek alındığında, isteğin geçerli olup olmadığını görmek için istek üzerine bir onay işlemi gerçekleştirilir.

Ek bilgi için [MFT REST API güvenliğinin yapılandırılması](#) başlıklı konuya bakın.

3. İstek geçerliyse, yanıt gövdesiyle birlikte başarılı bir yanıt alırsınız. Ters durumda, uygun bir hata kodu ve yanıt kodu oluşturulur.

Yanıtların listesi için bkz. [Yanıt durumu kodları](#) .

Bir aktarma işleminin durumunu almak için:

1. MFT veMFT web uygulamasını barındıran sunucuya bir [GET](#) isteği gönderin. Bir isteği gönderirken URL 'nin biçimini göz önünde bulundurmanız gerekir.
Ek bilgi için [admin/mft/transfer](#) başlıklı konuya bakın.
2. Bir istek alındığında, isteğin geçerli olup olmadığını görmek için istek üzerine bir onay işlemi gerçekleştirilir.
Ek bilgi için [MFT REST API güvenliğinin yapılandırılması](#) başlıklı konuya bakın.
3. İstek geçerliyse, yanıt gövdesiyle birlikte başarılı bir yanıt alırsınız. Ters durumda, uygun bir hata kodu ve yanıt kodu oluşturulur.
Yanıtların listesi için bkz. [Yanıt durumu kodları](#) .

Örnek yanıt gövdesi

Aktarım listesi iç depolamadan alınır ve aktarımların her biri için JSON biçiminde bir yanıt gövdesi oluşturulur. Bir aktarıma ilişkin yanıt gövdesi aşağıdakine benzer bir şey görünür:

```
{ "transfer": [
  {
    "destinationAgent": { "name": "AGENT.TRI.BANK" },
    "originator": {
      "host": "192.168.99.1",
      "userId": "johndoe"
    },
    "sourceAgent": { "name": "TESTAGENT" },
    "statistics": {
      "endTime": "2018-01-08T16:22:15.569Z",
      "numberOfFileFailures": 0,
      "numberOfFileSuccesses": 2,
      "numberOfFileWarnings": 0,
      "numberOfFiles": 2,
      "startTime": "2018-01-08T16:22:15.242Z"
    },
    "status": {
      "state": "successful"
    },
    "id": "414D51204D465444454D4F3320202020513E525A21109908"
  },
  {
    "destinationAgent": { "name": "AGENT.TRI.BANK" },
    "originator": {
      "host": "192.168.99.1",
      "userId": "ramsubbarao"
    },
    "sourceAgent": { "name": "TESTAGENT" },
    "statistics": {
      "endTime": "2018-01-08T16:22:13.573Z",
      "numberOfFileFailures": 0,
      "numberOfFileSuccesses": 2,
      "numberOfFileWarnings": 0,
      "numberOfFiles": 2,
      "startTime": "2018-01-08T16:22:13.167Z"
    },
    "status": {
      "state": "successful"
    },
    "id": "414D51204D465444454D4F3320202020513E525A21109702"
  }
]
```

Özniteliklerle ilgili ek bilgi için [REST API JSON yanıt gövdesi öznitelikleri](#) başlıklı konuya bakın.

V 9.0.5 MFT aracı durumu alma

REST API aracısını kullanarak Managed File Transfer aracı durumu nasıl elde edebilirsiniz?

Daha fazla bilgi için bkz. [REST API komutunu kullanarak yönetim](#).

Temsilcinin durumu almak için:

1. Send a GAL request to the agent under the default coordination queue manager that is using the MFT andMFT web application. Bir isteđi gönderirken URL ' nin biçimini göz önünde bulundurmanız gerekir.
Ek bilgi için [admin/mft/agent](#) başlıklı konuya bakın.
2. Bir istek alındığında, isteđin geçerli olup olmadığını görmek için istek üzerine bir onay işlemi gerçekleştirilir.
Ek bilgi için [MFT REST API güvenliğinin yapılandırılması](#) başlıklı konuya bakın.
3. İstek geçerliyse, yanıt gövdesiyle birlikte başarılı bir yanıt alırsınız. Ters durumda, uygun bir hata kodu ve yanıt kodu oluşturulur.
Yanıtların listesi için bkz. [Yanıt durumu kodları](#) .

Örnek URL adresleri

{baseURI}/ibmmq/rest/{version}/admin/mft/agent

Eşgüdüm kuyruk yöneticisi altındaki tüm araçların **name**, **state** ve **type** listesini görüntüler.

Bu özniteliklerle ilgili ek bilgi için [Aracı durumu REST API öznitelikleri](#) başlıklı konuya bakın.

URL dizgisinin sonuna `?attributes=*` eklerseniz, çıkış, eşgüdüm kuyruk yöneticisi altında bulunan her aracının tüm özniteliklerinin bir listesini içerir.

If you add `/{agentName}` to the end of the URL string, the output contains the **name**, **state**, and **type** of the agent that matches agentName.

Similarly, if you add `/{agentName}?attributes=*` to the end of the URL string, the output contains a list of all the attributes for the agent that matches agentName.

{baseURI}/ibmmq/rest/{version}/admin/mft/agent?name={<prefix>*<suffix> }

For example, `{baseURI}/ibmmq/rest/{version}/admin/mft/agent?name=AGENT*TEST` outputs the default attribute summary for the agents starting with the name *ARACI* and ending with the name *TEST*.

URL ' den `*TEST` öđesini çıkarırsanız, çıkış, *AGENT* adıyla başlayan tüm araçlar için varsayılan öznitelik özetini içerir.

{baseURI}/ibmmq/rest/{version}/admin/mft/agent?type={agentType}

Örneđin, `{baseURI}/ibmmq/rest/{version}/admin/mft/agent?type=standard` , *standart* tipindeki araçlar için varsayılan öznitelik özetini çıkarır.

Benzer bir şekilde, `type={agentType}` yerine `state={agentState}` deđerini deđiştirirseniz, çıkış, belirtilen durumun araçları için varsayılan öznitelik özetini görüntüler.

{baseURI}/ibmmq/rest/{version}/admin/mft/agent?type={agentType}&attributes=*

Örneđin, `{baseURI}/ibmmq/rest/{version}/admin/mft/agent?type=standard&attributes=*` bir aracının tipi *standart* olan tüm öznitelikleri çıkarır.

Ayrıca, öznitelikler birleşimine göre de süzgeç uygulayabilirsiniz. Örneđin:

- `{baseURI}/ibmmq/rest/{version}/admin/mft/agent?name=*bob&type=standard&state=ready&attributes=*`
- `{baseURI}/ibmmq/rest/{version}/admin/mft/agent?type=standard&state=ready&attributes=*`
- `{baseURI}/ibmmq/rest/{version}/admin/mft/agent?name=agent*&state=ready&attributes=*`

Örnek yanıt gövdesi

Tanımlanan özniteliklerle ilgili ek bilgi için [Aracı durumu REST API öznitelikleri](#) ve [Aracı durumu REST API yanıt gövdesi öznitelikleri](#) başlıklı konuya bakın.

Issuing the following command, `/ibmmq/rest/v1/admin/mft/agent/`, lists the basic agent status of agent name, type, and state. Örneğin:

```
{
  "agent": [ { "name": "AGENT1",
               "state": "ready",
               "type": "standard" },
             { "name": "AGENT2",
               "state": "ready",
               "type": "standard" },
             { "name": "BRIDGE_AGENT3",
               "type": "protocolBridge",
               "state": "ready" },
             { "name": "CD_AGENT",
               "type": "connectDirectBridge",
               "state": "ready" } ]
}
```

Aşağıdaki komutların her birini yayınlayın:

```
/ibmmq/rest/v1/admin/mft/agent?type=standard
/ibmmq/rest/v1/admin/mft/agent?state=stopped
/ibmmq/rest/v1/admin/mft/agent?name=AGENT*
```

her bir komutta, belirli bir ad, durum ve tip ile eşleşen bir aracının temel bilgileri listelenir. Örneğin:

```
{ "agent": [ { "name": "AGENT1",
               "state": "ready",
               "type": "standard" },
             { "name": "AGENT2",
               "state": "ready",
               "type": "standard" } ]
}
{ "agent": [ { "name": "AGENT1",
               "state": "stopped",
               "type": "standard" },
             { "name": "AGENT2",
               "state": "stopped",
               "type": "standard" } ]
}
{ "agent": [ { "name": "AGENT1",
               "state": "ready",
               "type": "standard" },
             { "name": "AGENT2",
               "state": "ready",
               "type": "standard" } ]
}
```

Aşağıdaki komutu yayınlayarak `/ibmmq/rest/v1/admin/mft/agent?attributes=general&type=standard`, general öznelikleriyle birlikte *standard* türdeki tüm iletileri listeler. Örneğin:

```
{
  "agent": [
    {
      "name": "AGENT1",
      "state": "ready",
      "type": "standard",
      "general": {
        "description": "Standard connected to the qmgr in client mode",
        "statusAge": "06:31:00",
        "version": "9.0.3.0",
        "level": "p903-L170513",
        "statusPublicationRate": 300,
        "statusPublishTime": "2017-10-31T06:57:07.000Z",
        "maximumQueuedTransfers": 1000,
        "maximumDestinationTransfers": 25,
        "maximumSourceTransfers": 25,
        "operatingSystem": "Windows7"
      }
    },
    {
      "name": "AGENT2",
      "state": "ready",
      "type": "standard",
      "general": {
        "description": "Standard connected to qmgr in Binding mode",
        "statusAge": "05:00:00",

```

```

        "version" : "9.0.3.0",
        "level" : "p903-L170513",
        "statusPublicationRate" : 300,
        "statusPublishTime" : "2017-09-13T09:10:09.000Z",
        "maximumQueuedTransfers" : 1000,
        "maximumDestinationTransfers":25,
        "maximumSourceTransfers":25,
        "operatingSystem" : "Windows7" }
    }
}

```

V 9.0.4 REST API komutunu kullanarak uzaktan yönetim

Uzak kuyruk yöneticilerini ve bu kuyruk yöneticileriyle ilişkili IBM MQ nesnelerini 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 yapılandırmabaşlıklı 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 68.
- 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 düzeyinde 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
- **V 9.0.5** /channel
- **V 9.0.5** /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. The gateway queue manager is specified by either setting the `mqRestGatewayQmgr` attribute by using the **setmqweb** command to the name of the gateway queue manager, or sending the name of the gateway queue manager in a header that is sent with the request. 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. [“Yerel kuyruk yöneticisinden uzak denetim” sayfa 186.](#)
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 grubundaydıysa, 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 özneteliklerini 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 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, `remoteQM` uzak 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.

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:

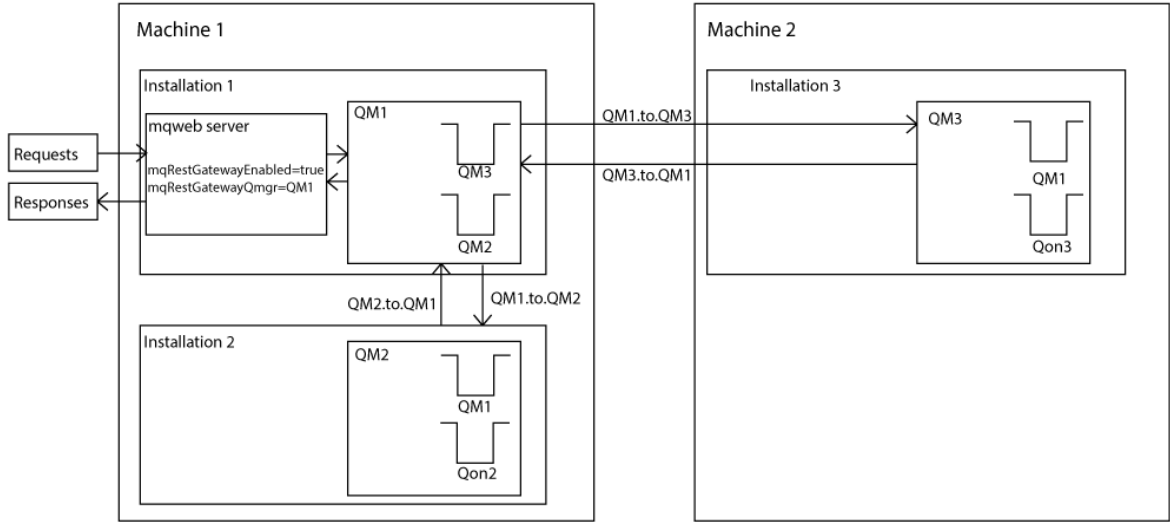
- Kuyruk yöneticisinde (QM1), Installation 1'da Machine 1'da:
 - Gönderen kanalı QM1.to.QM2
 - Günlük nesnesi kanalı QM2.to.QM1
 - Gönderen kanalı QM1.to.QM3
 - Günlük nesnesi kanalı QM3.to.QM1
 - İletim kuyruğu QM2
 - İletim kuyruğu QM3
 - 1414 kapısında yapılandırılan bir dinleyici
- Kuyruk yöneticisinde (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 15. 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"
}]
}
```

V 9.0.2 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.

IBM MQ 9.0.2' ta, 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 gerekiyorsa, 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 tanııcı aşağıdaki öğelere sahiptir:

MQWB

İletinin MQ Rest API 'sında ortaya çıkan bir örnek olduğunu gösteren bir örnek.

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

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

Dizgi.

Hatayı deneyimleyen kuyruk yöneticisinin adı.

V 9.0.4 Bu alan messaging REST API için geçerli değildir.

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

Numara.

Hata ile ilişkili MQ tamamlama kodu.

V 9.0.4 reasonCode

Bu alan yalnızca **type** pcf, 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.

Kuyruk paylaşım gruplarıyla ilgili hatalar

z/OS

In a queue sharing group, it is possible to specify an optional query parameter of **commandScope** for certain commands. Bu değiştirge, 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_UNKOWNN_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_UNKOWNN_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"
}]}
```

V 9.0.1 REST API Keşif

REST API için Documentation, IBM Documentation içinde ve Swagger biçiminde kullanılabilir. Swagger, REST API' lerini belgelemek için yaygın olarak kullanılan bir yaklaşımdır. REST API için Swagger belgeleri, mqweb sunucusu üzerinde API keşif özelliği etkinleştirilerek görüntülenebilir.

Başlamadan önce

API keşfi kullanarak Swagger belgelerini görüntülemek için mqweb sunucusu güvenliğini etkinleştirmeniz gerekir. Güvenliği etkinleştirmek için gereken adımlarla ilgili daha fazla bilgi için [IBM MQ Console güvenliğinin yapılandırılması](#) başlıklı konuya bakın.

Yordam

1. Aşağıdaki dizinlerden birindeki mqwebuser.xml dosyasını bulun:

- **ULW** MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb
- **Z/OS** WLP_user_directory/servers/mqweb

Where *WLP_user_directory* is the directory that was specified when the **crtmqweb.sh** script ran to create the mqweb server definition.

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

- mqwebuser.xml etiketinizde <featureManager> etiketleri varsa, <featureManager> etiketleri içinde aşağıdaki XML ' i ekleyin:

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

- <featureManager> etiketleriniz mqwebuser.xml dosyanızın içinde yoksa, <server> etiketleri içinde aşağıdaki XML ' i ekleyin:

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

3. Swagger belgelerini görüntülemek için aşağıdaki yöntemlerden birini kullanın:

- Bir tarayıcıda aşağıdaki URL 'yi girerek REST API 'yi göz atabileceğiniz ve deneyebileceğiniz bir web sayfası görüntüleyin:

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

Her isteğin doğrulamaya ek olarak, her POST, PATCH ya da DELETE isteği için bir ibm-mq-rest-csrf-token üstbilgisi eklemelisiniz.

V 9.0.5

Bu üstbilginin gerekli içeriği, IBM MQsürümüne bağlı olarak değişir:

- IBM MQ 9.0.5' tan, ibm-mq-rest-csrf-token HTTP üstbilgisinin istekte bulunması gerekir; değeri boşluk da dahil olmak üzere her şey olabilir.
- IBM MQ 9.0.5öncesinde, üstbilginin değeri csrfToken çerezinin içeriğidir. REST API ile bir HTTP GET yöntemi kullanıldığında bir csrfToken oluşturulur. Tarayıcınızdaki adres çubuğuna aşağıdaki metni girerek tanımlama bilgilerinin içeriğini görüntüleyebilirsiniz:

```
javascript:alert(document.cookie)
```

Bu istek üstbilgisi, isteğin kimlik doğrulaması için kullanılmakta olan kimlik bilgilerinin, kimlik bilgilerinin sahibi tarafından kullanıldığını onaylamak için kullanılır. Yani, simge, siteler arası istek sahteciliği saldırılarını önlemek için kullanılır.

- Aşağıdaki URL ' ye HTTP GET komutu vererek tüm REST API ögesini açıklayan tek bir Swagger 2 belgesi alın:

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

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

ana makine

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

Varsayılan deęer localhost'tur.

kapı

administrative REST API ' un kullandığı HTTPS kapı numarasını belirtir.

Varsayılan deęer 9443'tur.

Anasistem adı ya da kapı numarası varsayılandan deęiştirilirse, doęru deęerleri REST API URL adresinden belirleyebilirsiniz. URL ' nin saptanması hakkında daha fazla bilgi için bkz. [“REST API URL 'sinin belirlenmesi” sayfa 71.](#)

V 9.0.4 REST API ulusal dil desteęi

REST API , belirli niteliklere sahip, ulusal dilleri bir HTTP isteęinin bir parçası olarak belirtme yeteneęi saęlar.

Arka Plan

HTTP üstbilgileri , isteklerde belirli davranıřların belirtilmesine ve yanıtlarda saęlanacak ek bilgilerde belirtilmesine olanak saęlar.

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

Ulusal dil belirtilmesi

ACCEPT-LANGUAGE HTTP üstbilgisinde, bir ya da daha fazla dil etiketi saęlanabilir. İsteęe baęlı olarak, bir sırayı etiketlerle iliřkilendirebilir ve tercihe göre sıralanmıř bir liste belirtimine olanak tanıyabilirsiniz. [Bu sayfa](#) , ilkeye iliřkin yararlı bir tartıřmaya sahiptir.

REST API bu üstbilgiyi kabul eder, KABUL-DİL üstbilgisinden bir dil seęin ve bu dilde iletiler döndürür. ACCEPT-LANGUAGE üstbilgisi, REST API ' in destekleyebileceęi bir dil içermedięinde, iletiler varsayılan bir dilde döndürölür. Bu varsayılan dil, REST API web sunucusunun varsayılan yerel ayarına karřılık gelir.

[“Hangi veriler çevrilir?” sayfa 88](#) bölümü, hangi verilerin çevrildięini açıklar.

Yanıtlarda geęerli dili belirtir

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

Hangi veriler çevrilir?

Hata ve bilgi iletileri çevrilir, dięer metinler de çevrilmez.

- Bir kuyruk yöneticisinden döndürölren veriler çevrilmez; örneęin, REST API aracılıęıyla 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 ařaęıdaki dillere çevrilir.

Çince (Basitleřtirilmiř)

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.

V 9.0.1

IBM MQ Consolekomutunu kullanarak yönetim

IBM MQ Console , ortak yönetim görevlerini gerçekleştirmek için kullanılacak web tabanlı bir kullanıcı arabirimidir.

Başlamadan önce**Not:**

- IBM MQ Consolekullanırken kuyruk yöneticilerinizden herhangi birinde komut sunucusunu geçersiz kılmayın. Bir kuyruk yöneticisi için komut sunucusu devre dışı bırakılırsa, IBM MQ Console , komutların işlenmesine ilişkin uzun gecikmeler ile yanıt vermez duruma gelir. Kuyruk yöneticisine verilen komutların, komut sunucusunun devre dışı bırakıldığı zaman aşımı süresi.

- IBM MQ Console'a bağlandığında, tarayıcı, tarayıcı tarafından ayarlanan yerel ayarlarda IBM MQ Console 'ı görüntülemeye çalışır. IBM MQ Console 'ın belirtilen dili destekleyip desteklemediğini görmek için bir onay imi yapılır. Dil dosyaları bulunamazsa, ABD İngilizcesi kullanılır ve mqweb sunucusu bu dosyayı dosya bulunmadığı şekilde günlüğe kaydeder. Bu nedenle, tarayıcınız IBM MQ Console tarafından desteklenmeyen bir dilde ayarlandığında, aşağıdaki iletiler beklenir:

```
SRVE0190E: File not found: /nls/en_GB/labels.json
SRVE0190E: File not found: /nls/en_GB/pcf.json
SRVE0190E: File not found: /nls/en_GB/errors.json
SRVE0190E: File not found: /nls/en_GB/strings.json
```

Yordam

- [Yerel kuyruk yöneticileriyle çalışılması](#)
- [IBM MQ nesneleriyle çalışma](#)
- [Yetki kayıtlarıyla çalışma](#)
- [Sistem kaynağı kullanımının izlenmesi](#)
- [Gösterge panosu düzenlerini yapılandırma](#)

V 9.0.1 IBM MQ Console ile çalışmaya başlama

IBM MQ Console 'u başlatabilmeniz için önce doğru bileşenleri kurmalı ve mqweb sunucusunu başlatmalısınız. Daha sonra bir tarayıcıda IBM MQ Console 'ı başlatabilirsiniz.

Başlamadan önce

IBM i IBM üzerinde, komutların QSHHELL içinde çalıştırılması gerekir.

Bu görev hakkında

Bu göreve ilişkin yordam, IBM MQ Console ile hızlı bir şekilde başlamak için temel adımlar üzerinde odaklanır. Yapılandırmanızı özelleştirme hakkında daha fazla bilgi için, sonraki sırada listelenenler altında listelenen bağlantılara bakın.

Not: You have the option during installation of configuring the IBM MQ Console without security.

Yordam

1. IBM MQ Console ve REST API bileşenini kurun:

- **V 9.0.4** **AIX** AIX' ta mqm.web.rte kütük kümesini kurun.
- **Linux** Linux' ta MQSeriesWeb bileşenini kurun. For more information about installing components and features on Linux, see [Linux kuruluş görevleri](#).
- **Windows** Windows' ta Web Administration özelliğini kurun. For more information about installing components and features on Windows, see [Windows kuruluş görevleri](#).
- **z/OS** z/OS' ta IBM MQ for z/OS Unix System Web Services Components özelliğini kurun. For more information about installing components and features on z/OS, see [z/OS kuruluş görevleri](#).

2. Kullanıcıların ve grupların IBM MQ Console' e erişmelerini sağlamak için temel güvenliği yapılandırın.

a) Örnek basic_registry.xml dosyasını `MQ_INSTALLATION_PATH/web/mq/samp/configuration` dizininden şu dizine kopyalayın:

- **ULW** UNIX, Linux, and Windows üzerinde: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`

- **z/OS** z/OS üzerinde: `WLP_user_directory/servers/mqweb`

where `WLP_kullanıcı_dizini` is the directory that was specified when the `crtmqweb.sh` script ran to create the mqweb server definition.

b) Örnek XML dosyasını `mqwebuser.xml` olarak yeniden adlandırın.

Not: Yeniden adlandırılan bu dosya, administrative REST API için de kullanılan var olan bir dosyanın yerini alır. Therefore, if you changed the `mqwebuser.xml` file for the administrative REST API, copy your changes to the new XML file before you rename it.

3. Altyapıya bağımlı olarak, mqweb sunucusu ile uzak bağlantıları etkinleştirin:

- **Linux** İsteğe bağlı olarak Linux üzerinde
- **Windows** İsteğe bağlı olarak Windows üzerinde
- **z/OS** Açıkz/OS
- IBM MQ 9.0.4' tan **setmqweb** komutunu kullanın:
`setmqweb properties -k httpHost -v hostname`
- From IBM MQ 9.0.1, add the following XML to the `mqwebuser.xml` file, within the `<server>` tags:
`<variable name="httpHost" value="hostname" />`

Burada *anasistem*, etki alanı adı sonetine sahip IP adresini, etki alanı ad sunucusunu (DNS) anasistem adını ya da IBM MQ ' un kurulu olduğu sunucunun DNS ana makine adını belirtir. Kullanılabilir tüm ağ arabirimlerini belirlemek için yıldız imi (*) kullanın.



Uyarı: **z/OS** **V 9.0.4**

z/OS' ta **setmqweb** ya da **dspmqweb** komutlarını vermeden önce, değişken noktalarını mqweb sunucusu yapınıza gösteren `WLP_USER_DIR` ortam değişkenini ayarlamanız gerekir.

Bunu yapmak için şu komutu verin:

```
export WLP_USER_DIR=WLP_user_directory
```

Burada `WLP_user_directory`, `crtmqweb.sh` e aktarılan dizinin adıdır. Örneğin:

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

Ek bilgi için [Liberty sunucusu tanımlamasının yaratılması](#) başlıklı konuya bakın.

4. IBM MQ Console' u destekleyen mqweb sunucusunu başlatın:

- **Linux** **Windows** Windows ve Linux üzerinde, [ayrıcılıkli kullanıcı](#) olarak komut satırına aşağıdaki komutu girin:
`strmqweb`
- **z/OS** z/OS' ta, [Görev 29: IBM WLP sunucusu için bir yordam oluşturma](#) içinde oluşturduğunuz yordamı başlatın.

5. Tarayıcıda aşağıdaki URL 'yi girerek IBM MQ Console ' a bağlanın:

`https://hostname:9443/ibmmq/console`

Burada *anasistem*, etki alanı adı sonetine sahip IP adresi ya da etki alanı adı sunucusu (DNS) anasistem adını ya da IBM MQ ' un kurulu olduğu sunucunun DNS ana makine adını belirtir. 3. adımda uzak bağlantılar yapılandırmadıysanız, *anasistemadı* değeri localhosttur.

6. IBM MQ Console' da oturum açın. IBM MQ 9.0.2' tan `mqadmin` kullanıcı adını ve `mqadmin` parolasını kullanın. At IBM MQ 9.0.1, use the user name `admin`, and the password `admin`.

Sonraki adım

- Kullanıcıların ve grupların, LDAP ve istemci sertifikalarının nasıl yapılandırılacağı da dahil olmak üzere, IBM MQ Console güvenliğini yapılandırma hakkında daha fazla bilgi için [IBM MQ Console güvenliğinin yapılandırılması](#) başlıklı konuya bakın.
- HTTP bağlantılarının etkinleştirilmesi de içinde olmak üzere IBM MQ Console ayarlarının yapılandırılmasıyla ilgili daha fazla bilgi için bkz. [IBM MQ Console 'un yapılandırılması](#).
- Varsayılan URL değilse URL 'nin saptanması hakkında daha fazla bilgi için bkz. [“IBM MQ Console URL 'sinin belirlenmesi”](#) sayfa 92.

V 9.0.1 IBM MQ Console URL 'sinin belirlenmesi

IBM MQ Console 'a erişmek için kullanılan varsayılan URL şudur: `https://localhost:9443/ibmmq/console`. If the HTTP host or port is changed from the default, or if the HTTP port is enabled, you can determine the URL by using the **dspmweb** command.

Bu görev hakkında

V 9.0.4 From IBM MQ 9.0.4, you can use the **dspmweb status** command to determine the IBM MQ Console URL on Windows, Linux ve z/OS. IBM MQ 9.0.3 veearlier için, Windows ve Linux için **dspmweb** komutunu kullanabilirsiniz. z/OS üzerinde, URL 'yi belirlemek için `messages.log` dosyasını aramanız gerekir.

Yordam

V 9.0.4

URL adresini belirlemek için aşağıdaki yöntemlerden birini kullanın:

- IBM MQ 9.0.4' tan, **dspmweb status** komutunu [ayrıcılık](#) kullanıcı olarak kullanın:

1. Komut satırına **strmqweb** komutunu girerek, mqweb sunucusunun çalıştığından emin olun.
2. Komut satırındaki `dspmweb status` komutunu girerek URL 'yi görüntüleyin.

- IBM MQ 9.0.3 ve öncesi için, aşağıdaki yöntemlerden birini kullanın:

- **Linux** **Windows** Windows ya da Linux üzerinde, **dspmweb** komutunu [ayrıcılık](#) kullanıcı olarak kullanın:

1. Komut satırına **strmqweb** komutunu girerek, mqweb sunucusunun çalıştığından emin olun.
2. Komut satırındaki `dspmweb` komutunu girerek URL 'yi görüntüleyin.

- **z/OS** z/OS'ta, URL'yi bulmak için `messages.log` dosyasını kullanın:

1. `messages.log` dosyasını açın.

The `messages.log` file is on the `WLP_user_directory/servers/mqweb/logs` path, where `WLP_user_directory` is the directory that was specified when the **strmqweb.sh** script ran to create the mqweb server definition.

2. `ibmmq/console` ile biten en son `CWWKT0016I` ileti kodunu arayın. URL bu iletiye dahil edilir.

Not: The IBM MQ Console uses an internal URL of `/ibmmq/console/internal`. Bu URL bazı `CWWKT0016I` iletilerinde gösterilir, ancak yalnızca IBM MQ Console tarafından kullanılmak üzere gösterilir.

Örnek

The following example shows the message code `CWWKT0016I` with the URL as the default URL:

```
A CWWKT0016I: Web application available (default_host): https://localhost:9443/ibmmq/console
```

z/OS V 9.0.1 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 görevler

“IBM MQ Console komutunu kullanarak yönetim” sayfa 89

IBM MQ Console , ortak yönetim görevlerini gerçekleştirmek için kullanılacak web tabanlı bir kullanıcı arabirimidir.

İlgili bilgiler

Yerel kuyruk yöneticileriyle çalışılması

V 9.0.1 Yerel kuyruk yöneticileriyle çalışılması

Yerel kuyruk yöneticilerini yaratmak, yapılandırmak ve denetlemek için IBM MQ Console içindeki yerel kuyruk yöneticisi gerecini kullanabilirsiniz.

Bu görev hakkında

Yerel kuyruk yöneticisi gereci, yerel kuyruk yöneticilerini listeler. IBM MQ Console ' in çalışmakta olduğu IBM MQ kuruluşu. Aynı sistemde IBM MQ ' un farklı kuruluşlarıyla ilişkili kuyruk yöneticileri listelenmiyor. Listedeki çalışmak için listeden tek tek kuyruk yöneticileri seçebilirsiniz.

Not: V 9.0.4 IBM MQ Console , eşlenmiş veri kuyruğu yöneticilerini (RDQM ' ler) desteklemez.

gereç ekle V 9.0.5 Add widget ' i tıklayarak gösterge panonuzun yerel kuyruk yöneticisi pencere aracını ekleyebilirsiniz. Daha sonra **Yerel Kuyruk Yöneticileri** seçeneğini belirleyin.

V 9.0.5 Pencere aracını oluşturduktan sonra yeniden adlandırabilirsiniz. Fare gösterginizi başlık



çubuğunda dolaştır ve yeniden adlandır simgesi görüntülenir. Simgeyi tıklayın, Yeniden Adlandır pencere bileşeni penceresinde yeni bir ad girin ve **Yeniden Adlandır**' ı tıklayın.

z/OS z/OS üzerinde kuyruk yöneticilerini oluşturamaz, başlatamaz, durduramaz ya da silemezsiniz.

Yordam

- Yeni bir yerel kuyruk yöneticisi yaratmak için:
 - a) Yerel kuyruk yöneticisi pencere aracı araç çubuğunda oluştur simgesini

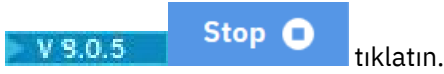
V 9.0.5 Create + tıkklatın.

- b) Yeni kuyruk yöneticisi için bir ad girin. Ad en çok 48 karakter içerebilir. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
- c) İsteğe bağlı: Kuyruk yöneticisinin dinlemesi için kullanılabilir bir TCP/IP kapısı girin. Kapı numarası 65535 'i aşmamalıdır.
- d) **Oluştur**'u tıklatın. Yeni kuyruk yöneticisi yaratılır ve başlatılır.
- Yerel kuyruk yöneticisini başlatmak için:
 - a) Yerel kuyruk yöneticisi gerecindeki listeden başlatmak istediğiniz kuyruk yöneticisini seçin.
 - b) Yerel kuyruk yöneticisi pencere aracı araç çubuğunda başlangıç simgesini



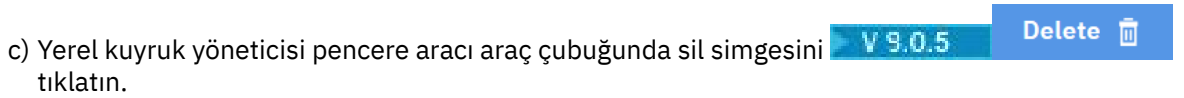
tıklatın.

- Yerel kuyruk yöneticisini durdurmak için:
 - a) Yerel kuyruk yöneticisi pencere bileşenindeki listeden durdurmak istediğiniz kuyruk yöneticisini seçin.
 - b) Yerel kuyruk yöneticisi pencere aracı araç çubuğunda durdur simgesini



tıklatın.

- c) Confirm that you want to stop the queue manager by clicking **Durdur**.
- Bir yerel kuyruk yöneticisini silmek için:
 - a) Yerel kuyruk yöneticisi gerecindeki listeden silmek istediğiniz kuyruk yöneticisini seçin.
 - b) Kuyruk yöneticisi çalışıyorsa, durdurun.



tıklatın.

- d) Kuyruk yöneticisini silmek istediğinizi doğrulamak için **Sildüğmesini** tıklatın. Kuyruk yöneticisi ve ilişkili tüm nesnelere silinir.
- Yerel bir kuyruk yöneticisine ilişkin özellikleri görüntülemek ve düzenlemek için:
 - a) Kuyruk yöneticisinin çalışır durumda olduğunu doğrulayın ve kuyruk yöneticisi listesinde bu yöneticiyi seçin.
 - b) Yerel kuyruk yöneticisi pencere aracı araç çubuğunda özellikler simgesi



tıklatın. Diğer bir seçenek olarak, kuyruk yöneticisini çift tıklatın.

- c) Özellikleri görüntüleyin ve gereken şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. For information about a property, you can view the property information in MQ Explorer belgelerindeki [Kuyruk yöneticisi özellikleri](#).
- Yerel kuyruk yöneticisine ilişkin güvenliği yenilemek için:
 - a) Yerel kuyruk yöneticisinin çalışır durumda olduğunu doğrulayın ve kuyruk yöneticisi listesinde bu kuyruk yöneticisini seçin.
 - b) Seç ... > **Güvenliği yenile**
 - c) Yenilemek için kuyruk yöneticisi güvenliğini seçin:
 - Yetkilendirme hizmetleri bileşeni tarafından dahili olarak tutulan yetkilerin listesini yenilemek için **Yetkilendirme hizmeti** seçeneğini belirleyin.
 - Bağlantı kimlik doğrulamasına ilişkin yapılandırmanın önbelleğe alınmış görünümünü yenilemek için **Bağlantı kimlik doğrulaması** seçeneğini belirleyin.
 - SSL ya da TLS anahtar havuzunun önbelleğe alınmış görünümünü yenilemek için **SSL** seçeneğini belirleyin. Bu seçenek, sertifikalı geri alma listeleri ve tüm şifreleme donanım parametreleri için kullanılan LDAP sunucularının yerlerini de yeniler.
- Yerel kuyruk yöneticisine ilişkin yetki kayıtlarıyla çalışmak için:

- a) Yerel kuyruk yöneticisinin çalışır durumda olduğunu doğrulayın ve kuyruk yöneticisi listesinde bu kuyruk yöneticisini seçin.
- b) Aşağıdaki seçeneklerden birini seçin:
 - Seç ... > **Yetki kayıtlarını yönet** , kuyruk yöneticisine ilişkin yetki kayıtlarıyla çalışmak ve kullanıcıların hangi işlem gruplarını alabileceğini belirtme.
 - Seç ... > **Yetki yaratma kayıtlarını yönet** kuyruk yöneticisine ilişkin yaratma yetkisi kayıtlarıyla çalışmak için, o kuyruk yöneticinde kullanıcı gruplarının yaratabileceği nesne gruplarını belirtin.
- Yerel kuyruk yöneticisi için otomatik olarak bir gösterge panosu sekmesi oluşturmak için:
 - a) Yerel kuyruk yöneticisi gerecindeki kuyruk yöneticisini seçin.
 - b) Seç ... > **Yeni gösterge panosu sekmesi ekle**
Yeni bir gösterge panosu sekmesi oluşturulur. Sekme, kuyruk yöneticisinin adını içerir.
- Yerel kuyruk yöneticileri listesine süzgeç uygulamak için:
 - a) Süzgeç metninizi arama kutusuna yazın.
 - b) Süzgeç uygulamayı durdurmak için, arama kutusundan metni silin.

V 9.0.1 IBM MQ nesneleriyle çalışma

You can use the IBM MQ object widgets in the IBM MQ Console to work with the different types of IBM MQ object.

Bu görev hakkında

Her IBM MQ nesnesi gereci, belirli bir kuyruk yöneticisiyle ilişkilendirilmiş nesnelere içerir. Gösterge panonuzun için aşağıdaki IBM MQ nesne pencere araçlarını ekleyebilirsiniz:

- Kuyruklar gereci
- Konular gereci
- Dinleyiciler gereci
- Kanallar pencere bileşeni
- İstemci bağlantı kanalları gereci
- Kimlik doğrulama bilgileri gereci
- Abonelikler gereci

Yordam

- Bir IBM MQ nesnesi pencere öğesi oluşturmak için:

- a) **Pencere aracı ekle** 'yi tıkkatın **V 9.0.5** **Add widget** .
- b) Listedeki uygun kuyruk yöneticisini seçin.
- c) Yaratmak istediğiniz nesne gereci tipinin adını tıkkatın.




- **V 9.0.5**
Nesne gerecini yeniden adlandırmak için:



- a) Fare göstergenizi başlık çubuğunda dolaştır ve yeniden adlandır simgesi görüntülenir. Simgeyi tıkkatın.
- b) Yeniden Adlandır pencere bileşeni penceresinde yeni bir ad girin ve **Yeniden Adlandır** 'ı tıkkatın.

- **V 9.0.5**

IBM MQ nesne gerecini yapılandırmak için:

- Pencere aracının başlık çubuğundaki yapılandır simgesini  tıklatın.
 - İsteğe bağlı: IBM MQ nesnelерinin görüntüleneceği kuyruk yöneticisini belirtin.
 - İsteğe bağlı: Sistem nesnelерinin gösterilip gösterilmeyeceğini ya da gizlenip gizlenmeyeceğini belirtin.
 - Kaydet**'i tıklatın.
- Gereçte görüntülenen nesnelere süzgeç uygulamak için:
 - Süzgeç metninizi arama kutusuna yazın.
 - Süzgeç uygulamayı durdurmak için, arama kutusundan metni silin.
 - Pencere bileşeninin içeriğini yenilemek için pencere aracının başlık çubuğundaki yenile simgesini  tıklatın.
 - Pencere aracını kaldırmak için, pencere bileşeninin başlık çubuğundaki kaldır simgesini  tıklatın.

V 9.0.1 Kuyruklarla Çalışma

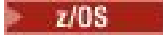
Belirli bir kuyruk yöneticisi için var olan kuyrukları göstermek için IBM MQ Console içindeki Kuyruklar pencere aracını kullanabilirsiniz. Daha sonra, kuyruklar ekleyebilir ve silebilir, bir kuyruğa ileti ekleyebilir ya da iletileri temizleyebilir, iletilere göz atabilir, kuyruğun özelliklerini görüntüleyebilir ve bir kuyruğa ilişkin özellikleri ayarlayabilir ve kuyruğun yetkili kayıtlarını yönetebilirsiniz.

Başlamadan önce



Kullanmak için önce bir kuyruk pencere aracı oluşturmalısınız. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. [“IBM MQ nesneleriyle çalışma” sayfa 95.](#)

Bu görev hakkında

Kuyruklar gereci, belirli bir kuyruk yöneticisi için var olan kuyrukları listeler. Birlikte çalışmak için listeden tek tek kuyruklar seçebilirsiniz.

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

Yordam

- Kuyruk eklemek için:
 - Kuyruklar pencere aracı araç çubuğunda oluştur simgesini  tıklatın.
 - Kuyruk için bir ad girin. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
 - Eklenecek kuyruğun tipini seçin.
 - Oluştur**'u tıklatın. Yeni kuyruk yaratıldı.
- Bir kuyruğa ileti koymak için:
 - Kuyruklar pencere aracındaki listeden ileti eklemek istediğiniz kuyruğu seçin. Bir model kuyruğu seçemezsiniz.
 - Kuyruklar pencere öğesi araç çubuğunda ileti ekle simgesini  tıklatın.
 - Kuyruğa koymak istediğiniz iletiyi girin.
 - Ekle**'yi tıklatın.
- Kuyruktan iletileri temizlemek için:

- a) İletileri kuyruklar pencere aracındaki listeden temizlemek istediğiniz yerel kuyruğu seçin.
- b) Seç  ... > **Kuyruğu temizle**.
- c) Confirm that you want to clear the queue by clicking **Kuyruğu Temizle**.
- Kuyruklardaki iletilere göz atmak için:
 - a) Kuyruklar pencere aracındaki listeden göz atmak istediğiniz yerel ya da diğer adı seçin.
 - b) Kuyruklar pencere aracı araç çubuğunda göz at simgesini   tıklatın. Kuyruktaki iletileri görüntüleyen göz atma iletisi penceresi açılır.
- Bir kuyruğu silmek için:
 - a) Silmek istediğiniz kuyruğu, kuyruklar pencere aracındaki listeden seçin.
 - b) Kuyruklar pencere öğesi araç çubuğunda sil simgesini   tıklatın.
 - c) İsteğe bağlı: If the queue has messages on it, confirm that the queue can be cleared by clicking **Kuyruğu temizle**.
 - d) Kuyruğu silmek istediğinizi doğrulamak için **Sildüğmesini** tıklatın. Kuyruk silindi.
- Bir kuyruğun özelliklerini görüntülemek ve düzenlemek için:
 - a) Kuyruklar pencere aracındaki kuyruğu seçin.
 - b) Kuyruklar pencere aracı araç çubuğunda özellikler simgesini   tıklatın. Diğer bir seçenek olarak, kuyruğu çift tıklatın.
 - c)  Özellikleri görüntüleyin ve gereken şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. For information about a property, view the property information in MQ Explorer belgelerindeki [Kuyruk özellikleri](#).
- Bir kuyruğa ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Pencere bileşenindeki kuyruğu seçin.
 - b) Tıklatın  ... > **Yetki kayıtlarını yönet**.
Yetki kayıtları, kullanıcıların ve yöneticilerin seçilen kuyruğun üzerinde olduğu izinleri gösterir.

Konularla çalışma


Konuları eklemek ve silmek, bir konunun özelliklerini görüntülemek ve ayarlamak için IBM MQ Console içindeki konular pencere aracını kullanabilirsiniz.

Başlamadan önce

Bir konuyu kullanabilmek için önce bir konu pencere öğesi oluşturmalısınız. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. [“IBM MQ nesneleriyle çalışma” sayfa 95](#).













Bu görev hakkında

Konular pencere aracında, belirli bir kuyruk yöneticisi için var olan konular listelenir. Birlikte 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 pencere ögesi araç çubuğunda oluştur simgesini   tıklatın.
- b) Yeni konu için bir ad girin. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
- c) Konu için, konu için iletileri yayınlayacağı konu dizesini belirtin. Daha fazla bilgi için bkz. [Konu özellikleri](#).
- d) **Oluştur**'u tıklatın. Yeni konu yaratılır.
- Bir konuyu silmek için:
 - a) Konular pencere aracındaki listeden silmek istediğiniz konuyu seçin.
 - b) Konular pencere aracı araç çubuğunda sil simgesini   tıklatın.
 - c) **Sil** seçeneğini tıklatarak konuyu silmek istediğinizi onaylayın. Konu silindi.
- Bir konuya ilişkin özellikleri görüntülemek ve düzenlemek için:
 - a) Konular pencere aracındaki konuyu seçin.
 - b) Konular pencere aracı araç çubuğunda özellikler simgesini   tıklatın. Diğer bir seçenek olarak, konuyu çift tıklatın.
 - c)  Özellikleri görüntüleyin ve gereken şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. For information about a property, view the property information in MQ Explorer belgelerindeki [Konu özellikleri](#).
- Bir konuyla ilgili bir iletiyi yayınlamak için:
 - a) Konular pencere aracı araç çubuğunda ileti ekle simgesini   tıklatın.
 - b) **İleti** alanına bir ileti girin.
 - c) **Konu dizesi** alanında iletiyi yayımlamak için konu dizesini girin.
 - d) **Yayınla**'yı tıklatın.
- Bir konuya abone olmak için:
 - a) Konular pencere aracı araç çubuğunda abone olma simgesini   tıklatın.
 - b) **Konu dizesi** alanına abone olmak için konu dizesini girin.
 - c) **Abone Oldüğmesini** tıklatın.
- Bir konuya ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Konular pencere aracındaki konuyu seçin.
 - b) Bu ögeyi tıklatın  ... > **Yetki kayıtlarını yönet**.
Yetki kayıtları, kullanıcıların ve yöneticilerin seçilen konu üzerinde sahip olduğu izinleri gösterir.

Dinleyicilerle çalışma

Dinleyicileri eklemek ve silmek, dinleyicileri başlatmak ve durdurmak, dinleyici özelliklerini görüntülemek ve ayarlamak ve bir dinleyiciye ilişkin yetki kayıtlarını yönetmek için IBM MQ Console içindeki dinleyici pencere aracını kullanabilirsiniz.

Başlamadan önce













Bir dinleyici aracını kullanabilmeniz için önce bir dinleyici gerci yaratmanız gerekir. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. ["IBM MQ nesneleriyle çalışma"](#) sayfa 95.

Bu görev hakkında

İletişimciler pencere aracı, belirli bir kuyruk yöneticisi için var olan dinleyicileri listeler. Birlikte çalışmak için listeden tek tek dinleyicileri seçebilirsiniz.

 z/OS üzerindeki dinleyici pencere aracını kullanamazsınız.

Yordam

- Bir TCP/IP dinleyicisi eklemek için:
 - a) Dinleyici pencere aracı araç çubuğunda oluştur simgesini   tıklatın.
 - b) Dinleyici için bir ad girin. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
 - c) Dinleyici için kullanılabilir bir TCP/IP kapısı girin. Kapı numarası 65535 'i aşmamalıdır.
 - d) **Oluştur**'u tıklatın. Yeni dinleyici yaratılır.
- Bir dinleyiciyi silmek için:
 - a) İletişimciler pencere aracındaki listeden silmek istediğiniz dinleyiciyi seçin.
 - b) Dinleyici pencere aracı araç çubuğunda sil simgesini   tıklatın.
 - c) Confirm that you want to delete the listener by clicking **Sil**. Dinleyici silinir.
- Bir dinleyici başlatmak için:
 - a) İletişimciler pencere aracındaki listeden başlatmak istediğiniz dinleyiciyi seçin.
 - b) Dinleyici pencere aracı araç çubuğunda başlangıç simgesini   tıklatın.
- Bir dinleyiciyi durdurmak için:
 - a) Dinleyici pencere aracındaki listeden durdurmak istediğiniz dinleyiciyi seçin.
 - b) Dinleyici pencere aracı araç çubuğunda durdur simgesini   tıklatın.
 - c) Confirm that you want to stop the listener by clicking **Durdur**.
- Bir dinleyiciye ilişkin özellikleri görüntülemek ve düzenlemek için:
 - a) Dinleyici pencere bileşenindeki dinleyiciyi seçin.
 - b) Dinleyici pencere aracı araç çubuğunda özellikler simgesi   tıklatın. Diğer bir seçenek olarak, dinleyiciyi çift tıklatın.
 - c)  Özellikleri görüntüleyin ve gereken şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. For information about a property, view the property information in MQ Explorer belgelerindeki [Dinleyici özellikleri](#).
- Bir dinleyiciye ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) Dinleyici pencere bileşenindeki dinleyiciyi seçin.
 - b) Tıklatın  ... > **Yetki kayıtlarını yönet**. Yetki kayıtları, kullanıcıların ve yöneticilerin seçilen dinleyiciye sahip olduğu izinleri gösterir.

Kanallarla çalışma


Kanalları eklemek ve silmek, kanalları başlatmak ve durdurmak, kanalları ilk duruma getirmek ve çözmek ve ping kanallarını silmek için IBM MQ Console içindeki kanallar pencere aracını kullanabilirsiniz. Ayrıca, bir kanalın özelliklerini görüntüleyebilir ve ayarlayabilir ve kanalın yetki kayıtlarını yönetebilirsiniz.

Başlamadan önce

Kullanabilmeniz için önce bir kanal pencere bileşeni oluşturmanız gerekir. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. [“IBM MQ nesneleriyle çalışma” sayfa 95.](#)


Bu görev hakkında

Kanallar pencere aracı, belirli bir kuyruk yöneticisi için var olan kanalları listeler. Listeden çalışmak için listeden tek tek kanallar seçebilirsiniz.


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

Yordam


- Kanal eklemek için:

- a) Kanallar pencere aracı araç çubuğunda oluştur simgesini  tıklatın.
- b) Kanal için bir ad girin. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
- c) Eklenecek kanala ilişkin tipi seçin.
- d) Gönderen, kümeyle gönderici ya da istekçi kanalı yaratıyorsanız, bağlantı adını belirtin. Bağlantı adı, hedef kuyruk yöneticisini bulandıran bilgisayarın adıdır. Adın biçimi şöyledir: *computer_name(port_number). computer_name* , hedef kuyruk yöneticisini barındıran bilgisayarın adı ya da IP adresidir ve *port_number* , hedef kuyruk yöneticisinin dinleyicisinin kullandığı kapıdır.
- e) Bir gönderen kanalı ya da sunucu kanalı yaratıyorsanız, kanalın alıcı ucundaki kuyruk yöneticisine karşılık gelen iletim kuyruğunu belirleyin.
- f) **Oluştur**'u tıklatın. Yeni kanal yaratıldı.


- Bir kanalı silmek için:

- a) Kanallar pencere aracındaki listeden silmek istediğiniz kanalı seçin.
- b) Pencere ögesi araç çubuğunda sil simgesini  tıklatın.
- c) Kanalı silmek istediğinizi doğrulamak için **Sildüğmesini** tıklatın. Kanal silindi.

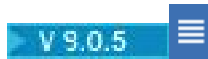
- Bir kanalı başlatmak için:


- a) Kanallar pencere aracındaki listeden başlatmak istediğiniz kanalı seçin.
- b) Pencere ögesi araç çubuğunda başlangıç simgesini  tıklatın.

- Bir kanalı durdurmak için:

- a) Kanallar pencere aracındaki listeden durdurmak istediğiniz kanalı seçin.
- b) Pencere ögesi araç çubuğunda durdur simgesini  tıklatın.
- c) Kanalı durdurmak istediğinizi doğrulamak için **Durdurdüğmesini** tıklatın.





- Bir kanala ilişkin özellikleri görüntülemek için:

- a) Kanallar pencere aracındaki kanalı seçin.
- b) Kanallar pencere aracı araç çubuğunda özellikler simgesi  tıklatın. Diğer bir seçenek olarak, kanalı çift tıklatın.

- c)  Özellikleri görüntüleyin ve gereken şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. For information about a property, view the property information in MQ Explorer belgelerindeki [Kanal özellikleri](#).

- Bir kanalı ilk durumuna getirmek için:

- a) Kanallar pencere aracındaki kanalı seçin.

- b) Tıklatın  ... > **İlk duruma getirme**.
- c) 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 kanalı çözmek için:
 - a) Kanallar pencere aracındaki kanalı seçin.
 - b) Tıklatın  ... > **Çözümle**.
 - c) Choose whether to commit or back out the current batch of messages by clicking **Kesinleştir** or **Arka çık**.
- Bir kanala ping komutu göndermek için:
 - a) Kanallar pencere aracındaki kanalı seçin.
 - b) Tıklatın  ... > **Ping**.
- Bir kanala ilişkin yetki kayıtlarını görüntülemek ya da düzenlemek için:
 - a) Pencere bileşenindeki kanalı seçin.
 - b) Tıklatın  ... > **Yetki kayıtlarını yönet**.
Yetki kayıtları, kullanıcıların ve yöneticilerin seçilen kanalda sahip olduğu izinleri gösterir.

İstemci bağlantı kanallarıyla çalışma


Bir kuyruk yöneticisine istemci-bağlantı kanalları eklemek ve silmek, özellikleri görüntülemek ve ayarlamak ve kanalın yetki kayıtlarını yönetmek için IBM MQ Console içindeki istemci bağlantı kanalları pencere bileşenini kullanabilirsiniz.

Başlamadan önce



Bir istemci bağlantı kanalları yaratmanız gerekir. Bu gereçlerin kullanılması için, gereç yaratmanız gerekir. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. [“IBM MQ nesneleriyle çalışma” sayfa 95](#).





Bu görev hakkında

İstemci-bağlantı kanalları gereci, belirli bir kuyruk yöneticisi için varolan istemci bağlantısı kanallarını listeler. Tek tek istemci-bağlantı kanallarını çalışmak için listeden seçebilirsiniz.

 z/OS üzerindeki bir istemci-bağlantı kanalı için yetki kayıtlarını görüntüleyemez ya da düzenleyemezsiniz.

Yordam

- İstemci-bağlantı kanalı eklemek için:
 - a) İstemci bağlantı kanalları pencere aracı araç çubuğunda oluştur simgesini   tıklatın.
 - b) İstemci-bağlantı kanalı için bir ad girin. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
 - c) Bağlantı adını belirtin. Bağlantı adı, hedef kuyruk yöneticisini bulunduran bilgisayarın adıdır. Biçim şöyledir: *bilgisayar_adi(kapı_numarası)*; burada *bilgisayar_adi* , hedef kuyruk yöneticisini barındıran bilgisayarın adı ya da IP adresidir ve *kapı_numarası* , hedef kuyruk yöneticisinin dinleyicisinin kullandığı kapıdır.
 - d) **Oluştur**'u tıklatın. Yeni istemci-bağlantı kanalı yaratılır.

- İstemci-bağlantı kanalını silmek için:
 - a) İstemci-bağlantı kanalları pencere bileşenindeki listeden silmek istediğiniz istemci bağlantısı kanalını seçin.
 - b) Pencere ögesi araç çubuğunda sil simgesini  tıklatın.
 - c) İstemci-bağlantı kanalını silmek istediğinizi doğrulamak için **Sildüğmesini** tıklatın. İstemci-bağlantı kanalı silindi.
- İstemci-bağlantı kanalının özelliklerini görüntülemek ve düzenlemek için:
 - a) İstemci-bağlantı kanalları pencere bileşenindeki istemci bağlantısı kanalını seçin.
 - b) İstemci-bağlantı kanalları pencere bileşeni araç çubuğunda özellikler simgesini  tıklatın. Diğer bir seçenek olarak, istemci-bağlantı kanalını çift tıklatın.
 - c)  Özellikleri görüntüleyin ve gereken şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir. For information about a property, view the property information in MQ Explorer belgelerindeki Kanal özellikleri.
- İstemci-bağlantı kanalına ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:
 - a) İstemci-bağlantı kanalları pencere bileşenindeki istemci bağlantısı kanalını seçin.
 - b) Tıklatın  ... > **Yetki Kayıtlarını Yönet**. Yetki kayıtları, kullanıcıların ve yöneticilerin seçilen istemci-bağlantı kanalı üzerinde sahip olduğu izinleri gösterir.

Kimlik doğrulama bilgileriyle çalışma

Bir kuyruk yöneticisinde kimlik doğrulama bilgileri nesnelere eklemek ve silmek için IBM MQ Console içindeki kimlik doğrulama bilgileri pencere bileşenini kullanabilirsiniz. Ayrıca, özellikleri görüntüleyebilir ve ayarlayabilir ve nesnelere ilgili yetki kayıtlarını yönetebilirsiniz.


Başlamadan önce

Bir kimlik doğrulama bilgileri pencere bileşeni kullanabilmeniz için önce oluşturmalısınız. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. ["IBM MQ nesnelere çalışma" sayfa 95.](#)


Bu görev hakkında

Kimlik doğrulama bilgileri pencere bileşeni, 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 denetimi gerçekleştirmek için gereken tanımlamaları ve kullanıcı kimliği ve parola denetimini etkinleştirmek için gerekli tanımlamalar içerir.

 IDPW LDAP 'ı kullanamazsınız, z/OS' ta bir kimlik doğrulama bilgisi nesnesine ilişkin yetki kayıtlarını görüntüleyebilir ya da düzenleyebilirsiniz.

Yordam

- Bir kimlik doğrulama bilgileri nesnesi eklemek için:
 - a) Kimlik doğrulama bilgileri pencere bileşeni araç çubuğunda oluştur simgesini  tıklatın.
 - 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.
- **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 ek gereksinim yoktur.
- **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.

e) **Oluştur**'u tıklatın.

- Bir kimlik doğrulama bilgileri nesnesini silmek için:

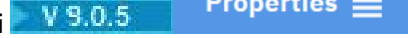
a) Pencere bileşenindeki listeden silmek istediğiniz kimlik doğrulama bilgileri nesnesini seçin.

b) Pencere ögesi araç çubuğunda sil simgesini  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) Gereçte kimlik doğrulama bilgileri nesnesini seçin.

b) Pencere ögesi araç çubuğunda özellikler simgesi  tıklatın. Diğer bir seçenek olarak, kimlik doğrulama bilgileri nesnesini çift tıklatın.

c) Özellikleri görüntüleyin ve gereken şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir.

- Bir kimlik doğrulama bilgileri nesnesine ilişkin yetki kayıtlarını görüntülemek ve düzenlemek için:

a) Kimlik doğrulama bilgileri pencere bileşenindeki kimlik doğrulama bilgileri nesnesini seçin.

b) Tıklatın  > **Yetki Kayıtlarını Yönet**. Yetki kayıtları, kullanıcıların ve yöneticilerin seçilen kimlik doğrulama bilgileri nesnesi üzerinde sahip olduğu izinleri gösterir.

Aboneliklerle çalışma

Bir kuyruk yöneticisine abonelik eklemek ve bunları silmek, özellikleri görüntülemek ve ayarlamak ve aboneliklere ilişkin yetki kayıtlarını yönetmek için IBM MQ Console içindeki abonelikler pencere aracını kullanabilirsiniz.

Başlamadan önce

Bir abonelik pencere aracını kullanabilmeniz için önce bir pencere bileşeni oluşturmalısınız. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. [“IBM MQ nesneleriyle çalışma” sayfa 95.](#)

Bu görev hakkında



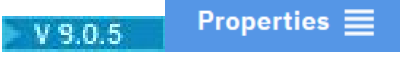
Abonelikler bir kuyruk yöneticisine verilir ve abonenin almak istediği yayınlarla ilgili bilgileri içerir:

- Abonenin ilgilendiği konu dizgisi; genel arama karakterleri kullanılırsa bu konu birden çok konu dizgisine çözümleyebilir.

- Yayınlanan iletilere uygulanacak isteğe bağlı bir seçim dizgisi.
- Seçilen yayınların yerleştirdiği kuyruğun adı.

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

Yordam

- Bir abonelik nesnesi eklemek için:
 - a) Abonelikler pencere aracı araç çubuğunda oluştur simgesini  tıklatın.
 - b) Nesnenin adını belirtin. Geçerli karakterler, harfler ve sayılar ve ".", "/", "_" ve "%" karakterleridir.
 - c) **Yönetilen** ya da **Sağlanan** için bir **Hedef sınıfı** seçin. **Yönetildi** seçeneğini belirlerseniz, yerel kuyruk yöneticisinde bir hedef yaratılır.
 - d) If you select a destination class of **Sağlandı**, in the **Hedef** field, specify the name of the queue to which messages for this subscription are forwarded.
 - e) **Konu dizesi** alanında, abone olmak için konu dizesini belirtin.
 - f) Bir **Genel arama karakteri kullanımı** ayarı seçin. Genel arama karakterlerinin dizgilerin kısımlarını temsil ettiğini belirtmek için **Karakter düzeyinde genel arama karakteri** seçeneğini belirleyin. Genel arama karakterlerinin, konu sıradüzeninin bölümlerini temsil ettiğini belirtmek için **Konu düzeyi genel arama karakteri** seçeneğini belirleyin.
 - g) Bir **Kapsam** seçin. Select **Tümü** so the subscription is forwarded to all queue managers directly connected through a publish/subscribe collective or hierarchy. Yalnızca bu kuyruk yöneticisi içinde konu üzerinde yayınlanan iletilerin iletilmesi için **Kuyruk yöneticisi** seçeneğini belirleyin.
 - h) İsteğe bağlı: Bir **seçici** belirtin. Seçim dizgisi, bir yayınla eşleşip eşleşmediğini belirlemek için yayına uygulanan bir ifadedir.
 - i) **Oluştur**'u tıklatın.
- Bir abonelik nesnesini silmek için:
 - a) Abonelikler pencere aracındaki listeden silmek istediğiniz abonelik nesnesini seçin.
 - b) Pencere öğesi araç çubuğunda sil simgesini  tıklatın.
 - c) Abonelik nesnesini silmek istediğinizi doğrulamak için **Sildüğmesini** tıklatın. Nesne silindi.
- Bir abonelik nesnesine ilişkin özellikleri görüntülemek ve düzenlemek için:
 - a) Gereçte abonelik nesnesini seçin.
 - b) Pencere öğesi araç çubuğunda özellikler simgesi  tıklatın. Alternatif olarak, abonelik nesnesini çift tıklatın.
 - c) Özellikleri görüntüleyin ve gereken şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir.

 **V 9.0.1**

Kanal kimlik doğrulama kayıtlarıyla çalışma

Bir kuyruk yöneticisine kanal kimlik doğrulama kayıtları eklemek ve silmek için IBM MQ Console içindeki kanal kimlik doğrulama kayıtları pencere bileşenini kullanabilirsiniz. Ayrıca, kanal doğrulama kayıtlarına ilişkin özellikleri de görüntüleyebilir ve ayarlayabilirsiniz.

Başlamadan önce



Bir kanal kimlik doğrulama kayıtları pencere bileşenini kullanabilmeniz için önce oluşturmalısınız. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. [“IBM MQ nesneleriyle çalışma” sayfa 95.](#)

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 kimlik doğrulama kayıtları](#).

Yordam

- SSL/TLS ayırt edici ad kimliğine sahip bir kanal kimlik doğrulaması kaydı eklemek için bkz. [“SSL/TLS Ayırt Edici Ad tanıtıcısı ile kanal kimlik doğrulama kayıtları oluşturma” sayfa 105](#).
- Bir istemci uygulaması kullanıcı kimliği kimliğine sahip bir kanal kimlik doğrulaması kaydı eklemek için bkz. [“İstemci uygulaması kullanıcı kimliği tanıtıcısı ile kanal kimlik denetimi kayıtları yaratılması” sayfa 106](#).
- Uzak kuyruk yöneticisi adı kimliğine sahip bir kanal kimlik doğrulaması kaydı eklemek için bkz. [“Uzak kuyruk yöneticisi adı kimliği ile kanal doğrulama kayıtları yaratılması” sayfa 107](#).
- Adres kimliğine sahip bir kanal kimlik doğrulaması kaydı eklemek için bkz. [“IP adresi tanıtıcısı ile kanal doğrulama kayıtları yaratılması” sayfa 108](#).
- Bir kanal kimlik doğrulaması kaydını silmek için:
 - a) Kanal kimlik doğrulama kayıtları pencere bileşenindeki listeden silmek istediğiniz kanal kimlik doğrulama kaydını seçin.
 - b) Pencere ögesi araç çubuğunda sil simgesini  tıklatın.
 - c) Kanal kimlik doğrulama kaydını silmek istediğinizi doğrulamak için **Sildüğmesini** tıklatın. Kanal kimlik doğrulama kaydı silindi.
- Bir kanal kimlik doğrulama kaydının özelliklerini görüntülemek ve düzenlemek için:
 - a) Kanal kimlik doğrulama kaydı pencere bileşenindeki listeden düzenlemek istediğiniz kanal kimlik doğrulama kaydını seçin.
 - b) Pencere ögesi araç çubuğunda özellikler simgesi  tıklatın. Diğer bir seçenek olarak, kanal kimlik doğrulama kaydını çift tıklatın.
 - c) Özellikleri görüntüleyin ve gereken şekilde düzenleyin. Özellik metin kutusu devre dışı bırakılırsa, özellik salt okunur olur ya da yalnızca komut satırından düzenlenebilir.

SSL/TLS Ayırt Edici Ad tanıtıcısı ile kanal kimlik doğrulama kayıtları oluşturma

Kanal doğrulama kayıtları pencere bileşenini, bir SSL/TLS Ayırt Edici Ad kimliğiyle kanal doğrulama kayıtlarını izin vermek, engellemek ve uyarlamak için kullanabilirsiniz. SSL/TLS ayırt edici ad kimliği, belirli bir Ayırt Edici Ad içeren bir SSL ya da TLS kişisel sertifikası veren kullanıcılarla eşleşir.

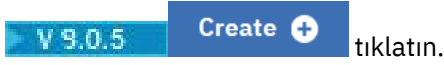
Başlamadan önce

Bir kanal kimlik doğrulama kayıtları pencere bileşenini kullanabilmeniz için önce oluşturmalısınız. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. [“IBM MQ nesneleriyle çalışma” sayfa 95](#).

Yordam

- Kanal kimlik doğrulaması kaydı eklemek için:

a) Kanal kimlik doğrulama kaydı pencere bileşeni araç çubuğunda oluştur simgesini



tıklatın.

b) Kanal doğrulama kaydında istediğiniz kural tipini belirtmek için **Kural Tipi** ' ne seçin:

- Gelen bağlantılara erişim izni vermek için **İzin Ver** seçeneğini belirleyin.
- Gelen bağlantılara erişimi engellemek için **Block** (Engelle) seçeneğini belirleyin.
- Engellenecek gelen bağlantılara erişim konusunda uyarmak için **Uyar** seçeneğini belirleyin. Bağlantıya izin verilir ve bir hata iletisi raporlanır. Olaylar yapılandırıldıysa, engellenecek neyin ayrıntılarını gösteren bir olay iletisi yaratılır. Yalnızca eşleşen kurallar raporlanır.

c) Listedeki **SSL/TLS Ayırt Edici Adı** kimlik tipini seçin.

d) **İleri** düğmesini tıklatın.

e) Bir **Kanal profili** belirtin.

Kanal tanıtımı, kanal kimlik doğrulamasını ayarladığınız kanalların ya da kanalların adıdır. Profil, bir kanal aralığını bloke edebilen için joker karakterler içerebilir. For example, the profile `a1phadelta*` blocks channels named `alphadelta1`, `alphadelta2`, `alphadelta3` and so on.

f) **Eşdüzey adı** belirtin. Örneğin, `CN=John Smith, O=IBM, OU=Test, C=GB`. Eş adları hakkında daha fazla bilgi için bkz. [SSLPEER değerleri için WebSphere MQ kuralları](#).

g) İsteğe bağlı: Kullanılan **Adres** süzgecini belirtin. Adres, kanalın diğer ucunda beklenen IP adresidir.

h) İsteğe bağlı: **SSL sertifika veren adı** değerini belirtin. SSL sertifika veren adı, SSL/TLS sertifikasının yayınlaması gereken sertifika yetkilisinin adıdır.

i) İsteğe bağlı: **İleri**'yi tıklatın.

j) İsteğe bağlı: Bir **İzin Ver** kural tipi için, isteğe bağlı olarak kanal kimlik doğrulaması kaydı için **Kullanıcı kaynağı** belirtebilirsiniz. Kullanıcı kaynağı, gelen bağlantı SSL/TLS Ayırt Edici Adı ile eşleştiğinde kullanılan kullanıcı kimliğinin kaynağını belirtir.

- **Kanal** seçeneği, eşlemelerle eşleşen gelen bağlantıların, akıcı kullanıcı kimliğini ya da kanal nesnesinde tanımlı olan herhangi bir kullanıcıyı kullandığını belirtir.
- **Eşle** seçeneği, eşlemelerle eşleşen gelen bağlantıların, **MCA kullanıcı kimliği** alanında belirtilen kullanıcı kimliğini kullandığından emin olur.

k) İsteğe bağlı: **İleri**'yi tıklatın.

l) İsteğe bağlı: Kanal kimlik doğrulaması kaydı için bir **Tanım** belirtin.

m) **Oluştur**'u tıklatın. Yeni kanal kimlik doğrulama kaydı yaratılır.

V 9.0.1 İstemci uygulaması kullanıcı kimliği tanıtıcısı ile kanal kimlik denetimi kayıtları yaratılması

Kanal kimlik doğrulama kayıtları pencere aracını, bir istemci uygulaması kullanıcı kimliği kimliğiyle kanal doğrulama kayıtlarını izin vermek, engellemek ve uyarlamak için kullanabilirsiniz. İstemci uygulaması kullanıcı kimliği kimlik tanıtıcısı, istemci bağlantı kanalından istemci uygulama tanıtıcılarıyla eşleşiyor.

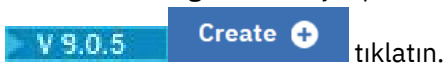
Başlamadan önce

Bir kanal kimlik doğrulama kayıtları pencere bileşenini kullanabilmeniz için önce oluşturmalısınız. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. [“IBM MQ nesneleriyle çalışma” sayfa 95.](#)

Yordam

- Kanal kimlik doğrulaması kaydı eklemek için:

a) Kanal kimlik doğrulama kaydı pencere bileşeni araç çubuğunda oluştur simgesini



tıklatın.

b) Kanal doğrulama kaydında istediğiniz kural tipini belirtmek için **Kural Tipi** ' ne seçin:

- Gelen bağlantılara erişim izni vermek için **İzin Ver** seçeneğini belirleyin.
 - Gelen bağlantılara erişimi engellemek için **Block** (Engelle) seçeneğini belirleyin.
 - Engellenecek gelen bağlantılara erişim konusunda uyarmak için **Uyar** seçeneğini belirleyin. Bağlantıya izin verilir ve bir hata iletisi raporlanır. Olaylar yapılandırıldıysa, engellenecek neyin ayrıntılarını gösteren bir olay iletisi yaratılır. Yalnızca eşleşen kurallar raporlanır.
- c) Listedeki **İstemci uygulaması kullanıcı kimliği** kimlik tipini seçin.
- d) **İleri**'yi tıklayın.
- e) Bir **Kanal profili** belirtin.
Kanal tanıtımı adı, kanal kimlik doğrulamasını ayarladığınız kanalların ya da kanalların kümesinin adıdır. Profil, bir kanal aralığını bloke edebilmenizi için joker karakterler içerebilir. For example, the profile `alphaDelta*` blocks channels named `alphaDelta1`, `alphaDelta2`, `alphaDelta3` and so on.
- f) **İstemci kullanıcı kimliği** değerini belirtin. İstemci kullanıcı kimliği, izin vermek, engellemek ya da uyarmak istediğiniz istemcinin kullanıcı kimliğidir.
- g) İsteğe bağlı: Kullanılan **Adres** süzgecini belirtin. Adres, kanalın diğer ucunda beklenen IP adresidir.
- h) İsteğe bağlı: **İleri**'yi tıklayın.
- i) İsteğe bağlı: Bir **İzin Ver** kural tipi için, isteğe bağlı olarak kanal kimlik doğrulaması kaydı için **Kullanıcı kaynağı** belirtebilirsiniz. Kullanıcı kaynağı, gelen bağlantı istemci kullanıcı kimliğiyle eşleştğinde kullanılan kullanıcı kimliğinin kaynağını belirtir.
- **Kanal** seçeneği, eşlemelerle eşleşen gelen bağlantıların, akıcı kullanıcı kimliğini ya da kanal nesnesinde tanımlı olan herhangi bir kullanıcıyı kullandığını belirtir.
 - **Eşle** seçeneği, eşlemelerle eşleşen gelen bağlantıların, **MCA kullanıcı kimliği** alanında belirtilen kullanıcı kimliğini kullandığından emin olur.
- j) İsteğe bağlı: **İleri**'yi tıklayın.
- k) İsteğe bağlı: Kanal kimlik doğrulaması kaydı için bir **Tanım** belirtin.
- l) **Oluştur**'u tıklayın. Yeni kanal kimlik doğrulama kaydı yaratılır.

V 9.0.1

Uzak kuyruk yöneticisi adı kimliği ile kanal doğrulama kayıtları yaratılması

Uzak bir kuyruk yöneticisi adı kimliğiyle kanal doğrulama kayıtlarını izin vermek, engellemek ve uyarlamak için kanal kimlik doğrulama kayıtları pencere bileşenini kullanabilirsiniz. Uzak kuyruk yöneticisi adı kimliği, belirtilen kuyruk yöneticisiniyle eşleşiyor.

Başlamadan önce

Bir kanal kimlik doğrulama kayıtları pencere bileşenini kullanabilmeniz için önce oluşturmalısınız. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. [“IBM MQ nesneleriyle çalışma” sayfa 95.](#)

Yordam

- Kanal kimlik doğrulaması kaydı eklemek için:

- a) Kanal kimlik doğrulama kaydı pencere bileşeni araç çubuğunda oluştur simgesini



tıklatın.

- b) Kanal doğrulama kaydında istediğiniz kural tipini belirtmek için **Kural Tipi** ' ne seçin:

- Gelen bağlantılara erişim izni vermek için **İzin Ver** seçeneğini belirleyin.
- Gelen bağlantılara erişimi engellemek için **Block** (Engelle) seçeneğini belirleyin.
- Engellenecek gelen bağlantılara erişim konusunda uyarmak için **Uyar** seçeneğini belirleyin. Bağlantıya izin verilir ve bir hata iletisi raporlanır. Olaylar yapılandırıldıysa, engellenecek neyin ayrıntılarını gösteren bir olay iletisi yaratılır. Yalnızca eşleşen kurallar raporlanır.

- c) Listedeki **Uzak kuyruk yöneticisi adı** kimlik tipini seçin.

- d) **İleri** düğmesini tıklatın.
- e) Bir **Profil Adı** belirtin.
Profil adı, kanal kimlik doğrulamasını ayarladığınız kanalların ya da kanalların adıdır. Profil, bir kanal aralığını bloke edebilmeyi için joker karakterler içerebilir. For example, the profile a1phade1ta* blocks channels named alphadelta1, alphadelta2, alphdelta3 and so on.
- f) **Kuyruk yöneticisi adı** belirtin. Kuyruk yöneticisi adı, izin vermek, engellemek ya da uyarmak istediğiniz uzak kuyruk yöneticisinin adını belirtir.
- g) İsteğe bağlı: Kullanılan **Adres** süzgecini belirtin. Adres, kanalın diğer ucunda beklenen IP adresidir.
- h) İsteğe bağlı: **İleri**'yi tıklatın.
- i) İsteğe bağlı: Bir **İzin Ver** kural tipi için, isteğe bağlı olarak kanal kimlik doğrulaması kaydı için **Kullanıcı kaynağı** belirtebilirsiniz. Kullanıcı kaynağı, gelen bağlantı uzak kuyruk yöneticisi adıyla eşleştiğinde kullanılan kullanıcı kimliğinin kaynağını belirtir.
- **Kanal** seçeneği, eşlemelerle eşleşen gelen bağlantıların, akıcı kullanıcı kimliğini ya da kanal nesnesinde tanımlı olan herhangi bir kullanıcıyı kullandığını belirtir.
 - **Eşle** seçeneği, eşlemelerle eşleşen gelen bağlantıların, **MCA kullanıcı kimliği** alanında belirtilen kullanıcı kimliğini kullandığından emin olur.
- j) İsteğe bağlı: **İleri**'yi tıklatın.
- k) İsteğe bağlı: Kanal kimlik doğrulaması kaydı için bir **Tanım** belirtin.
- l) **Oluştur**'u tıklatın. Yeni kanal kimlik doğrulama kaydı yaratılır.

V 9.0.1

IP adresi tanıtıcısı ile kanal doğrulama kayıtları yaratılması

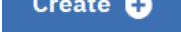
Kanal kimlik doğrulama kayıtları pencere aracını kullanarak, bir adres kimliği ile kanal kimlik doğrulama kayıtlarını izin vererek, engellemeyi ve uyarı olarak kullanabilirsiniz. Adres kimliği belirli IP adresleriyle eşleşir.

Başlamadan önce

Bir kanal kimlik doğrulama kayıtları pencere bileşenini kullanabilmeniz için önce oluşturmalısınız. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. [“IBM MQ nesneleriyle çalışma” sayfa 95.](#)

Yordam

- Kanal kimlik doğrulaması kaydı eklemek için:

- a) Kanal kimlik doğrulama kaydı pencere bileşeni araç çubuğunda oluştur simgesini  tıklatın.
- b) Kanal doğrulama kaydında istediğiniz kural tipini belirtmek için **Kural Tipi** ' ne seçin:
- Gelen bağlantılara erişim izni vermek için **İzin Ver** seçeneğini belirleyin.
 - Gelen bağlantılara erişimi engellemek için **Block** (Engelle) seçeneğini belirleyin.
 - Engellenecek gelen bağlantılara erişim konusunda uyarmak için **Uyar** seçeneğini belirleyin. Bağlantıya izin verilir ve bir hata iletisi raporlanır. Olaylar yapılandırıldıysa, engellenecek neyin ayrıntılarını gösteren bir olay iletisi yaratılır. Yalnızca eşleşen kurallar raporlanır.
- c) Listedeki **Adres** kimlik tipini seçin.
- d) **İleri** düğmesini tıklatın.
- e) İsteğe bağlı: Bir **Öbek** ya da **Uyarma** kural tipi için, **Ne zaman eşleştirileceğini** belirtin. Bu seçenekler arasından seçim yapabilirsiniz:
- **Dinleyiciye**. Bu seçenek, dinleyiciye kuralla eşleşmeyi dener.
 - **Kanalda**. Bu seçenek, kanaldaki kuralla eşleşmeyi dener.
- f) Bir **Profil Adı** belirtin.

Profil adı, kanal kimlik doğrulamasını ayarladığınız kanalların ya da kanalların adıdır. Profil, bir kanal aralığını bloke edebilmeyi için joker karakterler içerebilir. For example, the profile a1phade1ta* blocks channels named alphadelta1, alphadelta2, alphdelta3 and so on.

- g) Bir **Adres** belirtin. Adres, IP adresi ya da IP adresleri için izin verilen ya da engellenen, virgülle ayrılmış bir listedir.
- h) İsteğe bağlı: **İleri**'yi tıklayın.
- i) İsteğe bağlı: Bir **İzin Ver** kural tipi için, isteğe bağlı olarak kanal kimlik doğrulaması kaydı için **Kullanıcı kaynağı** belirtebilirsiniz. Kullanıcı kaynağı, gelen bağlantı uzak kuyruk yöneticisi adıyla eşleştiğinde kullanılan kullanıcı kimliğinin kaynağını belirtir.
 - **Kanal** seçeneği, eşlemelerle eşleşen gelen bağlantıların, akıcı kullanıcı kimliğini ya da kanal nesnesinde tanımlı olan herhangi bir kullanıcıyı kullandığını belirtir.
 - **Eşle** seçeneği, eşlemelerle eşleşen gelen bağlantıların, **MCA kullanıcı kimliği** alanında belirtilen kullanıcı kimliğini kullandığından emin olur.
- j) İsteğe bağlı: **İleri**'yi tıklayın.
- k) İsteğe bağlı: Kanal kimlik doğrulaması kaydı için bir **Tanım** belirtin.
- l) **Oluştur**'u tıklayın. Yeni kanal kimlik doğrulama kaydı yaratılır.


V 9.0.1 **Son atanan kullanıcı kimliği kimliğine sahip kanal kimlik doğrulama kayıtları oluşturma**

Son atanan kullanıcı kimliği kimliğine sahip engelleme ve uyarı kanalı kimlik doğrulama kayıtları oluşturmak için kanal kimlik doğrulama kayıtları pencere aracını kullanabilirsiniz. Son atanan kullanıcı kimliği kimliği, bir sunucu kanalındaki belirtilen kullanıcı kimlikleri listeleriyle eşleşir.

Başlamadan önce

Bir kanal kimlik doğrulama kayıtları pencere bileşenini kullanabilmeniz için önce oluşturmalısınız. IBM MQ nesne pencere araçlarını oluşturma hakkında daha fazla bilgi için bkz. [“IBM MQ nesneleriyle çalışma” sayfa 95.](#)

Yordam

- Kanal kimlik doğrulaması kaydı eklemek için:
 - a) Kanal kimlik doğrulama kaydı pencere bileşeni araç çubuğunda oluştur simgesini  tıklayın.
 - b) Kanal doğrulama kaydında istediğiniz kural tipini belirtmek için **Kural Tipi** ' ne seçin:
 - Gelen bağlantılara erişimi engellemek için **Block** (Engelle) seçeneğini belirleyin.
 - Engellenecek gelen bağlantılara erişim konusunda uyarılmak için **Uyar** seçeneğini belirleyin. Bağlantıya izin verilir ve bir hata iletisi raporlanır. Olaylar yapılandırıldıysa, engellenecek neyin ayrıntılarını gösteren bir olay iletisi yaratılır. Yalnızca eşleşen kurallar raporlanır.
 - c) Listedeki **Son atanan kullanıcı kimliği** kimlik tipini seçin.
 - d) **İleri** düğmesini tıklayın.
 - e) Bir **Profil Adı** belirtin.

Profil adı, kanal kimlik doğrulamasını ayarladığınız kanalların ya da kanalların adıdır. Profil, bir kanal aralığını bloke edebilmeyi için joker karakterler içerebilir. For example, the profile a1phade1ta* blocks channels named alphadelta1, alphadelta2, alphdelta3 and so on.
 - f) **Kullanıcı listesi** seçeneğini belirleyin. Kullanıcı listesi, kanaldan engellenecek kullanıcı kimliklerinin virgülle ayrılmış bir listesidir.
 - g) İsteğe bağlı: **İleri**'yi tıklayın.
 - h) İsteğe bağlı: Kanal kimlik doğrulaması kaydı için bir **Tanım** belirtin.
 - i) **Oluştur**'u tıklayın. Yeni kanal kimlik doğrulama kaydı yaratılır.

V 9.0.1 Yetki kayıtlarıyla çalışma

Bu gruba ilişkin bir yetki kaydı belirterek, grupların kuyruk yöneticilerine ve IBM MQ nesnelere erişimleri için erişim denetimi yapabilirsiniz.

Bu görev hakkında

Yetki kayıtlarını kullanarak, bir grup ileti sistemi kullanıcısının belirli bir kuyruk yöneticisine ya da IBM MQ nesnesine sahip olduğu erişimi ince ayar yapabilirsiniz. Yetki kaydını, yapılandırdığınız gerçek izinler nesne tipine bağlı olsa da, aynı yordamı kullanarak tüm nesne tipleri için aynı şekilde yetki kaydını yapılandırabilirsiniz.

Örneğin, aşağıdaki resimlerde gösterildiği gibi, kuyruk yöneticisi ve kuyruk için kullanılacak farklı izinlerin karşıtlığı: V 9.0.5

Authority records for 'qm3'

Entity name	Entity type
mqm	Group
mqsystem	User


Total: 2 Last updated: 3:40:18 PM

Administration	Context	MQI
<input checked="" type="checkbox"/> Change	<input checked="" type="checkbox"/> Set all context	<input checked="" type="checkbox"/> Alternate user authority
<input checked="" type="checkbox"/> Delete	<input checked="" type="checkbox"/> Set identity context	<input checked="" type="checkbox"/> Connect
<input checked="" type="checkbox"/> Display		<input checked="" type="checkbox"/> Inquire
<input checked="" type="checkbox"/> Ctrl		<input checked="" type="checkbox"/> Set
		<input checked="" type="checkbox"/> System

Check all Uncheck all Close Save

V 9.0.5

Authority records for 'q1' on qm3

Delete  1 item selected Cancel	
▲ Entity name	Entity type
mqm	Group
mqsystem	User

<input checked="" type="checkbox"/> Change	<input checked="" type="checkbox"/> Pass all context	<input checked="" type="checkbox"/> Browse
<input checked="" type="checkbox"/> Clear	<input checked="" type="checkbox"/> Pass identity context	<input checked="" type="checkbox"/> Inquire
<input checked="" type="checkbox"/> Delete	<input checked="" type="checkbox"/> Set all context	<input checked="" type="checkbox"/> Get
<input checked="" type="checkbox"/> Display	<input checked="" type="checkbox"/> Set identity context	<input checked="" type="checkbox"/> Put
		<input checked="" type="checkbox"/> Set

[Check all](#) [Uncheck all](#) [Close](#) [Save](#)

 You cannot work with authority records on z/OS.

Yordam

- Bir IBM MQ nesnesine ilişkin yetki kaydını görüntülemek ya da düzenlemek için:
 - a) Panodaki bir gereçte nesneyi seçin. İlişkili kuyruk yöneticisinin çalışır durumda olması gerekir.
 - b) Uygun pencere aracı araç çubuğundan, ... > **Yetki Kayıtlarını Yönet**.
 - c) Yetki kaydını görüntülemek istediğiniz grubu seçin. Bu gruba ilişkin yetkiler görüntülenir.
 - d) Yetkileri gerektiği şekilde seçin ya da temizleyin. Yetki kaydı yaratmakta olduğunuz nesnenin tipine bağlı olarak farklı yetkiler kullanılabilir.
 - e) **Kaydet**'i tıklatın.
- Bir kuyruk yöneticisine ilişkin yaratma yetkisi kaydını görüntülemek ya da düzenlemek için:
 - a) Panodaki kuyruk yöneticisi gerecinde kuyruk yöneticisini seçin. Kuyruk yöneticisi çalışıyor olmalıdır.
 - b) Gereç araç çubuğundan şu öğeleri seçin: ... > **Yetki Kayıtlarını Yaratmayı Yönet**.
 - c) Yaratma yetkisi kaydını görüntülemek istediğiniz grubu seçin. Bu gruba ilişkin yetkiler görüntülenir.
 - d) Yaratma ya da temizleme yetkilerine ilişkin yetkiyi seçin.
 - e) **Kaydet**'i tıklatın.
- IBM MQ nesnesine ilişkin bir yetki kaydı oluşturmak için:
 - a) Panodaki bir gereçte IBM MQ nesnesini seçin. İlişkili kuyruk yöneticisinin çalışır durumda olması gerekir.

b) Gereç araç çubuğundan şu öğeleri seçin: ... > **Yetki Kayıtlarını Yönet**.

c) Artı simgesini  tıklatın.

d) Yetki kaydını yaratmakta olduğunuz kullanıcı ya da grubun adını belirtin. Kullanıcı ya da grup var olmalıdır.

e) Varlığın bir kullanıcı mı, yoksa bir grup mu olduğunu belirtmek için **Varlık Tipi** ' yi seçin.

f) **Oluştur**'u tıklatın.

g) Kullanıcının ya da grubun sahip olmasını istediğiniz yetkileri seçin ya da bu yetkilerin işaretini kaldırın. Her nesne tipi için farklı yetkiler vardır.

h) **Kaydet**'i tıklatın.

- Bir kuyruk yöneticisine nesne yaratmak için yetki kaydı yaratmak için:

a) Panodaki bir gereçte kuyruk yöneticisini seçin. Kuyruk yöneticisi çalışıyor olmalıdır.

b) Gereç araç çubuğundan şu öğeleri seçin: ... > **Yetki Kayıtlarını Yaratmayı Yönet**.

c) Oluştur simgesini  tıklatın.

d) Yetki kaydını yaratmakta olduğunuz kullanıcı ya da grubun adını belirtin. Kullanıcı ya da grup var olmalıdır.

e) Varlığın bir kullanıcı mı, yoksa bir grup mu olduğunu belirtmek için **Varlık Tipi** ' yi seçin.

f) **Oluştur**'u tıklatın.

g) Kullanıcı ya da grubun sahip olmasını istediğiniz yaratma yetkilerini seçin ya da temizleyin.

h) **Kaydet**'i tıklatın.

V 9.0.1

Sistem kaynağı kullanımını izleme


Kuyruk yöneticilerine ilişkin izleme verilerini görüntülemek için IBM MQ Console içindeki Grafikler pencere bileşenini kullanıyorsunuz.

Bu görev hakkında

Gösterge panonuzun bir Grafikler pencere bileşenini ekliyorsunuz ve kaynak kullanımının belirli bir yönünü izlemek için bu pencere bileşenini yapılandırın. Farklı verileri görüntülemek için Grafikler pencere bileşeninin birçok örneğini oluşturabilirsiniz. Veriler, bir grafik biçiminde görüntülenir.

Veriler 10 saniyelik aralıklarla toplanır. Grafiğin X eksenini bir zaman çizelgesi görüntüler. Y ekseninde, bakmakta olduğunuz kaynağa uygun birimler görüntülenir. Y eksenini, iade edilen verileri barındırmak için dinamik olarak yeniden boyutlandırılır.

Bir grafik pencere bileşenini yapılandırmadan önce en az bir çalışan kuyruk yöneticisininiz olmalıdır.

 z/OS üzerinde sistem kaynağı kullanımını izleyemezsiniz.

Yordam

1. Gösterge panonuzun bir Grafik pencere bileşenini ekleyin:

a) Pencere öğesi ekle simgesini  tıklatın.

b) **Grafikler** seçeneğini belirleyin.

2. Verileri göstermek için Grafikler pencere aracını yapılandırın:

a) Grafikler pencere bileşeninin başlık çubuğunda yapılandır simgesini  tıklatın.

b) İsteğe bağlı: Bir **Pencere öğesi başlığı** girin. Bu başlık, pencere bileşeninin başlık çubuğunda gösterilir.

c) İzlenecek **Kaynak sınıfı** ' yı seçin:

Platform merkezi işlem birimleri

CPU ' nun kullanımını izleyin.

Platform kalıcı veri depoları

Disk kaynağı kullanımını izleyin.

API kullanım istatistikleri

API çağrılarını izleyin.

Kuyruk başına API kullanım istatistikleri

Tek tek kuyruklara göre API çağrılarını izleyin. When you choose this class, you specify the queue name to monitor in the **Nesne** field.

d) İzlemek için **Kaynak tipi** ' yi seçin.

Seçilebilecek kaynak tipleri, seçilen kaynak sınıfına bağlı olarak değişir. Aşağıdaki çizelge kaynak tiplerini gösterir:

<i>Çizelge 6. Kaynak tipleri</i>		
Sınıf	Tip	Tanım
Platform merkezi işlem birimleri	CPU performansı-platform genişliği	CPU ' lar ve belleğe ilişkin başarımların verilerini görüntülemek için bu tipi seçin.
	CPU performansı-çalışan kuyruk yöneticisi	İzlemekte olduğunuz kuyruk yöneticilerine ilişkin CPU ' lar ve bellek için başarımların verilerini görüntülemek üzere bu tipi seçin. İzlemeniz için kuyruk yöneticisinin çalışıyor olması gerekir. Sonuçları birden çok kuyruk yöneticisinden izliyorsanız, grafikteki performans verilerini ayırt etmek için farklı renkler kullanılır.
Platform kalıcı veri depoları	Disk kullanımı-altyapı genişliği	Genel disk kullanımına ilişkin başarımların verilerini görüntülemek için bu tipi seçin.
	Disk kullanımı-çalışan kuyruk yöneticileri	İzlemekte olduğunuz kuyruk yöneticileriyle ilgili disk kullanımına ilişkin başarımların verilerini görüntülemek için bu tipi seçin. İzlemeniz için kuyruk yöneticisinin çalışıyor olması gerekir. Sonuçları birden çok kuyruk yöneticisinden izliyorsanız, grafikteki performans verilerini ayırt etmek için farklı renkler kullanılır.
	Disk kullanımı-kuyruk yöneticisi kurtarma günlüğü	İzlemekte olduğunuz her bir kuyruk yöneticisinin kurtarma günlüğü için disk saklama alanının nasıl kullanılmakta olduğunu görmek için bu tipi seçin.
API kullanım istatistikleri	MQCONN ve MQDISC	MQCONN ve MQDISC çağrılarına ilişkin verileri görüntülemek için bu tipi seçin.
	MQOPEN ve MQCLOSE	MQOL ve MQCLOSE çağrılarına ilişkin verileri görüntülemek için bu tipi seçin.
	MQINQ ve MQSET	MQINQ ve MQSET çağrılarına ilişkin verileri görüntülemek için bu tipi seçin.

<i>Çizelge 6. Kaynak tipleri (devamı var)</i>		
Sınıf	Tip	Tanım
	MQPUT	MQPUT ile ilgili çağrılara ilişkin verileri görüntülemek için bu tipi seçin.
	MQGet	MQGET ile ilgili çağrılara ilişkin verileri görüntülemek için bu tipi seçin.
	Kesinleştirme ve geriye işleme	Kuyruk yöneticisi tarafından eşitleme noktalarının kullanımıyla ilgili bilgileri görüntülemek için bu tipi seçin.
	Abone Ol	MQSUB çağrılarıyla ilgili verileri görüntülemek için bu tipi seçin.
	Yayınla	Yayınlanan iletilere ilişkin verileri görüntülemek için bu tipi seçin.
Kuyruk başına API kullanım istatistikleri	MQOPEN ve MQCLOSE	Belirtilen kuyruk için MQOL ve MQCLOSE çağrılarının ilişkin verileri görüntülemek için bu tipi seçin.
	MQINQ ve MQSET	Belirtilen kuyruk için MQINQ ve MQSET çağrılarının ilişkin verileri görüntülemek için bu tipi seçin.
	MQPUT ve MQPUT1	Belirtilen kuyruk için MQPUT ile ilgili ve MQPUT1-related çağrılara ilişkin verileri görüntülemek için bu tipi seçin.
	MQGet	Belirlenen kuyruk için MQGET ile ilgili çağrılara ilişkin verileri görüntülemek için bu tipi seçin.

e) İzlenecek **Kaynak ögesi** ' yi seçin:

Seçilebilecek kaynak öğeleri, seçilen kaynak sınıfına ve kaynak tipine bağlı olarak değişir. Aşağıdaki tablolarda kaynak öğeleri gösterilmektedir:

<i>Çizelge 7. Platform merkezi işleme birimi kaynaklarına ilişkin öğeler</i>		
Tip	Öge	Tanım
CPU performansı- platform genişliği	Kullanıcı CPU süresi yüzdesi	Kullanıcı durumundaki CPU kullanım yüzdesini gösterir.
	Sistem CPU süresi yüzdesi	Sistem durumundaki CPU ' nun yüzdesini gösterir.
	CPU yükleme-bir dakikalık ortalama	Yükleme ortalamasını 1 dakikadan fazla gösterir.
	CPU yükleme-beş dakikalık ortalama	Yükleme ortalamasını 5 dakikadan fazla gösterir.
	CPU yükü-onbeş dakikalık ortalama	Yükleme ortalamasını on beş dakikadan fazla gösterir.
	RAM serbest yüzdesi	Boş RAM belleğinin yüzdesini gösterir.

<i>Çizelge 7. Platform merkezi işleme birimi kaynaklarına ilişkin öğeler (devamı var)</i>		
Tip	Öge	Tanım
	RAM toplam baytları	Yapılandırılan RAM ' in toplam baytlarını gösterir.
CPU performansı- çalışan kuyruk yöneticisi	Kullanıcı CPU süresi-kuyruk yöneticisi için yüzde tahmini	İzlenmekte olan kuyruk yöneticileriyle ilgili işlemler için kullanıcı durumundaki CPU kullanımı yüzdesini tahmin eder.
	Sistem CPU süresi-kuyruk yöneticisi için yüzde tahmini	İzlenmekte olan kuyruk yöneticileriyle ilgili işlemler için sistem durumundaki CPU kullanımı yüzdesini tahmin eder.
	RAM toplam bayt sayısı-kuyruk yöneticileri için tahmin	İzlenmekte olan kuyruk yöneticileri tarafından kullanılan RAM ' in toplam bayt miktarını tahmin eder.

<i>Çizelge 8. Platform kalıcı verilerine ilişkin öğeler kaynakları depolar</i>		
Tip	Öge	Tanım
Disk kullanımı-altyapı genişliği	MQ trace file system - bytes in use	İzleme kütüğü sistemi tarafından kullanılmakta olan disk saklama alanının bayt sayısını gösterir.
	MQ izleme kütüğü sistemi-boş alan	Serbest olan izleme kütüğü sistemi için ayrılmış disk saklama alanını gösterir.
	MQ errors file system - bytes in use	Hata verileri tarafından kullanılmakta olan disk depolama alanının bayt sayısını gösterir.
	MQ hata dosyası sistemi-boş alan	Serbest hata verileri için ayrılmış disk depolama alanını gösterir.
	MQ FDC dosya sayısı	Geçerli FDC dosyalarının sayısını gösterir.
Disk kullanımı-çalışan kuyruk yöneticileri	Kuyruk Yöneticisi dosya sistemi-kullanımdaki bayt sayısı	İzlemekte olduğunuz kuyruk yöneticilerine ilişkin kuyruk yöneticisi dosyaları tarafından kullanılan disk saklama alanı bayt sayısını gösterir.
	Kuyruk Yöneticisi dosya sistemi-boş alan	Serbest olan kuyruk yöneticisi dosyaları için ayrılmış disk depolama alanını gösterir.
Disk kullanımı-kuyruk yöneticisi kurtarma günlüğü	Günlük-bayt kullanımda	İzlemekte olduğunuz kuyruk yöneticilerinin kurtarma günlükleri için kullanılan disk depolama alanının bayt sayısını gösterir.
	Günlük-byte sayısı üst sınırı	Kuyruk yöneticisi kurtarma günlükleri için kullanılmak üzere yapılandırılmış disk depolama alanı üst sınırını gösterir.

<i>Çizelge 8. Platform kalıcı verilerine ilişkin ögeler kaynakları depolar (devamı var)</i>		
Tip	Öge	Tanım
	Günlük kütüğü sistem byte'larını kullanımda	Günlük dosyası sistemi için kullanılan toplam disk baytlarının sayısını gösterir.
	Günlük dosyası sistemi-byte sayısı üst sınırı	Günlük kütüğü sistemi için konfigürasyonu tanımlanmış disk baytlarının sayısını gösterir.
	Günlük-fiziksel baytlar yazıldı	Kurtarma günlüklerine yazılmakta olan baytların sayısını gösterir.
	Günlük-mantıksal baytlar yazıldı	Kurtarma günlüklerine yazılan mantıksal bayt sayısını gösterir.
	Günlük yazma gecikme süresi	Kuyruk yöneticisi kurtarma günlüğüne zamanuyumlu olarak yazılırken gecikmenin bir ölçüsünün gösterilmesini sağlar.

<i>Çizelge 9. API kullanım istatistikleri kaynaklarına ilişkin ögeler</i>		
Tip	Öge	Tanım
MQCONN ve MQDISC	MQCONN/MQCONNX sayısı	MQCONN ve MQCONNX çağrılarının sayısını gösterir.
	Başarısız olan MQCONN/MQCONNX sayısı	MQCONN ve MQCONNX ile ilgili başarısız çağrılarının sayısını gösterir.
	Eşzamanlı bağlantılar-yüksek su işareti	Yürürlükteki istatistik aralığında koştuzamanlı bağlantı sayısı üst sınırını gösterir.
	MQDISC sayısı	MQDISC çağrılarının sayısını gösterir.
MQOPEN ve MQCLOSE	MQOPER sayısı	MQOPER çağrılarının sayısını gösterir.
	Başarısız MQOPEN sayısı	MQOPER ' a yapılan başarısız çağrılarının sayısını gösterir.
	MQCLOSE sayısı	MQCLOSE çağrılarının sayısını gösterir.
	MQCLOSE sayısı başarısız oldu	MQCLOSE ' ye yönelik başarısız çağrılarının sayısını gösterir.
MQINQ ve MQSET	MQINQ sayısı	MQINQ çağrılarının sayısını gösterir.
	Başarısız olan MQINQ sayısı	MQINQ ' ya yönelik başarısız çağrılarının sayısını gösterir.

<i>Çizelge 9. API kullanım istatistikleri kaynaklarına ilişkin ögeler (devamı var)</i>		
Tip	Öge	Tanım
	MQSET sayısı	MQSET çağrılarının sayısını gösterir.
	Başarısız MQSET sayısı	MQSET ' e yapılan başarısız çağrılarının sayısını gösterir.
MQPUT	Aralık toplamı MQPUT/MQPUT1 sayısı	MQPUT ve MQPUT1 çağrılarının sayısını gösterir.
	Aralık toplam MQPUT/MQPUT1 byte sayısı	MQPUT ve MQPUT1 çağrıları tarafından konulan toplam veri baytlarını gösterir.
	Kalıcı olmayan ileti MQPUT sayısı	MQPUT tarafından kullanılan, kalıcı olmayan iletilerin sayısını gösterir.
	Kalıcı ileti MQPUT sayısı	MQPUT tarafından yerleştiren kalıcı iletilerin sayısını gösterir.
	Başarısız MQPUT sayısı	MQPUT ' e başarısız çağrılarının sayısını gösterir.
	Kalıcı olmayan ileti MQPUT1 sayısı	MQPUT1 tarafından gönderilen, kalıcı olmayan iletilerin sayısını gösterir.
	Kalıcı ileti MQPUT1 sayısı	MQPUT1 tarafından kullanılan kalıcı iletilerin sayısını gösterir.
	Başarısız MQPUT1 sayısı	Başarısız olan çağrılarının sayısını MQPUT1' e gösterir.
	Kalıcı olmayan ileti bayt sayısını girin	Kalıcı olmayan iletilerde bulunan bayt sayısını gösterir.
	Kalıcı ileti baytı sayısı-byte sayısı	Kalıcı iletilerde bulunan bayt sayısını gösterir.
	MQSTAT sayısı	MQSTAT çağrılarının sayısını gösterir.
	Başarısız olan MQSTAT sayısı	MQSTAT ' a yapılan başarısız çağrılarının sayısını gösterir.
MQGet	Aralık toplam yıkıcı get-count	MQGET işlemi tarafından kuyruklardan kaldırılan iletilerin sayısı.
	Aralık toplam yıkıcı get-byte sayısı	MQGET tarafından kuyruklardan kaldırılan veri byte 'ları.

<i>Çizelge 9. API kullanım istatistikleri kaynaklarına ilişkin öğeler (devamı var)</i>		
Tip	Öğ	Tanım
	Kalıcı olmayan ileti yıkıcı get-count	MQGET işlemi tarafından kuyruklardan kaldırılan, kalıcı olmayan iletilerin sayısı.
	Kalıcı ileti yıkıcı get-count	MQGET işlemi tarafından kuyruklardan kaldırılan kalıcı iletilerin sayısı.
	MQGET-count başarısız oldu	MQGET ' ye yönelik başarısız çağrılarının sayısını gösterir.
	Kalıcı olmayan iletiler alındı-bayt sayısı	MQGET ' e döndürülen, kalıcı olmayan iletilerin bayt sayısını gösterir.
	Kalıcı iletiler-byte sayısı alındı	MQGET ' e döndürülen kalıcı iletilerin bayt sayısını gösterir.
	Kalıcı olmayan ileti göz atma sayısı	Göz atılmamış, kalıcı olmayan iletilerin sayısını gösterir.
	Kalıcı ileti göz atma sayısı	Göz atılan kalıcı iletilerin sayısını gösterir.
	Başarısız göz atma sayısı	Başarısız olan ileti taramalarının sayısını gösterir.
	Kalıcı olmayan ileti göz atma-bayt sayısı	Göz atmayan, kalıcı olmayan iletilerin bayt sayısını gösterir.
	Kalıcı ileti göz atma-bayt sayısı	Göz atılan kalıcı iletilerin bayt sayısını gösterir.
	Süresi dolan ileti sayısı	Süresi dolan iletilerin sayısını gösterir.
	Temizlenen kuyruk sayısı	Temizlenen kuyrukların sayısını gösterir.
	MQCB sayısı	MQCB çağrılarının sayısını gösterir.
	Başarısız olan MQCB sayısı	MQCB ' ye yapılan başarısız çağrılarının sayısını gösterir.
	MQCTL sayısı	MQCTL çağrılarının sayısını gösterir.
	MQCTL sayısı başarısız oldu	MQCTL ' ye yapılan başarısız çağrılarının sayısını gösterir.
Kesinleştirme ve geriye işleme	Kesinleştirme sayısı	MQCMIT çağrılarının sayısını gösterir.
	Kesinleştirme sayısı başarısız oldu	MQCMIT ' e yapılan başarısız çağrılarının sayısını gösterir.

Çizelge 9. API kullanım istatistikleri kaynaklarına ilişkin ögeler (devamı var)

Tip	Öge	Tanım
	Geriye işleme sayısı	MQBACK çağrılarının sayısını gösterir.
Abone Ol	Sürekli abonelik sayısı yarat	Kalıcı abonelikler yaratmak için MQSUB çağrılarının sayısını gösterir.
	Kalıcı abonelik sayısını değiştir	Kalıcı abonelikleri değiştirmek için MQSUB çağrılarının sayısını gösterir.
	Sürekli abonelik sayısını sürdür	Kalıcı abonelikleri sürdürmek için MQSUB çağrılarının sayısını gösterir.
	Kalıcı olmayan abonelik sayısı yarat	Kalıcı olmayan abonelikler yaratmak için MQSUB çağrılarının sayısını gösterir.
	Kalıcı olmayan abonelik sayısını değiştir	Kalıcı olmayan abonelikleri değiştirmek için MQSUB çağrılarının sayısını gösterir.
	Kalıcı olmayan abonelik sayısını sürdür	Kalıcı olmayan abonelikleri sürdürmek için MQSUB çağrılarının sayısını gösterir.
	Başarısız yaratma/değiştirme/sürdürme abonelik sayısı	Abonelikleri yaratmak, değiştirmek ya da sürdürmek için MQSUBRQ ' ya yönelik başarısız çağrılarının sayısını gösterir.
	Kalıcı abonelik sayısını sil	Kalıcı abonelikleri silmek için MQSUB çağrılarının sayısını gösterir.
	Kalıcı olmayan abonelik sayısını sil	Kalıcı olmayan abonelikleri silmek için MQSUB çağrılarının sayısını gösterir.
	Abonelik silme başarısız sayısı	Abonelikleri silmek için MQSUB çağrılarının sayısını gösterir.
	MQSUBRQ sayısı	MQSUBRQ çağrılarının sayısını gösterir
	Başarısız olan MQSUBRQ sayısı	MQSUBRQ ' ya yönelik başarısız çağrılarının sayısını gösterir
	Dayanıklı abone-yüksek su işareti	Geçerli istatistik aralığında en fazla dayanıklı abonelik sayısını gösterir.

<i>Çizelge 9. API kullanım istatistikleri kaynaklarına ilişkin öğeler (devamı var)</i>		
Tip	Öğ	Tanım
	Dayanıklı abone-düşük su işareti	Geçerli istatistik aralığında en az dayanıklı abonelik sayısını gösterir.
	Dayanıklı olmayan abone-yüksek su işareti	Yürürlükteki istatistik aralığında, kalıcı olmayan abonelik sayısı üst sınırını gösterir.
	Dayanıklı olmayan abone-düşük su işareti	Yürürlükteki istatistik aralığında, kalıcı olmayan abonelik sayısı alt sınırını gösterir.
Yayınla	Konu MQPUT/MQPUT1 aralık toplamı	Konulara koyulan iletilerin sayısı.
	Aralık toplam konu byte sayısı	Konulara konulan ileti baytlarının sayısı.
	Abonelere yayınlandı-ileti sayısı	Abonelere yayınlanan ileti sayısını gösterir.
	Abonelere yayınlandı-bayt sayısı	Abonelere yayınlanan iletilerin bayt sayısını gösterir.
	Kalıcı olmayan konu MQPUT/MQPUT1 sayısı	Konulara koyulan kalıcı olmayan iletilerin sayısını gösterir.
	Kalıcı-konu MQPUT/MQPUT1 sayısı	Konulara koyulan kalıcı iletilerin sayısını gösterir.
	Başarısız olan MQPUT/MQPUT1 sayısı	Bir konuya konmak için başarısız deneme sayısını gösterir.

<i>Çizelge 10. API için, kuyruk başına kullanım istatistikleri kaynakları için öğeler</i>		
Tip	Öğ	Tanım
MQOPEN ve MQCLOSE	MQOPER sayısı	MQOPER çağrılarının sayısını gösterir.
	MQCLOSE sayısı	MQCLOSE çağrılarının sayısını gösterir.
MQINQ ve MQSET	MQINQ sayısı	MQINQ çağrılarının sayısını gösterir.
	MQSET sayısı	MQSET çağrılarının sayısını gösterir.
MQPUT ve MQPUT1	MQPUT/MQPUT1 sayısı	MQPUT ve MQPUT1 çağrılarının sayısını gösterir.

<i>Çizelge 10. API için, kuyruk başına kullanım istatistikleri kaynakları için öğeler (devamı var)</i>		
Tip	Öge	Tanım
	MQPUT byte sayısı	MQPUT ve MQPUT1 çağrıları tarafından konulan toplam veri baytlarını gösterir.
	MQPUT kalıcı olmayan ileti sayısı	MQPUT tarafından kullanılan, kalıcı olmayan iletilerin sayısını gösterir.
	MQPUT kalıcı ileti sayısı	MQPUT tarafından yerleştiren kalıcı iletilerin sayısını gösterir.
	MQPUT1 kalıcı olmayan ileti sayısı	MQPUT1 tarafından gönderilen, kalıcı olmayan iletilerin sayısını gösterir.
	MQPUT1 kalıcı ileti sayısı	MQPUT1 tarafından kullanılan kalıcı iletilerin sayısını gösterir.
	Kalıcı olmayan bayt sayısı	Kalıcı olmayan iletilerde bulunan bayt sayısını gösterir.
	Kalıcı byte sayısı	Kalıcı iletilerde bulunan bayt sayısını gösterir.
	Kuyruktan kaçınılan kuyruk	
	Kuyruktan kaçınan bayt	
	kilit çekişmesi	
MQGet	MQGET sayısı	
	MQGET byte sayısı	
	Yıkıcı MQGET kalıcı olmayan ileti sayısı	MQGET tarafından kuyruktan kaldırılan, kalıcı olmayan iletilerin sayısı.
	Yıkıcı MQGET kalıcı ileti sayısı	MQGET tarafından kuyruktan kaldırılan kalıcı iletilerin sayısı.
	Yıkıcı MQGET kalıcı olmayan byte sayısı	MQGET ' e döndürülen, kalıcı olmayan iletilerin bayt sayısını gösterir.
	Yıkıcı MQGET kalıcı byte sayısı	MQGET ' e döndürülen kalıcı iletilerin bayt sayısını gösterir.
	MQGET kalıcı olmayan ileti sayısına göz at	Göz atılmamış, kalıcı olmayan iletilerin sayısını gösterir.
	MQGET göz atma kalıcı ileti sayısı	Göz atılan kalıcı iletilerin sayısını gösterir.
	MQGET kalıcı olmayan byte sayısına göz at	Göz atmayan, kalıcı olmayan iletilerin bayt sayısını gösterir.
	MQGET göz atma kalıcı byte sayısı	Göz atılan kalıcı iletilerin bayt sayısını gösterir.

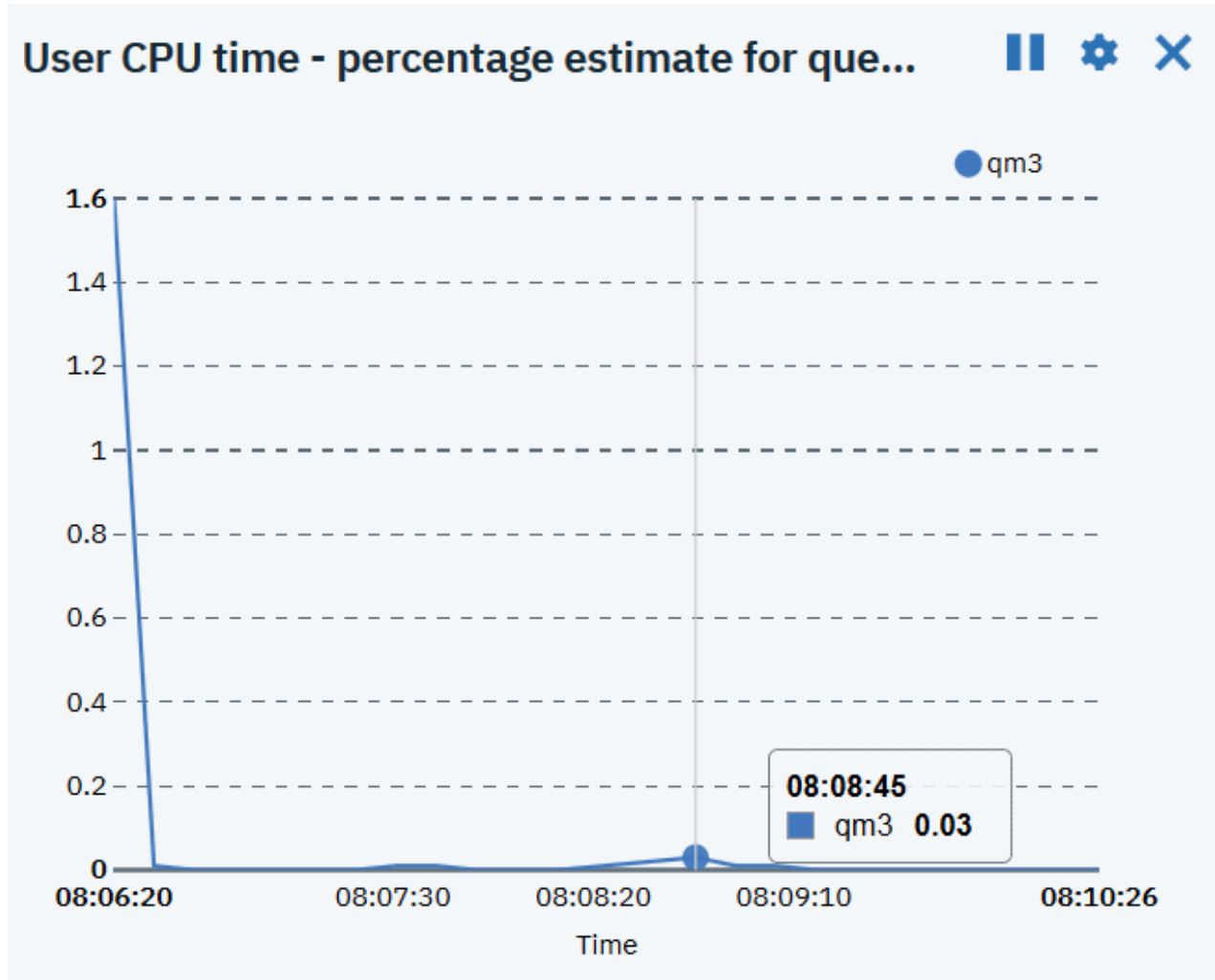
Çizelge 10. API için, kuyruk başına kullanım istatistikleri kaynakları için öğeler (devamı var)		
Tip	Öge	Tanım
	Süresi dolmuş iletiler	Süresi dolan iletilerin sayısını gösterir.
	Silinen kuyruk sayısı	Temizlenen kuyrukların sayısını gösterir.
	Ortalama Kuyruk Süresi	
	Kuyruk süresi	

f) İzlenecek kuyruk yöneticisini seçin ve o kuyruk yöneticisinin bilgilerini görüntülemek için kullanılacak rengi belirtin. Daha fazla kuyruk yöneticisi eklemek için **Ekle** düğmesini tıklayın. En çok beş kuyruk yöneticisi belirtebilirsiniz.

g) **Kaydet**'i tıklayın.

Sonuçlar


Pencere aracını yapılandırdıktan sonra, grafikte veriler görüntülenmeden önce kısa bir bekleme süresi vardır. Veriler bir zaman ekseni boyunca görüntülenir. Her veri noktası, verilerin toplandığı 10 saniyelik sürenin sonunu gösterir. Aşağıdaki örnekte gösterildiği gibi ayrıntılı bilgileri görmek için imleci grafikte veri noktalarının üzerine getirebilirsiniz: **V 9.0.5**



Gösterge panosu düzenlerinin yapılandırılması

Gösterge panosu, gereçlerin gösterildiği IBM MQ Console içindeki bir taşıyıcıdır. Farklı bilgi seçimlerini göstermek için birden çok gösterge panosu sekmesi oluşturabilirsiniz.

Bu görev hakkında

Her bir pano sekmesini  sekme adının yanındaki oku tıklatarak yapılandırabilirsiniz. Sekme adını değiştirebilir ve sekme için bir açıklama ekleyebilirsiniz. Sekmenin kaç sütuna sahip olduğunu da yapılandırabilirsiniz.

Pencere araçlarını sürükleyip bırakarak, pencere bileşenlerinin yerleşim düzenini bir gösterge panosu sekmesinin içinde yapılandırabilirsiniz.

Yordam

- “Gösterge panosu sekmelerinin yaratılması ve silinmesi” sayfa 123
- “Gösterge panosu düzenlerinin içe ve dışa aktarılması” sayfa 124

Gösterge panosu sekmelerinin yaratılması ve silinmesi

Belirli bir yerel kuyruk yöneticisine ilişkin bilgileri gösteren bir gösterge panosu sekmesi otomatik olarak yaratılabilir. Gösterge panosu sekmelerini el ile oluşturabilir ve silebilirsiniz.

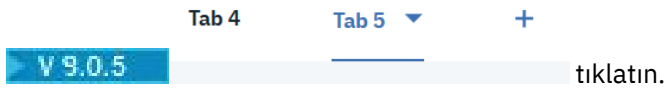
Bu görev hakkında

Belirli bir yerel kuyruk yöneticisine ilişkin bilgileri göstermek için otomatik olarak bir gösterge panosu sekmesi oluşturduğunuzda, aşağıdaki pencere araçları otomatik olarak eklenir:

- Kuyruklar gereci
- İstemci bağlantı kanalları gereci
- Kanallar pencere bileşeni
- Dinleyiciler gereci
- Abonelikler gereci
- Konular gereci
- Kimlik doğrulama bilgileri gereci

Yordam

- Gösterge panosu sekmesi oluşturmak için:
 - a) Var olan gösterge panosu sekmelerinizin yanındaki artı simgesini



- b) Yeni sekme için bir ad girin.
 - c) İsteğe bağlı: Yeni sekme için bir açıklama girin.
 - d) **Ekle**'yi tıklatın.
- Belirli bir kuyruk yöneticisi için otomatik olarak bir gösterge panosu sekmesi oluşturmak için:
 - a) Yerel kuyruk yöneticisi gerecindeki kuyruk yöneticisini seçin.
 - b) Seç ... > **Yeni gösterge panosu sekmesi ekle**
Yeni bir gösterge panosu sekmesi oluşturulur. Sekme, kuyruk yöneticisinin adını içerir.

- Bir gösterge panosu sekmesini silmek için:

- a) Gösterge panosu sekme adının yanındaki oku tıkklatın .
- b) **Sil sekmesi** seçeneğini belirleyin.
- c) Confirm that you want to delete the dashboard tab by clicking **Sil**. Sekme silinir.

V 9.0.1 Gösterge panosu düzenlerinin içe ve dışa aktarılması


Bir gösterge panosu düzenini IBM MQ Console' den dışa aktararak kaydedebilirsiniz. Kaydedilmiş bir pano düzenini IBM MQ Console'ine aktarabilirsiniz.

Bu görev hakkında


Bir gösterge panosunu dışa aktardığınızda, yerel diskinizde bir .json dosyası yaratassınız. Daha sonra, yerleşimi yeniden oluşturmak için bir gösterge panosuna .json dosyasını içe aktarabilirsiniz. Bir gösterge panosu düzenini içe aktardığınızda, içe aktarılan sekmeleri var olan bir pano düzenine eklemeyi seçebilirsiniz. Diğer bir seçenek olarak, var olan gösterge panosu düzenini içe aktarılan yerleşim düzeniyle değiştirebilirsiniz.

Yordam

- Gösterge panosu düzenini dışa aktarmak için:

- a) Gösterge panosu menüsü simgesini () tıkklatın.
- b) **Gösterge panosunu dışa aktar** seçeneğini belirleyin.
Dosya, tarayıcı karşıdan yükleme klasörünüze kaydedilir.

- Bir gösterge panosu düzenini içe aktarmak için:

- a) Gösterge panosu menüsü simgesini () tıkklatın.
- b) **Gösterge panosunu içe aktar** seçeneğini belirleyin.
Gösterge Panosunu İçe Aktar penceresi açılır.
- c) **Göz At** düğmesini tıkklatın ve yapılandırmanızı içeren dosyanın yerini bulmak için göz atın.
- d) Gösterge panosu sekmelerinin nasıl içe aktarılacağını seçin:
Aşağıdaki seçenekler arasından seçim yapabilirsiniz:
 - **İçe aktarılan gösterge panosu sekmelerinin var olan gösterge panosuna eklenmesi**
 - **İçe aktarılan gösterge panosu sekmeleriyle var olan gösterge panosunu değiştir**
- e) **İçe Aktar**'ı tıkklatın.
Gösterge panosu sekmeleri içe aktarılır.




V 9.0.1 Gösterge panosu denetimleri

IBM MQ Console izlemesini etkinleştirmek, çevrimiçi yardıma erişmek, IBM MQ Console ile ilgili bilgileri görüntülemek ve IBM MQ Console oturumunu kapatmak için gösterge panosunun üst kısmındaki denetimleri kullanın.

Bu görev hakkında

Gösterge panosu denetimleri, IBM MQ Console' in genel özellikleri için geçerlidir.

Yordam

- Yardım simgesini () tıklatarak IBM MQ için IBM Documentation olanağına erişin.
- Gösterge panosunu içe aktarmak, dışa aktarmak ya da ilk durumuna getirmek için menü simgesini tıklatın. Daha fazla bilgi için, bkz. “Gösterge panosu düzenlerinin yapılandırılması” sayfa 123.
- IBM MQ Console ile ilgili tanılama izlemesini etkinleştirmek ya da devre dışı bırakmak için ayarlar simgesini  tıklatın.
- Kişi simgesini  tıklatarak ve **Hakkında** seçeneğini belirleyerek, oturum açan kullanıcıyla ilgili bilgiler de dahil olmak üzere IBM MQ Console ile ilgili bilgileri görüntüleyin.
- **Logout**(Oturumu Kapat) düğmesini tıklatarak IBM MQ Console oturumunu kapatın.
Bir istemci sertifikası tarafından IBM MQ Console ' ta oturum açtığınız ya da güvenlik yapılandırma örneği no_security.xml kullanılıyorsa bu seçenek kullanılamaz. Güvenlik yapılandırmalarıyla ilgili daha fazla bilgi için [IBM MQ Console güvenliğinin yapılandırılması](#) başlıklı konuya bakın.

Klavye kısayolları

IBM MQ Console ile çalışmak için klavye kısayollarını kullanabilirsiniz.

Aşağıdaki tabloda kullanılabilir kısayollar listelenir. Tüm kısayollar bir pencere bileşeni içinden kullanılır.

Anahtar	İşlem
p	Seçilen nesneye ilişkin özellikleri görüntüler
c	Yeni Nesne Yarat
shift-d	Seçilen nesneyi siler
a	Bir gereçteki tüm nesnelere seçer
üst karakter-a	Nesnelerin seçimini kaldır

Linux

Windows

IBM MQ Explorer komutunu kullanarak yönetim

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

IBM MQ for Windows ve IBM MQ for Linux x86-64 , denetim ya da MQSC komutlarını kullanmaya alternatif olarak yönetim görevlerini yerine getirmek için IBM MQ Explorer adlı bir denetim arabirimi sağlar. [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 çalışan kuyruk yöneticilerine uzaktan bağlanabilir, tüm ileti sistemi omurgasının konsoldan görüntülenmesine, keşfedilmesine ve değiştirilebilmesine 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 127](#).

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.

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 makineniz için).
- Kuyruk yöneticisini başlatma ve durdurma (yalnızca yerel makineniz için).
- Kuyruklar ve kanallar gibi IBM MQ nesnelерinin tanımlarını tanımlayın, görüntüleyin ve değiştirin.
- Kuyruklardaki iletilere göz atın.
- Bir kanalı başlat ve durdur.
- Bir kanala, dinleyiciye, kuyruğa ya da hizmet nesnelere ilişkin durum bilgilerini görüntüleyin.
- 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 Kuyruk Yöneticisi) 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.
- Kanal başlatıcıları, tetikleyici izleyicileri ve dinleyicileri yaratmanızı ve silmesini sağlar.
- Komut sunucularını, kanal başlatıcılarını, tetikleyici izleyicileri ve dinleyicileri başlatın ya da durdurun.
- Bir kuyruk yöneticisi başlatıldığında otomatik olarak başlamak için belirli hizmetleri ayarlayın.
- Kuyruk yöneticilerinin özelliklerini değiştirin.
- Yerel varsayılan kuyruk yöneticisini değiştirin.
- TLS sertifikalarını yönetmek, sertifikaları kuyruk yöneticileriyle ilişkilendirmek ve sertifika depolarını yapılandırmak ve ayarlamak için **strmqikm** (ikeyMan) GUI 'sini çağırın (yalnızca yerel makinenizdeki).
- IBM MQ nesnelерinden JMS nesneleri ve JMS nesnelерinden IBM MQ nesneleri oluşturun.
- Şu anda desteklenen tiplerden herhangi biri için bir JMS Connection Factory (Bağlantı Üreticisi) yaratın.
- Bir dinleyiciye ilişkin TCP kapı numarası ya da kanal başlatıcı kuyruğu adı gibi, herhangi bir hizmete ilişkin parametreleri değiştirin.
- Hizmet izlemesini başlatır ya da durdurur.

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 kümeye ilişkin 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 uygulamalarıyla 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](#).

Linux Windows 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 127](#)
- [“IBM MQ Explorer güvenliği” sayfa 128](#)
- [“Showing and hiding queue managers and clusters in IBM MQ Explorer” sayfa 131](#)
- [“Küme üyeliği ve IBM MQ Explorer” sayfa 132](#)
- [“IBM MQ Explorer için veri dönüştürme” sayfa 133](#)

IBM MQ Explorer için önkoşul olan yazılımlar ve tanımlar

IBM MQ Explorer' yi 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 UNIX 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 Explorer güvenliği

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. [UNIX ya da Linux sistemlerinde IBM MQ nesnesine erişim verilmesi ve Windows](#) .

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 görünülmesine neden olur. 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 almazsanız, 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 belirtiyorsunuz. 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 129](#) and [“Tasks on the system that hosts the IBM MQ Explorer” sayfa 130](#).

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](#) özniteliği için uygun değeri belirtin. [SSLCIPH](#) özniteliğine ilişkin ek bilgi için [TLS ile kanalları korumabaşlıklı konuya](#) bakın.
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:
 - UNIX and Linux sistemleri için /var/mqm/qmgrs/+QMNAME+/SSL

- Windows sistemleri için C:\Program Files\IBM\MQ\qmgrs\+QMNAME+\SSL

Burada +QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.

5. 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 seçeneğini **runmqckm** komutlarıyla 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 [UNIX, Linux, and Windows üzerinde TLS ile çalışma](#).

Tasks on the system that hosts the IBM MQ Explorer

IBM MQ Explorer' i bulduran sistemde aşağıdaki görevleri gerçekleştirin:

1. key . jks adlı 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** ' ı 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 130, 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 bulduran sistem üzerindeki “Sistemdeki uzak kuyruk yöneticisini barındıran görevler” sayfa 129 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.

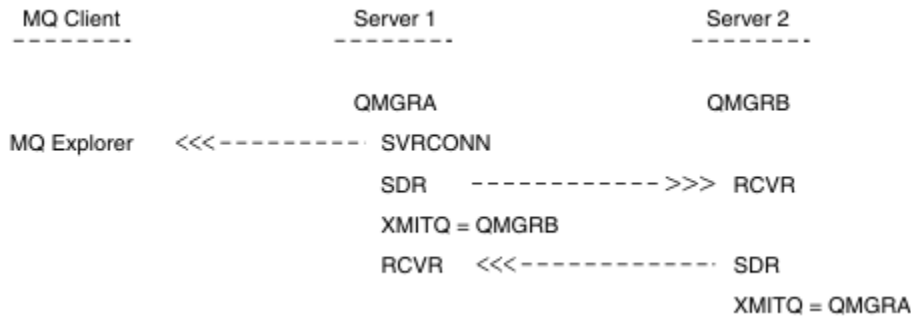
Bağlantı daha sonra bir ara kuyruk yöneticisi aracılığıyla hedef kuyruk yöneticisine bağlanmak için kullanıldıysa, userId ileti tanımlayıcısının (MQMD) *UserIdentifier* parametresine yeniden aktarılır. Hedef kuyruk yöneticisinde MCA dinleyicisinin bu iletiyi kabul etmesi için, MCAUSER özneliğinin ayarlanması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

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

Ařađıdaki örnek, bir ara kuyruk yöneticisi aracılıđıyla bir kuyruk yöneticisini IBM MQ Explorer' e nasıl bađlayacađını göstermektedir.

Kuyruk yöneticiyle uzak bir denetim bađlantısı kurun. Ařađıdakilerden emin olun:

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



Bu örnekte:

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

In this situation, there is no direct connection to QMGR B from IBM MQ Explorer; the connection to QMGR B is through QMGR A.

Server2 üzerindeki kuyruk yöneticisi QMGR B , gönderen alıcı kanallarını kullanarak Server1 üzerinde QMGR A ' a bađlı. The channel between QMGR A and QMGR B must be set up in such a way that remote administration is possible; see ["Uzaktan yönetim için kanallar ve iletim kuyrukları hazırlama" sayfa 187.](#)

Showing and hiding queue managers and clusters in IBM MQ Explorer

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

Uzak kuyruk yöneticisini göstermek için:

1. **Kuyruk Yöneticileri** ağaç düđümünü sađ tıkladın ve **Kuyruk Yöneticilerini Göster/Gizle** seçeneđini belirleyin.
2. **Ekle**'yi tıkladın. Kuyruk Yöneticileri Göster/Gizle panosu görüntülenir.
3. Sađlanan alanlardaki uzak kuyruk yöneticisinin adını ve anasistem adını ya da IP adresini girin.
Anasistem adı ya da IP adresi, uzak kuyruk yöneticisiyle bađlantı kurmak için varsayılan sunucu bađlantı kanalı olan SYSTEM.ADMIN.SVRCONNya da kullanıcı tanımlı bir sunucu bađlantısı kanalı.
4. **Bitir**'i tıkladın.

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 tıklatın.
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 tıklatın.

Uzak kuyruk yöneticilerinin listesini almak için:

1. IBM MQ Explorer' un Gezinme bölmesinde en yüksek **IBM MQ** ağaç düğümünü sağ tıklatın ve **IBM MQ Explorer Ayarlarını İçte Aktar**seçeneğini belirleyin.
2. **IBM MQ Explorer > IBM MQ Explorer Settings**öğesini tıklatın.
3. **Göz At**düğmesini tıklatın ve uzak kuyruk yöneticisi bağlantı bilgilerini içeren dosyanın yoluna gidin.
4. **Aç**'ı tıklatın. 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 (içte) aktarmak için **Son** düğmesini tıklatın.

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 yöneticisine, 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 görüntülenmesini 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 yönet

Yerel IBM MQ nesnelerini yönetme

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

Bu görev hakkında

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

Bu bölümde açıklanan yaklaşımlara ek olarak, yerel IBM MQ nesnelerini yönetmek için IBM MQ Explorer ' i kullanabilirsiniz. Daha fazla bilgi için, bkz. [“IBM MQ Explorerkomutunu kullanarak yönetim” sayfa 125.](#)

Yordam

- Yerel IBM MQ nesnelerini yönetmenize yardımcı olması için aşağıdaki konularda bilgileri kullanın.
 - [MQI kullanan uygulama programları](#)
 - [“MQSC komutlarını kullanarak MQ uygulamasını yönetme” sayfa 10](#)
 - [“Kuyruk yöneticileriyle çalışılması” sayfa 138](#)
 - [“Yerel kuyruklarla çalışma” sayfa 140](#)
 - [“Diğer ad kuyruklarıyla çalışma” sayfa 145](#)
 - [“Model kuyruklarıyla çalışma” sayfa 165](#)
 - [“Hizmetlerle çalışma” sayfa 172](#)
 - [“Tetikleme için nesnelere yönetme” sayfa 179](#)


Kuyruk yöneticisinin başlatılması ve durdurulması


Daha fazla bilgi için bağlantıları içeren bir kuyruk yöneticisini başlatma ve başlatma bilgileri.

Bu görev hakkında

Bir kuyruk yöneticisini aşağıdaki komutları kullanarak başlatabilir ve durdurabilirsiniz:

- Bir kuyruk yöneticisini başlatmak için **strmqm** komutunu kullanın.
- To stop a queue manager, use the **endmqm** command, which provides three ways to stop a queue manager: a controlled, or quiesced, shutdown, an immediate shutdown, and a preemptive shutdown.

 Diğer bir seçenek olarak, Windows ve Linuxüzerinde, kuyruk yöneticisini başlatmak ve durdurmak için IBM MQ Explorer ' i kullanabilirsiniz.

 On Windows you can configure a queue manager to start automatically when the system starts using the IBM MQ Explorer.

Yordam

1. Kuyruk yöneticisinin başlatılmasına ilişkin ek bilgi edinmek için [Kuyruk yöneticisinin başlatılması](#) başlıklı konuya bakın.
2. Kuyruk yöneticisinin nasıl durdurulacağı hakkında daha fazla bilgi için [Kuyruk yöneticisinin durdurulması](#) başlıklı konuya bakın.

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 yöneticisini durdurma](#) konusunda 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. [“Windowsüzerinde kuyruk yöneticisini el ile durdurma” sayfa 135.](#)

Linux UNIX

To stop a queue manager on UNIX or Linux, see [“Stopping a queue manager manually on UNIX and Linux” sayfa 136.](#)

İlgili bilgiler

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

Windows

Windowsüzerinde 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 <https://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):

İşlem adı	Tanım
AMQZMUC0	Kritik süreç yöneticisi
AMQZXMA0	Yürütme denetleyicisi
AMQZFUMA	OAM işlemi
AMQZLAA0	LQM araçları
AMQZLSA0	LQM araçları
AMQZMUFO	Yardımcı Program Yöneticisi
AMQZMGRO	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

Çizelge 12. Çalıştırılıyorsa, durdurulacak Windows işlemleri (devamı var)	
İşlem adı	Tanım
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

- Windows Control Panel (Denetim Masası) içindeki **Administration tools > Services** (Yönetim Araçları-Hizmetler) seçeneklerini kullanarak IBM MQ hizmetini durdurun.
- Tüm yöntemleri denediyseniz ve kuyruk yöneticisi durdurulmamışsa, sisteminizi yeniden başlatın.

Linux > UNIX **Stopping a queue manager manually on UNIX and Linux**

If you are unable to stop a queue manager on UNIX and Linux by using the **endmqm** command, you can try to stop the queue manager manually by ending any processes that are running and stopping the IBM MQ service.

Bu görev hakkında

UNIX ve Linux üzerinde kuyruk yöneticisini durdurmak için aşağıdaki adımları tamamlayın.

Kuyruk yöneticisini el ile durdurursanız, FFST alınabilir ve FDC dosyaları `/var/mqm/errors`ine yerleştirilebilir. Bu, kuyruk yöneticisinde bir hata olarak kabul edilmemelidir.

Kuyruk yöneticisi, el ile durdurmanın bu yöntemini kullanmanıza rağmen, olağan durumda yeniden başlatılacaktır.

Yordam

- Find the process IDs of the queue manager programs that are still running by using the **ps** command. Örneğin, kuyruk yöneticisi QMNAME olarak adlandırıldıysa, aşağıdaki komutu kullanın:

```
ps -ef | grep QMNAME
```

- 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:

Çizelge 13. UNIX and Linux processes to be stopped if running	
İşlem adı	Tanım
amqzmuc0	Kritik süreç yöneticisi
amqzma0	Yürütme denetleyicisi

Çizelge 13. UNIX and Linux processes to be stopped if running (devamı var)	
İşlem adı	Tanım
amqzfuma	OAM işlemi
amqzlaa0	LQM araçları
amqzlsa0	LQM araçları
amqzmuf0	Yardımcı Program Yöneticisi
amqzmur0	Yeniden başlatılabilir süreç yöneticisi
amqzmgr0	Süreç denetleyicisi
amqfqpub	Abone olma işlemi yayınlama
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.

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, ağın 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 bilgiler

[Kanallar](#)

Kuyruk yöneticileriyle çalışılması

Kuyruk yöneticisi özniteliklerini görüntülemek ya da değiştirmek için kullanabileceğiniz MQSC komutlarına örnekler.

Kuyruk yöneticisi özniteliklerinin görüntülenmesi

runmqsc komutunda belirtilen kuyruk yöneticisinin özniteliklerini görüntülemek için, **DISPLAY QMGR** MQSC komutunu kullanın:

```
DISPLAY QMGR
```

Bu komuttaki tipik çıkış [Şekil 16 sayfa 139](#) içinde gösterilir.

```

DISPLAY QMGR
  1 : DISPLAY QMGR
AMQ8408: Display Queue Manager details.
QMNAME(QM1)
ACCTINT(1800)
ACCTQ(OFF)
ACTVCONO (DISABLED)
ALTDATE(2012-05-27)
AUTHOREV(DISABLED)
CHAD(DISABLED)
CHADEXIT( )
CLWLDATA( )
CLWLEN(100)
CLWLUSEQ(LOCAL)
CMDLEVEL(800)
CONFIGEV(DISABLED)
CRTIME(16.14.01)
DEFXMITQ( )
DISTL(YES)
IPADDRV(IPV4)
LOGGERSV(DISABLED)
MAXHANDS(256)
MAXPROPL(NOLIMIT)
MAXUMSGS(10000)
MONCHL(OFF)
PARENT( )
PLATFORM(WINDOWSNT)
PSNPMMSG(DISCARD)
PSSYNCP(1FPER)
PSMODE(ENABLED)
REPOS( )
ROUTEREC(MSG)
SCMDSERV(QMGR)
SSLCRYP( )
SSLFIPS(NO)
MQ\Data\qmgrs\QM1\ssl\key)
SSLKEYC(0)
STATCHL(OFF)
STATMQI(OFF)
STRSTPEV(ENABLED)
TREELIFE(1800)
ACCTCONO(DISABLED)
ACCTMQI(OFF)
ACTIVREC(MSG)
ACTVTRC(OFF)
ALTTIME(16.14.01)
CCSID(850)
CHADEV(DISABLED)
CHLEV(DISABLED)
CLWLXIT( )
CLWLMRUC(999999999)
CMDEV(DISABLED)
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE)
CRDATE(2011-05-27)
DEADQ( )
DESCR( )
INHIBTEV(DISABLED)
LOCALEV(DISABLED)
MARKINT(5000)
MAXMSGL(4194304)
MAXPRTY(9)
MONACLS(QMGR)
MONQ(OFF)
PERFMEV(DISABLED)
  PSRTYCNT(5)
PSNPRES(NORMAL)
QMID(QM1_2011-05-27_16.14.01)
REMOETEVEV(DISABLED)
REPOSNL( )
SCHINIT(QMGR)
SSLCRLNL( )
SSLEV(DISABLED)
SSLKEYR(C:\Program Files\IBM\WebSphere
STATACLS(QMGR)
STATINT(1800)
STATQ(OFF)
SYNCP
TRIGINT(999999999)

```

Şekil 16. Bir DISPLAY QMGR komutundan tipik çıkış

Not: SYNCP, salt okunur bir kuyruk yöneticisi öznitelidir.

ALL parametresi, **DISPLAY QMGR** komutundaki varsayılan parametredir. Tüm kuyruk yöneticisi özniteliklerini görüntüler. Çıkış, özellikle, size varsayılan kuyruk yöneticisi adını, ölü harf kuyruğu adını ve komut kuyruğu adını gösterir.

Bu kuyrukların şu komutu girerek var olduğunu doğrulayabilirsiniz:

```
DISPLAY QUEUE (SYSTEM.*)
```

Bu, kök SYSTEM.* ile eşleşen kuyrukların listesini görüntüler. Parantezler gereklidir.

Kuyruk yöneticisi özniteliklerinin değiştirilmesi

runmqsc komutunda belirtilen kuyruk yöneticisinin özniteliklerini değiştirmek için, değiştirmek istediğiniz öznitelikleri ve değerleri belirterek **MQSC** komutunu **ALTER QMGR** 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.

İlgili bilgiler

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

[Kuyruk yöneticisine ilişkin öznitelikler](#)

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

[QMGR GÖRÜNTÜLE](#)

[ALTER QMGR](#)

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 bilgiler

[Kuyruklara ilişkin adlandırma kısıtları](#)

[Diğer nesnelere ilişkin adlandırma kısıtlamaları](#)

Yerel kuyruk tanımlanması

Bir uygulama için, yerel kuyruk yöneticisi, uygulamanın bağlı olduğu kuyruk yöneticidir. Yerel kuyruk yöneticisi tarafından yönetilen kuyruklara, o kuyruk yöneticisinin yerel olduğu söyleniyor.

Yerel kuyruk yaratmak için **DEFINE QLOCAL** MQSC komutunu kullanın. Varsayılan yerel kuyruk tanımlanmasında tanımlı olan varsayılan değeri kullanabilir ya da kuyruk özelliklerini varsayılan yerel kuyruklardaki gibi değiştirebilirsiniz.

Not: Varsayılan yerel kuyruk SYSTEM.DEFAULT.LOCAL.QUEUE ve sistem kurulumu sırasında yaratıldı.

Örneğin, izleyen **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)
```

Not:

1. Açıklamadaki değer dışında, örnekte gösterilen tüm öznitelik değerleri varsayılan değerlerdir. Bu örnekler, kesinlikle örnek purposesyes için yer alır. Varsayılan değerlerin istediğiniz ya da değiştirilmemiş olduğundan eminseniz, bu değerleri atlayabilirsiniz. Ayrıca bkz. [“Varsayılan nesne özniteliklerinin görüntülenmesi” sayfa 141.](#)
2. **USAGE(NORMAL)** , bu kuyruğun bir iletim kuyruğu olmadığını belirtir.
3. ORANGE.LOCAL.QUEUE, bu komut başarısız olur. Bir kuyruğun var olan tanımlanmasının üzerine yazmak istiyorsanız **REPLACE** özniteliğini kullanın, ancak aynı zamanda bkz. [“Yerel kuyruk özniteliklerinin değiştirilmesi” sayfa 142.](#)

İlgili bilgiler

[QLOCAL ' I TANIMLA](#)

Varsayılan nesne özniteliklerinin görüntülenmesi

Bir IBM MQ nesnesi tanımlandığında varsayılan nesneden alınan öznitelikleri görüntülemek için **DISPLAY QUEUE** komutunu kullanabilirsiniz.

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 aşağıdaki komutu kullanın:

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Bu komutun 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 öznitelige sahip, çünkü kuyruk derinliğini izleyerek, kuyruğun dolu olmasını sağlayabilirsiniz.

İlgili bilgiler

[GÖRÜNTÜLE](#)

[TANIMLAR](#)

Yerel kuyruk tanımlamasının kopyalanması

DEFINE komutundaki **LIKE** özniteliğini kullanarak bir kuyruk tanımlamasını kopyalayabilirsiniz.

Örneğin:

```
DEFINE QLOCAL (MAGENTA.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE)
```

Bu komut, özgün kuyruğumuzun (ORANGE.LOCAL.QUEUE)değeri, sistem varsayılan yerel kuyruğundan değil. Kuyruğu yaratırken tam olarak girildiği şekilde kopyalanacak kuyruğun adını girin. Ad küçük harfli karakterler içeriyorsa, adı tek tırnak işareti içine alın.

Ayrıca, bir kuyruk tanımlamasını kopyalamak için **DEFINE** komutunun bu formunu kullanabilir, ancak özgün özniteliklerde bir ya da daha fazla değişiklik yerine başkasını koyabilirsiniz. Örneğin:

```
DEFINE QLOCAL (THIRD.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE) +  
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 4194304.

Not:

1. Bir **DEFINE** komutunda **LIKE** özniteliğini kullandığınızda, yalnızca kuyruk özniteliklerini kopyalısınız. Kuyruklardaki iletileri kopyalamasınız.
2. Yerel bir kuyruk tanımladığınızda, **LIKE** belirtilmeden, **DEFINE LIKE (SYSTEM.DEFAULT.LOCAL.QUEUE)**.

İlgili bilgiler

TANIMLAR

Yerel kuyruk özniteliklerinin değiştirilmesi

You can change queue attributes in two ways, using either the **ALTER QLOCAL** command or the **DEFINE QLOCAL** command with the **REPLACE** attribute.

“Yerel kuyruk tanımlanması” sayfa 140’inde, kuyruk **ORANGE.LOCAL.QUEUE** tanımlandı. Örneğin, bu kuyruktaki ileti uzunluğu üst sınırını 10.000 bayt olarak azaltmak istediğinizi varsayalım.

- **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. Kuyruk artık etkin yerleştirildi, oysa daha önce engellendi. Put enabled is the default, as specified by the queue **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 bilgiler

ALTER kuyrukları

ALTER QLOCAL

TANIMLAR

QLOCAL ' I TANIMLA

Yerel kuyruğun temizlenmesi

Yerel bir kuyruğu temizlemek için **CLEAR** komutunu kullanabilirsiniz.

For example, to delete all the messages from a local queue called **MAGENTA.QUEUE**, use the following command:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

Not: Fikrinizi değiştirmenize olanak tanıyan bir bilgi istemi yoktur; Enter tuşuna bastığınızda iletiler kaybedilir.

Şu durumda bir kuyruğu temizleyemezsiniz:

- Eşitleme noktası altında kuyruğa yerleştirmeye ilişkin kesinleştirilmemiş iletiler var.
- Şu anda bir uygulama kuyruğu açık durumda.

İlgili bilgiler

QLOCAL ' I TEMİZLE

Yerel Kuyruğun Silinmesi

Yerel bir kuyruğu silmek için **DELETE QLOCAL** MQSC komutunu kullanabilirsiniz.

Kuyruğun üzerinde kesinleştirilmemiş iletiler varsa, kuyruk silinemez. Ancak, kuyruğun bir ya da daha çok kesinleştirilmiş ileti varsa ve kesinleştirilmemiş ileti varsa, yalnızca **PURGE** seçeneğini belirtseniz silinebilir. Örneğin:

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

PURGE yerine **NOPURGE** belirtilmesi, kuyruğun kesinleştirilmiş iletiler içerip içermediği için silinmemesini sağlar.

İlgili bilgiler

[QLOCAL SIL](#)

Tarama kuyrukları

IBM MQ , kuyrukta bulunan iletilerin içeriğine bakmak için kullanabileceğiniz bir örnek kuyruk tarayıcısı sağlar. Tarayıcı hem kaynak hem de yürütülür dosya biçimlerinde sağlanır.

MQ_INSTALLATION_PATH , IBM MQ ' in kurulu olduğu üst düzey dizini temsil eder.

Windows

Windows' ta, örnek kuyruk tarayıcısının dosya adları ve yolları aşağıdaki gibidir:

Kaynak

```
MQ_INSTALLATION_PATH\tools\c\samples\
```

Yürütülebilir

```
MQ_INSTALLATION_PATH\tools\c\samples\bin\amqsbcg.exe
```

Linux

UNIX

UNIX ve Linux' da dosya adları ve yollar şu şekildedir:

Kaynak

```
MQ_INSTALLATION_PATH/samp/amqsbcg0.c
```

Yürütülebilir

```
MQ_INSTALLATION_PATH/samp/bin/amqsbcg
```

Bu örnek için iki giriş değıştirgesi, kuyruk adı ve kuyruk yöneticisi adı gereklidir. Örneğin:

```
amqsbcg SYSTEM.ADMIN.QMGREVENT.tpp01 saturn.queue.manager
```

Bu komuttan gelen tipik sonuçlar [Şekil 17 sayfa 144](#) içinde 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

No more messages
MQCLOSE
MQDISC

```

Şekil 17. Kuyruk tarayıcısından tipik sonuçlar

İlgili bilgiler

[Tarayıcı örnek programı](#)

Büyük kuyrukların etkinleştirilmesi

IBM MQ , 2 GB ' den büyük kuyrukları destekler.

Windows Windows sistemlerinde, büyük dosyalar için destek, herhangi bir ek etkinleştirme yapılmaksızın kullanılabilir.

Linux **UNIX** UNIX ve Linux sistemlerinde, 2 GB ' den büyük kuyruk dosyaları yaratabilmeniz için öncelikle büyük dosya desteğini belirttik olarak etkinleştirmeniz gerekir. Bunun nasıl yapacağına ilişkin bilgi için işletim sistemi belgelerinize bakın.

Tar gibi bazı yardımcı programlar 2 GB ' den büyük dosyalarla başa çıkamaz. Büyük dosya desteğini etkinleştirmeden önce, kullandığınız yardımcı programlarla ilgili kısıtlamalar hakkında bilgi almak için işletim sistemi belgelerinizi denetleyin.

Kuyruklar için gereksinim duyduğunuz depolama miktarını planlamaya ilişkin bilgi için, platforma özgü performans raporları için [IBM MQ Family-Performance Reports](#) başlıklı konuya bakın.

Diğer ad kuyruklarıyla çalışma

Başka bir kuyruğa ya da konuya dolaylı olarak başvuruda bulunmak için bir diğer ad kuyruğu tanımlayabilirsiniz.

V9.0.0.1 V9.0.1



Uyarı: Dağıtım listeleri, konu nesnelerini gösteren diğer ad kuyruklarını kullanmaz. IBM MQ 9.0.1 ve IBM MQ 9.0.0 Fix Pack 1, bir diğer ad kuyruğu dağıtım listesindeki bir konu nesnesini gösteriyorsa, IBM MQ , MQRC_ALIAS_BASE_Q_TYPE_ERROR dizisini döndürür.

Bir diğer ad kuyruğunun başvurduğu kuyruk aşağıdakilerden herhangi biri olabilir:

- Yerel bir kuyruk (bkz. [“Yerel kuyruk tanımlanması” sayfa 140](#)).
- Uzak kuyruğun yerel tanımlaması (bkz. [“Uzak kuyruğun yerel tanımlaması yaratılması” sayfa 192](#)).
- Bir konu.

Diğer ad kuyruğu, gerçek bir kuyruk değil, yürütme sırasında gerçek (ya da hedef) kuyrukta çözülen bir tanımlamadır. 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 çözümlenemiyor. Ancak, bir diğer ad kuyruğu, yerel kuyruk yöneticisinin bir üyesi olduğu kümelerin başka bir yerinde tanımlanmış diğer ad kuyruklarına çözebilir. Ek bilgi için [Ad çözümlülüğü](#) konusuna bakın.

Diğer ad kuyrukları için kullanışlıdır:

- Farklı uygulamalara hedef kuyruk için farklı erişim düzeyleri düzeyleri verilsin.
- Farklı uygulamaların aynı kuyrukla aynı kuyrukla çalışması için izin verilmesi. (Farklı varsayılan öncelikler ya da farklı varsayılan kalıcılık değerleri atamak istersiniz.)
- Bakım, geçiş ve iş yükü dengelemesi basitleştiriyor. (Diğer adı kullanmaya devam eden, uygulamanızı değiştirmek zorunda kalmadan hedef kuyruk adını değiştirmek isteyebilirsiniz.)

For example, assume that 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 koyması 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.

TARGQ özniteliklerinin 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:

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGET (YELLOW.QUEUE)
```

Bu komut, MY.ALIAS.QUEUE (KUYRUK) kuyruğunu YELLOW.QUEUE. 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:

```
ALTER QALIAS (MY.ALIAS.QUEUE) TARGET (MAGENTA.QUEUE)
```

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(Kuyruk), ancak bu iletiye ileti koymasına izin verilmiyor.

Aşağıdaki komut, etkinleştirilen ve uygulama ALPHA için geçersiz kılınan bir diğer adı tanımlıyor:

```
DEFINE QALIAS (ALPHAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (ENABLED) +  
GET (DISABLED)
```

Aşağıdaki komut, devre dışı bırakılan ve uygulama BETA için etkinleştirilen bir diğer adı tanımlar:

```
DEFINE QALIAS (BETAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (DISABLED) +  
GET (ENABLED)
```

ALPHA, ALPHAS.ALIAS.QUEUE adlı MQI çağrılarında, BETABETA kuyruk adını kullanır. BETAS.ALIAS.QUEUE. Her ikisi de aynı kuyruğa erişir, ama farklı şekillerde.

Kuyruk diğer adlarını tanımladığınızda, bu öznitelikleri yerel kuyruklarla aynı şekilde tanımladığınızda LIKE ve REPLACE özniteliklerine de kullanabilirsiniz.

Diğer ad kuyruklarıyla diğer komutların 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 QUEUE (ALPHAS.ALIAS.QUEUE)
```

Use the **ALTER QALIAS** command to alter the base queue name, to which the alias resolves, where the **force** option forces the change even if the queue is open:

```
ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGQ(ORANGE.LOCAL.QUEUE) FORCE
```

Bu kuyruk diğer adını silmek için **DELETE QALIAS** komutunu kullanın:

```
DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

Kuyruk açık olan bir uygulama varsa, diğer ad kuyruğunu silemezsiniz.

İlgili bilgiler

[QALIAS ALTER](#)

[QALIAS ' YI](#)

[QALIAS SIL](#)

[Dağıtım listeleri](#)

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 147.

İlgili bilgiler

[Ölü-harf kuyrukları](#)

[Teslim edilmeyen iletiler sorun giderme](#)

[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 bilgiler

[Ölü-harf kuyrukları](#)

[Teslim edilmeyen iletiler sorun giderme](#)

IBM i' in ölü 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. DQ 'ların kullanımı ve önlenmesi hakkında bilgi için bkz. "[Ölü-harfli kuyruklarla çalışma](#)" sayfa 147.

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 iletisi, kural çizelgesindeki bir girişle eşleştiğinde, 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 bilgiler

[Ölü-harf kuyrukları](#)

[Teslim edilmeyen iletiler sorun giderme](#)

Ölü-harfli kuyruk işleyici kuralları çizelgesi, DLQ işleyicinin IBM i DLQ 'ya gelen iletileri nasıl işleyeceęini tanımlar.

DLQ işleyici kuralları çizelgesi, DLQ işleyicinin, DLQ 'ya gelen iletileri nasıl işleyeceęini 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.

Denetim verileri

Bu bölümde, bir DLQ işleyici kuralları çizelgesinde denetim verileri girdisine dahil edebildiğiniz anahtar sözcükler açıklanmaktadır. Aşağıdakileri unutmayın:

- Bir anahtar sözcüğün varsayılan değeri (varsa) altı çizilir.
- Dikey çizgi (|) alternatifleri ayırır. Bunlardan yalnızca birini belirtebilirsiniz.
- Tüm anahtar sözcükler isteğe bağlıdır.

INPUTQ (*QueueName* | ' ')

İşlemek istediğiniz DLQ ' nun adı:

1. **STRMQMDLQ** komutuna parametre olarak belirttiğiniz herhangi bir UDLMMSGQ değeri (ya da *DFT), kural çizelgesindeki INPUTQ değerini geçersiz kılar.
2. **STRMQMDLQ** komutuna parametre olarak boş bir UDLMMSGQ değeri belirtirseniz, kural çizelgesindeki INPUTQ değeri kullanılır.
3. **STRMQMDLQ** komutuna parametre olarak boş bir UDLMMSGQ değeri ve kural çizelgesinde boş bir INPUTQ değeri belirtirseniz, sistemin varsayılan ölü harf kuyruğu kullanılır.

INPUTQM (*QueueManagerName* | ' ')

INPUTQ anahtar sözcüğündeki DLQ adlı DLQ ' nun sahibi olan kuyruk yöneticisinin adı.

Bir kuyruk yöneticisi belirtmezseniz ya da kural çizelgesinde INPUTQM (') belirtmezseniz, sistem kuruluş için varsayılan kuyruk yöneticisini kullanır.

RETRYINT (*Aralık* | 60)

DLQ işleyicinin, ilk denemede işlenmeyen ve yinelenen girişimler istendiği, DLQ ' daki iletileri yeniden işleme girişiminde bulunan saniye cinsinden aralık. Varsayılan değer olarak, yeniden deneme aralığı 60 saniyedir.

WAIT (*EVET* | *HAYIR* | *nnn*)

DLQ işleyicinin, işlenebileceği başka ileti olmadığını saptadığında DLQ ' ya daha fazla ileti gelmesini beklemesi gerekip gerekmediğini belirtir.

EVET

DLQ işleyicinin süresiz olarak beklemesine neden olur.

HAYIR

DLQ 'nın, DLQ' un boş olduğunu saptadığında ya da işleyebileceği bir ileti içerip içermediği saptanırken, DLQ işleyicisine neden olur.

nnn

DLQ işleyicinin, kuyruğun boş olduğunu ya da işleyebileceği hiçbir ileti içermediği algıladıktan sonra, sonlandırılmadan önce yeni çalışma gelmesi için *nnn* saniye beklemesine neden olur.

Meşgul DLQ ' lar için WAIT (YES) ve WAIT (NO) ya da WAIT (*nnn*) belirleyin Düşük düzeyde etkinlik içeren DLQ ' lar için. DLQ işleyicinin sona erdirilmesine izin verilirse, tetikleme kullanarak yeniden başlatın.

You can supply the name of the DLQ as an input parameter to the **STRMQMDLQ** command, as an alternative to including control data in the rules table. If any value is specified both in the rules table and on input to the **STRMQMDLQ** command, the value specified on the **STRMQMDLQ** command takes precedence.

Not: Kurallar çizelgesinde bir denetim verileri girişi varsa, bu giriş, çizelgedeki ilk girişe girmelidir.

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 karar verir.

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 üzerindeki iletilerin eşleştirilmesiyle karşılaştırılır) bir tanımlamayla başlar. Daha sonra, işlem anahtar sözcüklerini (DLQ işleyicinin eşleşen bir iletiyi işleyeceğini saptayanlar) açıklanır.

Örüntü eşleşen anahtar sözcükler bir örnekte açıklanmıştır. IBM i 'un ölü-harfli kuyruğunda hangi iletilerin eşleştirileceği ile ilgili değerleri belirlemek için bu anahtar sözcükleri kullanın. Tüm kalıp eşleştirme anahtar sözcükleri isteğe bağlıdır.

APPLIDAT (*ApplIdentityData* |*)

İleti tanımlayıcısında belirtilen MQMD ' de belirtilen, DLQ üzerindeki iletinin *ApplIdentityData* değeri.

APPLNAME (*PutApplAdı* |*)

MQPUT ya da MQPUT1 çağrısını yayınlayan uygulamanın adı, DLQ üzerindeki iletinin MQMD ' nin *PutApplAdı* alanında belirtildiği gibi.

APPLTYPE (*PutApplTip* |*)

DLQ üzerindeki iletinin, ileti tanımlayıcısında, MQMD ' de belirtilen *PutApplType* değeri.

DESTQ (*QueueName* |*)

İletinin kaderinde bulunduğu ileti kuyruğunun adı.

DESTQM (*QueueManagerAdı* |*)

İletinin kaderinde bulunduğu ileti kuyruğunun kuyruk yöneticisi adı.

FEEDBACK (*Geribildirim* |*)

MsgType değeri MQMT_REPORT ise, *Geribildirim* raporun niteliyi açıklar.

Simgesel adları kullanabilirsiniz. Örneğin, MQFB_COA simgesel adını kullanarak, bunların hedef kuyruklarına ulaşmalarının doğrulanmasını gerektiren DLQ ' daki iletileri saptayın.

BIÇİM (*Biçim* |*)

İletinin göndericisinin, ileti verilerinin biçimini tanımlamak için kullandığı ad.

MSGTYPE (*MsgType* |*)

DLQ ' da iletinin ileti tipi.

Simgesel adları kullanabilirsiniz. Örneğin, yanıt gerektiren DLQ ' da bu iletileri tanımlamak için MQMT_REQUEST simgesel adını kullanabilirsiniz.

PERSIN (*Kalıcılık* |*)

İletinin kalıcılık değeri. (Bir iletinin kalıcılığı, kuyruk yöneticisinin yeniden başlatılıp yeniden başlatılıp başlatılmayacağını belirler.)

Simgesel adları kullanabilirsiniz. Örneğin, kalıcı olan DLQ ' da bu iletileri tanımlamak için MQPER_PERSISTENT simgesel adını kullanabilirsiniz.

NEDEN (ReasonCode|*)

İletinin DLQ ' ya neden konduğunu açıklayan neden kodu.

Simgesel adları kullanabilirsiniz. Örneğin, hedef kuyrukları dolu olduğu için, DLQ ' ya yerleştirilen iletileri tanımlamak için MQRC_Q_FULL simgesel adını kullanabilirsiniz.

REPLYQ (QueueName|*)

DLQ üzerindeki iletinin, ileti tanımlayıcısında, MQMD ' de belirtilen yanıtlama kuyruğu adı.

REPLYQM (QueueManagerAd|*)

REPLYQ anahtar sözcüğünde belirtilen yanıtlama kuyruğu kuyruk yöneticisi adı.

USERID (UserIDentifier|*)

İleti tanımlayıcısında belirtildiği gibi, DLQ ' da iletiyi oluşturan kullanıcının kullanıcı kimliği, MQMD.

IBM i

IBM üzerinde DLQ işlem anahtar sözcükleri

IBM i ölü-harf kuyruğunda eşleşen bir iletinin nasıl işleneceğini belirlemek için bu ölü harfli kuyruk işlemi anahtar sözcüklerini kullanın.

İŞLEM (AT | YOKSAYIN | YENİDEN DENE | FWD)

Bu kuralda tanımlanan örüntüyle eşleşen DLQ ' da 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.

GÇ

DLQ işleyicisine, iletiyi FWDQ anahtar sözcüğündeki belirtilen kuyruğa iletmesine neden olur.

ACTION anahtar sözcüğünü belirtmeniz gerekir. Bir işlemi gerçekleştirmek için yapılan deneme sayısı, RETRY anahtar sözcüğünün yönetilerek yönetilir. Denetim verilerinin RETRYINT anahtar sözcüğü, girişimler arasındaki aralığı denetler.

FWDQ (QueueName | & DESTQ | & REPLYQ)

ACTION anahtar sözcüğünü seçtiğinizde, iletinin iletileceği ileti kuyruğunun adı.

QueueName

İleti kuyruğunun adı. FWDQ (") geçerli değil.

& DESTQ

MQDLH yapısındaki *DestQName* alanında kuyruk adını alın.

& YANITLA

İleti tanımlayıcısındaki *ReplyToQ* alanından kuyruk adını alın, MQMD.

Hata iletilerini önlemek için, ileti örüntüsünde REPLYQ (? *), FWDQ (& REPLYQ) değerini belirten bir kural boş *ReplyToQ* alanı içeren bir iletiyle eşleşir.

FWDQM (QueueManagerAd | & DESTQM | & REPLYQM | ')

İletinin iletileceği kuyruğun kuyruk yöneticisi.

QueueManagerAd

ACTION (FWD) anahtar sözcüğünü seçtiğinizde, iletinin iletildiği kuyruğa ilişkin kuyruk yöneticisi adı.

& DESTQM

MQDLH yapısındaki *DestQMGrAd* alanından kuyruk yöneticisi adını alın.

& REPLYQM

İleti tanımlayıcısındaki *ReplyToQMGr* alanından kuyruk yöneticisi adını alın, MQMD.

..

Varsayılan değer olan FWDQM ("), yerel kuyruk yöneticisini tanıtır.

ÜSTBILGI (EVET|HAYIR)

MQDLH ' nin, ACTION (FWD) istendiği bir iletide kalması gerekip gerekmediğini belirleyin. Varsayılan olarak, MQDLH iletiyle kalır. HEADER anahtar sözcüğü, FWD dışındaki işlemler için geçerli değil.

PUUTAT (DEF| CTX)

DLQ işleyicisine hangi iletilerin yerleştirilmesi gerektiği konusunda yetki:

DÖF

DLQ işleyicisinin yetkisi olan iletileri yerleştirir.

CTX

İletilerin, ileti bağlamındaki kullanıcı kimliğinin yetkisiyle konmasına neden olur. PUTAUT (CTX) değerini belirlerseniz, diğer kullanıcıların kimliğini varsaymanız için yetki verilmelidir.

RETRY (RetryCount|1)

Bir işlemi (denetim verilerinin RETRYINT anahtar sözcüğünde belirlenen aralıkta) denemek için 1-999.999.999 aralığında yer alan sayı.

Not: Belirli bir kuralı uygulamak için DLQ işleyicisinin yaptığı girişimler sayısı, DLQ işleyicisinin yürürlükteki örneğine özgüdür; sayım, yeniden başlatma işlemleri boyunca devam etmez. DLQ işleyicisini yeniden başlatıyorsanız, bir kuralı uygulamak için yapılan girişlerin sayısı sıfıra 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 yoksayılır.
- 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 değiştirgeler önemli karakterler içermediği için geçersizdir:

FORMAT(' ')
FORMAT()
FORMAT()
FORMAT

- Genel arama karakterleri desteklenir. Soru işaretini (?) herhangi bir tek karakterin yerine, sonda boşluk dışında bir boşluk dışında kullanabilirsiniz. Sıfır ya da daha çok bitişik karakter yerine yıldız (*) işaretini 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 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, 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.

Ölü harf kuyruğu işleyicisi, IBM i ölü-harflı 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 bulunduğunda, 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 İletiler ve neden kodları içinde açıklanmıştır.) Kural çizelgesinde istediğiniz zaman değişiklik yapabilirsiniz, ancak DLQ işleyicisi yeniden başlatılınca kadar bu değişiklikler yürürlüğe girmez.
4. DLQ işleyicisi, iletilerin 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 koşturarak koşut zamanlı olarak çalışabilir. Ancak, bir DLQ ile DLQ işleyicisi arasında bire bir ilişki olması daha olağan bir durum.

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 iletilerin görülmesini 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 işleyicisini etkinleştirmek için DLQ 'yı (DLQ) boşaltmak için belirli önlemleri alın. Örneğin, DLQ 'da ileti bırakan ACTION (IGNORE) kullanmayın. (Tablodaki diğer kurallar tarafından açık bir şekilde ele alınmayan iletiler için ACTION (IGNORE) varsayıldığını 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 REALLYY . DEAD . QUEUEkuyruğuna iletilebilmesine neden olur. Böyle bir kuralınız yoksa, iletilerin DLQ ' da süresiz olarak kalabileceğini kabul edin.

IBM i IBM üzerinde bir DLQ işleyicisi kural çizelgesi örneği

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.rul
```

- 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 yerleştirilmesi için, diğer kullanıcıların kimliğini de üstlenmek için yetkilendirilmiş olmanız gerekir.

`runmqdlq` komutuna ilişkin ek bilgi için [runmqdlq](#) başlıklı konuya bakın.

İlgili bilgiler

[Ölü-harf kuyrukları](#)

[Teslim edilmeyen iletiler sorun giderme](#)

Örnek DLQ işleyicisi **amqsd1q**

In addition to the dead-letter queue handler invoked using the **runmqdlq** command, IBM MQ provides the source of a sample DLQ handler **amqsd1q** with a function that is similar to that provided by **runmqdlq**.

Gereksinimlerinizi karşılayan bir DLQ işleyicisi sağlamak için `amqsd1q` uygulamasını özelleştirebilirsiniz. Örneğin, iletileri ölü harf üstbilgileri olmadan işleyebilecek bir DLQ işleyicisi istediğinize karar verebilirsiniz. (Hem varsayılan DLQ işleyicisi hem de örnek, `amqsd1q`, yalnızca bir ölü-harfli üstbilgisiyle başlayan DLQ ' daki iletileri işleme alan `MQDLH`. `MQDLH` ile başlamayan iletiler hata olarak tanımlanır ve DLQ ' da süresiz olarak kalır.)

`MQ_INSTALLATION_PATH` , IBM MQ ' in kurulu olduğu üst düzey dizini temsil eder.

IBM MQ for Windows' ta, amqsdlq kaynağı dizinde sağlanıyor:

```
MQ_INSTALLATION_PATH\tools\c\samples\d1q
```

ve derlenen sürüm şu dizinde sağlanıyor:

```
MQ_INSTALLATION_PATH\tools\c\samples\bin
```

IBM MQ for UNIX ve Linux sistemlerinde, dizin içinde amqsdlq kaynağı sağlanır:

```
MQ_INSTALLATION_PATH/samp/d1q
```

ve derlenen sürüm şu dizinde sağlanıyor:

```
MQ_INSTALLATION_PATH/samp/bin
```

İstemci kipinde **amqsdlq** derlemesini de yapabilirsiniz. Daha fazla bilgi için bakınız: [Writing client proceural applications, Building applications for IBM MQ MQI clients, and Running applications in the IBM MQ MQI client environment.](#)

DLQ işleyici kuralları çizelgesi

Ölü-harflı 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 eşleştğinde 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 bilgiler

[Ölü-harf kuyrukları](#)

[Teslim edilmeyen iletiler sorun giderme](#)

DLQ denetim verileri

Bir denetim verileri girdisine, bir denetim verileri girdisine anahtar sözcükler dahil edebilirsiniz.

Not:

- Düşey çizgi (|) alternatifleri ayırır, yalnızca biri belirlenebilir.
- Tüm anahtar sözcükler isteğe bağlıdır.

INPUTQ (*QueueName* | ' ')

İşlemek istediğiniz DLQ ' nun adı:

1. runmqdlq komutuna değiştirge olarak sağladığınız herhangi bir INPUTQ değeri, kural çizelgesindeki INPUTQ değerini geçersiz kılar.
2. If you do not specify an INPUTQ value as a parameter to the Runmqdlq command, but you **yap** specify a value in the rules table, the INPUTQ value in the rules table is used.
3. If no DLQ is specified or you specify INPUTQ(' ') in the rules table, the name of the DLQ belonging to the queue manager with the name that is supplied as a parameter to the Runmqdlq command is used.
4. Runmqdlq komutuna değiştirge olarak bir Inputq değeri belirtmezseniz ya da kural çizelgesinde bir değer olarak, kural çizelgesindeki INPUTQM anahtar sözcüğünde adı belirtilen kuyruk yöneticisine ait DLQ kullanılır.

INPUTQM (*QueueManagerName* | ' ')

INPUTQ anahtar sözcüğündeki DLQ adlı DLQ ' nun sahibi olan kuyruk yöneticisinin adı:

1. runmqdlq komutuna deęiřtirge olarak saęladığınız INPUTQM deęeri, kural izelgesindeki INPUTQM deęerini geersiz kılar.
2. Runmqdlq komutuna parametre olarak bir INPUTQM deęeri belirtmezseniz, kural izelgesindeki INPUTQM deęeri kullanılır.
3. Kuyruk yneticisi belirtilmediyse ya da kural izelgesinde INPUTQM (") belirtilirse, kuruluř iin varsayılan kuyruk yneticisi kullanılır.

RETRYINT (Aralık|60)

DLQ iřleyicinin, ilk denemede iřlenememesi gereken ve yinelenen giriřimler istendięi iin, DLQ iřleyicinin DLQ zerindeki iletileri yeniden iřlemesi gerektięi saniye cinsinden aralık. Varsayılan deęer olarak, yeniden deneme aralıęı 60 saniyedir.

WAIT (EVET|HAYIR|nnn)

DLQ iřleyicinin, iřleyebileceęi bařka ileti olmadıęını saptadıęında DLQ ' ya daha fazla ileti gelmesini beklemesi gerekip gerekmedięini belirler.

EVET

DLQ iřleyicisi sresiz olarak bekler.

HAYIR

DLQ iřleyicisi, DLQ ' un boř olduęunu algıladıęında ya da iřleyebileceęi bir ileti ierip iermedięi saptanırken sona erer.

nnn

DLQ iřleyicisi, kuyruęun boř olduęunu ya da iřleyebileceęi hibir ileti iermedięi algıladıktan sonra, sona erdirilmeden nce yeni alıřma iin nnn saniye bekler.

Meřgul DLQ ' lar iin WAIT (YES) ve WAIT (NO) ya da WAIT (nnn) belirleyin Dřk dzeyde etkinlik ieren DLQ ' lar iin. DLQ iřleyicinin sona erdirilmesine izin verilirse, tetiklemeyi kullanarak yeniden aęırın. Tetikleme ile ilgili daha fazla bilgi iin [Tetikleyicileri kullanarak IBM MQ uygulamalarının bařlatılması](#) konusuna bakın.

An alternative to including control data in the rules table is to supply the names of the DLQ and its queue manager as input parameters to the Runmqdlq command. Hem kural izelgesinde hem de runmqdlq komutuna giriř olarak bir deęer belirtirseniz, runmqdlq komutunda belirtilen deęer ncelik kazanır.

Kural izelgesine bir denetim verileri giriři eklerseniz, bu giriř, izelgedeki **ilk** giriř olmalıdır.

DLQ kuralları (rntler ve iřlemler)

Kalıp eřleřtirme anahtar szcklerinin (l-harf kuyruęunda hangi iletilerin eřleřtięi) ve iřlem anahtar szcklerinin (DLQ iřleyicisinin eřleřen bir iletiyi nasıl iřleyeceęini belirleyen) bir aıklaması. rnek bir kural da saęlanmıřtır.

Kalıp eřleřtirme anahtar szckleri

DLQ zerindeki iletilerin eřleřtirildięi deęerleri belirlemek iin kullandıęınız kalıp eřleřtirmesi anahtar szckleri ařaęıdaki gibidir. (Tm rnt-eřleřen anahtar szckler isteęe baęlıdır):

APPLIDAT (ApplIdentityData|*)

DLQ zerindeki iletinin, ileti tanımlayıcısında, MQMD ' de belirtilen *ApplIdentityData* deęeri.

APPLNAME (PutApplAdi|*)

The name of the application that issued the MQPUT or MQPUT1 call, as specified in the *PutApplAd* field of the message descriptor, MQMD, of the message on the DLQ.

APPLTYPE (PutApplTip|*)

DLQ zerindeki iletinin ileti tanımlayıcısında, MQMD ' de belirtilen *PutApplType* deęeri.

DESTQ (QueueName|*)

İletin kaderinde bulunduęu ileti kuyruęunun adı.

DESTQM (QueueManagerAdi|*)

İletin kaderinde bulunduęu ileti kuyruęunun kuyruk yneticisinin adı.

FEEDBACK (Geribildirim|*)

MsgType değeri MQFB_REPORT ise, *Geribildirim* raporun niteliyi açıklar.

Simgesel adları kullanabilirsiniz. Örneğin, MQFB_COA simgesel adını kullanarak, hedef kuyruklarına varışlarının doğrulanabilmesi için DLQ ' da bu iletileri tanımlayabilirsiniz.

BIÇİM (Biçim|*)

İletinin göndericisinin, ileti verilerinin biçimini tanımlamak için kullandığı ad.

MSGTYPE (MsgType|*)

DLQ ' da iletinin ileti tipi.

Simgesel adları kullanabilirsiniz. Örneğin, yanıt ihtiyacı olan DLQ ' da bu iletileri tanımlamak için MQMT_REQUEST simgesel adını kullanabilirsiniz.

PERSIN (Kalıcılık|*)

İletinin kalıcılık değeri. (Bir iletinin kalıcılığı, kuyruk yöneticisinin yeniden başlatılıp yeniden başlatılıp başlatılmayacağını belirler.)

Simgesel adları kullanabilirsiniz. Örneğin, kalıcı olan DLQ ile ilgili iletileri tanımlamak için MQPER_PERSISTENT simgesel adını kullanabilirsiniz.

NEDEN (ReasonCode|*)

İletinin DLQ ' ya neden konduğunu açıklayan neden kodu.

Simgesel adları kullanabilirsiniz. Örneğin, hedef kuyrukları dolu olduğu için, DLQ ' ya yerleştirilen iletileri tanımlamak için MQRC_Q_FULL simgesel adını kullanabilirsiniz.

REPLYQ (QueueName|*)

DLQ üzerindeki iletinin, ileti tanımlayıcısında, MQMD ' de belirtilen yanıtlama kuyruğunun adı.

REPLYQM (QueueManagerAd|*)

DLQ üzerindeki iletinin, ileti tanımlayıcısında, MQMD ' de belirtildiği gibi, yanıtlanacak kuyruğa ilişkin kuyruk yöneticisinin adı.

USERID (UserIdentifier|*)

DLQ üzerindeki iletiyi oluşturan kullanıcının kullanıcı kimliği (DLQ ileti tanımlayıcısı, MQMD 'de belirtildiği gibi), DLQ 'daki iletinin tanıtıcısı.

İşlem anahtar sözcükleri

Eşleşen bir iletinin nasıl işleneceğini tanımlamak için kullanılan işlem anahtar sözcükleri şunlardır:

İŞLEM (AT | YOKSAYIN | YENİDEN DENE | FWD)

Bu kuralda tanımlanan örüntüyle eşleşen DLQ ' da herhangi bir ileti için yapılacak işlem.

At

DLQ ' dan iletiyi silin.

IGNORE

İletiyi DLQ ' ya bırakın.

Yeniden deneme

İletiyi hedef kuyruğuna yerleştirmeye yönelik ilk girişim başarısız olursa, yeniden deneyin.

RETRE anahtar sözcüğü, bir işlemi gerçekleştirmek için yapılan deneme sayısını ayarlar. Denetim verilerinin RETRYINT anahtar sözcüğü, girişimler arasındaki aralığı denetler.

GÇ

İletiyi, FWDQ anahtar sözcüğündeki kuyruğa iletin.

ACTION anahtar sözcüğünü belirtmeniz gerekir.

FWDQ (QueueName| & DESTQ | & REPLYQ)

ACTION (FWD) istendiğinde iletinin iletileceği ileti kuyruğunun adı.

QueueName

İleti kuyruğunun adı. FWDQ (") geçerli değil.

& DESTQ

MQDLH yapısındaki *DestQName* alanında kuyruk adını alın.

& YANITLA

İleti tanımlayıcısındaki *ReplyToQ* alanından kuyruk adını alın, MQMD.

Bir kural boş *ReplyToS* alanına sahip bir iletiyle eşleştğinde hata iletilerini önlemek için, ileti örüntüsünde REPLYQ (? *) değerini belirtin.

FWDQM (*QueueManagerAdı* | & DESTQM | & REPLYQM | ' ')

İletinin iletileceği kuyruğun kuyruk yöneticisi.

QueueManagerAdı

ACTION (FWD) istendiğinde iletinin iletileceği kuyruğun kuyruk yöneticisinin adı.

& DESTQM

MQDLH yapısındaki *DestQMGrAd* alanından kuyruk yöneticisi adını alın.

& REPLYQM

İleti tanımlayıcısındaki *ReplyToQMGr* alanından kuyruk yöneticisi adını alın, MQMD.

''

Varsayılan değer olan FWDQM (''), yerel kuyruk yöneticisini tanıtır.

ÜSTBILGI (EVET|HAYIR)

MQDLH ' nin, ACTION (FWD) istendiği bir iletide kalması gerekip gerekmediğini belirleyin. Varsayılan olarak, MQDLH iletiyle kalır. HEADER anahtar sözcüğü, FWD dışındaki işlemler için geçerli değil.

PUUTAT (DEF| CTX)

DLQ işleyicisine hangi iletilerin yerleştirilmesi gerektiği konusunda yetki:

DÖF

DLQ işleyicisinin yetkisini kullanarak iletileri girin.

CTX

İleti bağlamındaki, kullanıcı kimliğinin yetkisi olan iletileri yazın. PUTAUT (CTX) belirlerseniz, diğer kullanıcıların kimliğini üstlenmek için yetkili olmanız gerekir.

RETRY (*RetryCount*|1)

Bir işlemi (denetim verilerinin RETRYINT anahtar sözcüğünde belirlenen aralıkta) denemek için 1-999.999.999 aralığında yer alan sayı. Belirli bir kuralı uygulamak için DLQ işleyicisinin yaptığı girişimler sayısı, DLQ işleyicisinin yürürlükteki örneğine özgüdür; sayım, yeniden başlatma işlemleri boyunca devam etmez. DLQ işleyicisi yeniden başlatılırsa, bir kuralı uygulamak için yapılan girişlerin sayısı sıfıra sıfırlanır.

Örnek kural

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 tarafından engellendikleri için, hedef kuyruğa alma hedefine yönelik olarak DQ ' ya yönelik üç deneme girişiminde bulunmaya karar verir.

Bir kuralda kullanabileceğiniz tüm anahtar sözcükler bu bölümün geri kalan kısmında açıklanmıştır. Aşağıdakileri unutmayın:

- Bir anahtar sözcüğün varsayılan değeri (varsa) altı çizilir. Çoğu anahtar sözcük için, varsayılan değer * (yıldız işareti), herhangi bir değerle eşleşir.
- Düşey çizgi (!) alternatifleri ayırır, yalnızca biri belirlenebilir.
- 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 yoksayılır.
- 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 sınırlayıcı 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 sonun 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 kurala ilişkin yeniden deneme sayısını 1 artırır. İlk deneme başarısız olursa, DQ işleyicisi RETRY anahtar sözcüğünde belirtilen sayıyla eşleşmeye çalışılıncaya 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 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 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 koşut olarak koşut zamanlı olarak çalışabilir. Ancak, bir DLQ ile DLQ işleyicisi arasında bire bir ilişki olması daha olağan bir durum.

İlgili bilgiler

Ölü-harf kuyrukları

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; atılamayan ya da başka kuyruklara iletilenmeyen 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. 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 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 bilgiler

Ölü-harf kuyrukları

Teslim edilmeyen iletiler sorun giderme

runmqdlq (çalıştırma-harfli kuyruk işleyicisi çalıştır)

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, SYSYSTEM.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 bilgiler

[ALTER QMODEL](#)

[QMODEL ' I TANIMLA](#)

[QMODEMI SIL](#)

[GÖRÜNTÜLE](#)

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 tanımlanması” sayfa 166

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 166

Bir denetim konusu nesnesini görüntülemek için **DISPLAY TOPIC** MQSC komutunu kullanın.

“Denetim konusu özniteliklerinin değiştirilmesi” sayfa 167

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 168

DEFINE komutundaki **LIKE** özniteliğini kullanarak bir konu tanımını kopyalayabilirsiniz.

“Denetim konusu tanımlamasının silinmesi” sayfa 168

Bir yönetim konusunu silmek için **DELETE TOPIC** MQSC komutunu kullanabilirsiniz.

İlgili bilgiler

[Yönetimle ilgili konu nesneleri](#)

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 birleştirilmesibaşlıklı](#) 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 ağaçlarıbaşlıklı](#) 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 166.
- 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 167 konusuna da bakın.

İlgili bilgiler

[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 özneliği 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 özneliklerin ne olduğunu görmek için aşağıdaki komutu kullanın:

```
DISPLAY TOPIC (SYSTEM.DEFAULT.TOPIC)
```

İlgili bilgiler

[KONUYU GÖRÜNTÜLE
TANITIM](#)

Denetim konusu özneliklerinin 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 özneliği değiştirir; bu konuya, bu konuya ilişkin varsayılan öncelik önceliği 5 'e teslim edilir; diğer tüm öznelikler aynı kalır.

- **DEFINE** komutunu kullanarak:

```
DEFINE TOPIC(ORANGE.TOPIC) DEFPRTY(5) REPLACE
```

Bu komut, bu konuya teslim edilen iletilerin varsayılan önceliğini değiştirir. Diğer tüm özneliklere 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 bilgiler

[KONUYU DEĞİŞTİR](#)

[KONUYU GÖRÜNTÜLE](#)

Denetim konusu tanımlamasının kopyalanması

DEFINE komutundaki **LIKE** özniteliğini kullanarak bir konu tanımını kopyalayabilirsiniz.

Örneğin:

```
DEFINE TOPIC (MAGENTA.TOPIC) +  
LIKE (ORANGE.TOPIC)
```

Bu komut bir konu yaratır, **MAGENTA.TOPIC**, özgün konuyla aynı özniteliklerle **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 özniteliklerin özniteliklerinde 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 bilgiler

[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 dizesini 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 bilgiler

[KONUYU SIL](#)

Aboneliklerle çalışma

Abonelikleri yönetmek için MQSC komutlarını kullanın.

Abonelikler, **SUBTYPE** özniteisinde tanımlanan üç tipten biri olabilir:

ADMIN

Bir kullanıcı tarafından yönetimsel olarak tanımlanır.

proxy

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 169](#)

Denetim aboneliği yaratmak için **DEFINE SUB** MQSC komutunu kullanın. Varsayılan yerel abonelik tanımlamasında tanımlı olan varsayılan değeri de kullanabilirsiniz. Ya da, abonelik özelliklerini varsayılan yerel abonelikten (SYSTEM.DEFAULT.SUB , sistem kurulduğunda yaratılmış olur.

[“Aboneliklerin özniteliklerini görüntüleme” sayfa 170](#)

Kuyruk yöneticisi tarafından bilinen herhangi bir aboneliğin yapılandırılmış özniteliklerini görüntülemek için **DISPLAY SUB** komutunu kullanabilirsiniz.

[“Yerel abonelik özniteliklerinin değiştirilmesi” sayfa 171](#)

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 tanımlamasının kopyalanması” sayfa 171](#)

DEFINE komutundaki **LIKE** özniteliğini kullanarak bir abonelik tanımlamasını kopyalayabilirsiniz.

[“Yerel aboneliği silme” sayfa 172](#)

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 tanımlamasında 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 dizgisinde, 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 dizgisi adının eşleşmesi gerekmez.
- Hedef ve konu dizgisinin değerleri dışında, gösterilen tüm öznitelik değerleri varsayılan değerlerdir. Burada yalnızca bir şekil olarak gösterilir. Varsayılan değerlerin istediğiniz ya da değiştirilmediğinden eminseniz, bunları atlayabilirsiniz. Ayrıca bkz. [“Aboneliklerin özniteliklerini görüntüleme” sayfa 170](#).
- Aynı kuyruk yöneticisiyle ilgili olarak TURUNCU ile yerel bir aboneliğiniz varsa, bu komut başarısız olur. Bir kuyruğun var olan tanımlamasının üzerine yazmak istiyorsanız **REPLACE** özniteliğini kullanın, ancak aynı zamanda bkz. [“Yerel abonelik özniteliklerinin değiştirilmesi” sayfa 171](#).

- SUBQ KUYRUĞU yoksa, bu komut başarısız olur.

İlgili bilgiler

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 örnek 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 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 TÜMÜ **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(414D5120414141202020202020202020EE921E4E20002A03)
NUMMSG(0)
```

Bir yönetici aboneliği tanımladığınızda, bu abonelik varsayılan abonelikten belirttik olarak belirtmediğiniz herhangi bir özniteliği alır; bu, SYSTEM.DEFAULT.SUB adlı bir abonelikten explicitly olarak adlandırılır. Bu varsayılan özniteliklerin ne olduğunu görmek için aşağıdaki komutu kullanın:

```
DISPLAY SUB (SYSTEM.DEFAULT.SUB)
```

İlgili bilgiler

[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 bilgiler

[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 bilgiler

[ALT](#)

Yerel aboneliđi silme

You can use the MQSC command **DELETE SUB** to delete a local subscription.

```
DELETE SUB(ORANGE)
```

SUBID seçeneđini kullanarak da bir aboneliđi silebilirsiniz:

```
DELETE SUB SUBID(414D51204141412020202020202020EE921E4E20002A03)
```

İlgili bilgiler

[SUB SIL](#)

Abonelikte iletilerin denetlenmesi

Bir abonelik tanımlandığında, bu abonelik bir kuyrukla ilişkilendirilir. Bu abonelikte eşleşen yayınlanan iletiler bu kuyruđa konabiliyor.

Bu görev hakkında

Aşağıdaki **runmqsc** komutlarının yalnızca ileti alan abonelikleri gösterdiğine dikkat edin.

Bir aboneliđin řu anda kuyruđa alınmış olan iletileri denetlemek için aşağıdaki adımları izleyin:

Yordam

1. Bir abonelik tipi **DISPLAY SBSTATUS** (*sub_name*) **NUMMSG** için kuyruđa alınan iletileri denetlemek için bkz. [“Aboneliklerin özniteliklerini görüntüleme” sayfa 170](#).
2. **NUMMSG** değeri sıfırdan büyükse, abonelikle ilişkili kuyruđu **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 [“Tarama kuyrukları” sayfa 143](#).

İlgili bilgiler

[SBSTATUS GÖRÜNTÜLE](#)

Hizmetlerle çalışma

Hizmet nesneleri, ek işlemlerin bir kuyruk yöneticisinin parçası olarak yönetilebileceđi 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 belirtilmiş. Sunucu hizmeti nesnesi, belirlenen bir kuyruk yöneticisi başlatıldığında yürütölen bir programın tanımlamasıdır. Sunucu hizmeti nesneleri, genellikle uzun bir süre boyunca çalıştırılan programları tanımlar. Örneđin, bir sunucu hizmeti nesnesi, bir tetikleme izleme işlemi yürütmek için kullanılabilir (örneğin, **runmqtrm**).

Bir sunucu hizmeti nesnesinin yalnızca bir eşgörünümü kořutzamanlı 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 bilgiler

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 koşut zamanlı olarak yürütülebilir. 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 174

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 bilgiler

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 175

Bir hizmet başlatıldığında, hizmet işleminin başlatıldığı ortam, kuyruk yöneticisinin ortasından edinilir. It is possible to define additional environment variables to be set in the environment of the service process by adding the variables you want to define to one of the `service.env` environment override files.

İlgili bilgiler

[Hizmeti durdur](#)

[Hizmeti başlat](#)

Ek ortam değişkenleri

Bir hizmet başlatıldığında, hizmet işleminin başlatıldığı ortam, kuyruk yöneticisinin ortasından edinilir. It is possible to define additional environment variables to be set in the environment of the service process by adding the variables you want to define to one of the `service.env` environment override files.

Ortam değişkenleri ekleyebileceğiniz dosyalar

Ortam değişkenleri ekleyebileceğiniz olası iki dosya vardır:

Makine kapsamı `service.env` dosyası

Bu dosya şu konumda bulunur:

- **Linux** **UNIX** UNIX and Linux sistemlerinde `/var/mqm`.
- **Windows** The data directory selected during installation on Windows systems.

Kuyruk yöneticisi kapsamı `service.env` dosyası

Bu dosya, kuyruk yöneticisi verileri dizininde bulunur. Örneğin, QMNAME adlı bir kuyruk yöneticisine ilişkin ortam geçersiz kılma dosyasının yeri şöyledir:

- **Linux** **UNIX** UNIX 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, kuyruk yöneticisi kapsam dosyasındaki tanımların, makine kapsamı dosyasındaki tanımlardan öncelikli olarak işlenmesini sağlar.

`service.env` içinde belirtilebilecek ortam değişkenleri.

Herhangi bir ortam değişkeni `service.env` içinde belirtilebilir. For example, if the IBM MQ service runs a number of commands, it might be useful to set the `PATH` user variable in the `service.env` file. Değişkeni, ortam değişkenleri olamaz olarak ayarladığınız değerler; 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 bilgileri içerebilir. Bazı hizmetler (örneğin Telemetry) kendi sınıf yolunu ayarlar. `service.env` içinde tanımlanan `CLASSPATH` bu dosyaya eklenir.

The format of the variables defined in the file, `service.env` is a list of name and value variable pairs. Her değişken yeni bir satırda tanımlanmalıdır ve beyaz alan da içinde olmak üzere, her bir değişken belirttik olarak tanımlandığı gibi alınır.

`service.env` örneği

```
#*****#
##                                     *#
## <N_OCO_COPYRIGHT>                 *#
## Licensed Materials - Property of IBM *#
##                                     *#
## 63H9336                             *#
## (C) Copyright IBM Corporation 2005, 2023. *#
##                                     *#
## <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 tanımlarına değiştirilebilir ekler” sayfa 176

Bir hizmet nesnesi tanımlanmasında, 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 tanımlarına değiştirilebilir ekler

Bir hizmet nesnesi tanımlanmasında, 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:

-   UNIX and Linux sistemlerinde, IBM MQ veri dizini yeri /var/mqm/ olur.
-  Windows sistemlerinde, IBM MQ veri dizininin yeri, IBM MQ kurulumu sırasında seçilen veri dizini yeridir.

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 176.

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 değiştireceği 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 tanıtcısını gösteren bir simgedir.

Ortak belirteçlerin listesi için bkz. [“Hizmet tanımlarına değiştirilebilir ekler” sayfa 176](#) .

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 177](#) içinde yapılan değişiklikleri alır.

Not: MQSC komutu (**STOP SERVICE**) yalnızca hizmet tanımında bir **STOPCMD** bağımsız değişkeni belirtilirse kullanılabilir.

İlgili bilgiler

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 , sistem günlüğüne yazılacak UNIX and Linux sistemi tarafından sağlanan komuttur.
+QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.

İlgili bilgiler

HİZMET

Bir kuyruk yöneticisi yalnızca bir kuyruk yöneticisi sona erdiğinde, bir komut hizmeti nesnesi kullanılması

Bu örnek, bir kuyruk yöneticisi yalnızca durdurulduğunda, işletim sisteminin sistem günlüğüne giriş yazan bir programı başlatmak için bir komut hizmeti nesnesinin nasıl tanımlanacak şekilde tanımlamayı 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 bilgiler

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 biçem 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 bilgiler

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('runmqtrm') +
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 bilgiler

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** özniteliği (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) +  
TRIGDPTH (100)+  
TRIGMPRI (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 özniteliği değeri mi?

TRIGTYPE (DEPTH)

Gerekli önceliğin (TRIGMPRI) ileti sayısı TRIGDPTH içinde belirtilen sayıya ulaştığında tetikleme olayının oluşturulduğunu 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 işlenmek 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. Dizede 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:\appl\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 programını kullanma

The **dmpmqmsg** utility (formerly **qload**) is incorporated into the product from IBM MQ 8.0. Formerly the **qload** utility has been available as SupportPac MO03.

Genel Bakış

dmpmqmsg yardımcı programı, bir kuyruğun ya da iletilerin içeriğini bir dosyaya kopyalamanıza ya da taşımanıza olanak sağlar. Bu dosya, gerekli olduğu şekilde saklanabilir ve daha sonra, iletileri yeniden kuyruğa yeniden yüklemek için kullanılır.

Önemli: Dosya, yardımcı program tarafından anlaşılacak belirli bir biçime sahiptir. Ancak, dosyayı yeniden yüklemeye önce bir düzenleyicide güncelleştirmek için, dosya insan tarafından okunabilir. Dosyayı düzenliyorsanız, biçimini değiştirmemeniz gerekir.

Olası kullanım alanları şunlardır:

- Kuyruklardaki iletilerin kaydedilmesi, dosyaya kaydedilmesi. Büyük olasılıkla arşivleme amacıyla ve daha sonra yeniden bir kuyruğa yeniden yükleyin.
- Bir kuyruğa önceden kaydetmiş olduğunuz iletilerle bir kuyruk yeniden yükleniyor.
- Kuyruktan eski iletiler kaldırılıyor.
- ' Gerekli olursa iletiler arasında doğru zamanı bile korusanız da, saklanan bir konumdan gelen iletileri yeniden yanıtlayın.



Uyarı: SupportPac MO03 , yerel ya da istemci bağ tanımını belirtmek için **-1** değiştirgesini kullandı. **-1** , **-c** parametresiyle değiştirildi.

-P is now used for code page information instead of **-c**.

Komutla ve kullanılabilir parametrelerle ilgili ek bilgi için [dmpmqmsg](#) konusuna bakın.

Example of using the **dmpmqmsg** utility on Linux, using a Windows machine

Aynı kuyruk yöneticisinde başka bir kuyruğa (*Q2*) taşımak istediğiniz kuyrukta (*Q1*) iletileri olan bir Linux makinesinde kuyruk yöneticiniz var. **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.

Gördüğünüz Linux makinesinde:

```
display ql(Q1) CURDEPTH
      2 : display ql(Q1) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q1)
      TYPE(QLOCAL)
      CURDEPTH(4)
```

MQSERVER ortam değişkenini Linux'taki kuyruk yöneticisini işaret edecek şekilde ayarlayın. Örneğin:

```
set MQSERVER=SYSTEM.DEF.SVRCONN/TCP/veracruz.x.com(1414)
```

Burada *veracruz* makinenin adıdır.

Run the **dmpmqmsg** utility to read from the queue, *Q1*, and store the output in `c:\temp\mqqload.txt`.

Uzak istemci olarak kuyruk yöneticisine (*QM_VER*), MQSERVER tarafından kurulan Linux anasisteminde ve kapısında çalıştırılarak bağlantı kurun. Şu özniteliği kullanarak uzak istemci olarak bağlantı elde edin: `-c`.

```
dmpmqmsg -m QM_VER -i Q1 -f c:\temp\mqqload.txt -c
Read      - Files:      0  Messages:      4  Bytes:      22
Written - Files:      1  Messages:      4  Bytes:      22
```

The output file `c:\temp\mqqload.txt` contains text, using a format that the **dmpmqmsg** utility understands.

On the Windows machine, issue the **dmpmqmsg** command (using the `-o` option instead of the `-i` option) to load queue (*Q2*) on the Linux machine from a file on the Windows machine:

```
dmpmqmsg -m QM_VER -o Q2 -f c:\temp\mqqload.txt -c
Read      - Files:      1  Messages:      4  Bytes:      22
Written - Files:      0  Messages:      4  Bytes:      22
```

Linux makinesinde, kuyrukta dosyanın geri yüklendiği dört ileti olduğunu unutmayın.

```
display ql(Q2) CURDEPTH
      6 : display ql(Q2) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q2)
      TYPE(QLOCAL)
      CURDEPTH(4)
```

Linux makinesinde,

Özgün kuyruktan gelen iletileri silin.

```
clear qllocal(Q1)
      4 : clear qllocal(Q1)
AMQ8022: IBM MQ queue cleared.
```

Özgün kuyrukda başka ileti olmadığını doğrulayın:

```
display ql(Q1) CURDEPTH
      5 : display ql(Q1) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q1)
      TYPE(LOCAL)
      CURDEPTH(0)
```

Komutun ve parametrelerinin açıklaması için bkz. [dmpmqmsg](#) .

İlgili kavramlar

“[dmpmqmsg yardımcı programını kullanma örnekleri](#)” sayfa 183

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 183' 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 183, 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 183 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 183, 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 nesnelerini yönetme

Bu kısımda, MQSC komutları kullanılarak uzak kuyruk yöneticilerindeki IBM MQ nesnelerinin nasıl denetleneceği ve iletilerin ve yanıt iletilerinin hedefini denetlemek için uzak kuyruk nesnelerinin nasıl kullanılacağı anlatılıyor.

Yordam

- Uzak IBM MQ nesnelерinin nasıl yönetileceđi hakkında bilgi için ařađıdaki alt konulara bakın:
 - [“Kanallar ve uzak kuyruđa alma” sayfa 185](#)
 - [“Yerel kuyruk yöneticisinden uzak denetim” sayfa 186](#)
 - [“Uzak kuyruđun yerel tanımlaması yaratılması” sayfa 192](#)
 - [“Dađıtılmıř ađlara iliřkin zamanuyumsuz komutların sona erdiđi denetleniyor” sayfa 194](#)
 - [“Uzak kuyruk tanımlamalarının diđer adlar olarak kullanılması” sayfa 197](#)
 - [“Veri dönüřtürme” sayfa 197](#)

Kanallar ve uzak kuyruđa alma

Yerel bir kuyruk yöneticisi ile uzak kuyruk yöneticisi arasında tek yönlü iletiřim bađlantısı sađlamak için bir kanal ayarlayabilirsiniz. Kanal, uzak kuyruk yöneticisinde herhangi bir sayıda kuyruđa gönderilen iletileri taşıyabilir. Uzak kuyruk yöneticisinin yerel kuyruk yöneticisi tarafından gönderilen iletilere yanıt vermesini istiyorsanız, uzak kuyruk yöneticisinden gelen yanıtları yerel kuyruk yöneticisine geri göndermek için ikinci bir kanal ayarlayabilirsiniz.

Kuyruk yöneticisi, bir ileti göndererek, yanıt almak için bir ileti ve gerekiyorsa, başka bir kuyruk yöneticisiyle iletiřim kurar. Alma kuyruk yöneticisi ařađıdaki gibi olabilir:

- Aynı makinede
- Aynı konumdaki başka bir makinede (ya da dünyanın diđer tarafında bile)
- Yerel kuyruk yöneticisiyle aynı altyapıda çalıştırma
- IBM MQ tarafından desteklenen başka bir altyapıda çalıştırma

Bu iletiler ařađıdakilerden kaynaklanabilir:

- Verileri bir düđümden diđerine aktaran kullanıcı-yazılı uygulama programları
- PCF komutlarını ya da MQAI olanađını kullanan kullanıcı tarafından yazılan yönetim uygulamaları
- IBM MQ Explorer.
- Gönderen kuyruk yöneticileri:
 - Başka bir kuyruk yöneticisine kod ekleme olayı iletileri
 - Dolaylı kipte (komutların başka bir kuyruk yöneticisinde çalıştırıldıđı) bir **runmqsc** komutundan yayınlanan MQSC komutları

Uzak kuyruk yöneticisine bir ileti gönderilmeden önce, yerel kuyruk yöneticisinin iletilerin gelişini algılamak ve bunları ařađıdakilerden oluşan bir yöntemle aktarmak için bir mekanizmaya gerek vardır:

- En az bir kanal
- İletim kuyruđu
- Kanal başlatıcısı

Uzak kuyruk yöneticisinin bir ileti alması için dinleyici gereklidir.

Kanal, iki kuyruk yöneticisi arasında tek yönlü bir iletiřim bađlantısıdır ve uzak kuyruk yöneticisinde herhangi bir sayıda kuyruk için gönderilen iletileri taşıyabilir.

Kanalın her bir ucunda ayrı bir tanım vardır. Örneđin, bir son gönderen ya da sunucu ise, diđer ucun bir alıcı ya da istekte bulunanın olması gerekir. Basit bir kanal, yerel kuyruk yöneticisi uçtaki *gönderen kanal tanımlaması* ve uzak kuyruk yöneticisi uçtaki bir *alıcı kanalı tanımlaması* ile oluşur. İki tanım aynı ada sahip olmalı ve birlikte tek bir ileti kanalı oluşturur.

Uzak kuyruk yöneticisinin yerel kuyruk yöneticisi tarafından gönderilen iletilere yanıt vermesini istiyorsanız, yanıtları yerel kuyruk yöneticisine geri göndermek için ikinci bir kanal ayarlayın.

Kanalları tanımlamak için **DEFINE CHANNEL** MQSC komutunu kullanın. Bu bölümde, başka bir şekilde belirtilmediđi sürece, kanallarla ilgili örnekler varsayılan kanal özniteliklerini kullanır.

İletilerin her birinde bir ileti kanalı aracı (MCA) vardır ve iletilerin gönderilmesini ve alınmasını denetleyen bir ileti kanalı vardır. MCA ileti iletim kuyruğundan ileti alır ve bunları kuyruk yöneticileri arasındaki iletişim bağlantısına yerleştirir.

İletim kuyruğu, MCA 'nın bunları alması ve uzak kuyruk yöneticisine göndermeden önce iletileri geçici olarak tutan özelleştirilmiş bir yerel kuyruktur. Bir *uzak kuyruk tanımlaması* üzerinde iletim kuyruğunun adını belirtmenizi sağlar.

Bir MCA 'nın birden çok iş parçacığını kullanarak ileti aktarmasına izin verebilirsiniz. Bu işlem *Pipelining* olarak bilinir. Pipelining, MCA 'nın iletileri daha verimli bir şekilde aktarmasını, kanal performansını artırmasını sağlar. Bir kanalın pipelining kullanacak şekilde yapılandırılmasına ilişkin ayrıntılar için [Kanalların öznitelikleri](#) başlıklı konuya bakın.

“[Uzaktan yönetim için kanallar ve iletim kuyrukları hazırlama](#)” sayfa 187 , uzak denetimi ayarlamak için bu tanımların nasıl kullanılacağını size bildirir.

Dağıtılmış kuyruğa alma sırasının oluşturulmasıyla ilgili ek bilgi için [Dağıtılmış kuyruğa alma bileşenleri](#) başlıklı konuya bakın.

İlgili bilgiler

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

Kümeleri kullanarak uzaktan yönetim

Dağıtılmış kuyruklama kullanan bir IBM MQ ağında, her kuyruk yöneticisi bağımsızdır. Bir kuyruk yöneticisinin başka bir kuyruk yöneticisine ileti göndermesi gerekiyorsa, bu, ileti göndermek istediği her kuyruk için bir iletim kuyruğu, uzak kuyruk yöneticisine bir kanal ve uzak kuyruk tanımlaması tanımlamalıdır.

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.

Yerel kuyruk yöneticisinden uzak denetim

MQSC ve PCF komutlarını kullanarak, uzak kuyruk yöneticisini yerel bir kuyruk yöneticisinden denetleyebilirsiniz.

Hem MQSC hem de PCF komutları için kuyruklar ve kanalların hazırlanması aynı. Bu bölümde, örneklerin anlaşılması daha kolay olduğu için, MQSC komutlarını gösterirler. Denetim programlarını PCF komutları kullanarak yazmakla ilgili ek bilgi için bkz. [“IBM MQ Programları Komut Biçimlerinin Kullanılması” sayfa 21.](#)

MQSC komutlarını uzak bir kuyruk yöneticisine etkileşimli olarak ya da komutları içeren bir metin dosyasından göndermenizi sağlar. Uzak kuyruk yöneticisi aynı makinede ya da başka bir makinede, daha çok farklı bir makinede olabilir. Aşağıdakiler de içinde olmak üzere, diğer IBM MQ ortamlarındaki kuyruk yöneticilerini uzaktan yönetebilirsiniz:

-  UNIX
-  Linux

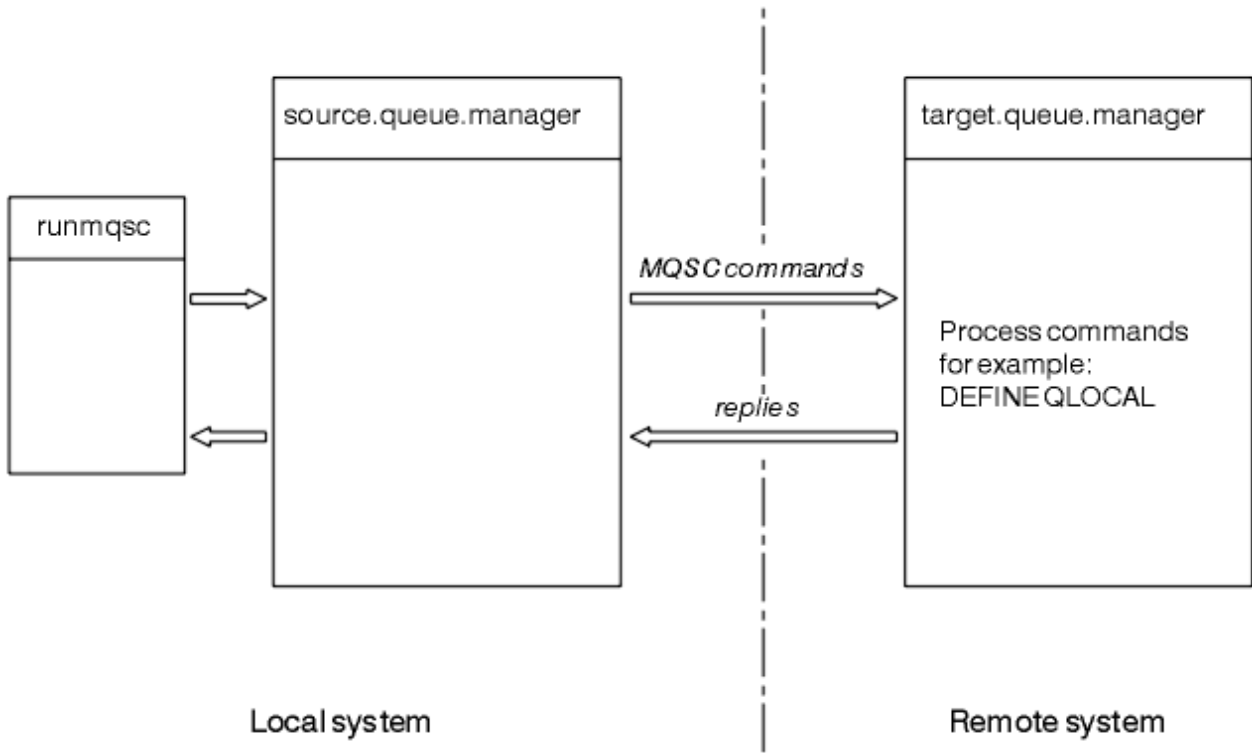
- **Windows** Windows
- **IBM i** IBM i
- **z/OS** z/OS

Uzak denetimi uygulamak için belirli nesnelere yaratmanız gerekir. Özel gereksiniminiz yoksa, varsayılan değerler (örneğin, ileti uzunluğu üst sınırı için) yeterlidir.

Uzak denetim için kuyruk yöneticilerinin hazırlanması

Uzak denetim için kuyruk yöneticilerini hazırlamak için MQSC komutları nasıl kullanılır.

Şekil 18 sayfa 187 shows the configuration of queue managers and channels that you need for remote administration using the **runmqsc** command. `source.queue.manager` nesnesi, MQSC komutlarını verebileceğiniz ve bu komutların sonuçlarının (işletmen iletileri) döndürüldüğü kaynak kuyruk yöneticisidir. `target.queue.manager` nesnesi, komutları işleyen ve herhangi bir işletmen iletilisi oluşturan hedef kuyruk yöneticisinin adıdır.



Şekil 18. MQSC komutlarını kullanarak uzaktan yönetim

Her iki sistemde de, henüz yapmadıysanız:

- `crtmqm` komutunu kullanarak kuyruk yöneticisini ve varsayılan nesnelere yaratın. Daha fazla bilgi için bkz. [crtmqm](#).
- Start the queue manager, using the `strmqm` command. Ek bilgi için bkz. [strmqm](#).

Hedef kuyruk yöneticisinde:

- Komut kuyruğu, `SYSTEM.ADMIN.COMMAND.QUEUE` (Kuyruk), var olmalıdır. Bu kuyruk, bir kuyruk yöneticisi yaratıldığında varsayılan olarak yaratılır.

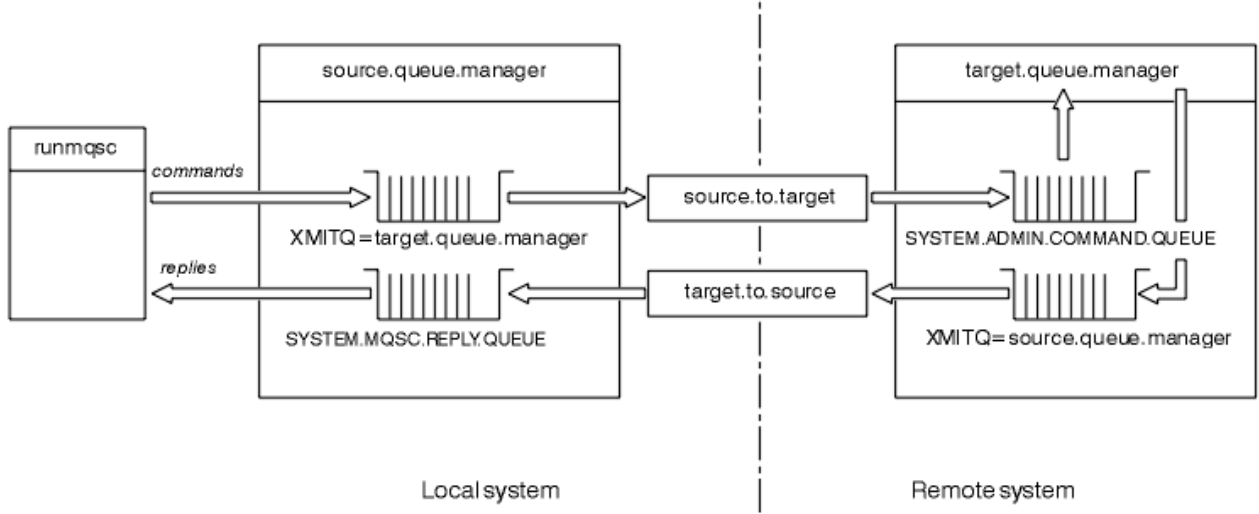
Bu komutları yerel olarak ya da Telnet gibi bir ağ olanağı üzerinden çalıştırmanız gerekir.

Uzaktan yönetim için kanallar ve iletim kuyrukları hazırlama

Uzak denetim için kanallar ve iletim kuyrukları hazırlamak üzere MQSC komutları nasıl kullanılır.

MQSC komutlarını uzaktan çalıştırmak için, her bir yön için bir kanal ve bunların ilişkili iletim kuyrukları için iki kanal ayarlayın. Bu örnek, iletim tipi olarak TCP/IP 'yi kullandığınızı ve ilişkili TCP/IP adresini bildiğinizi varsayar.

The channel `source.to.target` is for sending MQSC commands from the source queue manager to the target queue manager. Göndereni `source.queue.manager` konumunda ve günlük nesnesi `target.queue.manager` konumunda. The channel `target.to.source` is for returning the output from commands and any operator messages that are generated to the source queue manager. Ayrıca, her kanal için bir iletim kuyruğu da tanımlamanız gerekir. Bu kuyruk, alan kuyruk yöneticisinin adını verilen yerel bir kuyruğdur. Bir kuyruk yöneticisi diğer adı kullanmıyorsanız, uzak yönetimin çalışması için XMITQ adının uzak kuyruk yöneticisi adıyla eşleşmesi gerekir. [Şekil 19 sayfa 188](#) , bu yapılandırmayı özetler.



Şekil 19. Uzak denetim için kanalların ve kuyrukların ayarlanması

Kanalların ayarlanmasıyla ilgili ek bilgi için [Dağıtılmış kuyruklama yapılandırması](#) başlıklı konuya bakın.

Kanallar, dinleyiciler ve iletim kuyruklarının tanımlanması

Kaynak kuyruk yöneticisinde (`source.queue.manager`), kanalları, dinleyiciyi ve iletim kuyruğunu tanımlamak için aşağıdaki MQSC komutlarını verin:

1. Kaynak kuyruk yöneticisinde gönderici kanalını tanımlayın:

```
DEFINE CHANNEL ('source.to.target') +
CHLTYPE(SDR) +
CONNNAME (RHX5498) +
XMITQ ('target.queue.manager') +
TRPTYPE(TCP)
```

2. Kaynak kuyruk yöneticisinde günlük nesnesi kanalını tanımlayın:

```
DEFINE CHANNEL ('target.to.source') +
CHLTYPE(RCVR) +
TRPTYPE(TCP)
```

3. Kaynak kuyruk yöneticisinde dinleyici tanımlayın:

```
DEFINE LISTENER ('source.queue.manager') +
TRPTYPE (TCP)
```

4. Kaynak kuyruk yöneticisinde iletim kuyruğunu tanımlayın:

```
DEFINE QLOCAL ('target.queue.manager') +
USAGE (XMITQ)
```

Kanal, dinleyici ve iletim kuyruğunu yaratmak için, hedef kuyruk yöneticisine (`target.queue.manager`) aşağıdaki komutları verin:

1. Hedef kuyruk yöneticisinde gönderen kanalı tanımlayın:

```
DEFINE CHANNEL ('target.to.source') +
CHLTYPE(SDR) +
CONNNAME (RHX7721) +
XMITQ ('source.queue.manager') +
TRPTYPE(TCP)
```

2. Hedef kuyruk yöneticisinde günlük nesnesi kanalını tanımlayın:

```
DEFINE CHANNEL ('source.to.target') +
CHLTYPE(RCVR) +
TRPTYPE(TCP)
```

3. Hedef kuyruk yöneticisinde dinleyici tanımlayın:

```
DEFINE LISTENER ('target.queue.manager') +
TRPTYPE (TCP)
```

4. Hedef kuyruk yöneticisinde iletim kuyruğunu tanımlayın:

```
DEFINE QLOCAL ('source.queue.manager') +
USAGE (XMITQ)
```

Not: Gönderen kanalı tanımlamalarında CONNNAME özniteliği için belirtilen TCP/IP bağlantı adları yalnızca şekil içindir. Bu, bağlantının *diğer* uçta bulunan makinenin ağ adıdır. Ağınız için uygun değerleri kullanın.

Dinleyicilerin ve kanalların başlatılması

Dinleyicilere ve kanallara başlamak için MQSC komutlarının kullanılması.

Her iki dinleyici de aşağıdaki MQSC komutlarını kullanarak başlatın:

1. Start the listener on the source queue manager, `source.queue.manager`, by issuing the following MQSC command:

```
START LISTENER ('source.queue.manager')
```

2. Start the listener on the target queue manager, `target.queue.manager`, by issuing the following MQSC command:

```
START LISTENER ('target.queue.manager')
```

Aşağıdaki MQSC komutlarını kullanarak her iki gönderen kanalı da başlatın:

1. Aşağıdaki MQSC komutunu vererek, kaynak kuyruk yöneticisine (`source.queue.manager`) gönderen kanalını başlatın:

```
START CHANNEL ('source.to.target')
```

2. Aşağıdaki MQSC komutunu vererek, hedef kuyruk yöneticisinde (`target.queue.manager`) gönderen kanalını başlatın:

```
START CHANNEL ('target.to.source')
```

Kanalların otomatik tanımlaması

MQSC komutunu, ALTER QMGR (ya da PCF komutu Change Queue Manager) kullanarak kuyruk yöneticisi nesnesini güncelleyerek, alıcı ve sunucu bağlantısı tanımlamalarının otomatik tanımlamasını etkinleştirmenizi sağlar.

IBM MQ , gelen bağlantı isteği alırsa ve uygun bir alıcı ya da sunucu bağlantısı kanalı bulamazsa, otomatik olarak bir kanal yaratır. Otomatik tanımlar, IBM MQ: SYSTEM.AUTO.RECEIVER ve SYSTEM.AUTO.SVRCONN.

Kanal tanımlamalarının otomatik olarak yaratılmasına ilişkin ek bilgi için [Kanalların hazırlanması](#) başlıklı konuya bakın. Kümelere ilişkin kanalların otomatik olarak tanımlanması hakkında bilgi için bkz. [Otomatik tanımlı kanallarla çalışma](#).

Uzak denetim için komut sunucusunun yönetilmesi

Komut sunucusunun nasıl başlatılacağı, durdurulacağı ve görüntüleneceği. Bir komut sunucusu, PCF komutlarının, MQAI ' nin ve ayrıca uzak yönetim için de yer alan tüm denetim için zorunludur.

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.

Not: Uzaktan yönetim için, hedef kuyruk yöneticisinin çalışır durumda olduğundan emin olun. Ters durumda, komutlar içeren iletiler, gönderildikleri kuyruk yöneticisini bırakamaz. Bunun yerine, bu iletiler uzak kuyruk yöneticisine hizmet veren yerel iletim kuyruğunda kuyruğa alınır. Bu durumdan uzak dur.

Komut sunucusunu başlatmak ve durdurmak için ayrı denetim komutları vardır. Komut sunucusunun çalıştırıldığı, IBM MQ for Windows ya da IBM MQ for Linux (x86 ve x86-64 platformlarının) kullanıcıları, IBM MQ Explorer kullanılarak aşağıdaki bölümlerde açıklanan işlemleri gerçekleştirebilirler. Daha fazla bilgi için [“IBM MQ Explorer komutunu kullanarak yönetim”](#) sayfa 125 başlıklı konuya bakın.

Komut sunucusu başlatılıyor

Kuyruk yöneticisi özneliğinin değerine (*SCMDSERV*) bağlı olarak, komut sunucusu kuyruk yöneticisi başlatıldığında otomatik olarak başlatılır ya da el ile başlatılmalıdır. The value of the queue manager attribute can be altered using the MQSC command ALTER QMGR specifying the parameter SCMDSERV. Varsayılan olarak, komut sunucusu otomatik olarak başlatılır.

SCMDSERV MANUAL olarak ayarlandıysa, komutu kullanarak komut sunucusunu başlatın:

```
stmqscsv saturn.queue.manager
```

Burada saturn.queue.manager , komut sunucusunun başlatılmakta olduğu kuyruk yöneticidir.

Komut Sunucusunun Durumunun Görüntülenmesi

Uzak denetim için, hedef kuyruk yöneticindeki komut sunucusunun çalışmakta olduğundan emin olun. Çalıştırılmamışsa, uzak komutlar işlenemez. Komutları içeren iletiler, hedef kuyruk yöneticisinin komut kuyruğunda kuyruğa alınır.

Bir kuyruk yöneticisine ilişkin komut sunucusunun durumunu görüntülemek için, aşağıdaki MQSC komutunu verin:

```
DISPLAY QMSTATUS CMDSERV
```

Komut Sunucusunun Durdurulması

Bir önceki örnek tarafından başlatılan komut sunucusunu sona erdirmek için aşağıdaki komutu kullanın:

```
endmqcsv saturn.queue.manager
```

Komut sunucusunu iki şekilde durdurabilirsiniz:

- For a controlled stop, use the `sonmqcsv` command with the `-c` flag, which is the default.
- Anında durdurma için, `-i` işaretiyle `endmqcsv` komutunu kullanın.

Not: Kuyruk yöneticisinin durdurulması, kendisiyle ilişkili komut sunucusunu da sona erdirir.

Uzak kuyruk yöneticisinde MQSC komutları verilmesi

You can use a particular form of the `runmqsc` command to run MQSC commands on a remote queue manager.

MQSC komutlarını uzaktan işleyecekse, komut sunucusu hedef kuyruk yöneticisinde çalışıyor olmalıdır. (Bu, kaynak kuyruk yöneticisinde gerekli değildir). Bir kuyruk yöneticisine komut sunucusunun nasıl başlatılabilmesiyle ilgili bilgi için bkz. "[Uzak denetim için komut sunucusunun yönetilmesi](#)" sayfa 190.

Kaynak kuyruk yöneticisinde, aşağıdaki komutu yazarak, MQSC komutlarını dolaylı kipte etkileşimli olarak çalıştırabilirsiniz:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

`-w` işaretiyle `runmqsc` komutunun bu biçimi, komutların komut sunucusu giriş kuyruğunda (değiştirilmiş bir biçimde) konulduğu ve sırayla yürütüldüğü, dolaylı kipteki MQSC komutlarını çalıştırır.

Bir MQSC komutunu yazdığınızda, bu durumda uzak kuyruk yöneticisine yeniden yönlendirilir (bu durumda `target.queue.manager`). Zaman aşımı 30 saniye olarak ayarlanır; bir yanıt 30 saniye içinde alınmazsa, yerel (kaynak) kuyruk yöneticisinden aşağıdaki ileti oluşturulur:

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

MQSC komutlarını vermeyi durdurduğunuzda, yerel kuyruk yöneticisi, gelen tüm zaman aşımı yanıtlarını görüntüler ve başka yanıtları atar.

Kaynak kuyruk yöneticisi varsayılan yerel kuyruk yöneticisini varsayılan değer olarak kullanır. `runmqsc` komutunda `-m LocalQmgrName` seçeneğini belirtirseniz, komutları herhangi bir yerel kuyruk yöneticisi tarafından verilecek şekilde yönlendirebilirsiniz.

Dolaylı kipte, uzak kuyruk yöneticisinde bir MQSC komut dosyası da çalıştırabilirsiniz. Örneğin:

```
runmqsc -w 60 target.queue.manager < mycomds.in > report.out
```

Burada `mycomds.in`, MQSC komutlarını içeren bir dosyadır ve `report.out`, rapor dosyasıdır.

Komutların uzaktan verilmesi için önerilen yöntem

Uzak bir kuyruk yöneticisine komut verdiğinizde aşağıdaki yaklaşımı kullanmayı düşünün:

1. Bir komut dosyasında uzak sistemde çalıştırılacak MQSC komutlarını yerleştirin.
2. `runmqsc` komutundaki `-v` işaretini belirterek, MQSC komutlarınızı yerel olarak doğrulayın.
Başka bir kuyruk yöneticisinde MQSC komutlarını doğrulamak için `runmqsc` komutunu kullanamazsınız.
3. Komut dosyasının hatasız olarak çalışıp çalışmadığını denetleyin.
4. Uzak sistemde komut kütüğünü çalıştırın.

MQSC komutlarını uzaktan kullanarak sorunlarınız varsa

MQSC komutlarını uzaktan çalıştırmakta zorlanırsanız, aşağıdakine sahip olduğundan emin olun:

- Hedef kuyruk yöneticisinde komut sunucusu başlatıldı.
- Geçerli bir iletim kuyruğu tanımlıyor.
- Her ikisi için ileti kanallarının iki ucunu da tanımlayın:
 - Komutların gönderilmekte olduğu kanaldır.
 - Yanıtların döndürüleceği kanal.
- Kanal tanımlamasındaki doğru bağlantı adı (CONNNAME) belirlendi.
- İletişimciler, ileti kanallarını başlatmadan önce başlatıldı.
- Örneğin, bir kanal başlatıldıysa, ancak bir süre sonra sona erdirildikten sonra, bağlantı kesme aralığının süresinin dolmadığını kontrol edin. Bu, özellikle kanalları el ile başlatabilirseniz önemlidir.
- Hedef kuyruk yöneticisine (örneğin, uzak kuyruk yöneticisinden desteklenmeyen parametreleri içeren istekler) anlamlı olmayan bir kaynak kuyruk yöneticisinden gelen istekler gönderilir.

Ayrıca bkz. [“MQSC komutlarıyla ilgili sorunların çözülmesi” sayfa 18.](#)

Working with queue managers on z/OS

Bu kılavuzda açıklanan platformlardaki bir kuyruk yöneticisinden bir z/OS kuyruk yöneticisine MQSC komutları yayınlayabilirsiniz. However, to do this, you must modify the `runmqsc` command and the channel definitions at the sender.

Hedef kuyruk yöneticisinin z/OS'ta çalıştığını belirtmek için, özellikle kaynak düğümdeki `runmqsc` komutuna `-x` işaretini ekleyin:

```
runmqsc -w 30 -x target.queue.manager
```

Uzak kuyruğun yerel tanımlaması yaratılması

Uzak kuyruğun yerel tanımlaması, uzak kuyruk yöneticiliklerindeki bir kuyruğa gönderme yapan yerel bir kuyruk yöneticisinde bulunan tanımlamadır.

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 tanımların nasıl çalıştığını anlama

Bir uygulama yerel bir kuyruk yöneticisine bağlanır ve sonra bir MQOPEN çağrısını yayınlar. 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 çağrısı yayınlar. 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, `sa.tu.rn.queue.manager`. Hedef kuyruğun sahibi başka bir kuyruk yöneticisi var.

On the MQOPEN call, the application specifies these fields:

Alan değeri	Tanım
<i>ObjectName</i> CYAN.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 AUTOMOBILE.INSURANCE.QUOTE.QUEUE, uzak kuyruk yöneticisinde yerel bir kuyruk olarak tanımlanmalıdır.

RQMNAME (jupiter.queue.manager)

Hedef kuyruk AUTOMOBILE.INSURANCE.QUOTE.QUEUE.

XMITQ (INQUOTE.XMIT.QUEUE)

İletim kuyruğunun adını belirler. Bu isteğe bağlıdır; bir iletim kuyruğunun adı belirtilmediyse, uzak kuyruk yöneticisiyle aynı adı taşıyan bir kuyruk kullanılır.

Her iki durumda da, uygun iletim kuyruğu, bir ileti kuyruğu (MQSC komutlarında KULLANGE (XMITQ)) olduğunu belirten bir **Usage** özniteli yerel kuyruk olarak tanımlanmalıdır.

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 öznitelliklerini görüntülemek ya da değiştirmek için MQSC komutlarını kullanabilir ya da uzak kuyruk nesnesini silebilirsiniz. Örneğin:

- Uzak kuyruğun özniteliklerini görüntülemek için:

```
DISPLAY QUEUE (CYAN.REMOTE.QUEUE)
```

- 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.

İletim kuyruğunun tanımlanması

İletim kuyruğu, bir kuyruk yöneticisi iletileri uzak kuyruk yöneticisine bir ileti kanalı aracılığıyla ilettiğinde kullanılan yerel bir kuyruktur.

Kanal, uzak kuyruk yöneticisine tek yönlü bir bağlantı sağlar. İletiler, kanal bunları kabul edebilmesi için iletim kuyruğunda kuyruğa alınır. Bir kanal tanımladığınızda, ileti kanalının gönderme sonunda bir iletim kuyruğu adı belirlemeniz gerekir.

MQSC komut özneliği USAGE, bir kuyruğun iletim kuyruğu mu, yoksa normal bir kuyruk mu olduğunu tanımlar.

Varsayılan iletim kuyrukları

Kuyruk yöneticisi uzak bir kuyruk yöneticisine ileti gönderdiğinde, aşağıdaki sırayı kullanarak iletim kuyruğunu tanımlar:

1. Uzak bir kuyruğun yerel tanımlamasının XMITQ özneliğe adı geçen iletim kuyruğu.
2. Hedef kuyruk yöneticisiyle aynı adı taşıyan bir iletim kuyruğu. (Bu değer, uzak bir kuyruğun yerel tanımlamasının XMITQ ile ilgili varsayılan değerdir.)
3. Yerel kuyruk yöneticisinin DEFXMITQ özneliğe adı geçen iletim kuyruğu.

Örneğin, aşağıdaki MQSC komutu, `target.queue.manager'` a giden iletiler için `source.queue.manager` üzerinde varsayılan bir iletim kuyruğu yaratır:

```
DEFINE QLOCAL ('target.queue.manager') +  
DESCR ('Default transmission queue for target qm') +  
USAGE (XMITQ)
```

Uygulamalar, iletileri doğrudan bir iletim kuyruğuna ya da bir uzak kuyruk tanımlamasıyla dolaylı olarak gönderebilirler. Ayrıca bkz. [“Uzak kuyruğun yerel tanımlaması yaratılması” sayfa 192.](#)

Dağıtılmış ağlara ilişkin zamanuyumsuz komutların sona erdiği denetleniyor

Birçok komut, dağıtılmış bir ağda kullanıldığında zamanuyumsuz olur. Komuta bağlı olarak ve yayınlandığında ağ durumuna bağlı olarak, tamamlanması önemli ölçüde zaman alabilir. Kuyruk yöneticisi bir iletiyi tamamlamaya ilişkin bir ileti yayınlamıyor, bu nedenle komutun bittiğini denetmenin başka yollarına da ihtiyacınız var.

Bu görev hakkında

Bir kümede yaptığınız neredeyse herhangi bir yapılandırma değişikliği, zamanuyumsuz bir şekilde tamamlanır. Bunun nedeni, iç yönetimden kaynaklanır ve kümeler içinde çalışan döngüler güncelleştirilir.

Yayınlama/abone olma sıradüzenleri için, abonelikleri etkileyen herhangi bir yapılandırma değişikliğinin zamanuysuz olarak tamamlanacak olması olasılığı vardır. Bu, komutun adından her zaman belli değildir.

Aşağıdaki MQSC komutları zamanuysuz bir şekilde tamamlanabilir. Bu komutların her birinin bir PCF eşdeğeri vardır ve çoğu IBM MQ Explorer içinden de kullanılabilir. İş yükü olmayan küçük bir ağ üzerinde çalıştırıldığında, bu komutlar genellikle birkaç saniye içinde tamamlar. Ancak bu durum, daha büyük ve daha gürültülü ağlara yönelik bir durum değil. Ayrıca, **REFRESH CLUSTER** komutu, özellikle birden çok kuyruk yöneticisinde aynı anda yayınlandığında, çok daha uzun sürebilir.

Bu komutların sona erdiğine emin olmak için, uzak kuyruk yöneticilerindeki beklenen nesnelere var olduğunu doğrulayın.

Yordam

- ALTER QMGR

ALTER QMGR PARENT komutu için, istenen üst ilişkinin durumunu izlemek için `DISPLAY PUBSUB TYPE (PARENT) ALL` komutunu kullanın.

ALTER QMGR REPOS ve ALTER QMGR REPOSNL komutları için, tamamlanma işlemini doğrulamak için `DISPLAY CLUSQMGR QMTYPE` komutunu kullanın.

- KANALI TANIMLAYIN, ALTER KANALVE KANAL SIL

For all parameters listed in the table ALTER CHANNEL değiştirgeleleri, use the `DISPLAY CLUSQMGR` command to monitor when changes have been propagated to the cluster.

- AD LISTESINI TANIMLA, ALTER NAMELIST(ALTER NAMELIST) ve DELETE NAMELIST.

Bir **QMGR** nesnesinin **CLUSNL** özniteliğinde **NAMELIST** kullanırsanız, bir kuyruk ya da küme kanalı bu nesneyi etkileyebilir. Etkilenen nesne için uygun olan görüntü birimi.

`SYSTEM.QPUBSUB.QUEUE.NAMELIST` üzerindeki değişiklikler, yayınlama/abone olma sıradüzeninde yetkili sunucu aboneliklerinin oluşturulmasını ya da iptal edilmesini etkileyebilir. Bunu izlemek için `DISPLAY SUB SUBTYPE (PROXY)` komutunu kullanın.

- DEFINE kuyrukları, ALTER kuyrukları ve DELETE kuyrukları.

`DISPLAY QUEUE` komutu tarafından döndürülebilen parametrelerçizelgesinde listelenen tüm değiştirgeleler için, değişikliklerin kümeye ne zaman geçirildiğini izlemek için `DISPLAY QCLUSTER` komutunu kullanın.

- ALTve SUB SIL

Bir konu dizisine ilişkin ilk aboneliği tanımladığınızda, yayınlama/abone olma sıradüzeninde ya da yayınlama/abone olma kümesinde yetkili sunucu abonelikleri yaratabilirsiniz. Benzer şekilde, bir konu dizisine son aboneliği sildiğinizde, yayınlama/abone olma hiyerarşisinde ya da yayınlama/abone olma kümesinde yetkili sunucu aboneliklerini iptal edebilirsiniz.

Bir aboneliği tanımlayan ya da silmiş bir komutun tamamlandığını denetlemek için, dağıtılmış ağdaki diğer kuyruk yöneticilerinde beklenen proxy aboneliğinin olup olmadığını denetleyin. Bir kümede *doğrudan yönlendirme* kullanıyorsanız, beklenen yetkili aboneliğin kümedeki diğer kısmi havuzlarda var olup olmadığını denetleyin. Bir kümede *konu anasistem yönetmesi* kullanıyorsanız, beklenen yetkili aboneliğin eşleşen konu anasistemlerinde var olup olmadığını denetleyin. Aşağıdaki MQSC komutunu kullanın:

```
DISPLAY SUB(*) SUBTYPE (PROXY)
```

Bir küme ya da sıradüzeninde yayınlandığında, aşağıdaki eşdeğer abone olma ve MQI çağrılarına ilişkin aboneliği kaldırmak için aynı denetimi kullanın:

– MQSUB komutunu kullanarak abone olun.

– MQCO_REMOVE_SUB ile MQCLOSE komutunu kullanarak aboneliği kaldırın.

- KONUYU TANIMLAYIN, ALTER KONUSUVE KONUYU SIL

Kümelenmiş bir konuyu değiştirme, değiştirme ya da silme işlemini sona ermiş olarak denetlemek için, başlığı kümedeki diğer kısmı havuzlarda görüntüleyin (*doğrudan yönlendirme* kullanıyorsanız). ya da diğer konu anasistemlerinde (*konu anasistem yönetmesi* kullanıyorsanız).

For all parameters listed in the table DISPLAY TOPIC komutu tarafından döndürülebilen parametreler, use the DISPLAY TCLUSTER command to monitor when changes have been propagated to the cluster.

Not:

- **CLUSTER** parametresi, yayınlama/abone olma kümesindeki yetkili sunucu aboneliklerinin oluşturulmasını ya da iptal edilmesini etkileyebilir.
- **PROXYSUB** ve **SUBSCOPE** parametreleri, bir yayınlama/abone olma sıradüzeninde ya da yayınlama/abone olma kümesinde yetkili sunucu aboneliklerinin oluşturulmasını ya da iptal edilmesini etkileyebilir.
- Bunu izlemek için DISPLAY SUB SUBTYPE (PROXYSUB) komutunu kullanın.

• KÜMEYİ YENİLE

REFRESH CLUSTER komutunu çalıştırıyorsanız, küme komut kuyruğu derinliğini yoklatın. Nesneleri aramadan önce, sıfıra ulaşmasını bekleyin ve sıfır konumunda bekleyin.

1. Küme komut kuyruğu derinliğinin sıfır olup olmadığını denetlemek için aşağıdaki MQSC komutunu kullanın.

```
DISPLAY QL(SYSTEM.CLUSTER.COMMAND.QUEUE) CURDEPTH
```

2. Kuyruk derinliği sıfır değerine ulaşıncaya kadar denetimi yineleyin ve sonraki denetimde sıfır konumunda kalır.

REFRESH CLUSTER komutu nesneleri kaldırır ve yeniden oluşturur ve büyük yapılandırmalarda tamamlanması önemli bir zaman alabilir. Bkz. Yayınlama/abone olma kümeleri için Kümeyi Yenile.

• QMGR YENİLEMESİ TIPI (PROXYSUB)

REFRESH QMGR TYPE (PROXYSUB) komutunun bitmesini denetlemek için, dağıtılmış ağdaki diğer kuyruk yöneticilerinde yetkili sunucu aboneliklerinin düzeltilip düzeltildiğini denetleyin. Bir kümede *doğrudan yönlendirme* kullanıyorsanız, yetkili sunucu aboneliklerinin kümedeki diğer kısmı havuzlarda düzeltilip düzeltilmediğini denetleyin. Bir kümede *konu anasistem yönetmesi* kullanıyorsanız, beklenen yetkili aboneliklerin eşleşen konu anasistemlerinde düzeltilip düzeltilmediğini denetleyin. Aşağıdaki MQSC komutunu kullanın:

```
DISPLAY SUB(*) SUBTYPE(PROXYSUB)
```

• KÜMEYİ Sı

RESET CLUSTER komutunun tamamlandığını denetlemek için DISPLAY CLUSQMGR komutunu kullanın.

• QMGR RESET TIP (PUBSUB)

RESET QMGR komutunun tamamlandığını denetlemek için DISPLAY PUBSUB TYPE (PARENT | CHILD) komutunu kullanın.

Not: RESET QMGR komutu, yayınlama/abone olma sıradüzeninde ya da yayınlama/abone olma kümesinde yetkili sunucu aboneliklerinin iptaline neden olabilir. Bunu izlemek için DISPLAY SUB SUBTYPE (PROXYSUB) komutunu kullanın.

- Ayrıca, komutların tamamlandığı gibi, kuyruk derinliği sıfır olan diğer sistem kuyruklarını da izlemeniz gerekebilir.

Örneğin, SYSTEM.INTER.QMGR.CONTROL kuyruğunda ve SYSTEM.INTER.QMGR.FANREQ kuyruğunda izlemek isteyebilirsiniz. Bkz. Kümelerdeki yetkili abonelik trafiğini izleme ve Yayınlama/abone olma ağlarındaki tüketicileri ve tüketicileri dengeleme.

Sonraki adım

Bu denetimler, zamanuyumsuz bir komutun bittiğini onaylamazsa, bir hata oluşmuş olabilir. Araştırmak için, önce komutun yayınlandığı kuyruk yöneticisine ilişkin günlüğü denetleyin, sonra (bir küme için) küme tam havuz günlüklerini denetleyin.

İlgili bilgiler

 z/OS üzerinde CLUSTER komutlarının zamanuyumsuz davranışı

Uzak kuyruk tanımlamalarının diğer adlar olarak kullanılması

Başka bir kuyruk yöneticisinde kuyruk bulmanın yanı sıra, kuyruk yöneticisi diğer adları ve yanıtla kuyruğu diğer adları için uzak kuyruğun yerel tanımlamasını da kullanabilirsiniz. Her iki diğer ad tipi, uzak bir kuyruğun yerel tanımlamasıyla çözülür. İletinin hedefine ulaşması için uygun kanalları ayarlamanız gerekir.

Kuyruk yöneticisi diğer adları

Diğer ad, bir iletide belirtildiği gibi, hedef kuyruk yöneticisinin adının, ileti rotasındaki bir kuyruk yöneticisi tarafından 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.

Yanıtla 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 yanıtla kuyruğunun, ileti rotasındaki bir kuyruk yöneticisinde değiştirildiği işlemdir. Gönderme uygulaması, belirtilen yanıtla kuyruğu adının diğer ad olduğunun farkında değil.

Yanıt 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](#).

Yanıtla kuyruğu diğer adlarına ilişkin ek bilgi için [Yanıt kuyruğu diğer adları ve kümeler](#) başlıklı konuya bakın.

Veri dönüştürme

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 dönüştürülebilirler.

Örneğin, her ikisi de Batı Avrupa dilleri için geçerli olduğu için, CCSID (CCSID) 850 ve 500 olan kodlanmış karakter kümeleri arasında dönüştürme desteklenir.

ASCII ' ye EBCDIC newline (NL) karakter dönüştürmeleri için bkz. [Tüm kuyruk yöneticileri](#).

Desteklenen dönüştürmeler [Veri dönüştürme işlemi](#)' nde tanımlanır.

Kuyruk yöneticisi, iletileri yerleşik biçimlerde dönüştüremediğinde

CCSID 'leri farklı ulusal dil gruplarını gösteriyorsa, kuyruk yöneticisi iletileri yerleşik biçimlerde otomatik olarak dönüştüremez. Örneğin, CCSID 850 ile CCSID 1025 arasındaki dönüştürme (Kiril alfabesinde kullanan diller için EBCDIC kodlanmış karakter kümesi), kodlanmış karakter kümesindeki karakterlerin birçoğu diğerinde gösterilemiyorsa desteklenmez. Farklı ulusal dillerde çalışan bir kuyruk yöneticisi ağız varsa ve kodlanmış karakter kümelerinin bazıları arasında veri dönüştürmesi desteklenmiyorsa, varsayılan bir dönüştürmeyi etkinleştirebilirsiniz.

V 9.0.0 ccsid_part2.tbl 'in geçerli olduğu platformlar için ek bilgi için ccsid_part2.tbl 'i kullanarak “Varsayılan veri dönüştürmenin belirtilmesi” sayfa 201 'a bakın. ccsid_part2.tbl dosyasının geçerli olduğu platformlarda varsayılan veri dönüştürme işlemi, “Varsayılan veri dönüştürme” sayfa 199’ünde anlatılır.

IBM MQ 9.0’inde gelişmiş Unicode veri dönüştürme desteği

V 9.0.0

IBM MQ 9.0’öncesinde, ürünün önceki sürümleri, Basic Birden Çok Dilli Uçak (U+FFFF' un üzerindeki kod noktaları) dışındaki Unicode kod noktalarını içeren verilerin dönüştürülmesini 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.

IBM MQ 9.0' tan IBM MQ , veri dönüştürmede Unicode 8.0 standartında tanımlanmış olan tüm Unicode karakterlerini 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 için veri dönüştürme ve bazı platformlarda, bu CCSID' ler için tanımlanmış olan tüm kod noktalarını desteklemek için, Unicode tamamlayıcı uçaklarda kod noktalarıyla eşleştirilen tüm kod noktalarını destekleyin.

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

V 9.0.0

IBM MQ 9.0 , ek bir dosya (ccsid_part2.tbl) sağlıyor.

ccsid_part2.tbl dosyası, ccsid.tbl dosyasından önceliklidir ve:

- CCSID girişleri eklemenizi ya da deşiytirmenizi sağlar
- Varsayılan veri dönüştürmeyi belirtin
- Farklı komut düzeylerine ilişkin verileri belirtin

ccsid_part2.tbl yalnızca aşağıdaki altyapılar için geçerlidir:

- **Linux** Linux -tüm sürümler
- **Solaris** Solaris
- **Windows** Windows

Windows IBM MQ 9.0, IBM MQ for Windows, ccsid_part2.tbl , varsayılan olarak *MQDataRoot\conv\table* dizininde yer alır. Bununla birlikte, IBM MQ for Windows üzerinde, desteklenen tüm kod kümeleri de kaydedilir.

Solaris

Linux

IBM MQ 9.0, IBM MQ for Linux ve Solaris platformlarında, `ccsid_part2.tbl` , `MQDataRoot/conv/ta`bledizinde bulunur. Tüm Linux ve Solaris altyapıları için, desteklenen kod kümeleri IBM MQ tarafından 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 200.

ccsid.tbl dosya

V 9.0.0

`ccsid_part2.tbl` ' un geçerli olduğu kişiler dışındaki platformlarda, `ccsid.tbl` dosyası aşağıdaki amaçlarla kullanılır:

- HP-UX

AIX

 AIX ve HP-UX platformlarında, 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 kılavuz bilgiler 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

V 9.0.0

IBM MQ 9.0 ' den varsayılan veri dönüştürme yöntemi aşağıdaki altyapılarda değiştirildi:

- Linux -tüm sürümler
- Solaris
- Windows

See "[Varsayılan veri dönüştürmenin belirtilmesi](#)" sayfa 201 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.

V 9.0.0

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 nesnelerinde 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 diğerine 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.

V9.0.0 ccsid_part2.tbl dosyası

From IBM MQ 9.0, the `ccsid_part2.tbl` file replaces the existing `ccsid.tbl` file, used in previous versions of the product, to supply additional CCSID information. `ccsid.tbl` dosyasının IBM MQ tarafından ayrıştırılmaya devam ettiğini ve silinmemesi gerektiğini ve `ccsid_part2.tbl` içindeki girdilerin, `ccsid.tbl` içindeki diğer girdilere göre öncelik verdiğine dikkat edin.

`ccsid_part2.tbl`, şu konumdaki `MQDataRoot/conv/table` dizininde bulunur:

- Linux -tüm sürümler
- Solaris

ve Windows dizininde `MQDataRoot\conv\table` dizini

`ccsid_part2.tbl` dosyasının diğer platformlarda kullanılmadığını unutmayın. Dosya, aşağıdaki işlemleri gerçekleştirmek için kullanılır:

- IBM MQ veri dönüştürmesinde kullanmak üzere var olan CCSID girdilerini ekleme ya da değiştirme
- Varsayılan veri dönüştürmenin belirtilmesi

Bu dosya olarak `ccsid_part2.tbl` kullanmalısınız:

- Yeni Unicode kodlama değerleri için destek içerir.
- Girişlerin yalnızca seçilen komut düzeyleri için geçerli olması için, CCSID girişlerinin sürümünü belirtmenizi sağlar.

CCSID girişlerini ekleme ya da değiştirme

`ccsid_part2.tbl` dosyasındaki bir giriş aşağıdaki biçime sahiptir:

```
<CCSID number> <Base CCSID> <DBCS CodePage> <SBCS CodePage>  
<Type> <Encoding> <ACRI> <Name>
```

CCSID 1200 (UTF-16) için örnek bir giriş şöyledir:

```
1200 1200 1200 1200 3 8 0 UTF-16
```

Not: For more details on the value for ACRI see the comment in the `ccsid_part2.tbl` file.

`ccsid_part2.tbl` biçiminde:

Tip eşit olabilir:

- 1=SBCS
- 2=DBCS

3=MBCS

Kodlama 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:

- Bir satırın başlangıcındaki # simgesini kullanarak bir açıklama belirtebilir. Bu, IBM MQ ' un hattı ayrıştırmaya çalışmamasını önler.
- Satır içi yorumlar sağlanamıyor.
- Boş satırlar yaratmadığınızdan 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 ve EBCDIC CCSID 'ler arasında dönüştürme yapmak için kullanılan varsayılan dönüştürme CCSID' lerini tanımlayabilirsiniz.

Bu işlevi etkinleştirdiğinizde, varsayılan dönüştürme iletim ve ileti üstbilgileri için 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 'ler için varsayılan değeri 500 'e ayarlar ve ikinci satır, ASCII ve benzeri CCSID' ler için varsayılan değeri 850 'ye ayarlar.

Farklı Komut Düzeylerine İlişkin Verilerin Belirlenmesi

To specify CCSID entries for different command levels of IBM MQ you use a colon symbol followed by the command level (or command levels) of IBM MQ that you want the next section to be applicable to.

Bu sayı, kuyruk yöneticisinin ya da istemcinin çalışmakta olması gereken en düşük komut düzeyini gösterir. Örneğin, yürürlükteki kuyruk yöneticisi komuta düzeyi 900 ise ve bu değer 800 ya da 900 komut düzeyinde bir işaretlerle karşılaşırsa, CCSID ' ler okunurdu.

Ancak, 800 düzeyinde bir kuyruk yöneticisi, 900 kısımdaki CCSID ' leri yoksayar.

Belirtilen komut düzeyi, yeni bir komut düzeyi işareti bulununcaya kadar, bir komut düzeyi işaretiinden sonra saptanan tüm CCSID girişleri için geçerli olur.

Komut düzeyini tüm komut düzeylerine ayarlamanız gerekiyorsa, sıfır sayısını belirtin.

When first parsing ccsid_part2 .tbl, IBM MQ treats all CCSIDs encountered as valid for all command levels of IBM MQ.

Sürüm oluşturma yalnızca, IBM MQ ilk komut düzeyi işaretiyle karşılaştığında kullanılır.

Aşağıdaki kod parçacığı, sürüm oluşturma örneğini 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

```

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 can be completely administered at the command line using the IBM MQ [MQSC](#) commands.

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 bilgiler

[MQ Telemetry](#)

[MQXR özellikleri](#)

Linux

AIX

Linux ve AIX üzerinde telemetri için kuyruk yöneticisi

yapılandırılması

Follow these manual steps to configure a queue manager to run MQ Telemetry. You can run an automated procedure to set up a simpler configuration using the MQ Telemetry support for IBM MQ Explorer.

Başlamadan önce

1. IBM MQ ve MQ Telemetry özelliğinin nasıl kurulabilmesiyle ilgili bilgi için bkz. [MQ Telemetry kurulumu](#) .
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_unix.properties`. Normalde MQXR özellikleri dosyasını doğrudan düzenlemenize gerek yoktur; bunun nedeni, neredeyse tüm ayarların MQSC denetim komutları ya da IBM MQ Explorer yoluyla yapılandırılabilmesinden kaynaklanabilir. Dosyayı doğrudan düzenlemeye karar verdiyseniz, değişikliklerinizi yapmadan önce kuyruk yöneticisini durdurun. Bkz. [MQXR özellikleri](#).

Bu görev hakkında

IBM MQ Explorer için MQ Telemetry desteği bir sihirbaz ve örnek komut yordamı `sampleMQM` içerir. Konuk kullanıcı kimliğini kullanarak ilk yapılandırmayı kurdular; bkz. [MQ Telemetry kuruluşunun IBM MQ Explorer komutunu kullanarak doğrulanması](#) ve [IBM MQ Telemetry Transport örnek programları](#).

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.

```
echo "DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000)" | runmqsc qMgr
```

Telemetri (MQXR) hizmeti ilk kez başlatıldığında, `SYSTEM.MQTT.TRANSMIT.QUEUE` oluşturur.

Bu görevde el ile yaratılır; çünkü, telemetri (MQXR) hizmeti başlatılmadan önce, it'a erişim yetkisi vermek için bu hizmet `SYSTEM.MQTT.TRANSMIT.QUEUE` 'in var olması gerekir.

3. Varsayılan iletim kuyruğunu ayarla

When the telemetry (MQXR) service is first started, it does not alter the queue manager to make `SYSTEM.MQTT.TRANSMIT.QUEUE` the default transmission queue.

`SYSTEM.MQTT.TRANSMIT.QUEUE` 'u varsayılan iletim kuyruğu yapmak için varsayılan iletim kuyruğu özelliğini değiştirin. Alter the property using the IBM MQ Explorer or with the command in the following example:

```
echo "ALTER QMGR DEFEXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE')" | runmqsc qMgr
```

Varsayılan iletim kuyruğunu değiştirme, var olan yapılandırmanızı engelleyebilir. Varsayılan iletim kuyruğunu `SYSTEM.MQTT.TRANSMIT.QUEUE` 'e değiştirme nedeni, iletileri doğrudan MQTT istemcilerine göndermenin daha kolay hale gelmesidir. Varsayılan iletim kuyruğunu değiştirmeden, IBM MQ Explorer iletilerini alan her istemci için bir uzak kuyruk tanımı eklemelisiniz; bkz. [“İstemciye doğrudan ileti gönderme” sayfa 207](#).

4. Bir ya da daha çok kullanıcı kimliği yaratmak için [“MQTT istemcilerinin IBM MQ nesnelere erişmesi için yetki verme” sayfa 209](#) içindeki bir yordamı izleyin. Kullanıcı kimlikleri, yayınları yayınlama, abone olma ve MQTT istemcilerine gönderme yetkisine sahiptir.
5. Telemetri (MQXR) hizmetini kur

```
cat /opt/<install_dir>/mqxr/samples/installMQXRService_unix.mqsc | runmqsc qMgr
```

Ayrıca, [Şekil 20 sayfa 204](#) içindeki örnek kodla da bakın.

6. Hizmeti başlat

```
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, kimlikleri 4. adımda tanımlanan kullanıcı kimliklerinden biri olacak şekilde 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](#).

Örnek

Şekil 20 sayfa 204 shows the `runmqsc` command to create the `SYSTEM.MQXR.SERVICE` manually on Linux.

```
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')
```

Şekil 20. `installMQXRService_unix.mqsc`

Windows

Windows üzerinde telemetri kuyruk yöneticisi yapılandırılıyor

Follow these manual steps to configure a queue manager to run MQ Telemetry. You can run an automated procedure to set up a simpler configuration using the MQ Telemetry support for IBM MQ Explorer.

Başlamadan önce

1. IBM MQ ve MQ Telemetry özelliğinin nasıl kurulabilmesiyle ilgili bilgi için bkz. [MQ Telemetry kurulumu](#).
2. Kuyruk yöneticisi yarat ve başlat. Kuyruk yöneticisi bu görevde `qMgr` olarak anılır.
3. Bu görevin bir parçası olarak, telemetri (MQXR) hizmetini yapılandırılıyorsunuz. MQXR özellik ayarları, platforma özgü özellikler dosyasında depolanır: `mqxr_win.properties`. Normalde MQXR özellikleri dosyasını doğrudan düzenlemenize gerek yoktur; bunun nedeni, neredeyse tüm ayarların MQSC denetim komutları ya da IBM MQ Explorer yoluyla yapılandırılabilmesinden kaynaklanabilir. Dosyayı doğrudan düzenlemeye karar verdiyseniz, değişikliklerinizi yapmadan önce kuyruk yöneticisini durdurun. Bkz. [MQXR özellikleri](#).

Bu görev hakkında

IBM MQ Explorer için MQ Telemetry desteği bir sihirbaz ve örnek komut yordamı `sampleMQMiçerir`. Konuk kullanıcı kimliğini kullanarak ilk yapılandırmayı kurdular; bkz. [MQ Telemetry kuruluşunun IBM MQ Explorer komutunu kullanarak doğrulanması](#) ve [IBM MQ Telemetry Transport örnek programları](#).

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.

```
echo DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000) | runmqsc qMgr
```

Telemetri (MQXR) hizmeti ilk kez başlatıldığında, `SYSTEM.MQTT.TRANSMIT.QUEUE` oluşturulur.

Bu görevde el ile yaratılır; çünkü, telemetri (MQXR) hizmeti başlatılmadan önce, it'a erişim yetkisi vermek için bu hizmet SYSTEM.MQTT.TRANSMIT.QUEUE ' in var olması gerekir.

3. qMgr için varsayılan iletim kuyruğunu ayarla

```
echo ALTER QMGR DEFQMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```

Şekil 21. Varsayılan iletim kuyruğunu ayarla

When the telemetry (MQXR) service is first started, it does not alter the queue manager to make SYSTEM.MQTT.TRANSMIT.QUEUE the default transmission queue.

SYSTEM.MQTT.TRANSMIT.QUEUE ' u varsayılan iletim kuyruğu yapmak için varsayılan iletim kuyruğu özelliğini değiştirin. Alter the property using the IBM MQ Explorer or with the command in [Şekil 21 sayfa 205](#).

Varsayılan iletim kuyruğunu değiştirme, var olan yapılandırmanızı engelleyebilir. Varsayılan iletim kuyruğunu SYSTEM.MQTT.TRANSMIT.QUEUE ' e değiştirme nedeni, iletileri doğrudan MQTT istemcilerine göndermenin daha kolay hale gelmesidir. Varsayılan iletim kuyruğunu değiştirmeden, IBM MQ iletilerini alan her istemci için bir uzak kuyruk tanımı eklemelisiniz; bkz. "[İstemciye doğrudan ileti gönderme](#)" sayfa 207.

4. Bir ya da daha çok kullanıcı kimliği yaratmak için "MQTT istemcilerinin IBM MQ nesnelere erişmesi için yetki verme" sayfa 209 içindeki bir yordamı izleyin. Kullanıcı kimlikleri, yayınları yayınlama, abone olma ve MQTT istemcilerine gönderme yetkisine sahiptir.

5. Telemetri (MQXR) hizmetini kur

```
type  
installMQXRService_win.mqsc | runmqsc qMgr
```

6. Hizmeti başlat

```
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, kimlikleri 4. adımda tanımlanan kullanıcı kimliklerinden biri olacak şekilde 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](#).

Creating SYSTEM.MQXR.SERVICE manually

[Şekil 22 sayfa 206](#) shows the **runmqsc** command to create the SYSTEM.MQXR.SERVICE manually on Windows.

```

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+\."') +
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')

```

Şekil 22. installMQXRService_win.mqsc

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 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 14 sayfa 207.

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 yoksayılır.

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 degerine ayarlanmalıdır.

Çizelge 14. Bir MQTT kuyruk yöneticisi diğer adının çözülmesi					
	Giriş		Çıkış		
<i>ClientIdentifier</i>	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. SYSTEM.MQTT.TRANSMIT.QUEUE
<i>ClientIdentifier</i> adlı bir kuyruk yöneticisi diğer adıyla eşleşiyor	<i>ClientIdentifier</i>	<i>tanımsız</i>	<i>ClientIdentifier</i>	<i>tanımsız</i>	SYSTEM.MQTT.TRANSMIT.QUEUE

Ad çözümlemesiyle 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 23. 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 hiyerarşisine bağlı 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 24. 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 25. 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 'a çözümlenmelidir; bkz. Çizelge 15 sayfa 208. Uzak kuyruk adı herhangi bir değer olabilir. İstemci bunu bir konu dizgisi olarak alır.

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	ClientIdentifier	SYSTEM.MQTT.TRANSMIT.QUEUE

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 ClientIdentifierya 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. Aygıtı güvenilen, zaman sınırlandırılmış 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 atıfta bulunulan 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 tarafından sağlanan bir kanal kimliğine MQTTClient yetki 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 için 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 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. The administrator configures the telemetry channel to select the client `Kullanıcı Adı` as its identity.

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 210](#)
- [“Kaba erişim denetimi” sayfa 211](#)
- [“Orta-parçalı erişim denetimi” sayfa 211](#)
- [“Parçalı erişim denetimi” sayfa 211](#)

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. [Windowsüzerinde bir gruba kullanıcı ekleme](#) ya da [Linuxüzerinde bir gruba kullanıcı ekleme](#)

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 yayınladıysa 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.

clientyöneticisi, 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, MQTTClient'e herhangi bir konuyu yayınlayıp abone olmak için yetki verir:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

3. MQTT istemci uygulaması geliştiricisi, 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

Connections, her zaman MQTT istemcisi tarafından başlatılır. MQTT istemcisi her zaman SSL istemcisidir. MQTT istemcisine ilişkin sunucu ve sunucu kimlik doğrulamasının istemci kimlik doğrulaması her ikisi de isteğe bağlıdır.

İstemciyi özel imzalı bir dijital sertifikalıyla sağlayarak, MQTT istemcisinin kimliğini WebSphere MQ olarak doğrulayabilirsiniz. WebSphere MQ Denetimcisi, MQTT istemcilerini TLS kullanarak kuyruk yöneticisine kendilerini doğrulamaya zorlayabilir. İstemci kimlik denetimini yalnızca karşılıklı kimlik doğrulamanın bir parçası olarak isteyebilirsiniz.

SSL'yi kullanmaya 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. MQTT istemcilerini, VPN ağı üzerinden TCP/IP kullanarak telemetri kanallarına bağlayabilirsiniz.

TLS ' nin kullanıldığı istemci kimlik doğrulaması, istemcinin gizli bir gizeye sahip olmasını sağlar. Gizli, kendinden onaylı sertifika ya da bir sertifika yetkilisi tarafından sağlanan bir anahtarda istemcinin özel anahtardır. Anahtar, istemcinin dijital sertifikasını imzalamak için kullanılır. İlgili genel anahtara sahip olan herkes, sayısal sertifikayı doğrulayabilir. Sertifikalar güvenilir, ya da zincirlenmişse, sertifika zincirinden güvenilir bir kök sertifikadan geriye doğru izlenebilir. İstemci doğrulaması, istemci tarafından sunucuya sağlanan sertifika zincirindeki tüm sertifikaları sunucuya gönderir. Sunucu, sertifika zincirini güvendiği bir sertifika buluncaya kadar denetler. Güvenilir sertifika, kendinden onaylı bir sertifikadan oluşturulan genel sertifikadır ya da genellikle bir sertifika yetkilisi tarafından verilen bir kök sertifikadır. Son olarak, isteğe bağlı olarak, güvenilir sertifikayı "canlı" bir sertifika iptal listesiyle karşılaştırılabilir.

Güvenilen sertifika, bir sertifika yetkilisi tarafından yayınlanabilir ve önceden JRE sertifika deposuna eklenmiş olabilir. Kendinden onaylı bir sertifika ya da telemetri kanalı anahtar deposuna güvenilen bir sertifika olarak eklenmiş olan herhangi bir sertifika olabilir.

Not: Telemetri kanalı, bir ya da daha fazla telemetri kanalına özel anahtarları ve istemcileri doğrulamak için gerekli tüm genel sertifikalara sahip bir birleşik anahtar deposu/güvenilirlik deposu içerir. Bir SSL kanalının bir anahtar deposu olması ve kanal truststore ile aynı dosyaya sahip olması gerektiğinden, JRE sertifika deposuna hiçbir zaman gönderme yapılmamaktadır. Bunun anlamı, bir istemcinin kimlik doğrulaması bir CA kök sertifikası gerektiriyorsa, CA kök sertifikası zaten JRE sertifika deposunda olsa bile, kanal için kök sertifikayı, kanal için anahtar deposuna yerleştirmeniz gerekir. JRE sertifika deposuna hiçbir zaman gönderme yapılmamaktadır.

İstemci kimlik doğrulamasının sayacağı tehditleri ve müşterinin ve sunucunun tehditlere karşı koymak için rol oynadığı tehditleri düşünün. Tek başına istemci sertifikasının doğrulanması, bir sisteme yetkisiz erişimi önlemek için yeterli değildir. İstemci aygıtı başka biri tarafından tutulduysa, istemci aygıtının sertifika sahibinin yetkisiyle hareket etmesi gerekmez. İstenmeyen saldırılara karşı tek bir savunmaya asla güvenmeyin. En azından iki faktörlü kimlik doğrulama yaklaşımı ve özel bilgiler bilgisine sahip bir sertifikana ek sahip olma. Örneğin, JAAS' ı kullanın ve sunucu tarafından yayınlanan bir parola kullanarak istemcinin kimliğini doğrulayın.

Müşteri sertifikasına ilişkin birincil tehdit, yanlış ellere geçmektedir. Sertifika, istemcide bir parola korumalı anahtar deposunda tutulur. Anahtar depoya nasıl yerleştirilsin? MQTT istemcisi parolayı nasıl anahtar deposuna alır? Parola korumasının güvenliği ne kadar güvenli? Telemetri cihazları çoğu zaman kolayca çıkarılabilir ve daha sonra özel olarak hacklenebilirler. Aygıt donanımını kurcalaması gerekiyor mu? İstemci tarafı sertifikalarının dağıtılması ve korunması zor olduğu tanınmaktadır; bu, anahtar yönetimi sorunu olarak adlandırılır.

İkincil bir tehdit, aygıtın istenmeyen yollarla sunuculara erişmek için yanlış kullanılsa da olabilir. Örneğin, MQTT uygulaması kurcalandıysa, kimliği doğrulanmış istemci kimliğini kullanarak sunucu yapılandırmasındaki bir zayıflığı kullanmak mümkün olabilir.

SSL kullanan bir MQTT istemcisini doğrulamak için telemetri kanalını ve istemciyi yapılandırın.

İlgili kavramlar

“[TLS ' yi kullanarak MQTT istemci kimlik doğrulaması için telemetri kanalı yapılandırması](#)” sayfa 213 IBM MQ yöneticisi, sunucudaki telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasındaki bir TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar kütüklere geçiş tümceciyle korunan erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcecik ya da anahtar dosyası olmadan tanımlandıysa, kanal TLS bağlantılarını kabul etmez.

İlgili bilgiler

[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, sunucudaki telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasındaki bir TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar kütüklere geçiş tümceciyle korunan erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcecik ya da anahtar dosyası olmadan tanımlandıysa, 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 215.](#)

İlgili kavramlar

[“TLS kullanılarak kanal kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 215](#)

IBM MQ yöneticisi, sunucudaki telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasındaki bir TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar kütüklere geçiş tümcecisiyle korunan erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcecik ya da anahtar dosyası olmadan tanımlandıysa, kanal TLS bağlantılarını kabul etmez.

İlgili bilgiler

[KANAL TANIMLA \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

[CipherSpecs ve CipherSuites](#)

Linux Windows AIX TLS ' yi kullanarak telemetri kanalı kimlik doğrulaması

Connections, her zaman MQTT istemcisi tarafından başlatılır. MQTT istemcisi her zaman SSL istemcisidir. MQTT istemcisine ilişkin sunucu ve sunucu kimlik doğrulamasının istemci kimlik doğrulaması her ikisi de isteğe bağlıdır.

İstemci, anonim bağlantıyı destekleyen bir CipherSpec kullanmak üzere yapılandırılmadıkça, istemci her zaman sunucuyu doğrulamayı dener. Kimlik doğrulama başarısız olursa, bağlantı kurulmaz.

SSL ' yi kullanmaya 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. MQTT istemcilerini, VPN ağı üzerinden TCP/IP kullanarak telemetri kanallarına bağlayabilirsiniz.

Server kimlik doğrulaması, gizli bilgi göndermek üzere olduğunuz sunucuyu kimlik doğrulamasını gerçekleştirir. İstemci, sunucudan gönderilen sertifikalarla, güvenli depoya yerleştirilen sertifikalara ya da JRE cacerts deposunda eşleşen denetimleri gerçekleştirir.

JRE sertifika deposu bir JKS dosyasıdır, cacerts. It is located in JRE InstallPath\lib\security\. It is installed with the default password changeit. JRE sertifika deposunda ya da istemci güvenilirlik deposunda güvendiğiniz sertifikaları saklayabilirsiniz. Her iki mağaza da kullanamazsınız. Genel sertifikaların istemcilerin diğer Java uygulamalarının kullandığı sertifikalardan ayrı olarak güvendiği sertifikalarını alıkoymak 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 verirsiniz, içerdiği sertifikalar onlara güvendiğinizden emin olmak için bu sertifikayı gözden geçirin.

Farklı bir güven sağlayıcı belirterek JSSE yapılandırmasını değiştirebilirsiniz. Bir sertifika üzerinde farklı denetimler gerçekleştirmek için bir güven sağlayıcısını özelleştirebilirsiniz. In some OGSi environments that have used the MQTT client, the environment provides a different trust provider.

Telemetri kanalının TLS ' yi kullanarak kimlik doğrulamasını yapmak için sunucuyu ve istemciyi yapılandırın.

Linux Windows AIX TLS kullanılarak kanal kimlik doğrulaması için telemetri kanalı yapılandırması

IBM MQ yöneticisi, sunucudaki telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasındaki bir TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar kütüklere geçiş tümceciyle korunan erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcecik ya da anahtar dosyası olmadan tanımlandıysa, 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 tümceciğini belirtin. Kanalın TCP/IP kapı numarasını ayarlamadıysanız, TLS telemetri kanalı kapı numarası varsayılan değer olarak 8883 değerini alır.

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 215.](#)

İlgili kavramlar

[“TLS ' yi kullanarak MQTT istemci kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 213](#)

IBM MQ yöneticisi, sunucudaki telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasındaki bir TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar kütüklere geçiş tümceciyle korunan erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcecik ya da anahtar dosyası olmadan tanımlandıysa, kanal TLS bağlantılarını kabul etmez.

İlgili bilgiler

[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

“TLS kullanılarak kanal kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 215

IBM MQ yöneticisi, sunucudaki telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasındaki bir TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. TLS kanalları, anahtar kütüklere geçiş tümcecisiyle korunan erişim ile yapılandırılır. Bir TLS kanalı geçiş tümcecik ya da anahtar dosyası olmadan tanımlandıysa, kanal TLS bağlantılarını kabul etmez.

İlgili bilgiler

[Telemetri \(MQXR\) hizmeti](#)

[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 'yi kullanmaya 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 kuruyorsunuz. MQTT istemcilerini, VPN ağı üzerinden TCP/IP kullanarak telemetri kanallarına bağlayabilirsiniz.

Kanalı şifreleyen ve sunucuyu doğrulayan tipik bir yapılandırma için [“TLS 'yi kullanarak telemetri kanalı kimlik doğrulaması” sayfa 214' 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 tokası 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 tokasız 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 tümcecigi 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 koymaktır. 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.

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 'yi kullanmaya 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 kuruyorsunuz. MQTT istemcilerini, VPN ağı üzerinden TCP/IP kullanarak 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 212](#)
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 214.](#)
3. İstemciyle kuyruk yöneticisi arasındaki iletileri şifrele; bkz. [“telemetri kanallarında yayın gizliliği” sayfa 216.](#)

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 değiştirgelerini istemci ve sunucu üzerindeki JRE güvenlik dizinindeki dosyaları düzenleyerek değiştirin.

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 isteğe bağlı bir parametredir. Bu, TCP kanalları için 1883 , TLS kanalları için de 8883 değerine ayarlanır.

Bu kanalla ilişkilendirilen TCP/IP kapı numarası. MQTT istemcileri, kanal için tanımlanan kapıyı belirterek bir kanala bağlanır. Kanalda TLS özellikleri varsa, istemci TLS iletişim kuralını kullanarak bağlanmalıdır; örneğin:

```
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

Diğer dosya uzantılarına sahip bir anahtar deposunun JKS anahtar deposu olduğu varsayılır.

Sunucudaki bir anahtar deposu tipini, istemciye diğer anahtar deposu tipleriyle birleştirebilirsiniz.

Sunucunun özel sertifikasını anahtar depoya yerleştirin. 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 yerleştirin.

Sunucu sertifikası ve sertifika zincirindeki tüm sertifikalar, sunucunun kimliğini doğrulamak üzere istemcilere gönderilir.

ClientAuth ögesini Requiredolarak ayarladıysanız, anahtar deposunun istemcinin kimliğini doğrulamak 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 doğrulamaları ile doğrular. Bir sertifika zinciri kullanarak, bir sertifika, farklı istemci sertifikalarıyla yayınlansa bile, birçok istemciyi doğrulayabilir.

PassPhrase

PassPhrase , TLS kanalları için gerekli bir parametredir. TCP kanalları için atlanmalıdır.

Anahtar deposunu korumak için geçiş 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' i 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.

JSSE ' 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 SSLFIPS ayarlayın. SSLFIPS , yalnızca FIPS onaylı algoritmaların kullanılıp kullanılmayacağını belirtir.

İptal ad listesi

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 nesnelere 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 16. JRE TLS yapılandırma dosyaları için altyapıya göre dosya yolları	
Altyapı	filePath
Windows	<i>WMQ Installation Directory</i> \java\jre\lib\security
UNIX 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 `strmqikm`.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 `strmqikm` command, or download strong, but limited jurisdiction files from [IBM geliştirici setleri](#), [Güvenlik bilgileri](#).

Önemli: Ülkenizde, şifreleme yazılımların başka bir ülkeye alınması, bulundurmak, 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 26 sayfa 221 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 26. 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 27. 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 28. 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 29. 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. [Şekil 30 sayfa 222](#).

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 Kullanıcı1 adı ve Parola ' yı döndüren iki geri arama sağlar. The Kullanıcı1 Adı and Parola are set in the MqttConnectOptions object. Kullanıcı1 Adı ve Parolaerişimine nasıl erişileceğini gösteren bir örnek için bkz. [Şekil 31 sayfa 223](#) .

Örnekler

Adıconfigurationolan bir JAAS yapılandırma dosyası örneği: MQXRConfig.

```
MQXRConfig {
samples.JAASLoginModule required debug=true;
//com.ibm.security.auth.module.NTLoginModule required;
//com.ibm.security.auth.module.Krb5LoginModule required
//      principal=principal@your_realm
//      useDefaultCcache=TRUE
//      renewTGT=true;
//com.sun.security.auth.module.NTLoginModule required;
//com.sun.security.auth.module.UnixLoginModule required;
//com.sun.security.auth.module.Krb5LoginModule required
//      useTicketCache="true"
//      ticketCache="${user.home}/${}/tickets";
};
```

Şekil 30. Örnek jaas.config dosyası

Bir MQTT istemcisi tarafından sağlanan Kullanıcı1 adı ve Parola ' yı alacak şekilde kodlanmış bir JAAS Oturum 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;
}

```

Şekil 31. Örnek JAASLoginModule.Login() yöntemi

İlgili bilgiler

AuthCallback MQXR sınıfı

Sorun çözülüyor: JAAS oturum açma modülü telemetri hizmeti tarafından çağırılmadı

V 9.0.0 Yönetme IBM MQ Light

MQ Light programını IBM MQ Explorer komutunu kullanarak ya da bir komut satırında yönetebilirsiniz. Kanalları yapılandırmak ve IBM MQ' a bağlı MQ Light istemcilerini izlemek için Gezgin 'i kullanın. TLS ve JAASgüvenliğini kullanarak MQ Light güvenliğini yapılandırın.

Başlamadan Önce

Platformunuza AMQP ' nin kurulmasıyla ilgili bilgi için [Kurulacak olanı seçme](#) başlıklı konuya bakın. Install the AMQP Service component by using the IBM MQ 8.0.0.4 manufacturing refresh, not the 8.0.0.4 Fix Pack. You cannot install the AMQP component on a version of the queue manager earlier than 8.0.0.4.

IBM MQ Explorer komutunu kullanarak yönetme

AMQP kanallarını yapılandırmak ve IBM MQ' a bağlı MQ Light istemcilerini izlemek için Gezgin 'i kullanın. You can configure the security of MQ Light using TLS and JAAS.

Komut satırını kullanarak yönetme

You can administer MQ Light at the command line using the IBM MQ [MQSC](#) commands.

V 9.0.0 Viewing IBM MQ objects in use by MQ Light clients

You can view the different IBM MQ resources in use by MQ Light clients, for example connections and subscriptions.

Bağlantılar

AMQP hizmeti başlatıldığında yeni Hconns oluşturulur ve kuyruk yöneticisine bağlanır. Bu Hconns havuzu, MQ Light istemcileri ileti yayınladığında kullanılır. Hconns uygulamasını **DISPLAY CONN** komutunu kullanarak görüntüleyebilirsiniz. Örneğin:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (APPLDESC LK 'WebSphere MQ Advanced Message Queuing Protocol*')
```

Bu komut, istemciye özgü Hconns programlarını da gösterir. Boş bir istemci tanıtıcısı özniteliğine sahip olan Hconns (Hconns), havuzda kullanılan Hconns

Bir MQ Light istemcisi bir AMQP kanalına bağlandığında, kuyruk yöneticisine yeni bir Hconn bağlantısı kurulur. Bu Hconn, MQ Light istemcisinin yarattığı abonelikler için zamanuyumsuz olarak iletileri tüketmek için kullanılır. **DISPLAY CONN** komutunu kullanarak belirli bir MQ Light istemcisi tarafından kullanılan Hconn 'u görüntüleyebilirsiniz. Örneğin:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_abcd1234')
```

İstemciler tarafından yaratılan abonelikler

Bir MQ Light 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 ürünü
- Bir örnek. Bu örnek, istemci paylaşılmayan bir abonelik oluşturduysa private ya da istemci paylaşılan bir aboneliğe katılmışsa share olur.

Belirli bir MQ Light istemcisi tarafından kullanılanı abonelikleri görüntülemek için, **DISPLAY SUB** komutunu çalıştırın ve private önekinde süzgeç uygulayın:

```
DISPLAY SUB(:private:*)
```

Birden çok istemci tarafından kullanımda olan paylaşılan abonelikleri görüntülemek için **DISPLAY SUB** komutunu çalıştırın ve share önekinde süzgeç uygulayın:

```
DISPLAY SUB(:share:*)
```

Paylaşılan abonelikler birden çok MQ Light istemcisi tarafından kullanılabilmesi için, paylaşılan abonelikten iletileri tüketen istemcilerin görüntülenmesini isteyebilirsiniz. Bu işlemi, abonelik kuyruğunda bir tanıtıcısı açık olan Hconns ' leri listeleterek 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(414D5120514D3120202020202020202020202020202020202020202020202020)
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(414D5143514D312020202020202020)
TYPE(HANDLE)

OBJNAME(SYSTEM.BASE.TOPIC)      OBJTYPE(TOPIC)

OBJNAME(SYSTEM.MANAGED.DURABLE.560A7E7020002961)
OBJTYPE(QUEUE)
```

3. Tutamaçların her biri için, tutamaç açık olan MQ Light istemci tanıtıcısını görüntüleyin:

```
DISPLAY CONN(707E0A56642B0020) CLIENTID
23 : DISPLAY CONN(707E0A56642B0020) CLIENTID

AMQ8276: Display Connection details.
CONN(707E0A56642B0020)
EXTCONN(414D5143514D312020202020202020)
TYPE(CONN)
CLIENTID(recv_8f02c9d)
DISPLAY CONN(707E0A565F290020) CLIENTID
24 : DISPLAY CONN(707E0A565F290020) CLIENTID
AMQ8276: Display Connection details.
CONN(707E0A565F290020)
EXTCONN(414D5143514D312020202020202020)
TYPE(CONN)
CLIENTID(recv_86d8888)
```

V 9.0.0

MQ Light istemci tanıtıcısı, yetkilendirme 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 çalışıldığında, 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 yapılandırması 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 belirlemezse, 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.

V 9.0.0 MQ Light istemci kimliği ve yetkilendirmesi

Use the MQ Light client ID, the MQ Light user name, or a common client identity defined on the channel or in a channel authentication rule, for authorization to access IBM MQ objects.

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 bulunan 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
```

İstemci parolasını kullanarak MQ Light 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 ADOPTTCTX (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.

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 kuruyorsunuz. AMBP 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 değiş tokası 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 tokasız olarak kurulur.

Anonim bağlantılar güvensiz olduğu için, çoğu TLS somutlaması anonim CipherSpecs özelliğini kullanmak için varsayılan değer 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 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 koymanız. Güvenilir sağlayıcınız sunucu sertifikasının kimliğini doğrulamazdı, ancak bağlantı şifrelenir.

V 9.0.0 Configuring MQ Light clients with TLS

Ağ üzerinde akan verileri korumak ve istemcinin bağlandığı kuyruk yöneticisinin kimliğini doğrulamak için MQ Light istemcilerini TLS ' yi kullanacak şekilde yapılandırabilirsiniz.

Bir MQ Light istemcisinden AMQP kanalına bağlantı için TLS 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 sunması için bir MQ Light 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 MQ Light istemcisi, TLS tarafından korunan bir AMQP kanalına bağlandığında, istemci kuyruk yöneticisi tarafından sunulan sertifikana ilişkin kimliği doğrular. Bunu yapmak için, MQ Light istemcinizi kuyruk yöneticisinin sertifikasını içeren bir güvenli depo ile yapılandırmalısınız. Bunu yapma adımları, kullanmakta olduğunuz MQ Light istemcisine bağlı olarak değişir.

- Node JS API belgeleri için MQ Light istemcisi için bkz. <https://www.npmjs.com/package/mqlight>
- Java API belgeleri için MQ Light istemcisi için bkz. <https://mqlight.github.io/java-mqlight/>
- Ruby belgeleri için MQ Light istemcisi için bkz. <https://www.rubydoc.info/github/mqlight/ruby-mqlight/>
- MQ Light Client for Python belgeleri için bkz. <https://python-mqlight.readthedocs.org/en/latest/>

V 9.0.0 MQ Light istemcilerinin kuyruk yöneticisinden çıkarılması

MQ Light bağlantısını kuyruk yöneticisinden çıkarmak istiyorsanız, PURGE CHANNEL komutunu çalıştırın ya da MQ Light istemcisine bağlantıyı durdurun.

- **PURGE CHANNEL** komutunu çalıştırın. Örneğin:

```
PURGE CHANNEL(MYAMQP) CLIENTID('recv_28dbb7e')
```

- Alternatively, stop the connection that the MQ Light client is using to disconnect the client by completing the following steps:

1. Find the connection that the client is using by running the **DISPLAY CONN** command. Örneğin:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_28dbb7e')
```

Komut çıkışı aşağıdaki gibidir:

```
DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
40 : DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
AMQ8276: Display Connection details.
CONN(707E0A565F2D0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE (CONN)
CLIENTID(recv_28dbb7e)
```

2. Bağlantıyı durdurun. Örneğin:

```
STOP CONN(707E0A565F2D0020)
```

Çoklu 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 nesneleriyle 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 değiştiricileriyle 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 iletilerinde yayınlanan bilgilerin konusu ve bir konu TOPIC nesnesi yaratılarak tanımlanmıştır. KONU NESNELERİNİN, çoklu yayınlama kullanılarak kullanılmayacağını tanımlayan iki parametre vardır. Bu değiştiriciler şunlardır: **COMMINFO** ve **MCAST**.

- **COMMINFO** Bu parametre, çok hedefli iletişim bilgileri nesnesinin adını belirtir. COMMINFO nesne değiştiricileriyle 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 nesnelere yaratıldıktan sonra, bunlar amqsubc örneği ve amqssubc örneği kullanılarak sınavılabilir. 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 230adımında 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.

classful network IP adresleme şeması, çok hedefli adres için belirlenmiş adres alanına sahiptir. Tam çoklu yayın aralığı IP adresi 224.0.0.0 - 239.255.255.255 olur, ancak bu adreslerin bazıları ayrılmıştır. Ayrılmış adreslerin listesi için sistem denetimcinize başvurun ya da daha fazla bilgi için <https://www.iana.org/assignments/multicast-addresses> ' e bakın. Yerel olarak kapsamlı çok hedefli adresi 239.0.0.0 - 239.255.255.255 aralığında kullanmanız önerilir.

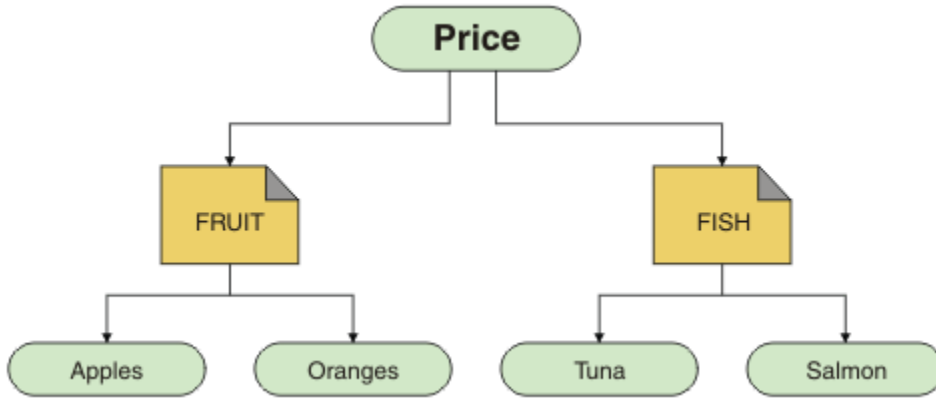
Aşağıdaki şemada, olası iki çok noktaya gönderim veri akışı vardır:

```
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 daha sonraki [Çizelge 17 sayfa 232'](#) te anlatılır.

COMMINFO nesnesi, MQMD alanlarının ve kullanıcı özelliklerinin birçoğunun iletiyle 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 iletiyle 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.

Çok hedefli ileti öznitelikleri

İleti öznitelikleri, MQMD gibi çeşitli yerlerden (örneğin, MQRFH2) ve ileti özelliklerindeki alanlara gelebilir.

Aşağıdaki tabloda, iletiler MCPROP değerine ve bir öznitelik gönderilmediğinde kullanılan varsayılan değere tabi olarak gönderildiğinde ne olacağını gösterir.

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
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	Dışlandı	Burada geçerli değil
MsgSeqNumarası	Dışlandı	Burada geçerli değil

Çizelge 17. İleti alışverişi öznelikleri ve bunların çok hedefli olarak nasıl ilişkilendirildikleri (devamı var)		
Öznelik	Çoklu yayın kullanan işlem	Varsayılan değer iletilmezse
Görelî Konum	Dışlandı	Burada geçerli değil
MsgFlags	Dışlandı	Burada geçerli değil
OriginalLength	Dışlandı	Burada geçerli değil
UserProperties	Dahil	Burada geçerli değil

İlgili başvurular

[ALTER KOMUT](#)

İlgili bilgiler

[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 özneliği, 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.

Yayınlayıcılarınızın ve abonelerinizin 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** özneliğiyle 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 eşiğinin 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 geçişinin nasıl ayarlanacağını öğrenmek için bu bilgileri kullanın.

Çok hedefli olarak gelen iletim hatasının önemli bir öğesi, iletilen verilerin arabelleğe alınmasını (bağlantının kesilen ucunda tutulacak iletilerin geçişi) 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 geçişi olduğu anlamına gelir, ancak çoklu yayının niteliği nedeniyle %100 garantili teslim desteklenemez.

IBM MQ Multicast ileti geçişi, iletişim bilgileri (COMMINFO) nesnesinde **MSGHIST** özneliğe göre denetlenir:

MSGHIST

Bu değer, NACK ' lar (negatif onaylar) durumunda, yeniden iletimleri işlemek için sistem tarafından tutulan, kilobayt cinsinden ileti geçişi 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 geçişi, bir yayın akışına katılan bir abonenin şu anda kullanılabilir olduğu kadar çok veri alıp almadığını ya da yalnızca abonelik zamanından yapılan yayınları almasını denetler.

YOK

NONE değeri, vericinin yalnızca abonelik zamanından yapılan yayını iletmemesine 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 bilgiler

[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** özniteliği, çok hedefli uygulamalar kullanmayan uygulamalardan yayınlar denetler. **BRIDGE** ENABETLE olarak ayarlanırsa ve konunun **MCAST** parametresi de ETKİN olarak ayarlandıysa, çoklu yayın kullanmayan uygulamalardan gelen yayınlar, yapılan uygulamalara köprü olarak ayarlanır. **BRIDGE** parametresine ilişkin ek bilgi için [DEFINE COMBILGİS](#) 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 trafiğinin bir yöneltici üzerinden geçebileceğini ya da geçebileceği yönelticilerin 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 şu konuya bakın: [Multicast uygulamalarının çoklu yayın olmayan bir ağ üzerinde test edilmesi](#) ve [Çoklu yayın 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 görenek geri bildirim.

HeartbeatTimeout

Sağlıklı işletim bildirim zamanı (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 atımı gönderilmemesinin gönderildiğini gösterir. Sağlıklı işletim bildirim zamanı 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 yayınla 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 özelliği	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
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 LLM ürün belgelerine bakın: [IBM MQ Low Latency Messaging\(Düşük Gecikme İleti Sistemi\)](#).

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 238](#)
- [“IBM MQ for IBM i yönetimine ilişkin diğer yöntemler” sayfa 252](#)
- [“Work management for IBM i” sayfa 256](#)

İlgili kavramlar

[“Availability, backup, recovery, and restart on IBM i” sayfa 264](#)

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.

İlgili başvurular

[“IBM MQ for IBM i susturulması” sayfa 308](#)

Bu bölümde, 'quiesce' un nasıl susturacağını açıklar. IBM MQ for IBM i.

İlgili bilgiler

[IBM üzerindeki yapılandırma bilgilerinin değiştirilmesi](#)

[IBM MQ for IBM i kuyruk yöneticisi kitaplık adlarının anlaşılması](#)

[IBM üzerinde güvenliğin ayarlanması](#)

[IBM üzerindeki ölüm harf kuyruğu işleyicisi](#)

[Determining problems with IBM MQ for IBM i applications](#)

[IBM üzerindeki kurulabilir hizmetler ve bileşenler](#)

[System and default objects on 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 nesnelere 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 bir 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 öğe için kullanılabilir:

- Kurtarma amacı
- Bakım
- IBM MQ uygulamalarını kaydetme

IBM MQ uygulama tanımlarını 1/2 yöntemden 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 nesnelere ilişkin daha fazla ayrıntı için bkz. [“Komut dosyası \(MQSC\) komutları” sayfa 10](#) ve [“IBM MQ Programları Komut Biçimlerinin Kullanılması” sayfa 21](#).

İlgili bilgiler

[IBM MQ for IBM i CL komutları başvurusu](#)

IBM i

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 geçerli olduğu 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 i 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 i susturulması” sayfa 308](#) içinde açıklanmaktadır.

IBM i

IBM MQ for IBM i nesnelere oluşturma

Use this information to understand the methods for creating IBM MQ objects for IBM i.

Başlamadan önce

Aşağıdaki görevler, komut satırından IBM MQ for IBM i ' u kullanabilmeniz için çeşitli yollar önermektedir.

Bu görev hakkında

IBM MQ nesnelere 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 CDMMQM 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, F4tuşuna 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 240](#) . 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 kuyruğulardan 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 242.

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 MQ for IBM i ölü-harfli kuyruk işleyicisi](#).

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)  
MAXMSGLEN(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 **CHGMQMQ** command or the **CPYMQMQ** command with the REPLACE *YES attribute. In "[Yerel kuyruk tanımlanması](#)" sayfa 241, you defined the queue `orange.local.queue`. Örneğin, bu kuyruktaki ileti uzunluğu üst sınırını 10.000 byte'a artırmamız gerekiyorsa.

- **CHGMQMQ** komutunu kullanarak:

```
CHGMQMQ QNAME('orange.local.queue') MQMNAME(MYQUEUEMANAGER) MAXMSGLEN(10000)
```

Bu komut, ileti uzunluğu üst sınırı olan tek bir özniteliği değiştirir; diğer tüm öznitelikler de aynı kalır.

- Using the **CRTMQMQ** command with the REPLACE *YES option, for example:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL) MQMNAME(MYQUEUEMANAGER)  
MAXMSGLEN(10000) REPLACE(*YES)
```

Bu komut yalnızca ileti uzunluğu üst sınırı değil, diğer tüm özniteliklerin 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 belgeleri, [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 özniteliği 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 koyması 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(MYQUEUEEMANAGER)
```

Bu komut, `my.alias.queue` ögesinispecifykuyruğuna (`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:

```
CHGMQMQ QNAME('my.alias.queue') TGTQNAME('magenta.queue') MQMNAME(MYQUEUEEMANAGER)
```

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 ileti üzerine 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(MYQUEUEEMANAGER)

/* This alias is put disabled and get enabled for application BETA */
CRTMQMQ QNAME('betas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*NO) GETENBL(*YES) MQMNAME(MYQUEUEEMANAGER)
```

ALPHA 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(MYQUEUEEMANAGER)

/* ALTER the base queue name, to which the alias resolves. */
/* FORCE = Force the change even if the queue is open. */
CHQMCMQ QNAME('alphas.alias.queue') TGTQNAME('orange.local.queue') FORCE(*YES)
MQMNAME(MYQUEUEEMANAGER)
```

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 **WRKMCMQ komut panosu**.

Kuyruk yöneticisi, bir uygulama tarafından model kuyruğu olarak tanımlanmış bir kuyruk adı belirleyen bir MQI çağrısı 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 tanımlarsınız. 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 özneliğinden, bu şablondan yaratılan gerçek kuyruklar kalıcı dinamik kuyruklardır. Belirtilmeyen öznitelikler, otomatik olarak SYSYTEM.DEFAULT.MODEL.QUEUE varsayılan kuyruğundan kopyalanır.

Model kuyruklarını tanımladığınızda, bunları yerel kuyruklarla aynı şekilde tanımladığınızda, REPLACE *YES özneliğini kullanabilirsiniz.

Model kuyruklarıyla diğer komutların kullanılması

Bir model kuyruğunun özneliklerini görüntülemek ya da değiştirmek için uygun komutları kullanabilirsiniz. Örneğin:

```
/* Display the model queue's attributes */
DSPMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('green.model.queue')

/* ALTER the model queue to enable puts on any */
/* dynamic queue created from this model. */
CHGMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('blue.model.queue') PUTENBL(*YES)
```

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, koşullardan biri örnek olarak verilebilir. 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 başarıyı yükselmesi) ve yeniden başlatma sırasında yinelemeleri en aza indirmenin etkisine sahiptir, bu nedenle yeniden başlatma süresini iyileştirir.

Tetikleme monitörü nedir?

Başlatma kuyruğunu işleyen program, tetikleme izleme uygulaması olarak adlandırılır ve tetikleme iletisinde yer alan bilgilere dayalı olarak, tetikleme iletisini okumanız ve uygun işlemi yapmak gerekir. Olağan durumda bu işlem, tetikleme iletisinin 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, sunulan işleri aşağıdaki gibi daha anlamlı iş adlarına sahip olmak için uyarlamak mümkündür:

1. İş tanımlamasında, istediğiniz tanımlamayı 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 tanımlamayı 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 kuyruktır. 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) TRGTYPE(*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.

MAXMSGLN(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 tanımlanmasında belirtilen başlangıç kuyruğuna koyar. Başlatma kuyruklarının özel ayarları yoktur, ancak yol göstermesi için yerel kuyruk motor.ins.init.queue'un aşağıdaki tanımlanmasını 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.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 **DSPMQMPRC** komutunu kullanın. Örneğin:

```
MQMNAME(MYQUEUEMANAGER) DSPMQMPRC('motor.insurance.quote.process')
```

Ayrıca, bir süreç tanımlamasını silmek için **CHGMQMPRC** komutunu ve var olan bir süreç tanımlamasını değiştirmek için **DLTMQMPRC** komutunu kullanabilirsiniz.

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 tıklayın.

```
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) +
RMTMQMNAME(QMGRB1) TMQNAME(XMITQ.TO.QMGRB1)

/* TCP/IP sender channel to send messages to the queue manager on SYSTEMB*/
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*SDR) +
MQMNAME(QMGRA1) TRPTYPE(*TCP) +
TEXT('Sender Channel From QMGRA1 on +
SYSTEMA to QMGRB1 on SYSTEMB') +
CONNNAME(SYSTEMB) TMQNAME(XMITQ.TO.QMGRB1)
```

4. SYSTEMB üzerinde bir kuyruk yöneticisi yaratmak için QMGRB1 adını tıklayın.

```
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 SYSTEMBarasında 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:

```
WRKMQMMSG QNAME(TARGETB) MQMNAME(QMGRB1).
```

IBM i IBM üzerindeki 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, 2023. 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
/*
/* *****
/*
/* Queue Manager Name Parameter
/* *****
/*
/* QSYS/PGM PARM(&QMGRNAME)
/*
/* *****
/*
/* 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 MQ for IBM i yönetimine ilişkin diğer yöntemler

CL komutlarının kullanılması, IBM MQ for IBM i denetimi için tercih edilen yöntemdir. Ancak, MQSC komutları, PCF komutları ve uzak denetim de içinde olmak üzere çeşitli diğer 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 238.

Kuyruk yöneticilerinin çalışmasını izlemek için IBM MQ özel işlemden geçirme olaylarını kullanabilirsiniz. IBM MQ özel işlemden 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 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 254.
2. Bir program, PCF iletilerini gönderen MQAI kullanılarak yazılıyor. Bkz. [“PCF 'lerin kullanımını kolaylaştırmak için MQAI' nin kullanılması”](#) sayfa 32.
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 i kullanılması”](#) sayfa 254.
4. Komutları dolaylı olarak bir uzak kuyruk yöneticisine göndermek için **STRMQMQSC** komutunu kullanın. Bkz. [“IBM üzerinde MQSC komutlarını kullanarak yönetim”](#) sayfa 252.

Örneğin, uzak bir kuyruk yöneticilerindeki 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 üzerinde MQSC komutlarını kullanarak yönetim

Use this information to learn about MQSC commands, and how to use them to administer IBM MQ for IBM i.

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, giriŸ 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 32 sayfa 253 içinde gösterilen panodaki etkin bir kuyruk yöneticisine karşı 26. seçeneği belirleniyor.

Böyle bir MQSC oturumunu sona erdirmek için endyazın.

Şekil 32 sayfa 253 , 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 32. 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, QMQNAME 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 bkz. “Komut dosyası (MQSC) komutları” sayfa 10.

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 yaratabilir 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 ağıdaki herhangi bir kuyruk yöneticisine PCF komutları yayınlayabileceğiniz şekilde yazabilirsiniz. Bu şekilde, yönetim görevlerini merkezileştirebilir ve otomatikleştirebilirsiniz.

Her PCF komutu, bir IBM MQ iletisinin 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 iletisi 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 iletisi 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 iletisini 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 32 ' e bakın.](#)

PCF veri yapılarının tam açıklaması ve bunların nasıl gerçekleştirileceği için bkz. [Komutlar ve yanıtlar için yapılar.](#)

IBM MQ Explorer ile IBM MQ for IBM 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 nesnelerinin tanımlarını tanımlayın, görüntüleyin ve değiştirin.
- Kuyruklardaki iletilere göz atın.
- Bir kanalı başlat ve durdur.
- Bir kanalla ilgili durum bilgilerini görüntüleyin.
- Bir kümedeki kuyruk yöneticilerini görüntüleyin.

- Hangi uygulamaların, kullanıcıların ya da kanalların belirli bir 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 istasyonlarınızdaki 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 **STRMQMSVR**.
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 kaçınınız.

Komut sunucusunu başlatmak ve durdurmak için ayrı denetim komutları vardır. IBM MQ Gezgini 'ni kullanarak, aşağıdaki bölümlerde açıklanan işlemleri gerçekleştirebilirsiniz.

Komut sunucusunun başlatılması ve durdurulması

Komut sunucusunu başlatmak için bu CL komutunu kullanın:

```
STRMQMSVR 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 33 sayfa 256](#) 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 33. MQM Komut Sunucusu panosunu görüntüle

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 nesnelere deęiřtirmeyin.

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. **deęil**, Integrated File System (IFS) içindeki nesnelere herhangi bir deęiřiklik 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 19 sayfa 257](#)' de kısaca açıklanmıştır.

Kuyruk Yöneticisiyle Çalış (WRKMQM) panosunda seçenek 22 'yi kullanarak, kuyruk yöneticisine bağlı tüm işleri görüntüleyebilirsiniz. WRKMQMLSR komutunu kullanarak dinleyicileri görüntüleyebilirsiniz.

Çizelge 19. IBM MQ Görevler.	
İş adı	İşlev
AMQALMPX	Günlük denetim noktalarını düzenli olarak alan denetim noktası işlemcisi.
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.

Çizelge 19. IBM MQ Görevler. (devamı var)	
İş adı	İşlev
AMQRPMPA	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.
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, QMQMJOBID varsayılan iş tanımlaması ile çalışır.

IBM MQ kurulumu sırasında sağlanan iş yönetimi nesnelere, Çizelge 20 sayfa 258 içinde listelenir ve bir kuyruk yöneticisi için oluşturulan nesnelere Çizelge 21 sayfa 259' ta listelenir.

Not: İş yönetimi nesnelere QMQM kitaplığında bulunabilir ve kuyruk yöneticisi nesnelere kuyruk yöneticisi kitaplığında bulunabilir.

Çizelge 20. İş yönetimi nesnelere		
Ad	Tip	Tanım
AMQALMPX	*JOBID	Denetim noktası işlemi tarafından kullanılan iş tanımlaması
AMQZLAA0	*JOBID	IBM MQ aracı işlemlere tarafından kullanılan iş tanımı
AMQZLSA0	*JOBID	Yalıtılmış bağ tanımları kuyruk yöneticisi aracısı
AMQZXMA0	*JOBID	IBM MQ yürütme denetleyiciler tarafından kullanılan iş tanımı
QMQM	*SBSD	Tüm IBM MQ işlerinin çalıştırıldığı altsistem
QMQM	*JOBQ	Belirtilen altsisteme bağlı iş kuyruğu
QMQMJOBID	*JOBID	Bir iş için belirli bir iş tanımlaması yoksa, varsayılan IBM MQ iş tanımlaması kullanılır.
QMQMMSG	*MSGQ	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

Çizelge 20. İş yönetimi nesneleri (devamı var)		
Ad	Tip	Tanım
QMQRUN50	*CLS	Düşük öncelikli IBM MQ işleri için bir sınıf açıklaması

Çizelge 21. Bir kuyruk yöneticisi için yaratılan iş yönetimi nesneleri		
Ad	Tip	Tanım
AMQA000000	*JRNRCV	Yerel günlük nesnesi
AMQAJRN	*JRN	Yerel günlük
AMQJRNINF	*USRSPC	Bir kuyruk yöneticisinin başlatılması ve ortam kurtarması için gereken en son günlük nesnelere güncellenen kullanıcı alanı. Bu kullanıcı alanı, hangi günlük nesnelere arşivlenmesini gerektireceğini ve hangilerinin güvenle silinebileceğini belirlemek için bir uygulama tarafından sorgulanabilir.
AJMQAJRNMSG	*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 nesnelere 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.

İş başlatmak için kullanılan iş tanımlaması, işin birçok özneliğini denetler. Örneğin:

- İşin kuyruğa alındığı iş kuyruğu ve işin çalıştırıldığı altsistem.
- İşin yürütüm değiştiricileri için kullandığı işi ve sınıfı başlatmak için kullanılan yönetme 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. QMQMJOBID 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 (QMQMJOBID) 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 (QMQRUN20, QMQRUN35ya da QMQRUN50) hangilerinin kullanıldığını tanımlar.

Not: IBM MQ işleri başlatılamıyorsa, altsistemin çalışır durumda olduğundan ve iş kuyruğunun tutulmadığından emin olun.

IBM MQ iş yönetimi nesnelere değiştirdiyse, 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 19 sayfa 257 ' 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 AMQZXMA0 exist in the queue manager library? Yes No
Does job description QMQMJOB0 exist in the queue manager library? Yes No
Does job description AMQZXMA0 exist in the QMQM library? Yes No
Does job description QMQMJOB0 exist in the QMQM library? Yes No
```

Bu soruların tümüne hayır yanıtını verdiyseniz, QMQM kitaplığında QMQMJOB0 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, izlenebilmesini 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, AMQZXMA0 yürütme denetleyicisi işi başlar.

1. Issue the **STRMQM** command for queue manager TESTQM.

2. IBM MQ , QMTESTQM adlı kuyruk yöneticisi kitaplığında, ilk olarak AMQZXMA0iş tanımlaması ve QMQMJOB0 iş tanımlaması için arama yapar.

Neither of these job descriptions exist, so IBM MQ looks for job description AMQZXMA0 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. AMQZXMA0 iş tanımındaki yöneltme verileri QMQRUN20 olduğundan, sistem, altsistem yöneltme girişlerinde bu verilerle eşleşen bir veri araması yapar.

Varsayılan olarak, 9900 sıra numarasına sahip yöneltme girdisinin QMQRUN20ile eşleşen karşılaştırma verileri vardır; bu nedenle iş, o yöneltme girişinde tanımlanan sınıfla (QMQRUN20adı da verilir) başlatılır.

5. QMQM/QMQRUN20 sınıfının çalıştırma önceliği 20 değerine ayarlandığı için, AMQZXMA0 işi, sistemdeki en etkileşimli işlerle aynı önceliğe sahip olan QMQM altsisteminde çalışır.

Daha sonra, AMQALMPX denetim noktası işlemi işi başlar.

1. IBM MQ , QMTESTQM adlı kuyruk yöneticisi kitaplığında, ilk olarak AMQALPMX iş tanımlaması ve QMQMJOBDB iş tanımlaması için arama yapar.

Neither of these job descriptions exist, so IBM MQ looks for job descriptions AMQALMPX and QMQMJOBDB in the product library QMQM.

İş tanımlaması AMQALMPX yok, ancak QMQMJOBDB yapıyor, bu nedenle işi sunmak için QMQMJOBDB kullanılır.

Not: QMQMJOBDB iş tanımlaması her zaman, kendi iş tanımlaması olmayan IBM MQ işleri için kullanılır.

2. İş tanımlaması, IBM MQ varsayılan iş kuyruğunu kullanır; bu nedenle, iş QMQM/QMQM iş kuyruğuna gönderilir.
3. QMQMJOBDB iş tanımlamasındaki yöneltme verileri QMQMRUN35' dir, bu nedenle sistem, altsistem yöneltme girişlerinde bu verilerle eşleşen bir veri arar.
Varsayılan olarak, 9910 sıra numarasına sahip yöneltme girdisinin QMQMRUN35 ile eşleşen karşılaştırma verileri vardır; bu nedenle iş, o yöneltme girişinde tanımlanan sınıfla (QMQMRUN35 adı da verilir) başlatılır.
4. QMQM/QMQMRUN35 sınıfının çalışma önceliği 35 olarak ayarlanmış; dolayısıyla, AMQALMPX işi, QMQM altsisteminde sistemdeki çoğu etkileşimli işten daha düşük bir önceliğe sahip, ancak çoğu toplu işten daha yüksek önceliğe sahip bir iş çalıştırıyor.

IBM i IBM i' ta iş yönetimi örneklerini yapılandırma

IBM MQ işlerinin yürütme ortamı özniteliklerini 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 yinelemelerini 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 çalıştırma 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: Yönelme sırası numarası için herhangi bir sayısal değer belirleyebilirsiniz, ancak değerlerin sıralı düzende olması gerekir. Bu sıra numarası, altsisteme, yönelme verilerinin bir yönelme verileri eşleşmesi için hangi yönelme girişlerinin aranacağı sıralamayı bildirir.

- d. Aşağıdaki komutu girerek, yerel ya da genel iş tanımlamasını yeni öncelik sınıfını kullanacak şekilde değiştirin:

```
CHGJOB JOB(QMQMLibname/QMQMJOB) RTGDTA('QMQMRUN10')
```

Şimdi QMLibraryname ile ilişkili tüm kuyruk yöneticisi işleri, 10 'un çalıştırma 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/QMQMJOB 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 yönelme girişleri yaratın ya da
- Hangi yönelme verilerinin kullanıldığına bakılmaksızın, QCMD ' yi çağıran, tüm yönelme 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(*SBSD) TOLIB(QMQM) NEWOBJ(QMQM2)
```

b. QMQM iş kuyruğunu kaldırmak için aşağıdaki komutu kullanın:

```
RMVJOBQE SBSB(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 SBSB(QMQM/QMQM2) JOBQ(QMQM/QMQM2) MAXACT(*NOMAX)
```

e. Aşağıdaki komutu girerek kuyruk yöneticisi kitaplığında yinelenen bir QMQMJOBQ yaratın:

```
CRTDUPOBJ OBJ(QMQMJOBQ) FROMLIB(QMQM) OBJTYPE(*JOBQ) TOLIB(QMlibraryname)
```

f. Yeni iş kuyruğunu kullanmak için, aşağıdaki komutu girerek iş tanımlamasını değiştirin:

```
CHGJOBQ JOBQ(QMlibraryname/QMQMJOBQ) JOBQ(QMQM/QMQM2)
```

g. Altsistemi başlatmak için aşağıdaki komutu kullanın:

```
STRSBS SBSB(QMQM/QMQM2)
```

Not:

- Altsistemi herhangi bir kitaplıkta belirleyebilirsiniz. Ürünün yeniden kurulması için herhangi bir nedenle ya da QMQM kitaplığı değiştirilirse, yaptığınız değişiklikler kaldırılır.
- QMlibraryname ile ilişkilendirilmiş tüm kuyruk yöneticisi işleri şimdi QMQM2alt sistemi altında çalışır.

5. Aşağıdaki örnek, bir iş tipine ilişkin tüm çıktıları toplar.

Birden çok kuyruk yöneticisi için tüm denetim noktası işlemini, AMQALMPX iş günlüklerini tek bir çıkış kuyruğunda toplamak için aşağıdaki adımları gerçekleştirin:

a. Örneğin, bir çıkış kuyruğu yaratabilirsiniz.

```
CRTOUTQ OUTQ(MYLIB/CHKPTLOGS)
```

b. Denetlemek istediğiniz IBM MQ işleminin adını kullanarak, QMQM/QMQMJOBQ iş tanımının genel bir kopyasını yaratın; örneğin,

```
CRTDUPOBJ OBJ(QMQMJOBQ) FROMLIB(QMQM) OBJTYPE(*JOBQ) NEWOBJ(AMQALMPX)
```

c. İş tanımlamasındaki çıkış kuyruğu parametresini, yeni çıkış kuyruğunuzu gösterecek şekilde değiştirin ve tüm iletilerin iş günlüğüne yazılacağı şekilde, iş günlüğü düzeyini değiştirin.

```
CHGJOBQ JOBQ(QMQM/AMQALMPX) OUTQ(MYLIB/CHKPTLOGS) LOG(4 00 *SECLVL)
```

Tüm kuyruk yöneticileri için tüm IBM MQ AMQALMPX işleri, yerel kuyruk yöneticisi kitaplığında yerel geçersiz kılma iş tanımlamaları olmaması koşuluyla, yeni genel AMQALMPX iş tanımlamasını kullanır.

Bu işlere ilişkin tüm iş günlüğü kuyruklama dosyaları artık MYLIB kitaplığındaki CHKPTLOGS çıkış kuyruğuna yazılıyor.

Not:

- a. Önceki örnek yalnızca, QPJOBLOG ya da herhangi bir yazdırma dosyasında çıkış kuyruğu parametresi için *JOB değeri varsa çalışır. Önceki örnekte, QSYS/QPDJOBLOG dosyasının OUTQ ' un *JOB olarak ayarlanması gerekir.
- b. Bir sistem yazdırma kütüğünü değiştirmek için CHGPRTF komutunu kullanın. Örneğin:

```
CHGPRTF PRTF(QJOBLOG) OUTQ(*JOB)
```

*JOB seçeneği, iş tanımlarınızın kullanılması gerektiğini belirtir.

- c. IBM MQ işleriyle ilişkili tüm kuyruğa yollanmış dosyaları belirli bir çıkış kuyruğunda gönderebilirsiniz. Ancak, kullanılmakta olan yazdırma dosyasının OUTQ parametresi için uygun değeri olduğunu doğrulayın.

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 kurtarılabilmesinin 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 eşgörünümü 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 dir.

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 büyüklüğüne ulaşırsa, bu değer ayırdır ve yeni bir günlük nesnesiyle değiştirilir. **CRTMQM** ve **THRESHOLD** parametresini kullanarak bir kuyruk yöneticisi yarattığınızda günlük nesnelere eşikini belirtebilirsiniz.

Yerel IBM MQ for IBM i günlük nesneleriyle ilişkili günlük nesneleri, her kuyruk yöneticisi kitaplığında bulunur ve aşağıdaki gibi bir adlandırma kuralını benimleyebilir:

AMQ Arnnnnn

burada:

B

A-Z bir harftir. Tek yönetim ortamı kuyruk yöneticileri için A . Bu, çok eşgörünümlü bir kuyruk yöneticisinin farklı eşgörünümlerine 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. AMQA1nnnnn , AMQA1 (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 269).
- 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 nesneleri 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 özniteliğinin başlangıç değeri, mqs.ini dosyasının LogDefaults kısmına yazılır.

Bir günlük nesnesi belirlenen eşik değerine geçtiğinde, günlük nesnesi ayrılır ve yeni bir günlük nesnesi yaratılır ve önceki günlük nesnesinden öznitelikleri devralır. Sistem otomatik olarak yeni bir günlük nesnesi bağlandığında, kuyruk yöneticisi yaratıldıktan sonra LogReceiverSize ya da LogASP özniteliklerinde 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:

QMGRLIB

Kuyruk yöneticisi kitaplığınızın adı

B

Eşgörünüm tanıtıcısıdır (genellikle A).

rnnnnn

Daha önce açıklanan adlandırma sırasında bir sonraki günlük nesnesi

xxxxxx

Yeni günlük nesnesi eşiği 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 269](#).

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 yöneticiliklerindeki 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) DSPJRNDTA(*YES)
```

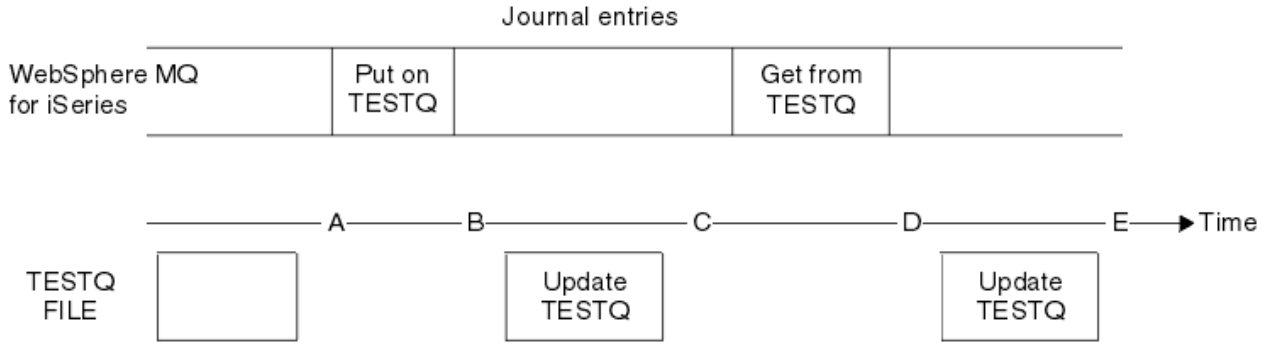
Kuyruklar daha fazla ileti işledikçe, denetim noktası kaydı kuyrukların yürürlükteki durumuyla tutarsız hale gelir.

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 34 sayfa 267](#)' da gösterilir.



Şekil 34. MQM nesnelerini 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 nesnelerinde 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 , nesnelerin 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 nesnelerini nasıl yönetmesine ilişkin ek bilgi için bkz. [“Availability, backup, recovery, and restart on IBM i”](#) sayfa 264.

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 269 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 nesnelerin resimleri alınır. Varsayılan olarak, ENDCCTJOB (*YES) parametresiyle **ENDMQM** komutu kullanılarak bir kuyruk yöneticisi sona erdirildiğinde, tüm nesnelerin 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 kaydetmeniz 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 nesneler 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 işlemi 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 kurtarılamadığını 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.

Denetim noktası işlemi AMQALMPX aşağıdaki noktalarda denetim noktasını almasından sorumludur:

- 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 konuya 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:

```
SAV0BJ 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 ayarladığınıza bağlıdır.

Yedekleme stratejinizin bir parçası olarak, günlük nesnelerinize göz kulak olun. Çeşitli nedenlerden dolayı, günlük nesnelere IBM MQ kitaplıklarından kaldırılması yararlı olur:

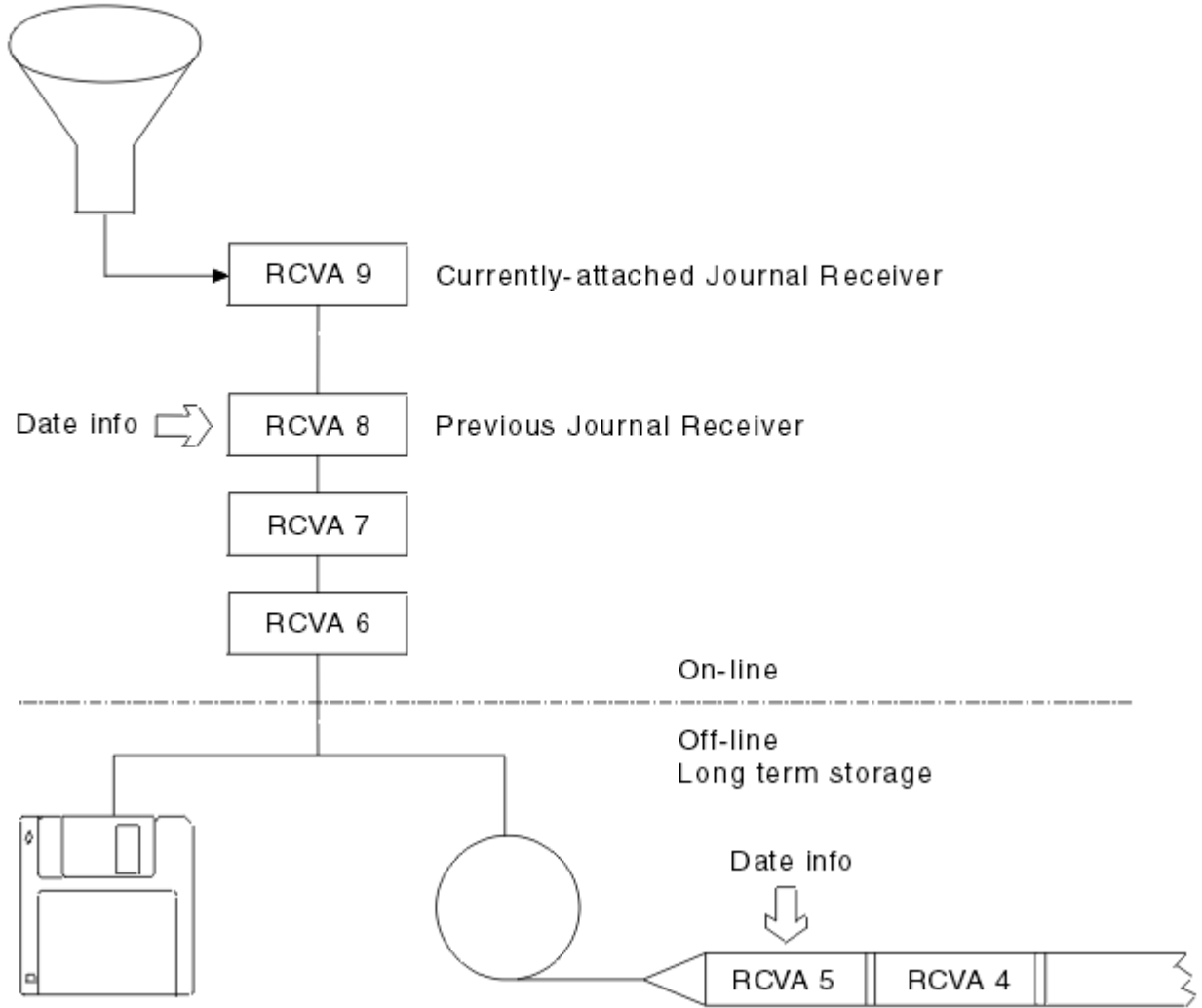
- Alanı serbest bırakmak için; bu, tüm günlük nesnelere için geçerlidir.
- Başlatma sırasında performansı artırmak için (STRMQM)
- Nesnelere yeniden yaratma başarımını yükseltmek için (RCRMQMOBJ)

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 35 sayfa 270 içinde gösterilir.

WebSphere MQ
for iSeries
Journal



Şekil 35. IBM tarihinde günlük kaydı

Yedeklenen bir günlük nesnesinin kuyruk yöneticisi kitaplığından ne zaman kaldırılabilirliğini 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 alkonması gereken en eski günlük nesnesinin adı. Bu kullanıcı alanına AMQJRNINF adı verilir ve veriler şu biçimde yazılır:

```
JJJJJJJJJLLLLLLLLLLLLYYYYMMDDHHMMSSmmm
```

Burada:

JJJJJJJJJ

IBM MQ ' un hala gereksinim duyduğu en eski günlük nesnesidir.

LLLLLLLLL

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: RCDMQMIMG 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 (QMQMMSG), 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 i **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 yedeklemesinde 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 yedeklemesi 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 kullanırız.)

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 üzerinde belirli bir kuyruk yöneticiye ilişkin günlük nesnelerinin geri yüklenmesi

Günlük nesnelerinin 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 nesnelerinin standart IBM i RSTOBJ komutu kullanılarak geri yüklenmesini gerektirir:

```
RSTOBJ OBJ(QMQMDATA/AMQA000005) OBJTYPE(*JRNRCV) .....
```

Günlük nesnelerinin, tek bir günlük nesnesi yerine geri yüklenmesi gerekebilir. Örneğin, AMQA000007 , IBM MQ kitaplıklarındaki en eski günlük nesnesidir 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 nesnelerini günlükle ilişkilendirmek için IBM i komutunu WRKJRNA komutunu kullanmanız gerekebilir.

Günlükleri geri yüklerken, sistem, günlük nesnesi sırasında yeni bir ad içeren bağlı bir günlük nesnesi yaratır. Ancak, yeni ad üretilen yeni ad, geri yüklemeniz gereken bir günlük nesnesiyle aynı olabilir. Bu sorunun üstesinden gelinmesi için el ile araya girilmesi gerekir; yeni bir günlük nesnesi ve günlük nesnesi geri yüklenmeden önce yeni bir günlük nesnesi yaratmak için gereklidir.

Örneğin, saklanmış AMQAJRN günlüğüyle ve aşağıdaki günlük nesnelere ilişkin 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 nesnesi, 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 264](#)):

```
CRTJRNRCV JRNRCV(QMQRLIB/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 üzerinde çok eşgörümlü kuyruk yöneticileri

Çok eşgörümlü 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 eşgörümlü 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 eşgörümlüde bulunan etkin eşgörümlüden 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 üzerinde güvenilirlik ve kullanılabilirlik

Çok eşgörümlü 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 eşleşmesini 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 eşgörümlü 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 eşgörümlü bir kuyruk yöneticisi, *kullanılabilirlik* için yapılandırılırken size yardımcı olur. Kuyruk yöneticisinin bir eşgörümlü bir kerede etkin olur. Bir standby eşgörümlünün 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 eşgörsümlü kuyruk yöneticilerini kullanarak yüksek düzeyde kullanılabilirlik çözümü oluşturun.

Çok eşgörsümlü 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 Javatafından 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 eşgörsümlü 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 Sürüm 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 eşgörümlü 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 eşgörümlüme ait yerel günlükle tam olarak zamanuyumlu olarak saklanan bir uzak günlük eşlemesidir.

Üç yapılandırma, çok eşgörsümlü bir kuyruk yöneticisine ilişkin günlüklerin nasıl yönetileceğini tasarlamaya ilişkin noktaları başlatıyor.

1. Etkin eşgörümlü ASP 'deki synchronized journal replication (günlük ikizlemenin), beklemedeki eşgörümler 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 [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 i 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 eşgörünümlerine 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.
- Paylaşım 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, paylaşım ile ilgili 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 fare ile sağ düğmesiyle tıklayın. **Dosya** > **İzinler** seçeneklerini tıklayın.

/QIBM/UserData/mqm/qmgrs nesnesi 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 tıklayın. Alternatif olarak, **CHFUSRPRF** ve **CHGPWD** komutlarını kullanın.

Sonuçlar

Share olanağını kullanarak diğer sunuculardaki GAMMA 'ya erişmenizi sağlar. Diğer görevleri gerçekleştiriyorsanız, /QNTC/GAMMA/WMQ yolunu 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 289 or “IBM üzerinde NetServer ve günlük ikizlemesi kullanılarak, tek bir eşgörümlü kuyruk yöneticisinin çok eşgörümlü kuyruk yöneticisine dönüştürülmesi” sayfa 293.

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 eşgörümlü bir kuyruk yöneticisinin günlük eşlemesini kullanacak şekilde yapılandırılıp yapılandırılmayacağına ya da bir IASP 'nin mi kullanılacağına belirlemeye karar vermede avantaj ve dezavantajlar vardır. İkizleme, kuyruk yöneticisinin bir uzak günlüğe zamanuyumlu olarak yazmasını gerektirir. Bir donanım bakış açısından, bu gereksinime başarıyı 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 eşgörümlü 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 eşgörümlü 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 eşgörümlü 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 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 işlemini 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 279](#)
2. Kuyruk yöneticisini taşıdığınız bağımsız yardımcı bellek havuzu (IASP); bkz. [“Bağımsız yardımcı bellek havuzları \(IASP ' ler\)” sayfa 279](#)
3. Bir küme kaynağı grubu (CRG); aşağıdakileri tanımladığınız [“Aygıt kümesi kaynak grupları” sayfa 280](#) ' a bakın:
 - Kurtarma etki alanı
 - IASP
 - Programdan çıkın; bkz. [“Aygıt CRG çıkış programı” sayfa 280](#)

IBM i Kümeler

Bir IBM i kümesi, mantıksal olarak birbirine bağlanmış olan, IBM i bilgisayarları ya da bölümleri olan eşgörünümün 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 bölümlenizi 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.

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 282 , 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 281 , bu işlemi gerçekleştirmenize yardımcı olacak bazı örnek kodları içerir.

İlgili kavramlar

“Bağımsız ASP 'ler ve yüksek kullanılabilirlik” sayfa 302

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:

```
ADDTCPFC INTNETADR(' [TAKEOVER IP]') LIND([LINE DESC])  
SUBNETMASK(' [SUBNET MASK]') AUTOSTART(*NO)
```

7. Devralınan IP adresini yalnızca birincil düğümde başlatmak için şu komutu kullanın:

```
STRTCPFC 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] *DEV *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] *DEV *ONLINE '[TAKEOVER IP]')
```

Bağımsız bir yardımcı bellek havuzuna taşıma ya da kaldırma, kuyruk yöneticisi ya da bağımsız bir yardımcı bellek havuzu

Kuyruk yöneticisini bağımsız bir yardımcı bellek havuzuna (IASP) ve bir kuyruk yöneticisini IASP ' den kaldırmak için gereken komutları gösteren 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 dizininizin 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 varı on olduğundan emin olun.

```
VRFCFG 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 kitaplığınızı 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 kitaplığınızı 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. Zamanuyumlu olmayan bir uzak günlüğü kullanarak bir kuyruk yöneticisini yeniden başlatmaya karar verdiğinizde, beklemedeki kuyruk yöneticisi yönetim ortamı, etkin eşgörünüm başarısız olmadan önce silinen iletileri yeniden işleyebilir ya da etkin eşgörünüm başarısız olmadan önce alınan iletileri işlenemez.

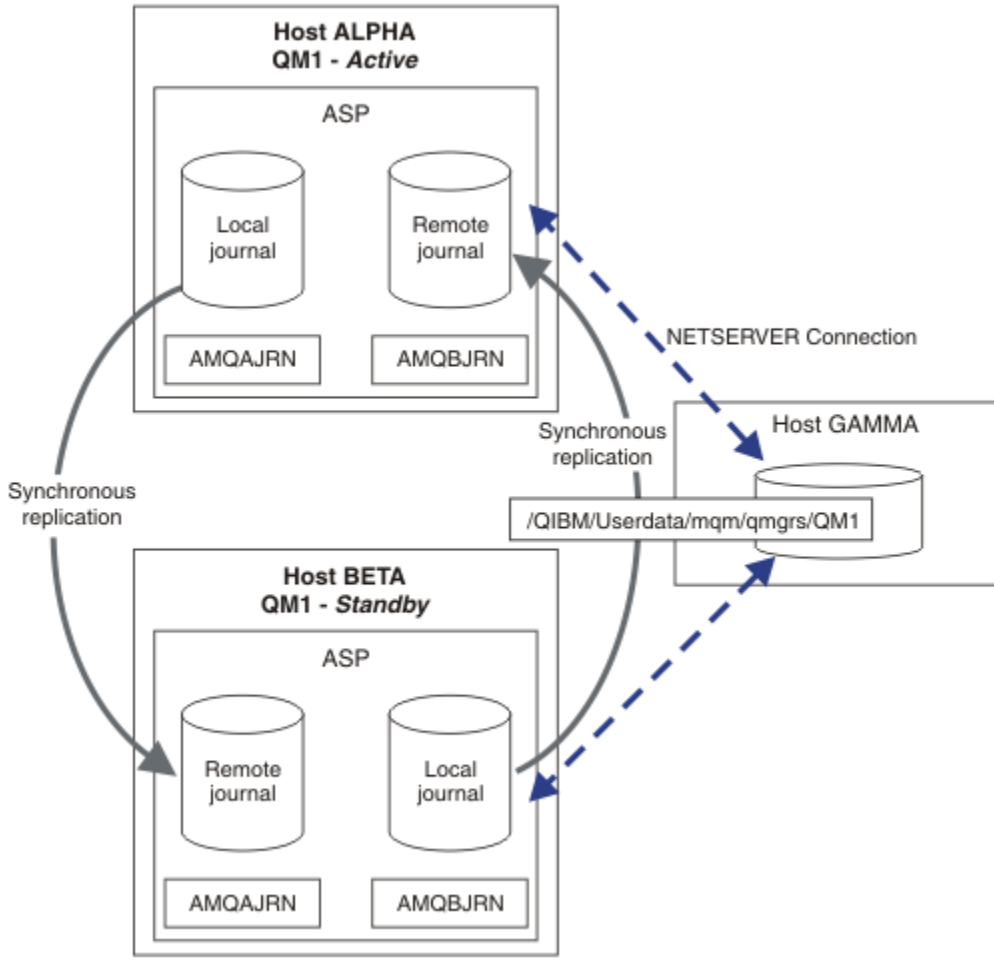
Başka bir olasılık da, dosya sisteminin en son denetim noktası kaydını içermesi ve yedek veritabanında eşzamanlanmamış bir uzak günlük içermemesi olabilir. Bu durumda kuyruk yöneticisi otomatik olarak yeniden başlatılmaz. Uzak günlük uyumlulaştırılıncaya ya da beklemedeki kuyruk yöneticisini dosya sisteminden 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.

[Şekil 36 sayfa 285](#)'ta ALPHA başarısız olursa, ikizlenmiş günlüğü kullanarak QM1 ' yi beta olarak el ile yeniden başlatabilirsiniz. Çoklu yönetim ortamı kuyruk yöneticisi yeteneğini QM1olarak ekleyerek, ALPHA 'daki etkin yönetim ortamı başarısız olursa, yedek QM1 yönetim ortamı BETA' ya otomatik olarak devam eder. QM1 , başarısız olan sunucu ALPHA ise, yalnızca etkin QM1yönetim ortamı değil, otomatik olarak da devam edebilir. BETA etkin kuyruk yöneticisi yönetim ortamının anasistemi olduğunda, beklemedeki yönetim ortamı ALPHA üzerinde başlatılabilir.

[Şekil 36 sayfa 285](#) , kuyruk yöneticisi verilerini saklamak için NetServer kullanarak bir kuyruk yöneticisinin iki eşgörünümü arasındaki günlüklerin kopyaladığına ilişkin bir konfigürasyon gösterir. Örüntüleri daha fazla günlük içerecek şekilde genişletebilir ve dolayısıyla daha fazla eşgörünüm ekleyebilirsiniz. Follow the journal naming rules explained in the topic, [“Queue manager journals on IBM i” sayfa 264](#). Şu anda bir kuyruk yöneticisinin çalışmakta olan eşgörünümlerinin sayısı iki ile sınırlıdır, biri etkin, diğeri ise yedekte bekleme durumunda.



Şekil 36. Kuyruk yöneticisi günlüğünü ikizle

ALPHA anasistem üzerindeki QM1 yerel günlüğü, AMQAJRN (ya da daha tam olarak, QMQM1/AMQAJRN) ve BETA ' da QMQM1/AMQBJRN adlı 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 *ASYNÇ uzak günlük eşlemesi

IBM i günlükleri zamanuyumlu (*SYNC) kullanılarak ikizlenir ya da zamanuyumsuz (*ASYNÇ) günlük kaydı; bkz. [Uzak günlük yönetimi](#).

Şekil 36 sayfa 285 içindeki eşleme kipi *SYNC, *ASYNÇdeğil. *ASYNÇ daha hızlıdır, ancak uzak günlük durumu *ASYNCPEND olduğunda bir hata oluşursa, yerel ve uzak günlük tutarlı değildir. Uzak günlüğün yerel günlükle birlikte olması gerekir. *SYNC seçeneğini belirlerseniz, tamamlanan yazma gerektiren bir çağrıdan dönmeden 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 sunucu yeniden kullanılabilir durumda olduğunda, zamanuyumlu uzak günlük kopyalamayı el ile yeniden etkinleştirmeniz gerekir. Günlükler daha sonra yeniden eşitlenir.

¹ Belirlenen süre, IBM i 5 ve IBM i 6.1 ' deki 1-3600 saniye aralığında belirlenen süre 60 saniyedir.

Şekil 36 sayfa 285 içindeki *SYNC / *SYNC yapılarındaki bir sorun, BETA 'daki yedek kuyruk yöneticisi yönetim ortamının denetime nasıl gireceğini içerir. BETA kuyruk yöneticisi yönetim ortamı ilk kalıcı iletilerini yazarken, uzak günlüğü ALPHA 'da güncelleme girişiminde bulunur. ALPHA 'dan BETA'ya geçiş kontrol nedeni ALPHA 'nın başarısızlığı ve ALPHA 'nın hala kapalı olması durumunda, uzaktan günlük kaydı ALPHA 'da hata oluşmasına neden olur. 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 seçeneğidir. Çizelge 22 sayfa 286 , bir kuyruk yöneticisi çifti arasında *SYNC ve *ASync günlük kaydı kullanılarak yapılan değiş tokuşları özetler.

Çizelge 22. Uzak günlük kaydı seçenekleri			
Etkin	Yedek	*SYNC	*ASync
*SYNC		<ol style="list-style-type: none"> 1. Yedek yedek sisteme geçiş ve yedek sisteme geçiş 2. Yedek yönetim ortamı, hata durumunda yedek sisteme geçiş işleminden hemen sonra devam etmez. 3. Uzak günlük kaydı her zaman kullanılabilir olmalıdır 4. Kuyruk yöneticisi başarımları uzak günlüğe kaydetmeye bağlıdır 	<ol style="list-style-type: none"> 1. Yedek yedek sisteme geçiş ve yedek sisteme geçiş 2. Yedek sunucu kullanılabilir durumda olduğunda, uzak günlüğe kaydetme işlemi *SYNC olarak değiştirilmelidir 3. Uzak günlüğe kaydetme işlemi yeniden başlatıldıktan sonra kullanılabilir durumda kalmalıdır 4. Kuyruk yöneticisi başarımları uzak günlüğe kaydetmeye bağlıdır
*ASync		<ol style="list-style-type: none"> 1. Mantıklı bir kombinasyon değil. 	<ol style="list-style-type: none"> 1. Hata durumunda yedek sisteme geçiş ya da yedek sisteme geçiş işleminden sonra bazı iletiler kaybolabilir ya da yinelenir 2. Etkin yönetim ortamının gecikmeden devam etmesi için, beklemedeki yönetim ortamının kullanılabilir durumda olması gerekmez. 3. Başarımları, uzak günlüğe kaydetmeye bağlı değildir

*SYNC / *ASync

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, zaman uyumlu 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 azaltan uzak günlük kaydı için beklemelidir.

Bir sunucu çifti arasındaki *SYNC / *SYNC yapılanışındaki hata durumunda yedek yönetim ortamını sürdürme gecikmesi de olumsuz olur. *SYNC / *ASYNC yapılanışında 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 işlemi *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 *ASYNC ' 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 eşgörünümlü kuyruk yöneticilerinin yönetimi perspektifinden, bir *SYNC / *ASYNC 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 geçiyse 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 *ASYNC 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 *SYNCdeğerine 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 barındırıldığı sunucu her zaman kullanılabilir olmalıdır.

Yedek kuyruk yöneticisinin hata durumunda yedek sisteme geçiş işleminden sonra hızlı bir şekilde devam etmesi için *SYNC / *ASYNC seçeneğini belirleyin. Yeni etkin yönetim ortamında uzak günlük ayarını *SYNC olarak el ile geri yüklemeniz gerekir. *SYNC / *ASYNC yapılanışı, bir çift çok eşgörünümlü 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.

*ASYNC / *ASYNC

Etkin ve beklemedeki kuyruk yöneticilerini barındıran sunucuların her ikisi de *ASYNC 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 *ASYNC / *ASYNC seçeneğini belirleyin ve hata durumunda yedek sisteme geçiş ya da yedek sisteme geçiş işleminden sonra bazı iletileri gevşetmeye ya da yinelemeye hazırlıksınız.

*ASYNC / *SYNC

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. İleti 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 *ASYNC ya da *SYNC 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 eşgörümlü bir kuyruk yöneticisi yaratılması

İki IBM i sunucusunda çalışmak için çok eşgörümlü 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 **ADDQMJRN** 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. IBM MQ must be at least at version 7.0.1.1.
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 37 sayfa 292 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 276.

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 eşgörümlerine 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) UDLSGQ(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.

```
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.

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.

```
ADDMQMINF 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.

```
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 290adımındaki komut yerine BETA 'dan ALPHA' ya zamanuyumsuz günlük kaydı ayarlamak için bu komutu kullanın.

```
ADDMQMJRN MQMNAME (QM1) RMTJRNRDB (ALPHA) RMTJRNDLV (*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 283

içindeki bilgileri kullanın. Varsayılan değer, uzak gnlkten gelen bir yanıt iin 60 saniyelik bekleme sresiyle zamanuyumlu olarak eŐlenmektedir.

8. ALPHA ve BETA 'daki gnlklerin etkinleŐtirildiĐini ve uzak gnlk kopyalamanın durumunun olduĐunu doĐrulayın.

a) ALFA ' DA:

```
WRKMQMJRN MQMNAME(QM1)
```

b) -Evet.

```
WRKMQMJRN MQMNAME(QM1)
```

9. ALPHA ve BETA 'da kuyruk yneticisi eŐgrnmlerini baŐlatın.

a) ALPHA 'da ilk ynetim ortamını baŐlatın ve etkin ynetim ortamı haline bakın. Yedek bir ynetim ortamına gemenin etkinleŐtirilmesi.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) BETA 'da ikinci eŐgrnm baŐlatın ve yedek ynetim ortamı haline getirsin.

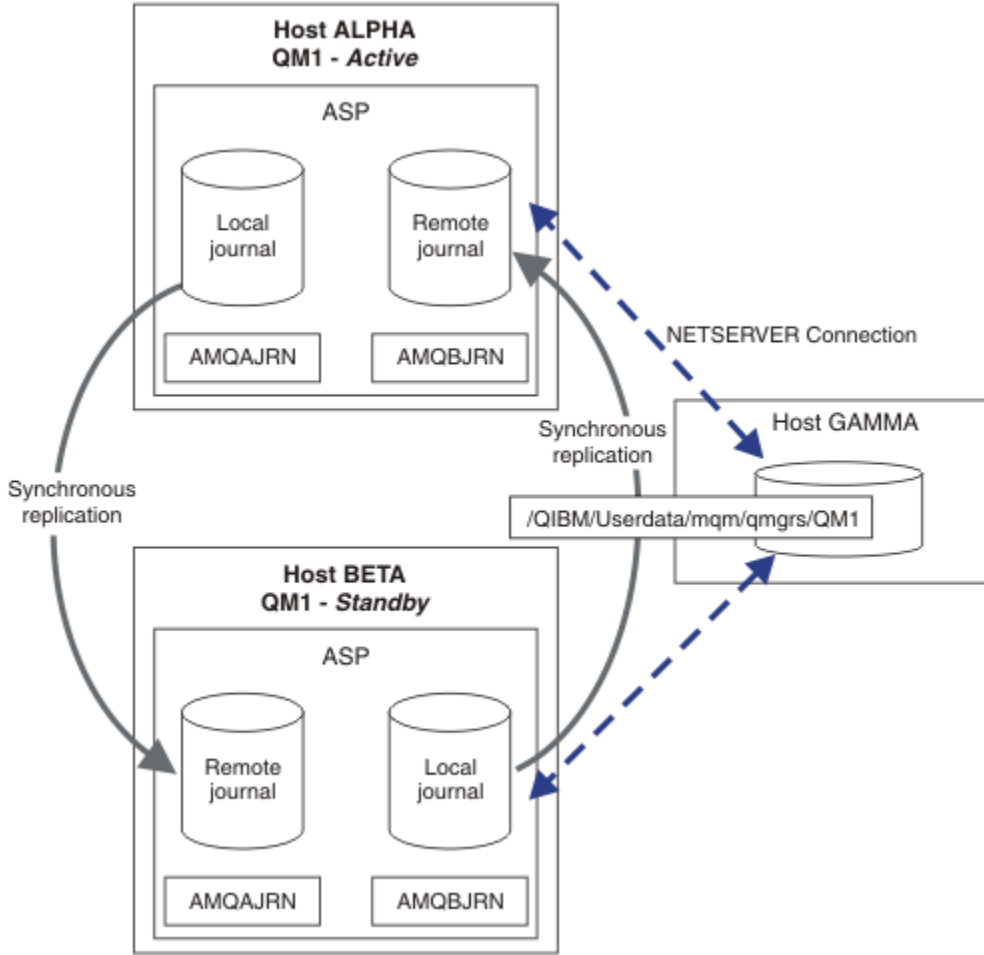
```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

Sonuçlar

Kuyruk yneticisi durumunu denetlemek iin kullan:

1. ALPHA 'daki kuyruk yneticisi rneĐinin durumu olmalıdır.
2. BETA 'da kuyruk yneticisi ynetim ortamının durumu olmalıdır.

Örnek



Şekil 37. 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 38 sayfa 296 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 çalıştıracak 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 'da 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. **ADDMQJRN** , günlüklerin ve bağlantıların eklenmesini basitleştirir.
- Kuyruk yöneticisini başlatın ve bir yedek yönetim ortamı izin veriyor.

Not:

Görevin “4” sayfa 294 . adımında, tek yönetim ortamı 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 294. Restore the queue manager library after step “5” sayfa 294.

Not:

Görevin “5” sayfa 294 . adımında QM1 yeniden yaratıyorsunuz. 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 276.

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 eşgörünümüne 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.

```
ADDQMQRN 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=QMOM1
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.

```
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.

11. BETA ' da kaç. Bu komut, BETA 'da yerel bir günlük ve ALPHA' da uzak bir günlük ekler.

```
ADDQMQRN 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 290adımındaki komut yerine BETA 'dan ALPHA' ya zamanuyumsuz günlük kaydı ayarlamak için bu komutu kullanın.

```
ADDQMQRN MQMNAME (QM1) RMTJRNRDB (ALPHA) RMTJRNDLV (*ASYNCR)
```

Sunucu ya da günlük kaydı ALPHA 'da hatanın kaynağı varsa, BETA yeni günlük girişlerinin ALPHA' ya kopyalanacağını beklemeden başlar.

Eşleme kipini *SYNCRolarak 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 283 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:

```
WRKMQRN 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.

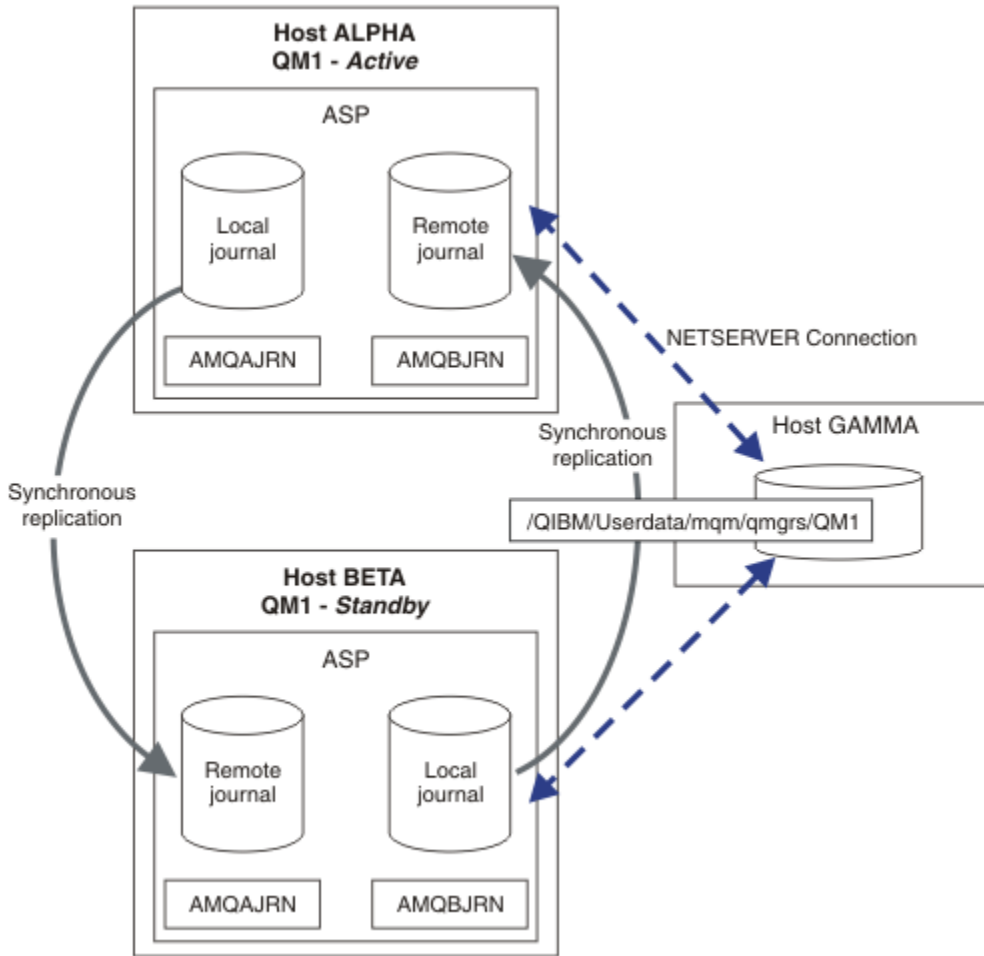
```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

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 38. 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)
```

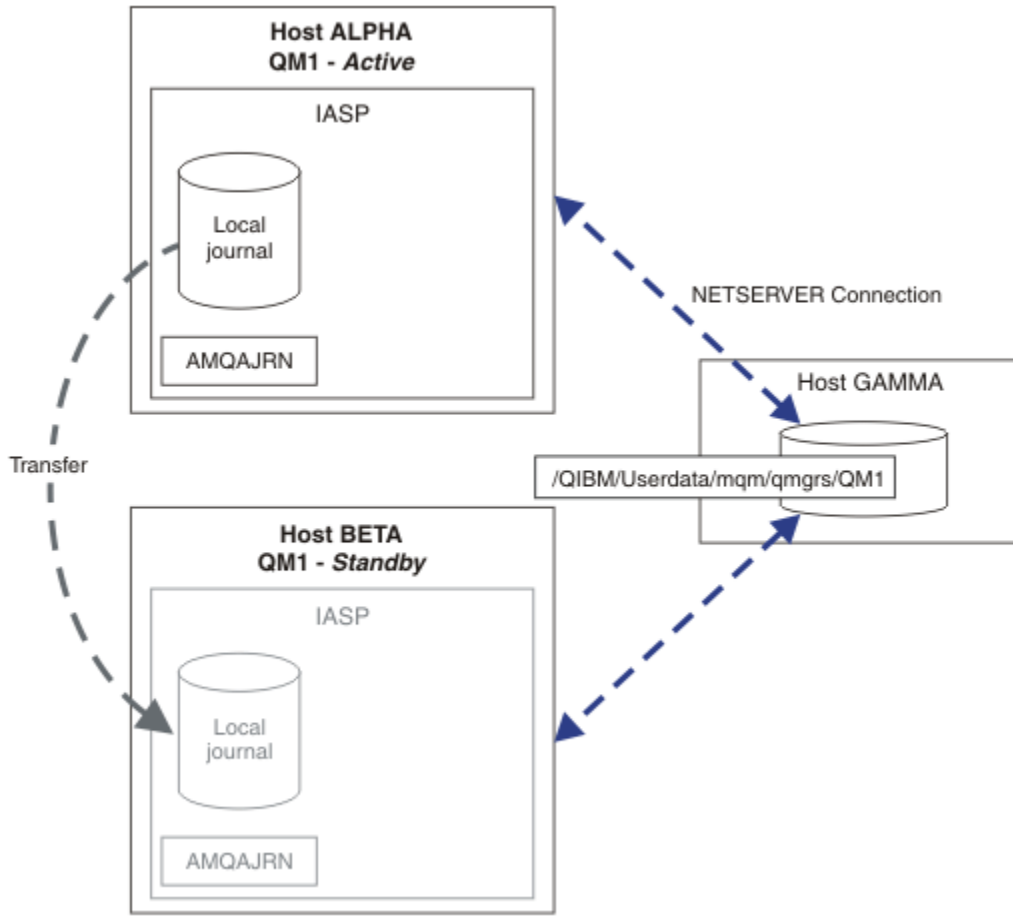
- Diğer yüksek kullanılabilirlik yapılandırmalarına bakın:
 1. Kuyruk yöneticisi verilerini bir Windows sunucusuna yerleştirmek için NetServer ' ı kullanın.
 2. Kuyruk yöneticisi günlüğünü ikizlemek için uzak günlük kaydı işlevini kullanmak yerine, günlüğü bağımsız bir ASP ' ye saklayın. Bağımsız ASP '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 eşgörünümlerini barındıran sunucular aynı küme kaynağı grubunda yer aldıysa, 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 39. Bir kuyruk yöneticisi günlüğünü bağımsız bir ASP kullanarak aktarma

Çok eşgörunümlü 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 eşgörunümlü kuyruk yöneticisi verilerini saklayamazsınız.

Paylaşılan kütük sistemi yapılarındaki iki rolü gerçekleştirir: Aynı kuyruk yöneticisi verileri, kuyruk yöneticisinin tüm yönetim ortamlarını daha iyi paylaşır. Dosya sisteminde, kuyruk yöneticisinin yalnızca bir eşgörunümünün, 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 eşgörunüm tarafından tutulan kuyruk yöneticisi verilerine kilitleme işlemi serbest bırakmalıdır. Bundan sonra, beklemedeki kuyruk yöneticisi yönetim ortamı kuyruk yöneticisi verilerine okuma/yazma erişimi kazanabilir. Dosya sistemi iletişim kuralı, çok eşgörunümlü kuyruk yöneticileriyle doğru bir şekilde çalışmak için bir kural kümesine uygun olmalıdır; bkz. ["IBM üzerinde yüksek kullanılabilirlik çözümünün bileşenleri"](#) sayfa 275.

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 geriletir. Kuyruk yöneticisi hizmetleri olarak başlatılacak şekilde yapılandırılan uygulamalar başlatılır.

Bağımsız ASP 'deki hatalı etkin kuyruk yöneticisi eşgörunümünden 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 eşgörunümlü 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 302](#) 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. IBM MQ must be at least at version 7.0.1.1.
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 tasarlarlarken *SupportPac MC41: Configuring IBM MQ for iSeries for High Availability* yöntemlerinden bazılarını bulabilirsiniz.

Bu görev hakkında

Şekil 40 sayfa 301 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 276](#).

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.

```
ADDMQMINF 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.

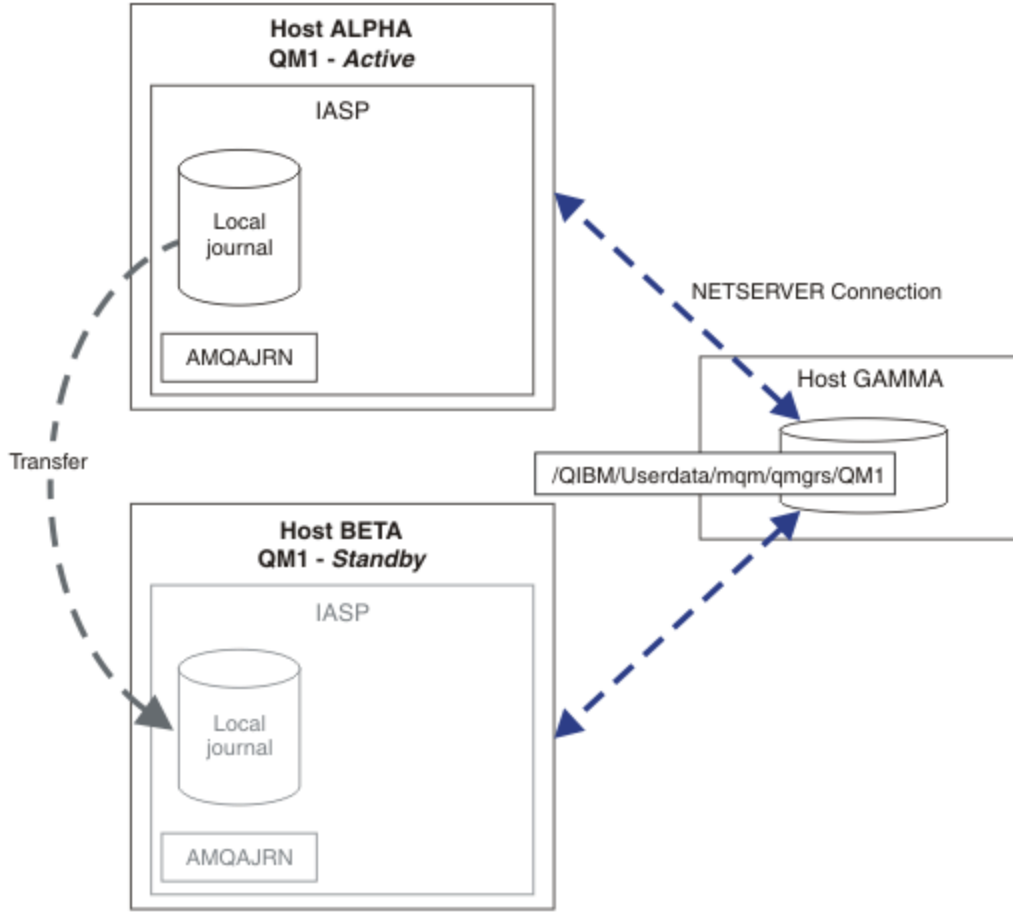
```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

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 40. 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.

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 performansı artırabilir ve kurtarma süresini kısaltabilirsiniz. Birden çok temel ASP ' nin kullanılması, disk hatasına karşı bir dereceyalıtım 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 çalıştırmanız 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 kullanabilir 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 289.
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')
RMTJRN RDB('BETA') RMTJRNSTS(*INACTIVE)
```

- b) -Evet.


```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('ALPHA') RMTJRNSTS(*INACTIVE)
```

- Her iki QM1 eşgörünümünü durdurmak için, etkin kuyruk yöneticisi yönetim ortamı olan ALPHA üzerinde **ENDMQM** komutunu çalıştırın.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) INSTANCE(*ALL) ENDCCTJOB(*YES)
```

- 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 yapılandırma bilgilerini ALPHA ' dan kaldırır. Günlük adının öneki AMQ ise, QM1 ile ilişkilendirilmiş yerel günlüğü ALPHA ' dan siler. Günlük adı öneki AMQ ise ve uzak bir günlük yaratıldıysa, uzak günlüğü BETA ' dan da kaldırır.

- 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 yapılandırma bilgilerini BETA' dan kaldırır. If the journal name is prefixed by AMQ, it deletes the local journal associated with QM1 from BETA. Günlük adı öneki AMQ ise ve uzak bir günlük yaratıldıysa, uzak günlüğü ALPHA ' dan da kaldırır.

Sonuçlar

DLTMQM ve **RMVMQMINF** , **CRTMQM** ve **ADDQMJRN** 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 mqsc.ini.

Sonraki adım

Alternatif bir yaklaşım, "1" sayfa 304 adımımda 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.

- ALFA ' DA:

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('BETA')
```

- Evet.

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('ALPHA')
```

Günlükleri sildikten sonra, diğer adımları da devam ettirin.

IBM üzerinde çok eşgörünümlü bir kuyruk yöneticisinin yedeklenmesi

Bu yordam, yerel sunucudaki kuyruk yöneticisi nesnelerini 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 289. IBM MQ , ALPHA ve BETA sunucularda 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')
```

IBM i

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 zamanuyumlu mu, yoksa zamanuyumsuz mu olacağını, zamanuyumlu zamanaşımını mı, yoksa uzak günlüğün hemen etkinleştirilip etkinleştirilmeyeceğini yapılandırabilirsiniz.

CHGMQMJRN

Komut, eşleme günlüklerini etkileyen zamanaşımı, durum ve teslim parametrelerini değiştirir.

RMVMQMJRN

Bir kuyruk yöneticisi eşgörümünden adlandırılmış *uzak* günlüklerini kaldırır.

WRKMQMJRN

Yerel kuyruk yöneticisi yönetim ortamı için yerel ve uzak günlüklerin durumunu listeler.

mqs.ini dosyasını deęiřtiren ařaęıdaki komutları kullanarak ek kuyruk yneticisi ynetim ortamları ekleyin ve bunları ynetin.

ADDMQMINF

Komut, farklı bir IBM i sunucusuna yeni bir kuyruk yneticisi rneęi eklemek iin, DSPMQMINF komutuyla mqs.ini dosyasından atıęınız bilgileri kullanır.

RMVMQMINF

Bir kuyruk yneticisi ynetim ortamını kaldırın. Bu komutu, varolan bir kuyruk yneticisinin eřęrnmn kaldırmak ya da farklı bir sunucudan silinmiř bir kuyruk yneticisine iliřkin yapılanıř bilgilerini kaldırmak iin kullanın.

ok eřęrnmliler bir kuyruk yneticisinin yapılandırılmasına yardımcı olmak iin **CRTMQM** komutunun  deęiřtirgesi vardır.

MQMDIRP (*DFT | *dizin-neki*)

Aę zerinde alıřan depolama zerinde kuyruk yneticisi verilerine eřlenmiř bir baęlama noktası semek iin bu parametreyi kullanın.

ASP (*SYSTEM|*ASPDEV| *yardımcı-depolama-havuz-no*)

Kuyruk yneticisi gnlęn sisteme ya da temel bir kullanıcı ASP ' ye yerleřtirmek iin *SYSTEMya da *yardımcı-saklama-havuzu-no* deęerlerini belirtin. Kuyruk yneticisi gnlęn baęımsız bir ASP ' ye yerleřtirmek iin, *ASPDEV seeneęini belirleyin ve **ASPDEV** parametresini kullanarak bir aygıt adı da ayarlayın.

ASPDEV (*ASP|*aygıt-adi*)

Birincil ya da ikincil baęımsız ASP aygıtı iin bir *aygıt adı* belirleyin. *ASP seeneęinin belirlenmesi, **ASP** (*SYSTEM) deęerinin belirlenmesine neden olur.

IBM i IBM zerinde performans ve disk kaybına uyum sorunları

Performansı ve gvenilirlięi artırmak iin farklı yardımcı depolama havuzları kullanın.

Uygulamalarınızda ok sayıda kalıcı ileti ya da byk ileti kullanıyorsanız, bu iletinin diske yazılması, sistemin bařarımında nemli bir faktr haline gelir.

Bu olanakla bař etmek iin yeterli disk etkinleřtirmenizin olduęundan emin olun ya da kuyruk yneticisi gnlk alıcılarınızın tutulacaęı ayrı bir Yardımcı Bellek Havuzu (ASP) dřnn.

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 yneticisi kitaplıęı ve gnlkler ve IFS verileri sistem ASP ' inde saklanır.

ASP ' ler, belirli bir ya da daha ok disk birimindeki nesnelerin yalıtılmasına olanak tanır. Bu, bir disk ortamı hatası nedeniyle veri kaybını da azaltabilir. oęu durumda, yalnızca etkilenen ASP ' deki disk birimlerinde saklanan veriler kaybedilir.

Ayrı kullanıcı ASP ' lerindeki kuyruk yneticisi kitaplıęını ve gnlk verilerini, hata durumunda yedek sisteme geiř saęlamak ve disk ekiřmesini azaltmak iin kk IFS dosya sistemindeki ayrı ayrı kullanıcı ASP' lerinde saklamanız nerilir.

Ek bilgi iin [Yedekleme ve kurtarmabařlıklı konuya](#) bakın.

IBM i IBM i kitaplıęındaki IBM MQ kitaplıklarını kaydetmek iin SAVLIB ' in kullanılması

IBM MQ kitaplıklarını saklamak iin SAVLIB LIB(*ALLUSR) kullanamazsınız; bu kitaplıkların adları Q ile bařlayan adlara sahiptir.

Tm kuyruk yneticisi kitaplıklarını saklamak iin SAVLIB LIB(QM*) komutunu kullanabilirsiniz; ancak, *SAVF dıřında bir saklama aygıtı kullanıyorsanız, bu kitaplıkları kullanabilirsiniz. DEV (*SAVF) iin, sisteminizdeki her kuyruk yneticisi kitaplıęı iin 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 308](#)). ya da tüm kuyruk yöneticilerine bakın (bkz. [“IBM MQ for IBM için tüm kuyruk yöneticilerini kapatma” sayfa 310](#)).

ENDMQM parametresi ENDCCTJOB (*YES)

ENDMQM parametresinde ENDCCTJOB (*YES) değeri, önceki sürümlerle karşılaştırıldığında, IBM MQ for IBM i 6.0 ve sonraki sürümlerde farklı bir şekilde çalışır.

Önceki sürümlerde, ENDCCTJOB (*YES) belirttiğinizde, MQ uygulamanızı sizin için zorla sonlandırmanızı sağlar.

IBM MQ for IBM i 6.0 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. Gerekliyse, 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) DSPJRNDTA(*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 işlemi 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 yöneticisini 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)
```

Olğan dışı koşullar altında sona erdirin

1. Kuyruk yöneticilerini sona erdirmek için aşağıdaki komutu yürütün:

```
ENDMQM MQMNAME(QMgı1) OPTION(*IMMED)
```

Sona erdirmek istediğiniz her kuyruk yöneticisi için bunu yineleyin; ayrı komutlar koşut olarak çalışabilir.

2. Aşağıdaki altsistemleri (örneğimizde SUBX) sona erdirin:

```
ENDSBS SUBX *IMMED
```

Sona erdirmek istediğiniz her altsistem için bu işlemi yineleyin; ayrı komutlar koşut olarak 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(QMgı1)
```

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 Programlanır 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 312.](#)

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 320.](#)

IBM MQ for z/OS ' in nasıl yönetileceği ve üstlenmeniz gereken farklı yönetim görevlerine ilişkin ayrıntılar için aşağıdaki bağlantılara bakın:

İlgili kavramlar

[“YönetmeIBM MQ” sayfa 5](#)

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

İlgili görevler

[“Yerel IBM MQ nesnelere yönetme” sayfa 133](#)

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

[“Uzak IBM MQ nesnelere yönetme” sayfa 184](#)

Bu kısımda, MQSC komutları kullanılarak uzak kuyruk yöneticilerindeki IBM MQ nesnelere nasıl denetleneceği ve iletiler ve yanıt iletilerinin hedefini denetlemek için uzak kuyruk nesnelere nasıl kullanılacağı anlatılıyor.

İlgili bilgiler

[IBM MQ for z/OS kavramlar](#)

[Planlama](#)

[z/OSüzerinde IBM MQ ortamınızın planlanması](#)

[Yapılandırılıyor](#)

[yapılandırmaz/OS](#)

[Programlanabilir komut biçimleri başvurusu](#)

[MQSC başvurusu](#)

[IBM MQ for z/OS yardımcı programlarının kullanılması](#)

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).
- Kullanıma hazırlama giriş veri kümeleri.
- Sıralı bir veri kümesindeki komutların bir listesini işlemek için sağlanan toplu iş yardımcı programı CSQUTIL.
- 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 24 sayfa 315](#) , 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 MQdenetimi 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 yayılıyor. PCF komutlarının kullanılmasıyla ilgili ek bilgi ve komutların ayrıntıları için [Programlanabilir komut biçimleri başvurusu](#) belgelerine bakın.

IBM MQ for z/OSüzerinde 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 nesnelere 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 nesnelere QMGR yok etme eğilimlerinden birini içerir.

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 tanımını değiştirir.

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, nesnelere 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 kuyruktaki 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* 'yı 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 değiştirmeden 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.

IBM MQ for z/OS için komut özeti

Bu konuyu ana MQSC ve PCF komutlarının bir başvurusu olarak kullanın.

[Çizelge 23 sayfa 315](#), IBM MQ nesnelerini değiştirmek, tanımlamak, silmek ve görüntülemek için IBM MQ for z/OS üzerinde kullanılabilir olan MQSC ve PCF komutlarını özetler.

Çizelge 23. Nesne tipine göre ana MQSC ve PCF komutlarının özeti

MQSC komutu	ALTER	Tanımla	GÖRÜNTÜLE	SİL
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 23 sayfa 315' ta özetlenenlere ek olarak diğer işlemleri de gerçekleştirmenize olanak sağlayan birçok başka MQSC ve PCF komutu vardır.

Çizelge 24 sayfa 315 , 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 24. 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

Çizelge 24. MQSC komutlarının çalıştırılacağı kaynaklar (devamı var)

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
ALTER CFSTRUCT		X	X	X
KANALı 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 LISTESİNİ 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
ALT			X	X
KONUUYU TANIMLA		X	X	X

Çizelge 24. MQSC komutlarının çalıştırılacağı kaynaklar (devamı var)

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
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
KONUYU 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 EKLAN		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
QMGR GÖRÜNTÜLE		X	X	X
QMODEL ' I GÖRÜNTÜLE		X	X	X

Çizelge 24. MQSC komutlarının çalıştırılacağı kaynaklar (devamı var)

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
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
BSS ' 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 Sü			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
CMDSERV BAŞLAT	X	X	X	
DINLEYİCİ BAŞLAT			X	X

Çizelge 24. MQSC komutlarının çalıştırılacağı kaynaklar (devamı var)

Komut	CSQINP1	CSQINP2	z/OS Konsol	Komut giriş kuyruğu ve sunucusu
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.

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 nesnelerini 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 nesnelerini 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 parametresini 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. Dağıtımli 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 deyiimiyle 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 kullanabilmeniz 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 gerçekleştirin.
- Komutları verin ve nesne tanımlamalarını işler.
- Veri dönüştürme çıkışları oluşturun.
- Önyükleme veri kümesini değiştirin.
- Günlüklere ilişkin bilgileri listele.
- Günlükleri yazdırın.
- Db2 tablolarını ve diğer Db2 yardımcı programlarını ayarlayın.
- 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.

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 kuyruklarını, 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 bir 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).

Çalışırken IBM MQ for z/OS

IBM MQ for z/OS' i çalıştırmak için bu temel yordamları kullanın.

Bu bölümde açıklanan işlemleri, IBM MQ for Windowsçindistributed, IBM MQ for Linux (x86 ve x86-64 platformları) ve SupportPac MSOTile birlikte dağıtılan IBM MQ Explorerişletim sistemi kullanılarak da gerçekleştirebilirsiniz. Daha fazla bilgi için bkz. [“IBM MQ Explorerkomutunu kullanarak yönetim” sayfa 125 ve IBM Destek ve yüklemeler](#).

Bu bölümde aşağıdaki konulara ilişkin bilgiler yer almaktadır:

IBM MQ denetim komutlarını bir z/OS konsolundan ya da yardımcı program CSQUTIL yardımcı programıyla yayınlatabilirsiniz. Komutlar, komutu hangi IBM MQ altsisteminin 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 özneliklerin nasıl belirtileceği anlatılır ve bu komut, PCF adları yerine MQSC komut adlarını kullanarak bu komutları ve öznelikleri belirtir. MQSC komutlarının sözdizimine ilişkin ayrıntılar için bkz. MQSC komutları. PCF komutlarının sözdizimine ilişkin ayrıntılar için bkz. [“IBM MQ Programları Komut Biçimlerinin Kullanılması” sayfa 21](#). 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 319](#) 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 342](#) .

Bu komutların işlevlerinin çoğu, operasyon ve denetim panelleri tarafından, TSO ve ISPF ' den erişilebilen ve [“IBM MQ for z/OS için işlemler ve denetim panoları” sayfa 328](#) içinde açıklandığı şekilde uygun bir şekilde sağlanır.

Ek bilgi için bkz.

- [“Issuing commands from a z/OS console or its equivalent” sayfa 323](#)
 - [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 yayınlatabilirsiniz. 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 MGCRE 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 yayınlamazsınız. 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 41 sayfa 324](#) içinde gösterildiği gibi bir komut öneki dizisiyle (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 dizilerinin 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 41 sayfa 324](#). Ancak, karmaşık komutlar için ya da sık sık kullandığınız komut kümeleri için, komut verme yöntemlerinin diğer yöntemleri daha iyi olur.

```
+CSQ1 DISPLAY QUEUE(TRANSMIT.QUEUE.PROD) TYPE(QLLOCAL)
```

Şekil 41. 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, MGCRE 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 yayınlatabilirsiniz. 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 Önce IBM MQ” sayfa 325](#)
- [“Kuyruk Yöneticisinin Başlatılması” sayfa 325](#)
- [“Kuyruk Yöneticisinin Durdurulması” sayfa 327](#)

Kuyruk yöneticisini başlatma ve durdurma görece olarak basittir. Kuyruk yöneticisi olağan koşullarda durduğunda, son işlemi sonlandırma denetim noktasını almalıdır. 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 Önce IBM MQ

IBM MQ kurulduktan sonra, bu ürün resmi bir z/OS altsistemi olarak tanımlanır. Bu ileti, z/OS' un ilk program yüklemesi (IPL) sırasında görüntülenir:

```
CSQ3110I +CSQ1 CSQ3UR00 - SUBSYSTEM ssnm INITIALIZATION COMPLETE
```

Burada *ssnm* , IBM MQ altsistem adıdır.

Şu andan itibaren, o altsistem için kuyruk yöneticisini başlatabilirsiniz *Sistem denetimi komutlarını verme yetkisi bulunan herhangi bir z/OS konsolundan* ; bu, bir z/OS SYS komut grubudur. 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 42 sayfa 325 , START komutuna ilişkin örnekleri gösterir. (Komut öneki dizgisi (CPF) olan bir IBM MQ komutuna örnek atmanız gerektiğini unutmayın.)

```
+CSQ1 START QMGR
+CSQ1 START QMGR PARM(NEWLOG)
```

Şekil 42. 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='ZPARM(CSQ1ZPRM)'
```

Bir sistem parametre modülü, kuyruk yöneticisi uyarlandığında belirtilen bilgileri sağlar.

V 9.0.3 IBM MQ 9.0.3 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.

V 9.0.3 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 değiştirmedir. Bu, başlatma yordamını değiştirmeden CSQINP2 DDadını değiştirebileceğiniz anlamına gelir. Bu, işlemler ve kuyruk yöneticisi işlemleri için değişiklikleri uygulamak, değişiklikleri uygulamak için kullanışlıdır.

Kuyruk yöneticisi CSQ1 için başlatma yordamınızın Şekil 43 sayfa 326 gibi görüldüğünü varsayın.

```
//CSQ1MSTR PROC INP2=NORM
//MQMESA EXEC PGM=CSQYASCP
//STEPLIB DD DISP=SHR,DSN=thlqual.SCSQANLE
// DD DISP=SHR,DSN=thlqual.SCSQAUTH
// DD DISP=SHR,DSN=db2qual.SDSNLOAD
//BDS1 DD DISP=SHR,DSN=myqual.BSDS01
//BDS2 DD DISP=SHR,DSN=myqual.BSDS02
//CSQP000 DD DISP=SHR,DSN=myqual.PSID00
//CSQP001 DD DISP=SHR,DSN=myqual.PSID01
//CSQP002 DD DISP=SHR,DSN=myqual.PSID02
//CSQP003 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 43. Örnek başlatma yordamı

Daha sonra, kuyruk yöneticinizi şu komutla başlatın:

```
+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 başlatışınızda CSQINP2 tanımlarının CSQ1NEW üyesinden alınması gerekir. Bunu yapmak için, şu komutla kuyruk yöneticisini başlatasınız:

```
+CSQ1 START QMGR ENVPARM('INP2=NEW')
```

CSQ1NORM yerine CSQ1NEW kullanılır. Not: z/OS , simgesel parametrelere (INP2=NEW' de olduğu gibi) ilişkin KEYWORD=value belirteçlerini 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 bu işlemi başlatmaya devam eder. 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 MQ ile ilgili tüm yazma işlemi yanıtı (WTOR) iletilerinin yanıt alınmalıdır; örneğin, günlük istekleri alınıyor. [Şekil 44 sayfa 327](#) 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 44. 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 402.

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 iletilerini elde edin. Örneğin:

```
CSQY009I +CSQ1 ' STOP QMGR' COMMAND ACCEPTED FROM  
USER(userid), STOP MODE(FORCE)  
CSQY002I +CSQ1 QUEUE MANAGER STOPPING
```

Burada *userid* , STOP QMGR komutunu veren kullanıcı kimliğidir ve MODE parametresi, komutta 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 işlevini 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
```

z/OS 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 kullanırsınız. 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 denetimleri 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. z/OS IBM MQ işlemleri ve denetimleri panoları (CSQOREXX), sürüm 7 'den başlayarak eklenen tüm yeni işlevi ve deęiřtirgeleri desteklemeyebilir. Örneęin, konu nesnelere ya da aboneliklerinin doğrudan işlenmesine ilişkin herhangi bir pano yoktur.

Ařaęıdaki desteklenen mekanizmalardan birini kullanarak, dięer panolardan doğrudan olmayan yayınlama/abone olma tanımlarını ve dięer sistem denetimlerini yönetmenize olanak saęlar:

- a. IBM MQ Gezgin
- b. z/OS Konsol
- c. Programlanabilir Komut Biçimi (PCF) iletileri
- d. CSQUTIL ' in COMMAND işlevi

CSQOREXX panolarındaki genel **Command** işleminin, SMDS ile ilgili komutlar da içinde olmak üzere geçerli bir MQSC komutu vermenize olanak tanıdığını 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 saęlayamazsınız. Bunun yerine, kullanıcı kimliğiniz MQCONN ' da BATCH profili için UPDATE yetkisi varsa, **CHCKLOCL** (*REQUIREND* ayarını atlayabilirsiniz). Ek bilgi için [Yerel olarak baęlı 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 ayarlamak](#) başlıklı 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 langletter kullanılacak ulusal dil kitaplıklarını tanımlayan harftir (örneğin, U.S için E). İngilizce). IBM MQ kitaplıkları ISPF kurulumunuza kalıcı olarak kurulduysa, parametreler atlanabilir. Dięer 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.CSQOREXX.* yaratılır.

İşlemlere ve denetim panolarına ilişkin kurallar

IBM MQ karakter dizgileri ve adlarına ilişkin genel kurallarla ilgili olarak [IBM MQ nesnelere 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 yoksayıdır. 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** işlemini 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 nesnelerini, bazı ekstralara 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 nesnelere dir. 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. (Nesneler, MANAGE işlemi kullanılarak silinir.)

Bu kategori aşağıdaki nesnelere den 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 den oluşur:

Kanal	Herhangi bir kanal tipi
Gönderen	Gönderen kanalı
SERVER	Sunucu kanalı
Alıcı	Alıcı kanalı
İsteyen	İstekte bulunanın kanalı
CLUSRCVR	Küme-alıcı kanalı
CLUSSDR	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 sıfırlama
START	Kanal başlatıcıyı ya da dinleyicilerini başlatır
DUR	Kanal başlatıcıyı ya da dinleyicilerini durdurur

İşlemler

Her nesne tipi için gerçekleştirebileceğiniz eylemler, aşağıdaki tabloda gösterilir:

Çizelge 25. IBM MQ nesneleri için geçerli işlemler ve denetim panosu işlemleri

Nesne	Değiştir	Benzer tanımla	Yönet (1)	Liste ya da Görüntü	Süzgeçle Liste	Gerçekleş tir	Başlangıç	Durdur
YAZAR BILGILERI	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
SVRCONN	X	X	X	X	X		X	X
STGCLASS	X	X	X	X	X			
SYSTEM				X		X	X	X

Not:

1. Si1 ve diğ er işlevleri sağ lar.

2. Liste ya da Görüntüle işlemini kullanma

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 yok etme biçiminde tanımlanmış.

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/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 710 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.

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 336
 - 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 45 sayfa 337 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) değiştirebilirsiniz 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, CSQ9022I onay 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 Using the operations and control panels on z/OS

CSQOREXX 'ten görüntülenen ilk denetim panosunu araştırmak için bu konuyu kullanın.

Şekil 45 sayfa 337 , bir pano oturumu 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, 2023. All rights reserved.
Command ==>
F1=Help      F2=Split   F3=Exit     F4=Prompt   F9=SwapNext F10=Messages
F12=Cancel

```

Şekil 45. IBM MQ işlemleri ve denetimi ilk panosu

Bu panodan aşağıdakiler gibi işlemler gerçekleştirebilirsiniz:

- İstedığınız yerel kuyruk yöneticisini ve o kuyruk yöneticisinde, uzak kuyruk yöneticisinde ya da yerel kuyruk yöneticisinde aynı kuyruk paylaşım grubundaki başka bir kuyruk yöneticisinde verilen komutların olmasını isteyip istemediğinizi seçin. Kuyruk yöneticisi adını değiştirmeniz gerekirse, bu adı değiştirmeniz gerekir.
- **Eylem** alanına uygun sayıyı yazarak gerçekleştirmek istediğiniz işlemi seçin.
- Çalışmak istediğiniz nesne tipini belirtin. Nesne tipleriyle ilgili yardım almak için F1 tuşuna basın; bunların ne olduğundan emin değilseniz.
- Çalışmak istediğiniz nesne tipine ilişkin yok etmeyi belirtin.
- Belirtilen tipteki nesnelerin listesini görüntüler. **Ad** alanına bir yıldız işareti (*) girin ve işlem kuyruğu yöneticisinde önceden tanımlanmış nesnelerin (belirtilen tipte) bir listesini görüntülemek için **Giriş** tuşuna basın. Daha sonra, sırayla çalışmak üzere bir ya da daha çok nesne seçebilirsiniz. Listedeki tüm işlemler kullanılabilir.

Not: Görüntülenen nesnelerin bir listesi ile sonuçlanan seçimleri yapmanız ve daha sonra, bu listeden çalışmak için önerilirsiniz. Tüm nesne tipleri için izin verilen için **Görüntüle** işlemini kullanın.

z/OS z/OSüzerinde 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 MQkomutlarına 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 çizmeye 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 işleme 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 tiplerine 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 tanımlamayı 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 nesneleri.
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 340](#) 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. İstedığınız 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 değiştiremeyeniz 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 basarsanız, 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 340.](#)
- Aşamalı bir yaklaşım kullanıyorsanız, bkz. [“Aşamalı bir yaklaşımı kullanarak kümeli gönderici kanallarınızı değiştirme” sayfa 341.](#)

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) için denetim erişimi

```
ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channelname
```

- Bu öğe için CLASSE (MQQUEUE) erişimi güncelleme

```
ssid.SYSTEM.CLUSTER.TRANSMIT.channelname
```

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 özneliğini 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 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 özneliği 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:

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 21.](#)

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.

DynamicQName

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 348](#) başlıklı konuya bakın. Komut sunucusu ayrıca, kanal başlatıcı ve kuyruk paylaşım grubu komutlarını, bunların yayımlandıkları 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 yanıtı 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

Yanıtlama kuyruğunuzun 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 açıklığı 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 .

z/OS 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 348.

Bir MQGET çağrısının, yanıt ileti oluşturulmasını bekleyeceği bir zaman aralığı belirtebilirsiniz. Bir yanıt almazseniz, “Yanıt alamazsanız” sayfa 349 başlıklı konuda başlayan denetim listesini kullanın.

MQGET çağrısını kullanmak için:

1. Bu deęiřtirgeleri 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 *Format* 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.

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 yer alıyor.

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.

z/OS 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 kuyukları doğru olarak tanımlıyor mu?
- Bu kuyuklara MQOPEN çağrıları başarılı oldu mu?
- Hem sistem komut girişi, hem de yanıt kuyukları, MQPUT ve MQGET çağrıları için etkinleştirilmiştir.
- Kuyuklarınızın MAXDEPTH ve MAXMSGL özniteliklerinin artırımı olduğuna 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.

z/OS 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 bakınız: [User ID for command security and command resource security](#) ve [“z/OSüzerinde kuyruk yöneticisi komutları yayınlıyor” sayfa 323](#).

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 iletileri 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                                ) TYPE(
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 özneliği DISABLE değerine ayarlanmış bir yerel kuyruktır.
- A2 , PUT özneliği 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 iletisi döndürülür:

```
CSQN205I  COUNT=    3, RETURN=00000000, REASON=00000000
CSQM407I +CSQ1 NAMELIST(N1
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
SQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQM134I !MQ26 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQ9022I !MQ26 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQ9022I !MQ25 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'STOP CHL' command for CMDSCOPE(*) normal completion

```

Bu iletiler, komutun 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 bağlaşımlı tesisi yapılarını yönetme.

IBM MQ for z/OS kullanırken tamamlamanız gereken farklı yönetim görevlerine ilişkin ayrıntılar için aşağıdaki bağlantıları kullanın:

- [“Günlüklerin yönetilmesi” sayfa 355](#)
- [“Önyükleme veri kümesinin yönetilmesi \(BSDS\)” sayfa 363](#)
- [“Sayfa kümelerinin yönetilmesi” sayfa 371](#)
- [“Sayfa kümelerini nasıl yedeklemeniz ve kurtarmanız” sayfa 377](#)
- [“CSQUTIL kullanarak kuyrukları yedekleme ve geri yükleme” sayfa 381](#)

- [“Arabellek havuzlarının yönetilmesi” sayfa 381](#)
- [“Managing queue sharing groups and shared queues on z/OS” sayfa 383](#)

İlgili kavramlar

[“YönetmeIBM MQ for z/OS” sayfa 311](#)

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/OSkomutlarına komut verilmesi” sayfa 312](#)

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 392](#)

IBM MQtarafından kullanılan kurtarma ve yeniden başlatma mekanizmalarını anlamak için bu konuyu kullanın.

İlgili başvurular

[“IBM MQ for z/OS yardımcı programları” sayfa 320](#)

IBM MQ for z/OS , sistem denetimi için kullanabilemeniz için kullanabileceğiniz bir dizi yardımcı program sağlar.

İlgili bilgiler

[IBM MQ for z/OS kavramlar](#)

[z/OSüzerinde IBM MQ ortamınızın planlanması](#)

[z/OSüzerinde kuyruk yöneticilerinin yapılandırılması](#)

[Programlanabilir komut biçimleri başvurusu](#)

[MQSC başvurusu](#)

[IBM MQ for z/OS yardımcı programlarının kullanılması](#)

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 işleç, ARCHIVE LOG komutunu kullanarak 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](#) komutunun MODE (QUIESCE) seçeneği vardır. Bu seçenikle, IBM MQ işleri ve kullanıcıları bir kesinleşme noktasından sonra susturulmuş olur ve sonuçta elde edilen tutarlılık noktası yüklenmeden önce geçerli etkin günlüğe kaydedilir.

Site kurtarma işlemi 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)
```

Bir TIME parametresi belirtmeden [ARCHIVE LOG](#) komutunu yayınladığınızda, susturma süresi, varsayılan olarak CSQ6ARVP makrosunun QUIESCE parametresinin değerine ayarlanır. ARCHIVE LOG MODE

(QUIESCE) komutunun tamamlanması için gereken süre belirtilenden daha azsa, komut başarıyla tamamlanır; tersi durumda, zaman süresi dolduğunda komut başarısız olur. Zaman aralığını, TIME seçeneğini kullanarak açık bir şekilde belirtebilirsiniz; örneğin:

```
ARCHIVE LOG MODE(QUIESCE) TIME(60)
```

Bu komut, ARCHIVE LOG işlemi gerçekleşmeden önce 60 saniyeye kadar susturulmuş bir dönemi belirtir.

Dikkat: Zaman seçeneğinin kritik olduğu zaman seçeneğinin kullanılması, IBM MQ kaynaklarını kullanan tüm işler ve kullanıcılar için IBM MQ kullanılabilirliğini önemli ölçüde bozabilir.

Komut varsaylan değer 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.

DISPLAY LOG komutundaki çıkış, susturulmuş durumdan bir susturma işlemi olduğunu göstermek için CSQV400I iletisini kullanır. Örneğin:

```
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 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
```



Uyarı: zHyperWrite is not enabled in IBM MQ 9.0 so *Hayır* is the only value permitted.

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 yük boşaltma işleminin başlatıldığını belirten CSQJ311I iletisi verir.

Susturma süresi sona ermeden güncellemeler durdurulamazsa, IBM MQ iletisi CSQJ317Ive 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 kümesiye, ARCHIVE LOG yayınlanırsa, komut işlenmez ve IBM MQ aşağıdaki iletiyi yayınlar:

```
CSQJ319I - csect-name CURRENT ACTIVE LOG DATA SET IS THE LAST
AVAILABLE ACTIVE LOG DATA SET. ARCHIVE LOG PROCESSING
WILL BE TERMINATED
```

Başka bir ARCHIVE LOG komutu devam ederken [ARCHIVE LOG](#) komutu verilirse, yeni komut işlenmez ve IBM MQ aşağıdaki iletiyi yayınlar:

```
CSQJ318I - ARCHIVE LOG COMMAND ALREADY IN PROGRESS
```

Arşivleme sırasında yayınlanan iletilerle ilgili bilgi için bkz. [IBM MQ for z/OSiletileri](#).

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 işlemi iptal edebilir ve ARCHIVE LOG CANCEL OFFLOAD komutunu kullanarak bu işlemi yeniden başlatabilirsiniz. Bu komut, devam etmekte olan tüm yük boşaltma işlemlerini iptal eder ve arşiv işlemi 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.

z/OS Arşivleme ve günlüğe kaydetme denetimi

Sıkıştırma, yazdırma, arşivleme, kurtarma ve günlüğe kaydetme işlemi, CSQ6LOGP, CSQ6ARVPve CSQ6SYSP makrolarını kullanarak denetleyebilirsiniz. Özel nesnelere yapılan değişikliklerin yalnızca IBM MQgünlüklerinde 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. [Görev 17: Sistem parametre modülünüzü uyarlayın](#) .

Bir kuyruk yöneticisi IBM MQ MQSC SET LOG, SET SYSTEM ve SET ARCHIVE komutları kullanılarak çalışırken bu parametrelerden bazıları değiştirilebilir. Bunlar [Çizelge 26 sayfa 357](#) içinde gösterilir:

<i>Çizelge 26. Kuyruk yöneticisi çalışırken değiştirilebilen parametreleri arşivleme ve günlüğe kaydetme</i>	
SET komutu	Parametreler
LOG	WRTHRSH, MAXARCH, DEALCT, MAXRTU, COMPLOG
arşiv	Tümü

Çizelge 26. Kuyruk yöneticisi çalışırken değiştirilebilen parametreleri arşivleme ve günlüğe kaydetme (devamı var)

SET komutu	Parametreler
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 çağırılıyor. Bakınız [“IBM MQ Programlanabilir Komut Biçimlerine Giriş” sayfa 20](#)
- 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 alikonacağı (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 ve veri kümesi manyetik bant yönetim sistemi tarafından çizildikten sonra, bir arşiv günlüğü veri kümesi hakkındaki bilgiler BSDS ' de hala yer alabiliyor. 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 365 içinde açıklandığı şekilde güncelleyebilirsiniz. Güncelleme şart değil. Eski arşiv günlükleri BSDS ' de 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 tür 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 367.

Ayrıca MAXARCH değerini artırmayı düşünün. Bilgi için bkz. CSQ6LOGP komutunu 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 teypteysel, 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 367 başlıklı konuya bakın.

z/OS **Günlüğün yıldırıcı etkisi**

Uzun süren işlemler, günlük veri kümelerine yayılan iş günlüğü kayıtlarına neden olabilir. IBM MQ bu senaryoya, günlük verilerinin alıkonacağı 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 orijinaleri 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 alıkonmanız 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, 3 parça kurtarma bilgisi sağlar:

1. İş birimi kurtarma birimi için günlüğün ne kadarı saklanmalıdır
2. sayfa kümelerinin ortam kurtarma işlemi için günlüğün ne kadarının korunması gerekir
3. Bir kuyruk paylaşım grubunda kuyruk yöneticisi için, CF yapılarının ortam kurtarma işlemi için günlüğün ne kadarının korunması gerektiğini

Bunların her biri için, veri kümesi için gereken en eski günlük verilerini eşleştirmek için bir girişimde bulunmanız gerekir. Yeni iş birimleri başlangıç ve durdurma birimi olarak, (1) günlük olarak daha yeni bir konuma taşınmasını bekardık. Hareket etmiyorsa, uzun süredir çalışan UOW iletisi, 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 tanımlanabileceği anlamına gelir. Olağan durumda, bu ya da kuyruk paylaşım grubundaki diğer kuyruk yöneticilerinde BACKUP CFSTRUCT komutlarıyla daha yeni bir konuma geçilir.

z/OS *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 FFFFFFFF (6 baytlık RBU kullanımında) ya da FFFFFFFFFF (8 baytlık RBA 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 'ları kullanımdaysa) ya da [FFFFFC00000000](#) (8 baytlık günlük RBA' ları kullanımdaysa) neden koduyla birlikte sona erer.

6 baytlık günlük RBA 'ları kullanımdaysa, kuyruk yöneticisinin [Daha büyük günlük Görelî Bayt Adresini uygulamayı](#) açıklanan işlemi 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 önceki süreyi 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 çevirirken **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 yöneticilerindeki 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 Görelî Bayt Adresini uygulama” sayfa 362](#)

IBM MQ 8.0öncesinde, IBM MQ for z/OS , günlük içindeki verilerin yerini belirlemek için 6 baytlık bir RBA günlüğü kullandı. IBM MQ 8.0' tan, günlük RBA 8 bayt uzunluğunda olabilir ve günlüğü sıfırlamak zorunda kalmaktan önceki süreyi uzatır.

z/OS Daha büyük günlük Görelî Bayt Adresini uygulama

IBM MQ 8.0öncesinde, IBM MQ for z/OS , günlük içindeki verilerin yerini belirlemek için 6 baytlık bir RBA günlüğü kullandı. IBM MQ 8.0' tan, günlük RBA 8 bayt uzunluğunda olabilir ve günlüğü sıfırlamak zorunda kalmaktan önceki süreyi uzatır.

Bu yeni özelliğın belirtik olarak etkinleřtirilmesi gerekir. 8 baytlık günlük RBA ' yı etkinleřtirmeyi planlarken dikkat edilmesi gereken noktalar için bkz. [Planning to arge the maximum addressable log range](#) .

Tek bir IBM MQ for z/OS kuyruk yöneticisinde 8 baytlık günlük RBA ' yı etkinleřtirmek için bu yönergeleri gösterildiğı sırayla gerçekleştirin:

1. **OPMODE'** u kullanarak IBM MQ 8.0 yeni işlevlerini etkinleřtirin.

CD Kuyruk paylaşım grubundaki kuyruk yöneticilerine ilişkin, toplam kuyruk paylaşım grubu kesintisi yapmanız gerekmez. Her kuyruk yöneticisini sırayla durdurabilirsiniz, OPMODE = (NEWFUNC,800) ya da OPMODE = (NEWFUNC,900) için etkinleřtirmeyi etkinleřtirin ve yeniden başlatın.

CD Kuyruk paylaşım grubundaki tüm kuyruk yöneticileri OPMODE = (NEWFUNC,800) ya da OPMODE = (NEWFUNC,900) ile çalıştırıldıktan sonra, kuyruk paylaşım grubundaki her kuyruk yöneticisi için, tüm kuyruk yöneticileri yeni BSDS ile çalışmaya devam edinceye 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 ' nizin özniteliklerini denetleyin. Değıştirebilecek tek öznitelik, BSDS ' nin büyüklüğüdür.
 - b. Yeni BSDS, geçerli BSDS ' nin içerdığı daha fazla veri içerir, bu nedenle yeni veri kümelerinin yeterli kullanılabilir alan ile ayrıldığından emin olmanız gerekir. Yeni bir BSDS tanımlarken önerilen değerler için [Günlüğe kaydetme ortamınızın planlanması](#) başlıklı konuya ve ilişkili konulara bakın.
3. Kuyruk yöneticisini temizleme işlemini sona erdirin.
 4. Var olan BSDS ' leri yeni BSDS veri kümelerine dönüřtürmek için [BSDS conversion utility \(CSQJUCNV\)](#) komutunu çalıştırın. Bu işlem genellikle birkaç saniye sürer.

Var olan BSDS ' niz bu işlem sırasında değıştirilmeyecek ve başarısız bir dönüřtürme 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, örneğın:

```
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 gerçekleştirileceğı süre içinde başlamalıdır.


Dönüřtürülen BSDS ' ye erişilememesi nedeniyle kuyruk yöneticisi başarıyla yeniden başlatılamazsa, başarısızlığın nedenini saptama girişiminde bulunmayı deneyin, sorunu çözün ve işlemi yeniden deneyin. Gerekliyorsa, yardım almak için IBM destek merkezimize başvurun.

Gerekirse, bu noktada deęişiklik řu noktada yedeklenebilir:

- a. Yürürlükteki BSDS ' ler yeni BSDS olacak řekilde yeniden adlandırılıyor.
- b. Eski BSDS ' ler yürürlükteki BSDS olacak řekilde yeniden adlandırılıyor.
- c. Kuyruk yöneticisi yeniden başlatılıyor.

Kuyruk yöneticisi dönüřtürülen 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 IBM MQ 9.0 kuyruk yöneticisinde 8 baytlık günlük RBA ' yı etkinleřtirmek için, ilk başlatılmadan önce, boş bir sürüm 1 biçimi BSDS yaratmalı ve bir sürüm 2 biçimi BSDS üretmek için BSDS dönüřtürme yardımcı programına giriş olarak kullanmanız gerekir. Bu süreci nasıl gerçekleřtirdiğiniz hakkında bilgi için [Önyükleme ve günlük veri kümelerinin oluşturulması](#) başlıklı konuya bakın.

İlgili bilgiler

[Adreslenebilir günlük aralığı üst sınırını artırmayı planlama](#)

[Daha büyük günlük Görelı Bayt Adresi](#)

[BSDS dönüřtürme yardımcı programı \(CSQJUCNV\)](#)

Önyükleme veri kümesinin yönetilmesi (BSDS)

Önyükleme veri kümesi (BSDS), günlük veri kümelerine ve günlük kayıtlarına gönderme yapmak için kullanılır. BSDS ' yi nasıl inceleyebileceğinizin, deęiřtirebileceğinizin ve kurtarmanın nasıl gerçekleřtirileceğini anlamak için bu konuyu kullanın.

Ek bilgi için [Önyükleme veri kümesibařlıklı konuya bakın](#).

Bu konuda, önyükleme veri kümesinin yönetilmesinde yer alan görevler açıklanmaktadır. Bu kısımları içerir:

- [“BSDS ' nin içerdiği bilgileri bulma” sayfa 363](#)
- [“BSDS ' nin Deęiřtirilmesi” sayfa 365](#)
- [“BSDS ' nin Kurtarılıyor” sayfa 369](#)

BSDS ' nin içerdiği bilgileri bulma

BSDS ' nin içeriğini incelemek için yazdırma günlüğü eřleme yardımcı programını (CSQJU004) kullanabilirsiniz.

Yazdırma günlüğü eřleme yardımcı programı (CSQJU004), BSDS ' de saklanan bilgileri listeleyen bir toplu iř yardımcı programıdır. Çalıştırılmasına ilişkin yönergeler için [Yazdırma günlüğü eřleme yardımcı programı](#) başlıklı konuya bakın.

BSDS řunları içerir:

- [Zaman damgaları](#)
- [Etkin günlük veri kümesi durumu](#)

BSDS ' deki zaman damgaları

Yazdırma günlüğü eřleme yardımcı programının çıkışı, BSDS ' de saklanan çeřitli sistem olaylarının tarih ve saatini kaydetmek için kullanılan zaman damgalarını gösterir.

Ařağıdaki 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 eşliğine 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 eşliğinin ayrıntıları için, [CSQ6LOGPkomutunu kullanma](#) içindeki 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 yapılırsa.

Arşiv günlüğü süresi

Arşiv günlüğü girişinin BSDS ' de yaratıldığı saat (CSQJU003 NEWLOG değeri yapıldığında ya da arşivin kendisi yapılırsa).

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 işlemi çıkışında, son YENIDEN kullanılabilir veri kümesine ilişkin RBA başlangıç RBA değeri, son arşiv günlüğü veri kümesinin başlangıç RBA değerine eşit.

YENİDEN 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 'sından 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 Değiştirilmesi**

IBM MQ bunu otomatik olarak yaptığı için, BSDS ' nin günlüğe kaydetme olaylarının kayıtlarıyla güncellenmesini korumak için özel adımlar atmanız gerekir.

Ancak, aşağıdakilerden herhangi birini yapmak isterseniz, BSDS ' yi değiştirmek isteyebilirsiniz:

- Daha fazla etkin günlük veri kümesi ekleyin.
- Etkin günlük veri kümelerini yeni ayrılan veri kümelerine kopyalar; örneğin, daha büyük etkin günlük ayrımları sağlanırken.
- Günlük verileri kümelerini diğer aygıtlara taşıyın.
- Hasar görmüş bir BSDS ' yi kurtarın.
- Eski arşiv günlüğü veri kümelerini atın.

Değişiklik günlüğü döküm yardımcı programını (CSQJU003) çalıştırarak BSDS ' yi değiştirebilirsiniz. Bu yardımcı programı yalnızca kuyruk yöneticisi etkin değilken çalıştır ya da tutarsız sonuçlar alabilirsiniz. Yardımcı programın işlemi, SYSIN veri kümesindeki deyimler tarafından denetlenir. Bu bölümde çeşitli örnekler gösterilir. Eksiksiz yönergeler için [Değişiklik günlüğü döküm yardımcı programı](#) başlıklı konuya bakın.

Etkin günlük verileri kümesini, IBM MQ etkin günlük veri kümelerini kuyruk yöneticisi başlatma sırasında dışlayıcı (DISP = OLD) olarak ayırdığı için, yalnızca kuyruk yöneticisi etkin değil olduğunda kopyalayabilirsiniz.

z/OS **Etkin günlüklere ilişkin değişiklikler**

BSDS ' yi kullanarak etkin günlüklerin nasıl değiştirileceğini anlamak için bu konuyu kullanın.

Değişiklik günlüğü yardımcı programını kullanarak etkin günlüklere ilişkin BSDS ' deki girişleri ekleyebilir, bunları silebilir ve kaydedebilir. Burada yalnızca örnekler gösterilir; kullanmak istediklerinizle gösterilen veri kümesi adlarını değiştirin. Yardımcı programın ayrıntıları için [Değişiklik günlüğü döküm yardımcı programı](#) başlıklı konuya bakın.

Daha fazla bilgi için aşağıdaki bölümlere bakın:

- [BSDS ' ye kayıt girişi eklenmesi](#)
- [BSDS ' den etkin günlük verileri kümesiyle ilgili bilgileri silme](#)
- [BSDS ' de ayarlanan günlük verileri hakkında bilgi kaydedilmesi](#)
- [Etkin günlük büyüklüğünün artırılması](#)
- [CSQJUFMT kullanımı](#)

BSDS ' ye kayıt girdileri ekleme

Etkin bir günlük "durduruldu" olarak işaretlendiyse, günlüğe kaydetme için yeniden kullanılmaz; ancak, okuma için kullanılmaya devam eder. Yeni etkin günlük veri kümelerini tanımlamak için erişim yöntemi hizmetlerini kullanın ve daha sonra, yeni veri kümelerini BSDS ' ye kaydetmek için günlük döküm alma yardımcı programını kullanın. Örneğin, şunları kullanın:

```
NEWLOG DSNAME=MQM111.LOGCOPY1.DS10,COPY1
NEWLOG DSNAME=MQM111.LOGCOPY2.DS10,COPY2
```

Eski bir etkin günlük verilerinin içeriğini yenisine kopyalyorsanız, NEWLOG işlevindeki RBA aralığını ve başlangıç ve bitiş zaman damgalarını da verebilirsiniz.

BSDS ' den etkin günlük verileri kümesiyle ilgili bilgileri silme

BSDS ' den etkin bir günlük veri kümesine ilişkin bilgileri silmek için aşağıdaki özellikleri 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.
- Etkin bir günlük verilerinin içeriğini başka bir veri kümesine kopyalyorsunuz.
- 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. IBM MQ , etkin olduğunda tüm etkin günlük veri kümelerini özel kullanım için 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 olmaksızın 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 zarar görse, kuyruk yöneticisi durur ve BSDS veri kümeleri onarılmaya kadar yeniden başlatılmaz. 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 369. Do not copy any data to the second replacement BSDS, you do that in step “5” sayfa 371.

- 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 gerekirse, 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 'yı 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 yapacağınızla ilgili 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 kümelerinin yönetilmesi

Bir kuyruk yöneticisiyle ilişkili sayfa kümelerinin nasıl yönetileceğini anlamak için bu konuyu kullanın.

Bu konuda, bir kuyruk yöneticisiyle ilişkili sayfa kümeleri nasıl ekleneceği, kopyalanacak ve genel olarak yönetileceği açıklanır. Bu kısımları içerir:

- [“Sayfa kümeleri için üst düzey niteleyici \(HLQ\) nasıl değiştirileceği” sayfa 371](#)
- [“Kuyruk yöneticisine sayfa kümesi eklenmesi” sayfa 372](#)
- [“Sayfa kümelerinden biri dolunca ne yapmanız gerekir” sayfa 372](#)
- [“Sayfa kümelerinde yükleme nasıl dengelenir” sayfa 372](#)
- [Sayfa kümesi büyüklüğünün nasıl artırılacağı](#)
- [“Sayfa kümesinin nasıl azaltılacağı” sayfa 376](#)
- [“Bir sayfa kümesini nasıl yeniden tanıttığını” sayfa 377](#)
- [“Sayfa kümelerini nasıl yedeklemeniz ve kurtarmanız” sayfa 377](#)
- [“Sayfa kümelerinin nasıl silineceği” sayfa 381](#)
- [“CSQUTIL kullanarak kuyrukları yedekleme ve geri yükleme” sayfa 381](#)

Sayfa kümelerinin, depolama sınıflarının, arabelleklerin ve arabellek havuzlarının bir açıklaması ve geçerli olan bazı performans konuları için [Sayfa kümeleri](#) başlıklı konuya bakın.

Sayfa kümeleri için üst düzey niteleyici (HLQ) nasıl değiştirileceği

Bu görev, sayfa kümelerine ilişkin HLQ ' un nasıl değiştirileceği hakkında bilgi verir. Bu görevi gerçekleştirmek için aşağıdakileri yapın:

1. Yeni HLQ sayfa kümelerini tanımlayın.
2. Boyut tahsisi eski sayfa kümeleriyle aynıysa, REPRO kullanarak var olan sayfa kümesini boş yeni HLQ sayfa kümelerine kopyalayın. Sayfa kümelerinin büyüklüğünü artırıyorsanız, hedef sayfa kümesini biçimlendirmek için CSQUTIL FORMAT işlevini kullanın. Daha fazla bilgi için bkz. [Sayfa kümelerini biçimlendirme \(FORMAT\)](#).
3. Kaynak sayfadaki tüm iletileri hedef sayfa kümesine kopyalamak için CSQUTIL ' in COPYPAGE işlevini kullanın. Ek bilgi için [Sayfa kümesinin \(COPYPAGE\) genişletmesi](#) başlıklı konuya bakın.
4. Kuyruk yöneticisi yordamında CSQP00xx DD deyimini değiştirerek, yeni HLQ sayfa kümelerini işaret edecek şekilde değiştirin.

Kuyruk yöneticisini yeniden başlatın ve sayfa kümelerinde yapılan değişiklikleri doğrulayın.

Kuyruk yöneticisine sayfa kümesi eklenmesi

Bu açıklama, çalışmakta olan bir kuyruk yöneticiye sahip olduğunuz varsayılır. Örneğin, kuyruk yöneticinizin yeni kuyruklar kullanarak yeni uygulamalarla baş etmesi 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 \(FORMAT\)](#).

Kullanımda olan herhangi bir sayfa kümesini biçimlendirmeye özen göstermeyin, bu sizin planladığınız gibi değilse. Böyle bir durumda, FORMAT yardımcı programı işlevinin FORCE seçeneğini kullanın.

2. Bir arabellek havuzuyla sayfa kümesini ilişkilendirmek için DSN seçeneğiyle DEFE PSID komutunu kullanın.
3. DEFINE STGCLASS komutlarını vererek, sayfanız için uygun depolama sınıfı tanımlarını ekleyin.
4. İsteğe bağlı olarak, kuyruk yöneticinizin nasıl yapılandırıldığı belgeyi belgelemek için:
 - a. Kuyruk yöneticinizin başlatılan görev yordamınıza yeni sayfa kümesi eklemenizi sağlar.
 - b. Yeni sayfa kümesi için CSQINP1 kullanıma hazırlama veri kümeniz için bir tanım ekleyin.
 - c. Add a definition for the new storage class to your CSQ4INYP initialization data set member.

DEFE PSID ve DEFINE STGCLASS komutlarının ayrıntıları için bakınız: [DEFINT PSID](#) and [DEFINE STGCLASS](#).

Sayfa kümelerinden biri dolunca ne yapmanız gerekir

You can find out about the utilization of page sets by using the IBM MQ command DISPLAY USAGE. Örneğin, komut:

```
DISPLAY USAGE PSID(03)
```

Sayfa kümesinin geçerli durumunu görüntüler 03. Bu sayfa, bu sayfa kümesinin kaç boş sayfa olduğunu gösterir.

Sayfa kümeleriniz için ikincil kapsamlar tanımladıysanız, bunlar her dolduruldukları her defasında dinamik olarak genişletilirler. Sonunda, tüm ikincil kapsamlar kullanılır ya da başka bir disk alanı kullanılabilir değildir. Bu gerçekleşirse, bir uygulama MQRC_STORAGE_MEDIUM_FULL dönüş kodunu alır.

Bir uygulama bir MQI çağrısından MQRC_STORAGE_MEDIUM_FULL ' un dönüş kodunu alırsa, bu, sayfa kümesinde yeterli yer kalmadığını gösteren açık bir göstergedir. Sorun devam ederse ya da yeniden onaylama olasıysa, sorunu çözmek için bir şeyler yapmanız gerekir.

Bu soruna çeşitli yollarla yaklaşılabilir:

- Kuyrukları bir sayfa kümesinden diğerine taşıyarak sayfa kümeleri arasındaki yükü dengeler.
- Sayfa kümesini genişletin. Yönergeler için bkz. [“Sayfa kümesinin büyüklüğünün artırılması” sayfa 374](#) .
- Sayfa kümesini, 4 GB 'nin ötesine, 64 GB' lik bir üst sınır değerine genişletmek için yeniden tanımlayın. Yönergeler için [4 GB ' den büyük olacak şekilde bir sayfa kümesinin tanımlanması](#) başlıklı konuya bakın.

Sayfa kümelerinde yükleme nasıl dengelenir

Sayfa kümelerinde yük dengelemesi, bir ya da daha çok kuyrukla ilişkilendirilmiş iletilerin bir sayfa kümesinden diğerine, daha az kullanılanla, sayfa kümesine taşınması anlamına gelir. Sayfa kümesini genişletmek için kullanışlı değilse bu tekniği kullanın.

Hangi kuyrukların sayfa kümesini kullandığını belirlemek için uygun IBM MQ komutlarını kullanın. Örneğin, hangi kuyrukların sayfa kümesi 02 ile eşlendiğini öğrenmek için önce şu komutu kullanarak, hangi depolama sınıflarının sayfa kümesi 02 'ye eşlendiğini öğrenin:

```
DISPLAY STGCLASS(*) PSID(02)
```

Daha sonra hangi kuyrukların hangi depolama sınıfını kullanabileceğini öğrenmek için aşağıdaki komutu kullanın:

```
DISPLAY QUEUE(*) TYPE(QLOCAL) STGCLASS
```

Paylaşılmayan Kuyruğun Taşınması

Kuyrukları ve bunların iletilerini bir sayfa kümesinden diğerine taşımak için, MQSC MOVE QLOCAL komutunu kullanın (QLOCAL ' I TAŞI içinde açıklanmaktadır). Yeni bir sayfa kümesine taşımak istediğiniz kuyruğu ya da kuyrukları tanımladığınızda, bu kuyrukların her biri için aşağıdaki yordamı izleyin:

1. Taşımak istediğiniz kuyruğun herhangi bir uygulama (DISPLAY QSTATUS komutundan IPPROCS ve OPPROCS değerleri sıfır olduğundan) ve kesinleştirilmemiş iletileri olmadığını doğrulayın (DISPLAY QSTATUS komutundan UNCOM değeri NO.).

Not: Bu durumun devam etmesini sağlamanın tek yolu, kuyruğun güvenlik yetkisini geçici olarak değiştirmesidir. Ek bilgi için [Kuyruk güvenliği için tanımlar başlıklı konuya](#) bakın.

Bunu yapamazsanız, PUT ayarı (DISABLE) gibi ihtiyati adımlara rağmen uygulamalar kuyrukta kullanmaya başlarsa, bu yordamda sonraki aşamalar başarısız olabilir. Ancak, iletiler bu yordam tarafından hiçbir zaman kaybolamaz.

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. Şu komutu kullanarak, taşınmakta olan kuyrukla aynı özniteliklere sahip bir geçici kuyruk tanımlayın:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED)
```

Not: Bu geçici kuyruk önceki bir çalışmadan önceden varsa, tanımlama yapmadan önce bu kuyruğu 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 kümesine eşlenen yeni bir depolama sınıfı tanımlayın, örneğin:

```
DEFINE STGCLASS(NEW) PSID(nn)
```

Yeni depolama sınıfı tanımlamasını, sonraki kuyruk yöneticisi yeniden başlatmaya hazır CSQINP2 veri kümelerine ekleyin.

7. Taşımakta olduğunuz kuyruğu, depolama sınıfı özneliğini değiştirerek yeniden tanımlayın:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_Queue) STGCLASS(NEW)
```

Kuyruk yeniden tanımlandığında, "3" sayfa 373 adımıyla yaratılan geçici kuyruğa dayalı olur.

8. Komutu kullanarak, iletileri yeni kuyruğa geri taşıyın:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

9. "3" sayfa 373 adımıyla yaratılan kuyruk artık gerekli değil. Silmek için aşağıdaki komutu kullanın:

```
DELETE QL(TEMP_Queue)
```

10. Taşınmakta olan kuyruk CSQINP2 veri kümelerinde tanımlandıysa, uygun DEFINE QLOCAL komutunun STGCLASS özneliğini, CSQINP2 veri kümelerinde değiştirin. Var olan kuyruk tanımlamasının yerine yenisi konması için REPLACE anahtar sözcüğünü ekleyin.

Şekil 46 sayfa 374 , yük dengeleme işinden bir alma işlemi 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(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 46. Bir sayfa kümesi için yük dengeleme işinden çıkarma

Sayfa kümesinin büyüklüğünün artırılması

Başlangıçta 4 GB 'den daha büyük bir sayfa kümesi ayırabilirsiniz, bkz. [4 GB' den büyük olacak şekilde bir sayfa kümesi tanımlama](#)

Bir sayfa kümesi, EXPAND (SYSTEM) ya da EXPAND (USER) belirtilerek tam olarak genişletilecek şekilde tanımlanabilirler. Sayfa ayarınız EXPAND (NONE) ile tanımlandıysa, bunu iki şekilde genişletebilirsiniz:

- Tanımlamayı, otomatik genişletmeye izin verecek şekilde değiştirin. Bkz. [Bir sayfa kümesinin otomatik genişletmeye izin verecek şekilde değiştirilmesi](#)
- Yeni, daha büyük bir sayfa kümesi yaratın ve eski sayfa kümesinden yeni bir sayfa kümesini yeni bir sayfa olarak kopyalayın. Bkz. [İletilerin yeni, daha büyük bir sayfa kümesine taşınması](#)

4 GB ' den büyük olacak şekilde bir sayfa kümesi tanımlama

IBM MQ , 64 GB 'ye kadar olan bir sayfayı kullanarak, veri kümesi VSAM' ye 'genişletilmiş adreslenebilirlik' olarak tanımlıdır. Ek adreslenebilirlik, bir SMS veri sınıfı tarafından verilen bir öznitelige sahip bir öznitedir. Aşağıdaki örnek JCL 'de gösterilen örnekte,' EXTENDED 'yönetim sınıfı' Genişletilmiş adreslenebilirlik 'ile SMS' ye tanımlıdır. Var olan sayfa kümeniz şu anda genişletilmiş adreslemeye sahip olarak tanımlanmadıysa, genişletilmiş bir adreslenebilirlik biçimi veri kümesine geçiş yapmak için aşağıdaki yöntemi kullanın.

1. Kuyruk yöneticisini durdurun.
2. Var olan sayfa kümesini yeniden adlandırmak için Erişim Yöntemi Hizmetleri olanağını kullanın.
3. Bir hedef sayfa kümesi, var olan sayfa kümesiyle aynı boyutu, ancak DATACLAS (EXTENDED) ile tanımlayın.

Not: Genişletilmiş biçimdeki veri kümeleri SMS ile yönetilmelidir. Bunlar, VSAM veri kümeleri için genişletilmiş biçim isteğinde bulunan mekanizmalardır:

- Gerekli ya da tercih edilen bir DSNTYPE değeri EXT, alt parametre R ya da P olan bir veri sınıfı kullanılıyor.
- DD deyiminde DSNTYPE=EXTREQ (genişletilmiş biçim gereklidir) ya da DSNTYPE=EXTPREF (genişletilmiş biçim tercih edilir) kodlamasını kullanın.
- DD deyiminde LIKE= parametresini kodlamak, var olan genişletilmiş bir veri kümesine başvuruda bulunmak için kodlayın.

Daha fazla bilgi için bakınız: [Restrictions on Defining Extended-Format Data Sets.](#)

4. Kaynak sayfadaki tüm iletileri hedef sayfa kümesine kopyalamak için CSQUTIL ' in COPYPAGE işlevini kullanın. Ek ayrıntılar için [Sayfa kümesinin \(COPYPAGE\) genişletmesi](#) başlıklı konuya bakın.
5. Kuyruk yöneticisini yeniden başlatın.
6. Sistem genişlemesini kullanmak üzere sayfa kümesini değiştirerek, yürürlükteki ayırmasının ötesinde büyümeye devam etmesini sağlar.

Aşağıdaki JCL, örnek Access Method Services 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))
/*
```

Bir sayfa kümesinin otomatik genişletmeye izin verecek şekilde değiştirilmesi

ALTER PSID komutunu EXPAND (USER) ya da EXPAND (SYSTEM) seçenekleri ile kullanın. Sayfa kümelerinin genişletilmesine ilişkin genel bilgi için bkz. [ALTER PSID](#) ve [Expanding a page set \(COPY PAGE\)](#) .

İletilerin yeni, daha büyük bir sayfa kümesine taşınması

Bu teknik, kuyruk yöneticisinin durdurulmasını ve yeniden başlatılmasını içerir. Bu işlem, yeniden başlatma sırasında paylaşılan kuyruklarda bulunmayan kalıcı olmayan iletileri siler. Silinmek

istemediğiniz kalıcı olmayan iletiler varsa, bunun yerine yük dengelemeyi kullanın. Daha fazla ayrıntı için bkz. “Sayfa kümelerinde yükleme nasıl dengelenir” sayfa 372. Bu açıklamada, genişletmek istediğiniz sayfa kümesi *kaynak* sayfa kümesi olarak adlandırılır; yeni, daha büyük sayfa kümesi *hedef* sayfa kümesi olarak adlandırılır.

Aşağıdaki adımları izleyin:

1. Kuyruk yöneticisini durdurun.
2. Hedef sayfa kümesini, ikincil kapsam değeri daha büyük olan kaynak sayfa kümesinden daha büyük olduğundan emin olarak tanımlayın.
3. Hedef sayfa kümesini biçimlendirmek için CSQUTIL FORMAT işlevini kullanın. Daha fazla ayrıntı için [Sayfa kümelerini biçimlendirme \(FORMAT\)](#) başlıklı konuya bakın.
4. Kaynak sayfadaki tüm iletileri hedef sayfa kümesine kopyalamak için CSQUTIL ' in COPYPAGE işlevini kullanın. Ek ayrıntılar için [Sayfa kümesinin \(COPYPAGE\) genişletmesi](#) başlıklı konuya bakın.
5. Aşağıdakilerden birini yaparak hedef sayfa kümesini kullanarak kuyruk yöneticisini yeniden başlatın:
 - Kuyruk yöneticisini değiştirerek, hedef sayfa kümesine gönderme yapmak için görev yordamını başlatmış olun.
 - Kaynak sayfa kümesini silmek için Access Method Services olanağını kullanın ve ardından hedef sayfa kümesini, kaynak sayfa kümesinin adıyla aynı adı vererek yeniden adlandırın.

Dikkat:

Herhangi bir IBM MQ sayfa kümesini silmeden önce, gerekli yedek kopyalarının yapıldığından emin olun.

Sayfa kümesinin nasıl azaltılacağı

Prevent all users, other than the IBM MQ administrator, from using the queue manager. Örneğin, erişim güvenliği ayarlarını değiştirerek.

Çoğunlukla boş olan büyük bir sayfa kümeniz varsa (DISPLAY USAGE komutuna göre gösterildiği gibi), büyüklüğünü azaltmak isteyebilirsiniz. Bunu yapmak için yordam, CSQUTIL (bkz. IBM MQ yardımcı programı) içindeki COPY, FORMAT ve LOAD işlevlerinin kullanılmasını içerir. Bu yordam, bu sayfa kümesinin büyüklüğünü azaltmak için kullanışlı olmadığı için, sayfa kümesi sıfır (0) için çalışmaz; bunu yapabilmeyen tek yolu kuyruk yöneticinizi yeniden başlatarak (bkz. “Kuyruk yöneticisi yeniden kullanıma hazırlanıyor” sayfa 400) kullanıma hazırlanır. Bu yordamın önkoşulu, tüm UOWs ' lerin tamamlanacak ve sayfa kümelerinin tutarlı olması için tüm kullanıcıları sistemden kaldırmaya çalışmadır.

1. Kuyruk yöneticisini durdurmak için STOP QMGR komutunu QUIESCE ya da FORCE öznitelikle birlikte kullanın.
2. Büyük sayfa kümesinden tüm ileti verilerini kopyalamak ve sıralı bir veri kümesinde saklamak için, PSID seçeneği ile CSCHEIL ' in SKOPI işlevini çalıştırın.
3. Büyük sayfa kümesini değiştirmek için yeni bir küçük sayfa kümesi veri kümesi tanımlayın.
4. Run the FORMAT TYPE(NEW) function of CSQUTIL against the page set that you created in step “3” [sayfa 376](#).
5. Restart the queue manager using the page set created in step “3” [sayfa 376](#).
6. “2” [sayfa 376](#) adımı sırasında kaydedilen tüm iletileri geri yüklemek için CSQUTIL ' in LOAD işlevini çalıştırın.
7. Tüm kullanıcıların kuyruk yöneticisine erişmesine izin verir.
8. Eski büyük sayfa kümesini silin.

Bir sayfa kümesini nasıl yeniden tanıtacağını

Belirli senaryolarda, kuyruk yöneticisine yeniden çevrimiçi eski bir sayfayı çevrimiçi duruma getirebilmek yararlı olur. Belirli bir işlem alınmadığı sürece, eski sayfa kümesi çevrimiçi duruma getirildiğinde kuyruk yöneticisi, sayfa kümesinin kendisini ve denetim noktası kayıtlarında saklanan sayfa ayarlama kurtarma RBA 'nın eski olduğunu algılar ve bu nedenle, sayfanın güncel olarak güncel duruma getirilmesi için ortam kurtarma işlemini otomatik olarak başlatır.

Bu tür ortam kurtarma işlemi yalnızca, kuyruk yöneticisi yeniden başlatıldığında gerçekleştirilebilir ve özellikle manyetik bantta tutulan arşiv günlükleri okunmalıdır. Ancak, normalde bu durumda sayfa kümesi, müdahalede bulunan dönem için çevrimdışıdır ve bu nedenle, günlük, sayfa kümesi kurtarması için uygun bilgi içermez.

Aşağıdaki üç seçenek kullanılabilir:

Tam ortam kurtarma işleminin gerçekleştirilmesine izin verin.

1. Kuyruk yöneticisini durdurun.
2. Kuyruk yöneticisinin hem başlatılan görev yordamında hem de CSQINP1 kullanıma hazırlama veri kümesinde ayarlanan sayfa için tanımların kullanılabilir olduğundan emin olun.
3. Kuyruk yöneticisini yeniden başlatın.

Sayfa kümesindeki iletilerin yok edilmesine izin verin.

Bu seçenek, bir sayfa kümesinin uzun süredir çevrimdışı olduğu (örneğin, birkaç ay) ve artık bunu farklı bir amaç için yeniden kullanma kararı aldıysa yararlı olur.

1. CSQUTIL biçim işlevini TYPE (NEW) seçeneğiyle kullanarak sayfa kümesini biçimlendirin.
2. Kuyruk yöneticisine ilişkin başlatılan görev yordamlarına ve CSQINP1 kullanıma hazırlama veri kümesi için hem başlatılan görev yordamlarına hem de bu sayfaya ilişkin tanımları ekleyin.
3. Kuyruk yöneticisini yeniden başlatın.

Biçimlendirme için TYPE (NEW) seçeneğinin kullanılması, sayfa kümesinin yürürlükteki içeriğini temizler ve kuyruk yöneticisine sayfa kümesiyle ilgili denetim noktasındaki tüm geçmiş bilgilerini yoksaymasını bildirir.

Ortam kurtarma sürecinden kaçınmak için sayfa kümesini çevrimiçi duruma getirin.

Bu tekniği yalnızca, kuyruk yöneticisinin temiz bir şekilde kapatıldığından bu yana sayfa kümesinin çevrimdışı duruma geçtiğinden eminseniz kullanın. Bu seçenek, genellikle kuyruk yöneticisi başlatıldığında yedekleme işlemi gibi işletim sorunları nedeniyle, sayfa kümesinin kısa bir süre için çevrimdışı olduğu durumlarda uygundur.

1. CSQUTIL biçim işlevini TYPE (REPLACE) seçeneğiyle kullanarak sayfa kümesini biçimlendirin.
2. DSN seçeneğiyle DEFE PSID komutunu kullanarak, yeniden kuyruk yöneticisine sayfa kümesini devingen olarak ekleyin ya da bir kuyruk yöneticisi yeniden başlatıldığında eklenmesine izin verin.

Biçimlendirme için TYPE (REPLACE) seçeneğinin kullanılması, sayfa kümesinin kuyruk yöneticisi tarafından cleanly tarafından kapatıldığını denetler ve ortam kurtarma işleminin gerçekleştirilmemesi için bunu işaretler. Sayfa kümesinin içeriğinde başka hiçbir değişiklik yapılmayacak.

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 378](#)
- [“Sayfa kümeleri yedekleniyor” sayfa 379](#)
- [“Sayfa kümeleri kurtarılıyor” sayfa 380](#)
- [Sayfa kümelerinin nasıl silineceği](#)

Paylaşılan kaynaklar için kurtarma noktasının nasıl oluşturulacağı hakkında bilgi için bkz. [“Paylaşılan kuyrukların kurtarılıyor” sayfa 386.](#)

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 şekilde devam edebilirsiniz.

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 (ek bilgi için [CSQI024I](#) ve [CSQI025I](#) başlıklı konuya bakın). 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 379](#)).

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 kullanılabileceği 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 379](#)).
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 sayı 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') ALLOCgerçekleştirdiyse, 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, sayfa kümelerinden ya da günlük veri kümelerinden herhangi biri kullanılabilir durumda değilse bu tür kurtarma işlemi yapılamaz. 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 379’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 ilişkili arabellek havuzunu olabildiğince büyük bir şekilde kurtarılması için 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 (susturma işlevini kullanarak) 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 382](#)
- [“Arabellek Havuzunun Silinmesi” sayfa 382](#)

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. [z/OSüzerinde planlama](#) .

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 başařsız olur.

Sayfa kümelerinin nasıl silineceği hakkında bilgi için bkz. [“Sayfa kümelerinin nasıl silineceği” sayfa 381](#) .

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 gruplarının yönetilmesi” sayfa 383](#)
- [“Paylaşılan kuyrukların yönetilmesi” sayfa 386](#)
- [“Grup nesnelere yönetilmesi” sayfa 391](#)
- [“Bağlaşım tesisinin yönetilmesi” sayfa 391](#)

z/OS Kuyruk paylaşım gruplarının yönetilmesi

Bir kuyruk yöneticisini bir kuyruk paylaşım grubuna (QSG) ekleyebilir ya da bir kuyruk paylaşım grubuna kaldırabilir ve ilişkili Db2 tablolarını yönetebilirsiniz.

Bu konunun aşağıdaki görevlerle ilgili bölümleri vardır:

- [“Kuyruk paylaşım grubu tanımlanması” sayfa 383](#)
- [“Kuyruk Paylaşım Grubuna Kuyruk Yöneticisi Eklenmesi” sayfa 384](#)
- [“Kuyruk Yöneticisinin Kuyruk Paylaşım Grubundan Kaldırılması” sayfa 385](#)
- [“Removing a queue sharing group from the Db2 tables” sayfa 385](#)
- [“Db2 tanımlarının tutarlılığı doğrulanıyor” sayfa 386](#)

Kuyruk paylaşım grubu tanımlanması

Her kuyruk paylaşım grubu en çok dört karakterden oluşan bir ada sahiptir. Adın ağızda benzersiz olması ve kuyruk yöneticisi adlarından farklı olması gerekir.

Bir kuyruk paylaşım grubu ayarlamak için aşağıdaki adımları izleyin:

1. Bu, Db2 veri paylaşma grubunu kullanan ilk kuyruk paylaşım grubuysa, [Db2 ortamını ayarlayın](#).
2. [Coupling Facility olanağını ayarlayın](#).
3. Kuyruk paylaşım grubunu Db2 tablolarına ekleyin. Kuyruk paylaşım grubu yardımcı programının (CSQ5PQSG) QSG ADD QSG işlevini kullanın. Bu program, [The queue sharing group yardımcı programı](#) başlıklı konuda açıklanmaktadır. thlqual.SCSQPROC(CSQ45AQS) içinde bir örnek verilmiştir.
4. Add a queue manager to the queue sharing group by following the steps in [“Kuyruk Paylaşım Grubuna Kuyruk Yöneticisi Eklenmesi” sayfa 384](#)
5. Define application structures to IBM MQ by following the steps in [“Bağlaşım olanağı yapısı eklenmesi” sayfa 391](#).
6. Gerekirse, [paylaşımlı olmayan kuyrukları paylaşılan kuyruklara geçirin](#).
7. Kullanılabilirlik için, paylaşılan kanalları kuyruk paylaşım grubunun içine ve dışına oluşturun.
 - Kuyruk paylaşım grubuna yönelik bağlantılar için:
 - QSG ' deki kullanılabilir kuyruk yöneticileri arasında iş yükünü dağıtmak için bir VIPA yuvası ya da donanım yönlendiricisi ayarlayın.
 - QSGDISP (GROUP) ile bir alıcı kanalı tanımlayın, kanal tanımlamasının QSG ' deki tüm kuyruk yöneticilerinde kullanılabilir olmasını sağlayın.
 - QSG ' ye ilişkin MCA kanalı bağlantıları için, her kuyruk yöneticisinde INDISP (GROUP) ile bir dinleyici başlatın. QSG ' ye ilişkin istemci bağlantıları, INDISP (QMGR) ile başlatılan bir dinleyiciye bağlanmaya devam etmelidir.

- Belirli bir kuyruk yöneticisi adı yerine, QSG adını kullanarak bağlantı kurulacak uygulamaları değiştirin.
- Uygulamaların QSG ' deki herhangi bir kuyruk yöneticisine bağlanmasına izin vermek için, QSG içindeki tüm kuyruk yöneticilerindeki kanal doğrulama kurallarının aynı olduğundan emin olun.
- Kuyruk paylaşım grubundan bağlantılar için:
 - Paylaşılan bir iletim kuyruğu tanımlayın.
 - Giden kanalı QSGDISP (GROUP) ve DEFCDISP (SHARED) tanımlamasıyla tanımlayın.

Var olan bir kanalı paylaşılan bir kanala dönüşte dönüştürseniz, kanal tarafından kullanılan eşitleme kuyruğu değiştiyse, kanalı başlatmadan önce RESET CHANNEL komutunu da yayınlayabilmeniz gerekir.

Kuyruk Paylaşım Grubuna Kuyruk Yöneticisi Eklenmesi

Var olan bir kuyruk paylaşım grubuna kuyruk yöneticisi eklenebilir.

Aşağıdakine dikkat edin:

- Kuyruk paylaşım grubu, kuyruk yöneticilerini ekleyebilmek için önce var olmalıdır.
- Kuyruk yöneticisi, tek bir kuyruk paylaşım grubunun üyesi olabilir.

Kuyruk paylaşım grubuna kuyruk yöneticisi eklemek için aşağıdaki adımları izleyin:

1. Kuyruk yöneticisine ve kanal başlatıcı kullanıcı kimliklerine uygun erişimi vermek için kuyruk paylaşım grubu için ESM güvenlik denetimlerini uygula içindeki görevleri gerçekleştirin.
2. If the queue sharing group has CF structures configured to offload data to SMDS, perform the tasks in SMDS ortamını ayarlama.
3. Kuyruk yöneticisini durdurun.
4. Kuyruk paylaşım grubu yardımcı programının (CSQ5PQSG) ADD QMGR işlevini kullanın. Bu program the queue-sharing group yardımcı programı içinde açıklanmıştır. thlqual.SCSQPROC(CSQ45AQM) içinde bir örnek verilmiştir.
5. Kuyruk paylaşım grubu verileri eklemek için Sistem parametre modülünüzü değiştirin :
 - a. QSGDATA parametresini belirlemek için CSQ6SYSP ' yi değiştirin. Ek bilgi için bkz. CSQ6SYSP .
 - b. Sistem parametre modülünü birleştirin ve bağlayın. Yükleme modülü için farklı bir ad kullanmak isteyebilirsiniz.
 - c. Başlatma işleminizi yeni modülü kullanacak şekilde değiştirin.
6. Copy and tailor sample member thlqual.SCSQPROC(CSQ4INSS), which defines required CF structures and SYSTEM queues. Özelleştirilmiş üyeyi kuyruk yöneticisi başlatma JCL 'deki CSQINP2 DD' ye ekleyin.
7. Kuyruk paylaşımı grubu sistem değiştirge modülünü kullanarak kuyruk yöneticinizi yeniden başlatın.
8. İsteğe bağlı olarak, kuyruk yöneticisi adı yerine kuyruk paylaşım grubu adı önekli güvenlik profillerine geçin.
9. Paylaşılan kanallar QSG 'ye bağlantılar için kullanılıyorsa, uygulamaların QSG' deki herhangi bir kuyruk yöneticisine bağlanmasına izin vermek için, QSG ' deki diğer kuyruk yöneticilerindeki bunları ikizlemek için kanal doğrulama kuralları yaratın.
10. İsteğe bağlı olarak, QSG 'deki kuyruk yöneticisine bağlı uygulamaların, QSG' deki diğer kuyruk yöneticilerinin barındırdığı kuyruklara ileti yerleştirmesini sağlamak için aşağıdakilerden birini yapın:
 - ALTER QMGR IGQ (ENABLE) komutunu vererek grup içi kuyruklama ' i 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 önlemektedir.

Not: To add a queue manager to an existing queue sharing group containing queue managers running earlier versions of IBM MQ, you must first apply the coexistence PTF for the highest version of IBM MQ in the group to every earlier version queue manager in the group.

Kuyruk Yöneticisinin Kuyruk Paylaşım Grubundan Kaldırılması

Kuyruk yöneticisinin günlüklerinin başka bir işlem tarafından gerekmiyorsa ve kuyruk yöneticisinin iyeliğindeki tüm SMDS ' ler boş olması durumunda, kuyruk yöneticisini yalnızca kuyruk paylaşım grubundan kaldırabilirsiniz.

Günlükler aşağıdaki durumlarda gereklidir:

- Kuyruk paylaşım grubu tarafından kullanılan bağlantı olanağı (CF) uygulama yapılarından birinin en son yedeklemesi
- Gelecekteki bir geri yükleme işlemi için gereken veriler; yani, kuyruk yöneticisi, son yedekleme zamanaşımı aralığı değeri tarafından tanımlanan zamandan bu yana kurtarılabilir bir yapı kullanmıştır.

Bu noktalardan biri ya da her ikisi geçerliyse ya da kuyruk yöneticisinin iyeliğindeki bir SMDS ileti içeriyorsa, kuyruk yöneticisi kaldırılamaz. İleride geri yükleme işlemi için hangi kuyruk yöneticilerinin günlüklerine gerek olduğunu saptamak için, TYPE (BACKUP) seçeneğiyle MQSC DISPLAY CFSTATUS komutunu kullanın (bu komutun ayrıntıları için [DISPLAY CFSTATUS](#) başlıklı konuya bakın).

Kuyruk yöneticisi bir kuyruk paylaşım grubundan bir kuyruk yöneticisini kaldırmak için aşağıdaki adımları kullanın:

1. İletileri paylaşılan kuyruklara yerleştiren kuyruk yöneticisine bağlı uygulamaları durdurun.
2. Bu kuyruk yöneticisini içeren tüm belirsiz iş birimlerini çözümleyin.
3. Komut DISPLAY USAGE TYPE (SMDS) komutunu vererek, kuyruk yöneticisinin sahip olduğu herhangi bir SMDS ' de herhangi bir ileti olup olmadığını saptayın.
4. Herhangi bir uygulama yapısı için yüklü iletiler varsa, bu iletilerin kuyruktan alınıp alınmasına kadar bekleyin. Devam etmeden önce, DISPLAY USAGE TYPE (SMDS) tarafından bildirilen ofsayt dışı iletilerin sayısı sıfır olmalıdır.
5. STOP QMGR MODE (QUIESCE) komutunu kullanarak, kuyruk yöneticisini temiz bir şekilde sona erdirin.
6. Bir sonraki adımda BACKUP CFSTRUCT komutunda belirleyeceği EXCLINT parametresindeki değer en az karşılığı olan bir aralığı bekleyin.
7. Başka bir kuyruk yöneticisinde, MQSC BACKUP CFSTRUCT komutunu kullanarak ve önceki adımda gerektiği gibi bir EXCLUINT değeri belirterek, her kurtarılabilir CF yapısı için bir CF yapısı yedeklemesi çalıştırın.
8. Herhangi bir CF yapılarını geri yüklemek için kuyruk yöneticisinin günlüklerinin, komutun DISPLAY CFSTATUS (*) TYPE (BACKUP) (yedekleme) tipindeki çıktıyı inceleyerek geri yüklenmesine gerek olmadığını doğrulayın.
9. Kuyruk yöneticisini kuyruk paylaşım grubundan kaldırmak için, CSQ5PQSG yardımcı programının REMOVE QMGR işlevini kullanın. Bu program, [The queue sharing group yardımcı programı](#) başlıklı konuda açıklanmaktadır. thlqual.SCSQPROC(CSQ45RQM) içinde bir örnek verilmiştir.
10. Kuyruk yöneticisini yeniden başlatmadan önce, QSGDATA sistem parametresini varsayılan değerine sıfırlayın ve sistem değıştirge modülünü yeniden yaratın. Sistem parametrelerinin nasıl uyarlanmasına ilişkin bilgi için bkz. [CSQ6SYSPkomutunu kullanma](#) .

Son kuyruk yöneticisini bir kuyruk paylaşım grubunda kaldırırken, REMOVE yerine, FORCE seçeneğini kullanmanız gerektiğini unutmayın. Bu, kurtarma işlemi için gerekli olan kuyruk yöneticisi günlüklerinin tutarlılık denetimlerini gerçekleştirilmemekle birlikte kuyruk yöneticisini kuyruk paylaşım grubundan kaldırır. Bu işlemi yalnızca, kuyruk paylaşım grubunu silerseniz gerçekleştirmeniz gerekir.

Removing a queue sharing group from the Db2 tables

Bir kuyruk paylaşım grubunu Db2 tablolarından kaldırmak için, kuyruk paylaşım grubu yardımcı programının (CSQ5PQSG) QSG REMOVE işlevini kullanın. Bu program [The queue-sharing group yardımcı programı](#) başlıklı konuda açıklanmaktadır. `thlqual.SCSQPROC(CSQ45RQS)` içinde bir örnek verilmiştir.

Kuyruk paylaşım grubundan tüm kuyruk yöneticilerini kaldırdıktan sonra ("[Kuyruk Yöneticisinin Kuyruk Paylaşım Grubundan Kaldırılması](#)" sayfa 385 ' ta açıklandığı gibi) ortak Db2 veri paylaşımı grubu tablolarından yalnızca bir kuyruk paylaşım grubunu kaldırabilirsiniz.

Kuyruk paylaşım grubu kaydı kuyruk paylaşım grubu yönetim tablosundan silindiğinde, o kuyruk paylaşım grubuyla ilgili tüm nesnelere ve yönetim bilgileri diğer IBM MQ Db2 tablolarından silinir. Bu, paylaşılan kuyruk ve grup nesnesi bilgilerini içerir.

Db2 tanımlarının tutarlılığı doğrulanıyor

Bir kuyruk paylaşım grubu içindeki paylaşılan kuyruklara ilişkin sorunlar, Db2 nesne tanımlarının herhangi bir nedenle tutarsız olması durumunda oluşabilir.

Kuyruk yöneticilerine, CF yapılarına ve paylaşılan kuyruklara ilişkin Db2 nesne tanımlamalarının tutarlılığını doğrulamak için, kuyruk paylaşım grubu yardımcı programının (CSQ5PQSG) kuyruk paylaşım grubu yardımcı programının VERIFY QSG işlevini kullanın. Bu program, [The queue sharing group yardımcı programı](#) başlıklı konuda açıklanmaktadır.

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 386
- "[Paylaşılan kuyrukların taşınması](#)" sayfa 387
- "[Paylaşılmayan kuyrukların paylaşılan kuyruklara geçirilmesi](#)" sayfa 389
- [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 kuyrukta 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 sağlamak 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 387 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 OPPROCS 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 387 adımıyla oluşturulan geçici kuyruğa dayalı olur.

7. Aşağıdaki komutu kullanarak iletileri yeni kuyruğa geri taşıyın:

```
MOVE QLOCAL(Temp) TOQLOCAL(Queue_To_Move)
```

8. “3” sayfa 387 adımıyla oluşturulan 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 yerisi konması için REPLACE anahtar sözcüğünü ekleyin.

Şekil 47 sayfa 388 , 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(Temp_Queue) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(Temp_Queue)
DELETE QL(Queue_To_Move)
DEFINE QL(Queue_To_Move) LIKE(Temp_Queue) CFSTRUCT(NEW) QSGDISP(Shared)
MOVE QLOCAL(Temp_Queue) TOQLOCAL(Queue_To_Move)
DELETE QL(Temp_Queue)
/*
```

Şekil 47. 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 387). Şekil 48 sayfa 389 , 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 48. 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 387](#)).

Şekil 49 sayfa 389 , 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 49. 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 48 sayfa 389 , 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 dizin tipi ile ilgili bazı kısıtlamalara tabi olur; bu nedenle, paylaşılmayan bazı kuyrukları paylaşılan bir kuyruğa taşıyamayabilirsiniz.
2. Paylaşılan kuyruklar için doğru dizin tipini kullanmanız gerekir. Bir iletim kuyruğunu paylaşılan bir kuyruk olması için yeni düzeye geçiriyorsanız, dizin tipi MSGID olmalıdır.

Kuyruk boşsa ya da üzerindeki iletileri alkoymak zorunda kalmayacaksa, kuyruğu yeni düzeye geçirmek daha kolaydır. [Şekil 50 sayfa 390](#) , 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 50. 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 48 sayfa 389](#) ya da [Şekil 50 sayfa 390](#) 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 51 sayfa 390](#).

```
MOVE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR) TOQLOCAL(Queue_TO_MOVE)
DELETE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR)
```

Şekil 51. 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 yönetilmesi

Grup nesnelere 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 için 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 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 391
- “Bağlaşım olanağı yapısının çıkarılması” sayfa 391

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 393](#)
- [“Olağandışı sona erdirme işleminden sonra yeniden başlatma” sayfa 393](#)
- [“Sayfa kümelerinizi kaybetmediyseniz yeniden başlatma” sayfa 393](#)

- [“Günlük veri kümelerinizi kaybetmediyseniz yeniden başlatma” sayfa 393](#)
- [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 324](#) 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 324](#) 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 406](#) 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 377](#) 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) kullanılarak 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 günlüğü veri kümelerinizi 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 yordamna 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/OS üzerinde 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](#)
 - [Başyerinde 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 398](#) .)

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* arşivlenen BSDS ' lerini 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 395.
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 395 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 kullanılmasıyla 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 404 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. Bir kuyruk paylaşım grubunu kurtarmak için, kurtarma işlemini, kuyruk paylaşım grubundaki tüm kuyruk yöneticileriyle koordine etmek ve öncelikle Db2 olmak üzere diğer kaynaklarla koordine etmek gerekir. 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.

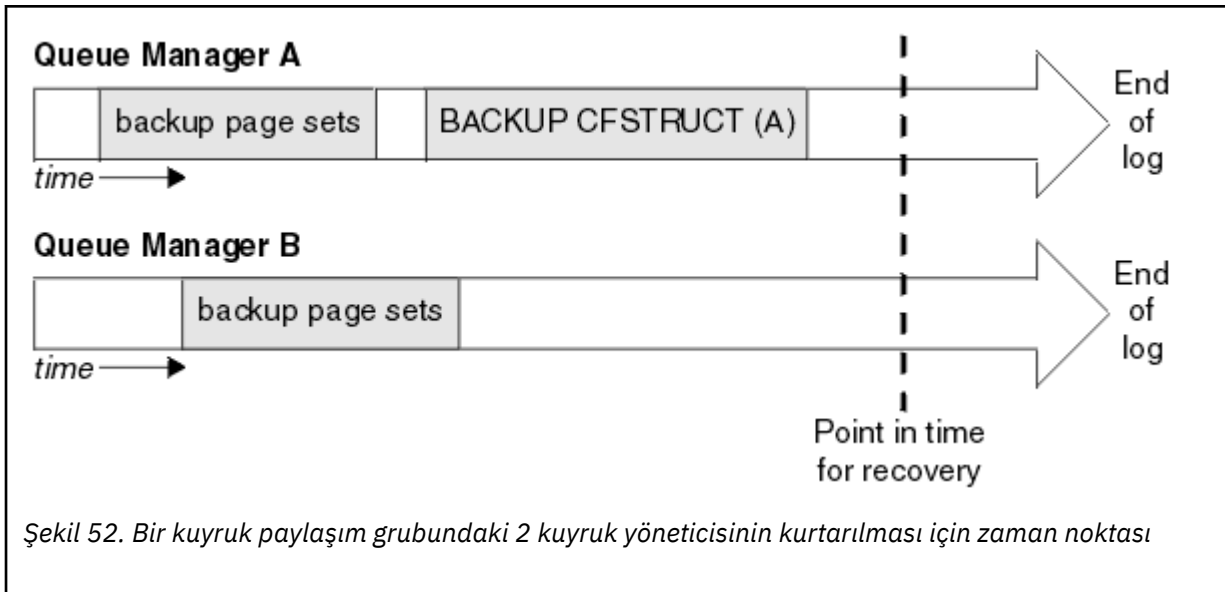
CF yapısı yedeklemesi, BACKUP CFSTRUCT komutunu işleyen kuyruk yöneticisinin günlüğüne yazılır; bu nedenle, toplanacak ve başka yere taşınmak üzere başka veri kümesi de yoktur.

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 olabileceği için, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinin günlükleri, paylaşılan kuyrukları kurtarmak için 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 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 paylaşımı 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* arşivlenen BSDS ' lerini 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 399.
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 399 . 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 399.

BSDS şimdi tüm etkin günlükleri boş olarak tanımlıyor, kullanılabilir durumda olduğunuz arşivlenmiş tüm günlükler ve günlüklerinizin sonundaki kontrol noktaları yok.

Kuyruk paylaşım grubu içindeki her kuyruk yöneticisi için CRESTART kaydını 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 komutu yayınlamaktan daha hızlı bir işlem gerçekleştirilir.

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 'deki eşdüzey BSDS' lerin 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.

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 ya da bozulduğu için olabilir. Bu gerçekleştiyse, 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 başlatıldığında, 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 tanımlanmalıdır.

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 sıfırsa, 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 kullanılmıyorsa, nesne tanımlarınızı en son yedeklediğinizde tanımları kullanın.
2. Kuyruk yöneticisi veri kümelerinizi yeniden tanımlayın ([“1” sayfa 400](#) adımı tamamlayıncaya kadar bunu gerçekleştirilmeyin).
- 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 400](#) 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 sorunlar 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 402](#)
- [“ARM ilkeleri” sayfa 402](#)
- [“IBM MQ ağında ARM ' nin kullanılması” sayfa 404](#)

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 gerçekleştirmeden 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 404](#)).

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 404](#).

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 gerekir. Her z/OS görüntüsünde, kuyruk yöneticisinin yeniden başlatılacağı, bir sistem birleşimi (sysplex) geniş benzersiz dört karakterlik altsistem adıyla tanımlanmalıdır.

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 53 sayfa 403, 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(SYSQMGRMQ*) /* These jobs to be restarted by ARM */
/*
```

Şekil 53. 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 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 olur.

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 404](#)
- [“LU 6.2 ile farklı bir z/OS görüntülerinde yeniden başlatma” sayfa 406](#)

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 bir 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 kapağı gereksinimi vardır; böylece, iki sistem z/OS görüntüsünde tek bir TCP/IP yığını 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 kapağı 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 (kapı) 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 kapı 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 kaydolur. Bu, mantıksal grup adının (kanalın bağlantı adı alanındaki) dinamik DNS ' 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 bir z/OS görüntüsü tarafından başvuru bir paylaşılan APPC yan tablosu ayarlayın. Bu, her kanal başlatıcısı LUNAME için bir giriş içermelidir. Bununla ilgili bilgi 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 öznitelikliğini kullanın. Bu, kanal başlatıcısının, 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 olursa, kanal başlatıcısını otomatik olarak yeniden başlatmayın.

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 407](#)
- [“Recovering CICS units of recovery manually” sayfa 407](#)
- [“Recovering IMS units of recovery manually” sayfa 411](#)

- “RRS kurtarma birimlerinin el ile kurtarılması” sayfa 412
- “Kuyruk paylaşım grubundaki başka bir kuyruk yöneticisinde kurtarma birimlerinin kurtarılması” sayfa 413

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 çözülmemiş 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)
```

Çözülmemiş 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 çözümlenmediği konusunda uyarır. Bu UOWs 'ler, yeniden başlatıldığında dağıtımlı kuyruğa alma bileşeniyle açık bir şekilde çözümlenmelidir.

Şekil 54 sayfa 408 , 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.A6E5A6EFFF60D425 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.A6E5A6EFFF60D425
+CSQC409I VICIC1 CSQCTRUE Resynchronization completed successfully
```

Şekil 54. Yeniden başlatma iletileri örneği

Toplam CSQC313I ileti sayısı, toplam CSQC402I artı CSQC403I iletilerine eşit olmalıdır. Topamlar eşit değilse, bağlantı sürecinin çözümleyememesi gereken UOWs 'ler vardır. Çözülemeyen UOWs 'ler, CICS (örneğin, soğuk başlatma) ya da IBM MQ ile ya da kuyruklama dağıtılarak 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 RESOLVE INDOUBT komutunu kullanarak her bir bekleyen UOW ' yi 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.

Çözülemeyen 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 güvence altına alınamaz. This is because IBM MQ treats the new instance of CICS as a different CICS (because the APPLID is different). Belirsiz çözümlene, 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 MQ tarafından sunulan listede artık görünmediğinde temizlenir.

CICS ' in çözemeyeceği herhangi bir kurtarma birimi, IBM MQ komutları kullanılarak el ile çözülmelidir. 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 ' tan 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 presents the list of unit of work (UOW) IDs that it believes are in doubt to the IBM MQ IMS adapter one at a time with a resolution parameter of Commit or Backout.
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 çözümlünceye 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 414](#)
- [“IMS denetim bölgesinden bağlanma” sayfa 414](#)
- [“Belirsiz kurtarma birimlerinde görüntüleme” sayfa 416](#)
- [“IMS bağımlı bölge bağlantılarını denetleme” sayfa 418](#)
- [“IMSbağlantısı kesiliyor” sayfa 420](#)
- [“IMS tetikleme izleyicisini denetleme” sayfa 421](#)

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 'dan IBM MQ' a bağlanmak için kullanılabilir mekanizmaları anlamak için bu konuyu kullanın.

IMS , denetim bölgesinden IMS' u kullanan her kuyruk yöneticisine bir bağlantı oluşturur. IMS must be enabled to make the connection in one of these ways:

- Aşağıdakilerden biri sırasında otomatik olarak:

- Soğuk bir başlangıç.
 - warmbağlantısı, IMS kapatıldığında IBM MQ bağlantısı etkinse, sıcak bir IMSbaşlangıcı olur.
- IMS komutuna yanıt olarak:

```
/START SUBSYS sysid
```

Burada *sysid* kuyruk yöneticisi adıdır.

Kuyruk yöneticisinin etkin olup olmamasından bağımsız olarak komut yayınlanabilir.

Kuyruk yöneticisine yönelik ilk MQ API çağrısı yapıncaya kadar bağlantı yapılmaz. Bu zamana kadar, IMS komutu /DIS SUBSYS, durumu 'NOT CONN' olarak gösterir.

IMS ' u ve kuyruk yöneticisini başladığınız sıra önemli değil.

Kuyruk yöneticisi bir STOP QMGR komutuyla durdurulduysa, IMS komutu /STOP SUBSYS ya da olağandışı sona erdirilirse,IMS , kuyruk yöneticisiyle bağlantıyı otomatik olarak yeniden etkinleştiremez. Therefore, you must make the connection by using the IMS command /START SUBSYS.

Bağdaştırıcı başlatılıyor ve kuyruk yöneticisine bağlanılıyor

Bağdaştırıcı, IMS dış Subsystem Attach Facility kullanılarak IMS denetim ve bağımlı bölgelerine yüklenen bir modül kümesidir.

Bu yordam, bağdaştırıcıyı başlatır ve kuyruk yöneticisine bağlanır:

1. Read the subsystem member (SSM) from IMS.PROCLIB. Seçilen SSM bir IMS EXEC parametresidir. There is one entry in the member for each queue manager to which IMS can connect. Her bir giriş, bir IBM MQ bağdaştırıcısına ilişkin denetim bilgilerini içerir.
2. IMS bağdaştırıcısını yükleyin.
Not: IMS , SSM üyesi tarafından tanımlanan her bir IBM MQ yönetim ortamı için bağdaştırıcı modüllerinin bir kopyasını yükler.
3. IBM MQ için dış altsistem görevini ekleyin.
4. Bağlantı adı olarak bağdaştırıcıyı CTL EXEC parametresiyle (IMSID) çalıştırın.

Bu işlem, bağlantının kullanıma hazırlamanın bir parçası mı, yoksa IMS komutunun /START SUBSYS komutunun bir sonucu mu olduğu da aynıdır.

IMS bağlantı girişiminde bulunmaya çalışıldığında kuyruk yöneticisi etkinse, aşağıdaki iletiler gönderilir:

- to the z/OS console:

```
DFS3613I ESS TCB INITIALIZATION COMPLETE
```

- IMS ana uçbirimine:

```
CSQQ000I IMS/TM imsid connected to queue manager ssm
```

IMS bağlantıyı yapmayı denediğinde ve *kuyruk yöneticisi etkin değilse*, uygulama bir MQI çağrısını yaptığında, IMS ana uçbirimine 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
```

If you get DFS3607I messages when you start the connection to IMS or on system startup, this indicates that the queue manager is not available. 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.

Hiçbirini yapmazsanız, bir DFS3607I iletisi ve ilişkili CSQQ001I iletisi her iş zamanlandığında yayınlanır ve kuyruk yöneticisine her bağlantı isteği bir uygulama tarafından yapılır.

İş parçacığı eki

Bir MPP ya da IFP bölgesinde, IMS bir uygulama programı IBM MQ çağrısı yapmasa da, ilk uygulama programı o bölgeye zamanlandığında bir iş parçacığı bağlantısı oluşturur. Bir BMP bölgesinde, uygulama ilk IBM MQ çağrısını (MQCONN ya da MQCONNX) durumuna getirdiğinde iş parçacığı bağlantısı yapılır. Bu iş parçacığı, bölge süresi boyunca ya da bağlantı durduruluncaya kadar saklanır.

Hem ileti odaklı, hem de ileti olmayan yönlendirilen bölgeler için, iş parçacığıyla ilişkili kurtarma iş parçacığı çapraz başvuru tanıtıcısı *Thread-xref*:

```
PSTid + PSBname
```

Burada:

PSTid

Bölüm belirtimi tablo bölgesi tanıtıcısı

PSBname

Program belirtimi blok adı

Bağlantı tanıtıcılarını, IBM MQ komutlarında benzersiz tanıtıcılar olarak kullanabilirsiniz; bu durumda IBM MQ , bu tanıtıcıyı otomatik olarak oluşturduğu herhangi bir işletmen iletisine otomatik olarak 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 411.](#)
- 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.

These commands reset the status of IMS ; they do not result in any communication with IBM MQ.

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 bağdaştırıcısı da bağımlı bölgelere yüklenir. Bağımlı bölgelerden IBM MQ' e bir bağlantı yapılır. Bu bağlantı, IBM MQ ve IMS çalışmalarının taahhüdlerini koordine etmek için kullanılır. Bağlantı başlatmak ve bağlantı 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, bağdaştırıcının yüklenmesini önlemek için, girdi içermeyen bir üye belirtin.

2. IBM MQ bağdaştırıcısını yükler.

Bir toplu ileti programı için, uygulama ilk ileti alışverişi komutunu yayınlayıncaya kadar yükleme işlemi gerçekleştirilmez. Bu sırada IMS , bağlantıyı 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 kaynak kullanılamıyorsa, işlem, SSM girişinde belirtilen hata seçeneğine bağlıdır. Seçenekler şunlardır:

R

Uygulamaya uygun dönüş kodu gönderilir.

Q

Uygulama, olağandışı bitiş kodu U3051 ile olağandışı sona eriyor. Giriş iletisi yeniden kuyruğa alındı.

A

Uygulama, olağandışı bitiş kodu U3047 ile olağandışı sona eriyor. Giriş iletisi atılır.

Bağlantılardaki 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
```

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 MQ arasındaki 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.

IMS tetikleme izleyicisini denetleme

Durdurmak için CSQQTRMN işlemini kullanabilir ve IMS tetikleme izleyicisini başlatabilirsiniz.

IMS tetikleme izleme programı (CSQQTRMN hareketi), [IMS tetikleme izleyicisini ayarlamak](#) 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
INITQUEUEUENAME=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.

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 422](#)
- [“Controlling IMS connections” sayfa 423](#)
- [Köprü kuyruklarının denetlenmesi](#)
- [“IMS köprüsünün yeniden eşzamanlanması” sayfa 424](#)
- [Tpipe adlarıyla çalışma](#)
- [IMS' dan iletilerin silinmesi](#)
- [Veri bağlantılarının silinmesi](#)
- [“IMS İşlem Süre Bitimi” sayfa 426](#)

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 *tüye* 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 aşağıdaki 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 iletilerinde 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 eşleştiğinde eşleşmeyen sıra numaralarını keşfederse, CSQ2020E iletileri 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 işlemi geri almak için kullanın. Kurtarma birimini kesinleştirirkeniz 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-iletler 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 bulabileceğinizdir.

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 kullanılmak üzere kullanılabilir bir veri bağlantısı için, ilişkili kuyruğu doğru XCF grubu ve üye adı alanlarına sahip 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 TMBER 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 iletisi
- giriş iletisi 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 DFSINSX0 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 ' ya (DFS2224I) ilişkin bir ileti.

z/OS

Operating Advanced Message Security on z/OS

Advanced Message Security adres alanı, z/OS MODIFY komutunu kullanarak komutları kabul eder.

Advanced Message Security adres alanına ilişkin komutları girmek için, z/OS MODIFY komutunu kullanın.

Örneğin,

```
F qmgr AMSM, cmd
```

Aşağıdaki MODIFY komutları kabul edilir:

Çizelge 27. Advanced Message Security address space MODIFY komutları		
Komut	Seçenek	Tanım
GÖRÜNTÜLE		Sürüm bilgilerini görüntüle

Çizelge 27. Advanced Message Security address space MODIFY komutları (devamı var)

Komut	Seçenek	Tanım
YENİLE	ANAHTARLIK ilke TÜMÜ	Anahtarlık sertifikalarını, güvenlik ilkelerini ya da her ikisini yenileyin.
SSMFAUDIT	BAŞARILI ARIZA TÜMÜ	AMS iletileri başarıyla korurken/korurken, AMS iletileri ya da her ikisini de korumayı/korumayı başaramadığında SMF denetiminin gerekli olup olmadığını belirleyin.
SMFTYPE	0 - 255	MS iletileri korurken/koruduğunda, SMF kayıt tipini üretilecek şekilde ayarlayın. SMF denetimini geçersiz kılmak için kayıt tipi olarak 0 belirtin.

Not: Bir seçeneği belirtmek için, bir virgül ile ayrılmalıdır. Örneğin:

```
F qmgIAMS,REFRESH KEYRING  
F qmgIAMS,SMFAUDIT ALL  
F qmgIAMS,SMFTYPE 180
```

REFRESH komutu.

Bir MQOPER çağrısı yayınlayan bir uygulama değişiklikleri alır. Var olan uygulamalar, o uygulamanın kuyruğu açtığı zamandan itibaren seçenekleri kullanmaya devam eder. Değişiklikleri almak için, bir uygulamanın kapatılması ve kuyruğun yeniden açılması gerekir.

z/OS V 9.0.1 IBM MQ for z/OS Service Provider for z/OS Connect

IBM MQ for z/OS Service Provider for z/OS Connect (MQ Service Provider), z/OS Connect aracılığıyla gelen istekleri işler. The MQ Service Provider allows REST aware applications to interact with z/OS assets, that are exposed using IBM MQ for z/OS queues and topics. Bu işlemi, zamanuyumsuz ileti sistemini kullanmak için gereken kodlamayı düşünmeden başarabilirsiniz.

Önemli: z/OS Connect EE version 3.0.21.0 and later, ships an enhanced version of the MQ Service Provider which supports service archive files. You should migrate to that version of z/OS Connect EE and use the built-in MQ Service Provider, instead of using the service provider shipped with the IBM MQ for z/OS product.

IBM Documentation'teki z/OS Connect EE bilgilerinde, z/OS Connect EE 'de MQ Service Provider için hızlı başlatma senaryoları [Hızlı başlangıç senaryoları](#) altında sağlanır ve ayrıntılı başvuru bilgileri [IBM MQ hizmet sağlayıcısını kullanma](#) altında sağlanır.

Bu bölüm aşağıdaki bilgilerden oluşur:

İlgili bilgiler

Video: [IBM MQ Service Provider for z/OS Connect \(YouTube\)](#)

z/OS V 9.0.1 IBM MQ for z/OS Service Provider for z/OS Connect-genel bakış

An overview of IBM MQ for z/OS Service Provider for z/OS Connect, explaining the principles used and the verbs the service provider uses.

Önemli: z/OS Connect EE version 3.0.21.0 and later, ships an enhanced version of the MQ Service Provider which supports service archive files. You should migrate to that version of z/OS Connect EE and use the built-in MQ Service Provider, instead of using the service provider shipped with the IBM MQ for z/OS product.

IBM Documentation içindeki z/OS Connect EE bilgilerinde, z/OS Connect EE ' de MQ Service Provider için hızlı başlatma senaryoları Hızlı başlangıç senaryoları altında sağlanır ve ayrıntılı başvuru bilgileri IBM MQ hizmet sağlayıcısını kullanma altında sağlanır.

MQ Service Provider , z/OS Connect 'in iki sürümünü destekler:

IBM z/OS Connect Sürüm 1 (z/OS Connect V1)

This is a component of WebSphere Liberty Profile (WLP) on z/OS, available at no extra cost, which exposes z/OS assets as a REST interface and allows remote applications, such as those running on a mobile device, to send them data formatted using JSON.

Daha fazla bilgi için, bkz. IBM z/OS Connect-Genel Bakış.

Önemli:

1. IBM için MQ Service Provider z/OS Connect V1 , IBM WebSphere MQ 7.5 kaynak bağdaştırıcısı gerektirir. Bu, 30th Nisan 2018 hizmet tarihi sona erme tarihine sahiptir.

Bu noktadan sonra MQ Service Provider , IBM z/OS Connect V1 ile desteklenmeyecek ve bunun yerine IBM z/OS Connect EE olanağını kullanmalısınız.

Bu nedenle, IBM z/OS Connect V1 , üretim ortamlarında değil, kavrama ve kavrama kanıtlanması için daha iyi bir şekilde yerleştirilmektedir.

2. MQ Service Provider yalnızca, WLP 8.5.5.9 ve sonraki yayın düzeylerinde verilen z/OS Connect V1 kodunu destekler.

IBM z/OS Connect Enterprise Edition (z/OS Connect EE)

Bu, IBM z/OS Connect V1 içindeki tüm işlevlerin yanı sıra, RESTful API 'ları oluşturmak için araçlar da dahil olmak üzere çeşitli geliştirmeler sağlayan ve IBM API Connect ile bütünleştirme de dahil olmak üzere, ayrı, ücretlendirilebilir bir ürün olan WLP' ye dayanır.

Daha fazla bilgi için bkz. IBM z/OS Connect EE.

Önemli: MQ Service Provider yalnızca z/OS Connect EE V2.0.3.0 (APAR PI66869) ya da üstünü destekler.

MQ Service Provider , hangi z/OS Connect sürümü kullanıldan bağımsız olarak aynı işlevi sağlar. All of the capabilities of z/OS Connect EE, such as the API Editor, are supported with the MQ Service Provider.

Belge boyunca, *z/OS Bağlan* terimi, belirli bir sürüme gönderme yapmak gerekmiyorsa, her iki sürümü de belirtmek için kullanılır; bu durumda, önceki metinde tanımlanan adlardan biri kullanılır.

z/OS Connect tarafından gösterilen z/OS varlıklarının REST veya JSON ' u anlamaları gerekmez; z/OS Connect, yerel çağrılara yönelik REST çağrılarını eşler ve ayrıca, COBOL kopya kitapları gibi yerel veri yapıları arasında dönüştürülecek dönüşümler de sağlar.

MQ Service Provider , ilgili z/OS Connect kurulumuna kurulabilen ayrı bir WLP özelliği olarak kullanılabilir. MQ Service Provider 'u ayarlamaya ilişkin çalışmanın büyük bölümünün yapılandırmaya dayalı olduğunu, böylece REST uygulamasının IBM MQ' in farkında olmaya gerek kalmadığını unutmayın. Ancak, daha ileri düzey uygulamalar için, MQMD alanları gibi öğelere erişim sağlanır.

MQ Service Provider , IBM MQ kuyruklarını ve konuları "Hizmetler" sayfa 429 olarak gösterir.

Desteklenen IBM MQ sürümleri

MQ Service Provider , IBM MQ for z/OS 8.0 ve sonraki yayın düzeyleriyle desteklenir.

IBM MQ 9.0.1 sürümünden önceki sürümler MQ Service Provider ile gönderilmez. Bunun yerine, MQ Service Provider dosyasını nasıl edinebileceğiyle ilgili bilgi için "MQ Service Provider' in elde edilmesi" sayfa 431 ' e bakın.

IBM Documentation ' in z/OS Connect bölümlerinde açıklanan belgeler, desteklenen tüm IBM MQ sürümlerine eşit olarak uygulanır.

Diğer önemli noktalar

MQ Service Provider aşağıdakilerden birini kullanır:

- Yıkıcı olmayan bir MQGET çağrısına ilişkin HTTP GET
- Yıkıcı MQGET çağrısına ilişkin HTTP DELETE
- MQPUT çağrısı için HTTP POST

Bunlar, HTTP için IBM MQ Köprüsü tarafından kullanılan aynı fiillerdir. Benzer şekilde, MQ Service Provider , MQMD değerleri gibi çeşitli şeyleri belirtmenin bir yolu olarak HTTP üstbilgilerini kullanır.

HTTP çağrılarına ilişkin daha fazla bilgi için bkz. [“MQ Service Provider ' in desteklediği fiiller” sayfa 430](#)

JSON verileri olarak gösterilen tek verinin, iletinin bilgi yükü olduğunu unutmayın. This means that applications do not unnecessarily get exposed to IBM MQ concepts if all they need is the message data.

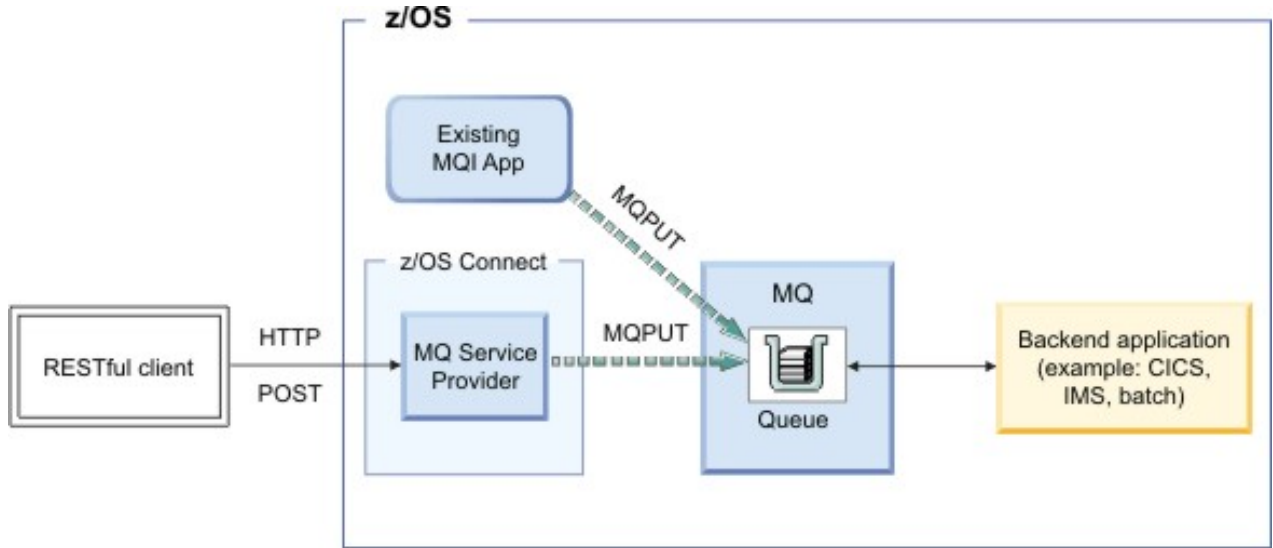
MQ Service Provider , WLP 'de IBM MQ kaynak bağdaştırıcısı desteğini kullanır; dolayısıyla, JMS' ye dayalıdır.

z/OS V 9.0.1 Hizmetler

The MQ Service Provider exposes IBM MQ queues and topics, and the applications that are behind them, as services. İki tip hizmet vardır: tek yönlü ve iki yönlü, bu bölümde anlatılır.

Tek yönlü hizmet

Tek yönlü hizmet, tek bir IBM MQ kuyruğu ya da konunun üzerine RESTful API sağlamak için kullanılabilir. RESTful istemcileri, tek yönlü bir hizmet için JSON bilgi yükü ile bir HTTP POST işlemi yayınlayabilir ve hizmet bilgi yükünü alır ve ileti gövdesi olarak bilgi yükü ile hedef kuyruğa ya da konuya ileti gönderir.



Şekil 55. Tek yönlü hizmet

Tek yönlü hizmetler, IBM MQ kuyrukları için HTTP DELETE ve HTTP GET isteklerine de izin verir.

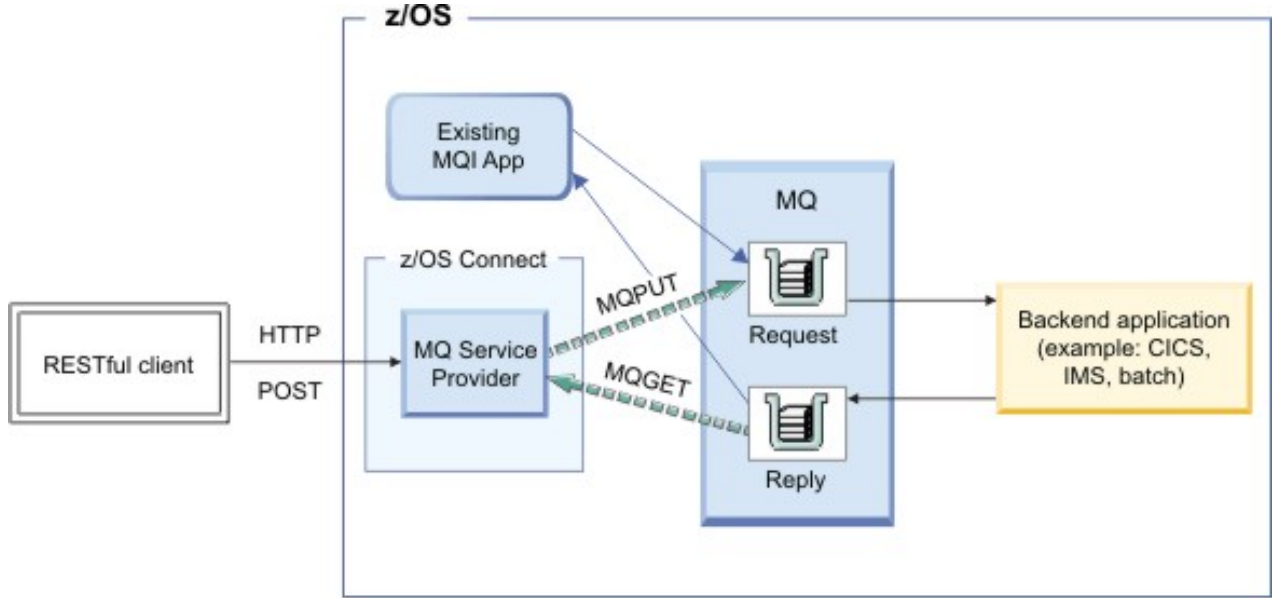
HTTP DELETE işlemi, kuyruktan kullanılabilir bir iletiyi yıkıcı bir şekilde alır. Bir HTTP GET işlemi, kuyruktan ilk kullanılabilir iletiyi göz atarak sonuçla sonuçlanır.

Not: İletiyi kuyruktan kaldırmak için, HTTP DELETE ya da ileti süre bitimi gibi başka bir işlem gerçekleşmediği sürece, iki HTTP GET çağrısına aynı iletiyi döndürür.

İletinin gövdesi, JSON biçiminde RESTful istemcisine geri döndürülür. İletinin gövdesi önceden JSON biçiminde değilse (örneğin, bir COBOL copybook), z/OS Connect 'i bir veri dönüştürmesi kullanarak JSON ' a dönüştürmek için yapılandırılabilir.

İki yönlü hizmet

İki yönlü hizmet, RESTful istemcisinin, bir çift kuyruğa karşı istek yanıtla iletilerini gerçekleştirmesini sağlar.



Şekil 56. İki yönlü hizmet

İstemci, bir JSON bilgi yükü belirterek bir HTTP POST isteği gönderir. Hizmet bilgi yükünü alır, isteğe bağlı olarak bunu COBOL copybook gibi farklı bir biçime dönüştürür ve bunu, bir istek kuyruğuna ileti olarak gönderir.

Bir arka uç uygulaması iletiyi tüketir, işlem işlemini işler ve bir yanıt kuyruğuna yerleştirilen bir yanıt oluşturur. Hizmet bu iletiyi bulur, bilgi yükünü alır, isteğe bağlı olarak JSON 'a dönüştürür ve bunu HTTP POST' un yanıt gövdesi olarak döndürür.

z/OS V 9.0.1 MQ Service Provider ' in desteklediği filler

MQ Service Provider , HTTP GET, HTTP DELETE ve HTTP POST fillerini, aynı anlamı taşıyan HTTP için IBM MQ Bridge 'de kullanılan HTTP POST (POST) ve HTTP POST (HTTP POST) verisini destekler. Bu fillerin kullanılması, IBM MQ for z/OS ile temel etkileşime izin verir ve bu fillerin bir sayısının bileşimi, daha karmaşık işlevlerin ortaya çıkarılmasına olanak tanır.

HTTP GET ya da HTTP DELETE işlemi yaparken, MQ Service Provider istekte sağlanan tüm gövdeleri yoksayar.

HTTP GET ya da HTTP DELETE başarılı olursa:

- HTTP yanıt gövdesinde ileti bilgi yükü döndürülür ve HTTP durum kodu 200 'dür.
- Herhangi bir HTTP gövdesi olmadan bilgi yükü olmayan bir ileti döndürüyor. Kullanılabilir bir ileti yoksa, HTTP gövdesi döndürülmez ve HTTP durum kodu 204 (İçerik Yok) olur.



Uyarı: z/OS Connect, her zaman nesne biçiminde bir JSON bilgi yükü olmasını bekler; bu, en düşük düzeyde geçerli bilgi yükü '{ }' tir.

HTTP POST işlemi yaparken, beklenen tek parametre, HTTP gövdesinde olacak ileti bilgi yükü ve mqzOSConnectHizmet ögesi içinde açıklanan çeşitli öğeler.

POST başarılı bir şekilde gerçekleşirse:

Tek yönlü hizmet

Boş bilgi yükü ile birlikte HTTP 204 (İçerik Yok) kodunu döndürür.

İki yönlü hizmet

JSON biçiminde yanıt iletilsinin içeriğini ya da yanıt iletilisi yoksa, HTTP yanıt kodu 204 (İçerik Yok) ile boş bir yanıt gövdesi döndürür.

HTTP üstbilgileri

Bir HTTP POST işlemi sırasında tek yönlü ya da iki yönlü bir hizmet verildiğinde, HTTP gövdesi geçerli bir JSON nesnesi olmalıdır. Şunları belirtmeniz gerekir:

- HTTP_POST ' a Content-Type=application/json HTTP üstbilgisi
- UTF-8 kodlaması

MQ Service Provider' in davranışını ayarlamak için kullanılan HTTP fiiline bağlı olarak, isteğe bağlı HTTP üstbilgileri de belirtilebilir. Daha ayrıntılı bilgi için bkz. [MQ Service Provider ile birlikte kullanılacak HTTP üstbilgileri](#) .

HTTP kodları

MQ Service Provider bir hata saptarsa, hizmet 400-599 aralığında bir HTTP durum kodu ayarlar. Ters durumda, önceden belgelenmiş olan durum kodları her zaman döndürülür.

İstemci tarafı hatası

Çağırılan uygulama tarafından geçerli olmayan veriler iletilirse, arayanın 400-499 aralığındaki bir HTTP durum kodu döndürülür. Tam olarak durum kodu hataya bağlıdır.

Geçerli olmayan veriler verilmekte olan bir kural dışı durum tarafından algılandıysa, sunucu tarafı örneğinde olduğu gibi JSON biçiminde bir yığın izlemesi sağlanır.

Sunucu tarafı hatası

MQ Service Provider beklenmeyen bir hata deneyimliyorsa, yığın izlemesi JSON biçimine dönüştürülecek ve 500 'e (İç Sunucu Hatası) HTTP durum kodu ile çağrıya dönecektir. z/OS Connect günlüklerine uygun tanılama programları da yazılacaktır.

Basitleştirilmiş örnek hata yanıtı bilgi yükü:

```
{
  "errorText": "CSQZ0006E: An unexpected JMSEException occurred while processing a request
for the 'mq7' service. ",
  "exceptionData": {
    "exceptions": [
      {
        "stackTrace": [
          "com.ibm.msg.client.jms.DetailedInvalidDestinationException: JMSWMQ2008:
Failed to open MQ queue 'ThisQueueDoesNotExist'.",
          <further content removed for brevity>
        ]
      },
      {
        "stackTrace": [
          "com.ibm.mq.MQException: JMSCMQ0001: WebSphere MQ call failed with
compcode '2' ('MQCC_FAILED') reason '2085' ('MQRC_UNKNOWN_OBJECT_NAME').",
          <further content removed for brevity>
        ]
      }
    ]
  }
}
```

MQ Service Provider' in elde edilmesi

MQ Service Provider , IBM MQ for z/OS 8.0 ve sonraki yayın düzeylerine göre kullanılmak üzere desteklenir. There are two ways to obtain the MQ Service Provider for use with any supported version of IBM MQ.

1. [“Obtaining the MQ Service Provider from the Unix Systems Services Components feature of IBM MQ 9.0.1 \(or later\)” sayfa 432](#)
2. [“MQ Service Provider 'in Fix Central' den elde edilmesi” sayfa 432](#)

Obtaining the MQ Service Provider from the Unix Systems Services Components feature of IBM MQ 9.0.1 (or later)

Programı, program dizinindeki yönergeleri izleyerek kurun. MQ Service Provider , dosya sisteminizde *PATHPREFIX/zosconnect* dizininde bulunur; burada *PATHPREFIX* , kuruluş sırasında seçilen değerdir. Varsayılan olarak, IBM MQ 9.0.1 için *PATHPREFIX* , */usr/lpp/mqm/V9R0M1*' dir.

MQ Service Provider 'in Fix Central' den elde edilmesi

[Fix Central](#) 'a gidin ve *IBM-MQ-zOSConnect-ServiceProvider* ' u arayarak MQ Service Provider ' u bulun.

Bu, iş istasyonunuza yüklenebilen bir tar dosyası (örneğin, 9.0.1-IBM-MQ-zOSConnect-ServiceProvider.tar) bulmalıdır.



Uyarı: Aynı tar dosyası da IBM MQ 8.0 ile birlikte kullanılır. IBM MQ 8.0 için ayrı bir tar dosyası yok.

Karşıdan yükledikten sonra, tar dosyasını uygun bir z/OS LPAR üzerinde *PATHPREFIX* dizinine aktarın. Bu dizin, z/OS Connect tarafından erişilebilir bir yerde olmalıdır.

Aşağıdaki komutu girerek dosyaları çıkarın:

```
tar -oxvf 9.0.1-IBM-MQ-zOSConnect-ServiceProvider.tar
```

Dizin yapısının izinlerini, kuruluşunuz için uygun olan şekilde değiştirin.

Dizin yapısı

MQ Service Provider ' ı elde etmenin her iki yolu da aynı izin yapısını oluşturur. Bu yapının kök dizini, bu belge boyunca *MQSP_ROOT* olarak anılır; burada *MQSP_ROOT* , *PATHPREFIX/zosconnect* olur.

MQSP_ROOT altında aşağıdaki dosyalar ve dizinler yer alıyor:

mqzosconnect.properties	Properties file that can be copied into z/OS Connect
v1.0/	Directory containing MQ Service Provider for z/OS
Connect V1	
lib/	
com.ibm.mq.zosconnect_1.0.0.jar	Feature jar file for MQ Service Provider for z/OS
Connect V1	
features/	
zosConnectMQ-1.0.mf	Feature manifest for MQ Service Provider for z/OS
Connect V1	
v2.0/	Directory containing MQ Service Provider for z/OS
Connect EE	
lib/	
com.ibm.mq.zosconnect_2.0.0.jar	Feature jar file for MQ Service Provider for z/OS
Connect EE	
features/	
zosConnectMQ-2.0.mf	Feature manifest for MQ Service Provider for z/OS
Connect EE	

z/OS V 9.0.1 İşlemsel noktalar

HTTP bir işlemsel iletişim kuralı değil, bu nedenle MQ Service Provider tarafından gerçekleştirilen ileti alışverişi işlemlerinin herhangi bir işlemsel koordinasyonu mümkün değil.

Bu, aşağıdaki sonuçları içerir:

- Tek yönlü bir hizmette HTTP POST işlemi yapılırsa ve istemci tarafından bir HTTP yanıtı alınmadan bağlantı başarısız olursa, istemci hemen yapılandırılan kuyruğa ya da konuya iletinin gönderilip gönderilmediğini hemen söyleyemez.

- Tek yönlü bir hizmette HTTP DELETE işlemi yapılırsa ve istemci tarafından bir HTTP yanıtı alınmadan bağlantı başarısız olursa, bir ileti, yıkıcı bir şekilde kuyruktan alınmış ve kaybedilmiş olabilir; bu durumda yıkıcı geri alma işlemi devam eder.
- Bir HTTP POST işlemi iki yönlü bir hizmetle yapılırsa ve istemci tarafından bir HTTP yanıtı alınmadan bağlantı başarısız olursa, hata oluştuğu yeri istemciye söyleyemez. İstek iletisi istek kuyruğuna gönderilmiş olabilir ya da yanıt iletisi yanıt kuyruğundan almış ve kaybedilmiş olabilir.
- Birden çok HTTP verisinin sonucunu, tek yönlü ya da iki yönlü bir hizmetle koordine etmek mümkün değildir.

IBM z/OS Connect EE - MQ Service Provider kurulum yordamı

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect EE.

Bu görev hakkında

MQ Service Provider ' yi ve tüm önkoşullarını kurmak için gösterilen sırayla aşağıdaki görevleri yerine getiriniz.

IBM z/OS Connect EE ' nin kurulması

IBM z/OS Connect EE kuruluşuna ilişkin gereksinimler ve yordam.

Başlamadan önce

[z/OS Connect EE V2 yerleşik WLP ' yi güncelleerişimine sahip olduğundan emin olun.](#)

Bu görev hakkında

Bu yordam, MQ Service Provider ile kullanım için uygun bir z/OS Connect EE sunucusunu ayarlar. Önceden ayarlanmış bir sunucunuz varsa, bunun yerine kullanılacak bir sunucu varsa.

Yordam

1. Burada açıklanan yönergeleri kullanarak bir sunucu oluşturun: [z/OS Connect EE Sunucusu Yaratılması](#)
2. Enable the TXRRS authorized service by following the procedure detailed in [Liberty Angel sürecini ve z/OS yetkili hizmetlerini yapılandırma](#).
3. TXRRS yetkili hizmetinin doğru şekilde ayarlandığını doğrulayın:
 - a) Sunucu başlatılıyor.
[z/OS Connect EE ' nin başlatılması ve durdurulması'](#) ta ayrıntılı olarak açıklanan yordamı izleyin.
 - b) Şu adreste bulunan sunucu günlüklerini görüntüleyin:

```
/var/zosconnect/servers/server_name/logs/messages.log
```

Bunlar ASCII dosyalarıdır ve aşağıdakine benzer bir çıkış içermelidir:

```
A CWWKE0001I: The server test has been launched.
I CWWKB0103I: Authorized service group LOCALCOM is available.
I CWWKB0103I: Authorized service group PRODMGR is available.
I CWWKB0103I: Authorized service group SAFCRE is available.
I CWWKB0103I: Authorized service group TXRRS is available.
I CWWKB0103I: Authorized service group WOLA is available.
I CWWKB0103I: Authorized service group ZOSDUMP is available.
I CWWKB0103I: Authorized service group ZOSWLM is available.
I CWWKB0103I: Authorized service group CLIENT.WOLA is available.
I CWWKB0108I: IBM CORP product z/OS Connect version 02.00 successfully registered with z/OS
```

TXRRS yetkili hizmet grubunun kullanılabilir olup olmadığını görmek için çıktığı denetleyin. Önceki örnekte, koyu metindeki satır hizmet grubunun kullanılabilir olduğunu gösterir.

MQ Service Provider ' in bu hizmet grubunun kullanılabilir olması gerekiyor

c) Sunucu durduruluyor.

Komutun komut satırından başlatılması ve durdurulması başlıklı bölümdeki yordamı izleyin.

Sonuçlar

z/OS Connect EE ' yi başarıyla kurdun.

Sonraki adım

IBM MQ kaynak bağdaştırıcısını takın.

İlgili görevler

“IBM z/OS Connect EE - MQ Service Provider kurulum yordamı” sayfa 433

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect EE.

“IBM MQ kaynak bağdaştırıcısının takılması” sayfa 434

The IBM MQ messaging provider feature in IBM z/OS Connect EE makes use of a component of IBM MQ called the IBM MQ resource adapter. Kaynak bağdaştırıcısı, IBM MQ for z/OS Unix System Services Components özelliğinin bir parçası olarak IBM MQ ile birlikte sağlanır.

IBM MQ kaynak bağdaştırıcısının takılması

The IBM MQ messaging provider feature in IBM z/OS Connect EE makes use of a component of IBM MQ called the IBM MQ resource adapter. Kaynak bağdaştırıcısı, IBM MQ for z/OS Unix System Services Components özelliğinin bir parçası olarak IBM MQ ile birlikte sağlanır.

Başlamadan önce

Bağlantı kuracağı kuyruk yöneticisi sürümü için IBM MQ for z/OS Unix System Services Components özelliğinin kurulu olduğundan emin olun.

Birden çok kuyruk yöneticisi sürümüne bağlanacaksa, bu özelliği en son sürümden kullanmanız gerekir.

Bu görev hakkında

Bu yordam, IBM MQ kaynak bağdaştırıcısını IBM z/OS Connect EE ' ye kurar.

Yordam

1. Bağlantı kuracağı IBM MQ sürümüne ilişkin Unix System Services Components dizinini bulun.
Örneğin, bu dizin /usr/lpp/mqm/V9R0M1/olarak adlandırılabilir. Dizin, bir dizi yerel kitaplık (.so dosyası) içeren bir java/lib alt dizini içermelidir.



Uyarı: Birden çok kuyruk yöneticisi sürümüne bağlanacaksa, en son sürümü kullanın.

2. z/OS Connect EE sunucusunun server.xml dosyasını düzenleyin.

Aşağıdaki satırları ekleyin:

```
<variable name="wmqJmsClient.rar.location"
  value="MQJAVA_LIB_DIR/jca/wmq.jmsra.rar"/>
<wmqJmsClient nativeLibraryPath="MQJAVA_LIB_DIR"/>
```

Burada MQJAVA_LIB_DIR , “1” sayfa 434adımında bulunan dizini temel alır;
örneğin, /usr/lpp/mqm/V9R0M1/java/lib.

- b. Değişiklikleri kaydedin.

Birinci satırda, IBM z/OS Connect EE ' ye IBM MQ kaynak bağdaştırıcısının nerede bulunacağı açıklanır.

The second line tells the IBM MQ resource adapter where to locate the native libraries that are used for bindings connections to IBM MQ.

Daha fazla bilgi için bakınız: [wmqJmsClient](#)ve [Deploying JMS applications to Liberty to use the IBM MQ Messaging provider](#) .

3. Sunucunuzun STEPLIB ' i IBM MQ kitaplıklarını içerecek şekilde ayarlayın.

Bunu, “1” sayfa 434 numaralı adımda bulunan yerel kitaplıkların kuyruk yöneticisine bağlanabilmesi için bunu yapmanız gerekir.

Bu genellikle, sunucunuzu başlatan JCL ' nin düzenlenmesini içerir; bu işlem aşağıdakileri içerir:

```
//STEPLIB DD DSN=HLQ.SCSQAUTL,DISP=SHR
// DD DSN=HLQ.SCSQANLE,DISP=SHR
```

Burada HLQ , IBM MQ kuruluşunu içeren veri kümelerinin üst düzey niteleyicidir.

Aynı sunucudan birden çok IBM MQ sürümüne bağlanabildiyseniz, en son sürümden veri kümelerini kullanın.

Sonuçlar

IBM MQ kaynak bağdaştırıcısını kısmen kurdunuz.

IBM MQ kaynak bağdaştırıcısının, “[z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme](#)” sayfa 436’inde açıklandığı şekilde, üst özelliği (wmqJmsClient-2.0) etkinleştirilinceye kadar tam olarak kurulmayacaklarını unutmayın.

Sonraki adım

MQ Service Provider 'i IBM z/OS Connect EE' ye kurun.

İlgili görevler

“[IBM z/OS Connect EE - MQ Service Provider kurulum yordamı](#)” sayfa 433

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect EE.

“[IBM z/OS Connect EE ' nin kurulması](#)” sayfa 433

IBM z/OS Connect EE kuruluşuna ilişkin gereksinimler ve yordam.

“[MQ Service Provider 'i IBM z/OS Connect EE' ye kurma](#)” sayfa 435

You must install the MQ Service Provider into IBM z/OS Connect EE before it is used.

MQ Service Provider 'i IBM z/OS Connect EE' ye kurma

You must install the MQ Service Provider into IBM z/OS Connect EE before it is used.

Bu görev hakkında

Bunu gerçekleştirmek için aşağıdaki yordamı gerçekleştirin:

Yordam

1. IBM z/OS Connect EE ürün uzantıları dizinini bulun.
Tipik bir kuruluş için bu, burada açıklanan `/var/zosconnect/v2r0/extensions` dizinidir: [Ürün uzantılarını ayarlama dizini](#).
2. `MQSP_ROOT/mqzosconnect.properties` dosyasını, “1” sayfa 435’adımından yer alan dizine kopyalayın.
3. Kopyalanan dosyayı düzenleyin.

Bu dosyanın bir ASCII dosyası olduğunu unutmayın. `PATH_TO_INSTALL` 'u `MQSP_ROOT/v2.0` olarak değiştirin ve değişikliklerinizi kaydedin.



Uyarı: “3” sayfa 435 adımı, sunucuyu çalıştıran kullanıcı kimliğinin `MQSP_ROOT` dizin yapısına okuma erişiminin olduğunu varsayar. Durum böyle değilse, kullanıcı kimliğine yeterli erişimi ekleyin ya da `MQSP_ROOT` içeriğini yeterli erişime sahip bir konuma taşıyın.

Sonuçlar

Şimdi, MQ Service Provider için tüm önkoşulları kurdunuz.

Sonraki adım

Şimdi MQ Service Provider ve z/OS Connect 'i etkinleştirmeniz gerekir.

İlgili görevler

“IBM z/OS Connect EE - MQ Service Provider kurulum yordamı” sayfa 433

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect EE.

“IBM MQ kaynak bağdaştırıcısının takılması” sayfa 434

The IBM MQ messaging provider feature in IBM z/OS Connect EE makes use of a component of IBM MQ called the IBM MQ resource adapter. Kaynak bağdaştırıcısı, IBM MQ for z/OS Unix System Services Components özelliğinin bir parçası olarak IBM MQ ile birlikte sağlanır.

“z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme” sayfa 436

z/OS Connect ve MQ Service Provider' i etkinleştirmek için yapmanız gereken

z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme

z/OS Connect ve MQ Service Provider' i etkinleştirmek için yapmanız gereken

Başlamadan önce

Yordamları tamamladığınızdan emin olun:

- “IBM z/OS Connect EE ' nin kurulması” sayfa 433
- “IBM MQ kaynak bağdaştırıcısının takılması” sayfa 434
- “MQ Service Provider 'i IBM z/OS Connect EE' ye kurma” sayfa 435

Bu görev hakkında

Bu yordam hem z/OS Connect, hem de MQ Service Provider' i etkinleştirir.

Yordam

1. Yarattığınız z/OS Connect EE sunucusunun `server.xml` dosyasını düzenleyin ve tüm **featureManager** ögesini şu satırlarla değiştirin:

```
<featureManager>
  <feature>zosconnect:zosconnect-2.0</feature>
  <feature>appSecurity-2.0</feature>
  <feature>jms-2.0</feature>
  <feature>mqzosconnect:zosConnectMQ-2.0</feature>
  <feature>wmqJmsClient-2.0</feature>
  <feature>zosTransaction-1.0</feature>
</featureManager>
```



Uyarı: Bu girişleri yalnızca önceden varolmasalar da sona erdirin.

2. z/OS Connect için güvenliği yapılandırın.

Bu yordamın gerçekleştirilmesine ilişkin ayrıntılar için [Configuring security for z/OS Connect EE](#) başlıklı konuya bakın.

3. Sunucuyu başlatın.

Sonraki adım

z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun.

İlgili görevler

[“IBM z/OS Connect EE - MQ Service Provider kurulum yordamı” sayfa 433](#)

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect EE.

[“z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun” sayfa 437](#)

z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun.

 **z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun**

z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun.

Başlamadan önce

Ensure that you have carried out the procedure detailed in [“z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme” sayfa 436](#)

Bu görev hakkında

z/OS Connect, kurulu olan hizmetleri sorgulamak ve bunları durdurma ve başlatma gibi yönetimle ilgili işlemleri gerçekleştirmek için kullanılacak RESTful API sağlar.

Yordam

1. Şu anda kurulu olan hizmetlerin listesini sorgulamak için z/OS Connect 'e yönelik bir HTTP GET sorunu yayınlayın.

Bunu, formun URL 'sini girerek bir web tarayıcısıyla yapabilirsiniz.

```
https://HOST_NAME:HTTPS_PORT/zosConnect/services
```

Burada *HOST_NAME* ve *HTTPS_PORT* , [“IBM z/OS Connect EE ' nin kurulması” sayfa 433](#) ' un [“1” sayfa 433](#) adımımda girdiğiniz değerlerdir.

Örneğin,

```
https://yourdomainname:12342/zosConnect/services
```

2. Tarayıcı tarafından istendiğinde kullanıcı kimliğini ve parolayı girin.

Bunlar, [“z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme” sayfa 436](#) ' un [“2” sayfa 436](#) . adımımda kullanıcı ögesinde girdiğiniz değerlerdir.

Sonuçlar

Bu, z/OS Connect 'in çalıştığını, ancak kurulu herhangi bir hizmet olmadığını gösteren aşağıdaki JSON yanıtına neden olur. Kurulu olan bir z/OS Connect sunucusu kullanıldıysa, bu hizmetlerin görüntüleneceği unutulmadır.

```
-----  
{  
  "zosConnectServices": [ ]
```

}

Sonraki adım

[z/OS Connect EE üzerinde basit bir şekilde IBM MQ hizmeti kurun.](#)

İlgili görevler

“IBM z/OS Connect EE - MQ Service Provider kurulum yordamı” sayfa 433

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect EE.

z/OS Connect EE üzerinde basit bir şekilde MQ Service Provider hizmeti ayarlama

Basit bir tek yönlü MQ Service Provider hizmeti ayarlamak için aşağıdaki adımları kullanın.

Başlamadan önce

Ensure that you have set up all the components correctly, as described in [z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme](#) and [z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun](#)

Yordam

1. MQSC ya da IBM MQ Explorer 'ı kullanarak, hedef z/OS kuyruk yöneticisinde ONE_WAY_QUEUE adlı bir kuyruk yaratın.
2. Bir IBM MQ ileti sistemi sağlayıcı bağlantı üreticisi ve bir kuyruk tanımlayın.
Bunu, yarattığınız z/OS Connect EE sunucusunun alt kısmına (`server.xml`) ekleyerek, ancak sunucu ögesinin içinde bunu yapabilirsiniz.

```
-----  
<jmsConnectionFactory id="cf1" jndiName="jms/cf1" connectionManagerRef="ConMgr1">  
  <properties.wmqJms  
    transportType="BINDINGS"  
    queueManager="MQ21" />  
</jmsConnectionFactory>  
  
<connectionManager id="ConMgr1" maxPoolSize="5" />  
  
<jmsQueue id="q1" jndiName="jms/d1">  
  <properties.wmqJms  
    baseQueueName="ONE_WAY_QUEUE" />  
</jmsQueue>  
-----
```

Notlar:

- a. **queueManager** özneliğinin değerini, doğru hedef kuyruk yöneticisi adına çevirin.
 - b. Bağ tanımlarının bir **transportType** değeri kullanılır. Kuyruk yöneticisiyle iletişim kurmak için bir çapraz bellek bağlantısının kullanıldığı anlamına gelir. Bu, MQ Service Provider kullanıldığında desteklenen tek **transportType** ' dir.
3. Yarattığınız z/OS Connect EE sunucusunun `server.xml` ögesine, ancak sunucu ögesinin içinde eklenerek tek yönlü bir tek yönlü IBM MQ hizmeti tanımlayın.

```
-----  
<zosconnect_zosConnectService id="zosconnMQ1"  
  invokeURI="/oneWay"  
  serviceName="oneWay"  
  serviceRef="oneWay " />  
  
<mqzosconnect_mqzOSConnectService id="oneWay "  
  connectionFactory="jms/cf1"
```

```
destination="jms/d1"/>
```

`zosConnectService` ögesi, `oneWay`'un `serviceName` ' u kullanarak z/OS Connect 'e yeni bir hizmet tanımlar. Sunucu:

- **invokeURI** özniteliği, hizmeti çağırmanın daha kolay olmasını sağlar.
- **serviceRef** özniteliği, bir z/OS Connect hizmet sağlayıcısının ID özniteiyle eşleşmelidir; bu durumda `mqsOSConnectService` ögesi bu durumda sağlanır.

`mqsOSConnectService` ögesi, MQ Service Providertarafından sağlanan tek bir hizmet eşgörünümünü tanımlar.

connectionFactory ve **destination** öznitelikleri, yönetim ortamına sırasıyla IBM MQ ileti alışverişi sağlayıcı bağlantı üreticinin ve kuyruğun nasıl bulunacağını anlatır.

Bu yapıdaki özniteliklere ilişkin ayrıntılar için bkz. [mqsOSConnectService ögesi](#) .

Sonuçlar

Basit bir tek yönlü hizmet kurdun.


Sonraki adım

Hizmeti test etmek gerekir.

İlgili görevler

[“z/OS Connect EE üzerinde tek yönlü hizmeti test etme” sayfa 439](#)

Tek yönlü hizmetinizin çalıştığından emin olmak için bir dizi adım.

 [z/OS Connect EE üzerinde tek yönlü hizmeti test etme](#)
Tek yönlü hizmetinizin çalıştığından emin olmak için bir dizi adım.

Başlamadan önce

[“z/OS Connect EE üzerinde basit bir şekilde MQ Service Provider hizmeti ayarlama” sayfa 438](#) işlemini başarıyla tamamladığınızı doğrulayın.

Yordam

1. z/OS Connect 'in yeni hizmeti tanıdığından emin olun.

Bunu yapmak için, [z/OS Connect 'in doğru bir şekilde ayarlandığına bakın](#)başlıklı konuda ayrıntılı adımları yeniden yayınlayın.

Şu anda tanımlı bir hizmet olduğu için, aşağıdaki çıkışa benzer bir şey görmemiz gerekir:

```
-----  
{  
  "zosConnectServices": [  
    {  
      "serviceName": "oneWay",  
      "serviceDescription": "DATA_UNAVAILABLE",  
      "serviceProvider": "IBM MQ for z/OS service provider for IBM z/OS Connect EE V2.0",  
      "serviceURL": "https://yourdomainname:12342/zosConnect/services/oneWay"  
    }  
  ]  
}
```

Not: The **ServiceDescription** is returned as DATA_UNAVAILABLE because the **serviceDescription** attribute was not specified on the `zosConnectService` element.

2. Hizmete bir HTTP POST işlemi yayınlayın. Bu, POST 'ta belirtilen bilgi yükü her neyse, ONE_WAY_QUEUE' ye gönderilen yeni bir iletiyle sonuçlanır.

Bu tür bir testi gerçekleştirmenin iyi bir yolu, Google Chrome için [Advanced REST istemcisi \(ARC\)](#) fişi gibi bir şey kullanmaktan başka bir yöntem değildir.

a) Aşağıdaki URL 'ye karşı bir HTTP POST işlemi yapmak için ARC' yi kullanın:

```
https://HOST_NAME:HTTPS_PORT/oneWay
```

b) Gövde için geçerli bir JSON örneği belirtin; örneğin:

```
{"name1" : "value1", "name2" : "value2"}
```

İki üstbilgi belirtmeniz gerekir:

i) Yetkilendirme = Basic ENCODED_USERID_PASSWORD

Burada ENCODED_USERID_PASSWORD , [Temel Kimlik Doğrulama Şeması](#)' nda açıklandığı gibi, temel 64 kodlu kullanıcı kimliği ve parolasıdır.

ii) İçerik Tipi = application/json

İlk üstbilgi, z/OS Connect 'te oturum açmak için kullanılır ve z/OS Connect ve MQ Service Providerbağlantısını etkinleştirmeinde kullanılan kullanıcı kimliği ve parolasının kodlanmış bir biçimidir.

İkinci üstbilgi, sunucuya, MQ Service Provider' e HTTP POST çağrılarını verilirken desteklenen tek değer olan bilgi yükünün JSON olduğunu bildirir.

Bu başlığı belirtmezseniz, desteklenmeyen ortam tipi sonuçlarını belirten bir HTTP 415 hatası oluştu.

Tek yönlü hizmet sonuçlarına boş bir HTTP yanıt gövdesinde bir HTTP POST işlemi verilmesi ve 204 numaralı HTTP yanıt kodu döndürülmesi (içerik yok).

ARC, yanıtın herhangi bir veri içermediğini bildiren bir ileti görüntüler.

To validate that a message has been put, use a tool like the IBM MQ Explorer to browse the contents of ONE_WAY_QUEUE. Gönderilen JSON bilgi yükünün bulunduğu kuyruğun MQSTR biçiminde bir IBM MQ iletisi olması gerekir.

3. Aşağıdaki URL 'ye karşı HTTP DELETE komutu vermek için ARC' yi kullanın:

```
https://HOST_NAME:HTTPS_PORT/oneWay
```

Bu yıkıcı bir şekilde, sadece tek yönlü servise gönderilen mesajı alır.

Bu, ilk olarak gönderilen JSON bilgi yükünü içeren bir yanıt gövdesi vermelidir. İletinin imhamı geçici olarak aldığını doğrulamak için IBM MQ Explorer 'ı kullanabilirsiniz.

İlgili görevler

[“z/OS Connect EE üzerinde basit bir şekilde MQ Service Provider hizmeti ayarlama” sayfa 438](#)
Basit bir tek yönlü MQ Service Provider hizmeti ayarlamak için aşağıdaki adımları kullanın.

z/OS Connect EE üzerinde basit bir iki yönlü IBM MQ hizmeti ayarlama

z/OS Connect EE ' de, var olan bir çift IBM MQ kuyruğu çiftine karşı MQ_REQUEST_Q ve MQ_REPLY_Q adı verilen bir çift yönlü IBM MQ hizmeti kurmanızı sağlar.

Başlamadan önce

Tüm bileşenleri doğru bir şekilde ayarladığınızdan emin olun ([“z/OS Connect ve MQ Service Providerbağlantısını etkinleştirme” sayfa 447](#) ve [“z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun” sayfa 448](#) içinde açıklandığı gibi).

Bu görev hakkında

Bu, z/OS Connect kullanılarak gönderilen istek iletilerini tüketmek ve yanıt iletileri oluşturmak için arka uç uygulamasına gereksinim duymanız gibi, tek yönlü bir hizmeti ayarlamaya göre biraz daha fazla bir işe bulaşmanızı sağlar.

Bu görev, MQ_REQUEST_Q ve MQ_REPLAY_Q kuyruklarının var olan bir arka uç uygulaması tarafından kullanıldığını varsayar; örneğin, bir CICS ya da IMS işlemi, bir istek iletilisini MQ_REQUEST_Q ' dan alır ve MQ_REPLY_Q üzerine yerleştirilen bir yanıt iletilisi oluşturur.

Yordam

1. Bir IBM MQ ileti alışverişi sağlayıcı bağlantı üreticisi ve z/OS Connect EE sunucusuna iki IBM MQ ileti alışverişi sağlayıcısı kuyruğu tanımlayın.

IBM MQ ileti alışverişi sağlayıcısı kuyrukları var olan MQ_REQUEST_Q ve MQ_REPLY_Q kuyruklarıyla eşlenir. Bunu, yarattığınız z/OS Connect EE sunucusunun alt kısmına (`server.xml`) ekleyerek, ancak sunucu ögesinin içinde bunu yapabilirsiniz.

```
<jmsConnectionFactory id="cf2" jndiName="jms/cf2"
  connectionManagerRef="ConMgr2">
  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21"/>
</jmsConnectionFactory>

<connectionManager id="ConMgr2" maxPoolSize="5"/>

<jmsQueue id="requestQueue" jndiName="jms/requestQueue">
  <properties.wmqJms baseQueueName="MQ_REQUEST_Q"/>
</jmsQueue>

<jmsQueue id="replyQueue" jndiName="jms/replyQueue">
  <properties.wmqJms baseQueueName="MQ_REPLY_Q"/>
</jmsQueue>
```

Notlar:

- a. **queueManager** özniteliğinin değerini, doğru hedef kuyruk yöneticisi adına çevirin.
 - b. Arka uç uygulamasının niteine bağlı olarak, bu öğelerin daha fazla yapılandırmasını gerçekleştirmeniz gerekebilir. Ek bilgi için [“MQ Service Provider’ in yapılandırılması” sayfa 453](#) başlıklı konuya bakın.
 - c. Koşut zamanlı istek sayısına bağlı olarak, **maxPoolSize** özniteliğinin ayarlanması gerekebilir. Bu öznitelige ilişkin ek bilgi için [JMS bağlantıları için bağlantı havuzlarının yapılandırılması](#) başlıklı konuya bakın.
2. Define the two-way IBM MQ service by adding the following to `server.xml`, of the z/OS Connect EE server that you have created, but inside the `server` element.

```
-----
<zosconnect_zosConnectService id="zosconnMQ2"
  invokeURI="/twoWay"
  serviceName="twoWay"
  serviceRef="twoWay"/>

<mqzosconnect_mqzOSConnectService id="twoWay"
  connectionFactory="jms/cf2"
  destination="jms/requestQueue"
  replyDestination="jms/replyQueue"
  waitInterval="10000"/>
-----
```

Notlar:

- a. Arka uç uygulamasının niteine bağlı olarak, bu öğelerin daha fazla yapılandırmasını gerçekleştirmeniz gerekebilir. Örneğin, veri dönüştürmeye gereksinim duyabilirsiniz. Ek bilgi için [“MQ Service Provider’ in yapılandırılması” sayfa 453](#) başlıklı konuya bakın.
- b. Daha fazla bilgi için, **waitInterval** değerini ayarlamanız gerekebilir; bkz. **waitInterval** .

- c. Bu örnekte, yanıt hedefine yerleştirilen iletilerin arka uç uygulaması tarafından, istek hedefindeki ileti tanıtıcısıyla aynı olan bir ilinti tanıtıcısına sahip olduğu varsayılır.

Böyle bir durumda değilse, **replySelection** özniteliğini `mqsosconnect_mqzOSConnectService` ögesine eklemeniz ve değeri uygun bir şekilde ayarlamanız gerekir.

Ek bilgi için **replySelection** başlıklı konuya bakın.

Sonuçlar

z/OS Connect EE ' de iki yönlü bir hizmet ayarladınız; bu hizmet, bir IBM MQ kuyruklarından oluşan, var olan bir arka uç uygulaması ile kullanılabilir.

Sonraki adım

You can invoke the two-way service issuing an HTTP POST command to the URL `https://host_name:port_no/twoWay`.

Bunu yapılacak adımlar, yalnızca HTTP POST komutu desteklendiğinden ve var olan arka uç uygulamasına uygun JSON bilgi yükü verilerini göndermeniz dışında, [“z/OS Connect EE üzerinde tek yönlü hizmeti test etme” sayfa 439](#)' ta anlatılanlarla benzerdir.

İlgili görevler

“z/OS Connect EE üzerinde basit bir şekilde MQ Service Provider hizmeti ayarlama” [sayfa 438](#)
Basit bir tek yönlü MQ Service Provider hizmeti ayarlamak için aşağıdaki adımları kullanın.

Hizmet arşivi (SAR) dosyası oluşturulması

z/OS Connect EE ile bir API oluşturabilmeniz için önce bir SAR dosyası oluşturmanız gerekir.

Bu görev hakkında

Bunu, IBM Documentation'teki *z/OS Connect EE* bilgilerinde açıklanan CICS-WOLA işlemi kullanarak IBM MQ ile başarabilirsiniz. Ek bilgi için [CICS COBOL copybook 'tan bir hizmet arşivi oluşturulması](#) başlıklı konuya bakın.

Önemli: Aşağıdaki metinde açıklanan işlem isteğe bağlıdır. API düzenleyicisinin sağladığı olanaklardan birini kullanmak istemiyorsanız, SAR dosyasına gerek yoktur. However, you might want to make use of the JSON schemas and bind files described in the preceding link, if you want to use the data transformation support provide by z/OS Connect EE.

Önceki bağlantıda açıklanan işlem, z/OS Connect EE ile sağlanan **BAQLS2JS** yardımcı programının kullanımını sağlar.

Bu yardımcı program:

1. z/OS Connect EE ' de sağlanan veri dönüştürme işleviyle kullanılacak JSON şemaları ve bağ tanımlama dosyaları oluşturur.

If you want to configure these files in z/OS Connect EE, you set them in the configuration through the `zosconnect_zosConnectDataXform` element.

Ek bilgi için [“Using data transformations with the MQ Service Provider” sayfa 456](#) başlıklı konuya bakın.

2. Önceki adımdan JSON şemalarını içeren bir SAR dosyası oluşturur ve belirli bir z/OS Connect EE hizmet eşgörünümünün **serviceName** özniteliğine başvuruyu oluşturur.

Başvuru, SERVICE-NAME PARAMETRESİ kullanılarak **BAQLS2JS** ' e sağlanır.

Örneğin, SERVICE-NAME=MQ1WayBackend , **BAQLS2JS** için bir öznitelik olarak ayarlandıysa, **BAQLS2JS** aşağıdaki gibi yapılandırılmış bir IBM MQ hizmet eşgörünümlü ile birlikte kullanılabilir bir SAR oluşturur:

```
<zosconnect_zosConnectService id="MQ1WayService"
  serviceName="MQ1WayBackend"
  serviceRef="mq1way"
  invokeURI="/mq1way"
  dataXformRef="xformJSON2Byte"/>

<mqzosconnect_mqZOSConnectService id="mq1way"
  connectionFactory="jms/cf1"
  destination="jms/oneWayRequestQ"/>
```

Sonuçlar

Oluşturulduktan sonra, SAR dosyası bir API oluşturmak için API Düzenleyicisi 'ne aktarılabilir. Ek bilgi için [z/OS Connect EE API Düzenleyicisi ile API yaratılması](#) başlıklı konuya bakın.

Once an API has been constructed in the API Editor, an API Archive (AAR) is generated which can then be deployed to z/OS Connect EE, where it will run against the service referenced by the SERVICE-NAME attributed in **BAQLS2JS**. Ek bilgi için [API 'nin z/OS Connect EE sunucusuna konuşlandırılması](#) başlıklı konuya bakın.

İlgili görevler

“z/OS Connect EE üzerinde basit bir şekilde MQ Service Provider hizmeti ayarlama” sayfa 438
Basit bir tek yönlü MQ Service Provider hizmeti ayarlamak için aşağıdaki adımları kullanın.

“z/OS Connect EE üzerinde basit bir iki yönlü IBM MQ hizmeti ayarlama” sayfa 440
z/OS Connect EE 'de, var olan bir çift IBM MQ kuyruğu çiftine karşı MQ_REQUEST_Q ve MQ_REPLY_Q adı verilen bir çift yönlü IBM MQ hizmeti kurmanızı sağlar.

IBM z/OS Connect V1 - MQ Service Provider kurulum yordamı

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect V1 .

Bu görev hakkında

MQ Service Provider 'yi ve tüm önkoşullarını kurmak için gösterilen sırayla aşağıdaki görevleri yerine getiriniz.

kurmaWebSphere Application Server Liberty

WebSphere Application Server Liberty (WLP) kuruluşuna ilişkin gereksinimler ve yordam.

Başlamadan önce

Ensure that you have access to a version of WLP for z/OS that includes z/OS Connect, for example 16.0.0.2.

Bu görev hakkında

Bu yordam WLP 'yi kurar ve bir sunucu kurar.

Yordam

1. [Installing Liberty using Installation Manager](#)'ta ayrıntılı olarak açıklanan yordamı izleyerek WLP'yi kurun.

Bu yönergelerin geri kalan kısmında, bu izin yapısının *WLP_ROOT* olarak anıldığını unutmayın.

2. [Creating a Liberty sunucusunu el ile oluşturmabaşlıklı bölümdeki yordamı izleyerek yeni bir sunucu oluşturun.](#)

Oluşturulan sunucuda z/OS Connect 'in nasıl etkinleştirileceğini açıklayan [z/OS Connect 'i ayarlama belgesine](#) bakın.



Uyarı: İsteğe bağlı *WebSphere eniyileştirilmiş yerel bağdaştırıcılar (WOLA)* adımları gerçekleştirmenize gerek yoktur.

3. Enable the TXRRS authorized service by following the procedure detailed in [Enabling z/OS authorized services on Liberty for z/OS](#)
4. TXRRS yetkili hizmetinin doğru şekilde ayarlandığını doğrulayın:

- a) Sunucu başlatılıyor.

[Komutun komut satırından başlatılması ve durdurulması](#) başlıklı bölümdeki yordamı izleyin.

- b) Şu adreste bulunan sunucu günlüklerini görüntüleyin:

```
WLP_ROOT/usr/servers/server_name/logs/message.log
```

Bunlar ASCII dosyalarıdır ve aşağıdakine benzer bir çıkış içermelidir:

```
A CWWKE0001I: The server server1 has been launched.
I CWWKB0103I: Authorized service group LOCALCOM is available.
I CWWKB0103I: Authorized service group PRODMGR is available.
I CWWKB0103I: Authorized service group SAFCREED is available.
I CWWKB0103I: Authorized service group TXRRS is available.
I CWWKB0103I: Authorized service group WOLA is available.
I CWWKB0103I: Authorized service group ZOSDUMP is available.
I CWWKB0103I: Authorized service group ZOSWLM is available.
I CWWKB0103I: Authorized service group CLIENT.WOLA is available.
I CWWKB0108I: IBM CORP product WAS FOR Z/OS version 16.0 successfully registered with z/OS.
```

TXRRS yetkili hizmet grubunun kullanılabilir olup olmadığını görmek için çıktığı denetleyin. Önceki örnekte, koyu metindeki satır hizmet grubunun kullanılabilir olduğunu gösterir.

MQ Service Provider ' in bu hizmet grubunun kullanılabilir olması gerekir.

- c) Sunucu durduruluyor.

[Komutun komut satırından başlatılması ve durdurulması](#) başlıklı bölümdeki yordamı izleyin.

Sonuçlar

WLP ' yi başarıyla kurdun.

Sonraki adım

IBM MQ kaynak bağdaştırıcısını takın.

İlgili görevler

[“IBM z/OS Connect V1 - MQ Service Provider kurulum yordamı” sayfa 443](#)

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect V1 .

[“IBM MQ kaynak bağdaştırıcısının takılması” sayfa 444](#)

The IBM MQ messaging provider feature in WLP makes use of a component of IBM MQ called the IBM MQ resource adapter. For compatibility reasons, users of z/OS Connect V1 must use the IBM WebSphere MQ 7.5 resource adapter, regardless of the version of the queue manager that it connects to.

IBM MQ kaynak bağdaştırıcısının takılması

The IBM MQ messaging provider feature in WLP makes use of a component of IBM MQ called the IBM MQ resource adapter. For compatibility reasons, users of z/OS Connect V1 must use the IBM WebSphere MQ 7.5 resource adapter, regardless of the version of the queue manager that it connects to.

Başlamadan önce

Fix Central'a gidin ve en son kaynak bağdaştırıcısını karşıdan yüklemek için [Obtaining the IBM MQ Resource Adapter for WLP](#)' e gidin ve en son kaynak bağdaştırıcısını yükleyin.

Kaynak bağdaştırıcısı şu biçimde olacak: `7.5.0.Fix_Level-WS-MQ-Java-InstallRA.jar`. Bu yordamın başvurduğu bu jar dosyasıdır.

Bu görev hakkında

This procedure installs the IBM MQ resource adapter in WLP.

Yordam

1. Jar dosyasının içeriğini açın, `wmq` dizinine gidin ve ikili olarak `wmq.jmsra.rar` dosyasını hedef dosya sistemindeki uygun bir dizine FTP ile FTP ' ye alın.
Bu dizin, bu yönergelerin geri kalanı için `MQRA_ROOT` olarak adlandırılır.
2. Hedef kuyruk yöneticisine ilişkin Unix System Services Components dizinini bulun.
Örneğin, bu dizin `/mqm/V9R0M1/`olarak adlandırılabilir. Dizin, bir dizi yerel kitaplık (`.so` dosyası) içeren bir `java/lib` alt dizini içermelidir.
3. Sunucunun `server.xml` dosyasını düzenleyin.
Aşağıdaki satırları ekleyin:

```
<variable name="wmqJmsClient.rar.location"
  value="MQRA_ROOT/wmq.jmsra.rar"/>
<wmqJmsClient nativeLibraryPath="MQJAVA_LIB_DIR"/>
```

Burada `MQJAVA_LIB_DIR` , “2” sayfa 445. adımda bulunan dizindir; örneğin, `/usr/lpp/mqm/V9R0M1/java/lib`.

b. Değişiklikleri kaydedin.

İlk satırda, WLP ' ye IBM MQ kaynak bağdaştırıcısının nerede bulunacağı açıklanır.

The second line tells the IBM MQ resource adapter where to locate the native libraries that are used for bindings connections to IBM MQ.

Daha fazla bilgi için bakınız: [wmqJmsClient](#)ve [Deploying JMS applications to Liberty to use the IBM MQ Messaging provider](#) .

4. STEPLIB ' i ayarlayın.

Bunu, “2” sayfa 445 numaralı adımda bulunan yerel kitaplıkların kuyruk yöneticisine bağlanabilmesi için bunu yapmanız gerekir.

WLP sunucusunu başlatmak için kullanılan kabukta aşağıdaki komutu çalıştırın:

```
export STEPLIB=HLQ.SCSQAUTH:HLQ.SCSQANLE
```

Burada `HLQ` , IBM MQ kuruluşunu içeren veri kümelerinin üst düzey niteleyicidir.

Sonuçlar

IBM MQ kaynak bağdaştırıcısını kısmen kurdunuz. IBM MQ kaynak bağdaştırıcısının, üst özelliği (`wmqJmsClient-1.1`) etkinleştirilinceye kadar tam olarak kurulmayacağına dikkat edin.

Sonraki adım

MQ Service Provider 'yi WLP' ye kurun.

İlgili görevler

[“IBM z/OS Connect V1 - MQ Service Provider kurulum yordamı” sayfa 443](#)

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect V1 .

[“kurmaWebSphere Application Server Liberty” sayfa 443](#)

WebSphere Application Server Liberty (WLP) kuruluşuna ilişkin gereksinimler ve yordam.

[“MQ Service Provider 'in WLP' ye kurulması” sayfa 446](#)

You must install the MQ Service Provider into WLP before it is used.

MQ Service Provider 'in WLP' ye kurulması

You must install the MQ Service Provider into WLP before it is used.

Başlamadan önce

Obtain, and install, the MQ Service Provider, using the information contained in [“MQ Service Provider' in elde edilmesi” sayfa 431.](#)

Bu görev hakkında

MQ Service Provider , Liberty özelliklerinin [paketlenmesi ve kurulması](#) içinde açıklanan yöntemlerden birini kullanarak kurabileceğiniz standart bir WLP özelliğidir.

`#{wlp.install.dir}` ve `#{wlp.user.dir}` ' in ne anlama olduğunu açıklamak için [Dizin konumları ve özellikleri](#) başlıklı konuya bakın.

Aşağıdaki metin, yöntemlere genel bir bakış sağlar.

Yordam

1. Özelliği WLP çekirdeğine kurun.



Uyarı: Bu yöntem, WLP kuruluş dizinine yazma erişimi gerektirir. WLP kuruluş dizinine yazma erişiminiz yoksa, [“2” sayfa 446](#) seçeneğinin daha uygun olması beklenir.

Bunu yapmak için, aşağıdaki kopyayı kopyalayın:

- a) `MQSP_ROOT/v1.0/lib/com.ibm.mq.zosconnect_1.0.0.jar - #{wlp.install.dir}/lib.`
 - b) `MQSP_ROOT/v1.0/lib/features/zosConnectMQ-1.0.mf - #{wlp.install.dir}/lib/features.`
2. Özelliği kullanıcı yapılandırmasına kurun.
Bu kopyayı yapmak için:
 - a) `MQSP_ROOT/v1.0/lib/com.ibm.mq.zosconnect_1.0.0.jar - {wlp.user.dir}/extension/lib.`
 - b) `MQSP_ROOT/v1.0/lib/features/zosConnectMQ-1.0.mf - {wlp.user.dir}/extension/lib/features.`
 3. Özelliği ürün uzantısı olarak kurun.
Bunu yapmak için:
 - a) `MQSP_ROOT/mqzosconnect.properties` dosyasını `#{wlp.install.dir}/etc/extensions` olarak kopyalayın.
 - b) Kopyalanan dosyayı düzenleyin.
Bu dosyanın bir ASCII dosyası olduğunu unutmayın. `PATH_TO_INSTALL` ' u `MQSP_ROOT/v1.0` olarak değiştirin ve değişikliklerinizi kaydedin.

Sonuçlar

Şimdi, MQ Service Provider için tüm önkoşulları kurdunuz.

Not: MQ Service Provider 'u kurmak için kullandığınız yaklaşım, `server.xml` ta gönderme yaptığı yöntemi etkiler. Örnekler [“z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme” sayfa 447](#) içinde gösterilir.

Sonraki adım

Şimdi MQ Service Provider ve z/OS Connect 'i etkinleştirmeniz gerekir.

İlgili görevler

[“IBM z/OS Connect V1 - MQ Service Provider kurulum yordamı” sayfa 443](#)

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect V1 .

[“IBM MQ kaynak bağdaştırıcısının takılması” sayfa 444](#)

The IBM MQ messaging provider feature in WLP makes use of a component of IBM MQ called the IBM MQ resource adapter. For compatibility reasons, users of z/OS Connect V1 must use the IBM WebSphere MQ 7.5 resource adapter, regardless of the version of the queue manager that it connects to.

[“z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme” sayfa 447](#)

z/OS Connect ve MQ Service Provider' yi etkinleştirmek için yapmanız gereken

 **z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme**
z/OS Connect ve MQ Service Provider' yi etkinleştirmek için yapmanız gereken

Başlamadan önce

Yordamları tamamladığınızdan emin olun:

- [“kurmaWebSphere Application Server Liberty” sayfa 443](#)
- [“IBM MQ kaynak bağdaştırıcısının takılması” sayfa 444](#)
- [“MQ Service Provider 'in WLP' ye kurulması” sayfa 446](#)

Bu görev hakkında

Bu yordam hem z/OS Connect, hem de MQ Service Provider' i etkinleştirir.

Yordam

1. Oluşturduğunuz `server.xml` dosyasını düzenleyin ve tüm **featureManager** ögesini bu satırlarla değiştirin.



Uyarı: `<feature>zosConnectMQ-1.0</feature>` satırının tam biçimi, [“MQ Service Provider 'in WLP' ye kurulması” sayfa 446](#)'ta açıklanan MQ Service Provider' i kurmak için hangi yönteme bağlı olduğunu bildirir.

Kullanılsa:

- [“1” sayfa 446](#) yöntemi, satır `<feature>zosConnectMQ-1.0</feature>` olarak görünür.
- [“2” sayfa 446](#) yöntemi, satır `<feature>usr:zosConnectMQ-1.0</feature>` olarak görünür.
- [“3” sayfa 446](#) yöntemi, satır `<feature>mqzosconnect:zosConnectMQ-1.0</feature>` olarak görünür.

Ayrıca, yalnızca bunlar önceden var olmadıysa, aşağıdaki girdileri de değiştirmeyin.

```
<featureManager>
<feature>zosConnect-1.0</feature>
<feature>appSecurity-2.0</feature>
<feature>zosConnectMQ-1.0</feature>
<feature>wmmqJmsClient-1.1</feature>
```

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

2. z/OS Connect için güvenliği yapılandırın.

Bu yordamın gerçekleştirilmesine ilişkin ayrıntılar için [Configuring security for z/OS Connect](#) başlıklı konuya bakın.

3. Sunucuyu başlatın.

[Komut satırından bir sunucunun başlatılması ve durdurulması](#)' ta ayrıntılı olarak açıklanan yordamı izleyin.

Sonraki adım

z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun.

İlgili görevler

“IBM z/OS Connect V1 - MQ Service Provider kurulum yordamı” sayfa 443

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect V1 .

“z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun” sayfa 448

z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun.

 **z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun**
z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun.

Başlamadan önce

Ensure that you have carried out the procedure detailed in [“z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme” sayfa 447](#)

Bu görev hakkında

z/OS Connect, kurulu olan hizmetleri sorgulamak ve bunları durdurma ve başlatma gibi yönetimle ilgili işlemleri gerçekleştirmek için kullanılacak RESTful API sağlar.

Yordam

1. Şu anda kurulu olan hizmetlerin listesini sorgulamak için z/OS Connect 'e yönelik bir HTTP GET sorunu yayınlayın.

Bunu, formun URL 'sini girerek bir web tarayıcısıyla yapabilirsiniz.

```
https://HOST_NAME:HTTPS_PORT/zosConnect/services
```

Burada *HOST_NAME* ve *HTTPS_PORT* , [“kurmaWebSphere Application Server Liberty” sayfa 443](#)' un [“2” sayfa 444](#) adımımda girdiğiniz değerlerdir.

Örneğin,

```
https://yourdomainname:12342/zosConnect/services
```

2. Tarayıcı tarafından istendiğinde kullanıcı kimliğini ve parolayı girin.

Bunlar, [“z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme” sayfa 447](#)' un [“2” sayfa 448](#) . adımımda kullanıcı ögesinde girdiğiniz değerlerdir.

Sonuçlar

Bu, z/OS Connect 'in çalıştığını belirten, ancak kurulu herhangi bir hizmet almadığını gösteren şu JSON yanıtına neden olur:


```
-----  
{  
  "zosConnectServices": [    ]  
}  
-----
```

Sonraki adım

[“z/OS Connect V1 üzerinde basit bir şekilde MQ Service Provider hizmeti ayarlama” sayfa 449](#)

İlgili görevler

[“IBM z/OS Connect V1 - MQ Service Provider kurulum yordamı” sayfa 443](#)

An overview of the components you need to install to enable the MQ Service Provider to be set up on IBM z/OS Connect V1 .

z/OS Connect V1 üzerinde basit bir şekilde MQ Service Provider hizmeti ayarlama

Basit bir tek yönlü MQ Service Provider hizmeti ayarlamak için aşağıdaki adımları kullanın.

Başlamadan önce

Ensure that you have set up all the components correctly, as described in [z/OS Connect ve MQ Service Provider bağlantısını etkinleştirme](#) and [z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun](#)

Yordam

1. Hedef z/OS kuyruk yöneticisinde ONE_WAY_QUEUE adlı bir kuyruk yaratın ve MQSC ya da IBM MQ Explorer komutunu kullanın.
2. Bir IBM MQ ileti sistemi sağlayıcı bağlantı üreticisi ve bir kuyruk tanımlayın.

You do this by adding the following to the bottom of `server.xml`, but inside the `server` element.

```
-----  
<jmsConnectionFactory id="cf1" jndiName="jms/cf1" connectionManagerRef="ConMgr1">  
  <properties.wmqJms  
    transportType="BINDINGS"  
    queueManager="MQ21"/>  
</jmsConnectionFactory>  
  
<connectionManager id="ConMgr1" maxPoolSize="5"/>  
  
<jmsQueue id="q1" jndiName="jms/d1">  
  <properties.wmqJms  
    baseQueueName="ONE_WAY_QUEUE"/>  
</jmsQueue>  
-----
```

Notlar:

- a. **queueManager** özniteliğinin değerini, doğru hedef kuyruk yöneticisi adına çevirin.
 - b. Bağ tanımlarının bir **transportType** değeri kullanılır. Kuyruk yöneticisiyle iletişim kurmak için bir çapraz bellek bağlantısının kullanıldığı anlamına gelir. Bu, MQ Service Provider kullanıldığında desteklenen tek **transportType** ' dir.
3. Define a simple one-way MQ Service Provider service by adding the following to `server.xml`, but inside the `server` element.

```
-----  
<zosConnectService id="zosconnMQ1"-----  
  invokeURI="/oneWay"  
  serviceName="oneWay"  
  serviceRef="oneWay"/>  
  
<mqzOSConnectService id="oneWay"
```

```
connectionFactory="jms/cf1"
destination="jms/d1"/>
```

`zosConnectService` ögesi, `oneWay`'un `serviceName` ' u kullanarak z/OS Connect 'e yeni bir hizmet tanımlar. Sunucu:

- **invokeURI** özniteliği, hizmeti çağırmanın daha kolay olmasını sağlar.
- **serviceRef** özniteliği, bir z/OS Connect hizmet sağlayıcısının ID özniteiyle eşleşmelidir; bu durumda `mqzOSConnectService` ögesi bu durumda sağlanır.

`mqzOSConnectService` ögesi, MQ Service Providertarafından sağlanan tek bir hizmet eşgörünümünü tanımlar.

connectionFactory ve **destination** öznitelikleri, yönetim ortamına sırasıyla IBM MQ ileti alışverişi sağlayıcı bağlantı üreticinin ve kuyruğun nasıl bulunacağını anlatır.

Bu yapıdaki özniteliklere ilişkin ayrıntılar için bkz. [mqzOSConnectService ögesi](#) .

Sonuçlar

Basit bir tek yönlü hizmet kurdun.


Sonraki adım

Hizmeti test etmek gerekir.

İlgili görevler

“z/OS Connect V1üzerinde tek yönlü hizmeti test etme” sayfa 450

Tek yönlü hizmetinizin çalıştığından emin olmak için bir dizi adım.

 z/OS Connect V1üzerinde tek yönlü hizmeti test etme

Tek yönlü hizmetinizin çalıştığından emin olmak için bir dizi adım.

Başlamadan önce

“z/OS Connect V1üzerinde basit bir şekilde MQ Service Provider hizmeti ayarlama” sayfa 449 işlemini başarıyla tamamladığınızı doğrulayın.

Yordam

1. z/OS Connect 'in yeni hizmeti tanıdığından emin olun.

Bunu yapmak için, [z/OS Connect 'in doğru bir şekilde ayarlandığına bakın](#) başlıklı konuda ayrıntılı adımları yeniden yayınlayın.

Şu anda tanımlı bir hizmet olduğu için, aşağıdaki çıkışa benzer bir şey görmemiz gerekir:

```
-----
{
  "zosConnectServices": [
    {
      "serviceName": "oneWay",
      "serviceDescription": "DATA_UNAVAILABLE",
      "serviceProvider": "IBM MQ for z/OS service provider for IBM z/OS Connect" V1.0,
      "serviceURL": "https://yourdomainname:12342/zosConnect/services/oneWay"
    }
  ]
}
-----
```

Not: The **ServiceDescription** is returned as DATA_UNAVAILABLE because the **serviceDescription** attribute was not specified on the `zosConnectService` element.

2. Hizmete bir HTTP POST işlemi yayınlayın. Bu, POST 'ta belirtilen bilgi yükü her neyse, ONE_WAY_QUEUE' ye gönderilen yeni bir iletiyle sonuçlanır.

Bu tür bir testi gerçekleştirmenin iyi bir yolu, Google Chrome için [Advanced REST istemcisi \(ARC\)](#) fişi gibi bir şey kullanmaktan başka bir yöntem değildir.

a) Aşağıdaki URL 'ye karşı bir HTTP POST işlemi yapmak için ARC' yi kullanın:

```
https://HOST_NAME:HTTPS_PORT/oneWay
```

b) Gövde için geçerli bir JSON örneği belirtin; örneğin:

```
{"name1" : "value1", "name2" : "value2"}
```

İki üstbilgi belirtmeniz gerekir:

i) Yetkilendirme = Basic ENCODED_USERID_PASSWORD

Burada ENCODED_USERID_PASSWORD , Temel Kimlik Doğrulama Şeması' nda açıklandığı gibi, temel 64 kodlu kullanıcı kimliği ve parolasıdır.

ii) İçerik Tipi = application/json

İlk üstbilgi, z/OS Connect 'te oturum açmak için kullanılır ve [“z/OS Connect ve MQ Service Providerbağlantısını etkinleştirme” sayfa 447](#) içinde kullanılan kullanıcı kimliği ve parolasının kodlanmış bir biçimidir.

İkinci üstbilgi, sunucuya, MQ Service Provider' e HTTP POST çağrılarını verilirken desteklenen tek değer olan bilgi yükünün JSON olduğunu bildirir.

Bu başlığı belirtmezseniz, desteklenmeyen ortam tipi sonuçlarını belirten bir HTTP 415 hatası oluştu.

Tek yönlü hizmet sonuçlarına boş bir HTTP yanıt gövdesinde bir HTTP POST işlemi verilmesi ve 204 numaralı HTTP yanıt kodu döndürülmesi (içerik yok).

ARC, yanıtın herhangi bir veri içermediğini bildiren bir ileti görüntüler.

To validate that a message has been put, use a tool like the IBM MQ Explorer to browse the contents of ONE_WAY_QUEUE. Gönderilen JSON bilgi yükünün bulunduğu kuyruğun MQSTR biçiminde bir IBM MQ iletisi olması gerekir.

3. Aşağıdaki URL 'ye karşı HTTP DELETE komutu vermek için ARC' yi kullanın:

```
https://HOST_NAME:HTTPS_PORT/oneWay
```

Bu yıkıcı bir şekilde, sadece tek yönlü servise gönderilen mesajı alır.

Bu, ilk olarak gönderilen JSON bilgi yükünü içeren bir yanıt gövdesi vermelidir. İletinin imhamı geçici olarak aldığını doğrulamak için IBM MQ Explorer 'ı kullanabilirsiniz.

İlgili görevler

[“z/OS Connect V1üzerinde basit bir şekilde MQ Service Provider hizmeti ayarlama” sayfa 449](#)
Basit bir tek yönlü MQ Service Provider hizmeti ayarlamak için aşağıdaki adımları kullanın.

z/OS Connect Sürüm 1 üzerinde basit bir iki yönlü IBM MQ hizmeti ayarlama

z/OS Connect V1içinde, var olan bir çift IBM MQ kuyruğunda, MQ_REQUEST_Q ve MQ_REPLY_Q adı verilen bir çift yönlü IBM MQ hizmeti nasıl ayarladığınızı.

Başlamadan önce

Tüm bileşenleri doğru bir şekilde ayarladığınızdan emin olun ([“z/OS Connect ve MQ Service Providerbağlantısını etkinleştirme” sayfa 447](#) ve [“z/OS Connect 'in doğru bir şekilde ayarlandığından emin olun” sayfa 448](#)içinde açıklandığı gibi).

Bu görev hakkında

Bu, z/OS Connect kullanılarak gönderilen istek iletilerini tüketmek ve yanıt iletileri oluşturmak için bir arka uç uygulaması gerektirdiğinden, tek yönlü bir hizmeti ayarlamaya göre biraz daha fazla işe karışan bir hizmettir.

Bu görev, MQ_REQUEST_Q ve MQ_REPLAY_Q kuyruklarının var olan bir arka uç uygulaması tarafından kullanıldığını varsayar; örneğin, bir CICS ya da IMS işlemi, bir istek iletilisini MQ_REQUEST_Q ' dan alır ve MQ_REPLY_Q üzerine yerleştirilen bir yanıt iletilisi oluşturur.

Yordam

1. Bir IBM MQ ileti sistemi sağlayıcı bağlantı üreticisi ve z/OS Connect V1 sunucusuna iki IBM MQ ileti alışverişi sağlayıcısı kuyruğu tanımlayın.

IBM MQ ileti alışverişi sağlayıcısı kuyrukları var olan MQ_REQUEST_Q ve MQ_REPLY_Q kuyruklarıyla eşlenir. You do this by adding the following to the bottom of `server.xml`, but inside the `server` element.

```
<jmsConnectionFactory id="cf2" jndiName="jms/cf2"
  connectionManagerRef="ConMgr2">
  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21"/>
</jmsConnectionFactory>

<connectionManager id="ConMgr2" maxPoolSize="5"/>

<jmsQueue id="requestQueue" jndiName="jms/requestQueue">
  <properties.wmqJms baseQueueName="MQ_REQUEST_Q"/>
</jmsQueue>

<jmsQueue id="replyQueue" jndiName="jms/replyQueue">
  <properties.wmqJms baseQueueName="MQ_REPLY_Q"/>
</jmsQueue>
```

Notlar:

- a. **queueManager** özniteliğinin değerini, doğru hedef kuyruk yöneticisi adına çevirin.
 - b. Arka uç uygulamasının niteine bağlı olarak, bu öğelerin daha fazla yapılandırmasını gerçekleştirmeniz gerekebilir. Ek bilgi için [“MQ Service Provider’ in yapılandırılması” sayfa 453](#) başlıklı konuya bakın.
 - c. Koşut zamanlı istek sayısına bağlı olarak, **maxPoolSize** özniteliğinin ayarlanması gerekebilir. Bu öznitelige ilişkin ek bilgi için [JMS bağlantıları için bağlantı havuzlarının yapılandırılması](#) başlıklı konuya bakın.
2. Define the two-way IBM MQ service by adding the following to `server.xml`, but inside the `server` element.

```
-----
<zosConnectService id="zosconnMQ2"
  invokeURI="/twoWay"
  serviceName="twoWay"
  serviceRef="twoWay"/>

<mqzOSConnectService id="twoWay"
  connectionFactory="jms/cf2"
  destination="jms/requestQueue"
  replyDestination="jms/replyQueue"
  waitInterval="10000"/>
-----
```

Notlar:

- a. Arka uç uygulamasının niteine bağlı olarak, bu öğelerin daha fazla yapılandırmasını gerçekleştirmeniz gerekebilir. Örneğin, veri dönüştürmeye gereksinim duyabilirsiniz. Ek bilgi için [“MQ Service Provider’ in yapılandırılması” sayfa 453](#) başlıklı konuya bakın.
- b. Daha fazla bilgi için, **waitInterval** değerini ayarlamanız gerekebilir; bkz. **waitInterval**.

c. Bu örnekte, yanıt hedefine yerleştirilen iletilerin arka uç uygulaması tarafından, istek hedefindeki ileti tanıtıcısıyla aynı olan bir ilinti tanıtıcısına sahip olduğu varsayılır.

Böyle bir durumda değilse, **replySelection** özneliğini mqzOSConnectService ögesine eklemeniz ve değeri uygun bir şekilde ayarlamamız gerekir.

Ek bilgi için **replySelection** başlıklı konuya bakın.

Sonuçlar

You have set up a two-way service in z/OS Connect V1, that can be used with an existing backend application that makes use of a pair of IBM MQ queues.

Sonraki adım

You can invoke the two-way service issuing an HTTP POST command to the URL `https://host_name:port_no/twoWay`.

Bunu yapılacak adımlar, yalnızca HTTP POST komutu desteklendiğinden ve var olan arka uç uygulamasına uygun JSON bilgi yükü verilerini göndermeniz dışında, "[z/OS Connect V1 üzerinde tek yönlü hizmeti test etme](#)" sayfa 450' ta anlatılanlarla benzerdir.

İlgili görevler

"[z/OS Connect V1 üzerinde basit bir şekilde MQ Service Provider hizmeti ayarlama](#)" sayfa 449
Basit bir tek yönlü MQ Service Provider hizmeti ayarlamak için aşağıdaki adımları kullanın.

z/OS V 9.0.1 MQ Service Provider' in yapılandırılması

MQ Service Provider' u kullanmadan önce göz önünde bulundurmanız gereken çeşitli alanlar vardır.

Bu bölümde aşağıdakiler yer alır:

- "[MQ Service Provider kullanırken dikkat edilmesi gereken noktalar](#)" sayfa 453
- "[Using data transformations with the MQ Service Provider](#)" sayfa 456
- "[MQI uygulamalarıyla çalışma](#)" sayfa 461

z/OS V 9.0.1 MQ Service Provider kullanırken dikkat edilmesi gereken noktalar

MQ Service Provider için iki güvenlik açısından dikkat edilmesi gereken noktalar vardır.

- Belirli bir kullanıcının RESTful isteklerini, z/OS Connect tarafından gösterilen MQ Service Provider hizmetlerine gönderip göndermeyeceğini belirleyin.
- Belirli bir MQ Service Provider hizmetinin kuyruk yöneticisine ve kaynaklarına erişmeye yetkili olup olmadığı.

Belirli bir kullanıcının RESTful isteklerini z/OS Connect 'ten gösterilen MQ Service Provider hizmetlerine gönderip göndermeyeceği

This is controlled by configuration in the `server.xml` file of z/OS Connect.

z/OS Connect EE için bkz. [z/OS Connect EE güvenliğinin yapılandırılması](#).

z/OS Connect V1 için [z/OS Connect için güvenliğinin yapılandırılması](#) başlıklı konuya bakın.

Belirli bir MQ Service Provider hizmetine, kuyruk yöneticisine ve kaynaklarına erişme yetkisi verilir verilmediğini

MQ Service Provider temel olarak, bağ tanımlama kipi bağlantıları kullanan bir ya da daha çok kuyruk yöneticisine bağlanan WLP ile birlikte gelen IBM MQ ileti alışverişi sağlayıcısını temel alan bir JMS uygulamasıdır.

Sonuç olarak MQ Service Provider , bu özellikleri paylaşan uygulamalarla aynı şekilde güvenli hale getirilebilir. Bu konu, bu benzerlikleri ve farkları işaret ettikçe tanımlıyor.

MQ Service Provider bir bağ tanımları uygulaması olduğu için, bir kuyruk yöneticisine bağlanır ve bir kullanıcı kimliği ve isteğe bağlı parola sunar. Bunlar isteğe bağlı olarak Bağlantı kimlik doğrulaması kullanılarak doğrulanır.

Geçerliyse, kullanıcının kuyruk yöneticisinin güvenlik yapılandırmasına bağlı olarak bağlanmasına izin verilir. Daha fazla bilgi için z/OS üzerinde güvenliğin ayarlanması başlıklı konuya bakın.

Configuration attributes on the mqzOSConnectHizmet öğesi, and the properties.wmqJMS sub-element of the **jmsConnectFactory** element in server.xml, affect which user ID, and optional password, are presented to the queue manager.

Çeşitli olasılıklar aşağıdaki tabloda ayrıntılı olarak açıklanmıştır.



Uyarı: Güvenlik yapılandırması özellikleri arasında bir öncelik sırası vardır. **mqzOSConnectService** öznitelikleri **properties.wmqJms** özniteliklerini geçersiz kılar ve **useCallerPrincipal** özniteliği diğer tüm öznitelikleri geçersiz kılar.

Her iki parola özniteliği düz metin biçiminde ya da kodlanmış biçimde sağlanabilir. You should use the encoded format, as anyone with access to the server.xml can view the password in plain text.

Both versions of z/OS Connect ship with a tool called **securityUtility** which can be used to encode passwords. Ek bilgi için Liberty: securityUtility komutunun başlıklı konuya bakın.

mqzOSConnectHizmet öğesi		properties.wmqJms öğesi		Sonuç
useCallerBirincil Kullanıcı	Kullanıcı Adı ve Parola	userName	Parola	
ayarlama/yanlış	ayarlama/boşluk	ayarlam a/boşluk	ayarlam a/boşluk	z/OS Connect adres alanı ile ilişkili kullanıcı adı, yetki ve kimlik doğrulama amacıyla kuyruk yöneticisine sunulur. Parola sunulmadı.
ayarlama/yanlış	ayarlama/boşluk	küme	ayarlam a/boşluk	properties.wmqJms öğesinde kullanıcı adı, yetki ve kimlik doğrulama amacıyla kuyruk yöneticisine sunulur. Parola sunulmadı.
ayarlama/yanlış	ayarlama/boşluk	küme	küme	properties.wmqJms öğesindeki kullanıcı adı ve parola, yetki ve kimlik doğrulama amacıyla kuyruk yöneticisine sunulur.
ayarlama/yanlış	her iki değer kümesi	küme varsa yoksayılı r	küme varsa yoksayılı r	mqzOSConnectService öğesindeki kullanıcı adı ve parola, yetki ve kimlik doğrulama amacıyla kuyruk yöneticisine sunulur.

Çizelge 28. Hizmet yetkileri (devamı var)

mqzOSConnectHizmet öğesi		properties.wmqJms öğesi		Sonuç
doğru	küme varsa yoksayılr	küme varsa yoksayılr	küme varsa yoksayılr	z/OS Connect için kimliği doğrulanan kullanıcı birincil kullanıcısı alınır ve kullanıcı adı yetkilendirme ve kimlik doğrulama amacıyla kuyruk yöneticisine sunulur. Parola sunulmadı. z/OS Connect EE güvenliğinin yapılandırılmasına ilişkin bilgiler burada ¹ yer alıyor. z/OS Connect Sürüm 1 ile ilgili bilgiler burada ² .

Notlar:

1. [z/OS Connect EE için güvenlik yapılandırılması](#)
2. [z/OS Connect için güvenliğin yapılandırılması](#)

Örnekler

Bu, hem bağlantı üreticisinin hem de mqzOSConnectService ' in hem **userID** hem de **password** olarak belirtileceği tek yönlü bir hizmeti gösterir. Her iki durumda da şifreler kodlanır. mqzOSConnectService tanımlamasında tanımlı olan **userID** ve **password** kullanılır.

z/OS Connect V1

```
<zosConnectService
id="samplezOSConnectService1"
invokeURI="/samplezOSConnectService1"
serviceName="samplezOSConnectService1_name"
serviceRef="samplezOSConnectService1_MQ"/>

<mqzOSConnectService
id="samplezOSConnectService1_MQ"
connectionFactory="jms/sampleCF1"
userName="bill"
password="{aes}AJ+DdZ+1u0KEG5KIwUz4LvHBAQ8nTd3y8K8HAI+48Tt"
destination="jms/sampleQ1"/>

<jmsConnectionFactory
id="sampleCF1"
jndiName="jms/sampleCF1"
connectionManagerRef="sampleCF2ConnectionManager1">

<properties.wmqJms
transportType="BINDINGS"
queueManager="MQ21"
userName="matt"
password="{xor}GBMeEg9uERg=" />
</jmsConnectionFactory>

<jmsQueue
id="sampleQ1"
jndiName="jms/sampleQ1">

<properties.wmqJms
baseQueueName="SampleQ1"/>
</jmsQueue>
```

z/OS Connect EE

```
<zosconnect_zosConnectService
```

```

id="samplezOSConnectService1"
invokeURI="/samplezOSConnectService1"
serviceName="samplezOSConnectService1_name"
serviceRef="samplezOSConnectService1_MQ"/>

<mqzosconnect_mqzOSConnectService
id="samplezOSConnectService1_MQ"
connectionFactory="jms/sampleCF1"
userName="bill"
password="{aes}AJ+DdZ+1u0KEG5KIwUz4LvHBAQ8nTd3y8K8HAI+48Tt"
destination="jms/sampleQ1"/>

<jmsConnectionFactory
id="sampleCF1"
jndiName="jms/sampleCF1"
connectionManagerRef="sampleCF2ConnectionManager1">

<properties.wmqJms
transportType="BINDINGS"
queueManager="MQ21"
userName="matt"
password="{xor}GBMeEg9uERg="/>
</jmsConnectionFactory>

<jmsQueue
id="sampleQ1"
jndiName="jms/sampleQ1">

<properties.wmqJms
baseQueueName="SampleQ1"/>
</jmsQueue>

```

This example shows a one way service definition that uses the same JMS objects (the queue sampleQ1, and connection factory sampleCF1) as the first example. The principal that is authenticated to z/OS Connect is presented to the queue manager because **useCallerPrincipal="doğru"** is specified.

z/OS Connect V1

```

<zosConnectService
id="samplezOSConnectService2"
invokeURI="/samplezOSConnectService2"
serviceName="samplezOSConnectService2_name"
serviceRef="samplezOSConnectService2_MQ"/>

<mqzOSConnectService
id="samplezOSConnectService2_MQ"
connectionFactory="jms/sampleCF1"
destination="jms/sampleQ1"
useCallerPrincipal="true"/>

```

z/OS Connect EE

```

<zosconnect_zosConnectService
id="samplezOSConnectService2"
invokeURI="/samplezOSConnectService2"
serviceName="samplezOSConnectService2_name"
serviceRef="samplezOSConnectService2_MQ"/>

<mqzosconnect_mqzOSConnectService
id="samplezOSConnectService2_MQ"
connectionFactory="jms/sampleCF1"
destination="jms/sampleQ1"
useCallerPrincipal="true"/>

```

Using data transformations with the MQ Service Provider

z/OS Connect provides the ability to transform JSON data to an arbitrary format before calling a back-end z/OS asset, such as a CICS transaction, and transform the response from the z/OS asset back into JSON.

Bu yetenek, eklenebilir veri dönüştürme sağlayıcılar tarafından sağlanır. Yerleşik bir sağlayıcı, JSON ' un COBOL, PLI ya da C yapılarına dönüştürülmesini ya da bu yapılardan dönüştürülmesini sağlar.

MQ Service Provider , z/OS Connect veri dönüşümleriyle çalışır, ancak göz önünde bulundurulması gereken bazı dikkat edilmesi gereken noktalar vardır.

z/OS Connect veri dönüştürmeleri hakkında daha fazla bilgi

z/OS Connect Enterprise Edition için bkz. [Veri dönüştürücülerinin tanımlanması](#).

z/OS Connect Sürüm 1 için [z/OS Connect ileti bilgi yükü dönüşümlerinin tanımlanması](#) başlıklı konuya bakın.

Sending messages to IBM MQ

Bir MQ Service Provider hizmeti bir veri dönüştürmesiyle yapılandırıldığında ve bir kuyruk yöneticisine ileti gönderecekse (örneğin, JSON içeren bir HTTP POST işlemi almış olan tek yönlü ya da iki yönlü bir hizmetse), MQ Service Provider aşağıdaki adımları gerçekleştirir:

1. HTTP isteğinden JSON bilgi yükünü alır.
2. JSON ' dan bayt dizisine veri dönüştürme işlemi gerçekleştirilmek için bilgi yükünü z/OS Connect ' e geçirir.
3. Veri dönüşümünün sonucunu alır ve bunu JMS BytesMessage biçiminde kuyruk yöneticisine gönderir.

Varsayılan olarak, gönderilen iletinin boş bir MQMD **Format** alanı vardır. In many cases this is not appropriate, so you can set the `mqmdFormat` attribute in the `mqzOSConnectHizmet` ögesi to an appropriate value.

Yerleşik veri dönüştürme desteğinde her zaman 37 CCSID ile çıkış oluşturur. Bu bilgilerin MQMD `CodedCharSetId` alanında belirtilmesi gerekir; tersi durumda, kuyruktan alma işlemi iletinin kodunu çözemeyebilir. Bunu yapmak için, `mqzOSConnectService` hizmet ögesinin gönderme yaptığı kuyruktaki CCSID özneliğini ayarlayarak yapabilirsiniz.

The following example configuration illustrates appropriate configuration for a one-way service, that is going to be used to send messages to a queue called `SampleQ1`.

İletiler, MQMD `Format` alanı `BIÇIM` ve `CodedCharSetId` alanı 37 ile gönderilmek üzere gönderilecektir.

The `zosConnectDataXform` element tells z/OS Connect where to locate configuration for data transformations, and is referred to using the `dataXformRef` attribute of the `zosConnectService` element.

İki yönlü hizmet için yapılandırma için bkz. ["IBM MQ' dan ileti alma" sayfa 458](#).

```
<jmsConnectionFactory
  id="sampleCF1"
  jndiName="jms/sampleCF1"
  connectionManagerRef="sampleCF1ConnectionManager">

  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21" />
</jmsConnectionFactory>

<connectionManager
  id="sampleCF1ConnectionManager"
  maxPoolSize="5" />

<jmsQueue id="sampleQ1"
  jndiName="jms/sampleQ1">

  <properties.wmqJms
    baseQueueName="SampleQ1"
    CCSID="37" />
</jmsQueue>

<zosConnectService
  id="samplezOSConnectService1"
  invokeURI="/samplezOSConnectService1"
  serviceName="samplezOSConnectService1_name"
  serviceRef="samplezOSConnectService1_MQ"
  dataXformRef="xformJSON2Byte"/>
```

```

<mqzOSConnectService
  id="samplezOSConnectService1_MQ"
  connectionFactory="jms/sampleCF1"
  mqmdFormat="AFORMAT"
  destination="jms/sampleQ1" />

<zOSConnectDataXform id="xformJSON2Byte"
  bindFileLoc="/XFORM_ROOT/bindfiles" bindFileSuffix=".bnd"
  requestSchemaLoc="/XFORM_ROOT/json" requestSchemaSuffix=".json"
  responseSchemaLoc="/XFORM_ROOT/json"
  responseSchemaSuffix=".json"/>

```

IBM MQ' dan ileti alma

Bir MQ Service Provider yönetim ortamı bir veri dönüştürmesiyle yapılandırıldığında ve bir kuyruk yöneticisinden (örneğin, bir HTTP GET ya da DELETE işlemi için tek yönlü bir hizmet ya da HTTP POST ' u almış olan iki yönlü bir hizmetse) bir ileti alır. MQ Service Provider , aşağıdaki adımları gerçekleştirir.



Uyarı: İki yönlü hizmet kullanıyorsanız, hizmet [“Sending messages to IBM MQ” sayfa 457](#) içinde açıklanan adımları önceden gerçekleştirmiş olacaktır.

1. Kuyruktan iletiyi alır.
2. İletinin bir JMS **BytesMessage** ya da bir JMS **TextMessage** olduğunu denetler. İleti ikisi de değilse, bir hata üretilir ve çağıranlara döndürülür. "
3. Bir bayt dizisinden JSON ' a veri dönüştürme işlemi gerçekleştirmek için ileti bilgi yükünü z/OS Connect 'e geçirir.
4. Veri dönüşümünün sonucunu alır ve HTTP yönteminin yanıtı olarak döndürür.

Alınan ileti tipine bağlı olarak, fazladan bir yapılandırma gerçekleştirmeniz gerekebilir. Bunun nedeni, MQ Service Provider ' in alınan iletiyi yapılandırılan veri dönüşümünü aktarmak için doğru biçime dönüştürmesi gerektiğinden kaynaklanır.

Varsayılan veri dönüştürme işlemi, bilgi yükü 37 CCSID 'de olmasını bekler, ancak z/OS varlığı bu CCSID' de ileti oluşturulamayabilir.

Gerekten yapılandırma, **BytesMessage** ya da **TextMessage** 'un alınıp alınmadığına ve z/OS Connect V1ya da z/OS Connect EE' yi kullanıp kullanmadığına bağlıdır.

z/OS Connect V1üzerindeBytesMessage alındı

Bir **BytesMessage** alınacaksa, iletiyi almak için kullanılan kuyruk tanımlamasındaki **receiveConversion= "QMGR"** ve **receiveCCSID= "37"** özniteliklerini belirtebilirsiniz.

Bu, aşağıdaki örnekteki gösterilmektedir. Bu durumda, sampleQ2Receive tanımlamasında hem **receiveConversion** , hem de **receiveCCSID** öznitelikleri kümesi vardır.

```

<jmsConnectionFactory
  id="sampleCF2"
  jndiName="jms/sampleCF2"
  connectionManagerRef="sampleCF2ConnectionManager">

  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21" />
</jmsConnectionFactory>

<connectionManager
  id="sampleCF2ConnectionManager"
  maxPoolSize="5"/>

<jmsQueue id="sampleQ2Send"
  jndiName="jms/sampleQ2Send">

  <properties.wmqJms
    baseQueueName="SampleQ2Send"
    CCSID="37" />

```

```

</jmsQueue>
<jmsQueue id="sampleQ2Receive"
  jndiName="jms/sampleQ2Receive">
  <properties.wmqJms
    baseQueueName="SampleQ2Receive"
    receiveCCSID="37"
    receiveConversion="QMGR" />
</jmsQueue>

<zosConnectService
  id="samplezOSConnectService2"
  invokeURI="/samplezOSConnectService2"
  serviceName="samplezOSConnectService2_name"
  serviceRef="samplezOSConnectService2_MQ"
  dataXformRef="xformJSON2Byte" />

<mqzOSConnectService
  id="samplezOSConnectService2_MQ"
  connectionFactory="jms/sampleCF2"
  mqmdFormat="AFORMAT"
  destination="jms/sampleQ2Send"
  replyDestination="jms/sampleQ3Receive" />

<zosConnectDataXform id="xformJSON2Byte"
  bindFileLoc="/XFORM_ROOT/bindfiles" bindFileSuffix=".bnd"
  requestSchemaLoc="/XFORM_ROOT/json" requestSchemaSuffix=".json"
  responseSchemaLoc="/XFORM_ROOT/json"
  responseSchemaSuffix=".json" />

```

z/OS Connect EE ' deBytesMessage alındı

Bir **BytesMessage** alınacaksa, iletiyi almak için kullanılan kuyruk tanımlamasındaki **receiveConversion= "QMGR"** ve **receiveCCSID= "37"** özniteliklerini belirtebilirsiniz.

Bu, aşağıdaki örnekteki gösterilmektedir. Bu durumda, sampleQ2Receive tanımlamasında hem **receiveConversion** , hem de **receiveCCSID** öznitelikleri kümesi vardır.

```

<jmsConnectionFactory
  id="sampleCF2"
  jndiName="jms/sampleCF2"
  connectionManagerRef="sampleCF2ConnectionManager">
  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21" />
</jmsConnectionFactory>

<connectionManager
  id="sampleCF2ConnectionManager"
  maxPoolSize="5" />

<jmsQueue id="sampleQ2Send"
  jndiName="jms/sampleQ2Send">
  <properties.wmqJms
    baseQueueName="SampleQ2Send"
    CCSID="37" />
</jmsQueue>

<jmsQueue id="sampleQ2Receive"
  jndiName="jms/sampleQ2Receive">
  <properties.wmqJms
    baseQueueName="SampleQ2Receive"
    receiveCCSID="37"
    receiveConversion="QMGR" />
</jmsQueue>

<zosconnect_zosConnectService
  id="samplezOSConnectService2"
  invokeURI="/samplezOSConnectService2"
  serviceName="samplezOSConnectService2_name"
  serviceRef="samplezOSConnectService2_MQ"

```

```

dataXformRef="xformJSON2Byte"/>
<mqzosconnect_mqzOSConnectService
  id="samplezOSConnectService2_MQ"
  connectionFactory="jms/sampleCF2"
  mqmdFormat="AFORMAT"
  destination="jms/sampleQ2Send"
  replyDestination="jms/sampleQ3Receive"/>
<zosconnect_zosConnectDataXform id="xformJSON2Byte"
  bindFileLoc="/XFORM_ROOT/bindfiles" bindFileSuffix=".bnd"
  requestSchemaLoc="/XFORM_ROOT/json" requestSchemaSuffix=".json"
  responseSchemaLoc="/XFORM_ROOT/json"
  responseSchemaSuffix=".json"/>

```

z/OS Connect V1 üzerinde TextMessage alındı

Bir **TextMessage** alınacaksa, iletiyi varsayılan 37 ile veri dönüştürme işlemi tarafından beklenen CCSID 'ye dönüştürmeniz gerekir.

Özel bir veri dönüştürme kullanıyorsanız ve dönüştürme, 37 'den farklı bir CCSID beklerken, uygun bir CCSID belirlemek için [mqzOSConnectHizmet ögesi](#) içinde [receiveTextCCSID](#) özneliğini belirtebilirsiniz.

Bu, aşağıdaki örnekteki gösterilmektedir. Bu durumda örnek zOSConnectService3_MQ tanımı, [mqzOSConnectService element](#) özneliği için 1208 (UTF-8) değerine ayarlanmış [receiveTextCCSID](#) özneliğine sahiptir.

```

<jmsConnectionFactory
  id="sampleCF3"
  jndiName="jms/sampleCF3"
  connectionManagerRef="sampleCF3ConnectionManager">
  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21"/>
</jmsConnectionFactory>
<connectionManager
  id="sampleCF3ConnectionManager"
  maxPoolSize="5"/>
<jmsQueue id="sampleQ3Send"
  jndiName="jms/sampleQ3Send">
  <properties.wmqJms
    baseQueueName="SampleQ3Send"
    CCSID="37"/>
</jmsQueue>
<jmsQueue id="sampleQ3Receive"
  jndiName="jms/sampleQ3Receive">
  <properties.wmqJms
    baseQueueName="SampleQ3Receive"/>
</jmsQueue>
<zosConnectService
  id="samplezOSConnectService3"
  invokeURI="/samplezOSConnectService3"
  serviceName="samplezOSConnectService3_name"
  serviceRef="samplezOSConnectService3_MQ"
  dataXformRef="customDataXForm"/>
<mqzOSConnectService
  id="samplezOSConnectService3_MQ"
  connectionFactory="jms/sampleCF3"
  mqmdFormat="AFORMAT"
  destination="jms/sampleQ3Send"
  replyDestination="jms/sampleQ3Receive"
  receiveTextCCSID="1208"/>

```

z/OS Connect EE ' deTextMessage alındı

Bir **TextMessage** alınacaksa, iletiyi varsayılan 37 ile veri dönüştürme işlemi tarafından beklenen CCSID ' ye dönüştürmeniz gerekir.

Özel bir veri dönüştürme kullanıyorsanız ve dönüştürme, 37 'den farklı bir CCSID beklerken, uygun bir CCSID belirlemek için [mqzOSConnectHizmet ögesi](#) içinde [receiveTextCCSID](#) özneliğini belirtebilirsiniz.

Bu, aşağıdaki örnekteki gösterilmektedir. Bu durumda örnek zOSConnectService3_MQ tanımı, [mqzOSConnectService element özneliği](#) için 1208 (UTF-8) değerine ayarlanmış [receiveTextCCSID](#) özneliğine sahiptir.

```
<jmsConnectionFactory
  id="sampleCF3"
  jndiName="jms/sampleCF3"
  connectionManagerRef="sampleCF3ConnectionManager">

  <properties.wmqJms
    transportType="BINDINGS"
    queueManager="MQ21" />
</jmsConnectionFactory>

<connectionManager
  id="sampleCF3ConnectionManager"
  maxPoolSize="5" />

<jmsQueue id="sampleQ3Send"
  jndiName="jms/sampleQ3Send">

  <properties.wmqJms
    baseQueueName="SampleQ3Send"
    CCSID="37" />
</jmsQueue>

<jmsQueue id="sampleQ3Receive"
  jndiName="jms/sampleQ3Receive">

  <properties.wmqJms
    baseQueueName="SampleQ3Receive" />
</jmsQueue>

<zosconnect_zosConnectService
  id="samplezOSConnectService3"
  invokeURI="/samplezOSConnectService3"
  serviceName="samplezOSConnectService3_name"
  serviceRef="samplezOSConnectService3_MQ"
  dataXformRef="customDataXForm" />

<mqzosconnect_mqzOSConnectService
  id="samplezOSConnectService3_MQ"
  connectionFactory="jms/sampleCF3"
  mqmdFormat="AFORMAT"
  destination="jms/sampleQ3Send"
  replyDestination="jms/sampleQ3Receive"
  receiveTextCCSID="1208" />
```

MQI uygulamalarıyla çalışma

MQ Service Provider , IBM MQ' u kullanan var olan uygulamalara RESTful arabirimi sağlar. Bu uygulamalar, IBM MQ classes for JMS ya da Message Queue Interface (MQI) olanağını kullanarak IBM MQ ile etkileşimli çalışabilirler.

Varsayılan olarak IBM MQ classes for JMS , bir MQRFH2 üstbilgisi içeren iletileri gönderir. Ancak, çoğu MQI uygulaması MQRFH2 üstbilgilerini kullanmaz.

If an IBM MQ classes for JMS application interacts with an MQI application that has not been designed to work with MQRFH2 headers, configuration is required to prevent the IBM MQ classes for JMS sending messages containing an MQRFH2 header. Ek bilgi için [JMS iletilerinin IBM MQ iletilerine eşleme başlıklı konuya](#) bakın.

If the MQ Service Provider is going to send messages to queues, which will be consumed by MQI applications, that do not expect an MQRFH2 header you need to configure z/OS Connect to prevent an MQRFH2 header being sent.

Bu yapılandırmayı, `server.xml`'taki ilgili IBM MQ Messaging Provider kuyruğuna bir **targetClient="MQ"** özniteliğini ekleyerek elde edin.

The following sample configuration shows a one-way MQ Service Provider service with an **id** of *mqiService*. In turn, this has been configured to use the MQ messaging provider queue with an **id** of *mqiQueue*.

mqiQueue , **targetClient="MQ"** öznitelikle yapılandırılmıştır. Bu, bir uygulama söz konusu kuyruğa bir JMS iletisi gönderdiğinde (örneğin, bir uygulama *mqiService* için HTTP POST işlemi yayınlıyorsa), iletinin eklendiği bir MQRFH2 üstbilgisi olmaz.

z/OS Connect V1

```
<mqzOSConnectService
  id="mqiService"
  connectionFactory="jms/mqiCF"
  destination="jms/mqiQueue" />

<jmsQueue
  id="mqiQueue"
  jndiName="jms/mqiQueue">

  <properties.wmqJms
    baseQueueName="MQIQueue"
    targetClient = "MQ" />
</jmsQueue>
```

z/OS Connect EE

```
<mqzosconnect_mqzOSConnectService
  id="mqiService"
  connectionFactory="jms/mqiCF"
  destination="jms/mqiQueue" />

<jmsQueue
  id="mqiQueue"
  jndiName="jms/mqiQueue">

  <properties.wmqJms
    baseQueueName="MQIQueue"
    targetClient = "MQ" />
</jmsQueue>
```

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 sınamamıştır 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 sınanmamıştır. 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 (<http://www.eclipse.org/>) tarafından geliřtirilen yazılımları ierir.

Java ve Java tabanlı t÷m markalar ve logolar, Oracle firmasının ve/ya da iřtiraklerinin markaları ya da tescilli markalarıdır.



Parça numarası:

(1P) P/N: