

7.5

IBM WebSphere MQ ' nun denetlenmesi

IBM

Not

Bu bilgileri ve desteklediđi ürünü kullanmadan önce, "[Özel notlar](#)" sayfa 159 bölümündeki bilgileri okuyun.

Bu basım, yeni basımlarında tersi belirtilmediđi sürece, IBM® WebSphere MQ 'ın 7. yayın düzeyi 5 'i ve sonraki tüm yayın ve deđişiklik düzeyleri için geçerlidir.

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

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

İçindekiler

Yönetme.....	5
Yerel ve Uzak Denetim.....	7
IBM WebSphere MQ denetim komutları nasıl kullanılır?.....	8
Yönetim görevlerinin otomatikleştirilmesi.....	8
Programlanabilir Komut Biçimlerine Giriş.....	9
PCF 'lerin kullanımını kolaylaştırmak için MQAI' nin kullanılması.....	19
IBM WebSphere MQ Administration Interface (MQAI) ürününe giriş.....	19
IBM WebSphere MQ Yönetim Arabirimi (MQAI).....	20
IBM WebSphere MQ Gezgini 'ni kullanarak yönetim.....	55
IBM WebSphere MQ Explorer ile yapabildiğiniz.....	55
IBM WebSphere MQ Explorer 'ın kurulması.....	57
Windows üzerinde Güvenlik.....	62
IBM WebSphere MQ Explorer (yalnızca Windows ve Linux x86 platformlarının genişletmesi).....	66
IBM WebSphere MQ Taskbar uygulamasının kullanılması (yalnızca Windows).....	66
IBM WebSphere MQ uyarı izleme uygulaması (yalnızca Windows).....	66
Yerel IBM WebSphere MQ nesnelerini yönetme.....	67
Kuyruk yöneticisinin başlatılması ve durdurulması.....	67
Kuyruk yöneticilerinin el ile durdurulması.....	69
MQSC komutlarını kullanarak yerel denetim görevlerini gerçekleştirme.....	71
Kuyruk yöneticileriyle çalışılması.....	79
Yerel kuyruklarla çalışma.....	81
Diğer ad kuyruklarıyla çalışma.....	86
Model kuyruklarıyla çalışma.....	87
Yönetimle ilgili konularla çalışma.....	88
Aboneliklerle çalışma.....	91
Hizmetlerle çalışma.....	94
Tetikleme için nesnelere yönetme.....	100
Uzak IBM WebSphere MQ nesnelerini yönetme.....	102
Kanallar, kümeler ve uzak kuyruğa alma.....	102
Yerel kuyruk yöneticisinden uzak denetim.....	104
Uzak kuyruğun yerel tanımlaması yaratılması.....	110
Uzak kuyruk tanımlamalarının diğer adlar olarak kullanılması.....	112
Veri dönüştürme.....	113
IBM WebSphere MQ TelemetryYönetimi.....	114
Linux ve AIXüzerinde telemetri için kuyruk yöneticisi yapılandırılması.....	115
Windows üzerinde telemetri için kuyruk yöneticisi yapılandırılması.....	117
MQTT istemcilerine ileti göndermek için kuyruk yöneticisi yapılandır.....	118
İstemci Kimliği, yetkilendirme ve kimlik doğrulaması.....	121
SSL kullanan telemetri kanalı kimlik doğrulaması.....	127
SSL kullanarak yayın gizliliği.....	129
SSL Yapılandırması.....	130
JAAS yapılandırması.....	135
Aygıtlar kavramları içinIBM WebSphere MQ Telemetry cini.....	137
Çoklu yayını yönetme.....	148
Çok noktaya yayın ile çalışmaya başlama.....	148
IBM WebSphere MQ Multicast konu topolojisi.....	149
Çok Hedefli iletilerin boyutunu küçültür.....	150
Multicast ileti sistemi için veri dönüştürmenin etkinleştirilmesi.....	151
Çok noktaya yayın yönetimi ve izleme.....	152
Çok hedefli abonelik ileti geçmişi ayarlama.....	153
Gelişmiş çok noktaya gönderim görevleri.....	153
HP Integrity NonStop Sunucusunu Yönetme.....	156

Pathway 'den TMF/Gateway 'i el ile başlatma.....	156
Yol Yolu 'ndan TMF/Gateway durduruluyor.....	157
Özel notlar.....	159
Programlama arabirimi bilgileri.....	160
Ticari Markalar.....	160

Yönetme IBM WebSphere MQ

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 WebSphere MQ nesnelerini yerel olarak ya da uzaktan yönetebilirsiniz; bkz. [“Yerel ve Uzak Denetim” sayfa 7](#).

Kuyruk yöneticilerinizi ve ilgili kaynaklarını IBM WebSphere MQ' ta yaratmak ve denetlemek için kullanabileceğiniz çeşitli yöntemler vardır. Bu yöntemler, komut satırı arabirimlerini, grafik kullanıcı arabirimini ve bir denetim API 'sini içerir. Bu arabirimlerin her biriyle ilgili daha fazla bilgi için bu konudaki bölümlere ve linklere bakın.

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

- [“IBM WebSphere MQ denetim komutları” sayfa 5](#)
- [“IBM WebSphere MQ Script \(MQSC\) komutları” sayfa 5](#)
- [“Programlanabilir Komut Biçimleri \(PCF ' ler\)” sayfa 6](#)

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

- [“The IBM WebSphere MQ Explorer” sayfa 6](#)
- [“Windows Varsayılan Yapılanış uygulaması” sayfa 6](#)
- [“Microsoft Cluster Service \(MSCS\)” sayfa 7](#)

PCF komutlarını kullanarak hem yerel hem de uzak kuyruk yöneticilerine ilişkin bazı yönetim ve izleme görevlerini otomatikleştirebilirsiniz. Bu komutlar, bazı altyapılarda IBM WebSphere MQ Administration Interface (MQAI) kullanımı yoluyla da basitleştirilebilir. Yönetim görevlerini otomatikleştirme hakkında daha fazla bilgi için bkz. [“Yönetim görevlerinin otomatikleştirilmesi” sayfa 8](#).

IBM WebSphere MQ denetim komutları

Denetim komutları, kuyruk yöneticilerindeki denetim görevlerini gerçekleştirmenizi sağlar.

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

Denetim komutları [Kuyruk yöneticilerinin yaratılması ve yönetilmesi](#) başlıklı konu altında açıklanmıştır. Denetim komutlarına ilişkin komut başvurusu için bkz. [IBM WebSphere MQ Control komutları](#).

IBM WebSphere 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 üzerinde ayarlanan işaretlere bağlı olarak, `runmqsc` komutunu üç kipte çalıştırabilirsiniz:

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

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

MQSC commands are available on all platforms. MQSC komutları [Karşılaştırma komut kümelerinin karşılaştırılması](#) içinde özetlenir.

Windows, UNIX ya da Linux üzerinde, MQSC 'yi sistem komut satırında verilen tek komutlar olarak kullanabilirsiniz. Daha karmaşık ya da birden çok komut yayınlamak için, MQSC, Windows, UNIX ya da Linux sistem komut satırından çalıştırdığınız bir dosyaya oluşturulabilir. MQSC uzak bir kuyruk yöneticisine gönderilebilir. Tüm ayrıntılar için bkz. [MQSC başvurusu](#).

“Komut Dosyası (MQSC) Komutları” [sayfa 71](#) , her bir MQSC komutunun ve sözdiziminin bir açıklamasını içerir.

Yerel yönetimde MQSC komutlarının kullanılmasına ilişkin ek bilgi için [“MQSC komutlarını kullanarak yerel denetim görevlerini gerçekleştirme” sayfa 71](#) ' e bakın.

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 WebSphere 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. [“Programlanabilir Komut Biçimlerine Giriş” sayfa 9](#).

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

The IBM WebSphere MQ Explorer

IBM WebSphere 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 denetleyebilirsiniz.
- 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 WebSphere MQ Explorer by using the system menu, the MQExpLorer executable file, or the **strmqcfig** command.

Linux üzerinde, IBM WebSphere MQ Explorer olanağını başarıyla başlatmak için giriş dizininize bir dosya yazabilmeniz ve ana dizinin var olması gerekir.

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

Ek bilgi için [“IBM WebSphere MQ Explorer komutunu kullanarak yönetim” sayfa 55](#) başlıklı konuya bakın.

Windows Varsayılan Yapılanış uygulaması

You can use the Windows Default Configuration program to create a *başlangıç* (or default) set of IBM WebSphere MQ objects. Yaratılan varsayılan nesnelere bir özet [Çizelge 1: Windows varsayılan yapılanış uygulaması tarafından yaratılan nesnelere listesinde](#) yer almaktadır.

Microsoft Cluster Service (MSCS)

Microsoft Cluster Service (MSCS), sunucuları bir *kümeye* bağlamanızı, daha yüksek veri ve uygulama kullanılabilirliği vermenizi ve sistemin yönetilmesini kolaylaştırmanızı sağlar. MSCS, sunucu ya da uygulama hatalarını otomatik olarak algılayabilir ve bu hataları kurtarabilir.

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

IBM WebSphere MQ Kümeler

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

Bilgisayar grupları, birbirine bağlı ve bu şekilde yapılandırılmış bir şekilde, bir hata oluşursa, MSCS bir *hata durumunda yedek sisteme geçiş* gerçekleştirir, uygulamaların durum verilerini bozuk bilgisayardan kümedeki başka bir bilgisayara aktarmakta ve bu işlem burada yeniden başlatılabilmekte.

Microsoft Cluster Service (MSCS) 'nin desteklenmesi, MSCS' yi kullanmak üzere Windows sistemi için IBM WebSphere MQ ' nizin nasıl yapılandırılacağı hakkında ayrıntılı bilgi sağlar.

İlgili kavramlar

[WebSphere MQ teknik genel bilgileri](#)

[“Yerel IBM WebSphere MQ nesnelere yönetme” sayfa 67](#)

Bu bölümde, Message Queue Interface (MQI) olanağını kullanan uygulama programlarını desteklemek için yerel IBM WebSphere MQ nesnelere nasıl denetlemeniz gerekir. Bu bağlamda yerel yönetim, IBM WebSphere MQ nesnelere oluşturma, görüntüleme, değiştirme, kopyalama ve silme anlamına gelir.

[“Uzak IBM WebSphere MQ nesnelere yönetme” sayfa 102](#)

[Kişi XA kaynak yöneticisiyle kaybolduğunda dikkat edilecek noktalar](#)

İlgili görevler

[Planlama](#)

[Yapılandırılıyor](#)

İlgili başvurular

[Hareket desteği senaryoları](#)

Yerel ve Uzak Denetim

WebSphere MQ nesnelere yerel olarak ya da uzaktan yönetebilirsiniz.

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. WebSphere MQ' da, hiçbir kanal dahil olmadığından, bu iletişim işletim sistemi tarafından yönetildiği için bunu yerel yönetim olarak düşünebilirsiniz.

WebSphere MQ , *uzaktan yönetim* olarak bilinen bir iletişim noktasından yönetmeyi destekler. Bu, yerel sisteminizden, başka bir sistemde işlenen ve WebSphere MQ Explorer için de geçerli olan komutları yayınlamanıza olanak 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 oradan yayınlamanız ya da sizin için komutları yayınlatabilecek bir işlem yapmanız gerekir. Bu kısıtlama, WebSphere MQ Explorer için de geçerlidir.

[“Uzak IBM WebSphere MQ nesnelere yönetme” sayfa 102](#) , uzak yönetimin konusunu daha ayrıntılı olarak açıklar.

IBM WebSphere MQ denetim komutlarının kullanılması

Bu bölümde, IBM WebSphere MQ denetim komutlarının nasıl kullanılacağı ele alınmıştır.

Denetim komutlarını vermek istiyorsanız, kullanıcı kimliğinizin mqm grubunun bir üyesi olması gerekir. Bu konuda daha fazla bilgi için bkz. [Authority to administer WebSphere MQ on UNIX, Linux and Pencereler systems](#) . Buna ek olarak, ortama özgü aşağıdaki bilgileri de dikkate alın:

WebSphere MQ for Windows

Tüm denetim komutları bir komut satırından yayınlanabilir. Komut adları ve işaretleri büyük/küçük harfe duyarlı değildir: bunları büyük harf, küçük harf ya da büyük ve küçük harf birleşimine girebilirsiniz. Ancak, denetim komutları (kuyruk adları gibi) için bağımsız değişkenler büyük/küçük harfe duyarlıdır.

Sözdizimi açıklamalarında, tire işareti (-) bir işaret göstergesi olarak kullanılır. Tire işareti yerine eğik çizgi (/) kullanabilirsiniz.

WebSphere MQ for UNIX and Linux systems

Tüm WebSphere MQ denetim komutları bir kabuktan yayınlanabilir. Tüm komutlar büyük ve küçük harfe duyarlıdır.

Denetim komutlarının bir alt kümesi, IBM WebSphere MQ Explorer kullanılarak yayınlanabilir.

Ek bilgi için [WebSphere MQ denetim komutları](#) konusuna bakın.

Yönetim görevlerinin otomatikleştirilmesi

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, WebSphere MQ nesnelerini yönetme deneyiminiz olduğu varsayılır.

PCF komutları

WebSphere MQ programlanabilir komut biçimi (PCF) komutları, yönetim görevlerini bir denetim programına programlamak için kullanılabilir. Bu şekilde, bir programdan, kuyruk yöneticisi nesnelerini (kuyruklar, süreç tanımlamaları, ad listeleri, kanallar, istemci bağlantı kanalları, dinleyiciler, hizmetler ve kimlik doğrulama bilgileri nesneleri) yönlendirebilir ve hatta kuyruk yöneticilerini de yönlendirebilirsiniz.

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

Her PCF komutu, bir WebSphere MQ iletisinin uygulama verileri kısmına gömülü bir veri yapısıdır. Her komut, başka bir iletiyle aynı şekilde, MQPUT MQI işlevini kullanarak hedef kuyruk yöneticisine gönderilir. Komut sunucusu, iletiyi alan kuyruk yöneticinde çalıştırıyor; komut sunucusu bu iletiyi bir komut iletisi olarak yorumluyor ve komutu çalıştırır. Yanıtları almak için, uygulama bir MQGET çağrısı 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 iletisi yaratmak için gerekli olan bazı şeyler şunlardır:

İleti tanımlayıcısı

Bu, aşağıdaki standart bir WebSphere MQ ileti tanımlayıcısına sahip:

- İleti tipi (*MsqType*) 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).

PCF veri yapılarının tam açıklaması ve bunların nasıl gerçekleştirileceği için bkz. "[Programlanabilir Komut Biçimlerine Giriş](#)" sayfa 9.

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.

Programlanabilir Komut Biçimlerine Giriş

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

Bu ürün belgelerinde açıklanan Programlanabilir Komut Biçimleri aşağıdaki gibi desteklenmektedir:

- IBM WebSphere MQ for AIX
- IBM WebSphere MQ for HP-UX
- Linux için IBM WebSphere MQ
- Solaris için IBM WebSphere MQ
- IBM WebSphere MQ for Windows
- IBM WebSphere MQ for HP Integrity NonStop Server

Sorun PCF komutlarının çözmesi

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

İleti alışverişi ve kuyruğa alma ile ilgili denetim örnekleri şunlardır:

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

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

PF ' ler nedir?

PCF 'ler, bir program ile ağ içindeki herhangi bir kuyruk yöneticisi (PCF ' leri destekleyen) arasında deęiş tokuş edilebilen komut ve yanıt iletilerini tanımlar. You can use PCF commands in a systems management application program for administration of WebSphere 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 kuyruęu vardır ve uygulamanız PCF komut iletilerini o kuyruęa gönderebilir. Her kuyruk yöneticisinin, denetim kuyruęundan 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 kuyruęunuzu kullanarak uygulamanızı geri döndürülebilirler. PCF komutları ve yanıt iletileri, olaęan Message Queue Interface (MQI) kullanılarak gönderilir ve alınır.

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

Programlanabilir Komut Biçimlerinin Kullanılması

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

Bu bölümde aşağıdakiler yer alır:

- [“PCF komut iletileri” sayfa 10](#)
- [“Yanıtlar” sayfa 12](#)
- [IBM WebSphere MQ nesnelerinin adlandırılmasına ilişkin kurallar](#)
- [“PCF komutları için yetki denetimi” sayfa 14](#)

PCF komut iletileri

PCF komut iletileri, bir PCF üstbilgisinden, o üstbilgide tanımlanan deęiştirgelerden 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 dizgi deęiştirgesi. PCF üstbilgisi, komutu ve aynı iletide izleyen parametre yapılarının sayısını tanımlar. Her parametre yapısı komutta bir parametre sağlar.

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.

z/OS dışındaki platformlarda, PCF komutlarının gönderileceęi kuyruk her zaman SYSTEM.ADMIN.COMMAND.QUEUE.

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 çalıştığını doğrulayın.

Saęlanan üstbilgi dosyalarının bir listesi için, [WebSphere MQ COPY, HEADER, include and module files](#) başlıklı konuya bakın.

Bir PCF komutuna ilişkin ileti tanımlayıcısı

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

Bir PCF komut iletisi, ileti tanımlayıcısında aşağıdaki alanları içerir:

Report

Herhangi bir geçerli değer, gerektiği gibi.

MsgType

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

Expiry

Herhangi bir geçerli değer, gerektiği gibi.

Feedback

MQFB_NONE değerine ayarla

Encoding

Windows, UNIX ya da Linux sistemlerine gönderiyorsanız, bu alanı ileti verileri için kullanılan kodlamaya ayarlayın; gerekirse dönüştürme gerçekleştirilir.

CodedCharSetId

Gönderme yapmak için Windows, UNIX ya da Linux sistemleri, bu alanı ileti verileri için kullanılan kodlanmış karakter takımı tanıtcısı olarak ayarlar; gerekirse dönüştürme gerçekleştirilir.

Format

MQFMT_ADMIN değerine ayarlayın.

Priority

Herhangi bir geçerli değer, gerektiği gibi.

Persistence

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

MsgSeqNumber

1 olarak ayarla

Offset

0 olarak ayarla

MsgFlags

MQMF_NONE değerine ayarla

OriginalLength

MQOL_UNDEFINED değerine ayarla

Kullanıcı Verilerinin Gönderilmesi

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

PCF iletilerinin belirlenen bir kuyruğa gönderilmesi ve alınması

PCF iletileri belirtilen bir kuyruğa gönderiliyor

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

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](#).

Yanıtlar

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 WebSphere MQ Pencereler Explorer is one such application (however, the Version 6.0 or later IBM WebSphere 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 (Programlanabilir Komut Biçimlerinin Tanımlamaları) *her zaman iade* olarak gösterilir. Bu nedenle, bu yanıtlar için, verilerin hangi nesneye uygulanacağı saptanması için yanıtta nesnelerin adının belirlenmesi gerekir.

Bir yanıt için ileti tanımlayıcısı

Yanıt iletisi, ileti tanımlayıcısında aşağıdaki alanları içerir:

MsgType

Bu alan MQMT_REPLY ' dir.

MsgId

Bu alan kuyruk yöneticisi tarafından oluşturulur.

CorrelId

Bu alan, komut iletisine ilişkin rapor seçeneklerine göre oluşturulur.

Format

Bu alan, MQFMT_ADMIN 'dir.

Encoding

MQENC_NATIVE değerine ayarlayın.

CodedCharSetId

MQCCSI_Q_MGR olarak ayarlayın.

Persistence

Komut iletisinde olduğu gibi.

Priority

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

- MQIACF_PARAMETER_ID

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

- MQIACF_ERROR_ID

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

- MQIACF_SELECTOR

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

- MQIACF_ERROR_OFFSET

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

- MQIA_CODED_CHAR_SET_ID

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

Son (ya da yalnızca) hata yanıtı iletisi bir özet yanıtıdır; *CompCode* alanı MQCC_FAILED ve bir *Reason* alanı MQRCFC_COMMAND_FAILED (başarısız). Bu ileti, üstbilgiyi izleyen bir parametre yapısına sahip değil.

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 sırada döndürülmekte olan bu ek parametre yapılarına bağlı kalmamalıdır.

PCF komutları için yetki denetimi

Bir PCF komutu işlendiğinde, gerekli WebSphere MQ nesne yetkisi denetimleri için komut iletisinde ileti tanımlayıcısından *UserIdentifier* 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.

IBM WebSphere MQ for Pencereler, UNIX and Linux systems



Herhangi bir PCF komutunun işlenmesi için, hedef sistemdeki kuyruk yöneticisi nesnesi için kullanıcı kimliği *dsp* yetkisine sahip olmalıdır. In addition, WebSphere MQ object authority checks are performed for certain PCF commands, as shown in Çizelge 1 sayfa 16.

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

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

- Kanalı Değiştir
- Kanalı Kopyala
- Kanal Yarat
- Kanalı Sil
- Ping Kanalı
- Kanalı Sıfırla
- Başlangıç Kanalı
- Kanalı Durdur
- Kanal Başlatıcısını Başlat
- Kanal Dinleyicisi Başlat
- Kanalı Çözümle
- Kümeyi Sıfırla
- Kümeyi Yenile
- Kuyruk Yöneticisini Askıya Al
- Kuyruk Yöneticisini Sürdür

WebSphere MQ for HP Integrity NonStop Server

Herhangi bir PCF komutunun işlenmesi için, hedef sistemdeki kuyruk yöneticisi nesnesi için kullanıcı kimliği *dsp* yetkisine sahip olmalıdır. In addition, IBM WebSphere MQ object authority checks are performed for certain PCF commands, as shown in [Çizelge 1 sayfa 16](#).

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

- Kanalı Değiştir
- Kanalı Kopyala
- Kanal Yarat
- Kanalı Sil
- Ping Kanalı
- Kanalı Sıfırla
- Başlangıç Kanalı
- Kanalı Durdur
- Kanal Başlatıcısını Başlat
- Kanal Dinleyicisi Başlat
- Kanalı Çözümle
- Kümeyi Sıfırla
- Kümeyi Yenile
- Kuyruk Yöneticisini Askıya Al
- Kuyruk Yöneticisini Sürdür

WebSphere MQ Nesnesi yetkileri

<i>Çizelge 1. Windows, HP Integrity NonStop Server, UNIX and Linux sistemleri-nesne yetkileri</i>		
Komut	WebSphere MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
Kimlik Doğrulama Bilgilerini Değiştir	dsp ve chg	yok
Kanalı Değiştir	dsp ve chg	yok
Kanal Dinleyiciyi Değiştir	dsp ve chg	yok
İstemci Bağlantı Kanalını Değiştir	dsp ve chg	yok
Ad listesini değiştir	dsp ve chg	yok
Süreci Değiştir	dsp ve chg	yok
Kuyruğu Değiştir	dsp ve chg	yok
Kuyruk Yöneticisini Değiştir	chg <i>bkz. Not 3 ve Not 5</i>	yok
Hizmeti Değiştir	dsp ve chg	yok
Kuyruğu Temizle	klr	yok
Kimlik Doğrulama Bilgilerini Kopyala	dsp	crt
Kimlik Doğrulama Bilgilerini Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
Kanalı Kopyala	dsp	crt
Kanal Kopyala (Değiştir) <i>bkz. Not 1</i>	<i>from: dsp to: chg</i>	crt
Kanal Dinleyicisi Kopyala	dsp	crt
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

<i>Çizelge 1. Windows, HP Integrity NonStop Server, UNIX and Linux sistemleri-nesne yetkileri (devamı var)</i>		
Komut	WebSphere MQ nesne yetkisi	Sınıf yetkisi (nesne tipi için)
Kanal Yarat	(sistem varsayılan kanalı) dsp	crt
Kanal Yarat (Değiştir) <i>bkz. Not 1</i>	(sistem varsayılan kanalı) dsp to: chg	crt
Kanal Dinleyicisi Yarat	(sistem varsayılan dinleyicisi) dsp	crt
Kanal Dinleyici Yarat (Değiştir) <i>bkz. Not 1</i>	(sistem varsayılan dinleyicisi) dsp to: chg	crt
İstemci Bağlantı Kanalı Oluştur	(sistem varsayılan kanalı) dsp	crt
İstemci Bağlantı Kanalı Oluştur (Değiştir) <i>bkz. Not 1</i>	(sistem varsayılan kanalı) dsp to: chg	crt
Ad Listesi Yarat	(sistem varsayılan ad listesi) dsp	crt
Ad Listesi Yarat (Değiştir) <i>bkz. Not 1</i>	(sistem varsayılan ad listesi) dsp to: dsp ve chg	crt
Süreç Yarat	(sistem varsayılan süreci) dsp	crt
Süreç Yarat (Değiştir) <i>bkz. Not 1</i>	(sistem varsayılan süreci) dsp to: chg	crt
Kuyruk Yarat	(sistem varsayılan kuyruğu) dsp	crt
Kuyruk Yarat (Değiştir) <i>bkz. Not 1</i>	(sistem varsayılan kuyruğu) dsp to: dsp ve chg	crt
Hizmet oluştur	(sistem varsayılan kuyruğu) dsp	crt
Hizmet Oluştur (Değiştir) <i>bkz. Not 1</i>	(sistem varsayılan kuyruğu) dsp to: chg	crt
Kimlik Doğrulama Bilgilerini Sil	dsp ve dlt	yok
Yetki Kaydını Sil	(kuyruk yöneticisi nesnesi) chg <i>bkz. Not 4</i>	<i>bkz. Not 4</i>
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	<i>bkz. Not 4</i>	<i>bkz. Not 4</i>
Kanal Sorgula	dsp	yok
Kanal İletişimcisi ile	dsp	yok
Sorgu Kanalı Durumu (ChannelType MQCHT_CLSSDR için)	inq	yok

Çizelge 1. Windows, HP Integrity NonStop Server, UNIX and Linux sistemleri-nesne yetkileri (devamı var)

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

Notlar:

1. Değiştirilecek nesne varsa, bu komut geçerli olur; tersi durumda, yetki denetimi Create ya da Copy withod Replace gibi olur.
2. Gereken yetki, çıkış metninin tanımladığı MQSC komutu tarafından belirlenir ve önceki komutlardan birine eşdeğerdir.
3. Herhangi bir PCF komutunu işleyebilmek için, kullanıcı kimliğinin hedef sistemdeki kuyruk yöneticisi nesnesine ilişkin dsp yetkisi olmalıdır.
4. Bu PCF komutu, komut sunucusu -a değiştirgesiyle başlatılmadıkça yetkilendirilir. Varsayılan olarak komut sunucusu, Kuyruk Yöneticisi başlatıldığında ve -a değiştirgesi olmadan başlar. Ek bilgi için System Administration Guide (Sistem Yönetimi Kılavuzu) belgesine bakın.

5. Kuyruk yöneticisi için bir kullanıcı kimliği *chg* yetkisi verilmesi, tüm gruplar ve kullanıcılar için yetki kayıtlarını belirleme yeteneği verir. Bu yetkiyi sıradan kullanıcılara ya da uygulamalara atamayın.

WebSphere MQ , güvenlik denetimi için kendi kullanıcı çıkış programlarınızı sağlayabilmeniz için bazı kanal güvenliği çıkış noktaları da sağlar. Ayrıntılar [Kanal görüntüleme elkitabında](#) verilmiştir.

PCF 'lerin kullanımını kolaylaştırmak için MQAI' nin kullanılması

MQAI, AIX, HP-UX, IBM i, Linux, Solaris ve Windows platformlarında bulunan WebSphere MQ ' a yönetim arabirimidir.

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 kullanarak aşağıdaki yöntemleri kullanın:

PCF iletilerinin kullanımını kolaylaştırmak için

MQAI, WebSphere MQ' u yönetmenin kolay bir yoludur; karmaşık veri yapılarıyla ilişkili sorunlardan kaçınarak kendi PCF iletilerinizi yazmanıza gerek yoktur.

MQI çağrılarını kullanarak yazılmış programlardaki parametreleri geçirmek için, PCF iletisi dizginin ya da tamsayı verilerinin adını ve ayrıntılarını içermelidir. Bunu yapmak için, programınızda her yapı için birkaç deyim ve bellek alanı ayrılmalıdır. Bu görev uzun ve zahmetli olabilir.

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

Hata koşullarını daha kolay ele almak için

PCF komutlarından geri dönüş kodları almak zordur, ancak MQAI programı, programın hata koşullarını işlemlerini kolaylaştırır.

Veri çantasını oluşturduktan ve doldurduktan sonra, herhangi bir yanıt iletisi için bekleyen mqExecute çağrısını kullanarak, bir kuyruk yöneticisinin komut sunucusuna bir yönetim komut iletisi gönderebilirsiniz. mqExecute çağrısı, komut sunucusu ile deęişimi gerçekleştirir ve yanıtları bir *yanıt çantası*na döndürür.

MQAI ile ilgili daha fazla bilgi için bkz. [“IBM WebSphere MQ Denetim Arabirimi 'ne Giriş \(MQAI\)” sayfa 19.](#)

IBM WebSphere MQ Denetim Arabirimi 'ne Giriş (MQAI)

IBM WebSphere MQ Denetim Arabirimi (MQAI), IBM WebSphere MQ için bir programlama arabirimidir. Nesneleri (ya da deęiřtirgeleri) Programlanabilir Komut Biçimleri (PCF ' ler) kullanmaktan daha kolay bir biçimde işlemek için, veri torbalarını kullanarak bir IBM WebSphere MQ kuyruk yöneticisinde yönetim görevlerini gerçekleştirir.

MQAI kavramları ve terminolojisi

MQAI, C dilini ve Pencereler için Visual Basic kullanarak, WebSphere MQ' a bir programlama arabirimidir. z/OS dışındaki platformlarda kullanılabilir.

Veri torbalarını kullanarak bir WebSphere MQ kuyruk yöneticisindeki yönetim görevlerini gerçekleştirir. Veri torbaları, nesnelerin özelliklerini (ya da parametrelerini), dięer yönetim arabirimini, Programlanabilir Komut Biçimlerini (PCF ' ler) kullanmaktan daha kolay bir şekilde işlemenize olanak sağlar. MQAI, PCF ' lerin MQGET ve MQPUT çağrılarını kullanmaktan daha kolay kullanılmasını sağlar.

Veri torbalarıyla ilgili daha fazla bilgi için bkz. [“Veri torbaları” sayfa 45.](#) PCF ' ler hakkında daha fazla bilgi için bkz. [“Programlanabilir Komut Biçimlerine Giriş” sayfa 9.](#)

MQAI ' nin kullanılması

MQAI ' yi aşağıdaki gibi kullanabilirsiniz:

- PCF iletilerinin kullanımını basitleştirin. MQAI, WebSphere MQ' u yönetmenin kolay bir yoludur; kendi PCF iletilerinizi yazmanıza ve bu nedenle karmaşık veri yapılarıyla ilgili sorunlardan kaçınmanıza gerek yoktur.
- Hata koşullarını daha kolay ele al. It is difficult to get return codes back from the WebSphere MQ script (MQSC) commands, but the MQAI makes it easier for the program to handle error conditions.
- Uygulamalar arasında veri alışverişi. Uygulama verileri, PCF biçiminde gönderilir ve MQAI tarafından paketlenmez ve paketlenmez. İleti verileriniz tamsayılar ve karakter dizgilerinden oluşuyorsa, PCF verileri için WebSphere MQ yerleşik veri dönüştürmesinden yararlanmak için MQAI ' yi kullanabilirsiniz. Bu, veri dönüştürme çıkışlarını yazma gereksinmesini önler. For more information on using MQAI to administer WebSphere MQ and to exchange data between applications, see [“PCF 'lerin kullanımını kolaylaştırmak için MQAI' nin kullanılması” sayfa 19.](#)

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 21](#)
2. Ekrandaki olayları basit bir olay izleyicisini kullanarak görüntüler. [“Olay izleyicisini kullanarak olayları görüntülemek için örnek C programı \(amqsaiem.c\)” sayfa 24](#)
3. Tüm yerel kuyrukların ve bunların yürürlükteki derinliklerinin listesini yazdırır. [“Kuyruklar ve yazdırma bilgileri hakkında bilgi almak için örnek C programı \(amqsailq.c\)” sayfa 36](#)
4. Tüm kanalların ve tiplerinin bir listesini yazdırır. [“Kanal nesnelere hakkında bilgi almak için örnek C programı \(amqsaicl.c\)” sayfa 31](#)

MQAI uygulamanızı oluşturma

Uygulamanızı MQAI kullanarak oluşturmak için, WebSphere MQ' da olduğu gibi aynı kitaplıklara bağlan. WebSphere MQ uygulamalarınızı nasıl oluşturabilmeye ilişkin bilgi için bkz. [Bir WebSphere MQ uygulaması oluşturulması .](#)

MQAI kullanarak WebSphere MQ ' un yapılandırılmasına ilişkin ipuçları

MQAI, komut sunucusuyla doğrudan ilgilenmek yerine, denetim komutlarını komut sunucusuna göndermek için PCF iletilerini kullanır. Tips for configuring WebSphere MQ using the MQAI can be found in [“IBM WebSphere MQ' u yapılandırmaya ilişkin ipuçları ve öneriler” sayfa 40](#)

IBM WebSphere MQ Denetim Arabirimi (MQAI)

IBM WebSphere MQ for Windows, AIX, Linux, HP-UX ve Solaris, IBM WebSphere MQ Administration Interface (MQAI) özelliğini destekler. MQAI, PCF 'leri göndermek ve almak için MQI' ye alternatif bir programlama arabirimi olan IBM WebSphere MQ programlama arabirimidir.

MQAI, nesnelere (ya da değiştiricileriyle) MQAI yoluyla doğrudan PCF 'leri kullanmaktan daha kolay bir şekilde işlemenizi sağlayan *veri torbaları* ' ı kullanır.

MQAI, parametreleri veri torbasına geçirerek PCF iletilerine daha kolay programlama erişimi sağlar, böylece her yapı için tek bir deyim gereklidir. Bu erişim, programcının dizileri işleme ve depolama alanı ayırma gereksinmesini ortadan kaldırır ve PCF ' nin ayrıntılarından biraz yalıtım sağlar.

The MQAI administers WebSphere MQ by sending PCF messages to the command server and waiting for a response.

MQAI, bu elkitabının ikinci kısmında açıklanmıştır. MQAI ' ye bir bileşen nesne modeli arabiriminin açıklaması için [Java 'nın kullanılması](#) belgelerine bakın.

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

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

```

/*****
/*
/* Program name: AMQSAICQ.C
/*
/* Description: Sample C program to create a local queue using the
/*               WebSphere MQ Administration Interface (MQAI).
/*
/* Statement:    Licensed Materials - Property of IBM
/*
/*               84H2000, 5765-B73
/*               84H2001, 5639-B42
/*               84H2002, 5765-B74
/*               84H2003, 5765-B75
/*               84H2004, 5639-B43
/*
/*               (C) Copyright IBM Corp. 1999, 2024.
/*
*****/
/*
/* Function:
/* AMQSAICQ is a sample C program that creates a local queue and is an
/* example of the use of the mqExecute call.
/*
/* - The name of the queue to be created is a parameter to the program.
/*
/* - A PCF command is built by placing items into an MQAI bag.
/*   These are:-
/*     - The name of the queue
/*     - The type of queue required, which, in this case, is local.
/*
/* - The mqExecute call is executed with the command MQCMD_CREATE_Q.
/*   The call generates the correct PCF structure.
/*   The call receives the reply from the command server and formats into
/*   the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/*   is a failure from the command server then the code returned by the
/*   command server is retrieved from the system bag that is
/*   embedded in the response bag to the mqExecute call.
/*
/* Note: The command server must be running.
/*
*****/
/*
/* AMQSAICQ has 2 parameters - the name of the local queue to be created
/* - the queue manager name (optional)
/*
*****/
/*****
/* Includes
*****/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h>           /* MQI           */
#include <cmqcfh.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 WebSphere 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(MQCB0_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(MQCB0_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, /* WebSphere 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: <stimqcsv 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
          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 izleyicisini kullanarak olayları görüntülemek için örnek C programı (amqsaieem.c)

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

```

/*****
/*
/* Program name: AMQSAIEM.C
/*
/* Description: Sample C program to demonstrate a basic event monitor
/*              using the WebSphere MQ Admin Interface (MQAI).
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 1999, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/*****

```



```

/*                                                                    */
/* Function:                                                            */
/* AMQSAIEM is a sample C program that demonstrates how to write a simple */
/* event monitor using the mqGetBag call and other MQAI calls.          */
/*                                                                    */
/* The name of the event queue to be monitored is passed as a parameter */
/* to the program. This would usually be one of the system event queues:- */
/* SYSTEM.ADMIN.QMGR.EVENT      Queue Manager events                  */
/* SYSTEM.ADMIN.PERFM.EVENT     Performance events                   */
/* SYSTEM.ADMIN.CHANNEL.EVENT   Channel events                      */
/* SYSTEM.ADMIN.LOGGER.EVENT    Logger events                       */
/*                                                                    */
/* To monitor the queue manager event queue or the performance event queue, */
/* the attributes of the queue manager needs 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 <cmqcfh.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, &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 nesneleri hakkında bilgi almak için örnek C programı (amqsaicl.c)

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

```

/*****
/*
/* Program name: AMQSAICL.C
/*
/* Description: Sample C program to inquire channel objects
/*              using the WebSphere MQ Administration Interface (MQAI)
/*
/* <N_OCO_COPYRIGHT>
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 2008, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* <NOC_COPYRIGHT>
/*****
/*
/* Function:
/* AMQSAICL is a sample C program that demonstrates how to inquire
/*

```

```

/* attributes of the local queue manager using the MQAI interface. In */
/* particular, it inquires all channels and their types. */
/* */
/* - A PCF command is built from items placed into an MQAI administration */
/* bag. */
/* These are:- */
/* - The generic channel name "*" */
/* - The attributes to be inquired. In this sample we just want */
/* name and type attributes */
/* */
/* - The mqExecute MQCMD_INQUIRE_CHANNEL call is executed. */
/* The call generates the correct PCF structure. */
/* The default options to the call are used so that the command is sent */
/* to the SYSTEM.ADMIN.COMMAND.QUEUE. */
/* The reply from the command server is placed on a temporary dynamic */
/* queue. */
/* The reply from the MQCMD_INQUIRE_CHANNEL is read from the */
/* temporary queue and formatted into the response bag. */
/* */
/* - The completion code from the mqExecute call is checked and if there */
/* is a failure from the command server, then the code returned by the */
/* command server is retrieved from the system bag that has been */
/* embedded in the response bag to the mqExecute call. */
/* */
/* Note: The command server must be running. */
/* */
/*****
/* AMQSAICL has 2 parameter - the queue manager name (optional) */
/* - output file (optional) default varies */
*****/

/*****
/* Includes */
*****/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#if (MQAT_DEFAULT == MQAT_OS400)
#include <recio.h>
#endif

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

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

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

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

```



```

const struct
{
  char name[9];
} ChlTypeMap[9] =
{
  "sdr      ", /* MQCHT_SENDER */
  "svr      ", /* MQCHT_SERVER */
  "rcvr     ", /* MQCHT_RECEIVER */
  "rqstr    ", /* MQCHT_REQUESTER */
  "all      ", /* MQCHT_ALL */
  "cltconn  ", /* MQCHT_CLNTCONN */
  "svrcn    ", /* MQCHT_SVRCONN */
  "clusrcvr ", /* MQCHT_CLUSRCVR */
  "clusdr   ", /* MQCHT_CLUSSDR */
};
#endif

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

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

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

#endif

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

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

  /*****
  /* Connect to the queue manager */
  *****/
  if (argc > 1)

```

```

    strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(qmName, &hConn, &compCode, &connReason);

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

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

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

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

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

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

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

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

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

/*****
/* Check the command server is started. If not exit. */
*****/
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{

```

```

printf("Please start the command server: <strmqcsv QMgrName="">\n");
goto MOD_EXIT;
}

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

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

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

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

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

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

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

```

```

    }
}

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

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

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

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

return 0;
}

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

```

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

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

```

/*****
/*
/* Program name: AMQSAILQ.C
/*
/* Description: Sample C program to inquire the current depth of the local
/* queues using the WebSphere MQ Administration Interface (MQAI)
/*
/* Statement: Licensed Materials - Property of IBM
*/

```

```

/*
/*      84H2000, 5765-B73
/*      84H2001, 5639-B42
/*      84H2002, 5765-B74
/*      84H2003, 5765-B75
/*      84H2004, 5639-B43
/*
/*      (C) Copyright IBM Corp. 1999, 2024.
/*
/*****
/*
/* Function:
/* AMQSAILQ is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires the current depths of all the local queues.
/*
/* - A PCF command is built by placing items into an MQAI administration
/* bag.
/* These are:-
/* - The generic queue name "*"
/* - The type of queue required. In this sample we want to
/* inquire local queues.
/* - The attribute to be inquired. In this sample we want the
/* current depths.
/*
/* - The mqExecute call is executed with the command MQCMD_INQUIRE_Q.
/* The call generates the correct PCF structure.
/* The default options to the call are used so that the command is sent
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.
/* The reply from the command server is placed on a temporary dynamic
/* queue.
/* The reply from the MQCMD_INQUIRE_Q command is read from the
/* temporary queue and formatted into the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/* is a failure from the command server, then the code returned by
/* command server is retrieved from the system bag that has been
/* embedded in the response bag to the mqExecute call.
/*
/* - If the call is successful, the depth of each local queue is placed
/* in system bags embedded in the response bag of the mqExecute call.
/* The name and depth of each queue is obtained from each of the bags
/* and the result displayed on the screen.
/*
/* Note: The command server must be running.
/*
/*****
/* AMQSAILQ has 1 parameter - the queue manager name (optional)
/*
/*****

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

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

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

/*****
/* Function: main
/*****
int main(int argc, char *argv[])
{
    /*****
    /* MQAI variables
    /*****
    MQHCONN hConn;          /* handle to WebSphere 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, /* WebSphere 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);
}

```

IBM WebSphere MQ' u yapılandırmaya ilişkin ipuçları ve öneriler

MQAI olanağını kullanırken ipuçları ve öneriler programlama.

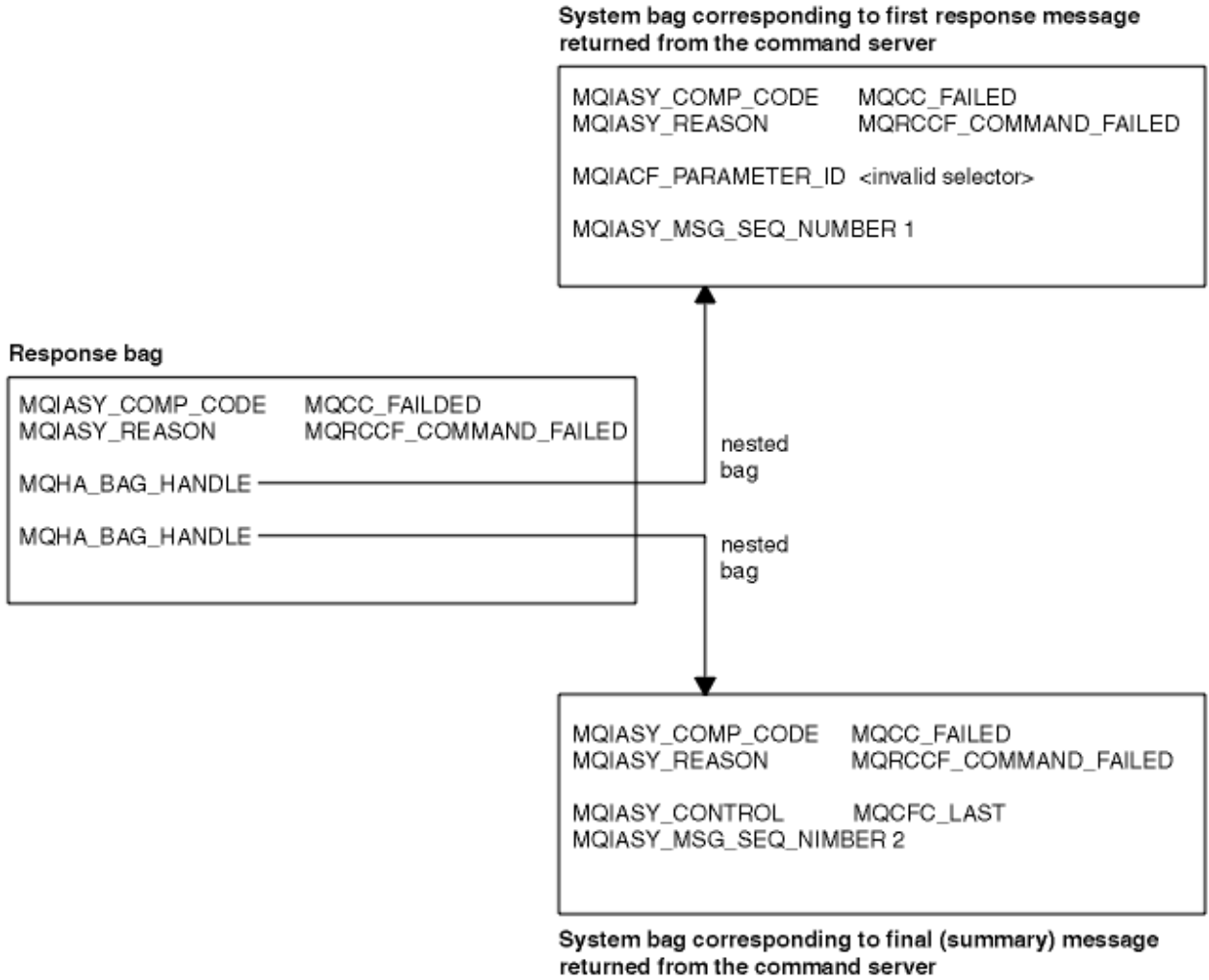
MQAI, komut sunucusuyla doğrudan ilgilenmek yerine, denetim komutlarını komut sunucusuna göndermek için PCF iletilerini kullanır. Burada, MQAI kullanılarak WebSphere MQ ' un yapılandırılmasına ilişkin bazı ipuçları bulunur:

- WebSphere MQ ' daki karakter dizgileri, sabit uzunluğa kadar doldurulmuş boğundur. C kullanıldığında, boş karakterle biten dizgiler olağan durumda WebSphere MQ programlama arabirimlerine giriş değıştirmeleri olarak sağlanabilir.
- Bir dizgi özniteliğinin değeri temizlemek için, özniteliği boş bir dizgi yerine tek bir boşluğa ayarlayın.
- Değıştirmek istediğiniz öznitelikleri önceden göz önünde bulundurun ve yalnızca bu öznitelikleri sorgulayın.

- Bazı öznitelikler değiştirilemez; örneğin, bir kuyruk adı ya da kanal tipi. Yalnızca değiştirilebilecek öznitelikleri değiştirmeyi denediğinizden emin olun. Belirli bir PCF değişiklik nesnesine ilişkin gerekli ve isteğe bağlı parametrelerin listesine bakın. Bkz. [Programlanabilir Komut Biçimlerinin Tanımlamaları](#).
- Bir MQAI çağrısı başarısız olursa, hataya ilişkin bazı ayrıntılar yanıt torbasına döndürülür. Daha sonra, MQHA_BAG_HANDLE seçicisi tarafından erişilebilen içiçe yerleştirilmiş bir çantada daha ayrıntılı bilgi bulunabilir. Örneğin, bir mqExecute çağrısı MQRCCF_COMMAND_FAILED neden koduyla başarısız olursa, yanıt torbasında bu bilgi döndürülür. Bu neden kodunun olası bir nedeni, belirtilen bir seçicinin komut iletisi tipi için geçerli olmadığından ve bu bilgi ayrıntısının, bir torba tanıtıcısı tarafından erişilebilen içiçe yerleştirilmiş bir torbada bulunması olabilir.

MQExecute ile ilgili daha fazla bilgi için bkz. [“Sending administration commands to the command server using the mqExecute call”](#) sayfa 54

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 42’inde 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” sayfa 43 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 44 içinde bulunabilir.

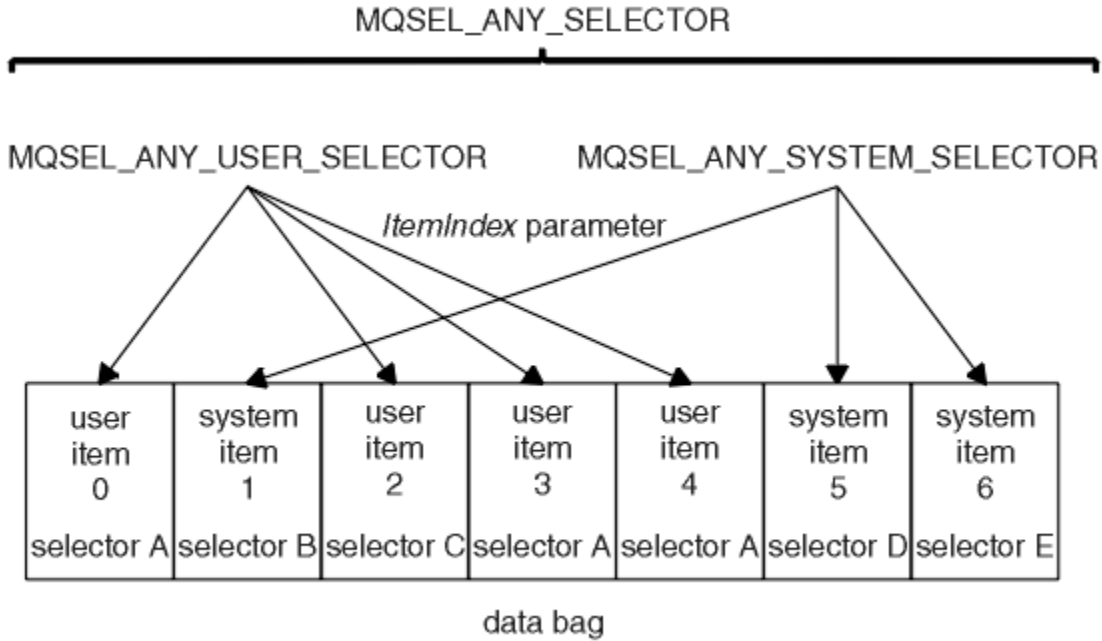
MQAI içinde dizin oluşturma

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

Bu, kullanıcı seçicileri, sistem seçicileri ya da her ikisi tarafından Şekil 1 sayfa 42' ta gösterildiği şekilde dizinlemeye olanak sağlar.



Şekil 1. Dizin oluşturma

Şekil Şekil 1 sayfa 42' 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 silinmesibaşlıklı konuya bakın.

Üç tip dizin oluşturma, veri öğelerinin kolayca alınmasına olanak sağlar. Örneğin, bir çantada belirli bir seçicinin üç eşgörünümlü varsa, mqCountÖğe çağrısı o seçiciye ilişkin eşgörünümlerin sayısını sayabilir ve mqInquire* çağrıları 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

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 iletilinde 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 43 , 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 (1)	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 (1)	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 (2)
mqPutBag	Çanta CCSID 'si (1)	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 (3)
mqGetBag	Çanta CCSID 'si (1)	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:

- Çanta CCSID 'si, selector MQIASY_CODED_CHAR_SET_ID olan sistem ögesidir.
- MQCCSI_Q_MGR, gerçek kuyruk yöneticisi CCSID 'si olarak değiştirildi.
- 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.
- CCSID ' nin CCSID değeri MQCC_DEFAULT ise, kuyruk yöneticisinin CCSID değeri kullanılır.

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 44](#).

Çizelge 4. MQMD 'nin biçimi ve MsgType değıştirgeleri		
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
MQCFT_*	MQFMT_PCF	MQMT_DATAGRAM

Çizelge 4 sayfa 44 , 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 45 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 44
<i>Expiry</i>	30 saniye (not “1” sayfa 45)
<i>Feedback</i>	MQFB_YOK
<i>Encoding</i>	MQENC_NATIVE
<i>CodedCharSetId</i>	torba CCSID ' ye bağlıdır (not “2” sayfa 45)
<i>Format</i>	Bkz. Çizelge 4 sayfa 44
<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 45
<i>ReplyToQMgr</i>	blank (boşluk)
<p>Notlar:</p> <ol style="list-style-type: none"> 1. This value can be overridden on the mqExecute call by using the OptionsBag parameter. Bu konuda bilgi için bkz. mqExecute. 2. Bkz. “MQAI ' de veri dönüştürme” sayfa 43. 3. Name of the user-specified reply queue or MQAI-generated temporary dynamic queue for messages of type MQMT_REQUEST. Aksi takdirde boştur. 	

Veri torbaları

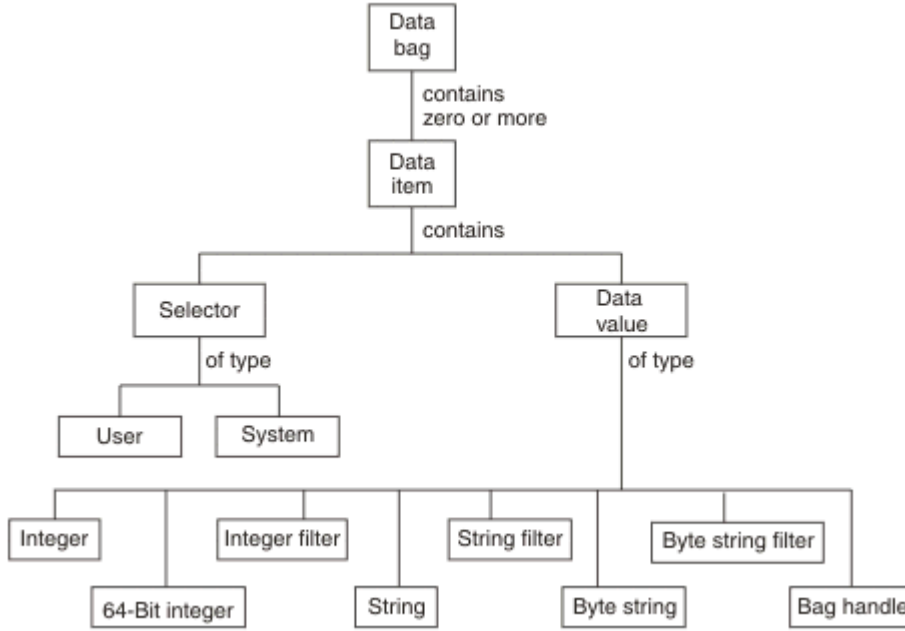
Veri torbası, MQAI kullanan nesnelerin özelliklerini ya da değiştirgelerini işlemek için kullanılan bir araçtır.

Veri Çantaları

- Veri torbası sıfır ya da daha fazla *veri ögesi* içerir. Bu veri ögeleri, 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 ögeleri, “Veri ögesi” sayfa 48 içindeki 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 42](#) içinde açıklanmıştır.

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



Şekil 2. 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 WebSphere MQ nesnelerini yönetmek için kullanılan veriler için oluşturulan bir çanta. Yönetim çantası, [“Veri torbalarının yaratılması ve silinmesi” sayfa 47](#) içinde açıklandığı gibi, belirli seçenekleri otomatik olarak belirtir.

komut çantası

WebSphere 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 47](#).

grup çantası

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

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

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

Veri Torbalarının Kullanılması

Veri torbalarını kullanmanın farklı yolları aşağıdaki listede gösterilmiştir:

- Veri torbaları [“Veri torbalarının yaratılması ve silinmesi”](#) sayfa 47 yaratabilir ve silebilirsiniz.
- Veri torbalarını kullanarak uygulamalar arasında veri gönderebilirsiniz [“Veri torbalarının alınması ve alınması”](#) sayfa 47.
- Veri çantalarına [“Çantalara veri öğeleri ekleme”](#) sayfa 49 veri öğeleri ekleyebilirsiniz.
- Bir veri torbası içinde bir sorgu komutu ekleyebilirsiniz [“Bir çantaya sorgu komutu eklenmesi”](#) sayfa 49.
- You can inquire within data bags [“Veri torbaları araştırılıyor”](#) sayfa 50.
- Veri öğelerini bir veri torbası [“Veri öğelerini sayma”](#) sayfa 52 içinde sayabilirsiniz.
- Bir veri torbası içindeki bilgileri değiştirebilirsiniz [“Bir çantadaki bilgilerin değiştirilmesi”](#) sayfa 50.
- Bir veri torbasını temizleyebilirsiniz [“Clearing a bag using the mqClearBag call”](#) sayfa 51.
- [“Truncating a bag using the mqTruncateBag call”](#) sayfa 52 veri torbasını kesebilirsiniz.
- Torbaları ve arabellekleri ([“Çantalar ve arabellekler dönüştürülüyor”](#) sayfa 52) 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 yinleme 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. [“Veri öğesi”](#) sayfa 48.
- 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 command server using the mqExecute call”](#) sayfa 54. Kullanıcı kodu bir sistem torbasını silemiyor.

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

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, MQAI tarafından uygulama yerine arabelleği işleyebilmesini sağlar. mqPutBag çağrısı, belirtilen çantanın içindekileri bir PCF iletisine dönüştürür ve iletii belirtilen kuyruğa gönderir ve mqGetBag çağrısı, iletii 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 12](#) başlıklı konuya bakın.

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

Veri ögesi

Veri ögeleri oluşturulduğunda veri torbalarını doldurmak için kullanılan veri ögeleri kullanılır. Bu veri ögeleri, kullanıcı ya da sistem ögeleri olabilir.

Bu kullanıcı ögeleri, denetlenmekte olan nesnelere öznitelikleri gibi kullanıcı verilerini içerir. Oluşturulan iletiler üzerinde daha fazla denetim için sistem ögeleri kullanılmalıdır: örneğin, ileti üstbilgilerinin oluşturulması. Sistem ögeleri hakkında daha fazla bilgi için bkz. [“Sistem ögeleri” sayfa 48](#).

Veri Ögesi Tipleri

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

Veri ögesi tamsayı ya da karakter dizilimi ögeleri olabilir. MQAI içinde kullanılacak veri ögesi tipleri şunlardır:

- Tamsayı
- 64 bitlik tamsayı
- Tamsayı süzgeci
- Karakter dizilimi
- Dizgi süzgeci
- Bayt dizilimi
- Bayt dizilimi süzgeci
- Çanta tutamacı

Veri Ögelerini Kullanma

Bu, veri ögelerini kullanmanın aşağıdaki yollarıdır:

- [“Veri ögelerini sayma” sayfa 52](#).
- [“Veri ögelerinin silinmesi” sayfa 53](#).
- [“Çantalara veri ögeleri ekleme” sayfa 49](#).
- [“Veri ögelerini süzme ve sorgulama” sayfa 49](#).

Sistem ögeleri

Sistem ögeleri şunlar için kullanılabilir:

- PCF üstbilgilerinin oluşturulması. Sistem ögeleri, PCF komut tanıtıcısını, denetim seçeneklerini, ileti sıra numarasını ve komut tipini denetleyebilir.
- Veri dönüştürme. Sistem ögeleri, çantadaki karakter dizisiyle ilgili karakter kümesi tanıtıcısını işler.

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

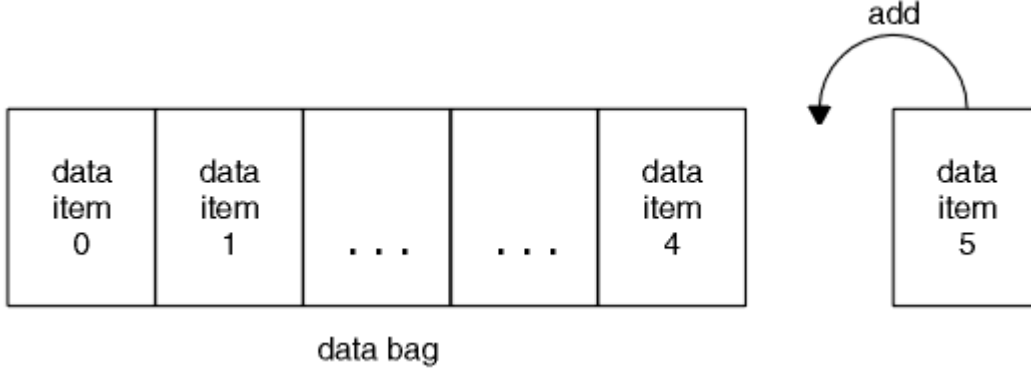
Sistem ögeleri benzersizdir. Bir ya da daha çok sistem ögesi bir sistem seçiciyle tanımlanabilir. Her sistem seçicisinin tek bir örneği vardır.

Çoğu sistem ögesi değiştirilebilir (bkz. [“Bir çantadaki bilgilerin değiştirilmesi” sayfa 50](#)), ancak çanta-yaratma seçenekleri kullanıcı tarafından değiştirilemez. Sistem ögelerini silemezsiniz. (Bkz. [“Veri ögelerinin silinmesi” sayfa 53](#).)

Çantalara veri öğeleri ekleme

Bir veri torbası oluşturulduğunda, veri torbasına veri öğeleri yerleştirebilirsiniz. Bu veri öğeleri, kullanıcı ya da sistem öğeleri olabilir. Veri öğeleri hakkında daha fazla bilgi için bkz. “Veri öğesi” sayfa 48.

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 3 sayfa 49’ünde gösterilir. Öğeler bir seçiciyle tanıtlır. 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 3. Veri öğeleri eklenmesi

mqAdd* çağrılarını kullanarak bir torbaya veri öğeleri ekleme:

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

Bir torbaya veri öğeleri eklemeye ilişkin daha fazla bilgi için bkz. “Sistem öğeleri” sayfa 48.

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 WebSphere MQ.

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

Veri öğelerini süzme ve sorgulama

MQAI olanağını kullanırken, WebSphere MQ nesnelerinin 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, "[Programlanabilir Komut Biçimlerine Giriş](#)" sayfa 9 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 **orgu** 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 öğelerini süzgeçten geçirme ve sorgulamak için daha fazla örnek için bkz. "[MQAI 'nin kullanılmasına ilişkin örnekler](#)" sayfa 20.

Veri torbaları araştırılıyor

Aşağıdaki bilgileri araştırabilirsiniz:

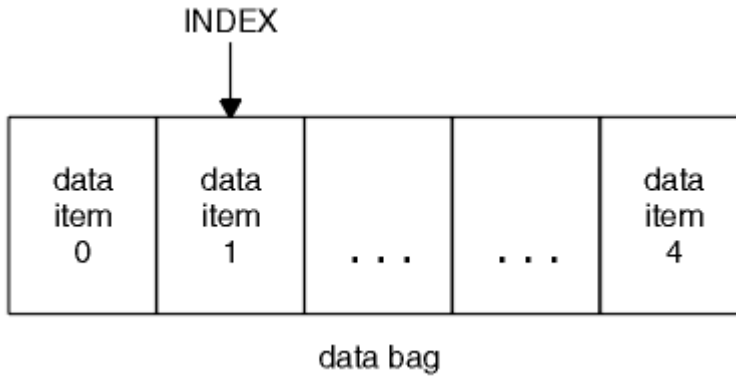
- The value of an integer item using the `mqInquireInteger` call. Bkz. [mqInquireInteger](#).
- `mqInquireInteger64` çağrısını kullanan 64 bitlik bir tamsayı öğesinin değeri. Bkz. [mqInquireInteger64](#).
- `mqInquireIntegerFilter` çağrısını kullanan bir tamsayı süzgeci öğesinin değeri. Bkz. [mqInquireIntegerFilter](#).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

`mqInquireItemInfo` çağrısını kullanarak, belirli bir öğenin 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

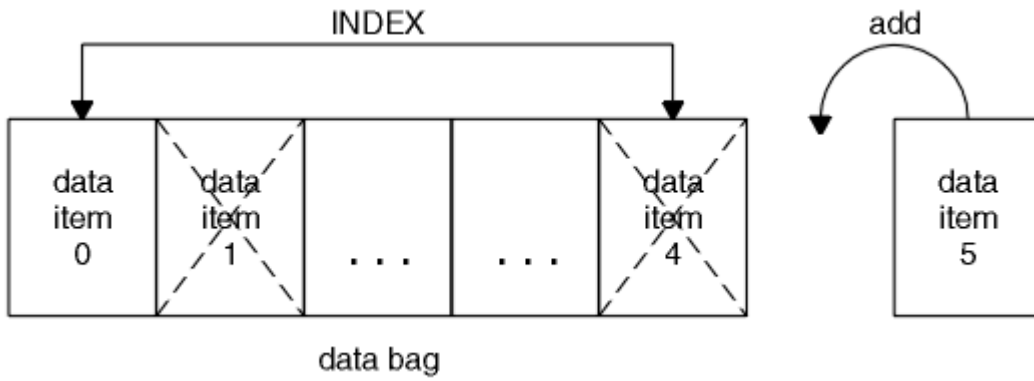
MQAI, `mqSet*` çağrılarını kullanarak bir çanta içindeki bilgileri değiştirmenize olanak sağlar. Yapabilecekleriniz:

1. Bir çantadaki veri öğelerini değiştirin. Dizin, değiştirilecek öğenin oluşumunun belirlenerek bir parametrenin tek tek bir eşgörünümünün değiştirilmesine olanak sağlar (bkz. [Şekil 4 sayfa 51](#)).



Şekil 4. Tek bir veri ögesinin değiştirilmesi

2. Belirtilen seçicinin var olan tüm oluşumlarını silin ve çantanın sonuna yeni bir geçiş ekleyin. (Bkz. Şekil 5 sayfa 51.) Özel bir dizin değeri, bir değiştirgenin **Tümü** yönetim ortamlarının değiştirilmesine olanak sağlar.



Şekil 5. 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](#)

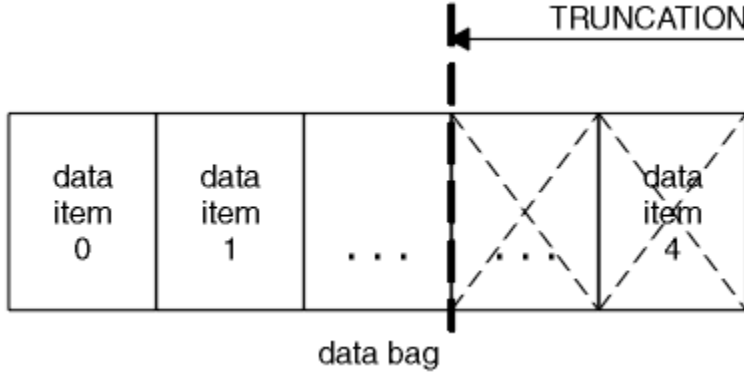
Clearing a bag using the `mqClearBag` call

`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.

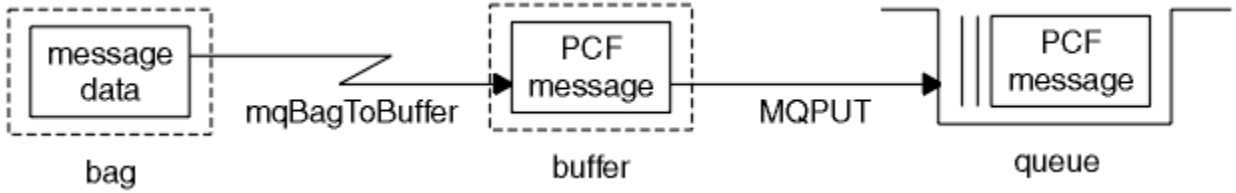


Şekil 6. Çantanın kesilmesi

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

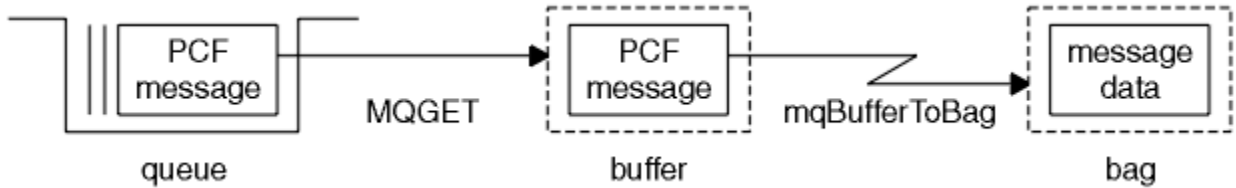
Çantalar ve arabellekler dönüştürülüyor

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 7 sayfa 52 içinde gösterilir. mqBagToBuffer çağrısının tam açıklaması için bkz. [mqBagToBuffer](#).



Şekil 7. 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 8 sayfa 52 içinde gösterilir. mqBufferToBag çağrısının tam açıklaması için bkz. [mqBufferToBag](#).



Şekil 8. 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

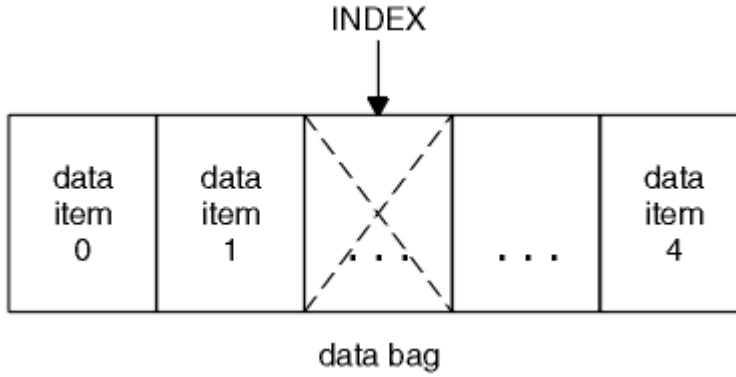
Çantalardaki öğeleri çeşitli şekillerde silebilirsiniz. Yapabilecekleriniz:

- Bir çantadan bir ya da daha fazla kullanıcı öğesini kaldırın. Ayrıntılı bilgi için bkz. [“mqDeleteÖğe çağrısını kullanarak bir çantadan veri öğelerinin silinmesi”](#) sayfa 53.
- **Tümü** kullanıcı öğelerini bir çantadan, yani *temizle* bir torbadan silin. Ayrıntılı bilgi için bkz. [“Clearing a bag using the mqClearBag call”](#) sayfa 51.
- Bir çantanın sonundaki kullanıcı öğelerini silin; bu, bir torbayı *kestir* . Ayrıntılı bilgi için bkz. [“Truncating a bag using the mqTruncateBag call”](#) sayfa 52.

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:

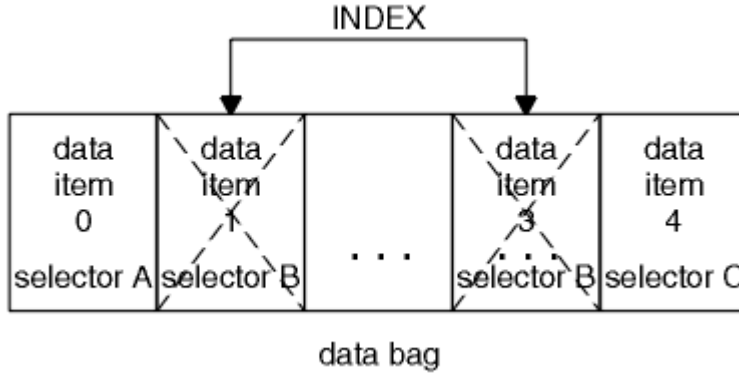
1. Belirtilen seçiciye ilişkin tek bir oluşum. (Bkz. [Şekil 9 sayfa 53.](#))



Şekil 9. Tek bir veri öğesinin silinmesi

ya da

2. Belirtilen seçicinin tüm oluşumları. (Bkz. [Şekil 10 sayfa 53.](#))



Şekil 10. 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](#).

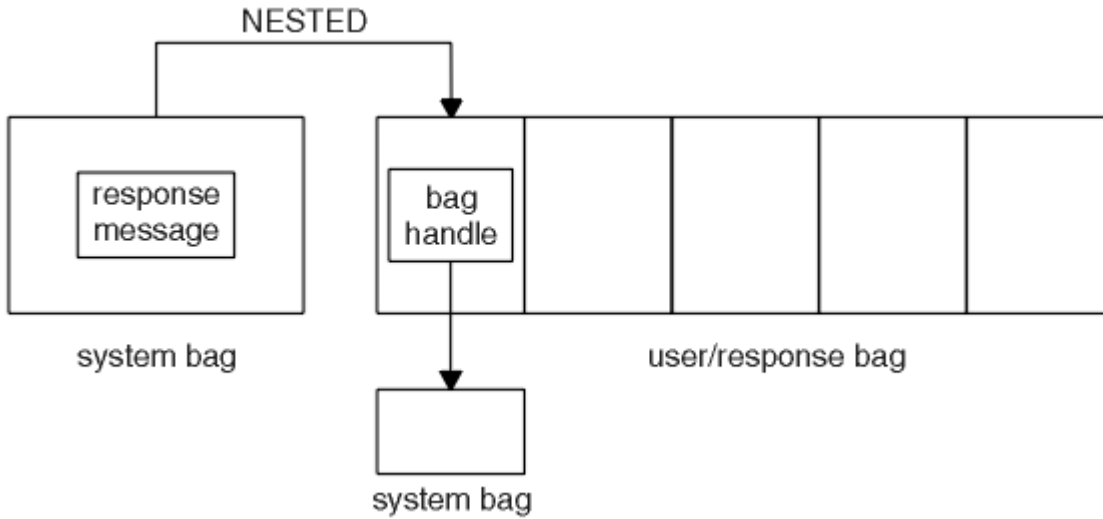
Sending administration commands to the command server using the mqExecute call

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 WebSphere MQ nesnelere ya da bir dizi PCF hata yanıtı iletilisiyle ilgili 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.

iç içe yerleştirme kavramı Şekil 11 sayfa 54 içinde gösterilir.



Şekil 11. 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.

Aşağıdakiler 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.

IBM WebSphere MQ Explorerkomutunu kullanarak yönetim

IBM WebSphere MQ Explorer , ağınızın yerel ya da uzak denetimini, yalnızca Windowsya da Linux (x86 ve x86-64 altyapılarında) çalıştıran bir bilgisayardan gerçekleştirmenizi sağlar.

IBM WebSphere MQ for Windows, and IBM WebSphere MQ for Linux (x86 ve x86-64 platformları), denetim ya da MQSC komutlarını kullanmaya alternatif olarak yönetim görevlerini yerine getirmek için IBM WebSphere MQ Explorer adlı bir yönetim arabirimi sağlar. [Komut takımlarının karşılaştırılması](#) , IBM WebSphere MQ Explorerolanağını kullanarak yapabildiğiniz bilgileri gösterir.

The IBM WebSphere MQ Explorer allows you to perform local or remote administration of your network from a computer running Windows, or Linux (x86-64 platforms), by pointing the IBM WebSphere MQ Explorer at the queue managers and clusters you are interested in. IBM WebSphere MQ Explorer kullanılarak denetlenebilen IBM WebSphere MQ platformları ve düzeyleri, [“Uzak kuyruk yöneticileri” sayfa 56](#) içinde açıklanmıştır.

To configure remote IBM WebSphere MQ queue managers so that IBM WebSphere MQ Explorer can administer them, see [“Önkoşul olan yazılım ve tanımlar” sayfa 57](#).

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

Linuxüzerinde, birden fazla Eclipse kurulumunuz varsa, IBM WebSphere MQ Explorer başlatılamayabilir. Bu gerçekleşirse, diğer Eclipse kurulumu için kullandığınız bir kullanıcı kimliğini kullanarak IBM WebSphere MQ Explorer uygulamasını başlatın.

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

IBM WebSphere MQ Explorerile yapabildiğiniz

Bu, IBM WebSphere MQ Explorerkomutunu kullanarak gerçekleştirebileceğiniz görevlerin listesidir.

IBM WebSphere MQ Gezgini ile şunları yapabilirsiniz:

- Bir kuyruk yöneticisi yaratın ve silin (yalnızca yerel makineniz için).
- Kuyruk yöneticisini başlatma ve durdurma (yalnızca yerel makineniz için).
- WebSphere MQ nesnelere, kuyruklar ve kanallar gibi 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 Yöneticisine Kuyruk Yöneticisi Ekle) sihirbazını kullanarak bir kümeye kuyruk yöneticisi ekleyin.
- SSL (Secure Sockets Layer; Güvenli Yuva Katmanı) kanal güvenliğiyle 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.
- Güvenli yuva katmanı (SSL) sertifikalarını yönetmek, sertifikaları kuyruk yöneticileriyle ilişkilendirmek ve sertifika depolarını yapılandırmak ve ayarlamak (yalnızca yerel makinenizde) için ikeyman GUI 'sini çağırın.
- Create JMS objects from WebSphere MQ objects, and WebSphere MQ objects from JMS objects.
- Şu anda desteklenen tiplerden herhangi biri için bir JMS 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.

Yönetim görevlerini, bir dizi *İçerik Görünümü* ve *Özellik iletişim kutuları* kullanarak gerçekleştirmenizi sağlar.

İçerik Görünümü

İçerik Görünümü, aşağıdakileri görüntüleyebilen bir panodur:

- Öznitelikler ve WebSphere 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.

Navigator görünümü kullanarak WebSphere MQ Gezginini 'nde gezinebilirsiniz. Navigator , gerek duyduğunuz İçerik Görünümlerini seçmenize olanak tanır.

Uzak kuyruk yöneticileri

Bağlantı kurabileceğiniz, desteklenen kuyruk yöneticilerine ilişkin iki kural dışı durum vardır.

Bir Windows ya da Linux (x86 ve x86-64 platformları) sisteminden, WebSphere MQ Explorer, desteklenen tüm kuyruk yöneticilerine aşağıdaki istisnalar dışında bağlanabilir:

- WebSphere MQ for z/OS queue managers earlier than Version 6.0.
- Şu anda desteklenen MQSeries V2 kuyruk yöneticileri destekleniyor.

IBM WebSphere MQ Gezginini, farklı komut düzeyleri ve platformlar arasındaki yeteneklerdeki farkları işler. Ancak, bu öznitelik tanınmadığı bir öznitelikçe rastlarsa, öznitelik görünmez.

If you intend to remotely administer a V6.0 or later queue manager on Pencereler using the IBM WebSphere MQ Explorer on a WebSphere MQ V5.3 computer, you must install Fix Pack 9 (CSD9) or later on your WebSphere MQ for Pencereler V5.3 computer.

If you intend to remotely administer a V5.3 queue manager on iSeries using the WebSphere MQ Explorer on a WebSphere MQ V6.0 or later computer, you must install Fix Pack 11 (CSD11) or later on your WebSphere MQ for iSeries V5.3 computer. Bu düzeltme paketi, WebSphere MQ Explorer ve iSeries kuyruk yöneticisi arasındaki bağlantı sorunlarını giderir.

IBM WebSphere MQ Explorer' in kullanılmasının belirlenmesine karar verilmesi

Kuruluşunuzda IBM WebSphere 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 WebSphere MQ Explorer, when you work with the objects using MQSC commands, you must enclose the object names in single quotation marks, or WebSphere MQ does not recognize them.

Büyük kuyruk yöneticileri

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

Kümeler

WebSphere MQ kümeleri potansiyel olarak yüzlerce ya da binlerce kuyruk yöneticisi içerebilir. WebSphere 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 WebSphere MQ Explorer dramatically because the IBM WebSphere MQ Explorer does not connect to the queue managers in the cluster until you select them.

IBM WebSphere MQ Explorer' in ayarlanması

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

- [“Önkoşul olan yazılım ve tanımlar” sayfa 57](#)
- [“Güvenlik” sayfa 57](#)
- [“Kuyruk yöneticilerinin ve kümelerinin gösterilmesi ve gizlenmesi” sayfa 61](#)
- [“Küme üyeliği” sayfa 62](#)
- [“Veri dönüştürme” sayfa 62](#)

Önkoşul olan yazılım ve tanımlar

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

IBM WebSphere 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 WebSphere 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.

Güvenlik

If you are using WebSphere 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 WebSphere MQ Explorer.

IBM WebSphere MQ Explorer' ı kullanma yetkisi

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

To perform local administrative tasks using the WebSphere 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 WebSphere MQ Explorer, the user executing the WebSphere 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.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 WebSphere MQ for z/OS and perform remote administrative tasks using the IBM WebSphere MQ Explorer, the following must be provided:

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

Buna ek olarak, WebSphere MQ Explorer 'ı yürüten kullanıcının aşağıdaki yetkilerin olması gerekir:

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

WebSphere MQ nesnelere nasıl yetki verileceği hakkında bilgi için bkz. UNIX ya da Linux sistemlerinde ve Windows sistemlerinde bir WebSphere MQ nesnesine erişim verilmesi.

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

WebSphere MQ Explorer 'da varsayılan süzgeç, tüm WebSphere MQ nesnelere görüntülenmektedir. Kullanıcının GÖRÜNTÜLEME yetkisi olmayan herhangi bir WebSphere MQ nesnesi varsa, yetkilendirme hataları oluşur. Yetki olayları kaydediliyorsa, görüntülenen nesne aralığını, kullanıcının GÖRÜNTÜLEME yetkisine sahip olduğu nesnelere sınırlayın.

Uzak kuyruk yöneticilerine bağlanmak için güvenlik

Kanalı, IBM WebSphere MQ Gezgini ile her uzak kuyruk yöneticisi arasında güvenli bir şekilde sağlamalısınız.

IBM WebSphere MQ Gezgini, uzak kuyruk yöneticilerine MQI istemci uygulaması olarak bağlanır. Bu, her uzak kuyruk yöneticisinin bir sunucu bağlantı kanalı tanımlaması ve uygun bir TCP/IP dinleyicisinin olması gerektiği anlamına gelir. Sunucu bağlantı kanalınızı güvenceye almezseniz, kötü amaçlı bir uygulamanın aynı sunucu bağlantısı kanalına bağlanması ve kuyruk yöneticisi nesnelere sınırsız yetkisi elde edilmeleri mümkündür. Sunucu bağlantı kanalınızı güvenceye almak için, kanalın MCAUSER özniteliği için boş olmayan bir değer belirleyin, kanal doğrulama kayıtlarını kullanın ya da bir güvenlik çıkışı kullanın.

MCAUSER özniteliğinin varsayılan değeri yerel kullanıcı kimliğidir. Sunucu bağlantı kanalının MCAUSER özniteliği olarak boş olmayan bir kullanıcı adı belirtirseniz, bu kanalı kullanarak kuyruk yöneticisiyle bağlantı kuran tüm programlar, adı belirtilen kullanıcının kimliğiyle çalışır ve aynı yetki düzeyine sahip olur. Kanal kimlik doğrulaması kayıtları kullanıyorsanız bu durum gerçekleşmez.

Using a security exit with the WebSphere MQ Explorer

WebSphere MQ Explorer olanağını kullanarak, varsayılan bir güvenlik çıkışı ve kuyruk yöneticisi özel güvenlik çıkışları belirtebilirsiniz.

Varsayılan bir güvenlik çıkışı tanımlayabilirsiniz; bu çıkış, WebSphere MQ Explorer 'dan 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, WebSphere MQ Explorer kullanarak belirtmenizi sağlar. Daha fazla bilgi için, WebSphere MQ Yardım Merkezi olanağına bakın.

SSL kullanabilen MQI kanallarını kullanarak uzak kuyruk yöneticisine bağlanmak için IBM WebSphere MQ Explorer komutunu kullanma

IBM WebSphere MQ Explorer , bir MQI kanalını kullanarak uzak kuyruk yöneticilerine bağlanır. SSL güvenliğini kullanarak MQI kanalını güvenceye almak istiyorsanız, bir istemci kanal tanımlama çizelgesini kullanarak kanalı oluşturmanız gerekir.

İstemci kanal tanımlama çizelgesini kullanarak bir MQI kanalının nasıl oluşturulacağı hakkında bilgi için bkz. [IBM WebSphere MQ MQI istemcilerine genel bakış](#).

When you have established the channel using a client channel definition table, you can use the IBM WebSphere MQ Explorer to connect to a remote queue manager using SSL-enabled MQI channel, as described in [“Sistemdeki uzak kuyruk yöneticisini barındıran görevler” sayfa 59](#) and [“IBM WebSphere MQ Explorer' ı barındıran sistemdeki görevler” sayfa 59](#).

Sistemdeki uzak kuyruk yöneticisini barındıran görevler

Uzak kuyruk yöneticisini bulandıran sistemde, aşağıdaki görevleri gerçekleştirin:

1. Bir sunucu bağlantısı ve istemci bağlantısı çifti tanımlayın ve her iki kanalda da sunucu bağlantısında *SSLIPH* değişkeni için uygun değeri belirtin. *SSLIPH* değişkeniyle ilgili daha fazla bilgi için bkz. [SSL ile kanalları koruma](#)
2. Kuyruk yöneticisinin @ipcc dizininde bulunan AMQCLCHL . TAB kanal tanımlama tablosunu IBM WebSphere MQ Explorer' i barındıran sisteme gönderin.
3. Belirlenen kapı üzerinde bir TCP/IP dinleyicisi başlatın.
4. CA ve kişisel SSL sertifikalarını kuyruk yöneticisinin SSL dizinine yerleştirin:
 - UNIX and Linux sistemleri için /var/mqm/qmgrs/+QMNAME+/SSL
 - Windows sistemleri için C:\Program Files\WebSphere MQ\qmgrs\+QMNAME+\SSLBurada +QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.
5. Create a key database file of type CMS named key . kdb . Stash the password in a file either by checking the option in the iKeyman GUI, or by using the -zu1a option with the **runmqckm** commands.
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 the Secure Sockets Layer on Windows systems, see [Working with SSL or TLS on UNIX, Linux and Pencereler systems](#) .

IBM WebSphere MQ Explorer' ı barındıran sistemdeki görevler

IBM WebSphere MQ Explorer' i bulandıran sistemde aşağıdaki görevleri gerçekleştirin:

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

IBM WebSphere MQ Explorer , SSL güvenliği için Java anahtar deposu dosyalarını (JKS) kullanır ve bu nedenle IBM WebSphere MQ Explorer için SSL ' yi yapılandırmak için kullanılan anahtar deposu dosyası bu dosya ile 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 MQ Explorer ' ı başlatın.
5. IBM WebSphere MQ Explorer araç çubuğundan, **Window-> Preferences**(Pencere-Tercihler) öğelerini tıklattıktan sonra **WebSphere 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 "IBM WebSphere MQ Explorer" ' ı barındıran sistemdeki görevler" sayfa 59, 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 bulandıran sistemdeki "Sistemdeki uzak kuyruk yöneticisini barındıran görevler" sayfa 59 içindeki 2. adımda uzak kuyruk yöneticisinden aktardığınız kanal çizelge dosyasının yerini belirtin.
9. **Son** düğmesini tıklatın. Artık uzak kuyruk yöneticisine IBM WebSphere MQ Explorer adresinden erişebilirsiniz.

Başka bir kuyruk yöneticisiyle bağlantı kuruluyor

IBM WebSphere MQ Gezgini, bir kuyruk yöneticisine IBM WebSphere MQ Gezgini ' nin önceden bağlı olduğu bir ara kuyruk yöneticisi (ara kuyruk yöneticisi) aracılığıyla bağlanmanıza olanak sağlar.

Bu durumda, IBM WebSphere MQ Gezgini, 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.

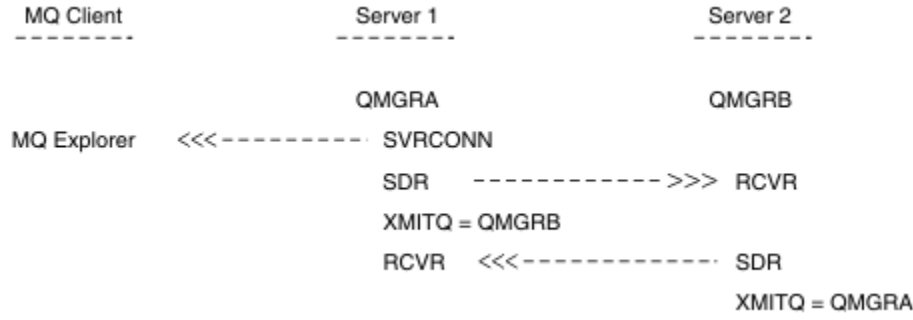
Bağlantı daha sonra bir ara kuyruk yöneticisi aracılığıyla hedef kuyruk yöneticisine bağlanmak için kullanıldıysa, userId ileti tanımlayıcısının (MQMD) *UserIdentifier* parametresine yeniden aktarılır. Hedef kuyruk yöneticisinde MCA dinleyicisinin bu iletiyi kabul etmesi için, MCAUSER özniteliğinin ayarlanması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

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

Aşağıdaki örnekte, bir ara kuyruk yöneticisi aracılığıyla bir kuyruk yöneticisini WebSphere MQ Explorer ' a nasıl bağlayacağını gösteren örnek gösterilmektedir.

Kuyruk yöneticisiyle uzak bir denetim bağlantısı kurun. Aşağıdakilerden emin olun:

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



Bu örnekte:

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

Bu durumda, IBM WebSphere MQ Explorer 'dan QMGRB ile doğrudan bağlantı yoktur; QMGRB ile bağlantı QMGRA ile bağlantılıdır.

Server2 üzerindeki kuyruk yöneticisi QMGRB , gönderen alıcı kanallarını kullanarak Server1 üzerinde QMGRA 'a bağlı. The channel between QMGRA and QMGRB must be set up in such a way that remote administration is possible; see [“Uzaktan yönetim için kanallar ve iletim kuyrukları hazırlama” sayfa 105.](#)

Kuyruk yöneticilerinin ve kümelerinin gösterilmesi ve gizlenmesi

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

Uzak kuyruk yöneticisini göstermek için:

1. Queue Managers ağaç düğümünü sağ tıklayın ve *Kuyruk Yöneticilerini Göster/Gizle ...* seçeneğini belirleyin.
2. **Ekle**'yi tıklayın. Kuyruk Yöneticileri Göster/Gizle panosu görüntülenir.
3. Sağlanan alanlardaki uzak kuyruk yöneticisinin adını ve anasistem adını ya da IP adresini girin.
Anasistem adı ya da IP adresi, uzak kuyruk yöneticisiyle bağlantı kurmak için varsayılan sunucu bağlantı kanalı olan SYSTEM.ADMIN.SVRCONN'a da kullanıcı tanımlı bir sunucu bağlantısı kanalı.
4. **Bitir**'i tıklayın.

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

IBM WebSphere MQ Explorer tarayıcısında bir küme üyesi olan bir kuyruk yöneticisi görüntülenirse, küme algılanır ve otomatik olarak görüntülenir.

Uzak kuyruk yöneticileri listesini bu panodan dışa aktarmak için:

1. Kuyruk Yöneticileri Göster/Gizle panosunu kapatın.
2. WebSphere MQ Explorer 'ın Dolaşma bölmesindeki üst **IBM WebSphere MQ** ağaç düğümünü farenin sağ düğmesiyle tıklayın ve **MQ Gezgini Ayarlarını Dışa Aktar** öğesini seçin.
3. **MQ Explorer > MQ Explorer Ayarları**' yi tıklayın.
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. WebSphere MQ Explorer 'in Dolaşma bölmesinde en üstteki **IBM WebSphere MQ** ağaç düğümünü farenin sağ düğmesiyle tıklatın ve **MQ Gezgini Ayarlarını İçer Aktar** öğesini seçin.
2. **MQ Explorer > MQ Explorer Ayarları**' yı tıklatın.
3. **Göz Atdüğmesini** 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 WebSphere MQ Gezgini 'ne (içer) aktarmak için **Son** düğmesini tıklatın.

Küme üyeliği

IBM WebSphere MQ Gezgini, 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 WebSphere MQ Explorer is running, then you must maintain the IBM WebSphere 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, WebSphere MQ Gezgini 'nin aşağıdaki bilgilere gereksinim duyması gerekir:

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

Bu bilgiler ile, WebSphere 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. WebSphere MQ Explorer otomatik olarak alternatif bir havuza geçmiyor.
- Seçilen havuz TCP/IP üzerinden iletişim kurulamaz.
- Seçilen havuz, WebSphere 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 WebSphere MQ Gezgini, istemci bağlantısı kullanmadan, doğrudan bir kümenin üyesi olan yerel kuyruk yöneticilerine bağlanır.

Veri dönüştürme

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

An error message is issued if you try to establish a connection between the IBM WebSphere MQ Explorer and a queue manager with a CCSID that the IBM WebSphere MQ Explorer does not recognize.

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

Windowsüzerinde güvenlik

Prepare WebSphere MQ sihirbazı, Windows hizmetinin, kullanılması gereken süreçlerle paylaşılabilmesi için özel bir kullanıcı hesabı yaratır.

Bir Windows hizmeti, IBM WebSphere MQ kurulumu için istemci işlemleri arasında paylaşılır. Her kuruluş için bir hizmet yaratılır. Each service is named MQ_InstallationName, and has a display name of IBM WebSphere MQ(InstallationName). Before Version 7.1, with only one installation on a server the single, Windows service was named MQSeriesServices with the display name IBM MQSeries.

Her hizmetin etkileşimli olmayan ve etkileşimli oturum açma oturumları arasında paylaşılması gerektiğinden, her birini özel bir kullanıcı hesabı altında başlatmanız gerekir. Tüm hizmetler için bir özel kullanıcı hesabı kullanılabilir ya da farklı özel kullanıcı hesapları oluşturabilirsiniz. Each special user account must have the user right to "Bir hizmet olarak oturum açma", for more information see "Bir IBM WebSphere MQ Windows Hizmeti için gereken kullanıcı hakları" sayfa 63. Kullanıcı kimliği, hizmeti çalıştırma yetkisine sahip değilse, hizmet başlatılmaz ve Windows sistem olay günlüğünde bir hata döndürür. Typically, you will have run the Prepare IBM WebSphere MQ wizard, and set up the user ID correctly. Ancak, kullanıcı kimliğini el ile yapılandırdıysanız, çözümlenmesi gereken bir sorun olabilir mi?

IBM WebSphere MQ 'u kurduğunuzda ve Prepare IBM WebSphere MQ sihirbazını ilk kez çalıştırdığınızda, MUSR_MQADMIN adlı hizmet için gerekli ayarlar ve izinlerle ("Bir hizmet olarak oturum açma"de içinde olmak üzere) yerel bir kullanıcı hesabı yaratır.

Sonraki kurulumlarda, Prepare IBM WebSphere MQ sihirbazı, MUSR_MQADMINx adlı bir kullanıcı hesabı yaratır; burada x, var olmayan bir kullanıcı kimliğini temsil eden sonraki kullanılabilir sayıdır. MUSR_MQADMINx için parola, hesap yaratıldığında rasgele oluşturulur ve hizmete ilişkin oturum açma ortamını yapılandırmak için kullanılır. Oluşturulan parolanın süresi dolmaz.

Bu IBM WebSphere MQ hesabı, belirli bir dönemden sonra hesap parolalarının değiştirilmesini zorunlu kılacak sistemde ayarlanan hesap ilkelerinden etkilenmez.

Parola, bu tek seferlik işlem dışında bilinmez ve kaydın güvenli bir bölümünde Windows işletim sistemi tarafından depolanır.

Active directory kullanılıyor (yalnızcaWindows)

Kullanıcı hesaplarının Active Directorykullanan etki alanı denetleyicilerinde tanımlı olduğu bazı ağ yapılandırmalarında, IBM WebSphere MQ yerel kullanıcı hesabı, diğer etki alanı kullanıcı hesaplarının grup üyeliğini sorgulamak için gereken yetkiye sahip olmayabilir. Prepare IBM WebSphere MQ Sihirbazı, sınamaları gerçekleştirerek ve kullanıcı sorularına ağ yapılandırmasıyla ilgili sorular sorarak bu durumda olup olmadığını tanımlar.

IBM WebSphere MQ yerel kullanıcı hesabının altında çalıştırılıyorsa, gerekli yetkiye sahip değilse, Prepare IBM WebSphere MQ Sihirbazı, kullanıcının belirli kullanıcı haklarına sahip bir etki alanı kullanıcı hesabına ilişkin hesap ayrıntıları için kullanıcıdan bilgi isteminde bulunur. For the user rights that the domain user account requires see "Bir IBM WebSphere MQ Windows Hizmeti için gereken kullanıcı hakları" sayfa 63. Kullanıcı, etki alanı kullanıcı hesabı için Hazırla IBM WebSphere MQ Sihirbazı 'na ilişkin geçerli hesap ayrıntılarını girdikten sonra, yeni hesap altında çalışmak üzere bir IBM WebSphere MQ Windows hizmetini yapılandırır. Hesap ayrıntıları, Kayıt Defterinin güvenli bölümünde tutulur ve kullanıcılar tarafından okunamaz.

Hizmet çalışırken, bir IBM WebSphere MQ Windows hizmeti başlatılır ve hizmet çalışır durumda olduğu sürece çalışır durumda kalır. Windows hizmeti başlatıldıktan sonra sunucuda oturum açan bir IBM WebSphere MQ yöneticisi, sunucuda kuyruk yöneticilerini yönetmek için IBM WebSphere MQ Explorer 'yi kullanabilir. Bu işlem, IBM WebSphere MQ Explorer 'yi var olan Windows hizmet sürecine bağlar. Bu iki işlem, çalışabilmesi için farklı izin düzeylerine gereksinim duyarlar:

- Başlatma işlemi için bir başlatma izni gerekiyor.
- IBM WebSphere MQ yöneticisi Access izni gerektirir.

Bir IBM WebSphere MQ Windows Hizmeti için gereken kullanıcı hakları

Bu konudaki tabloda, IBM WebSphere MQ kurulumu için Windows hizmetinin altında olduğu yerel ve etki alanı kullanıcı hesabı için gerekli olan kullanıcı hakları listelenir.

Toplu iş olarak oturum aç	Bu kullanıcı hesabı altında çalışmasını sağlamak için bir IBM WebSphere MQ Windows hizmetini etkinleştirir.
---------------------------	---

Hizmet olarak oturum aç	Kullanıcıların, yapılandırılan hesabı kullanarak oturum açmak için IBM WebSphere MQ Windows hizmetini ayarlamasını sağlar.
Sistemi sona erdirin	Bir hizmetin kurtarılması başarısız olduğunda, IBM WebSphere MQ Windows hizmetinin sunucuyu yeniden başlatmasına olanak tanır.
Kotayı büyüt	İşletim sistemi CreateProcessAsUser araması için gereklidir.
İşletim sisteminin bir parçası olarak davran	İşletim sistemi LogonUser araması için gereklidir.
Geçiş denetimini atla	İşletim sistemi LogonUser araması için gereklidir.
Bir işlem düzeyi anahtarını değiştir	İşletim sistemi LogonUser araması için gereklidir.

Not: ASP ve IIS uygulamalarının çalışan ortamlarda hata ayıklama programları hakları gerekli olabilir.

Etki alanı kullanıcı hesabınızın bu Windows kullanıcı hakları, Yerel Güvenlik İlkesi uygulamasında listelendiği şekilde etkin kullanıcı hakları olarak ayarlanmış olmalıdır. Aksi takdirde, yerel güvenlik ilkesi uygulamasını sunucuda yerel olarak ya da Etki Alanı Güvenlik Uygulaması etki alanını geniş kullanarak ayarlayın.

IBM WebSphere MQ Hizmeti ile ilişkili kullanıcı adının değiştirilmesi

You might need to change the user name associated with the IBM WebSphere MQ Service from MUSR_MQADMIN to something else. (Örneğin, kuyruk yöneticiniz 8 karakterden uzun kullanıcı adlarını kabul etmeyen DB2 ile ilişkilendirilmişse, bunu yapmanız gerekebilir.)

Yordam

1. Yeni bir kullanıcı hesabı yarat (örneğin, **NEW_NAME**)
2. Yeni kullanıcı hesabının ayrıntılarını girmek için Prepare IBM WebSphere MQ Wizard olanağını kullanın.

IBM WebSphere MQ Windows hizmeti kullanıcı hesabının parolasının değiştirilmesi

Bu görev hakkında

IBM WebSphere MQ Windows hizmet yerel kullanıcı hesabının parolasını değiştirmek için aşağıdaki adımları gerçekleştirin:

Yordam

1. Hizmetin çalışmakta olduğu kullanıcıyı belirleyin.
2. Stop the IBM WebSphere MQ service from the Computer Management panel.
3. Gereken parolayı, tek bir kişinin parolasını değiştirdiğiniz şekilde değiştirin.
4. Computer Management (Bilgisayar Yönetimi) panosundan IBM WebSphere MQ hizmeti için özelliklere gidin.
5. **Oturum Aç** sayfasını seçin.
6. Belirtilen hesap adının, parolanın değiştirildiği kullanıcıyla eşleştiğini doğrulayın.
7. Type the password into the **Parola** and **Parolayı onayla** fields and click **Tamam**.

Etki alanı kullanıcı hesabı altında çalışan bir kuruluş için IBM WebSphere MQ Windows hizmeti

Bu görev hakkında

Kuruluş için IBM WebSphere MQ Windows hizmeti bir etki alanı kullanıcı hesabı altında çalışıyorsa, hesabın parolasını aşağıdaki gibi de değiştirebilirsiniz:

Yordam

1. Etki alanı denetleyicide etki alanı hesabına ilişkin parolayı değiştirin. Bunu sizin için etki alanı denetimcinizden sormanız gerekebilir.
2. IBM WebSphere MQ hizmetine ilişkin **Oturum Açma** sayfasını değiştirmek için aşağıdaki adımları izleyin.

IBM WebSphere MQ Windows hizmetinin çalıştığı kullanıcı hesabı, kullanıcı arabirimi uygulamaları tarafından yayınlanan ya da sistem başlatma, sona erdirme ya da hizmet kurtarma işlemi sırasında otomatik olarak gerçekleştirilen herhangi bir MQSC komutu yürütür. Bu nedenle, bu kullanıcı hesabının IBM WebSphere MQ yönetim haklarına sahip olması gerekir. Varsayılan olarak sunucu üzerindeki yerel **mqm** grubuna eklenir. Bu üyelik kaldırılırsa, IBM WebSphere MQ Windows hizmeti işe yaramaz. Kullanıcı haklarıyla ilgili daha fazla bilgi için bkz. [“Bir IBM WebSphere MQ Windows Hizmeti için gereken kullanıcı hakları” sayfa 63.](#)

Bir güvenlik sorunu, IBM WebSphere MQ Windows hizmetinin altında çalıştığı kullanıcı hesabıyla ortaya çıksa, hata iletileri ve tanımlar sistem olay günlüğünde görünür.

İlgili kavramlar

[“Active directory kullanılıyor \(yalnızca Windows\)” sayfa 63](#)

Kullanıcı hesaplarının Active Directory kullanan etki alanı denetleyicilerinde tanımlı olduğu bazı ağ yapılandırmalarında, IBM WebSphere MQ yerel kullanıcı hesabı, diğer etki alanı kullanıcı hesaplarının grup üyeliğini sorgulamak için gereken yetkiye sahip olmayabilir. Prepare IBM WebSphere MQ Sihirbazı, sınamaları gerçekleştirerek ve kullanıcı sorularına ağ yapılandırmasıyla ilgili sorular sorarak bu durumda olup olmadığını tanımlar.

Kaynak yöneticisi olarak Db2 ile IBM WebSphere MQ eşgüdümü

Kuyruk yöneticilerinizi IBM WebSphere MQ Explorer olanağından başlatırken ya da IBM WebSphere MQ V7'i kullanıyorsanız ve Db2 koordinasyonunda sorun yaşıyorsanız, kuyruk yöneticisi hata günlüklerini denetleyin.

Aşağıdaki gibi bir hata için kuyruk yöneticisi hata günlüklerinizi denetleyin:

```
23/09/2008 15:43:54 - Process(5508.1) User(MUSR_MQADMIN) Program(amqzma0.exe)
Host(HOST_1) Installation(Installation1)
VMRF(7.1.0.0) QMgr(A.B.C)
AMQ7604: The XA resource manager 'DB2 MQBankDB database' was not available when called
for xa_open. The queue manager is continuing without this resource manager.
```

Açıklama: The user ID (default name is MUSR_MQADMIN) which runs the IBM WebSphere MQ Service process amqsvc.exe is still running with an access token which does not contain group membership information for the group DB2USERS.

Çözüm: IBM WebSphere MQ Hizmeti kullanıcı kimliğinin bir DB2USERS üyesi olmasını sağladıktan sonra, aşağıdaki komut sırasını kullanın:

- Hizmeti durdurun.
- Aynı kullanıcı kimliği altında çalışmakta olan diğer işlemleri durdurun.
- bu işlemleri yeniden başlatın.

Makinenin yeniden yüklenmesi önceki adımlardan emin olur, ancak gerekli değildir.

IBM WebSphere MQ Explorer' in Genişletmesi

IBM WebSphere MQ for Pencereler, and IBM WebSphere MQ for Linux (x86 ve x86-64 platformları), denetim ya da MQSC komutlarını kullanmaya alternatif olarak yönetim görevlerini yerine getirmek için IBM WebSphere MQ Explorer adlı bir yönetim arabirimi sağlar.

This information applies to WebSphere MQ for Windows, and WebSphere MQ for Linux (x86 and x86-64 platforms) only.

IBM WebSphere MQ Explorer, bilgileri Eclipse çerçevesinin ve Eclipse tarafından desteklenen diğer eklenti uygulamalarının bir stiliyle tutarlı bir şekilde sunar.

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

Daha fazla bilgi için, IBM WebSphere MQ Explorer ürün belgelerindeki *IBM WebSphere MQ Explorer 'in Genişletmesi* başlıklı konuya bakın.

IBM WebSphere MQ Taskbar uygulamasını kullanma (yalnızca Windows)

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

Windows üzerinde, WebSphere MQ simgesi, sunucudaki sistem tepsisinde bulunur ve renkle kodlanmış durum simgesiyle, aşağıdaki anlamlardan birine sahip olabilir:

Yeşil

Doğru biçimde çalışıyor; şu anda uyarı yok

Mavi

Belirsiz; WebSphere MQ başlatıyor ya da sona eriyor

Sarı

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

Menüyü görüntülemek için, WebSphere MQ simgesini farenin sağ düğmesiyle tıklatın. Menüden aşağıdaki işlemleri gerçekleştirebilirsiniz:

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

IBM WebSphere MQ uyarı izleyicisi uygulaması (yalnızca Windows)

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

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

WebSphere MQ uyarı izleyicisinden şunları yapabilirsiniz:

- IBM WebSphere MQ Explorer ' a doğrudan erişme
- Tüm bekleyen uyarılarla ilgili bilgileri görüntüle
- Yerel makineden WebSphere MQ hizmetini sona erdirin

- Ağ üzerinden uyarı iletilerini yapılandırılabilir bir kullanıcı hesabına ya da bir Windows iş istasyonuna ya da sunucusuna yönett

Yerel IBM WebSphere MQ nesnelere yönetme

Bu bölümde, Message Queue Interface (MQI) olanağını kullanan uygulama programlarını desteklemek için yerel IBM WebSphere MQ nesnelere nasıl denetlemeniz gerekir. Bu bağlamda yerel yönetim, IBM WebSphere MQ nesnelere oluşturma, görüntüleme, değiştirme, kopyalama ve silme anlamına gelir.

Bu kısımda ayrıntılı olarak açıklanan yaklaşımlara ek olarak, yerel WebSphere MQ nesnelere yönetmek için IBM WebSphere MQ Explorer 'ı kullanabilirsiniz; bkz. [“IBM WebSphere MQ Explorerkomutunu kullanarak yönetim” sayfa 55.](#)

Bu bölüm aşağıdaki bilgileri içerir:

- [MQI kullanan uygulama programları](#)
- [“MQSC komutlarını kullanarak yerel denetim görevlerini gerçekleştirme” sayfa 71](#)
- [“Kuyruk yöneticileriyle çalışılması” sayfa 79](#)
- [“Yerel kuyruklarla çalışma” sayfa 81](#)
- [“Diğer ad kuyruklarıyla çalışılması” sayfa 86](#)
- [“Model kuyruklarıyla çalışma” sayfa 87](#)
- [“Hizmetlerle çalışma” sayfa 94](#)
- [“Tetikleme için nesnelere yönetme” sayfa 100](#)

Kuyruk yöneticisinin başlatılması ve durdurulması

Kuyruk yöneticisini durdurmak ve başlatmak için bu konuyu bir giriş olarak kullanın.

Kuyruk Yöneticisinin Başlatılması

Bir kuyruk yöneticisini başlatmak için `strmqm` komutunu aşağıdaki şekilde kullanın:

```
strmqm saturn.queue.manager
```

WebSphere MQ for Windows ve WebSphere MQ for Linux (x86 ve x86-64 altyapıları) sistemlerinde, aşağıdaki gibi bir kuyruk yöneticisi başlatabilirsiniz:

1. IBM WebSphere MQ Explorer 'ı açın.
2. Select the queue manager from the Navigator View.
3. Startöğesini tıkklatın. Kuyruk yöneticisi başlatılır.

Kuyruk yöneticisi başlatma işlemi birkaç saniyeden daha uzun sürerse WebSphere MQ , başlatma ilerleyişini ayrıntılı olarak açıklayan bilgi iletileri verir.

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

Kuyruk yöneticisinin otomatik olarak başlatılması

WebSphere MQ for Windows içinde, sistem WebSphere MQ Explorer 'ı kullanmaya başladığında otomatik olarak bir kuyruk yöneticisi başlatabilirsiniz. Daha fazla bilgi için [“IBM WebSphere MQ Explorerkomutunu kullanarak yönetim” sayfa 55](#) başlıklı konuya bakın.

Kuyruk Yöneticisinin Durdurulması

Kuyruk yöneticisini durdurmak için `endmqm` komutunu kullanın.

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

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

```
endmqm QMB
```

WebSphere MQ for Windows ve WebSphere MQ for Linux (x86 ve x86-64 platformları için) sistemlerinde, kuyruk yöneticisini aşağıdaki gibi durdurabilirsiniz:

1. IBM WebSphere MQ Explorer 'ı açın.
2. Select the queue manager from the Navigator View.
3. Stop . . . öğesini tıklayın. End Queue Manager panosu görüntülenir.
4. Denetlenen ya da Hemen seçeneğini belirleyin.
5. OKögesini tıklayın. Kuyruk yöneticisi durur.

Susturulmuş sona erdirme

Varsayılan olarak, **endmqm** komutu, belirtilen kuyruk yöneticisinin susturulmuş bir şekilde sona erdirilmesini sağlar. Bu işlem tamamlanacak bir süre alabilir. Tüm bağlı uygulamaların bağlantısı kesilinceye kadar, susturulmuş bir kapatma işlemi bekler.

Uygulamaların durmasını bildirmek için bu sona erdirme tipini kullanın. Bu sorunu ortaya çıkardığınızda:

```
endmqm -c QMB
```

Tüm uygulamalar durduğunda size söylenmez. (Bir `endmqm -c QMB` komutu, bir `endmqm QMB` komutuna eşdeğerdir.)

Ancak, aşağıdaki durumlarda sorun varsa:

```
endmqm -w QMB
```

komut, tüm uygulamalar duruncaya ve kuyruk yöneticisi sona erinceye kadar bekler.

Anında kapatma

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

Hemen sona erdirmek için şunu yazın:

```
endmqm -i QMB
```

Önleyici kapatma

Not: **endmqm** komutunu kullanarak kuyruk yöneticisini durdurma girişimleri başarısız olursa, bu yöntemi kullanmayın. Bu yöntem, bağlı uygulamalar için önceden kestirilemeyecek sonuçlar verebilir.

Hemen sona erdirme çalışmazsa, -p işaretini belirterek bir *önleyici* kapatmaya başvurmanız gerekir.

Örneğin:

```
endmqm -p QMB
```

Bu işlem kuyruk yöneticisini hemen durdurur. Bu yöntem yine de işe yaramazsa, alternatif bir çözüm için bkz. [“Kuyruk yöneticisini el ile durdurma” sayfa 69](#).

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

Kuyruk yöneticisini kapatmaya ilişkin sorunlarınız varsa

Kuyruk yöneticisini kapatma sırasında sorunlar genellikle uygulamalar tarafından neden kaynaklanır. Örneğin, uygulamalar:

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

Kuyruk yöneticisini durdurduğunuzda bir sorun oluşursa, Ctrl-C ' yi kullanarak **endmqm** komutundan ayrılabilirsiniz. Daha sonra başka bir **endmqm** komutu yayınlayabilirsiniz, ancak bu kez gereksinim duyduğunuz sona erdirmeye ilişkin bir işarette birlikte bu işaret kullanılabilir.

Kuyruk yöneticisini el ile durdurma

Kuyruk yöneticilerinin durdurulmasına ilişkin standart yöntemler başarısız olursa, burada açıklanan yöntemleri deneyin.

Kuyruk yöneticilerini durdurmanın standart yolu, endmqm komutunu kullanmaktır. Bir kuyruk yöneticisini el ile durdurmak için, bu bölümde açıklanan yordamlardan birini kullanın. Denetim komutlarını kullanarak kuyruk yöneticilerine ilişkin işlemlerin nasıl gerçekleştirileceği konusunda ayrıntılı bilgi için [Kuyruk yöneticilerinin yaratılması ve yönetilmesine ilişkin konuya](#) bakın.

Windows üzerinde kuyruk yöneticilerini durdurma

How to end the processes and the IBM WebSphere MQ service, to stop queue managers in IBM WebSphere MQ for Pencereler.

Windows için WebSphere MQ altında çalışan bir kuyruk yöneticisini durdurmak için:

1. List the names (IDs) of the processes that are running, by using the Pencereler Task Manager.
2. End the processes by using Pencereler Task Manager, or the **taskkill** command, in the following order (if they are running):

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 yayınlama
AMQFCXBA	Aracı işçi süreci
AMQRPMPPA	Havuz oluşturma işlemi
AMQCRSTA	İş parçacıklı yanıtlayıcı olmayan iş süreci
AMQCRS6B	LU62 alıcı kanalı ve istemci bağlantısı
AMQRRMFA	Havuz işlemi (kümeler için)
AMQZDMAA	Ertelenmiş ileti işlemcisi
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
AMQZTRCN	İz

3. Stop the WebSphere MQ service from **Denetim araçları > Hizmetler** on the Pencereleer Control Panel.
4. Tüm yöntemleri denediyseniz ve kuyruk yöneticisi durdurulmamışsa, sisteminizi yeniden başlatın.

Windows Task Manager ve **tasklist** komutu, görevlerle ilgili sınırlı bilgi verir. Hangi işlemlerin belirli bir kuyruk yöneticisiyle ilgili olduğunu belirlemeye yardımcı olacak ek bilgi için, *Process Explorer* (procepx.exe) gibi bir aracı kullanarak <https://www.microsoft.com> adresindeki Microsoft Web sitesinden yükleyebilirsiniz.

UNIX and Linux sistemlerindeki kuyruk yöneticilerinin durdurulması

How to end the processes and the IBM WebSphere MQ service, to stop queue managers in IBM WebSphere MQ for UNIX and Linux. Burada açıklanan yöntemleri, kuyruk yöneticilerini durdurmak ve kaldırmak için standart yöntemler başarısız olursa deneyebilirsiniz.

UNIX and Linux sistemleri için WebSphere MQ altında çalışan bir kuyruk yöneticisini durdurmak için:

1. Halen ps komutunu kullanarak çalışmakta olan kuyruk yöneticisi programlarının işlem tanıtıcılarını bulun. Örneğin, kuyruk yöneticisi QMNAMEolarak adlandırıldıysa, aşağıdaki komutu kullanın:

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

2. Çalışmakta olan kuyruk yöneticisi işlemlerini sona erdirin. ps komutunu kullanarak keşfedilen süreç tanıtıcılarını belirterek kill komutunu kullanın.

İşlemleri aşağıdaki sırada sona erdirin:

amqzmuc0	Kritik süreç yöneticisi
amqzma0	Yürütme denetleyicisi
amqzfuma	OAM işlemi
amqzlaa0	LQM araçları
amqzlsa0	LQM araçları
amqzmuf0	Yardımcı Program Yöneticisi
amqzmur0	Yeniden başlatılabilir süreç yöneticisi
amqzmgr0	Süreç denetleyicisi
amqfqpub	Abone olma işlemi yayınla
amqfcxba	Aracı işçi süreci
amqrmppa	Havuz oluşturma işlemi
amqcrsta	İş parçacıklı yanıtlayıcı olmayan iş süreci
amqcrs6b	LU62 alıcı kanalı ve istemci bağlantısı
amqrrmfa	Havuz işlemi (kümeler için)
amqzdmaa	Ertelenmiş ileti işlemcisi
amqpcdeniz	Komut sunucusu
runmqtrm	Bir sunucu için tetikleme izleyicinin başlatılması
Runmqdlq	Ölü-mektup kuyruğu işleyicini çağır

runmqchi	Kanal başlatıcı işlemi
Runmqclr	Kanal dinleyicisi işlemi

Not: Durdurmak için başarısız olan işlemleri sona erdirmek için **kill -9** komutunu kullanabilirsiniz.

Kuyruk yöneticisini el ile durdurursanız, FFSTs alınabilir ve /var/mqm/errors. içine yerleştirilen FDC dosyaları bu durumu kuyruk yöneticisinde bir hata olarak dikkate almayabilir.

Bu yöntemi kullanarak durdursanız bile, kuyruk yöneticisi olağan bir şekilde yeniden başlatılır.

MQSC komutlarını kullanarak yerel denetim görevlerini gerçekleştirme

Bu bölümde, MQSC komutları tanıtılır ve bazı genel görevler için bu komutları nasıl kullanacağını bildirir.

Linux (x86 ve x86-64 platformları için) IBM WebSphere MQ for Windows ya da IBM WebSphere MQ kullanıyorsanız, IBM WebSphere MQ Explorer 'ı kullanarak bu bölümde açıklanan işlemleri de gerçekleştirebilirsiniz. Daha fazla bilgi için bkz. [“IBM WebSphere MQ Explorer komutunu kullanarak yönetim” sayfa 55](#).

Kuyruk yöneticisi nesnelere, kuyrukları, süreç tanımlamalarını, kanalları, istemci bağlantı kanallarını, dinleyicileri, hizmetleri, ad listelerini, kümeleri ve kimlik doğrulama bilgileri nesnelere de içeren kuyruk yöneticisi nesnelere yönetmek için MQSC komutlarını kullanabilirsiniz. Bu kısım kuyruk yöneticileriyle, kuyruklarla ve süreç tanımlamalarıyla ilgilidir; kanal, istemci bağlantı kanalı ve dinleyici nesnelere denetlenmesine ilişkin bilgi edinmek için [Nesneler](#) konusuna bakın. Kuyruk yöneticisi nesnelere yönetilmesine ilişkin tüm MQSC komutlarına ilişkin bilgi için bkz. [“Komut Dosyası \(MQSC\) Komutları” sayfa 71](#).

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 75](#).)

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 (RQMNAME gibi) büyük harfli olarak gösterilir. MQSC komut özniteliği adları sekiz karakterle sınırlıdır. MQSC komutları, IBM i ve z/OSde içinde olmak üzere diğer platformlarda kullanılabilir.

MQSC komutları, [MQSC başvurusu](#) bölümündeki konuların derleminde özetlenir.

Komut Dosyası (MQSC) Komutları

MQSC komutları, WebSphere MQ altyapılarında insan tarafından okunabilen komutların yayınına ilişkin tek bir yöntem sağlar. [programlanabilir komut biçimi](#) (PCF) komutlarıyla ilgili bilgi için bkz. [“Programlanabilir Komut Biçimlerine Giriş” sayfa 9](#).

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

MQSC komutlarını kullanırken aşağıdaki kuralları dikkate almalısınız:

- Her komut bir birincil değiştirgeyle (fil) başlar ve bu, onu izleyen ikincil bir parametre (bir isim) izler. Bu, en çok komutta yer alan bir nesnenin (parantez içinde) adını ya da soysal adını (parantez içinde) izler.

Bunun ardından, parametreler genellikle herhangi bir sırada olabilir; bir parametrenin karşılık gelen bir değeri varsa, değer, ilgili olduğu parametrenin hemen ardından gerçekleşmelidir.

- Anahtar sözcükler, ayraçlar ve değerler, herhangi bir sayıda boşluk ve virgül ile ayrılabilir. Sözdizimi şemalarında gösterilen bir virgül her zaman bir ya da daha çok boşlukla değiştirilebilir. There must be at least one blank immediately preceding each parameter (after the primary parameter).
- Komutun başında ya da sonunda ve parametreler, noktalama işaretleri ve değerler arasında herhangi bir sayıda boşluk oluşabilir. Örneğin, aşağıdaki komut geçerlidir:

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

Bir çift tırnak işareti içindeki boşluk karakterlere önem vermektedir.

- Ek virgüller, boşluklara izin verilen ya da boşlukmuş gibi işlem gören herhangi bir yerde görünebilir (elbette, bunlar tırnak işaretleri içindeki dizgilerde değilse).
- Yinelenen parametrelere izin verilmez. REPLACE NOREPLACE gibi "NO" sürümüne sahip bir değiştirenin yinelenmesine de izin verilmez.
- Boşluk, küçük harf karakterleri ya da özel karakterler içeren dizilimler:

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

şu ana kadar tek tırnak işareti içine alınmalıdır:

- Yıldız işaretiyle biten genel değerler
- Tek bir yıldız işareti (örneğin, TRACE (*))
- İki nokta üst üste (örneğin, CLASS (01:03)) içeren bir aralık belirtimi

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.

- z/OS dışındaki platformlarda, hiçbir karakter içermeyen (yani, arasında boşluk içermeyen iki tek tırnak işareti) tek tırnak işareti içine alınmış boş bir alan olarak yorumlanır. Bu, (") ile aynı şekilde yorumlanan bir boşluk olarak yorumlanır. 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.

- v7.0'da, SELECTOR, alt kullanıcı verileri gibi MQCHARV tiplerini temel alan bu dizgi özniteliklerindeki sondaki boşluklar, 'abc ' anlamına gelen önemli bir değer olarak değerlendirilir. 'abc' ile eşit değildir.
- 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.
- Eşanlımlılar bazı parametreler için tanımlanır. Örneğin, DEF her zaman DEFE eşanlımlısıdır, bu nedenle DEF QLOCAL geçerlidir. Eşanlımlılar, yalnızca minimum dizgiler değil; DEFI, DEFE için geçerli bir eşanlımlı değil.

Not: DELETE deęiřtirgesinin eřanlamlısı yok. Bu, DEF eřanlamlısı olan DEF ' yi kullanırken nesnelere yanlıřlıkla silinmesini önlemektedir.

IBM WebSphere MQYönetimi için MQSC komutlarını kullanmaya iliřkin bir genel bakıř için bkz. “MQSC komutlarını kullanarak yerel denetim görevlerini gerekleřtirme” sayfa 71.

MQSC komutları, bazı özel karakterleri belirli anlamlara sahip olacak řekilde kullanır. Bu özel karakterlerle ve bunların nasıl kullanılacaęı hakkında daha fazla bilgi için Özel anlamlar ieren karakterlerbařlıklı konuya bakın.

MQSC komutlarını kullanarak komut dosyalarını nasıl oluřturabileceęiniz hakkında bilgi almak için Building command scriptsbařlıklı konuya bakın.

MQSC komutlarının tam listesi için bkz. MQSC komutları.

İlgili görevler

Komut komut dosyaları oluřturuluyor

WebSphere MQ nesne adları

MQSC komutlarında nesne adlarının kullanılması.

Örneklere, nesnelere için bazı uzun adlar kullanıyoruz. Bu, uęrařtıęınız nesne tipini tanımlamanıza yardımcı olur.

MQSC komutlarını verdięinizde, yalnızca kuyruęun yerel adını belirtmeniz gerekir. Örneğimizde, ařaęıdaki kuyruk adlarını kullanırız:

```
ORANGE . LOCAL . QUEUE
```

Adın LOCAL . QUEUE kısmı, bu kuyruęun yerel bir kuyruk olduęunu göstermekten bařka bir neden deęildir. Bu, genel olarak yerel kuyrukların adları için **deęil** gereklidir.

Ayrıca, kuyruk yöneticisi adı olarak `saturn . queue . manager` adını da kullanırız. Adın `queue . manager` kısmı, bu nesnenin bir kuyruk yöneticisi olduęunu göstermek için gösterilir. Genel olarak kuyruk yöneticilerinin adları için **deęil** gereklidir.

MQSC komutlarında büyük ve küçük harfe duyarlılık

MQSC komutları, öznitelikleri de iinde olmak üzere büyük harfli ya da küçük harfli olarak yazılabilir. Adları tek tırnak iine alınmadıka, MQSC komutlarındaki nesne adları büyük harfe çevrilir (yani, QUEUE ve kuyruk ayırt edilmiyor). Tırnak iřaretleri kullanılmazsa, nesne büyük harfle bir ad ile iřlenir. Ek bilgi için MQSC bařvurusu bařlıklı konuya bakın.

Tüm WebSphere MQ denetim komutlarıyla ortak olan `runmqsc` komut aęırısı, bazı WebSphere MQ ortamlarında büyük/küçük harfe duyarlıdır. Ek bilgi için Denetim komutlarının kullanılması bařlıklı konuya bakın.

Standart giriř ve ıkıř

`stdin` olarak da adlandırılan *standart giriř aygıtı*, sisteme giriřin alındıęı aygıttır. Genellikle bu klavye olur, ancak giriřin bir dizeysel kapıdan ya da bir disk dosyasından geldięini belirtebilirsiniz. `stdout` olarak da bilinen *standart ıkıř aygıtı*, sistemin ıktısının gönderildięi aygıttır. Genellikle bu bir görüntüdür, ancak ıkıřı bir dizeysel kapıya ya da bir dosyaya yeniden yönlendirebilirsiniz.

İřletim sistemi komutlarında ve WebSphere MQ denetim komutlarında, `<` iřle giriři yeniden yönlendirir. Bu iřle bir dosya adıyla takip edildiyse, giriřten giriř alınır. Benzer řekilde, `>` iřlei ıkıřı yeniden yönlendirir; bu iřle bir dosya adıyla takip edildiyse, ıkıř o dosyaya yönlendirilir.

MQSC komutlarının etkileřimli olarak kullanılması

MQSC komutlarını bir komut penceresi ya da kabuk kullanarak etkileřimli 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 belirttik 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: WebSphere 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 girmediyse, doğru sözdizimi için [MQSC başvurusu](#) başlıklı konuya 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.

MQSC komutlarını metin dosyalarından çalıştırma

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.

“Standart giriş ve çıkış” sayfa 73 , stdin ve stdout ile ilgili bilgileri içerir. stdin dosyasını bir metin dosyasından yeniden yönlendirmek için, önce olağan metin düzenleyicinizi kullanarak MQSC komutlarını içeren bir metin dosyası yaratın. runmqsc komutunu kullandığınızda yeniden yönlendirme işlemlerini kullanın. Örneğin, aşağıdaki komut myprog.in metin dosyasında bulunan bir komut dizisini çalıştırır:

```
runmqsc < myprog.in
```

Benzer şekilde, çıktıyı bir dosyaya da yönlendirebilirsiniz. Giriş için MQSC komutlarını içeren bir dosya, *MQSC komut dosyası* adı verilir. Kuyruk yöneticisinden yanıt içeren çıkış dosyasına *çıkış dosyası* adı verilir.

stdin ve stdout komutlarını runmqsc komutuna yeniden yönlendirmek için komutun bu biçimini kullanın:

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

Bu komut, myprog.in . MQSC komut dosyasında bulunan MQSC komutlarını çağırır; bir kuyruk yöneticisi adı belirtmediğimizden, MQSC komutları varsayılan kuyruk yöneticisine karşı çalıştırılıyor. Çıkış, myprog.out metin kütüğüne gönderilir. Şekil 12 sayfa 76 shows an extract from the MQSC command file myprog.in and Şekil 13 sayfa 77 shows the corresponding extract of the output in myprog.out.

Varsayılan olmayan bir kuyruk yöneticisi (saturn.queue.manager) için, runmqsc komutuna stdin ve stdout 'ı yeniden yönlendirmek için bu komutun bu biçimini kullanın:

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

MQSC komut dosyaları

MQSC komutları, ASCII metin biçiminde, insan tarafından okunabilen bir biçimde yazılır. Şekil 12 sayfa 76 Bir MQSC komut kütüğünden, öznitelikleriyle bir MQSC komutu (DEFINE QLOCAL) gösteren bir özettir. [MQSC başvurusu](#) , MQSC komutlarının ve sözdiziminin bir açıklamasını içerir.

```

.
.
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 12. MQSC komut kütüğünden veri al

WebSphere 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 *rapord* 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 13 sayfa 77](#) içinde 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, `DEFINE QLOCAL` komutunun başarıyla tamamlanmasına ilişkin işletmen iletisi şöyledir:

```
AMQ8006: WebSphere 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: WebSphere MQ queue created.
:
.

```

Şekil 13. Bir MQSC komut raporu dosyasından çıkarma

Sağlanan MQSC komut dosyalarının çalıştırılması

Aşağıdaki MQSC komut dosyaları WebSphere MQ ile birlikte sağlanır:

amqscos0.tst

Örnek programlar tarafından kullanılan nesnelere ilişkin tanımlar.

amqscic0.tst

CICS hareketlerine ilişkin kuyrukların tanımları.

WebSphere MQ for Windows içinde bu dosyalar,

`MQ_INSTALLATION_PATH\tools\mqsc\samplesdizininde` bulunur. `MQ_INSTALLATION_PATH` WebSphere MQ 'un kurulu olduğu üst düzey dizini temsil eder.

UNIX and Linux sistemlerinde bu dosyalar, `MQ_INSTALLATION_PATH/sampdizininde` bulunur.

`MQ_INSTALLATION_PATH` WebSphere MQ 'un 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` komutundaki `-v` işaretini ayarlayın; örneğin:

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

`runmqsc` 'ı 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 13 sayfa 77 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 WebSphere MQ waits for replies from the remote queue manager.

Toplu iş dosyalarından MQSC komutları çalıştırılıyor

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

Bir toplu iş kütüğünden stdin 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. amqsputc örnek programını kullanarak, kuyruğa ileti koyar.

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

crtmqm $MYTEMPQM
strmqm $MYTEMPQM
runmqclsr -m $MYTEMPQM -t TCP -p $MYPORT &

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

amqsputc TESTQ $MYTEMPQM << EOF
hello world
EOF

endmqm -i $MYTEMPQM
```

Şekil 14. 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 uygulansa da bu konudaki bilgileri kullanın. Bir komutun ürettiği hatayı okuduğunuzda sorunun ne olduğu her zaman açık değildir.

MQSC komutlarını çalıştırmak için alamıyorsanız, bu ortak sorunlardan birinin sizin için geçerli olup olmadığını görmek için aşağıdaki bilgileri kullanın. Oluşturulan hatayı okuduğunuzda sorunun ne olduğu her zaman açık değildir.

runmqsc komutunu kullandığınızda, aşağıdakileri 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 yayınlr:

```
AMQ8118: WebSphere MQ queue manager does not exist.
```

- Çıkışı bir dosyaya yeniden yönlendiriyorsanız, > yeniden yönlendirme işlecini kullanın. Varsayılan olarak, dosya, `runmqsc` ' un çağrıldığı zaman yürürlükteki 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:

```
AMQ8146: WebSphere 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 `strmqm` komutunu yayınlamazsınız. Bunu yapmazsanız, aşağıdakine benzer hata iletileri alırsınız:

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

  1 : strmqm 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
```

Kuyruk yöneticileriyle çalışılması

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

Kuyruk yöneticisi özneliklerinin görüntülenmesi

runmqsc komutunda belirtilen kuyruk yöneticisinin özneliklerini görüntülemek için, aşağıdaki MQSC komutunu kullanın:

```
DISPLAY QMGR
```

Bu komuttaki tipik çıkış Şekil 15 sayfa 80’inde gösterilir.

```
DISPLAY QMGR
 1 : DISPLAY QMGR
AMQ8408: Display Queue Manager details.
QMNAME(QM1)
ACCTINT(1800)
ACCTQ(OFF)
ACTVCONO (DISABLED)
ALTDATA(2012-05-27)
AUTHOREV(DISABLED)
CHAD(DISABLED)
CHADEXIT( )
CLWLDATA( )
CLWLLEN(100)
CLWLUSEQ(LOCAL)
CMDLEVEL(750)
CONFIGEV(DISABLED)
CRTIME(16.14.01)
DEFXMITQ( )
DISTL(YES)
IPADDRV(IPV4)
LOGGEREV(DISABLED)
MAXHANDS(256)
MAXPROPL(NOLIMIT)
MAXUMSGS(10000)
MONCHL(OFF)
PARENT( )
PLATFORM(WINDOWSNT)
PSNPMMSG(DISCARD)
PSSYNCP(IFPER)
PSMODE(ENABLED)
REPOS( )
ROUTEREC(MSG)
SCMDSERV(QMGR)
SSLCRYP( )
SSLFIPS(NO)
MQData\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)
PSNMPRES(NORMAL)
QMID(QM1_2011-05-27_16.14.01)
REMOVEDEV(DISABLED)
REPOSNL( )
SCHINIT(QMGR)
SSLCRLNL( )
SSLEV(DISABLED)
SSLKEYR(C:\Program Files\IBM\WebSphere
STATACLS(QMGR)
STATINT(1800)
STATQ(OFF)
SYNCP
TRIGINT(999999999)
```

Şekil 15. Bir DISPLAY QMGR komutundan tipik çıkış

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 varolduğunu 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 başlıklar

Kuyruk yöneticisine ilişkin öznitelikler

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 başvurusu](#) ' e bakın.

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 MQSC komutunu TANIMLA QLOCAL 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 kuruluşu 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. Tanımın değeri dışında, gösterilen tüm öznitelik değerleri varsayılan değerlerdir. Onları burada göstermemiz için gösterdik. Varsayılan değerlerin istediğiniz ya da değiştirilmemiş olduğundan eminseniz, bu değerleri atlayabilirsiniz. Ayrıca bkz. [“Varsayılan nesne özniteliklerinin görüntülenmesi”](#) sayfa 82.
2. KULLANIM (NORMAL), bu kuyruğun iletim kuyruğu olmadığını gösterir.
3. ORANGE.LOCAL.QUEUE, bu komut başarısız olur. Bir kuyruğun var olan tanımlanmasının üzerine yazmak istiyorsanız REPLACE özniteliğini kullanın; ancak bkz. [“Yerel kuyruk özniteliklerinin değiştirilmesi”](#) sayfa 83.

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 kuyruğundan bahsetmelisiniz.

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. Kullanılmadan önce, ölü-mektup 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)

WebSphere MQ , bir ölü-mektup kuyruğunda bulunan iletilerin nasıl işleneceğini ya da kaldırıldığını belirtmenize olanak tanıyan bir ileti kuyruğu işleyicisi sağlar. Daha fazla bilgi için bkz. [Teslim edilmeyen iletilerin WebSphere MQ ölme-harflı kuyruk işleyicisiyle işlenmesi](#) .

Varsayılan nesne özniteliklerinin görüntülenmesi

Bir WebSphere MQ nesnesi tanımlandığında varsayılan nesneden alınmış öznitelikleri görüntülemek için DISPLAY QUEUE komutunu kullanabilirsiniz.

Bir WebSphere 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, karşılık gelen DEFINE komutunun sözdiziminden farklıdır. DISPLAY komutunda yalnızca kuyruk adını verebilirsiniz; DEFINE komutunda ise, kuyruğun tipini belirtmeniz gerekir; örneğin, QLOCAL, QALIAS, Qmodel ya da QREMOTE.

Öznitelikleri tek tek belirterek seçmeli olarak görüntüleyebilirsiniz. Örneğin:

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +
  MAXDEPTH +
  MAXMSGL +
  CURDEPTH;
```

Bu komut, belirtilen üç özniteliği aşağıdaki gibi görüntüler:

```
AMQ8409: Display Queue details.
  QUEUE (ORANGE.LOCAL.QUEUE)          TYPE (QLOCAL)
  CURDEPTH (0)                          MAXDEPTH (5000)
  MAXMSGL (4194304)
```

CURDEPTH, yürürlükteki kuyruk derinliğini, yani kuyruktaki ileti sayısını içerir. Bu, görüntülemek için yararlı bir özniteliğe sahip, çünkü kuyruk derinliğini izleyerek, kuyruğun dolu olmamasını sağlayabilirsiniz.

Yerel kuyruk tanımlamasının kopyalanması

DEFE komutundaki LIKE özniteliğini kullanarak bir kuyruk tanımlamasını kopyalayabilirsiniz.

Örneğin:

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

Bu komut, özgün kuyruğumuzun (ORANGE.LOCAL.QUEUEdeğeri, sistem varsayılan yerel kuyruğundan değil. Kuyruğu oluşturduğunuz sırada girildiği şekilde, **tam olarak** kopyalanacak olan kuyruğun adını girin. Ad küçük harflı karakterler içeriyorsa, adı tek tırnak işareti içine alın.

Ayrıca, bir kuyruk tanımlamasını kopyalamak için DEFINE komutunun bu formunu kullanabilir, ancak özgün özniteliklere bir ya da daha fazla değişiklik yerine 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. DEFE komutundaki LIKE özniteliğini kullandığınızda, yalnızca kuyruk özniteliklerini kopyalısınız. Kuyruklardaki iletileri kopyalamazsınız.
2. LIKE belirtmeden yerel bir kuyruk tanımladığınızda, DEFINE LIKE (SYSTEM.DEFAULT.LOCAL.QUEUE).

Yerel kuyruk özniteliklerinin değiştirilmesi

ALTER QLOCAL komutunu ya da REPLACE özniteliği ile DEFINE QLOCAL komutunu kullanarak, kuyruk özniteliklerini iki şekilde değiştirebilirsiniz.

“Yerel kuyruk tanımlanması” sayfa 81 içinde, kuyruk ORANGE.LOCAL.QUEUE tanımlandı. Örneğin, bu kuyruktaki ileti uzunluğu üst sınırını 10.000 bayt olarak azaltmak istediğinizi varsayalım.

- ALTER komutunun kullanılması:

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

Bu komut, ileti uzunluğu üst sınırı olan tek bir özniteliği değiştirir; diğer tüm öznitelikler de aynı kalır.

- REPLACE seçeneğiyle DEFE komutunun kullanılması, 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 **azalt**, var olan iletiler bundan etkilenmez. Ancak yeni mesajların yeni kriterlere uygun olması gerekir.

Yerel kuyruğun temizlenmesi

Yerel bir kuyruğu temizlemek için CLEAR komutunu kullanabilirsiniz.

Tüm iletileri MAGENTA.QUEUE, aşağıdaki komutu kullanın:

```
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.

Yerel Kuyruğun Silinmesi

Bir yerel kuyruğu silmek için MQSC komutunu DELETE QLOCAL komutunu kullanabilirsiniz.

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

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

PURGE yerine NOPURGE belirtilmesi, kuyruğun kesinleştirilmiş ileti içeriyorsa silinmemesini sağlar.

Tarama kuyrukları

WebSphere 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 , WebSphere MQ ' un kurulu olduğu üst düzey dizini temsil eder.

In WebSphere MQ for Pencereler, the file names and paths for the sample queue browser are as follows:

Kaynak

MQ_INSTALLATION_PATH\tools\c\samples

Yürütülebilir

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

In WebSphere MQ for UNIX and Linux, the file names and paths are as follows:

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 16 sayfa 85](#) içinde gösterilir.

Kuyruklar için gereksinim duyduğunuz depolama miktarını planlamaya ilişkin bilgi için, platforma özgü performans raporlarına ilişkin IBM WebSphere MQ web sitesini ziyaret edin:

<https://www.ibm.com/software/integration/ts/mqseries/>

Diğer ad kuyruklarıyla çalışılması

Başka bir kuyruğa ya da konuya dolaylı olarak gönderme yapmak için bir diğer ad kuyruğu tanımlayabilirsiniz.

V 7.5.0.8



Uyarı: Dağıtım listeleri, konu nesnelərini işaret eden diğer ad kuyruklarının kullanılmasını desteklemez. Version 7.5.0, Fix Pack 8' den bir diğer ad kuyruğu dağıtım listesindeki bir konu nesnesini gösterirse, IBM WebSphere MQ MQRC_ALIAS_BASE_Q_TYPE_ERROR döndürür.

Diğer ad kuyruğunun gönderme yaptığı kuyruk aşağıdakilerden biri olabilir:

- Yerel bir kuyruk (bkz. [“Yerel kuyruk tanımlanması”](#) sayfa 81).
- Uzak kuyruğun yerel tanımlaması (bkz. [“Uzak kuyruğun yerel tanımlaması yaratılması”](#) sayfa 110).
- Bir konu.

Diğer ad kuyruğu gerçek bir kuyruk değil, yürütme sırasında gerçek (ya da hedef) bir kuyruğa çözülen bir tanımlama. Diğer ad kuyruğu tanımlaması hedef kuyruğu belirtir. Bir uygulama bir diğer ad kuyruğuna MQOPEN çağrısı yaptığında, kuyruk yöneticisi diğer adı hedef kuyruk adına çözer.

Bir diğer ad kuyruğu, yerel olarak tanımlanmış başka bir diğer ad kuyruğuna çözülemiyor. Ancak, bir diğer ad kuyruğu, yerel kuyruk yöneticisinin üyesi olduğu kümelerde başka bir yerde tanımlanan diğer ad kuyruklarına çözülebilir. Ek bilgi için [Ad çözümlülüğü](#) konusuna bakın.

Diğer ad kuyrukları aşağıdakiler için kullanışlıdır:

- Farklı uygulamalara hedef kuyruk için farklı erişim yetkisi düzeyleri verme.
- Farklı uygulamaların aynı kuyrukla farklı şekillerde çalışmasına izin verilmesi. (Farklı varsayılan öncelikler ya da farklı varsayılan kalıcılık değerleri atamak isteyebilirsiniz.)
- Bakım, geçiş ve iş yükü dengelemeyi basitleştirme. (Diğer adı kullanmaya devam eden uygulamanızı değiştirmek zorunda kalmadan hedef kuyruk adını değiştirmek istiyor olabilir.)

Örneğin, iletilerin MY.ALIAS.QUEUE. Bu kuyruk, bir MQOPEN isteği yaptığında ve dolaylı olarak bu kuyruğa bir ileti koyarsa, bu kuyruğun adını belirtir. Uygulama, kuyruğun bir diğer ad kuyruğu olduğunu farkında değil. Bu diğer adı kullanan her MQI çağrısı için, kuyruk yöneticisi gerçek kuyruk adını çözer; bu, yerel bir kuyruk ya da bu kuyruk yöneticisinde tanımlanmış bir uzak kuyruk olabilir.

TARGET özniteliğinin değerini değiştirerek, MQI çağrılarını büyük olasılıkla başka bir kuyruk yöneticisinde başka bir kuyruğa yeniden yönlendirebilirsiniz. Bu, bakım, geçiş ve yük dengeleme için yararlıdır.

Diğer ad kuyruğu tanımlanması

Aşağıdaki komut bir diğer ad kuyruğu yaratır:

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

Bu komut, MY.ALIAS.QUEUE , YELLOW.QUEUE. Komut hedef kuyruğu yaratmaz; kuyruk YELLOW.QUEUE yürütme sırasında yok.

Diğer ad tanımlamasını değiştirirseniz, MQI çağrılarını başka bir kuyruğa yönlendirebilirsiniz. Örneğin:

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

Bu komut, MQI çağrılarını başka bir kuyruğa (MAGENTA.QUEUE.

Tek bir kuyruğun (hedef kuyruk) farklı uygulamalara ilişkin farklı özniteliklere sahip görünmesini sağlamak için diğer ad kuyruklarını da kullanabilirsiniz. Bunu, her uygulama için bir tane olmak üzere iki diğer ad tanımlayarak yaparsınız. İki uygulama olduğunu varsayalım:

- Uygulama ALPHA, YELLOW.QUEUE, ancak bu kuyruktan ileti almasına izin verilmiyor.
- Uygulama BETA, YELLOW.QUEUE, ancak üzerine ileti koymasına izin verilmiyor.

Aşağıdaki komut, ALPHA uygulaması için etkinleştirilen ve devre dışı bırakılan bir diğer adı tanımlar:

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

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

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

ALPHA, ALPHAS.ALIAS.QUEUE ; BETA, BETAS.ALIAS.QUEUE. İkisi de aynı kuyruğa farklı yollarla erişiyor.

Kuyruk diğer adlarını tanımlarken LIKE ve REPLACE özniteliklerini, bu öznitelikleri yerel kuyruklarda kullandığınız gibi kullanabilirsiniz.

Diğer komutların diğer ad kuyruklarıyla kullanılması

Diğer ad kuyruğu özniteliklerini görüntülemek ya da değiştirmek ya da diğer ad kuyruğu nesnesini silmek için uygun MQSC komutlarını kullanabilirsiniz. Örneğin:

Diğer ad kuyruğunun özniteliklerini görüntülemek için aşağıdaki komutu kullanın:

```
DISPLAY QALIAS (ALPHAS.ALIAS.QUEUE)
```

Diğer adın çözüldüğü temel kuyruk adını değiştirmek için aşağıdaki komutu kullanın; burada `force` seçeneği, kuyruk açık olsa bile değişikliği zorlar:

```
ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGET(ORANGE.LOCAL.QUEUE) FORCE
```

Bu kuyruk diğer adını silmek için aşağıdaki komutu kullanın:

```
DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

Bir uygulamada kuyruk açıksa, diğer ad kuyruğunu silemezsiniz. Bu ve diğer ad kuyruğu komutlarıyla ilgili ek bilgi için [MQSC başvurusu](#) başlıklı konuya bakın.

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

Bir öznitelikler kümesiyle birlikte, bir yerel kuyruğu tanımladığınız şekilde bir model kuyruğu tanımladınız. Model kuyrukları ve yerel kuyruklar, yaratılan dinamik kuyrukların geçici mi kalıcı mı, yoksa kalıcı mı olduğunu belirleyebileceğiniz model kuyruklarında aynı öznitelikler kümesine sahiptir. (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 özneliğinden, bu şablondan yaratılan gerçek kuyrukların kalıcı dinamik kuyruklar olduğunu görebilirsiniz. Belirtilmeyen öznitelikler, SYSYTEM.DEFAULT.MODEL.QUEUE varsayılan kuyruğu.

Model kuyrukları tanımladığınızda, bu öznitelikleri 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 aşağıdaki komutu kullanın:

```
DISPLAY QUEUE (GREEN.MODEL.QUEUE)
```

Bu modelden yaratılan herhangi bir dinamik kuyruğa yerleştirmeyi etkinleştirmek üzere modeli değiştirmek için aşağıdaki komutu kullanın:

```
ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)
```

Bu model kuyruğunu silmek için aşağıdaki komutu kullanın:

```
DELETE QMODEL (RED.MODEL.QUEUE)
```

Yönetimle ilgili konularla çalışma

Yönetimle ilgili konuları yönetmek için MQSC komutlarını kullanın.

Bu komutlarla ilgili ayrıntılı bilgi için bkz. [MQSC başvurusu](#) .

İlgili kavramlar

[Yönetimle ilgili konu nesneleri](#)

[“Yönetimle ilgili konu tanımlanması” sayfa 89](#)

Bir yönetim konusu yaratmak için **DEFINE TOPIC** MQSC komutunu kullanın. Yönetimle ilgili bir konu tanımlarken, isteğe bağlı olarak her konu özneliğini ayarlayabilirsiniz.

[“Yönetimle ilgili konu nesnesi özniteliklerinin görüntülenmesi” sayfa 89](#)

Bir denetim konusu nesnesini görüntülemek için **DISPLAY TOPIC** MQSC komutunu kullanın.

[“Denetim konusu özniteliklerinin değiştirilmesi” sayfa 90](#)

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 90

DEFINE komutundaki **LIKE** özneliğini kullanarak bir konu tanımını kopyalayabilirsiniz.

“Denetim konusu tanımlamasının silinmesi” sayfa 91

Bir yönetim konusunu silmek için **DELETE TOPIC MQSC** komutunu kullanabilirsiniz.

Yönetimle ilgili konu tanımlanması

Bir yönetim konusu yaratmak için **DEFINE TOPIC MQSC** komutunu kullanın. Yönetimle ilgili bir konu tanımlarken, isteğe bağlı olarak her konu özneliğini ayarlayabilirsiniz.

Belirtik olarak ayarlanmayan konunun herhangi bir özneliği, 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 öznelik, bu konunun üst konularının tanımladığı şekilde özneliği kullanır. Bu işlem, kök konu **SYSTEM.BASE.TOPIC** bulunduğu sürece konu ağacını yinelemeli. Konu ağaçlarıyla ilgili daha fazla bilgi için [Konu ağaçlarıbaşlıklı konuya](#) bakın.

```
DEFINE TOPIC (ORANGE.TOPIC) +  
        TOPICSTR (ORANGE) +  
        DEFPRTY(ASPARENT) +  
        NPMSGDLV(ASPARENT)
```

Not:

- Konu dizgisinin değeri dışında, gösterilen tüm öznelik değerleri varsayılan değerlerdir. Burada yalnızca bir şekil olarak gösterilir. Varsayılan değerlerin istediğiniz ya da değiştirilmemiş olduğundan eminseniz, bu değerleri atlayabilirsiniz. Ayrıca bkz. [“Yönetimle ilgili konu nesnesi özneliklerinin görüntülenmesi” sayfa 89](#).
- 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** özneliğini kullanın, ancak aynı zamanda bkz. [“Denetim konusu özneliklerinin değiştirilmesi” sayfa 90](#)

Yönetimle ilgili konu nesnesi özneliklerinin görüntülenmesi

Bir denetim konusu nesnesini görüntülemek için **DISPLAY TOPIC MQSC** komutunu kullanın.

Tüm konuları görüntülemek için şunu kullanın:

```
DISPLAY TOPIC (ORANGE.TOPIC)
```

Öznelikleri 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 kullanılan **ASPARENT** değerlerini görüntülemek için [DISPLAY TPSTATUS](#) seçeneğini kullanın. Örneğin, şunları kullanın:

```
DISPLAY TPSTATUS (ORANGE) DEFPRTY NPMSGDLV
```

Komut, aşağıdaki ayrıntıları görüntüler:

```
AMQ8754: Display topic status details.
        TOPICSTR(ORANGE)          DEFPRTY(0)
        NPMSGDLV(ALLAVAIL)
```

Bir denetim konusu tanımlarken, belirttiğiniz varsayılan denetim konusundan (SYSTEM.DEFAULT.TOPIC. Bu varsayılan özniteliklerin ne olduğunu görmek için aşağıdaki komutu kullanın:

```
DISPLAY TOPIC (SYSTEM.DEFAULT.TOPIC)
```

Denetim konusu özniteliklerinin değiştirilmesi

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

Örneğin, ORANGE.TOPIC, 5 olması için aşağıdaki komutlardan birini kullanın.

- **ALTER** komutunu kullanarak:

```
ALTER TOPIC(ORANGE.TOPIC) DEFPRTY(5)
```

Bu komut tek bir özniteliği değiştirir; bu konuya, bu konuya ilişkin varsayılan öncelik önceliği 5 'e teslim edilir; diğer tüm öznitelikler aynı kalır.

- **DEFINE** komutunu kullanarak:

```
DEFINE TOPIC(ORANGE.TOPIC) DEFPRTY(5) REPLACE
```

Bu komut, bu konuya teslim edilen iletilerin varsayılan önceliğini değiştirir. Diğer tüm özniteliklere varsayılan değerleri verilir.

Bu konuya gönderilen iletilerin önceliğini değiştiriyorsanız, var olan iletiler bundan etkilenmez. Ancak, yeni bir ileti, yayınlama uygulaması tarafından sağlanmıyorsa, belirtilen önceliği kullanın.

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 tanımını 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) +
        USEDQ(NO)
```

Denetim konusu tanımlamasının silinmesi

Bir yönetim konusunu silmek için **DELETE TOPIC** MQSC komutunu kullanabilirsiniz.

```
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 çözülen konu dizgisini kullanan uygulamalar, bu örnekte 'TURUNCU' olarak gösterilen 'TURUNCU' başlıklı konuyu kullanarak çalışmaya devam eder. Bu durumda, özellikler, konu ağacındaki bir konu nesnesindeki özellikleri devralır. Konu ağaçlarıyla ilgili daha fazla bilgi için [Konu ağaçları başlıklı konuya](#) bakın.

Aboneliklerle çalışma

Abonelikleri yönetmek için MQSC komutlarını kullanın.

Abonelikler, **SUBTYPE** öznelisinde 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 başvurusu](#) ' e bakın.

İlgili kavramlar

[“Yönetimle ilgili abonelik tanımlanması” sayfa 91](#)

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 özneliliklerini görüntüleme” sayfa 92](#)

Kuyruk yöneticisi tarafından bilinen herhangi bir aboneliğin yapılandırılmış özneliliklerini görüntülemek için **DISPLAY SUB** komutunu kullanabilirsiniz.

[“Yerel abonelik özneliliklerinin değiştirilmesi” sayfa 93](#)

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 93](#)

DEFINE komutundaki **LIKE** özneliliğini kullanarak bir abonelik tanımlamasını kopyalayabilirsiniz.

[“Aboneliği silme” sayfa 94](#)

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.
- Tanım ve konu dizgisinin değerleri dışında, gösterilen tüm öznitelik değerleri varsayılan değerlerdir. Burada yalnızca bir şekil olarak gösterilir. Varsayılan değerlerin istediğiniz ya da değiştirilmemiş olduğundan eminseniz, bu değerleri atlayabilirsiniz. Ayrıca bkz. [“Aboneliklerin özniteliklerini görüntüleme” sayfa 92.](#)
- TEST adlı kuyruk yöneticisiyle aynı kuyruk yöneticisiyle ilgili zaten 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 93.](#)
- SUBQ KUYRUĞU yoksa, bu komut başarısız olur.

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: WebSphere MQ subscription inquired.
SUBID(414D51204141412020202020202020EE921E4E20002A03)
SUB(ORANGE)                                TOPICSTR(ORANGE)
DURABLE(YES)
```

TOPICSTR, bu abonenin üzerinde çalıştığı çözümlenen konu dizgisidir. Bir abonelik, bir konu nesnesini kullanmak için tanımlandığında, o nesneden konu dizgisi, abonelik yapılırken sağlanan konu dizgisine önek olarak kullanılır. SUBID, abonelik yaratıldığında kuyruk yöneticisi tarafından atanan benzersiz bir tanıttıcıdır. Bazı abonelik adları uzun ya da farklı bir karakter kümelerinde kullanışsız olabileceği için, bu özellik görüntülenmek üzere yararlı bir öznitelige sahip olabilir.

Abonelikleri görüntülemek için kullanılacak diğer bir yöntem SUBID ' yi kullanmandır:

```
DISPLAY SUB +
  SUBID(414D512041414120202020202020EE921E4E20002A03) +
  TOPICSTR +
  DURABLE
```

Bu komut, aşağıdakiyle aynı çıkışa verir:

```
AMQ8096: WebSphere MQ subscription inquired.
SUBID(414D512041414120202020202020EE921E4E20002A03)
SUB(ORANGE)                                TOPICSTR(ORANGE)
DURABLE(YES)
```

Bir kuyruk yöneticisindeki yetkili sunucu abonelikleri varsayılan olarak görüntülenmez. Bunları görüntülemek için bir yetkili sunucu ya da TÖMÜ **SUBTYPE** belirtin.

Yürütme ortamı özniteliklerini görüntülemek için `DISPLAY SBSTATUS` komutunu kullanabilirsiniz. Örneğin, şu komutu kullanın:

```
DISPLAY SBSTATUS(ORANGE) NUMMSG
```

Aşağıdaki çıkış görüntülenir:

```
AMQ8099: WebSphere MQ subscription status inquired.  
SUB(ORANGE)  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
NUMMSG(0)
```

Bir yönetici aboneliği tanımladığınızda, bu abonelik varsayılan abonelikten belirtik 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)
```

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** komutunun kullanılması:

```
ALTER SUB(ORANGE) PUBPRTY(5)
```

Bu komut tek bir özniteliği değiştirir, bu aboneliğe teslim edilen iletilerin önceliğinin 5'e kadar kalmasını sağlar; diğer tüm öznitelikler aynı kalır.

- **DEFINE** komutunun kullanılması:

```
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.

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)
```

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(414D5120414141202020202020202020EE921E4E20002A03)
```

Abonelikte iletilerin denetlenmesi

Bu görev hakkında

Bir abonelik tanımlandığında, bu abonelik bir kuyrukla ilişkilendirilir. Bu abonelikte eşleşen yayınlanan iletiler bu kuyruđa konabiliyor.

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 öz niteliklerini görüntüleme” sayfa 92.
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 84.

Hizmetlerle çalışma

Hizmet nesneleri, ek işlemlerin bir kuyruk yöneticisinin parçası olarak yönetilebileceđi bir araçtır. Hizmetler ile, kuyruk yöneticisi başlatıldığında ve sona erdirildiğinde başlatılmış ve durdurulmuş programlar tanımlayabilirsiniz. IBM WebSphere MQ services are always started under the user ID of the user who started the queue manager.

Hizmet nesneleri aşağıdaki tiplerden biri olabilir:

Sunucu

Sunucu, **SUNUCU** olarak belirtilen **SERVTYPE** parametresine sahip bir hizmet nesnesidir. Sunucu hizmeti nesnesi, belirlenen bir kuyruk yöneticisi başlatıldığında yürütülen 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 (örneđin, **runmqtrm**).

Bir sunucu hizmeti nesnesinin yalnızca bir eşgörünümü kořutzamanlı olarak çalışabilir. Çalışmakta olan sunucu hizmeti nesnelерinin durumu, MQSC komutu kullanılarak izlenebilir, **DISPLAY SVSTATUS**.

Komut

Komut, **COMMAND** olarak belirtilen **SERVTYPE** parametresine sahip bir hizmet nesnesidir. Komut hizmeti nesneleri sunucu hizmeti nesnelерine benzerdir, ancak bir komut hizmeti nesnesinin birden çok eşgörünümü kořutzamanlı olarak çalışabilir ve bu nesnelерin durumu, MQSC komutu **DISPLAY SVSTATUS** kullanılarak izlenemez.

MQSC komutu, **STOP SERVICE** komutu yürütülürse, durdurma programı yürütülmeden önce MQSC komutu, **START SERVICE** tarafından başlatılmış olan programın hala etkin olup olmadığını saptamak için hiçbir denetim gerçekleştirilmez.

Hizmet Nesnesi Tanımlama

Çeşitli özniteliklere sahip bir hizmet nesnesi tanımlanınız.

Öznitelikler aşağıdaki gibidir:

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. Sunucu hizmeti nesnelerinin durumu, MQSC komutu kullanılarak izlenebilir, DISPLAY SVSTATUS.

Komut

Bir komut hizmeti nesnesi.

Bir komut hizmeti nesnesinin birden çok eşgörünümü koşutzamanlı 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 denetim, START SERVICE VE STOP SERVICE komutlarının kullanılmasıyla 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 95](#)

CONTROL parametresini kullanarak, bir hizmet nesnesinin yönetim ortamı başlatılır ve kuyruk yöneticisi tarafından otomatik olarak durdurulabilir ya da durdurulur ve MQSC komutları START SERVICE ve STOP SERVICE komutunu kullanarak başlatılır ve durdurulur.

Hizmetlerin yönetilmesi

CONTROL parametresini kullanarak, bir hizmet nesnesinin yönetim ortamı başlatılır ve kuyruk yöneticisi tarafından otomatik olarak durdurulabilir ya da durdurulur ve MQSC komutları START SERVICE ve STOP SERVICE komutunu kullanarak başlatılır ve durdurulur.

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 bir 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 96](#)

Bir hizmet başlatıldığında, hizmet işleminin başlatıldığı ortam kuyruk yöneticisinin ortamından devralınır. Tanımlamak istediğiniz değişkenleri `service.env` ortamı geçersiz kılma dosyalarından birine ekleyerek hizmet sürecinin ortamında ayarlanacak ek ortam değişkenleri tanımlanabilir.

Ek ortam değişkenleri

Bir hizmet başlatıldığında, hizmet işleminin başlatıldığı ortam kuyruk yöneticisinin ortamından devralınır. Tanımlamak istediğiniz değişkenleri `service.env` ortamı geçersiz kılma dosyalarından birine ekleyerek hizmet sürecinin ortamında ayarlanacak ek ortam değişkenleri tanımlanabilir.

Not:

Ortam değişkenlerini ekleyebileceğiniz iki olası dosya vardır:

- UNIX and Linux sistemlerinde `/var/mqm` dizininde ya da Windows sistemlerinde kuruluş sırasında seçilen veri dizininde bulunan makine kapsamı `service.env` kütüğü.
- Kuyruk yöneticisi veri dizininde bulunan kuyruk yöneticisi kapsamı `service.env` dosyası. Örneğin, `QMNAME` adlı bir kuyruk yöneticisi için ortam geçersiz kılma dosyasının yeri:
 - UNIX and Linux sistemlerinde, `/var/mqm/qmgrs/QMNAME/service.env`
 - Windows sistemlerinde, `C:\Program Files\IBM\WebSphere MQ\qmgrs\QMNAME\service.env`

Her iki dosya da (varsa) işlenir ve kuyruk yöneticisi kapsam dosyasındaki tanımlamalar, makine kapsamı dosyasındaki tanımlamalardan önceliklidir.

service.env içinde herhangi bir ortam değişkeni belirtilebilir. Örneğin, IBM WebSphere MQ hizmeti bir dizi komut çalıştırır, service.env dosyasında PATH kullanıcı değişkenini ayarlamak yararlı olabilir. Değişkeni ortam değişkenleri olarak ayarladığınız değerler olamaz; örneğin, CLASSPATH=%CLASSPATH% yanlış. Benzer şekilde, Linux PATH=\$PATH:/opt/mqm/bin üzerinde beklenmeyen sonuçlar verir.

CLASSPATH büyük harfle yazılmalı ve sınıf yolu deyimi yalnızca hazır bilgi içerebilir. Bazı hizmetler (örneğin Telemetri) kendi sınıf yollarını ayarlar. service.env içinde tanımlanan CLASSPATH buna eklenir.

Dosyada tanımlı değişkenlerin biçimi olanservice.env , ad ve değer değişkeni çiftlerinin bir listesidir. Her değişken yeni bir satırda tanımlanmalıdır ve her değişken, beyaz alan da içinde olmak üzere, belirtik olarak tanımlandığı şekilde alınır. service.env dosyasına bir örnek:

```
#*****#
#*                                     *#
#* <N_OCO_COPYRIGHT>                 *#
#* Licensed Materials - Property of IBM *#
#*                                     *#
#* 63H9336                             *#
#* (C) Copyright IBM Corporation 2005, 2024. *#
#*                                     *#
#* <NOC_COPYRIGHT>                   *#
#*                                     *#
#*****#
#*****#
#* Module Name: service.env           *#
#* Type        : WebSphere 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 97](#)

Bir hizmet nesnesi tanımlamasında, simgelerin yerine konması mümkündür. Hizmet programı yürütüldüğünde yerine koyulan simgeler, otomatik olarak genişletilmiş metniyle değiştirilir. Yerine koyma simgeleri, aşağıdaki ortak belirteçler listesinden ya da dosyada tanımlı olan değişkenlerden (service.env) alınabilir.

Hizmet tanımlarına değiştirilebilir ekler

Bir hizmet nesnesi tanımlamasında, simgelerin yerine konması mümkündür. Hizmet programı yürütüldüğünde yerine koyulan simgeler, otomatik olarak genişletilmiş metniyle değiştirilir. Yerine koyma simgeleri, aşağıdaki ortak belirteçler listesinden ya da dosyada tanımlı olan değişkenlerden (service.env) alınabilir.

Aşağıda, bir hizmet nesnesinin tanımındaki simgeleri yerine koymak için kullanılacak ortak simgeler yer alıyor:

MQ_INSTALL_PATH

WebSphere MQ ' un kurulu olduğu yer.

MQ_DATA_PATH

WebSphere MQ veri dizininin yeri:

- UNIX and Linux sistemlerinde, WebSphere MQ veri dizini yeri /var/mqm/olur.
- Windows sistemlerinde, WebSphere MQ veri dizininin yeri, WebSphere MQ kurulumu sırasında seçilen veri dizinidir.

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 değişkenleriyle başlatılan 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 daha fazla bilgi için bkz. [Understanding WebSphere MQ file names](#).

Değiştirilebilir eklemeler kullanmak için, simgeyi + karakterleri içine, STARTCMD, STARTARG, STOPCMD, STOPARG, STDOUT ya da STDERR dizilimlerinden birine takın. Buna örnek olarak bkz. [“Hizmet nesnelere ilişkin örnekler” sayfa 98](#).

Hizmet nesnelere ilişkin örnekler

Bu bölümdeki hizmetler, başka bir şekilde belirtilmediği sürece, UNIX stili yol ayırıcı karakterleriyle yazılır.

Sunucu hizmeti nesnesinin kullanılması

Bu örnekte, bir tetikleme izleyiciyi başlatmak için sunucu hizmeti nesnesinin nasıl tanımlanacak, kullanılacağı ve değiştirileceği gösterilmektedir.

1. Aşağıdaki MQSC komutu kullanılarak bir sunucu hizmeti nesnesi tanımlıdır:

```
DEFINE SERVICE(S1) +
  CONTROL(QMGR) +
  SERVTYPE(SERVER) +
  STARTCMD('+MQ_INSTALL_PATH+bin/!runmqtm') +
  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 WebSphere 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 işlem tanıtıcısını gösteren bir simgedir.

Ortak belirteçlerin listesi için bkz. [“Hizmet tanımlarına değiştirilebilir ekler” sayfa 97](#) .

2. Kuyruk yöneticisi bir sonraki başlatılırsa, sunucu hizmeti nesnesinin bir eşgörünümü yürütülür. Ancak, aşağıdaki MQSC komutu ile sunucu hizmeti nesnesinin bir eşgörünümünü hemen başlatacağız:

```
START SERVICE(S1)
```

3. Aşağıdaki MQSC komutu kullanılarak, sunucu hizmeti işleminin durumu görüntülenir:

```
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. Aşağıdaki 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. Sunucu hizmeti işlemi, aşağıdaki MQSC komutları kullanılarak değiştirilmek üzere yeniden başlatıldığı için yeniden başlatılır:

```
STOP SERVICE(S1)
```

Takip eden:

```
START SERVICE(S1)
```

Sunucu hizmeti işlemi yeniden başlatılır ve “4” sayfa 98’inde yapılan değişiklikleri alır.

Not: MQSC komutu, STOP SERVICE, yalnızca hizmet tanımında bir STOPCMD bağımsız değişkeni belirtilirse kullanılabilir.

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. Komut hizmeti nesnesi, aşağıdaki MQSC komutu kullanılarak tanımlanır:

```
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.

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. Komut hizmeti nesnesi, aşağıdaki MQSC komutu kullanılarak tanımlanır:

```
DEFINE SERVICE(S3) +
  CONTROL(QMGR) +
  SERVTYPE(COMMAND) +
  STOPCMD('/usr/bin/logger') +
  STOPARG('Queue manager +QMNAME+ stopping')
```

Burada:

logger , işletim sisteminin sistem günlüğüne giriş yazabilen WebSphere MQ ile verilen örnek bir programdır.
+QMNAME+ , kuyruk yöneticisinin adını gösteren bir simgedir.

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 stili yol ayırıcı karakterleri ile 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, aşağıdaki 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.

Hizmetin Otomatik Başlatılması

Bu örnekte, kuyruk yöneticisi başlatıldığında Tetikleme İzleyicisi 'nin otomatik olarak başlatılması için kullanılabilir bir sunucu hizmeti nesnesinin nasıl tanımlanabileceği gösterilmektedir.

1. Sunucu hizmeti nesnesi, aşağıdaki 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.

Tetikleme için nesneleri yönetme

WebSphere 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 nesneleri tanımlamanız gerekir.

Tetikleme, [WebSphere MQ uygulamalarının tetikleyicileri kullanarak başlatılması](#) içinde 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 özneliklerinin sayısını gerektirir.

Tetikleme, *Trigger* özneliği (MQSC komutlarındaki tetikleme) tarafından etkinleştirilir. In this example, a trigger event is to be generated when there are 100 messages of priority 5 or greater on the local queue MOTOR.INSURANCE.QUEUE, as follows:

```
DEFINE QLOCAL (MOTOR.INSURANCE.QUEUE) +
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
MAXMSGL (2000) +
DEFPSIST (YES) +
INITQ (MOTOR.INS.INIT.QUEUE) +
TRIGGER +
TRIGTYPE (DEPTH) +
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 özneliği 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 işlemek üzere kullanılacak uygulamayla ilişkilendirir. This is done through the PROCESS attribute on the

application queue MOTOR.INSURANCE.QUEUE. Aşağıdaki MQSC komutu gereken işlemi tanımlıyor: MOTOR.INSURANCE.QUOTE.PROCESS, bu örnekte tanımlanır:

```
DEFINE PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
DESCR ('Insurance request message processing') +
APPLTYPE (UNIX) +
APPLICID ('/u/admin/test/IRMP01') +
USERDATA ('open, close, 235')
```

Burada:

MOTOR.INSURANCE.QUOTE.PROCESS

Süreç tanımlamasının adı.

DESCR ('Insurance request message processing')

Bu tanımın ilişkilendirdiği uygulama programını açıklar. Bu metin, DISPLAY PROCESS komutunu kullandığınızda görüntülenir. Bu, işlemin ne yaptığını tanımlamanıza yardımcı olabilir. 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.

Uzak IBM WebSphere MQ nesnelere yönetme

Bu kısımda, MQSC komutları kullanılarak uzak kuyruk yöneticilerindeki IBM WebSphere MQ nesnelere nasıl denetleneceği ve iletilerin ve yanıt iletilerinin hedefini denetlemek için uzak kuyruk nesnelere nasıl kullanılacağı anlatılıyor.

Bu bölümde aşağıdakiler anlatılır:

- [“Kanallar, kümeler ve uzak kuyruğa alma” sayfa 102](#)
- [“Yerel kuyruk yöneticisinden uzak denetim” sayfa 104](#)
- [“Uzak kuyruğun yerel tanımlaması yaratılması” sayfa 110](#)
- [“Uzak kuyruk tanımlamalarının diğer adlar olarak kullanılması” sayfa 112](#)
- [“Kodlanmış karakter kümeleri arasında veri dönüştürme” sayfa 113](#)

Kanallar, kümeler ve uzak kuyruğa alma

Kuyruk yöneticisi, bir ileti göndererek, yanıt almak için bir ileti ve gerekiyorsa, başka bir kuyruk yöneticisiyle iletişim kurar. Alma kuyruk yöneticisi aşağıdaki gibi olabilir:

- Aynı makinede
- Aynı konumdaki başka bir makinede (ya da dünyanın diğer tarafında bile)
- Yerel kuyruk yöneticisiyle aynı altyapıda çalıştırma
- WebSphere MQ tarafından desteklenen başka bir altyapıda çalıştırma

Bu iletiler aşağıdakilerden kaynaklanabilir:

- Verileri bir düğümden diğerine aktaran kullanıcı-yazılı uygulama programları
- PCF komutlarını ya da MQAI olanağını kullanan kullanıcı tarafından yazılan yönetim uygulamaları
- IBM WebSphere MQ Gezgini.
- Gönderen kuyruk yöneticileri:
 - Başka bir kuyruk yöneticisine kod ekleme olayı iletileri
 - Dolaylı kipte (komutların başka bir kuyruk yöneticisinde çalıştırıldığı) bir runmqsc komutundan yayınlanan MQSC komutları

Uzak kuyruk yöneticisine bir ileti gönderilmeden önce, yerel kuyruk yöneticisinin iletilerin gelişini algılamak ve bunları aşağıdakilerden oluşan bir yöntemle aktarmak için bir mekanizmaya gerek vardır:

- En az bir kanal
- İletim kuyruğu
- Kanal başlatıcısı

Uzak kuyruk yöneticisi için bir ileti alınmak üzere dinleyici gereklidir.

Kanal, iki kuyruk yöneticisi arasında tek yönlü bir iletişim bağlantısıdır ve uzak kuyruk yöneticisinde herhangi bir sayıda kuyruk için gönderilen iletileri taşıyabilir.

Kanalın her bir ucunda ayrı bir tanım vardır. Örneğin, bir son gönderen ya da sunucu ise, diğer 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 MQSC komutu DEFINE CHANNEL ' ' ı kullanın. Bu bölümde, başka bir şekilde belirtilmediği sürece, kanallarla ilgili örnekler varsayılan kanal özniteliklerini kullanır.

İletilerin her birinde bir ileti kanalı aracısı (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 105 , 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.

Kümeleri kullanarak uzaktan yönetim

Dağıtılmış kuyruklama kullanan bir WebSphere 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.

Kümeler, öznitelikleri ve bunların ayarlarıyla ilgili daha fazla bilgi için bkz. [Kuyruk yöneticisi kümeleri](#).

Yerel kuyruk yöneticisinden uzak denetim

Bu kısımda, uzak kuyruk yöneticisinin MQSC ve PCF komutlarını kullanarak yerel bir kuyruk yöneticisinden nasıl denetleneceği anlatılıyor.

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 yazma hakkında daha fazla bilgi için, bkz. [“Programlanabilir Komut Biçimlerinin Kullanılması” sayfa 10](#).

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. You can remotely administer queue managers in other WebSphere MQ environments, including UNIX and Linux systems, Pencereler systems, IBM i, and z/OS.

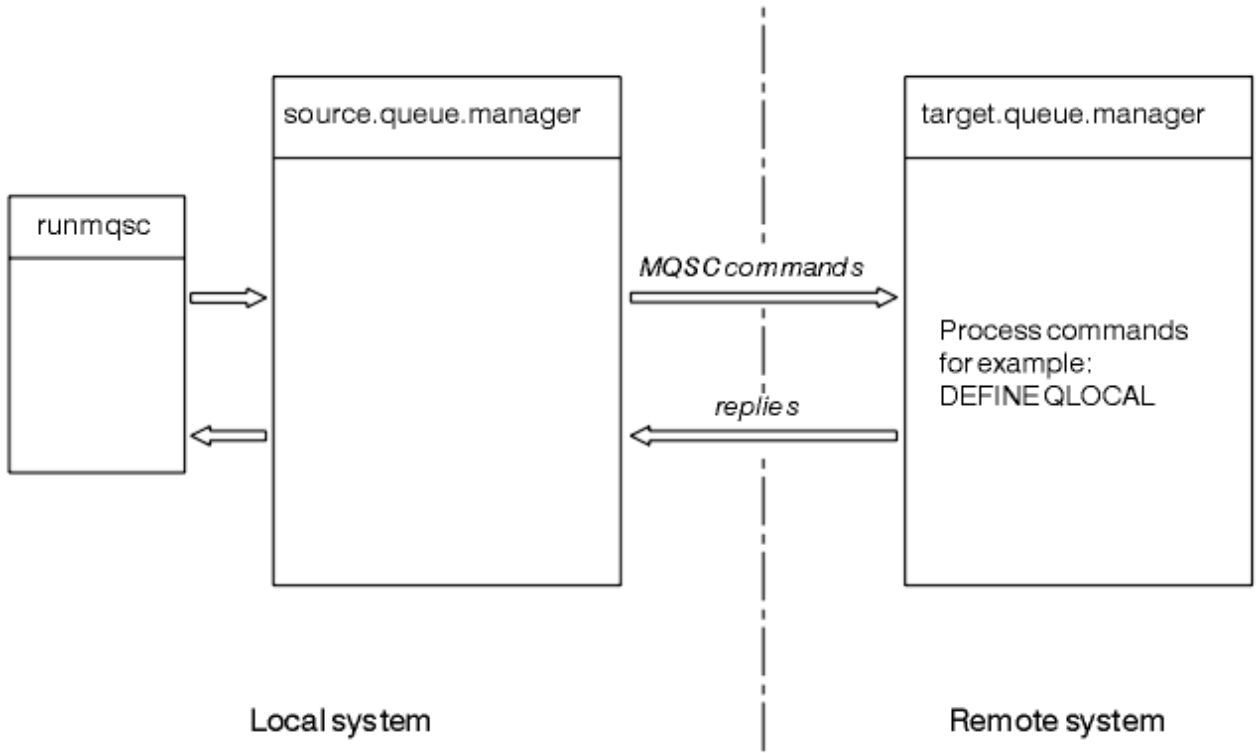
Uzak denetimi uygulamak için belirli nesnelere yaratmanız gerekir. Özel gereksiniminiz yoksa, varsayılan değerler (örneğin, ileti uzunluğu üst sınırı için) yeterli olur.

Uzak denetim için kuyruk yöneticilerinin hazırlanması

Uzak denetim için kuyruk yöneticilerini hazırlamak için MQSC komutları nasıl kullanılır.

Şekil 17 sayfa 105 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.

Not: -w seçeneği ile **runmqsc** kullanıyorsanız, varsayılan kuyruk yöneticisi `source.queue.manager` **gerekir** olmalıdır. Kuyruk yöneticisi yaratılmasına ilişkin ek bilgi edinmek için `crtmqmdosyasına` bakın.



Şekil 17. MQSC komutlarını kullanarak uzaktan yönetim

Her iki sistemde de, henüz yapmadıysanız:

- `crtmqm` komutunu kullanarak kuyruk yöneticisini ve varsayılan nesnelere yaratın.
- Start the queue manager, using the `strmqm` command.

Hedef kuyruk yöneticisinde:

- Komut kuyruğu, `SYSTEM.ADMIN.COMMAND.QUEUE`(Kuyruk), var olmalıdır. Bu kuyruk, bir kuyruk yöneticisi yaratıldığında varsayılan olarak yaratılır.

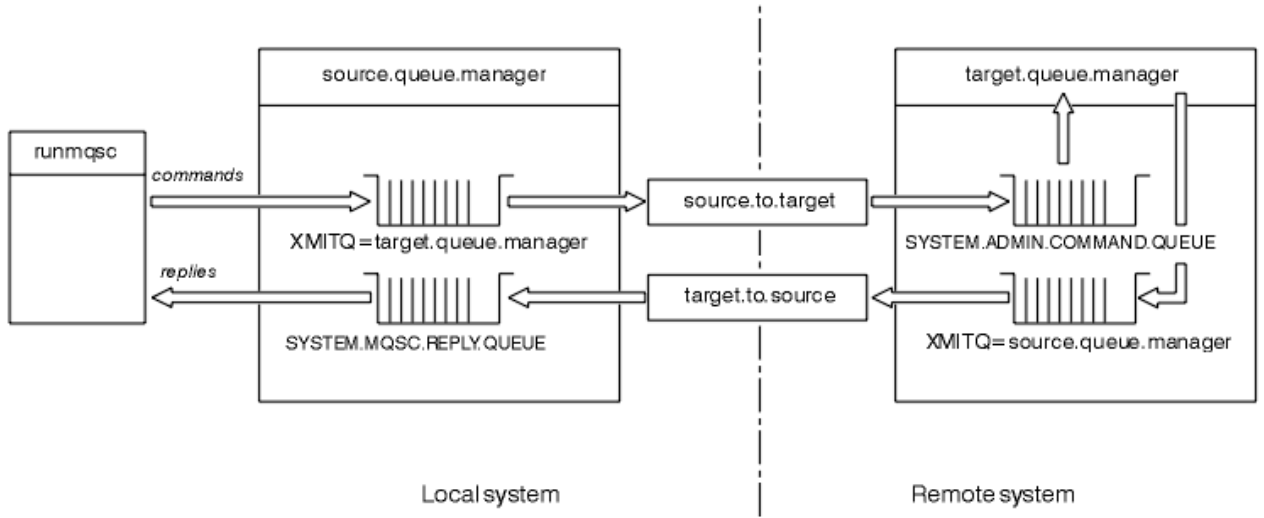
Bu komutları yerel olarak ya da Telnet gibi bir ağ olanağı üzerinden çalıştırmanız gerekir.

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 18 sayfa 106](#) bu yapılandırmayı özetler.



Şekil 18. Uzak denetim için kanalların ve kuyrukların ayarlanması

Kanalların ayarlanmasıyla ilgili daha fazla bilgi için [Dağıtılmış kuyruklama kullanarak uygulamaların bağlanması](#) 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 kanalı tanımlayın:

```
DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(SDR) +
  CONNAME (RHX5498) +
  XMITQ ('target.queue.manager') +
  TRPTYPE(TCP)
```

2. Kaynak kuyruk yöneticisinde günlük nesnesi kanalı 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) +
  CONNAME (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 CONNAME özneliği 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 kanalı başlatın:

```
START CHANNEL ('source.to.target')
```

2. Aşağıdaki MQSC komutunu vererek, hedef kuyruk yöneticisinde (`target.queue.manager`) gönderen kanalı 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.

WebSphere MQ , gelen bağlantı isteği alırsa ve uygun bir alıcı ya da sunucu bağlantısı kanalı bulamazsa, otomatik olarak bir kanal yaratır. Otomatik tanımlamalar, WebSphere MQ: SYSTEM.AUTO.RECEIVER ve SYSTEM.AUTO.SVRCONN.

Kanal tanımlamalarının otomatik olarak yaratılmasına ilişkin ek bilgi için [Kanalların hazırlanması](#) başlıklı konuya bakın. Kümelere ilişkin kanalları otomatik olarak tanımlama hakkında bilgi için [Küme kanallarının otomatik olarak tanımlanması](#) başlıklı konuya bakın.

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. Providing the command server is running, users of WebSphere MQ for Pencereler or WebSphere MQ for Linux (x86 and x86-64 platforms) can perform the operations described in the following sections using the IBM WebSphere MQ Explorer. Daha fazla bilgi için [“IBM WebSphere MQ Explorerkomutunu kullanarak yönetim”](#) sayfa 55 başlıklı konuya bakın.

Komut sunucusu başlatılıyor

Kuyruk yöneticisi özniteliğinin 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:

```
stmqcsv 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 endmqcsv 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.

The command server **gerekir** be running on the target queue manager, if it is going to process MQSC commands remotely. (Bu, kaynak kuyruk yöneticisinde gerekli değildir). Bir kuyruk yöneticisine komut sunucusunun nasıl başlatılabilmesiyle ilgili bilgi için bkz. [“Uzak denetim için komut sunucusunun yönetilmesi” sayfa 108.](#)

Kaynak kuyruk yöneticisinde, aşağıdaki komutu yazarak, MQSC komutlarını dolaylı kipte etkileşimli olarak çalıştırabilirsiniz:

```
runmqsc -w 30 target.queue.manager
```

`runmqsc` komutunun bu biçimi, `-w` işareti ile birlikte MQSC komutlarını dolaylı kipte çalıştırır; burada komutlar komut sunucusu giriş kuyruğunda (değiştirilmiş bir biçimde) konulsun ve sırayla yürütülür.

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` 'ı 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 78.

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 yapabildiğini sağlar.

Uzak kuyruklar için yerel tanımların nasıl çalıştığını anlama

Bir uygulama yerel bir kuyruk yöneticisine bağlanır ve daha 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 bir ileti yerleştirmek için uygulama, MQOPEN çağrısından döndürülen tutamacı 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 noktasına yönlendirmek için kullanılır.

Denetimci olarak, uzak kuyruk tanımlamasını değiştirerek iletinin hedef noktasını denetleyebilirsiniz.

Aşağıdaki örnekte, bir uygulamanın uzak kuyruk yöneticisine ait bir kuyruğa nasıl ileti yerleştirdiğini gösteren örnek gösterilmektedir. Uygulama bir kuyruk yöneticisine bağlanıyor; örneğin, saturn.queue.manager. Hedef kuyruğun sahibi başka bir kuyruk yöneticisi var.

On the MQOPEN call, the application specifies these fields:

Alan değeri	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) +
  QMGRNAME (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 yöneticilerindeki 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 kuyrukunun 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 özniteliklerini görüntülemek ya da değiştirmek için MQSC komutlarını kullanabilir ya da uzak kuyruk nesnesini silebilirsiniz. Örneğin:

- Uzak kuyruğun özniteliklerini görüntülemek için:

```
DISPLAY QUEUE (CYAN.REMOTE.QUEUE)
```

- 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 kuyruktır.

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 özniteliği 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 özniteliğe 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 özniteliğe 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 110.](#)

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 bir 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 kümeleri](#) başlıklı konuya bakın.

Kodlanmış karakter kümeleri arasında veri dönüştürme

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

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 dönüştürmesi](#)' 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. Varsayılan veri dönüştürme işlemi "[Varsayılan veri dönüştürme](#)" [sayfa 113](#) içinde açıklanmıştır.

Dosya ccsid.tbl

ccsid.tbl dosyası aşağıdaki amaçlar için kullanılır:

- WebSphere MQ ' da Windows için, desteklenen tüm kod kümelerini kaydeder.
- AIX ve HP-UX platformlarında, desteklenen kod kümeleri işletim sistemi tarafından dahili olarak tutulur.
- Diğer tüm UNIX and Linux platformlarında, desteklenen kod kümeleri WebSphere MQ tarafından sağlanan dönüştürme çizelgelerinde 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 içinde kaydedilen bilgileri güncelleştirebilirsiniz; örneğin, işletim sisteminizin gelecekteki bir yayın düzeyi ek kodlanmış karakter kümelerini destekliyorsa bunu yapmak isteyebilirsiniz.

WebSphere MQ for Pencereleri içinde, ccsid.tbl varsayılan olarak C:\Program Files\IBM\WebSphere MQ\conv\tabl dizininde bulunur.

WebSphere MQ for UNIX and Linux systems, ccsid.tbl , /var/mqm/conv/taabledizinde bulunur.

Varsayılan veri dönüştürme

Veri dönüştürmenin normalde desteklenmediği iki makine arasında kanal ayarlarsanız, kanalların çalışabilmesi için varsayılan veri dönüştürme özelliğini etkinleştirmeniz gerekir.

Varsayılan veri dönüştürme işlemi geçerli kılmak için, ccsid.tbl kütüğünü düzenleyerek, varsayılan EBCDIC CCSID değerini ve varsayılan ASCII CCSID değerini belirleyin. 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, WebSphere 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 WebSphere MQ nesne adları için geçerli karakterler kümesi (Adlandırma IBM WebSphere MQ nesnelere tanımlandığı şekilde) kullanıyorsanız, genel olarak bu gereksinimi karşılayın. Japonya 'da kullanılan EBCDIC CCSID 'ler 290, 930, 1279 ve 5026 ile özel durumlar ortaya çıkar; burada, küçük harflerin diğer 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.

Yönetme IBM WebSphere MQ Telemetry

IBM WebSphere MQ Telemetry , IBM WebSphere MQ Explorer ya da bir komut satırı kullanılarak yönetilir. Telemetri kanallarını yapılandırmak, telemetri hizmetini denetlemek ve IBM WebSphere MQ' a bağlı MQTT istemcilerini izlemek için kaşifi kullanın. Configure the security of IBM WebSphere MQ Telemetry using JAAS, SSL and the IBM WebSphere MQ object authority manager.

IBM WebSphere MQ Explorer komutunu kullanarak yönetme

Telemetri kanallarını yapılandırmak, telemetri hizmetini denetlemek ve IBM WebSphere MQ' a bağlı MQTT istemcilerini izlemek için kaşifi kullanın. Configure the security of IBM WebSphere MQ Telemetry using JAAS, SSL and the IBM WebSphere MQ object authority manager.

Komut satırını kullanarak yönetme

IBM WebSphere MQ Telemetry can be completely administered at the command line using the IBM WebSphere MQ [MQSC](#) commands.

IBM WebSphere MQ Telemetry belgelerinde ayrıca, MQ Telemetry Transport v3 Client uygulamasının temel kullanımını gösteren örnek komut dosyaları da vardır.

Read and understand the samples in [IBM WebSphere MQ Telemetry örnek programları](#) in the [IBM WebSphere MQ Telemetry için uygulama geliştirilmesi](#) section before using them.

İlgili kavramlar

[WebSphere MQ Telemetry](#)

[“MQTT istemcilerine ileti göndermek için dağıtım kuyruklama yapılandırma” sayfa 118](#)

IBM WebSphere MQ uygulamaları, MQTT v3 istemcileri iletilerini bir istemci tarafından oluşturulan aboneliğe yayınlayarak 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.

[“MQTT istemcisi tanıtıcısı, yetki kimliği ve kimlik doğrulaması” sayfa 121](#)

[“SSL kullanan telemetri kanalı kimlik doğrulaması” sayfa 127](#)

[“telemetri kanallarında yayın gizliliği” sayfa 129](#)

[“MQTT istemcilerinin ve telemetri kanallarının SSL yapılandırması” sayfa 130](#)

[“Telemetri kanalı JAAS yapılandırması” sayfa 135](#)

İstemci tarafından gönderilen `Kullanıcı` adı'nın kimliğini doğrulamak için JAAS 'ı yapılandırın.

[“Aygıtlar kavramları için IBM WebSphere MQ Telemetry cini” sayfa 137](#)

Aygıtlar için IBM WebSphere MQ Telemetry cini gelişmiş bir MQTT V3 istemci uygulamasıdır. İletileri diğer MQTT istemcilerinden saklamak ve iletmek için kullanın. Bu, IBM WebSphere MQ ile bir MQTT istemcisi gibi bağlanır, ancak diğer MQTT istemcilerini de bu istemciye bağlayabilirsiniz.

İlgili görevler

[“Linux ve AIX üzerinde telemetri için kuyruk yöneticisi yapılandırılması” sayfa 115](#)

Follow these manual steps to configure a queue manager to run IBM WebSphere MQ Telemetry. You can run an automated procedure to set up a simpler configuration using the IBM WebSphere MQ Telemetry support for IBM WebSphere MQ Explorer.

[“Windows üzerinde telemetri için kuyruk yöneticisi yapılandırılması” sayfa 117](#)

Follow these manual steps to configure a queue manager to run IBM WebSphere MQ Telemetry. You can run an automated procedure to set up a simpler configuration using the IBM WebSphere MQ Telemetry support for IBM WebSphere MQ Explorer.

İlgili başvurular

[MQXR özellikleri](#)

Linux ve AIX üzerinde telemetri için kuyruk yöneticisi yapılandırılması

Follow these manual steps to configure a queue manager to run IBM WebSphere MQ Telemetry. You can run an automated procedure to set up a simpler configuration using the IBM WebSphere MQ Telemetry support for IBM WebSphere MQ Explorer.

Başlamadan önce

1. IBM WebSphere MQ ve IBM WebSphere MQ Telemetry özelliğinin nasıl kurulabilmesiyle ilgili bilgi için bkz. [IBM WebSphere 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`. Hemen hemen tüm ayarların MQSC denetim komutları ya da MQ Gezgini ile yapılandırılabilirdiği için, MQXR özellikler dosyasını doğrudan düzenlemenize gerek yoktur. Dosyayı doğrudan düzenlemeye karar verdiyseniz, değişikliklerinizi yapmadan önce kuyruk yöneticisini durdurun. Bkz. [MQXR özellikleri](#).

Bu görev hakkında

IBM WebSphere MQ Explorer için IBM WebSphere MQ Telemetry desteği bir sihirbaz ve örnek komut yordamı `sampleMQM` içerir. Konuk kullanıcı kimliğini kullanarak ilk yapılandırmayı kurdular; bkz. [IBM WebSphere MQ Telemetry kuruluşunun IBM WebSphere MQ Explorer komutunu kullanarak doğrulanması ve IBM WebSphere MQ Telemetry örnek programları](#).

Follow the steps in this task to configure IBM WebSphere 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` 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. 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 WebSphere MQ Explorer or with the command in the following example:

```
echo "ALTER QMGR DEFQMTT('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 göndermekten daha kolay hale getirmemektir. Varsayılan iletim kuyruğunu değiştirmeden, IBM WebSphere MQ iletilerini alan her istemci için bir uzak kuyruk tanımı eklemelisiniz; bkz. [“İstemciye doğrudan ileti gönderme” sayfa 120.](#)

4. Bir ya da daha çok kullanıcı kimliği yaratmak için “MQTT istemcilerinin WebSphere MQ nesnelere erişmesi için yetki verme” sayfa 122 içindeki bir yordamı izleyin. Kullanıcı kimlikleri, bu 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 19 sayfa 116](#) 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 WebSphere MQ Explorer komutunu kullanarak, telemetri kanallarını MQTT istemcilerinden 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 WebSphere MQ Telemetry örnek programları](#).

Örnek

[Şekil 19 sayfa 116](#) 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 19. installMQXRService_unix.mqsc

Windowsüzerinde telemetri için kuyruk yöneticisi yapılandırılması

Follow these manual steps to configure a queue manager to run IBM WebSphere MQ Telemetry. You can run an automated procedure to set up a simpler configuration using the IBM WebSphere MQ Telemetry support for IBM WebSphere MQ Explorer.

Başlamadan önce

1. IBM WebSphere MQve IBM WebSphere MQ Telemetry özelliğinin nasıl kurulabilmesiyle ilgili bilgi için bkz. [IBM WebSphere MQ Telemetrykuruluyor](#) .
2. Kuyruk yöneticisi yarat ve başlat. Kuyruk yöneticisi bu görevde *qMgr* olarak anılır.
3. Bu görevin bir parçası olarak, telemetri (MQXR) hizmetini yapılandırıyorsunuz. MQXR özellik ayarları, platforma özgü özellikler dosyasında depolanır: `mqxr_win.properties`. Hemen hemen tüm ayarların MQSC denetim komutları ya da MQ Gezgini ile yapılandırılabilirdiği için, MQXR özellikler dosyasını doğrudan düzenlemenize gerek yoktur. Dosyayı doğrudan düzenlemeye karar verdiyseniz, değişikliklerinizi yapmadan önce kuyruk yöneticisini durdurun. Bkz. [MQXR özellikleri](#).

Bu görev hakkında

IBM WebSphere MQ Explorer için IBM WebSphere MQ Telemetry desteği bir sihirbaz ve örnek komut yordamı `sampleMQMıçerir`. Konuk kullanıcı kimliğini kullanarak ilk yapılandırmayı kurdular; bkz. [IBM WebSphere MQ Telemetry kuruluşunun IBM WebSphere MQ Explorer komutunu kullanarak doğrulanmasıve IBM WebSphere MQ Telemetry örnek programları](#).

Follow the steps in this task to configure IBM WebSphere 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` 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

```
echo ALTER QMGR DEFQMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```

Şekil 20. 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 WebSphere MQ Explorer or with the command in [Şekil 20 sayfa 117](#).

Varsayılan iletim kuyruğunu değiştirme, var olan yapılandırmanızı engelleyebilir. Varsayılan iletim kuyruğunu `SYSTEM.MQTT.TRANSMIT.QUEUE` ' e değiştirme nedeni, iletileri doğrudan MQTT istemcilerine göndermekten daha kolay hale getirmemektir. Varsayılan iletim kuyruğunu değiştirmeden, IBM WebSphere MQ iletilerini alan her istemci için bir uzak kuyruk tanımı eklemelisiniz; bkz. [“İstemciye doğrudan ileti gönderme” sayfa 120](#).

4. Bir ya da daha çok kullanıcı kimliği yaratmak için “MQTT istemcilerinin WebSphere MQ nesnelere erişmesi için yetki verme” sayfa 122 içindeki bir yordamı izleyin. Kullanıcı kimlikleri, bu 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 WebSphere MQ Explorer komutunu kullanarak, telemetri kanallarını MQTT istemcilerinden 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 WebSphere MQ Telemetry örnek programları](#).

Creating SYSTEM.MQXR.SERVICE manually

Şekil 21 sayfa 118 , Windows üzerinde SYSTEM.MQXR.SERVICE komutunu el ile oluşturmak için **runmqsc** komutunu gösterir.

```
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 21. *installMQXRService_win.mqsc*

MQTT istemcilerine ileti göndermek için dağıtımlı kuyruklama yapılandır

IBM WebSphere 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.

Bir MQTT istemcisi aboneliğine yanıt olarak bir ileti yayınlıyor

teletext (MQXR) hizmeti, MQTT istemcisi adına bir abonelik yaratır. İstemci, istemci tarafından gönderilen aboneliğe eşleşen yayınların hedefidir. Telemetri hizmetleri, eşleşen yayınları istemciye iletir.

Bir MQTT istemcisi WebSphere MQ 'ya kuyruk yöneticisi olarak, kuyruk yöneticisi adı da ClientIdentifier değerine ayarlanmış olarak bağlanır. İstemciye gönderilecek yayınların hedefi bir iletim kuyruğudur (SYSTEM.MQTT.TRANSMIT.QUEUE). Telemetri hizmeti, hedef kuyruk yöneticisi adını belirli bir istemcinin anahtarı olarak kullanarak, SYSTEM.MQTT.TRANSMIT.QUEUE üzerindeki iletileri MQTT istemcilerine iletir.

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 WebSphere MQ nesne adı çözünürlüğünü kullanarak, *ClientIdentifier* şu şekilde çözümlenir; bkz. Çizelge 6 sayfa 119.

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 eşleşiyor.

Kuyruk adı tanımlı değil. Şu anda, telemetri (MQXR) hizmeti, kuyruğun adı olarak SYSTEM.MQTT.PUBLICATION.QUEUE 'yi ayarlar. MQTT v3 istemcisi kuyrukları desteklemediğinden, çözülen kuyruk adı istemci tarafından yoksayılr.

Yerel kuyruk yöneticisi özelliği, Varsayılan iletim kuyruğu, ad SYSTEM.MQTT.TRANSMIT.QUEUE olarak ayarlanmalıdır; böylece, yayın istemciye gönderilecek SYSTEM.MQTT.TRANSMIT.QUEUE 'a konmalıdır.

2. *ClientIdentifier*, *ClientIdentifier* adlı bir kuyruk yöneticisi diğer adıyla eşleşiyor.

ClientIdentifier, uzak bir kuyruk yöneticisi adıdır. Kuyruk yöneticisi diğer adı adıyla eşleşir. Kuyruk yöneticisi diğer adı, uzak kuyruk yöneticisi adı olarak *ClientIdentifier* ile tanımlanmalıdır.

Kuyruk yöneticisi diğer adı tanımlamasındaki iletim kuyruğu adını ayarlayarak, varsayılan iletim için SYSTEM.MQTT.TRANSMIT.QUEUE değerine ayarlanmalıdır.

Çizelge 6. MQTT kuyruk yöneticisi diğer adına ilişkin ad çözümlemesi					
	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 WebSphere MQ programı aynı konuya yayınlanabilir. Bu yayın abonelerine gönderilir. Bu, konuya aboneliği olan MQTT v3 istemcileri de içinde olmak üzere 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 22. Windows' da bir küme başlığı tanımlama

Not: SYSTEM.MQTT.TRANSMIT.QUEUE bir küme özniteliği vermemek için.

MQTT istemcisi 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 WebSphere MQ.

İstemciye doğrudan ileti gönderme

Abonelik yaratan ve abonelik konularıyla eşleşen bir yayını alan bir istemciye alternatif olan bir alternatif, doğrudan MQTT v3 istemcisine bir ileti gönderir. MQTT V3 istemci uygulamaları doğrudan ileti gönderemez; ancak, WebSphere MQ uygulamaları gibi başka bir uygulama da gönderilebilir.

WebSphere MQ uygulaması, MQTT v3 istemcisinin `ClientIdentifier` ögesini bilmelidir. MQTT v3 istemcilerinin kuyrukları olmadığından, hedef kuyruk adı, bir konu adı olarak MQTT v3 uygulama istemcisi `messageArrived` yöntemine geçirilir. Örneğin, bir MQI programında, istemciyle `ObjectQmgrAdı` olarak bir nesne tanımlayıcısı yaratın:

```
MQOD.ObjectQmgrName = ClientIdentifier;  
MQOD.ObjectName = name;
```

Şekil 23. MQI Nesne tanımlayıcısı, bir MQTT v3 istemci hedefine ileti gönderisin

Uygulama JMS kullanılarak yazıldıysa, bir noktadan noktaya iletişim hedefi yaratın; örneğin:

```
javax.jms.Destination jmsDestination =  
    (javax.jms.Destination)jmsFactory.createQueue  
    ("queue://ClientIdentifier/name");
```

Şekil 24. 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 çözülmelidir. İletim kuyruğu `SYSTEM.MQTT.TRANSMIT.QUEUE` 'e çözülmelidir; bkz. [Çizelge 7 sayfa 120](#). Uzak kuyruk adı herhangi bir değer olabilir. İstemci bunu bir konu dizgisi olarak alır.

Çizelge 7. MQTT istemcisi uzak kuyruk tanımlamasının ad çözümlenmesi				
Giriş		Çıkış		
Kuyruk adı	Kuyruk yöneticisi adı	Kuyruk adı	Kuyruk yöneticisi adı	İletim kuyruğu
Uzak kuyruk tanımlamasının adı	Boş ya da yerel kuyruk yöneticisi adı	Bir konu dizgisi olarak kullanılan uzak kuyruk adı	<code>ClientIdentifier</code>	<code>SYSTEM.MQTT.TRANSMIT.QUEUE</code>

İstemci bağlıysa, ileti doğrudan `messageArrived` yöntemini çağıran MQTT istemcisine gönderilir; bkz. `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 "at most once" quality of service, QoS=0. Kalıcı bir iletiyi doğrudan bir istemciye gönderdiyseniz, varsayılan olarak bu ileti "tam olarak bir kez" hizmet kalitesi QoS=2 ile gönderilir. İstemci kalıcı bir kalıcılık düzeneğine sahip olmayabilir gibi, istemci doğrudan gönderilen iletiler için kabul ettiği hizmet kalitesini düşürebilir. Bir istemciye doğrudan gönderilen iletiler için hizmet kalitesini düşürmek için, `DEFAULT.QoS` konusuna bir abonelik yapın. İstemcinin destekleyebileceği hizmet kalitesi üst sınırını belirtin.

MQTT istemcisi tanıtıcısı, yetki kimliği ve kimlik doğrulaması

Telemetri (MQXR) hizmeti, MQTT istemcileri adına, MQTT kanallarını kullanarak WebSphere MQ konularına abone olur ya da abone olur. WebSphere MQ yöneticisi, WebSphere MQ yetkilendirmesi için kullanılan MQTT kanal kimliğini yapılandırır. Yönetici, kanal için ortak bir kimlik tanımlayabilir ya da kanala bağlı bir istemcinin `Kullanıcı Adı` ya da `ClientIdentifier` '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, bir MQTT kanalını kullanarak sunucuya gönderilen `Kullanıcı adı` ve `Parola` adını ayarlar. Ayrıca, bağlantıyı şifrelemek ve bağlantıyı doğrulamak için gereken SSL özelliklerini de ayarlayabilirler. Yönetici, MQTT kanalının doğrulanıp doğrulanmayacağını ve kanalın nasıl doğrulanıp doğrulanmayacağını belirler.

Bir MQTT istemcisine WebSphere MQ nesnelere erişim yetkisi vermek için, istemcinin `ClientIdentifier` ya da `Kullanıcı Adı` 'ini yetkilendirin ya da ortak bir istemci kimliğini yetkilendirin. Bir istemcinin WebSphere MQ'ya bağlanmasına izin vermek için, `Kullanıcı adı` 'nın kimliğini doğrulayın ya da bir istemci sertifikası kullanın. `Kullanıcı Adı` kimliğini doğrulamak için JAAS 'u yapılandırın ve bir istemci sertifikasının kimliğini doğrulamak için SSL' yi yapılandırın.

İstemcide bir `Parola` ayarlıysa, VPN 'yi kullanarak bağlantıyı şifreleyin ya da parolayı özel tutmak için MQTT kanalını SSL 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. Aygıtta güvenilen, zaman sınırlanmış bir istemci sertifikası kopyalanmıştır. Aygıt, kullanılabileceği yere konuşlandırılır. Aygıt her kullanımında, parola kullanılarak ya da akıllı karttan başka bir sertifika kullanılarak daha fazla kimlik doğrulaması gerçekleştirilir.

MQTT istemcisi kimliği ve yetkilendirmesi

WebSphere MQ nesnelere erişmek için yetkilendirme için `ClientIdentifier`, `Kullanıcı adı` ya da ortak bir istemci kimliği kullanın.

WebSphere MQ denetimcisinin, MQTT kanalının kimliğini seçmek için üç seçeneği vardır. Yönetici, istemci tarafından kullanılan MQTT kanalını tanımlarken ya da değiştirirken seçimi yapar. Kimlik, WebSphere MQ konularına erişimi yetkilendirmek için kullanılır. Seçenekler şunlardır:

1. İstemci tanıtıcısı.
2. Yöneticinin kanal için sağladığı kimlik.
3. The `Kullanıcı Adı` passed from the MQTT client.

`Kullanıcı adı` , `MqttConnectSeçenekleri` sınıfının bir öznesidir. İstemcinin hizmete bağlamadan önce ayarlanması gerekir. Varsayılan değeri boş değerli.

MQTT kanalına ilişkin kimliğin hangi nesnelere ve hangi işlemlerin kullanılacağını seçmek için WebSphere MQ **setmqaut** komutunu kullanın. For example, to authorize a channel identity, MQTTClient, provided by the administrator of queue manager, QM1:

```
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
```

MQTT istemcilerinin WebSphere MQ nesnelere erişmesi için yetki verme

MQTT istemcilerine yayınlama ve WebSphere MQ Nesnelere abone olma yetkisi vermek için bu adımları izleyin. Bu adımlar, dört alternatif erişim denetimi örüntülerini izlemektedir.

Başlamadan önce

MQTT istemcilerinin, bir telemetry kanalına bağlandığında bir kimlik atanarak WebSphere MQ 'daki nesnelere erişim yetkisi vardır. The WebSphere MQ Administrator configures the telemetry channel using WebSphere 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 WebSphere MQ olarak tanımlanmalıdır. The default authorization service on Pencereler 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, WebSphere MQ' da tanımlanan konuları yayınlama ya da abone olma izni vermek için bu tanıtıcıyı kullanın. Bir MQTT istemcisi bir konuya abone olmuşsa, elde edilen yayınları almak için gereken izni vermek için kimliği kullanın.

Her biri bireysel erişim izinlerine gereksinim duyan, onbinlerce 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.

MQTT istemcilerini, OAM kullanarak iki şekilde ortak kimlikler halinde birleştirebilirsiniz:

1. Her biri, denetimcinin WebSphere MQ Explorer kullanarak ayırdığı farklı bir kullanıcı kimliği ile birden çok telemetry kanalı tanımlayın. Farklı TCP/IP kapı numaralarını kullanarak bağlanan istemciler farklı telemetry kanallarıyla ilişkilendirilir ve farklı kimlikler atanır.
2. Tek bir telemetry 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, telemetry kanalının kimliği *mqttUser* olarak adlandırılır. İstemcilerin derlemleri farklı kimlikler kullanırsa, her istemci toplaması için bir tane olmak üzere birden çok *mqttUser* 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 denetimin ayrıntı düzeylerine göre farklılık gösterir.

- [“Erişim denetimi yok” sayfa 123](#)
- [“Kaba erişim denetimi” sayfa 123](#)
- [“Orta-parçalı erişim denetimi” sayfa 123](#)
- [“Parçalı erişim denetimi” sayfa 123](#)

The result of the models is to assign *mqttUsers* sets of permissions to publish and subscribe to WebSphere MQ, and receive publications from WebSphere MQ.

Erişim denetimi yok

MQTT istemcilerine WebSphere MQ yönetim yetkisi verilir ve herhangi bir nesne üzerinde herhangi bir işlem gerçekleştirilebilir.

Yordam

1. Tüm MQTT istemcilerinin kimliği olarak hareket etmek için *mqttUser* kullanıcı kimliği oluşturun.
2. *mqm* grubuna *mqttUser* ekleyin; bkz. Pencerelerüzerinde bir gruba kullanıcı eklemeye da Linuxüzerinde bir gruba kullanıcı ekleme

Kaba erişim denetimi

MQTT istemcilerinin yayınlama ve abone olma yetkisi vardır ve MQTT istemcilerine ileti göndermeleri gerekir. 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 bu 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 gruplar halinde yayınlamak ve farklı konu kümelerine 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. Konuları MQTT istemcilerine göndermek için *mqtt* ' i yetkilendirin.

```
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. WebSphere MQ uygulamaları, MQTT istemcileriyle aynı konu alanına yayınlanırken ve abone olursa, bu modeli kullanın. Gruplar, PublishX, SubscribeY ve mqtt olarak adlandırılır.

PublishX

PublishX gruplarının üyeleri *topicX* ' a yayınlanabilir.

SubscribeY

SubscribeY gruplarının üyeleri *topicY* ' e abone olabilir.

mqtt

mqtt grubunun üyeleri, bu yayınları MQTT istemcilerine gönderebilirler.

Yordam

1. Yayınlama/abone olma konu ağacında birden çok yönetici konu için ayrılmış birden çok grup, PublishX ve SubscribeY oluşturun.
2. Create a group *mqtt*.
3. Birden çok kullanıcı kimliği (*mqttUsers*) yaratın ve yetkilendirildikleri şeye bağlı olarak, kullanıcıları herhangi bir gruba ekleyin.
4. Farklı konulara farklı PublishX ve SubscribeX gruplarını yetkilendirin ve *mqtt* grubuna, MQTT istemcilerine ileti göndermesini yetkilendirin.

```
setmqaut -m qMgr -t topic -n topic1 -p PublishX -all +pub
setmqaut -m qMgr -t topic -n topic1 -p SubscribeX -all +pub +sub
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

Parola kullanarak MQTT istemcisi kimlik doğrulaması

İstemci parolasını kullanarak Kullanıcı1 adlı '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 Kullanıcı1 adlı kimliğini doğrulamak için JAAS 'ı kullanır. JAAS , MQTT istemcisi tarafından sağlanan Parola 'yı kullanır.

The WebSphere MQ administrator decides whether to authenticate the Kullanıcı1 Adı, or not to authenticate at all, by configuring the MQTT channel a client connects to. 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 WebSphere MQ administrator sets the MQTT channel identity to any name, such as MQTTClientUser, using WebSphere MQ Explorer.
2. WebSphere MQ yöneticisi, MQTTClient ' e herhangi bir konuya abone olma ve abone olma yetkisi verir:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

3. MQTT istemci uygulaması geliştiricisi, sunucuya bağlanmadan önce bir MqttConnectOptions nesnesi yaratır ve Kullanıcı1 adı ve Parola seçeneğini ayarlar.
4. The security developer creates a JAAS LoginModule to authenticate the Kullanıcı1 Adı with the Parola and includes it in the JAAS configuration file.
5. The WebSphere MQ administrator configures the MQTT channel to authenticate the UserName of the client using JAAS.

SSL kullanan MQTT istemcisi 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 sertifikayla sağlayarak, MQTT istemcisinin kimliğini IBM WebSphere MQ olarak doğrulayabilirsiniz. IBM WebSphere MQ Administrator, MQTT istemcilerini SSL 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 kuruyorsunuz. MQTT istemcilerini, VPN ağı üzerinden TCP/IP kullanarak telemetri kanallarına bağlayabilirsiniz.

Client kimlik doğrulaması, istemciye bir sır vermesinden kaynaklanıyor. 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.

-
-

SSL kullanan MQTT istemci kimlik doğrulaması için telemetri kanalı yapılandırması

IBM WebSphere MQ yöneticisi, sunucudaki telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasındaki bir TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. SSL kanalları, anahtar kütüklere geçiş tümcecisiyle korunan erişim ile yapılandırılır. Bir SSL kanalı geçiş tümcecik ya da anahtar dosyası olmadan tanımlandıysa, kanal SSL bağlantılarını kabul etmez.

Set the property, com.ibm.mq.MQTT.ClientAuth of an SSL 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 WebSphere 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 WebSphere 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 WebSphere MQ yöneticisi tarafından yaratılan bir kimdir.

İstemci kimlik doğrulaması için SSL şifreleme takımlarını da kullanabilirsiniz. Burada, şu anda desteklenmekte olan SSL şifreleme takımlarının alfabetik listesi gösterilmektedir:

- `SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA`
- `SSL_DH_anon_EXPORT_WITH_RC4_40_MD5`
- `SSL_DH_anon_WITH_3DES_EDE_CBC_SHA`
- `SSL_DH_anon_WITH_AES_128_CBC_SHA`
- `SSL_DH_anon_WITH_DES_CBC_SHA`
- `SSL_DH_anon_WITH_RC4_128_MD5`
- `SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA`
- `SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA`
- `SSL_DHE_DSS_WITH_AES_128_CBC_SHA`
- `SSL_DHE_DSS_WITH_DES_CBC_SHA`
- `SSL_DHE_DSS_WITH_RC4_128_SHA`
- `SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA`
- `SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA`
- `SSL_DHE_RSA_WITH_AES_128_CBC_SHA`
- `SSL_DHE_RSA_WITH_DES_CBC_SHA`
- `SSL_KRB5_EXPORT_WITH_DES_CBC_40_MD5`
- `SSL_KRB5_EXPORT_WITH_DES_CBC_40_SHA`
- `SSL_KRB5_EXPORT_WITH_RC4_40_MD5`
- `SSL_KRB5_EXPORT_WITH_RC4_40_SHA`
- `SSL_KRB5_WITH_3DES_EDE_CBC_MD5`
- `SSL_KRB5_WITH_3DES_EDE_CBC_SHA`
- `SSL_KRB5_WITH_DES_CBC_MD5`
- `SSL_KRB5_WITH_DES_CBC_SHA`

- SSL_KRB5_WITH_RC4_128_MD5
- SSL_KRB5_WITH_RC4_128_SHA
- SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
- SSL_RSA_EXPORT_WITH_RC4_40_MD5
- SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
- **V7.5.0.2** SSL_RSA_FIPS_WITH_AES_128_CBC_SHA256
- **V7.5.0.2** SSL_RSA_FIPS_WITH_AES_256_CBC_SHA256
- SSL_RSA_FIPS_WITH_DES_CBC_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_AES_128_CBC_SHA
- **V7.5.0.2** SSL_RSA_WITH_AES_128_CBC_SHA256
- **V7.5.0.2** SSL_RSA_WITH_AES_256_CBC_SHA256
- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_NULL_MD5
- SSL_RSA_WITH_NULL_SHA
- **V7.5.0.2** SSL_RSA_WITH_NULL_SHA256
- SSL_RSA_WITH_RC4_128_MD5
- SSL_RSA_WITH_RC4_128_SHA

V7.5.0.2 SHA-2 şifreleme takımlarını kullanmayı planlıyorsanız, bkz. [System requirements for using SHA-2 cipher suites with MQTT channels.](#)

İlgili kavramlar

“SSL kullanan kanal kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 128

IBM WebSphere MQ yöneticisi, sunucudaki telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasındaki bir TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. SSL kanalları, anahtar kütüklere geçiş tümcecisiyle korunan erişim ile yapılandırılır. Bir SSL kanalı geçiş tümcecik ya da anahtar dosyası olmadan tanımlandıysa, kanal SSL bağlantılarını kabul etmez.

[CipherSpecs ve CipherSuites](#)

İlgili başvurular

[KANAL TANIMLA \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

SSL kullanan 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\cacerts. 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 kanalını SSL kullanarak doğrulamak için sunucuyu ve istemciyi yapılandırın.

-
-

SSL kullanan kanal kimlik doğrulaması için telemetri kanalı yapılandırması

IBM WebSphere MQ yöneticisi, sunucudaki telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasındaki bir TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. SSL kanalları, anahtar kütüklere geçiş tümcecisiyle korunan erişim ile yapılandırılır. Bir SSL kanalı geçiş tümcecik ya da anahtar dosyası olmadan tanımlandıysa, kanal SSL 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. SSL kullanmak için WebSphere MQ gezginini kullanarak telemetri kanalını yapılandırın. Anahtar deposunun yolunu ve anahtar deposuna erişmek için geçiş tümcecisini belirtin. Kanal TCP/IP kapı numarasını ayarlamadıysanız, SSL telemetri kanalı kapı numarası varsayılan olarak 8883 değerine ayarlanır.

Kanal kimlik doğrulaması için SSL şifreleme takımlarını da kullanabilirsiniz. Burada, şu anda desteklenmekte olan SSL şifreleme takımlarının alfabetik listesi gösterilmektedir:

- SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
- SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
- SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
- SSL_DH_anon_WITH_AES_128_CBC_SHA
- SSL_DH_anon_WITH_DES_CBC_SHA
- SSL_DH_anon_WITH_RC4_128_MD5
- SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
- SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
- SSL_DHE_DSS_WITH_AES_128_CBC_SHA
- SSL_DHE_DSS_WITH_DES_CBC_SHA
- SSL_DHE_DSS_WITH_RC4_128_SHA
- SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
- SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_DHE_RSA_WITH_AES_128_CBC_SHA
- SSL_DHE_RSA_WITH_DES_CBC_SHA
- SSL_KRB5_EXPORT_WITH_DES_CBC_40_MD5
- SSL_KRB5_EXPORT_WITH_DES_CBC_40_SHA
- SSL_KRB5_EXPORT_WITH_RC4_40_MD5
- SSL_KRB5_EXPORT_WITH_RC4_40_SHA

- SSL_KRB5_WITH_3DES_EDE_CBC_MD5
- SSL_KRB5_WITH_3DES_EDE_CBC_SHA
- SSL_KRB5_WITH_DES_CBC_MD5
- SSL_KRB5_WITH_DES_CBC_SHA
- SSL_KRB5_WITH_RC4_128_MD5
- SSL_KRB5_WITH_RC4_128_SHA
- SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
- SSL_RSA_EXPORT_WITH_RC4_40_MD5
- SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
- **V7.5.0.2** SSL_RSA_FIPS_WITH_AES_128_CBC_SHA256
- **V7.5.0.2** SSL_RSA_FIPS_WITH_AES_256_CBC_SHA256
- SSL_RSA_FIPS_WITH_DES_CBC_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_AES_128_CBC_SHA
- **V7.5.0.2** SSL_RSA_WITH_AES_128_CBC_SHA256
- **V7.5.0.2** SSL_RSA_WITH_AES_256_CBC_SHA256
- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_NULL_MD5
- SSL_RSA_WITH_NULL_SHA
- **V7.5.0.2** SSL_RSA_WITH_NULL_SHA256
- SSL_RSA_WITH_RC4_128_MD5
- SSL_RSA_WITH_RC4_128_SHA

V7.5.0.2 SHA-2 şifreleme takımlarını kullanmayı planlıyorsanız, bkz. [System requirements for using SHA-2 cipher suites with MQTT channels.](#)

İlgili kavramlar

“SSL kullanan MQTT istemci kimlik doğrulaması için telemetri kanalı yapılandırması” sayfa 125 IBM WebSphere MQ yöneticisi, sunucudaki telemetri kanallarını yapılandırır. Her kanal, farklı bir kapı numarasındaki bir TCP/IP bağlantısını kabul edecek şekilde yapılandırılır. SSL kanalları, anahtar kütüklere geçiş tümcecisiyle korunan erişim ile yapılandırılır. Bir SSL kanalı geçiş tümcecik ya da anahtar dosyası olmadan tanımlandıysa, kanal SSL bağlantılarını kabul etmez.

[CipherSpecs ve CipherSuites](#)

İlgili başvurular

[KANAL TANIMLA \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

telemetri kanallarında yayın gizliliği

Telemetri kanallarında her iki yönde gönderilen MQTT yayınlarının gizliliği, bağlantı üzerinden iletimleri şifrelemek için SSL kullanılarak güvenli kılınmaktadır.

Telemetri kanallarına bağlanan MQTT istemcileri, kanalda iletilen yayınların gizliliğini simetrik anahtar şifrelemesi kullanarak güvenli kılmak için SSL ' 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 kurmuyorsunuz. MQTT istemcilerini, VPN ağı üzerinden TCP/IP kullanarak telemetri kanallarına bağlayabilirsiniz.

Kanalı şifreleyen ve sunucuyu doğrulayan tipik bir yapılandırma için [“SSL kullanan telemetri kanalı kimlik doğrulaması” sayfa 127'](#) a danışın.

SSL bağlantılarının, sunucunun kimlik doğrulaması olmadan şifrelenmesi, ortadaki saldırılara karşı olan bağlantıyı sağlar. 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 SSL 'yi destekleyen bir Diffie-Hellman anahtar değiş tokuş Cipherspec kullanarak sunucuyu doğrulamadan şifrelenmiş bir SSL bağlantısı oluşturabilirsiniz. İstemci ile sunucu arasında paylaşılan ve SSL iletimlerini şifrelemek için kullanılan ana güvenlik dizgisi, özel olarak imzalanmış bir sunucu sertifikası değiş tokuş olarak kurulur.

Anonim bağlantılar güvensiz olduğu için, çoğu SSL somutlaması anonim Cipherspec özelliğini kullanmak için varsayılan değer değildir. Bir telemetri kanalı tarafından SSL bağlantısı için bir istemci isteği kabul edilirse, kanala geçiş tümceciği tarafından korunan bir anahtar deposu bulunmalıdır. Varsayılan olarak, SSL uygulamaları anonim Cipherspec kullanmadığı için, anahtar deposunun istemcinin doğrulayabileceği özel olarak imzalanmış bir sertifika içermesi gerekir.

Anonim Cipherspec değerini kullanırsanız, sunucu anahtar deposu var olmalıdır, ancak özel olarak imzalanmış herhangi bir sertifika içermemelidir.

Şifrelenmiş bir bağlantı kurmanın başka bir yolu da, istemciye güven sağlayıcısının yerine kendi uygulamanızı koymanızdır. Güvenilir sağlayıcınız sunucu sertifikasının kimliğini doğrulamazdı, ancak bağlantı şifrelenir.

MQTT istemcilerinin ve telemetri kanallarının SSL yapılandırması

MQTT istemcileri ve WebSphere MQ Telemetry (MQXR) hizmeti, SSL kullanarak telemetri kanallarını bağlamak için Java Secure Socket Extension (JSSE) olanağını kullanır. Aygıtlar için IBM WebSphere MQ Telemetry cini SSL 'yi desteklemiyor.

Telemetri kanalını, MQTT istemcisini doğrulamak için SSL 'yi yapılandırın ve müşteriler ile telemetri kanalı arasındaki iletilerin aktarılıp şifrelenmesini şifreleyin.

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.

TCP/IP üzerinden SSL protokolünü kullanmak için bir Java MQTT istemcisi ile telemetri kanalı arasında bağlantı yapılandırabilirsiniz. Güvenli olan, SSL 'yi JSSE' yi kullanmak için nasıl yapılandırmanıza bağlıdır. En güvenli yapılandırmayla başlayarak, üç farklı güvenlik düzeyi yapılandırabilirsiniz:

1. Yalnızca güvenilen MQTT istemcilerinin bağlanmasına izin verir. Bir MQTT istemcisini yalnızca güvenilir bir telemetri kanalına bağlayın. İstemciyle kuyruk yöneticisi arasındaki iletileri şifrele; bkz. [“SSL kullanan MQTT istemcisi kimlik doğrulaması” sayfa 124](#)
2. Bir MQTT istemcisini yalnızca güvenilir bir telemetri kanalına bağlayın. İstemciyle kuyruk yöneticisi arasındaki iletileri şifrele; bkz. [“SSL kullanan telemetri kanalı kimlik doğrulaması” sayfa 127](#).
3. İstemciyle kuyruk yöneticisi arasındaki iletileri şifrele; bkz. [“telemetri kanallarında yayın gizliliği” sayfa 129](#).

JSSE yapılandırma parametreleri

Bir SSL bağlantısının konfigürasyonunun tanımlanını değiştirmek için JSSE parametrelerini değiştirin. JSSE yapılandırma parametreleri üç küme halinde düzenlenmiştir:

1. [IBM WebSphere MQ Telemetri kanalı](#)
2. [MQTT Java istemcisi](#)

3. JRE

IBM WebSphere MQ Explorer 'ı kullanarak telemetri kanalı parametrelerini yapılandırın. `MqttConnectionOptions.SSLProperties` öznitelide MQTT Java istemcisi deęiřtirgelerini ayarlayın. JRE güvenlik deęiřtirgelerini istemci ve sunucu üzerindeki JRE güvenlik dizinindeki dosyaları düzenleyerek deęiřtirin.

IBM WebSphere MQ Telemetry kanalı

WebSphere MQ Explorer 'ı kullanarak tüm telemetri kanalı SSL parametrelerini ayarlayın.

ChannelName

`ChannelName` , tüm kanalarda zorunlu bir parametredir.

Kanal adı, belirli bir kapı numarasıyla ilişkilendirilen kanalı tanımlar. MQTT istemcilerinin gruplarını yönetmenize yardımcı olacak ad kanalları.

PortNumber

`PortNumber` , tüm kanalarda isteęe baęlı bir parametredir. Varsayılan deęer olarak TCP kanalları için 1883 , SSL kanalları için de 8883 deęeri kullanılır.

Bu kanalla ilişkilendirilen TCP/IP kapı numarası. MQTT istemcileri, kanal için tanımlanan kapıyı belirterek bir kanala baęlanır. Kanalda SSL özellikleri varsa, istemci SSL protokolünü kullanarak baęlanmalıdır; örneęin:

```
MqttClient mqttClient = new MqttClient( "ssl://www.example.org:8884", "clientId");
mqttClient.connect();
```

KeyFileAd

`KeyFileAd` , SSL kanalları için gerekli bir deęiřtirgedir. TCP kanalları için atlanmalıdır.

`KeyFileAd1` , sağladığınız sayısal sertifikalar içeren Java anahtar deposunun yoludur. Sunucuda anahtar deposu tipi olarak JKS, JCEKS ya da PKCS12 ' yi kullanın.

Anahtar deposu tipini, aşağıdaki dosya uzantılarından birini kullanarak tanımlayın:

- .jks
- .jceks
- .p12
- .pkcs12

Diđer dosya uzantılarına sahip bir anahtar deposunun JKS anahtar deposu olduęu varsayılır.

Sunucudaki bir anahtar deposu tipini, istemcideki diđer anahtar deposu tipleriyle birleřtirebilirsiniz.

Sunucunun özel sertifikasını anahtar depoya yerleřtirin. Sertifika, sunucu sertifikası olarak bilinir. Sertifika, bir imzalama yetkilisi tarafından imzalanmış bir sertifika zincirinin ya da sertifika zincirinin bir parçası olabilir.

Bir sertifika zinciri kullanıyorsanız, ilişkili sertifikaları sunucu anahtar deposuna yerleřtirin.

Sunucu sertifikası ve sertifika zincirindeki tüm sertifikalar, sunucunun kimlięini doęrulamak üzere istemcilere gönderilir.

`ClientAuth` öęesini `Required` olarak ayarladıysanız, anahtar deposunun istemcinin kimlięini doęrulamak için gereken sertifikaları içermesi gerekir. İstemci kendinden onaylı bir sertifika ya da bir sertifika zinciri gönderir ve istemciyi, bu malzemenin anahtar depodaki bir sertifikaya ilişkin ilk doęrulamaları ile doęrulanır. Bir sertifika zinciri kullanarak, bir sertifika, farklı istemci sertifikalarıyla yayınlansa bile, birçok istemciyi doęrulayabilir.

PassPhrase

`PassPhrase` , SSL kanalları için gerekli bir parametredir. TCP kanalları için atlanmalıdır.

Anahtar deposunu korumak için geçiř tümcecisi kullanılır.

ClientAuth

ClientAuth , isteğe bağlı bir SSL 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 ayarladıysanız, istemci sunucuya SSL kullanarak bağlanmalıdır ve sunucunun kimliğini 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 SSL parametresidir. Varsayılan olarak, tüm etkin CipherSpecs ' ı deneyecek şekilde ayarlanır. TCP kanalları için atlanmalıdır.

If you want to use a particular CipherSpec, set CipherSuite to the name of the CipherSpec that must be used to establish the SSL 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.

JSE ' ye ek sağlayıcılar ekleyerek ek şifreleri kurun.

Federal Bilgi İşleme Standartları (FIPS)

FIPS, isteğe bağlı bir ayardır. Varsayılan olarak ayarlanmaz.

Kuyruk yöneticisinin özellikler panosunda ya da **runmqsc**komutunu kullanarak SSLFIPSayarlayın. SSLFIPS , yalnızca FIPS onaylı algoritmaların kullanılıp kullanılmayacağını belirtir.

İptal adlistesi

İptal ad listesi isteğe bağlı bir ayardır. Varsayılan olarak ayarlanmaz.

Kuyruk yöneticisinin özellikler panosunda ya da **runmqsc**komutunu kullanarak SSLCRLNL ' yi ayarlayın. SSLCRLNL , sertifika iptal konumlarını sağlamak için kullanılan kimlik doğrulama bilgileri nesnelerinin ad listesini belirtir.

SSL özelliklerini ayarlayan başka bir kuyruk yöneticisi değiştirgesi kullanılmadı.

MQTT Java istemcisi

MqttConnectionOptions.SSLPropertiesçinde Java istemcisi için SSL özellikleri 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 üzerindeki SSL davranışını etkileyen diğer yönleri JRE ' de yapılandırılır. Windows üzerindeki yapılandırma dosyaları *Java Installation Directory*\jre\lib\security içinde yer alıyor. IBM WebSphere MQ ile birlikte gönderilen JRE ' yi kullanıyorsanız, yol aşağıdaki çizelgede gösterildiği gibidir:

<i>Çizelge 8. JRE SSL yapılandırma dosyaları için altyapıya göre dosya yolları</i>	
Altyapı	filePath
Windows	<i>WMQ Installation Directory</i> \java\jre\lib\security
Diğer UNIX and Linux platformları	<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.

cacerts programını, IBM Key Management yardımcı programını çalıştıran WebSphere MQ komutunu `stmqikm` komutunu kullanarak açabilirsiniz. Open cacerts as a JKS file, using the password `changeit`. Dosyayı güvenli kılmak için parolayı değiştirin.

Güvenlik sınıflarının yapılandırılması

Ek güvenlik sağlayıcıları ve diğer varsayılan güvenlik özelliklerini kaydetmek için `java.security` dosyasını kullanın.

İzinler

Kaynaklara verilen izinleri değiştirmek için `java.policy` dosyasını kullanın. `javaws.policy`, `javaws.jar` için izin verir.

Şifreleme gücü

Bazı JRES 'ler güç şifrelemesi azaltılmış. Anahtarları anahtar depolarına aktaramıyorsanız, azaltılmış güvenlik düzeyi şifreleme nedeni olabilir. Either, try starting **ikeyman** using the **strong** command, or download strong, but limited jurisdiction files from [IBM geliştirici setleri](#), Güvenlik bilgileri.

Önemli: Ülkenizde, şifreleme yazılımların başka bir ülkeye alınması, 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

Örnekte, bir güven sağlayıcısının nasıl ekleneceği ve bu sağlayıcının MQTT istemci kodundan nasıl başvurulacağı gösterilmektedir. Bu örnek, istemci ya da sunucu için kimlik doğrulaması gerçekleştirmez. Sonuçtaki SSL bağlantısı, kimlik doğrulaması yapılmadan şifrelenir.

Şekil 25 sayfa 134 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 25. MQTT İstemcisi kod 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 26. `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 27. `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 28. AcceptAllX509TrustManager.java

Telemetri kanalı JAAS yapılandırması

İstemci tarafından gönderilen `Kullanıcı` adı'nın kimliğini doğrulamak için JAAS 'ı yapılandırın.

WebSphere MQ yöneticisi, JAAS ile istemci kimlik doğrulaması gerektiren MQTT kanallarının konfigürasyonunu tanımlar. 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 29 sayfa 136](#).

JAAS, dört standart oturum açma modülü sağlar. Standart NT ve UNIX Oturum Açma modülleri sınırlı değerlerdir.

JndiLoginBirimi

JNDI (Java Naming and Directory Interface; Java Adlandırma ve Dizin Arabirimi) 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. Örnek JAASLoginModule.java, WebSphere MQ Telemetry ile birlikte sağlanır. 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 WebSphere MQ dizinlerine yerleştirmeyin. Kendi dizinlerinizi oluşturun ve telemetri (MQXR) hizmeti için tüm sınıf yolunu tanımlayın.

service.env dosyasında sınıf yolunu ayarlayarak telemetri (MQXR) hizmeti tarafından kullanılan sınıf yolunu genişletebilirsiniz. CLASSPATH büyük harfle yazılmalı ve sınıf yolu deyimi yalnızca hazır bilgiler içerebilir. CLASSPATH değişkeninde değişkenleri kullanamazsınız; örneğin, CLASSPATH=%CLASSPATH% yanlış. Telemetri (MQXR) hizmeti kendi sınıf yolunu (classpath) ayarlar. service.env içinde tanımlanan CLASSPATH bu dosyaya eklenir.

Telemetri (MQXR) hizmeti, MQTT kanalına bağlı bir istemci için Kullanıcı adı ve Parola değerini döndüren iki geri arama olanağı sağlar. The Kullanıcı Adı and Parola are set in the MqttConnectOptions object. Kullanıcı Adı ve Parola erişimine nasıl erişileceğini gösteren bir örnek için bkz. [Şekil 30 sayfa 136](#).

Örnekler

Adiconfiguration olan 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 29. Örnek jaas.config dosyası

Bir MQTT istemcisi tarafından sağlanan Kullanıcı adı ve Parola 'yı almak için 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 30. Örnek JAASLoginModule.Login() yöntemi

Aygıtlar kavramları için IBM WebSphere MQ Telemetry cini

Aygıtlar için IBM WebSphere MQ Telemetry cini gelişmiş bir MQTT V3 istemci uygulamasıdır. İletileri diğer MQTT istemcilerinden saklamak ve iletmek için kullanın. Bu, IBM WebSphere MQ ile bir MQTT istemcisi gibi bağlanır, ancak diğer MQTT istemcilerini de bu istemciye bağlayabilirsiniz.

Yardımcı program bir yayınlama/abone olma aracıdır. MQTT V3 istemcileri, yayınlamak ve konuları yayınlamak için konu dizgileri ve abone olmak için konu süzgeçlerini kullanarak, konulara abone olur ve bu istemcilere abone olabilirler. The topic string is hierarchical, with topic levels divided by /. Konu süzgeçleri, tek düzeyli + genel arama karakterlerini ve çok düzeyli bir # genel arama karakterini, konu dizgisinin son bölümü olarak içerebilen konu dizgileridir.

Not: Yardımcı program için genel arama karakterleri, WebSphere Message Broker 'ın daha kısıtlayıcı kuralları (v6) izler. IBM WebSphere MQ farklı. Birden çok düzeyli birden çok genel arama karakteri destekler; genel arama karakterleri, sıradüzenin herhangi bir sayıdaki düzeyi için, konu dizgisinin herhangi bir yerinde durabilir.

Birden çok MQTT v3 istemcisi, dinleyici kapısını kullanarak yardımcı programa bağlanır. Varsayılan dinleyici kapısı değiştirilebilir. Birden çok dinleyici kapısı tanımlayabilir ve bunlara farklı ad alanları ayırabilirsiniz, bkz. [“Aygıtlar dinleyici kapıları için WebSphere MQ Telemetry cini” sayfa 144](#). Yardımcı programın kendisi bir MQTT v3 istemcisi olabilir. Yardımcı programı başka bir yardımcı programın dinleyici kapısına ya da bir WebSphere MQ Telemetry (MQXR) hizmetine bağlamak için bir yardımcı program köprüsü bağlantısı yapılandırın.

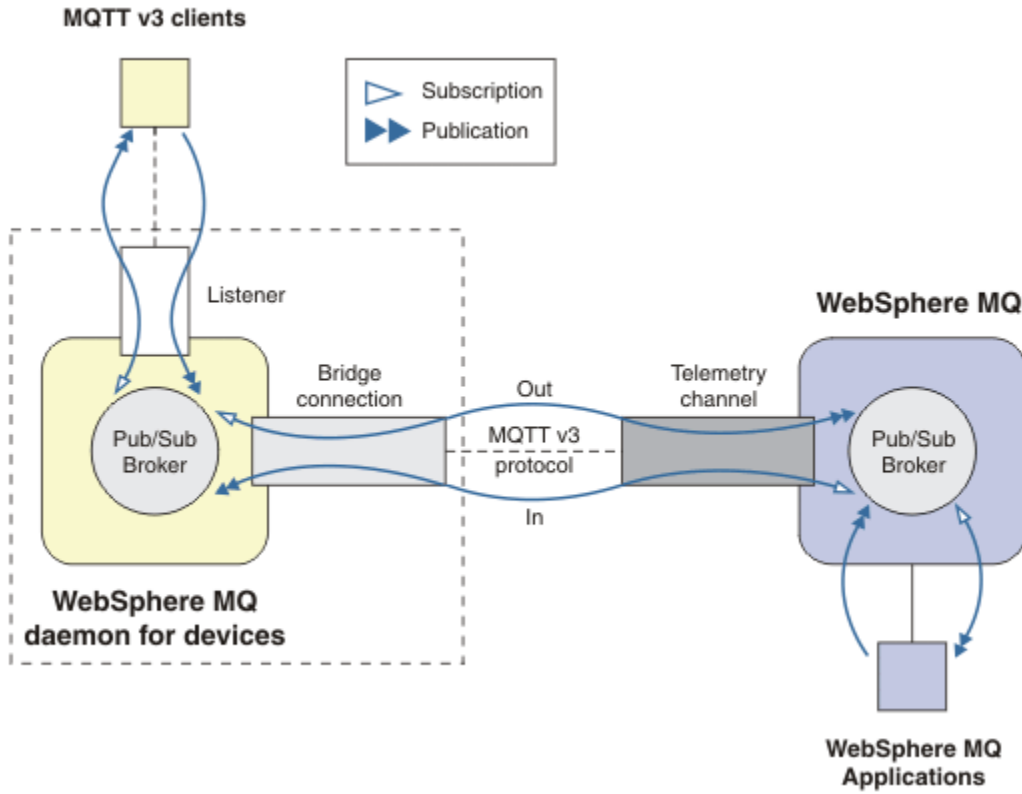
Aygıtlar için WebSphere MQ Telemetry cini için birden çok köprü yapılandırabilirsiniz. Yayınları değişik tokuş edebilen bir dizi yardımcı programı birbirine bağlamak için köprüleri kullanın.

Her bir köprü, yerel yardımcı programındaki konulara abone olabilir ve bu konulara abone olabilir. Ayrıca, başka bir yardımcı program, bir WebSphere MQ yayınlama/abone olma/abone olma aracısındaki ya da bağlı olduğu başka bir MQTT v3 aracısındaki konuları yayınlayabilir ve bunlara abone olabilir. Bir konu süzgecini kullanarak, yayınların bir araçından başka bir aracıya dağıtılmasını seçebilirsiniz. Yayınları her iki yöne de geçirebilirsiniz. You can propagate publications from the local daemon to each of its attached remote brokers, or from any of the attached brokers to the local daemon; see [“Aygıtlar köprüleri için IBM WebSphere MQ Telemetry cini” sayfa 137](#).

Aygıtlar köprüleri için IBM WebSphere MQ Telemetry cini

Bir IBM WebSphere MQ Telemetry cini for devices bridge, MQTT v3 iletişim kuralını kullanarak iki yayınlama/abone olma aracısını bağlar. Köprü, yayınları bir araçından diğerine, her iki yönde de yaylar. Bir uçta, aygıtlar köprü bağlantısı için bir WebSphere MQ Telemetry cini ve diğerinde bir kuyruk yöneticisi ya da başka bir yardımcı program olabilir. Bir kuyruk yöneticisi, telemetri kanalı kullanarak köprü bağlantısına bağlanır. Yardımcı program dinleyicisi kullanılarak köprü bağlantısına bir yardımcı program bağlanır.

IBM WebSphere MQ Telemetry cini aygıtları, diğer araçlara bir ya da daha fazla eşzamanlı bağlantıyı destekler. Yardımcı programdaki bağlantılar köprü adı verilir ve yardımcı program yapılandırma dosyasındaki bağlantı girişleriyle tanımlanır. IBM WebSphere MQ ile kurulan bağlantılar, aşağıdaki şekilde gösterildiği gibi IBM WebSphere MQ telemetri kanalları kullanılarak yapılır:



Şekil 31. IBM WebSphere MQ Telemetry daemon for devices ile IBM WebSphere MQ arasında bağlantı kuruluyor

Bir köprü, cini başka bir aracıya MQTT v3 istemcisi olarak bağlar. Köprü parametreleri, bir MQTT v3 istemcisinin özneliklerini ikizlemektedir.

Köprü bir bağlantıdan daha fazlasıdır. Bu, iki yayınlama/abone olma aracı arasında yer alan bir yayınlama ve abone olma aracı görevi görür. Yerel aracı, aygıtlar için IBM WebSphere MQ Telemetry yardımcı programsıdır ve uzak aracı, MQTT v3 iletişim kuralını destekleyen bir yayınlama/abone olma aracıdır. Genellikle uzak aracı başka bir yardımcı program ya da IBM WebSphere MQ' dir.

Köprü'nün işi, iki komisyoncu arasındaki yayınları yaymak. Köprü iki yönlü. Yayınları her iki yönde de yayıyor. Şekil 31 sayfa 138 , köprü'nün IBM WebSphere MQ Telemetry cinini aygıtlar için IBM WebSphere MQ' e nasıl bağlatacağı şekilde gösterir. "Köprüye ilişkin örnek konu ayarları" sayfa 139 , köprüyü yapılandırmak için konu değiştirgenin nasıl kullanılacağını göstermek için örnekleri kullanır.

Şekil 31 sayfa 138 içindeki In ve Out okları, köprü'nün iki yönlülüğü olduğunu belirtir. Okun bir ucunda abonelik yaratılır. Abonelik eşleşen yayınlar, okun ters ucundaki aracıya yayınlanır. Ok, yayınların akışına göre etiketlenir. Yayın akışı Giriş cinine ve yardımcı programdan Çıkış ' a (Windows) akışını sağlar. Etiketlerin önemi, komut sözdiziminde kullanılırlar. In (Giriş) ve Out (Çıkış) başlıklı konuda, yayınların gönderileceği yere değil, aboneliğin gönderileceği yere başvurmasını unutmayın.

Other clients, applications, or brokers might be connected either to IBM WebSphere MQ or to WebSphere MQ Telemetry daemon for devices. Bunlar, bağlı oldukları aracıdaki konuları yayınlayıp abone olarak yayınlarlar. Aracı IBM WebSphere MQ ise, konular kümelenmiş ya da dağıtılmış olabilir ve yerel kuyruk yöneticisinde belirtik olarak tanımlanmaz.

Köprülerin kullanımı

Yardımcı bilgi işlem bağlarını, köprü bağlantılarını ve dinleyicilerini kullanarak birbirine bağlayın. Köprü bağlantılarını ve telemetri kanallarını kullanarak yardımcı bilgi işlem ve kuyruk yöneticilerini bir

araya bağlar. Birden çok aracıyı bir araya bağladığınızda, döngüler yaratılabilir. Dikkatli olun: Yayınlar, durmaksızın araçların döngülerinin çevresinde dolaştırılabilir, algısız olarak dağıtılabilir.

IBM WebSphere MQ ' e köprülü yardımcı programların kullanılmasının bazı nedenleri şunlardır:

MQTT istemci bağlantılarının sayısını WebSphere MQolarak azaltın.

Yardımcı programların sıradüzenini kullanarak birçok istemciyi WebSphere MQ' ya bağlayabilirsiniz; tek bir kuyruk yöneticisinin aynı anda bağlanabildiği sayı sayısından daha fazla istemci kullanılabilir.

MQTT istemcileri ile WebSphere MQarasında iletileri saklama ve iletme

İstemciler kendi depolamalarına sahip değilse, istemciler ve IBM WebSphere MQarasındaki sürekli bağlantıların korunmasını önlemek için mağazana ve ileriye doğru ilerleyebilirsiniz. MQTT istemcisi ile WebSphere MQarasında birden çok bağlantı tipi kullanılabilir; bkz. [İzleme ve denetim için telemetri kavramları ve senaryoları](#).

MQTT istemcileri ile WebSphere MQarasında değiş tokuş edilen yayınlara süzgeç uygula

Yaygın olarak, yayınlar yerel olarak işlenen iletilere ve diğer uygulamaları içeren iletilere bölünmesini sağlar. Yerel yayınlar, algılayıcılar ve çalıştırıcılar arasında denetim akışlarını içerebilir ve uzak yayınlar okuma, durum ve yapılandırma komutlarına ilişkin istekleri içerir.

Yayınlara konu alanlarını değiştirme

Farklı dinleyici kapılarına bağlı istemcilerden gelen konu dizgilerini birbiriyle çakışmadan kaçının. Bu örnek, farklı binalardan gelen ölçüm okuma değerlerini etiketlemek için yardımcı programı kullanır; bkz. [Farklı istemci gruplarının konu alanlarını ayırma](#).

Köprüye ilişkin örnek konu ayarları

Uzak aracıya her şeyi yayınlama-varsayılan değerler kullanılıyor

Varsayılan yön out(çıkış) olarak adlandırılır ve köprü konuları uzak aracıya yayınlamaktadır. konu değiştirgesi, konu süzgeçlerini kullanarak hangi konuların dağıtılmasını denetler.

Köprü, yerel cini MQTT istemcilerine ya da diğer araçlara göre yayınlanan her şeye abone olmak için [Şekil 32 sayfa 139](#) içindeki konu değiştirgesini kullanır. Köprü, konuları köprünün bağlı olduğu uzak aracıya yayınlamaktadır.

```
connection Daemon1
topic #
```

Şekil 32. Her şeyi uzak aracıya yayınlama

Her şeyi uzak aracıya yayınlama-belirtik

Aşağıdaki kod parçasındaki konu ayarı, varsayılan değerlerin kullanılmasından sonra aynı sonucu verir. Tek fark, **direction** parametresinin açık olması olabilir. Yerel aracıya, yardımcı programa abone olmak ve uzak aracıya yayınlamak için out yönünü kullanın. Köprünün abone olduğu yerel yardımcı program üzerinde oluşturulan yayınlar, uzak aracıya yayınlamaktadır.

```
connection Daemon1
topic # out
```

Şekil 33. Her şeyi uzak aracıya yayınlama-belirtik

Her şeyi yerel aracıya yayınlama

dışarıyönünü kullanmak yerine, ters yönü (içinde) ayarlayabilirsiniz. Aşağıdaki kod parçası, köprünün, köprünün bağlı olduğu uzak aracıya yayınlanan her şeye abone olacak şekilde yapılandırılarak yapılandırılarak yapılandırılabilir. Köprü, konuları yerel aracıya, yardımcı programdan yayınlamaktadır.

```
connection Daemon1
topic # in
```

Şekil 34. Her şeyi yerel aracıya yayınla

Yerel aracıdaki dışa aktarma konusundan uzak aracıdaki içe aktarma konusuna ilişkin her şeyi yayınla

Use two additional topic parameters, **local_prefix** and **remote_prefix**, to modify the topic filter, # in the previous examples. Abonelikte kullanılan konu süzgecini değiştirmek için bir değiştirge kullanılır; diğer değiştirge ise, yayının yayınlandığı konuyu değiştirmek için kullanılır. Bu etki, diğer aracıdaki başka bir konu dizgisi olan bir aracıda kullanılan konu dizgisinin başlangıcının yerini alır.

Konunun yönlerine bağlı olarak, **local_prefix** ve **remote_prefix** 'nin anlamı geri çevrilmesine neden olur. Yön out ise, varsayılan değer olan **local_prefix**, konu aboneliğinin bir parçası olarak kullanılır ve **remote_prefix**, uzak yayındaki konu dizgisinin **local_prefix** bölümünün yerini alır. Yön in ise, **remote_prefix** uzak aboneliğin bir parçası olur ve **local_prefix**, konu dizgisinin **remote_prefix** kısmını değiştirir.

Bir konu dizgisinin ilk bölümü genellikle bir konu alanı tanımlama olarak düşünülmektedir. Bir konunun yayınlandığı konu alanını değiştirmek için ek parametreleri kullanın. Bu işlemi, konunun hedef aracıdaki başka bir konuyla çakışmasını önlemek ya da bir bağlama noktası konu dizesini kaldırmak için bunu yapabilirsiniz.

Örnek olarak, aşağıdaki kod parçasında, yardımcı programdaki `export/#` konu dizgisine ilişkin tüm yayınlar uzak aracıda `import/#` olarak yeniden yayınlanır.

```
topic # out export/ import/
```

Şekil 35. Yerel aracıdaki dışa aktarma konusundan uzak aracıdaki içe aktarma konusuna ilişkin her şeyi yayınla

Uzak aracıdaki dışa aktarma konusundan yerel aracıdaki içe aktarma konusuna ilişkin her şeyi yayınla

Aşağıdaki kod parçası yapılandırmayı tersine çevrilmiş olarak gösterir; köprü, uzak aracıdaki `export/#` konu dizgisiyle yayınlanan her şeye abone olur ve yerel aracıda bunu `import/#` olarak yayınlar.

```
connection Daemon1
topic # in import/ export/
```

Şekil 36. Uzak aracıdaki dışa aktarma konusundan yerel aracıdaki içe aktarma konusuna ilişkin her şeyi yayınla

Publish everything from the 1884/ mount point to the remote broker with the original topic strings

Aşağıdaki kod parçasında, köprü yerel yardımcı programdaki `1884/` sisteme bağlama noktasına bağlı istemciler tarafından yayınlanan her şeye abone olur. Köprü, bağlama noktasına yayınlanan her şeyi uzak aracıya yayınlar. The mount point string `1884/` is removed from the topics published to the remote broker. The *yerel_önek* is the same as the mount point string `1884/`, and the *remote_prefix* is a blank string.

```
listener 1884
mount_point 1884/
connection Daemon1
topic # out 1884/ ""
```

Şekil 37. 1884/ bağlama noktasından, özgün konu dizgileriyle uzak aracıya her şeyi yayınlayın.

Farklı istemcilere bağlı farklı istemcilerin konu alanlarını ayırma

Bir bina için ölçüm okuma değerlerini yayınlamak üzere elektrik enerjisi sayaçları için bir uygulama yazılmıştır. Okuma değerleri, MQTT istemcileri kullanılarak aynı binada barındırılan bir yardımcı programa yayınlanır. Yayınlar için seçilen konu `power`. Aynı uygulama, karmaşık bir alan içinde bir dizi binaya da dağıtılır. Site izleme ve veri depolama için, tüm binalardan gelen okuma değerleri köprü bağlantıları kullanılarak toplanır. Bağlantılar, oluşturma yardımcı programı merkezi bir yerdeki WebSphere MQ 'ya bağlar.

Her bir binadaki istemci uygulamaları aynı, ancak veriler oluşturularak farklılaştırılmalıdır. Her okumanın bir `power` konusu vardır ve bunu ayırt etmek için bina numarasının başına önek olarak eklenmelidir. The bridge from the first building in the complex uses the prefix `meters/building01/`, from building two the prefix is `meters/building02/`. Diğer binalardan gelen okumalar aynı modeli takip ediyor. WebSphere MQ , `meters/building01/power` gibi konuları içeren okumaları alır.

Örnek tutarlıdır; uygulamada, uygulamanın yayınlandığı konu alanının yapılandırılabilir olma olasılığı yüksektir.

Her bir yardımcı program için yapılandırma dosyası, aşağıdaki kod parçasındaki örüntüden sonra gelen bir konu deyimine sahiptir:

```
connection Daemon1
topic power out "" meters/building01/
```

Şekil 38. Farklı yardımcı dillere bağlı istemcilerin konu alanlarını birbirinden ayırın

Kullanılmayan `local_prefix` parametresi için yer tutucu olarak boş bir dizgi belirtin.

Aynı yardımcı programa bağlı istemcilerin konu alanlarını ayır

Tüm güç ölçümlerini bağlamak için tek bir yardımcı program kullanıldığını varsayın. Uygulamada farklı kapılara bağlanmak üzere yapılandırılabilirliğine göre, aşağıdaki kod parçasındaki gibi metreleri farklı binalardan farklı dinleyici kapılarına bağlayarak binaları ayırt edebilirsiniz. Yine, örnek, bağlama noktalarının nasıl kullanılabileceğini gösterir.

```
listener 1884
mount_point meters/building01/
listener 1885
mount_point meters/building02/
connection Daemon1
topic meters/+/power out
```

Şekil 39. Aynı yardımcı programa bağlı istemcilerin konu alanlarını ayır

Her iki yönde akan yayınlar için farklı konuları yeniden eşleyin

In the configuration in the following code fragment, the bridge subscribes to the single topic `b` at the remote broker and forwards publications about `b` to the local daemon, changing the topic to `a`. The bridge also subscribes to the single topic `x` at the local broker and forwards publications about `x` to the remote broker, changing the topic to `y`.

```
connection Daemon1
topic "" in a b
topic "" out x y
```

Şekil 40. Her iki yönde akan yayınlar için farklı konuları yeniden eşleyin

Bu örneğe ilişkin önemli bir nokta, her iki aracıda da farklı konuların abone olunması ve yayınlanma noktalarının yayınlanması ile ilgilidir. Her iki aracıdaki konu alanları da dağılır.

Her iki yönde akan yayınlar için de aynı konuları yeniden eşleyin (döngüleme)

Önceki örnekten farklı olarak, Şekil 41 sayfa 142 içindeki yapılandırma, genel olarak bir döngüdeki sonuçlarla sonuçlanır. `topic "" in a b` adlı konu deyiminde, köprü b 'a uzaktan abone olur ve yerel olarak a 'a yayınlar. In the other topic statement, the bridge subscribes to a locally, and publishes to b remotely. Aynı yapılandırma, Şekil 42 sayfa 142 içinde gösterildiği gibi yazılabilir.

The general result is that if a client publishes to b remotely, the publication is transferred to the local daemon as a publication on topic a. However, on being published by the bridge to the local daemon on the topic a, the publication matches the subscription made by the bridge to local topic a. Abonelik `topic "" out a b`. As a result, the publication is transferred back to the remote broker as a publication on topic b. The bridge is now subscribed to the remote topic b, and the cycle begins again.

Bazı araçlar, döngülerin gerçekleşmesini önlemek için döngü algılamasını gerçekleştirir. Ancak, farklı tipte araçlar birleştiğinde döngü algılama mekanizmasının çalışması gerekir. Loop detection does not work if WebSphere MQ is bridged to the WebSphere MQ Telemetry daemon for devices. Aygıtlar için iki IBM WebSphere MQ Telemetry yardımcı programı birbirine sıkışmışsa, bu iş çalışır. Varsayılan döngü algılaması açık olarak açıktır; bkz. [try_private](#).

```
connection Daemon1
topic "" in a b
topic "" out a b
```

Şekil 41. !Her iki yönde akan yayınlar için aynı konuları yeniden eşleyin

```
connection Daemon1
topic "" both a b
```

Şekil 42. !`both` kullanarak, her iki yönde akan yayınlarla aynı konuları yeniden eşleyin.

Şekil 40 sayfa 142 içindeki yapılandırma, Şekil 41 sayfa 142 ile aynıdır.

IBM WebSphere MQ Telemetry daemon for devices köprüsü bağlantılarının kullanılabilirliği

Var olan ilk uzak aracıya bağlanmak için birden çok IBM WebSphere MQ Telemetry daemon for devices köprüsü bağlantı adresi yapılandırın. Aracı çok eşgörsümlü bir kuyruk yöneticisiyse, her iki TCP/IP adresini de belirtin. Kullanılabilir olduğunda birincil sunucuya bağlanmak ya da yeniden bağlanmak için bir birincil bağlantı yapılandırın.

Bağlantı köprüsü parametresi (adresler), TCP/IP yuva adreslerinden oluşan bir listedir. Köprü, başarılı bir bağlantı kuruncaya kadar sırayla her adrese bağlanmayı dener. `yuvarlak _robin` ve `start_type` bağlantı parametreleri, başarılı bir bağlantı yapıldıktan sonra adreslerin nasıl kullanıldığını denetler.

`start_type` , `auto`(otomatik), `manual`(otomatik) ya da `lazy`(tembel) ise, bağlantı başarısız olursa, köprü yeniden bağlanmayı dener. Her bir adresi sırayla kullanır, her bağlantı girişimi arasında yaklaşık 20 saniyelik bir gecikme olur. `start_type` , `once`(kez) ise, bağlantı başarısız olursa, köprü otomatik olarak yeniden bağlanmayı denemez.

round_robin trueise, köprü bağlantısı, listedeki ilk adresten başlar ve listedeki her adresi sırayla dener. Liste çok tükendiğinde, ilk adresten başlar. Listede yalnızca bir adres varsa, her 20 saniyede bir yeniden dener.

round_robin değeri falseise, listedeki birincil sunucu adı verilen ilk adres tercihtir. İlk sunucuya bağlanma girişimi başarısız olursa, köprü arka plandaki birincil sunucuya yeniden bağlanmayı denemeye devam eder. Aynı zamanda köprü, listedeki diğer adresleri kullanarak bağlanmayı dener. Arka plan birincil sunucuya bağlanmayı denediğinde, köprü geçerli bağlantıdan bağlantıyı keser ve birincil sunucu bağlantısına geçiş yapar.

Bir bağlantının kendi isteğiyle bağlantısı kesilirse (örneğin, bir **connection_stop** komutu yayınlayarak), bağlantı yeniden başlatılırsa, aynı adresi kullanmaya devam eder. Bağlanmadaki bir hata nedeniyle bağlantı kesilirse ya da uzak aracıya bağlantıyı bıraktıysa, köprü 20 saniye bekler. Daha sonra listede yalnızca bir adres varsa, listedeki bir sonraki adrese ya da aynı adrese bağlanmaya çalışır.

Çok eşgörünümlü bir kuyruk yöneticisiyle bağlantı kurulması

Çok eşgörünümlü kuyruk yöneticisi yapılandırılmasında, kuyruk yöneticisi farklı IP adresleriyle iki farklı sunucuda çalışır. Tipik olarak telemetri kanalları, belirli bir IP adresi olmadan yapılandırılır. Bunlar yalnızca bir kapı numarasıyla yapılandırılırlar. telemetri kanalı başlatıldığında, varsayılan olarak yerel sunucudaki ilk kullanılabilir ağ adresini seçer.

Kuyruk yöneticisi tarafından kullanılan iki IP adresi ile köprü bağlantısının adresler parametresini yapılandırın. round_robin ögesini true olarak ayarlayın.

Etkin kuyruk yöneticisi yönetim ortamı başarısız olursa, kuyruk yöneticisi yedek yönetim ortamına geçiş yapar. Yardımcı program, etkin eşgörünümlü bağlantının bozuk olduğunu saptar ve yedek yönetim ortamına yeniden bağlanmayı dener. Köprü bağlantısı için yapılandırılan adresler listesinde yer alan diğer IP adresini kullanır.

Köprünün bağlandığı kuyruk yöneticisi hala aynı kuyruk yöneticisindedir. Kuyruk yöneticisi kendi durumunu kurtarır. cleansession false olarak ayarlanırsa, köprü bağlantı oturumu, hata durumunda yedek denetleyiciye geçiş işleminden önce aynı duruma geri yüklenir. Bir gecikmeden sonra bağlantı devam eder. Messages with "en az bir kez" or "en çok bir kez" quality of service are not lost, and subscriptions continue to work.

Yeniden bağlanma süresi, yedek yönetim ortamı başlatıldığında yeniden başlatılan kanal sayısına ve istemcilerin sayısına ve ne kadar ileti ışığının yandığını bağlıdır. Köprü bağlantısı, bağlantı yeniden oluşturulmadan önce her iki IP adresine yeniden bağlanmayı deneyebilir.

Belirli bir IP adresine sahip çok eşgörünümlü bir kuyruk yöneticisi telemetri kanalı yapılandırmayın. IP adresi yalnızca tek bir sunucu üzerinde geçerlidir.

IP adresini yöneten alternatif bir yüksek kullanılabilirlik çözümü kullanıyorsanız, bir telemetri kanalını belirli bir IP adresiyle yapılandırmak için doğru olabilir.

cleansession

Köprü bağlantısı, MQTT v3 istemci oturumdur. Bir bağlantının yeni bir oturum mı başlatılacağını, yoksa varolan bir oturumu mı geri yükleyeceğini denetleyebilirsiniz. Var olan bir oturumu geri yüklerse, köprü bağlantısı abonelikleri korur ve önceki oturumdan yayınları alıkoyar.

adresler birden çok IP adresi listeleniyorsa ve IP adresleri farklı kuyruk yöneticileri tarafından barındırılan telemetri kanallarına ya da farklı telemetri yardımcı kanallarına bağlanıyorsa, temizoturum ' yi false olarak ayarlamayın. Oturum durumu, kuyruk yöneticileri ya da damasonlar arasında aktarılmaz. Farklı bir kuyruk yöneticisinde ya da yardımcı programında varolan bir oturumu yeniden başlatmaya çalışmak, yeni bir oturum başlatılmakta olan bir oturumla sonuçlanır. Belirsiz iletiler kaybedilir ve abonelikler beklendiği gibi davranılamayabilir.

bildirimler

Bir uygulama, bildirimler kullanılarak köprü bağlantısının çalışıp çalışmadığına ilişkin izi tutabilir. Bildirim, 1değerine, bağlı ya da 0değerine sahip, bağlantısı kesik olan bir yayındır. Bu, `notification_topic` parametresi tarafından tanımlanan `topicString` olarak yayınlanır. Varsayılan değer olan `topicString`, '\$SYS/broker/connection/clientIdentifier/state' dir. Varsayılan `topicString`, '\$SYS' önekini içerir. `Subscribe to topics beginning with $SYS` by defining a topic filter beginning with \$SYS. #konu süzgeci, her şeye abone olabilir, yardımcı program üzerinde \$SYS ile başlayan konulara abone olmaz. Uygulama konusu alanından farklı bir özel sistem konu alanı tanımlanırken '\$SYS ' i düşünün.

Bildirimler , bir köprü bağlandığında ya da bağlantısı kesildiğinde MQTT istemcilerine bildirimde bulunmak için IBM WebSphere MQ Telemetry daemon for devices ' i etkinleştirin.

Keepalive_interval

Keepalive_interval köprü bağlantı parametresi, uzak sunucuya TCP/IP ping komutu gönderen köprü arasındaki aralığı belirler. Varsayılan aralık 60 saniyedir. Ping, TCP/IP oturumunun uzak sunucu tarafından ya da bir güvenlik duvarı tarafından kapatılmasını önler, bu da bağlantıda bir etkinlik dışı durum olduğunu saptar.

clientid

Bir köprü bağlantısı, bir MQTT v3 istemci oturumdur ve clientidköprü bağlantı parametresi tarafından ayarlanan bir clientIdentifier değerine sahiptir. If you intend reconnections to resume a previous session by setting the temizoturum parameter to yanlış, the clientIdentifier used in each session must be the same. Varsayılan değer olan clientid , `hostname.connectionName` değeridir. Bu değer aynı kalır.

Aygıtlara ilişkin WebSphere MQ Telemetry cininin kurulması, doğrulanması, yapılandırılması ve denetlenmesi

Yardımcı programın kurulumu, yapılışı ve denetimi dosya tabanlıdır.

Yardımcı programı, Software Development Kit 'i (Software Development Kit), yardımcı programını çalıştıracak aygıtı kopyalayarak kurun.

Örnek olarak, MQTT istemci yardımcı programını çalıştırın ve yayınlama/abone olma aracı olarak aygıtlara ilişkin WebSphere MQ Teletext cinine bağlanın; bkz. [Belirli bir MQTT v3 istemcisine ileti yayınlayın](#) .

Bir yapılandırma dosyası oluşturarak cini yapılandırın; bkz. [Aygıtlar yapılandırma dosyası için WebSphere MQ Telemetry cini](#).

Control a running daemon by creating commands in the file, `amqtd.d.upd`. Yardımcı programın dosyayı okuduğu her 5 saniyede bir, komutları çalıştırır ve dosyayı siler; bkz. [Aygıtlar komut dosyası için WebSphere MQ Telemetry cini](#).

Aygıtlar dinleyici kapıları için WebSphere MQ Telemetry cini

Dinleyici kapıları kullanan aygıtlar için MQTT V3 istemcilerini WebSphere MQ Telemetry cinine bağlayın. Dinleyici kapısını bir bağlama noktası ve bağlantı sayısı üst sınırı ile niteleyebilirsiniz.

Bir dinleyici kapısı, bu kapiya bağlanan bir istemcinin MQTT istemcisi `connect(serveURI)` yönteminde belirtilen kapı numarasına karşılık gelmelidir. Varsayılan değer olarak hem istemci, hem de yardımcı program için '1883' a ayarlanır.

You can change the default port for the daemon by setting the global definition `kapı` in the daemon configuration file. Yardımcı program yapılandırma dosyasına bir `dinleyici` tanımlaması ekleyerek belirli kapıları ayarlayabilirsiniz.

Varsayılan kapı dışında her bir dinleyici kapısı için, istemcileri yalıtım için bir bağlama noktası belirtebilirsiniz. Bağlama noktası bulunan bir kapiya bağlı istemciler diğer istemcilerden yalıtılır; bkz. ["Aygıtlara bağlama noktaları için WebSphere MQ Telemetry cini"](#) sayfa 145.

Herhangi bir kapiya bağlanabilecek istemcilerin sayısını sınırlayabilirsiniz. Set the global definition `bağlantı_sayısı` to limit connections to the default port, or qualify each listener port with `bağlantı_sayısı`.

Örnek

An example of a configuration file that changes the default port from 1883 to 1880, and limits connections to port 1880 to 10000. 1884 bağlantı noktasına yönelik bağlantılar 1000 ile sınırlıdır. 1884 kapisına bağlanan istemciler, diğer kapılara bağlı istemcilerden yalıtılır.

```
port 1880
max_connections 10000
listener 1884
mount_point 1884/
max_connections 1000
```

Aygıtlara bağlama noktaları için WebSphere MQ Telemetry cini

Bir bağlama noktasını, aygıtlar için bir WebSphere MQ Telemetry cinine bağlanmak için MQTT istemcileri tarafından kullanılan bir dinleyici kapisıyla ilişkilendirebilirsiniz. Bir sisteme bağlama noktası, farklı bir dinleyici kapisına bağlı MQTT istemcilerinden bir dinleyici kapisını kullanarak, MQTT istemcilerinin deęiş tokuş edilen yayınlarını ve abonelikleri yalıtır.

Bir bağlama noktasıyla bir dinleyici kapisına bağlanan istemciler, diğer dinleyici kapislarına bağlı istemcilerle konuları doğrudan deęiştiremez. Bir bağlama noktası olmayan bir dinleyici kapisına bağlanan istemciler, herhangi bir istemcinin konularına yayınlayabilir ya da bu konuları abone olabilirler. Müşteriler, bir bağlama noktası aracılığıyla bağlı olup olmadıkları konusunda bilgi sahibi deęildir; müşteriler tarafından yaratılan konu dizgileriyle hiçbir fark yaratmaz.

Bağlama noktası, yayın ve aboneliklerin konu dizgisine önek olarak eklenen bir metin dizesidir. Dinleyici kapisına bir bağlama noktası olan istemciler tarafından yaratılan tüm konu dizgilerinin başına önek olarak eklenir. Metin dizesi, dinleyici kapisına bağlı istemcilere gönderilen tüm konu dizgilerinden kaldırılır.

Bir dinleyici kapisında bağlama noktası yoksa, kapiya bağlı istemciler tarafından oluşturulan ve alınan yayınların ve aboneliklerin konu dizgileri deęiştirilmez.

Sonda /olan bağlama noktası dizgileri oluşturun. Bu şekilde bağlama noktası, bağlama noktası için konu ağacının üst konudur.

Örnek

Yapılandırma dosyası, aşağıdaki dinleyici kapislarını içerir:

```
listener 1883
mount_point 1883/
listener 1884 127.0.0.1
mount_point 1884/
listener 1885
```

1883 kapisına bağlı bir istemci, MyTopic için bir abonelik oluşturur. Yardımcı program abonelięi 1883/MyTopic olarak kaydettirir. Another client attached to port 1883 publishes a message on the topic, MyTopic. Yardımcı program, konu dizesini 1883/MyTopic olarak deęiştirir ve eşleşen abonelikleri arar. The subscriber on port 1883 receives the publication with the original topic string MyTopic. Yardımcı program, bağlama noktası önekini konu dizgisinden kaldırdı.

Another client, attached to port 1884, also publishes on the topic MyTopic. Bu kez, yardımcı programı konuyu 1884/MyTopic olarak kaydettirir. Farklı bağlama noktası, farklı bir konu dizgisine sahip bir abonelikte sonuçlandıęından, 1883 numaralı kapidaki abone yayını almaz.

A client, attached to port 1885, publishes on the topic, 1883/MyTopic. Yardımcı program konu dizgisini deęiştirmiyor. The subscriber on port 1883 receives the publication to MyTopic.

Hizmet kalitesi, dayanıklı abonelikler ve alıkonan yayınlar için WebSphere MQ Telemetry cini

Hizmet kalitesi ayarları, yalnızca çalışan bir yardımcı program için geçerlidir. Bir yardımcı program durdurursa, denetimli bir şekilde ya da bir hata nedeniyle beliren iletilerin durumu kaybedilir. Bir iletinin teslim edilmesi en az bir kez ya da en azından bir kez durdurulacaksa garanti edilemez. Aygıtlar için WebSphere MQ Telemetry cini sınırlı kalıcılığı destekler. Yardımcı program sona erdirildiğinde alıkonan yayınları ve abonelikleri saklamak için **retained_persistence** yapılandırma seçeneğini ayarlayın.

Unlike WebSphere MQ, the WebSphere MQ Telemetry daemon for devices does not journal persistent data. Oturum durumu, ileti durumu ve alıkonan yayınlar iletsel olarak kaydedilmez. Varsayılan olarak, yardımcı program durduğunda tüm verileri atar. Abonelikleri düzenli olarak kontrol etmek ve yayınları alıkonmak için bir seçenek ayarlayabilirsiniz. Yardımcı program durduğunda ileti durumu her zaman kaybedilir. Alıkonmayan tüm yayınlar kaybedilir.

Alıkonan yayınları belirli aralıklarla bir dosyaya kaydetmek için yardımcı program yapılandırma seçeneğini `Retained_persistence` ögesini `true` olarak ayarlayın. Yardımcı program yeniden başlatıldığında, en son otomatik olarak kaydedilen yayınların geri alınması gerekir. Varsayılan olarak, istemciler tarafından yaratılan alıkonan iletiler, yardımcı program yeniden başlatıldığında yeniden yürürlüğe alınmaz.

Kalıcı bir oturumda oluşturulan abonelikleri bir dosyaya düzenli olarak kaydetmek için, yardımcı program yapılandırma seçeneğini `Retained_persistence` `true` olarak ayarlayın. `Retained_persistence` değeri `true` olarak ayarlanırsa, istemcilerin `CleanSession` ile `false`, bir "kalıcı oturum" olarak ayarlanmış bir oturumda oluşturulacağı abonelikler geri yüklenir. Yardımcı program, yayınları yeniden başlattığında, bu abonelikleri yeniden saklar. The client receives the publications when it restarts with `CleanSession` to `false`. By default, client session state is not saved when a daemon stops, and so subscriptions are not restored, even if the client sets `CleanSession` to `false`.

`Retained_persistence`, otomatik kaydetme mekanizmasıdır. En son tutulan yayınlar ya da abonelikler saklanmayabilir. Tutulan yayınların ve aboneliklerin hangi sıklıkta kaydedileceğini değiştirebilirsiniz. `autosave_on_changes` yapılandırma seçeneklerini ve `autosave_interval` yapılandırma seçeneklerini kullanarak, kaydetme arasındaki aralığı ya da kaydetme arasındaki değişiklik sayısını ayarlayın.

Kalıcılık ayarına ilişkin örnek yapılandırma

```
# Sample configuration
# Daemon listens on port 1882 with persistence in /tmp
# Autosave every minute
port 1882
persistence_location /tmp/
retained_persistence true
autosave_on_changes false
autosave_interval 60
```

Aygıt güvenliği için WebSphere MQ Telemetry cini

Aygıtlar için WebSphere MQ Telemetry cini, buna bağlanan istemcilerin kimliklerini doğrulayabilir, diğer araçlara bağlanmak için kimlik bilgilerini kullanabilir ve konulara erişimi denetleyebilir. Yardımcı programın sağladığı güvenlik, SSL desteği sağlamayan WebSphere MQ Telemetry C istemcisi kullanılarak oluşturularak sınırlandırılır. Sonuç olarak, yardımcı programa bağlantılar şifrelenmez ve sertifikalar kullanılarak kimliği doğrulanamaz.

Varsayılan değer olarak, hiçbir güvenlik açık değildir.

İstemcilerin kimlik doğrulaması

MQTT istemcileri, `MqttConnectOptions.setUsername` ve `MqttConnectOptions.setPassword` yöntemlerini kullanarak bir kullanıcı adı ve parola ayarlayabilirler.

Parola dosyasındaki girişlere karşı istemci tarafından sağlanan kullanıcı adını ve parolayı denetleyerek yardımcı programa bağlanan bir istemciyi doğrulayın. Kimlik doğrulamasını etkinleştirmek için, bir

parola dosyası oluşturun ve yardımcı program yapılandırma dosyasında `password_file` parametresini ayarlayın; bkz. [password_file](#).

İstemci adı ya da parola olmadan bağlanan istemcilerin kimlik doğrulaması yapan bir yardımcı programa bağlanmasına izin vermek için, yardımcı program yapılandırma dosyasında `allow_anonymous` deęiřtirgesini ayarlayın; bkz. [allow_anonymous](#). Bir istemci kullanıcı adı ya da parola saęlıyorsa, `password_file` parametresi ayarlandıysa, bu kullanıcı her zaman parola dosyası ile karşılaştırılır.

Yardımcı program yapılandırma dosyasındaki `clientid_prefixes` parametresini, belirli istemcilerle bağlantıları sınırlandırmak için ayarlayın. İstemcilerin `client_prefixes` deęiřtirgesinde listelenen örneklerden biriyle başlayan `clientIdentifiers` (`clientIdentifiersclientIdentifiers`)haveolmalıdır; bkz. [clientid_prefixes](#).

Köprü bağlantısı güvenlięi

Aygıtlar köprü bağlantısı için her WebSphere MQ Telemetry cini bir MQTT V3 istemcidir. Her bir köprü bağlantısı için kullanıcı adı ve parolayı, yardımcı program yapılandırma dosyasında bir köprü bağlantı parametresi olarak ayarlayabilir; bkz. [kullanıcı adı](#) ve [parola](#). Daha sonra bir köprü kendisini bir aracıya doğrulayabilir.

Konulara ilişkin erişim denetimi

İstemcilerin kimlięi doğrulanırsa, yardımcı program her kullanıcıya ilişkin konulara denetim erişimi de saęlayabilir. Yardımcı program, bir istemcinin erişim denetimi dosyasındaki bir erişim konu dizisiyle yayınlanmasını ya da abone olduęu konuyu eşleřtirmeye dayalı olarak erişim denetimi verir; bkz. [acl_file](#).

Eriřim denetimi listesinin iki bölümü vardır. İlk parça, anonim istemciler de dahil olmak üzere tüm istemcilere erişimi denetler. İkinci bölüm, parola dosyasındaki herhangi bir kullanıcı için bir bölüm içerir. Her kullanıcı için belirli erişim denetimini listeler.

Örnek

Güvenlik deęiřtirgeleri ařaęıdaki örnekte gösterilmiřtir.

```
acl_file c:\WMQTDaemon\config\acl.txt
password_file c:\WMQTDaemon\config\passwords.txt
allow_anonymous true
connection Daemon1
username daemon1
password daemonpassword
```

řekil 43. Yardımcı program yapılandırma dosyası

```
Fred:Fredpassword
Barney:Barneypassword
```

řekil 44. Parola dosyası, passwords.txt

```
topic home/public/#
topic read meters/#
user Fred
topic write meters/fred
topic home/fred/#
user Barney
topic write meters/barney
topic home/barney/#
```

řekil 45. Eriřim denetimi dosyası, acl.txt

Çoklu yayını yönetme

Çok hedefli iletilerin boyutunu küçültür ve veri dönüştürmeyi etkinleştirme gibi WebSphere MQ Multicast yönetimi görevleri hakkında bilgi edinmek için bu bilgileri kullanın.

Çok noktaya yayın ile çalışmaya başlama

WebSphere MQ Multicast konuları ve iletişim bilgileri nesneleriyle çalışmaya başlamak için bu bilgileri kullanın.

Bu görev hakkında

WebSphere MQ Multicast Messaging, 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 iletiminde yayınlanan bilgilerin konusu ve bir konu TOPIC nesnesi yaratılarak tanımlanmıştır. KONU NESNELERİNİN, çoklu yayınlama kullanılıp 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 verilir verilmeyeceğini belirtir.

Çok noktaya gönderim için KONU nesnesi tanımlamak için aşağıdaki komut satırı örneğini kullanın:

```
DEFINE TOPIC(ALLSPORTS) TOPICSTR('Sports') COMMINFO(MC1) MCAST(ENABLED)
```

ALLSPOR adı verilen yeni bir KONU nesnesi yaratılır. It has a topic string *Spor*, its related communication information object is called *MC1* (which is the name you specified when defining a COMMINFO object in the previous example), and multicast is enabled.

Çok noktaya yayın yayınlama/abone olma testi

TOPIC ve COMMINFO nesneleri yaratıldıktan sonra, bunlar amqspubc örneği ve amqssubc örneği kullanılarak sınanabilir. Bu örneklerle ilgili daha fazla bilgi için bkz. [Yayınlama/Abone Olma örnek programları](#).

1. İki komut satırı penceresi açın; ilk komut satırı, amqspubc yayınlama örneği içindir ve ikinci komut satırı amqssubc abone olma örneği içindir.
2. Komut satırı 1 'de aşağıdaki komutu girin:

```
amqspubc Sports QM1
```

Burada *Spor* , daha önceki bir örnekte tanımlanan KONU nesnesinin konu dizgisi ve *QM1* , kuyruk yöneticisinin adıdır.

3. Komut satırı 2 'de aşağıdaki komutu girin:

Burada *Spor* ve *QM1* , “2” sayfa 148adımında 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 WebSphere MQ Multicast konu topolojisi

IBM WebSphere MQ Multicast konu topolojisini anlamak için bu örneği kullanın.

IBM WebSphere MQ Multicast desteği, her bir alt ağacın, toplam sıradüzen içinde kendi çoklu yayın grubu ve veri akışı 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 adresin listesi için sistem denetimcinize başvurun ya da daha fazla bilgi için [IPv4 Multicast Address Space Registry \(IPv4-Çoklu Yayın Adresi Alanı Kaydı\)](#) başlıklı konuya bakın. Yerel olarak kapsamlı çok hedefli adresi 239.0.0.0 - 239.255.255.255 aralığında kullanmanız önerilir.

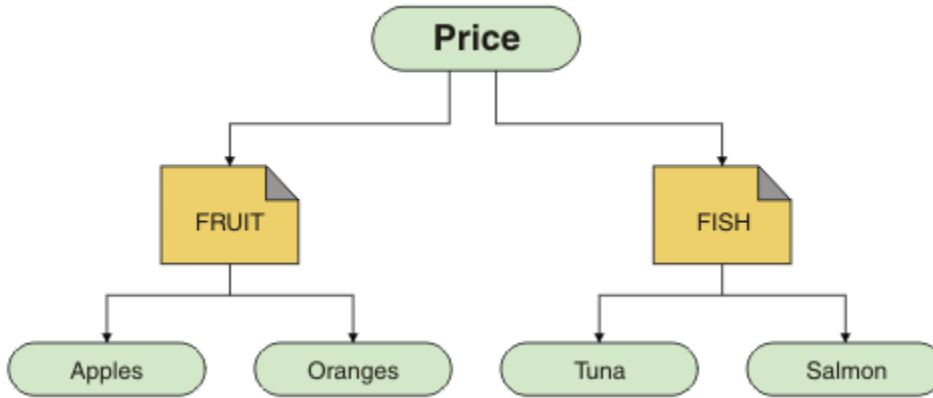
Aşağıdaki şemada, olası iki çok noktaya gönderim veri akışı vardır:

```
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.

WebSphere 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

Bu bilgileri, WebSphere MQ ileti biçimi hakkında bilgi edinmek ve WebSphere MQ iletilerinin boyutunu küçültmek için kullanın.

WebSphere MQ iletilerinin, ileti tanımlayıcısında yer alan özniteliklerle ilişkili bir dizi özniteliği vardır. Küçük iletiler için, bu öznitelikler veri trafiğinin çoğunu gösterebilir ve iletim hızı üzerinde önemli bir zarar verici etkiye sahip olabilir. WebSphere MQ Multicast, bu özniteliklerden herhangi biri iletiyle birlikte iletilirse, kullanıcının konfigürasyonunu tanımlamalarını sağlar.

İleti özniteliklerinin varlığı, konu dizgisinden farklı olarak, COMMINFO nesnesinin gönderilip gönderilmemesine bağlı olup olmadığına bağlıdır. Bir öznitelik iletilmezse, alan uygulaması varsayılan bir değer uygular. Varsayılan MQMD değerleri, MQMD_XX_ENCODE_CASE_ONE default değeriyle aynı olmak zorunda değildir ve [Çizelge 9 sayfa 150](#)' ta açıklanmıştır.

COMMINFO nesnesi, MQMD alanlarının ve kullanıcı özelliklerinin birçoğunun iletiyle nasıl akıp geçeceğini denetleyen MCPROP özniteliğini içerir. Bu özniteliğin değerini uygun bir düzeye ayarlayarak, WebSphere 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:

- MessageType
- 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 WebSphere Message Broker RMM uygulamalarıyla 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.

<i>Çizelge 9. İleti alışverişi öznitelikleri ve bunların çok hedefli olarak nasıl ilişkilendirildikleri</i>		
Öznitelik	Çoklu yayın kullanan işlem	Varsayılan değer iletilmezse
TopicString	Her Zaman Dahil	Burada geçerli değil
MQMQ StrucId	İletilmedi	Burada geçerli değil
MQMD Sürümü	İletilmedi	Burada geçerli değil
Rapor	Varsayılan değilse dahil edilir	0

Çizelge 9. İleti alışverişi öznelikleri ve bunların çok hedefli olarak nasıl ilişkilendirildikleri (devamı var)

Öznitelik	Çoklu yayın kullanan işlem	Varsayılan değer iletilmezse
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
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](#)

[DEFINE KOMUTU](#)

Multicast ileti sistemi için veri dönüştürmenin etkinleştirilmesi

WebSphere MQ Multicast ileti sistemi için veri dönüştürme çalışmalarının nasıl çalıştığını anlamak için bu bilgileri kullanın.

WebSphere MQ Multicast paylaşılan, bağlantısız bir iletişim kuralıdır ve bu nedenle her istemcinin veri dönüştürmesi için belirli istekler hazırlamaması mümkün değildir. Aynı çok noktaya yayın akışına abone olan her istemci aynı ikili verileri alır; bu nedenle, WebSphere 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, multicasst COMMINFO nesnesinin **CCSID** ve **ENCODING** değerleri, verimlilik için ileti iletiminin kodlamasını tanımlamak için kullanılabilir.

WebSphere 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 ClientExitPath ile ilgili daha fazla bilgi için, istemci yapılandırma dosyasının [ClientExitYol](#) kısmına bakın.

Çok hedefli uygulama izleme

WebSphere MQ Multicast uygulamasını 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.

Yayıncıların ve abonelerin durumunu görüntülemek için MQSC komutunu **DISPLAY TPSTATUS** kullanın. **DISPLAY TPSTATUS** komutuna ilişkin ek bilgi için [DISPLAY TPSTATUS](#) başlıklı konuya bakın.

COMMEV ve çok hedefli ileti güvenilirliği göstergesi

COMMINFO nesnesinin **COMMEV** özneliğiyle birlikte kullanılan *güvenilirlik göstergesi*, WebSphere MQ Multicast yayıncıların ve abonelerinin izlenmesinde önemli bir öğedir. Güvenilirlik göstergesi (Yayınlama ya da Abone Olma durumu komutlarında döndürülen **MSGREL** alanı), hata içermeyen iletimlerin yüzdesini gösteren bir WebSphere MQ göstergesidir. Bazen iletilerin iletim hatası nedeniyle yeniden iletilmesi gerekir; bu da **MSGREL** değerine yansıtılır. 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** değıştirgesinde tanımlı bir sıklıkla yazılır.

KURAL DIŞI DURUM

İleti güvenilirliği güvenilirlik eşığının 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

WebSphere MQ Multicast aboneliğinin ve ileti geçmişinin nasıl ayarlanacağını öğrenmek için bu bilgileri kullanın.

Çok hedefli iletim hatasının önemli bir ögesi WebSphere MQ' nun iletilen verileri arabelleğe alma (bağlantının iletilmesinin bitiminde tutulacak ileti geçmişi). This process means that no buffering of messages is required in the putting application process because WebSphere MQ provides the reliability. Bu tarihin boyutu, aşağıdaki bilgilerde anlatıldığı gibi, iletişim bilgileri (COMMINFO) nesnesi aracılığıyla yapılandırılır. Daha büyük bir iletim arabelleği, gerekirse yeniden iletilecek daha fazla iletim geçmişi olduğu anlamına gelir, ancak çoklu yayının niteliği nedeniyle %100 güvenli teslim desteklenemez.

WebSphere MQ Multicast ileti geçmişi, 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 geçmişi miktarıdır.

0 değeri, en az güvenilirlik düzeyini verir. Varsayılan değer 100 KB 'dir.

WebSphere MQ Multicast yeni abonelik geçmişi, iletişim bilgileri (COMMINFO) nesnesinde **NSUBHIST** özneliğe göre denetlenir:

NSUBHIST

Yeni abone geçmişi, bir yayın akışına katılan bir abonenin şu anda kullanılabilir olduğu kadar veri alıp almadığını ya da yalnızca abonelik süresinden yapılan yayınları aldığını denetler.

YOK

NONE değeri, vericinin yalnızca abonelik zamanından yapılan yayını iletmeye neden olur. Varsayılan değer, 'NONE ' değeridir.

TÜMÜ

ALL değeri, vericinin, bilindiği gibi konunun çok geçmiş tarihini yeniden aktarmasına neden olur. bazı durumlarda bu durum, alıkonan yayınlara da benzer bir davranış verebilir.

Not: Büyük bir konu geçmişi varsa, ALL değerinin kullanılması, tüm konu geçmişinin yeniden iletilmesinden dolayı büyük bir konu geçmişi, başarımlar üzerinde zarar verebilir.

İlgili başvurular

[DEFINE KOMUTU](#)

[ALTER KOMUT](#)

Gelişmiş çok noktaya gönderim görevleri

.ini dosyalarını yapılandırma ve WebSphere MQ LLM ile birlikte çalışabilirlik gibi gelişmiş WebSphere MQ Multicast yönetim 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 bir WebSphere MQ Multicast etkinleştirilmiş bir konuya yayınlandığında ne olacağını anlamak için bu bilgileri kullanın.

Çok hedefli olmayan bir yayıncı, **MCAST** etkin ve **BRIDGE** etkin olarak tanımlanmış bir konuya yayınlarsa, kuyruk yöneticisi iletiyi çok hedefli olarak doğrudan dinlemiş olabilecek tüm abonelere iletir. Çok noktaya yayın yapan bir yayıncı, çoklu yayın özelliğinin etkinleştirilmemiş konulara yayınlamaz.

Var olan konular, bir konu nesnesinin **MCAST** ve **COMMINFO** parametreleri ayarlanarak çoklu yayın özelliği etkinleştirilebilir. Bu parametrelerle ilgili ek bilgi için [İlk çoklu yayın kavramları](#) başlıklı konuya bakın.

COMMINFO nesnesi **BRIDGE** özneliği, çok hedefli uygulamalar kullanmayan uygulamalardan yayınlar denetler. **BRIDGE** ENABETLE olarak ayarlanırsa ve konunun **MCAST** parametresi de ETKİN olarak ayarlandıysa, çoklu yayın kullanmayan uygulamalardan gelen yayınlar, yapılan uygulamalara köprülülük olarak ayarlanır. **BRIDGE** parametresine ilişkin ek bilgi için [DEFINE COMBILGIS](#) başlıklı konuya bakın.

Çoklu yayın için .ini dosyaları yapılandırılıyor

.ini kütüklerindeki WebSphere MQ Multicast alanlarını anlamak için bu bilgileri kullanın.

Ek WebSphere 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` , WebSphere MQ veri dizininin yeri (`/var/mqm/mqclient.ini`) ve `QMNAME` , .ini dosyasının geçerli olduğu kuyruk yöneticisinin adıdır.

.ini dosyası, WebSphere MQ Multicast: davranışını ince ayar olarak kullanılan alanlar 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

Hangi protokolün kullanılabilir olduğuna bağlı olarak, IPv4, IPv6ya da her ikisi kullanılarak iletişim kurun.

her ikisi

Hem IPv4 hem de IPv6 kullanarak iletişimi destekler.

LimitTransOranı

DEVRE DIŐI

Herhangi bir iletim hızı denetimi yoktur. Bu varsayılan deęerdir.

STATIC

Statik iletim hızı denetimini gerekleřtirir. Verici, TransRateLimit parametresi tarafından belirlenen oranı aŐan bir hızda iletilemedi.

Devingen

Verici iletim hızını alıcılardan aldıęı geri bildirimlere gre uyarlar. Bu durumda, iletim hızı sınırı, TransRateLimit parametresi tarafından belirtilen deęerden daha fazla olamaz. Verici, optimum iletim hızına ulaŐmaya alıŐır.

TransRateSınırı

Kb/s cinsinden iletim hızı sınırı.

SocketTTL

SocketTTL deęeri, oklu yayın trafięin bir yneltici zerinden geebileceęini ya da geebileceęi ynelticilerin sayısını belirler.

Toplu

İletilerin toplu mı, yoksa hemen mi gnderileceęini denetler: 2 olası deęer vardır:

- *NO* (Hayır) İletiler toplu olarak gnderilmez, hemen gnderilir.
- *YES* (EVET) İleti vuruŐlanmış.

Dng

oklu yayın dnglerini etkinleŐtirmek iin deęeri 1 olarak ayarlayın. oklu yayın dngs, gnderilen verilerin anasisteme geri gnderilip geri gnderilmedięini tanımlar.

Arabirim

ok hedefli trafik akıŐlarının zerindeki arabirim IP adresi. Daha fazla bilgi ve sorun giderme iin Őu konuya bakın: [Multicast uygulamalarının oklu yayın olmayan bir aę zerinde test edilmesi](#) ve [oklu yayın trafięi iin uygun aęin ayarlanması](#)

FeedbackMode

NACK

Olumsuz onaylarla geri bildirim. Bu varsayılan deęerdir.

ACK

Olumlu kabul grerek geri bildirim.

WAIT1

Vericinin, alıcıların herhangi birinden yalnızca 1 adet ACK ' yi bekledięi olumlu kabul grerek geri bildirim.

HeartbeatTimeout

Saęlıklı iŐletim bildirim zamanasını (milisaniye). 0 deęeri, saęlıklı iŐletim bildirim olaylarının alıcı ya da konu alıcıları tarafından ykseltilmedięini belirtir. Varsayılan deęer 20000 'dir.

HeartbeatInterval

Saęlıklı iŐletim bildirim aralıęı (milisaniye). 0 deęeri, herhangi bir kalp atımı gnderilmemesinin gnderildięini gsterir. Saęlıklı iŐletim bildirim zamanasını olaylarını nlemek iin saęlıklı iŐletim bildirim aralıęı, **HeartbeatTimeout** deęerinden byk lde daha kk olmalıdır. Varsayılan deęer 2000 'dir.

WebSphere MQ Low Latency Messaging ile ok hedefli birlikte iŐlerlik

WebSphere MQ Multicast ve WebSphere MQ Low Latency Messaging (LLM) arasındaki birlikte iŐlerlięi anlamak iin bu bilgileri kullanın.

LLM kullanan bir uygulama iin temel bilgi yk aktarımı, iletileri her iki ynde de deęiŐ tokuŐ etmek iin oklu yayın kullanan baŐka bir uygulama ile mmkndr. oklu yayın, LLM teknolojisini kullansa da, LLM rnnn kendisi yerleŐik deęildir. Bu nedenle, hem LLM hem de WebSphere MQ Multicast programını kurmak ve iki rn ayırarak alıŐtırmak ve alıŐtırmak mmkndr.

Çoklu yayınla iletişim kuran LM uygulamalarının ileti özelliklerini gönderip alması gerekebilir. WebSphere MQ ileti özellikleri ve MQMD alanları, aşağıdaki tabloda gösterildiği gibi, belirli LLM ileti özelliği kodlarına sahip LLM ileti özellikleri olarak iletilir:

Çizelge 10. WebSphere MQ LLM özellik eşlemelerineWebSphere MQ ileti özellikleri

WebSphere MQ özelliği	WebSphere 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, bkz. LLM ürün belgelerine bakın: [WebSphere MQ Low Latency Messaging](#).

YönetmeHP Integrity NonStop Server

Use this information to learn about administration tasks for the IBM WebSphere MQ client for HP Integrity NonStop Server.

İki yönetim görevi size sağlanır:

1. Pathway 'den TMF/Gateway 'i el ile başlatma.
2. Pathway 'den TMF/Gateway durduruluyor.

Pathway 'den TMF/Gateway 'i el ile başlatma

First enlistment isteğinde otomatik olarak TMF/Gateway olanağını başlatmak için Pathway 'e izin verebilir ya da TMF/Gateway olanağını Yol 'dan el ile başlatabilirsiniz.

Yordam

TMF/Gateway 'i Yol Yolundan el ile başlatmak için aşağıdaki PATHCOM komutunu girin:

```
START SERVER <server_class_name>
```

Bir istemci uygulaması, TMF/Gateway belirsiz hareketlerin kurtarılması işlemini tamamlamadan önce bir kayıt isteği gerçekleştirirse, istek 1 saniyeye kadar tutulur. Kurtarma işlemi o süre içinde tamamlanmazsa, belge reddedilen bir öge reddedilir. İstemci, bir işlemsel MQI kullanımından MQRC_UOW_ENLISTMENT_ERROR hatası alır.

Yol Yolu 'ndan TMF/Gateway durduruluyor

Bu kısımda, TMF/Gateway ' in Pathway 'den nasıl durdurulacağı ve durdurmadan sonra TMF/Gateway 'in nasıl yeniden başlatılacağı ele alınmıştır.

Yordam

1. TMF/Gateway 'e yeni bir giriş isteğinde bulunmayı önlemek için aşağıdaki komutu girin:

```
FREEZE SERVER <server_class_name>
```

2. Uçuş içi işlemleri tamamlamak ve sona erdirmek için TMF/Gateway 'i tetiklemek için şu komutu girin:

```
STOP SERVER <server_class_name>
```

3. To allow the TMF/Gateway to restart either automatically on first enlistment or manually, following steps 1 and 2, enter the following command:

```
THAW SERVER <server_class_name>
```

Applications are prevented from making new enlistment requests and it is not possible to issue the **START** command until you issue the **THAW** command.

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 kullanılabilmesini göstermez. Aynı işlevi gören ve IBM'in fikri mülkiyet haklarına zarar vermeyen herhangi bir ürün, program ya da hizmet de kullanılabilir. Ancak, IBM dışı kaynaklardan sağlanan 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 Licensing 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:

Fikri Mülkiyet Lisanslama Yasal ve Fikri Mülkiyet Hukuku IBM Japonya, 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 yayında IBM dışı Web sitelerine yapılan göndermeler yalnızca bilgilendirme amacıyla yapılmıştır ve bu, hiçbir şekilde ilgili Web sitelerinin onaylandığı anlamına gelmeyebilir. 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 Software Interoperability Coordinator, Department 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

sonular ngr yntemiyle elde edilmiř olabilir. Dolayısıyla, gerek sonular farklı olabilir. Bu belgenin kullanıcıları, kendi ortamları iin geerli verileri kendileri doėrulamaladırlar.

IBM dıřı rnlerle ilgili bilgiler, bu rnleri saėlayan firmalardan, bu firmaların yayın ve belgelerinden ve genel kullanıma aık diėer kaynaklardan alınmıřtır. IBM bu rnleri test etmemiřtir ve IBM dıřı rnlerle ilgili performans doėruluėu, uyumluluk gibi iddiaları doėrulamaz. IBM dıřı rnlerin yeteneklerine iliřkin sorular, bu rnleri saėlayan firmalara yneltilmelidir.

IBM' in gelecekteki ynelim ve kararlarına iliřkin tm bildirimler deėiřebilir ya da herhangi bir duyuruda bulunulmadan bunlardan vazgeilir; bu ynelim ve kararlar yalnızca ama ve hedefleri gsterir.

Bu belge, gnlk iř ortamında kullanılan veri ve raporlara iliřkin rnekler ierir. rneklerin olabildiėince aıklayıcı olması amacıyla kiři, řirket, marka ve rn adları belirtilmiř olabilir. Bu adların tm gerek dıřıdır ve gerek iř ortamında kullanılan ad ve adreslerle olabilecek herhangi bir benzerlik tmyle rastlantıdır.

YAYIN HAKKI LİSANSI:

Bu belge, eřitli iřletim platformlarında programlama tekniklerini gsteren, kaynak dilde yazılmıř rnek uygulama programları ierir. Bu rnek programları, IBM'e herhangi bir demede bulunmadan, rnek programların yazıldıėı iřletim altyapısına iliřkin uygulama programlama arabirimiyle uyumlu uygulama programlarının geliřtirilmesi, kullanılması, pazarlanması ya da daėıtılması amacıyla herhangi bir biimde kopyalayabilir, deėiřtirebilir ve daėıtabilirsiniz. Bu rnekler her kořul altında tm ayrıntılarıyla sınanmamıřtır. Dolayısıyla, IBM bu programların gvenilirliėi, bakım yapılabilirliėi ya da iřlevleri konusunda aık ya da rtk gvence veremez.

Bu bilgileri elektronik kopya olarak grntlediyseniz, fotoėraflar ve renkli resimler grnmeyebilir.

Programlama arabirimi bilgileri

Programlama arabirimi bilgileri (saėlandıysa), bu programla birlikte kullanılmak zere uygulama yazılımları yaratmanıza yardımcı olmak zere hazırlanmıřtır.

Bu kitap, mřterinin IBM WebSphere MQ hizmetlerini edinmek zere program yazmasına olanak tanıyan, amalanan programlama arabirimlerine iliřkin bilgiler ierir.

Ancak, bu bilgiler tanılama, deėiřtirme ve ayarlama bilgilerini de ierebilir. Tanılama, deėiřtirme ve ayarlama bilgileri, uygulama yazılımlarınızda hata ayıklamanıza yardımcı olur.

nemli: Bu tanılama, deėiřtirme ve ayarlama bilgilerini bir programlama arabirimi olarak kullanmayın; bu, deėiřiklik sz konusu olduėunda kullanılır.

Ticari Markalar

IBM, IBM logosu, ibm.com, IBM Corporation 'ın dnya apında birok farklı hukuk dzeninde kayıtlı bulunan ticari markalarıdır. IBM ticari markalarının gncel bir listesini Web zerinde "Telif hakkı ve ticari marka bilgileri" www.ibm.com/legal/copytrade.shtml adresinde bulabilirsiniz. Diėer rn ve hizmet adları IBM'in veya diėer řirketlerin ticari markaları olabilir.

Microsoft ve Windows, Microsoft Corporation'ın ABD ve/veya diėer lkelerdeki ticari markalarıdır.

UNIX, The Open Group řirketinin ABD ve diėer lkelerdeki tescilli ticari markasıdır.

Linux, Linus Torvalds'ın ABD ve/ya da diėer 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: