

7.5

IBM WebSphere MQ planlaması

IBM

Not

Bu bilgileri ve desteklediđi ürünü kullanmadan önce, "[Özel notlar](#)" sayfa 153 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

Planlama.....	5
IBM MQ ve IBM MQ Appliance şirket içinde GDPR ' ye hazır olma ile ilgili önemli noktalar.....	5
IBM WebSphere MQ mimarisinin tasarlanması.....	14
Tek kuyruk yöneticisi mimarileri.....	15
Tek kuyruk yöneticisi mimarileri.....	15
Tek kuyruk yöneticisi mimarileri.....	15
Tek kuyruk yöneticisi mimarileri.....	15
Birden çok kuyruk yöneticisi mimarisi.....	15
Noktadan Noktaya İleti Sistemi.....	19
Yayınlama/abone olma giriş.....	19
Teslim edilmeyen iletilerin, kuyruk-kuyruk işleyicisiyle işlenmesi.....	105
Birden çok kuruluş için planlama.....	114
Birincil kuruluş seçilmesi.....	116
Depolama ve performans gereksinimlerinizin planlanması.....	120
Disk alanı gereksinimleri.....	121
Planlama dosya sistemi desteği.....	122
IBM WebSphere MQ ve UNIX System V IPC kaynakları.....	147
AIXüzerinde paylaşılan bellek.....	148
UNIX Process Priority değerlerini ayarlama.....	148
HP Integrity NonStop Serverüzerinde IBM WebSphere MQ istemci ortamınızın planlanması.....	149
HP Integrity NonStop Server ortamını hazırlama.....	149
IBM WebSphere MQ ve HP NonStop TMF.....	149
HP NonStop TMF ' yi kullanma.....	150
Özel notlar.....	153
Programlama arabirimi bilgileri.....	154
Ticari Markalar.....	154

Planlama

IBM WebSphere MQ ortamınızı planlarken, yapılandırmak istediğiniz IBM WebSphere MQ mimarisini, kaynak gereksinmelerini, günlüğe kaydetme gereksinmesini ve yedekleme olanaklarını göz önünde bulundurmanız gerekir. Use the links in this topic to plan the environment where IBM WebSphere MQ runs.

IBM WebSphere MQ ortamınızı planlamadan önce, temel IBM WebSphere MQ kavramlarını öğrenin ve [Teknik genel bakış](#) başlıklı konuda konuya bakın.

İlgili kavramlar

[Kullanılabilirlik, kurtarma ve yeniden başlatma](#)

İlgili görevler

[Geçiş yapıyor](#)

[Kuruluyor](#)

[Yapılandırılıyor](#)

[WebSphere MQ' un denetlenmesi](#)

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

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

PID ' ler için:

- 5724-H72 IBM MQ
- 5655-AV9 IBM MQ Advanced for z/OS
- 5655-AV1 IBM MQ Advanced for z/OS, Value Unit Edition
- 5655-AM9 IBM MQ Advanced Message Security for z/OS
- 5725-Z09 IBM MQ Appliance M2001
- 5725-S14 IBM MQ Appliance M2000
- 5655-MQ9 IBM MQ for z/OS
- 5655-VU9 IBM MQ for z/OS Value Unit Edition
- 5639-L92 IBM MQ Internet Pass-Thru
- 5655-MF9 IBM MQ Managed File Transfer for z/OS
- 5655-ADV IBM WebSphere MQ Advanced for z/OS
- 5655-AMS IBM WebSphere MQ z/OS
- 5724-R10 IBM WebSphere MQ File Transfer Edition for Multiplatforms
- 5724-A39 IBM WebSphere MQ for HP NonStop Server
- 5724-A38 IBM WebSphere MQ for HP OpenVMS
- 5655-W97 IBM WebSphere MQ for z/OS
- 5655-VU8 IBM WebSphere MQ for z/OS Value Unit Edition
- z/OS Value Unit Edition için 5655-VUE IBM WebSphere MQ
- 5725-C79 IBM WebSphere MQ Hypervisor Edition for Red Hat Enterprise Linux® for x86
- 5725-F22 IBM WebSphere MQ AIX
- 5655-MFT IBM WebSphere MQ Managed File Transfer for z/OS

Bildirim:

Bu belgenin amacı, GDPR için hazır olma çalışmalarınıza yardımcı olmaktır. Yapılandırabileceğiniz IBM MQ özellikleri ve kuruluşunuzun GDPR 'ye hazır olmalarına yardımcı olmak için göz önünde bulundurmanız gereken ürün kullanımı özellikleri hakkında bilgi sağlar. Müşterilerin özellikleri seçip yapılandırabileceği ve ürünün kendi içinde ya da üçüncü taraf uygulamaları ve sistemleri ile kullanılabileceği birçok yol olduğundan, bu bilgiler çok kapsamlı değildir.

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

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

İçindekiler

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

GDPR/GVKY

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

GDPR neden önemli?

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

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

GDPR hakkında daha fazla bilgi edinin:

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

Ürün Yapılandırması - GDPR'ye hazırlıkla ilgili önemli noktalar

Aşağıdaki bölümlerde, IBM MQ 'nun GDPR' ye hazır olma konusunda kuruluşunuza yardımcı olacak şekilde yapılandırılmasına ilişkin önemli noktalar sağlanır.

Veri Yaşam Çevrimi

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

IBM MQ aracılığıyla hangi veri akışı tipleri?

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

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

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

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

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

IBM ile çevrimiçi iletişim için kullanılan kişisel veriler

IBM MQ müşterileri, IBM MQ konuları hakkında IBM ile iletişim kurmak için çeşitli şekillerde çevrimiçi yorumlar/feedback/talepler gönderebilir:

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

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

Veri Toplama

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

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

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

Veri Depolama

İleti verileri kuyruk yöneticileri aracılığıyla hareket ettiğinde, IBM MQ bu verileri (belki birden çok kopyası) doğrudan durumlu ortama aktarmaya devam eder. IBM MQ kullanıcıları, dururken ileti verilerinin güvenliğini sağlamak isteyebilir.

Aşağıdaki öğeler, IBM MQ ' nun uygulama tarafından sağlanan verileri kalıcı olarak içerdiği ve kullanıcıların GDPR ile uyumluluğu sağlarken dikkate almak isteyecekleri alanları vurgular.

- Uygulama İleti Kuyrukları:

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

- Dosya Aktarma Aracısı Kuyrukları:

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

- İletim Kuyrukları:

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

- Gönderilmeyen Mektup Kuyrukları:

Kuyruk yöneticisinde yapılandırılmış bir ileti varsa, iletilerin hedef kuyruğa konamayacağı ve gitmeyen iletiler kuyruğunda saklanacağı bazı durumlar vardır.

- Geriletme Kuyrukları:

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

- AMS Hata Kuyruğu:

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

- Alıkonan Yayınlar:

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

Daha fazla bilgi:

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

Aşağıdaki öğeler, IBM MQ 'nun kullanıcıların GDPR' ye uygunluğu sağlarken dikkate almak isteyecekleri uygulama verilerini dolaylı olarak saklayabileceği alanları vurgular.

- İzleme rotası ileti alışverişi:

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

- Uygulama etkinliği izlemesi:

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

- Hizmet izleme:

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

- Kuyruk yöneticisi olayları:

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

Daha fazla bilgi:

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

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

- Ayrıcalıklı kullanıcı erişimini dosya sistemindeki IBM MQ verileriyle sınırlandırın; örneğin, UNIX platformlarında 'mqm' grubunun kullanıcı üyeliğini kısıtlayın.
- Özel olarak ayrılmış kuyruklar ve erişim denetimi aracılığıyla IBM MQ verilerine uygulama erişimini kısıtlayın. Uygun olduğunda, kuyruklar gibi kaynakların uygulamalar arasında gereksiz yere paylaşılmasından kaçının ve kuyruk ve konu kaynakları için ayrıntılı erişim denetimi sağlayın.
- İleti verilerinin uçtan uca imzalanmasını ve/veya şifrelenmesini sağlamak için IBM MQ Advanced Message Security olanağını kullanın.
- İzleme günlüklerini saklamak için kullanılan dizinin içeriğini korumak için dosya ya da birim düzeyinde şifreleme kullanın.
- Hizmet izlemesini IBM' e yükledikten sonra, kişisel verileri içerebilecek içerikten endişe duyuyduysanız, hizmet izleme dosyalarınızı ve FFST verilerinizi silebilirsiniz.

Daha fazla bilgi:

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

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

Bir IBM MQ kuyruk yöneticisi yaratıldığında, veri dizini grup tabanlı erişim denetimiyle ayarlanır; böylece IBM MQ yapılandırma dosyalarını okuyabilir ve bu sistemlere bağlanmak için kimlik bilgilerini kullanabilir. IBM MQ yöneticileri ayrıcalıklı kullanıcılar olarak kabul edilir ve bu grubun üyeleridir; bu nedenle dosyalara okuma erişimi vardır. Bazı dosyalar gizlenmiş, ancak şifrelenmemiş. Bu nedenle, kimlik bilgilerine erişimi tam olarak korumak için aşağıdaki işlemleri göz önünde bulundurmanız gerekir:

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

Daha fazla bilgi:

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

Veri Erişimi

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

- IBM MQ Konsolu [Yalnızca Uzak]
- IBM MQ REST API [Yalnızca Uzak]
- MQI [Yerel ve Uzak]
- JMS [Yerel ve Uzak]
- XMS [Yerel ve Uzak]
- IBM MQ Telemetry (MQTT) [Yalnızca Uzak]
- IBM MQ Light (AMQP) [Yalnızca Uzak]
- IBM MQ IMS köprüsü [Yalnızca Yerel]
- IBM MQ CICS köprüsü [Yalnızca Yerel]
- IBM MQ bridge for HTTP [Yalnızca Uzak]
- IBM MQ MFT İletişim Kuralı köprüleri [Yalnızca Uzak]
- IBM MQ Connect:Direct köprüleri [Yalnızca Uzak]
- IBM MQ Bridge to Salesforce [Yalnızca Uzak]
- IBM MQ Bridge to Blockchain [Yalnızca Uzak]
- IBM MQ MQAI [Yerel ve Uzak]
- IBM MQ PCF komutları [Yerel ve Uzak]
- IBM MQ MQSC komutları [Yerel ve Uzak]
- IBM MQ Explorer [Yerel ve Uzak]

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

- Kimlik denetimi
- Rol eşleme
- Yetkilendirme

Kimlik denetimi:

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

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

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

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

Rol eşleme:

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

Yetki:

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

Günlüğe kaydetme etkinliği:

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

Bu gereksinimi gerçekleştirmek için aşağıdaki bilgi kaynakları kullanılabilir:

1. IBM MQ kuyruk yöneticisi, bir denetim komutu başarıyla çalıştırıldığında komut olayları üretecek şekilde yapılandırılabilir.
2. Bir IBM MQ kuyruk yöneticisi, bir kuyruk yöneticisi kaynağı yaratıldığında, değiştirildiğinde ya da silindiğinde yapılandırma olayları üretecek şekilde yapılandırılabilir.
3. IBM MQ kuyruk yöneticisi, bir kaynak için yetki denetimi başarısız olduğunda bir yetki olayı üretecek şekilde yapılandırılabilir.
4. Başarısız yetki denetimlerinin kuyruk yöneticisi hata günlüklerine yazıldığını gösteren hata iletileri.
5. IBM MQ Web konsolu, kimlik doğrulama, yetkilendirme denetimleri başarısız olduğunda ya da kuyruk yöneticileri oluşturulduğunda, başlatıldığında, durdurulduğunda ya da silindiğinde denetim iletilerini günlüklerine yazar.

Bu tür çözümleri düşünürken, IBM MQ kullanıcıları aşağıdaki noktaları dikkate almak isteyebilirler:

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

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

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

Verilerin İşlenmesi

Genel Anahtar Altyapısı kullanılarak şifreleme:

TLS ' yi kullanmak için IBM MQ ile ağ bağlantılarının güvenliğini sağlayabilirsiniz; bu, bağlantının başlangıç tarafının karşılıklı kimlik doğrulamasını da sağlayabilir.

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

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

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

Ağ bağlantılarınızın güvenliğini sağlamak için SSL ve sertifikaların kullanılmasıyla ilgili daha fazla bilgi için IBM MQ V9 ürün belgelerinde aşağıdaki konulara bakın:

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

Verilerin Silinmesi

IBM MQ , ürüne sağlanan verileri silmek için komutlar ve kullanıcı arabirimi işlemleri sağlar. Bu, olanaklara sahip IBM MQ kullanıcılarının belirli kişilerle ilgili verileri silmesini sağlar.

- IBM MQ ' nun GDPR Müşteri Verilerini silme işlemine uymak için göz önünde bulundurulması gereken alanları
 - Uygulama kuyruğunda saklanan ileti verilerini silmek için:
 - İleti alışverişi API 'sini ya da araçlarını kullanarak ya da ileti süre bitimini kullanarak tek tek iletileri kaldırma.
 - İletilerin kalıcı olmadığını, kalıcı olmayan ileti sınıfının normal olduğu bir kuyrukta tutulduğunu ve kuyruk yöneticisini yeniden başlattığı belirtiliyordu.
 - Yönetim tarafından kuyruk temizleniyor.
 - Kuyruk siliniyor.
 - Bir konuda saklanan alıkonan yayın verilerini silmek için:
 - İletilerin kalıcı olmadığını belirtme ve kuyruk yöneticisini yeniden başlatma.
 - Alıkonan verileri yeni verilerle ya da ileti süre bitimini kullanarak değiştirme.
 - Yönetimsel olarak konu dizesi temizleniyor.
 - Tüm kuyruk yöneticisini silerek bir kuyruk yöneticisinde saklanan verileri silin.
 - İzleme dizinindeki dosyaları silerek, Hizmet izleme komutlarıyla saklanan verileri silin.
 - Hatalar dizinindeki dosyaları silerek saklanan FFST verilerini silin.
 - Adres alanını ve Coupling Facility dökümlerini silin (z/OS üzerinde).

- Bu tür verilerin arşivini, yedeklemesini ya da diğer kopyalarını silin.
- IBM MQ ' nun GDPR Hesap Verilerini silme işlemine uymak için göz önünde bulundurulması gereken alanları
 - Kuyruk yöneticilerine ve 3rd taraf hizmetlerine bağlanmak için IBM MQ tarafından depolanan hesap verilerini ve tercihlerini silmek için bunları silebilirsiniz (arşivleme, yedekleme ya da başka bir şekilde eşlenmiş kopyaları da içinde olmak üzere):
 - Kimlik bilgilerini saklayan kuyruk yöneticisi kimlik doğrulama bilgileri nesnelere.
 - Kullanıcı tanıtıcılarına başvuran kuyruk yöneticisi yetkisi kayıtları.
 - Belirli IP adreslerini, sertifika DN ' lerini ya da kullanıcı tanıtıcılarını eşleyen ya da engelleyen kuyruk yöneticisi kanalı kimlik doğrulama kuralları.
 - IBM MQ Managed File Transfer Agent, Logger ve MQ Explorer MFT Plugin tarafından kuyruk yöneticisi ve dosya sunucularıyla kimlik doğrulaması için kullanılan kimlik bilgileri dosyaları.
 - X.509 , SSL/TLS bağlantıları ya da IBM MQ Advanced Message Security (AMS) tarafından kullanılacak anahtar depolarından bir kişiyi temsil eden ya da bu kişiye ilişkin bilgileri içeren dijital sertifikalar.
 - Sistem günlük dosyalarında bu hesaplara başvuru da dahil olmak üzere, IBM MQ Appliance' dan tek tek kullanıcı hesapları.
 - IBM MQ Explorer çalışma alanı meta verileri ve Eclipse ayarları.
 - IBM MQ Explorer parola deposu, Parola Tercihleri' nde belirtildiği gibi.
 - IBM MQ Console ve mqweb sunucusu yapılandırma dosyaları.
 - Salesforce bağlantı verileri yapılandırma dosyaları.
 - blockchain bağlantı verileri yapılandırma dosyaları.
 - IBM Cloud Product Insights bağlantı verileri qm.ini ve APIKeyFileiçindeki ReportingService kısmı altında.

Daha fazla bilgi:

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

Verilerin İzlenmesi

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

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

Daha fazla bilgi:

- IBM MQ ağınızın izlenmesi
- Tanımlama iletisi hizmetleri
- QMErrorLog hizmeti

IBM MQ , kullanıcıların bir IBM Cloud Product Insights hizmetine bilgi yayınlamalarını sağlayan bir özellik sağlar; böylece, IBM MQ kullanıcısı kuyruk yöneticisi başlatma ve kullanım bilgilerini görüntüleyebilir.

Daha fazla bilgi:

- IBM MQ ürününü IBM Cloud içinde IBM Cloud Product Insights hizmetiyle kullanmak üzere yapılandırma

Kişisel Verilerin Kullanımını Kısıtlama Yeteneği

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

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

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

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

Dosya İşleme

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

IBM WebSphere MQ mimarisinin tasarlanması

Noktadan noktaya iletişim ve yayınlama/abone olma ileti stilleri için IBM WebSphere MQ tarafından desteklenen farklı mimariler hakkında bilgi edinin.

IBM WebSphere MQ mimarinizi planlamadan önce, kendinizi temel IBM WebSphere MQ kavramlarına alıştırdığınızda, [IBM WebSphere MQ Teknik genel bakış](#) başlıklı konuda ele geçirdiğiniz konulara bakın.

IBM WebSphere MQ mimarileri, tek bir kuyruk yöneticisi kullanan basit mimarilerden, birbirine bağlı kuyruk yöneticilerine ilişkin daha karmaşık ağlara kadar çeşitli mimariler sağlar. Birden çok kuyruk yöneticisi dağıtılmış kuyruklama tekniklerini kullanarak birbirine bağlanır. Tek kuyruk yöneticisi ve birden çok kuyruk yöneticisi mimarisinin planlanması hakkında ek bilgi için aşağıdaki konulara bakın:

- [“Tek bir kuyruk yöneticisine dayalı mimariler” sayfa 15](#)
- [“Birden çok kuyruk yöneticisine dayalı mimariler” sayfa 15](#)
- [“Ağlar ve Ağ Planlaması” sayfa 16](#)
- [WebSphere MQ dağıtımlı ileti sistemi teknikleri](#)

Mantıksal olarak ilgili olan birden çok kuyruk yöneticisine gereksiniminiz varsa ve veri ve uygulamaları paylaşmanız gerekiyorsa, bunlar bir kümede birlikte gruplandırılabilir. Kümelerin kullanılması, kuyruk yöneticilerinin, ek kanal tanımlamalarını ya da uzak kuyruk tanımlamalarını ayarlama gereksinimi olmadan birbirleriyle iletişim kurmalarını sağlayabilir, yapılandırma ve yönetimlerini basitleştirebilir. Kümeleri kullanma hakkında daha fazla bilgi için bkz. [Kümelerin çalışma şekli](#).

İlgili kavramlar

[“Planlama” sayfa 5](#)

IBM WebSphere MQ ortamınızı planlarken, yapılandırmak istediğiniz IBM WebSphere MQ mimarisini, kaynak gereksinimlerini, günlüğe kaydetme gereksinimini ve yedekleme olanaklarını göz önünde bulundurmanız gerekir. Use the links in this topic to plan the environment where IBM WebSphere MQ runs.

İlgili görevler

[Yapılandırılıyor](#)

Tek bir kuyruk yöneticisine dayalı mimariler

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

IBM WebSphere MQ mimarinizi planlamadan önce, temel IBM WebSphere MQ kavramlarını tanıyın ve [IBM WebSphere MQ' a giriş' i](#) inceleyin.

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

- [“Bir hizmete erişen yerel uygulamalara sahip tek kuyruk yöneticisi” sayfa 15](#)
- [“Bir hizmete istemci olarak erişen uzak uygulamalara sahip tek kuyruk yöneticisi” sayfa 15](#)
- [“Yayınlama/abone olma yapılandırması içeren tek kuyruk yöneticisi” sayfa 15](#)

Bir hizmete erişen yerel uygulamalara sahip tek kuyruk yöneticisi

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

Bir hizmete istemci olarak erişen uzak uygulamalara sahip tek kuyruk yöneticisi

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

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

Yayınlama/abone olma yapılandırması içeren tek kuyruk yöneticisi

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

İlgili kavramlar

[“IBM WebSphere MQ mimarisinin tasarlanması” sayfa 14](#)

Noktadan noktaya iletişim ve yayınlama/abone olma ileti stilleri için IBM WebSphere MQ tarafından desteklenen farklı mimariler hakkında bilgi edinin.

İlgili bilgiler

[WebSphere MQ' a giriş](#)

[Kuyruk yöneticilerinin yaratılması ve yönetilmesi](#)

Birden çok kuyruk yöneticisine dayalı mimariler

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

IBM WebSphere MQ mimarinizi planlamadan önce, temel IBM WebSphere MQ kavramlarını tanıyın ve [IBM WebSphere MQ' a giriş' i](#) inceleyin.

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

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

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

Bir IBM WebSphere MQ ağı planlamaya ilişkin yüksek düzeyli bir genel bakış için bkz. [“Ağlar ve Ağ Planlaması”](#) sayfa 16.

IBM WebSphere MQ mimarisine ilişkin kanalların nasıl planlanabileceği hakkında bilgi için bkz. [WebSphere MQ dağıtım kuyruğu ileti sistemi teknikleri](#).

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

İlgili kavramlar

[WebSphere MQ' a giriş](#)

[“IBM WebSphere MQ mimarisinin tasarlanması”](#) sayfa 14

Noktadan noktaya iletişim ve yayınlama/abone olma ileti stilleri için IBM WebSphere MQ tarafından desteklenen farklı mimariler hakkında bilgi edinin.

İlgili görevler

[Kuyruk yöneticilerinin yaratılması ve yönetilmesi](#)

Ağlar ve Ağ Planlaması

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

Kanallar, sisteminizle ve iletişimde bulunmanız gereken diğer sistemler arasında yaratılabilir. Doğrudan bağlantıların olmadığı sistemlere bağlanmak için çoklu sekme kanalları yaratılabilir. Senaryolarda açıklanan ileti kanalı bağlantıları, [Şekil 1 sayfa 17](#) içinde bir ağ çizgesi olarak gösterilir.

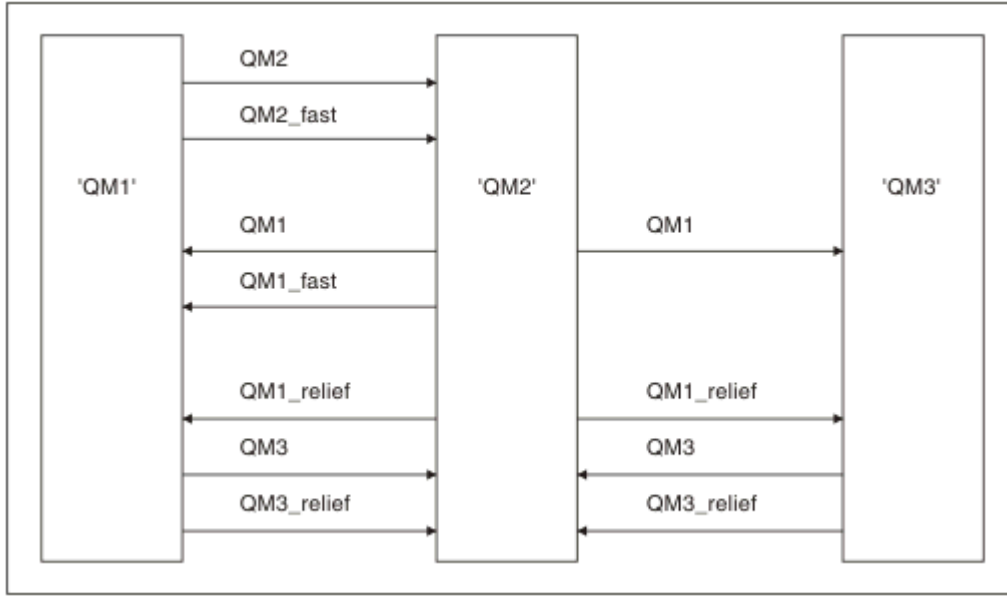
Kanal ve iletim kuyruğu adları

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

Kanal isimleri için o kadar da açık değil. The channel names in [Şekil 1 sayfa 17](#) for QM2, for example, must be different for incoming and outgoing channels. Tüm kanal adları yine de iletim kuyruğu adlarını içerebilir, ancak bunların benzersiz olması için nitelenmeleri gerekir.

Örneğin, QM2'de QM1' den gelen bir QM3 kanalı ve QM3 kanalına giden bir QM3 kanalı vardır. Adları benzersiz yapmak için, birincinin adı 'QM3_from_QM1', ikincinin adı 'QM3_from_QM2' olabilir. Bu şekilde, kanal adları, adın ilk kısmında iletim kuyruğu adını gösterir. Yön ve bitişik kuyruk yöneticisi adı, adın ikinci kısmında gösterilir.

A table of suggested channel names for [Şekil 1 sayfa 17](#) is given in [Çizelge 1 sayfa 17](#).



Şekil 1. Tüm kanalları gösteren ağ çizgesi

Çizelge 1. Kanal adları örneği

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

Not:

1. On WebSphere MQ for z/OS, queue manager names are limited to four characters.
2. Ağınızdaki tüm kanalları benzersiz bir şekilde adlayın. As shown in Çizelge 1 sayfa 17, including the source and target queue manager names in the channel name is a good way to do so.

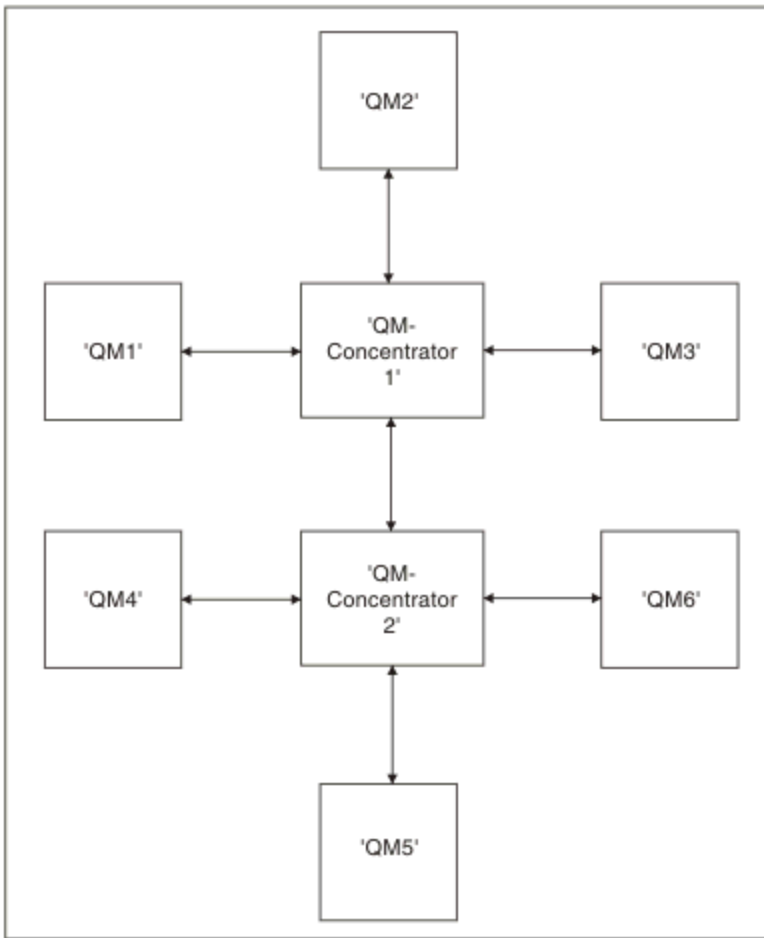
Ağ planlayıcısı

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

For widely used applications, it is more economical to think in terms of local access sites for the concentration of message traffic, using wide-band links between the local access sites, as shown in Şekil 2 sayfa 18.

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

- QM yoğunlaştırıcısı 1'in, üç yerel kuyruk yöneticisinin (QM1, QM2 ve QM3) her birine ileti kanalları vardır. Bu kuyruk yöneticilerini kullanan uygulamalar, QM-yoğunlaştırıcılar aracılığıyla birbirleriyle iletişim kurabilirler.
- QM yoğunlaştırıcısı 2'nin, üç yerel kuyruk yöneticisinin (QM4, QM5 ve QM6) her birine ileti kanalları vardır. Bu kuyruk yöneticilerini kullanan uygulamalar, QM-yoğunlaştırıcılar aracılığıyla birbirleriyle iletişim kurabilirler.
- QM yoğunlaştırıcıların, kuyruk yöneticisinde herhangi bir uygulamanın başka bir kuyruk yöneticisinde başka bir uygulama ile ileti alışverişi yapmasına olanak sağlayarak, QM yoğunlaştırıcılarının kendileriyle ileti kanalları vardır.



Şekil 2. QM yoğunlaştırıcılarını gösteren ağ çizgesi

Kümeleme

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

Aşağıdaki bilgiler, kullanıcının IBM WebSphere MQ kümelerine ilişkin temel bir anlayışına sahip olduğunu varsayar. Bu bilgiler, bir "bir boyut tümüne uyuyor" çözümü olarak tasarlanmamaktadır, ancak bunun yerine ortak sorunlara ortak yaklaşımlar paylaşılmaya çalışılır.

Kümeler, sistemi kurmak için gereken ilk yapılandırmayı ve gereken yönetimi basitleştiren bir şekilde, kuyruk yöneticilerini birbirine bağlamak için bir mekanizma sağlar. Yapılandırma ne kadar büyükse, avantaj da o kadar artar.

Kümeleme sistemlerinin doğru şekilde çalışmasını sağlamak ve sistemin, özellikle daha büyük veya daha karmaşık kümelmiş sistemler için gerekli kullanılabilirlik ve yanıt verebilirliği sağlamak için kümeleme planlamasında dikkatli olun.

Başarılı bir küme kurulumu, iyi bir planlama ve iyi uygulama yönetimi ve ağ tasarımı gibi temel IBM WebSphere MQ temel kavramlarıyla ilgili ayrıntılı bir anlayışla bağlıdır. Ensure that you are familiar with the information in [İletişim kavramları](#) and [Kümeler nasıl çalışır](#).

Kümeler ve neden kullanılırlar?

Kümeleme, iki temel avantaj sağlar:

- Kümeler, genellikle kanallar, iletim kuyrukları ve konfigürasyonu tanımlanacak uzak kuyruklar için birçok nesne tanımlaması gerektiren IBM WebSphere MQ ağlarının yönetimini kolaylaştırır. Bu durum, özellikle çok sayıda kuyruk yöneticisinin birbiriyle bağlantılı olması gereken büyük, potansiyel olarak değişen ağlarda da geçerlidir. Bu mimari, özellikle yapılandırılması ve etkin bir şekilde sürdürülmesi güçtür.
- Kümeler, ileti trafiğindeki iş yükünü kümedeki kuyruklar ve kuyruk yöneticilerine dağıtmak için kullanılabilir. Bu tür bir dağıtım, tek bir kuyruğun ileti iş yükünün, birden çok kuyruk yöneticisi üzerinde bulunan o kuyruğun eşdeğer eşgörünümlerine dağıtılmasını sağlar. İş yükünün bu dağılımı, sistem hatalarına karşı daha fazla esneklik sağlamak ve özellikle bir sistemdeki etkin ileti akışlarının ölçekleme performansını artırmak için kullanılabilir. Böyle bir ortamda, dağıtılmış kuyrukların her bir eşgörünümlü, iletileri işleyen uygulamaları tüketir.

İlgili bilgiler

[Kümeleme: En iyi uygulamalar](#)

Noktadan Noktaya İleti Sistemi

IBM WebSphere MQ içindeki en basit ileti sistemi noktadan noktaya ileti alışverişi biçimidir.

Noktadan noktaya ileti sisteminde, bir gönderme uygulamasının, bu uygulamaya bir ileti gönderebilmesi için önce alma uygulamasına ilişkin bilgileri bilmesi gerekir. Örneğin, gönderme uygulamasının, bilgilerin gönderileceği kuyruğun adını bilmesi ve kuyruk yöneticisi adını da belirtmesi gerekebilir.

IBM WebSphere MQ ile kullanabileceğiniz alternatif bir ileti sistemi stili, yayınlama/abone olma ileti alışverişi olur. Yayınlama/abone olma ileti sistemi, bilgi sağlayıcısını bu bilgilerin tüketicilerinden çözümlenize olanak sağlar. Gönderme uygulaması ve alma uygulamasının, gönderilecek ve alınacak bilgiler için birbiriyle ilgili herhangi bir bilgi sahibi olması gerekmez. Yayınlama/abone olma ileti alışverişi hakkında daha fazla bilgi için bakınız: [Introduction to WebSphere MQ publish/subscribe Messaging](#).

İlgili bilgiler

[Uygulamaların geliştirilmesi](#)

[WebSphere MQ iletileri](#)

IBM WebSphere MQ yayınlama/abone olma mesajlarına giriş

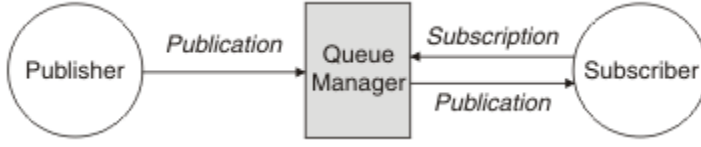
Yayınlama/abone olma ileti sistemi, bilgi sağlayıcısını bu bilgilerin tüketicilerinden çözümlenize olanak sağlar. Gönderme uygulaması ve alma uygulamasının, gönderilecek ve alınacak bilgiler için birbiriyle ilgili herhangi bir bilgi sahibi olması gerekmez.

Bir noktadan noktaya IBM WebSphere MQ uygulaması başka bir uygulamaya ileti göndermeden önce, bu uygulama hakkında bir bilgi sahibi olması gerekir. Örneğin, bilgilerin gönderileceği kuyruğun adını bilmesi ve kuyruk yöneticisi adını da belirtmesi gerekir.

IBM WebSphere MQ yayınlama/abone olma, uygulamanızın hedef uygulama hakkında herhangi bir bilgi sahibi olması gereksinmesini kaldırır. Tüm gönderme uygulamasının yapması gerekir, istediği bilgileri içeren bir IBM WebSphere MQ iletisi koyar ve bilgi konusunu belirten bir konu atayabilir ve IBM

WebSphere MQ ' in bu bilgilerin dağıtımını işlemesine izin vermeleri gerekir. Benzer şekilde, hedef uygulamanın aldığı bilgilerin kaynağı hakkında bilgi sahibi olmak zorunda değildir.

Şekil 3 sayfa 20 , en basit yayınlama/abone olma sistemini gösterir. Bir yayınlayıcı, bir kuyruk yöneticisi ve bir abone var. Aboneden kuyruk yöneticisine bir abonelik gönderilir, yayıncıdan kuyruk yöneticisine bir yayın gönderilir ve daha sonra, yayın kuyruk yöneticisi tarafından aboneye iletilir.



Şekil 3. Basit yayınlama/abone olma yapılandırması

Tipik bir yayınlama/abone olma sisteminde birden çok yayınlayıcı ve birden çok abone ve çoğu zaman birden çok kuyruk yöneticisi vardır. Bir uygulama hem yayınlayıcı hem de bir abone olabilir.

Yayınlama/abone olma bileşenlerine genel bakış

Yayınlama/abone olma, abonelerin yayıncılardan ileti biçiminde, bilgi alabilecekleri mekanizmadır. Yayıncılar ve aboneler arasındaki etkileşimler, standart WebSphere MQ olanaklarını kullanarak kuyruk yöneticilerine göre denetlenir.

Tipik bir yayınlama/abone olma sisteminde birden çok yayınlayıcı ve birden çok abone ve çoğu zaman birden çok kuyruk yöneticisi vardır. Bir uygulama hem yayınlayıcı hem de bir abone olabilir.

Bilgi sağlayıcıya *yayınlayıcı* adı verilir. Yayıncılar, bu bilgilerle ilgilenen uygulamalar hakkında herhangi bir bilgi sahibi olmamaya gerek kalmadan bir konu hakkında bilgi sağlar. Yayıncılar bu bilgileri, bu iletilerin konusunu yayınlamak ve tanımlamak istedikleri *yayımlar* adı verilen iletiler biçiminde oluşturur.

Bilgilerin tüketicisi bir *abone* olarak adlandırılır. Aboneler, abonenin ilgilendiği konuyu açıklayan *abonelikler* oluşturur. Bu nedenle, abonelik, aboneye hangi yayınların iletileceğini belirler. Aboneler birden çok aboneliği gerçekleştirebilirler ve birçok farklı yayıncıdan bilgi alabilirler.

Yayımlanan bilgiler bir WebSphere MQ iletisinde gönderilir ve bilgiler konusu *konu* ile tanımlanır. Yayıncı, bilgileri yayımlarken konuyu belirtir ve abone, yayımları almak istediği konuları belirtir. Aboneye, yalnızca abone olduğu konular hakkında bilgi gönderilir.

Bu, her iletiye noktadan noktaya ileti sisteminde gerekli olduğu şekilde, her iletiye belirli bir hedef dahil etme gereksinimini kaldırarak, sağlayıcıların ve tüketicilerinin yayınlama/abone olma mesajlarında ayrılmasına izin veren konuların varoluşunu sağlar.

Yayıncılar ve aboneler arasındaki etkileşimler bir kuyruk yöneticisi tarafından denetlenir. Kuyruk yöneticisi yayıncılardan ileti alır ve abonelerden abonelikler (bir dizi konu aralığıyla). Kuyruk yöneticisinin işi, yayımlanan iletileri, iletilerin konusuna ilgi kaydetmiş olan abonelere yöneltmek.

İletileri dağıtmak için standart WebSphere MQ olanakları kullanılır; böylece, uygulamalarınız var olan WebSphere MQ uygulamaları için kullanılabilir olan tüm özellikleri kullanabilir. Bu, yalnızca bir kez garantili teslim almak için kalıcı iletileri kullanabileceğiniz ve iletilerin, yalnızca yayınlayıcı tarafından kesinleştirildikleri durumlarda aboneye teslim edildiğinden emin olmak için iletilerin işlem biriminin bir parçası olabileceğinin anlamına gelir.

Tek bir kuyruk yöneticisi yayınlaması/abone olma yapılandırması örneği

Şekil 4 sayfa 21 , temel tek bir kuyruk yöneticisi yayınlama/abone olma yapılandırmasını gösterir. Bu örnek, yayıncılardan birkaç konu hakkında bilgilerin bulunduğu bir haber hizmetine ilişkin yapılandırmayı gösterir:

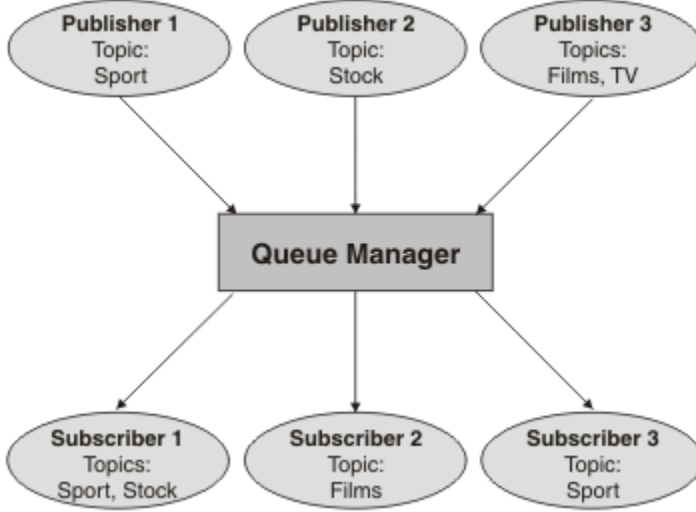
- Yayıncı 1 Spor konusunu kullanarak spor sonuçlarıyla ilgili bilgileri yayınlıyor
- Yayıncı 2, hisse senedi fiyatlarıyla ilgili bilgileri Stoğun bir konusu kullanarak yayınlıyor

- Yayıncı 3, Filmler konusu kullanılarak film eleştirileri hakkında bilgi yayınlıyor ve televizyon listeleriyle ilgili bir konuyu yayınlıyor.

Üç abone farklı konulara ilgi gösterdiler, bu nedenle kuyruk yöneticisi ilgilendikleri bilgileri şu bilgileri gönderir:

- Abone 1 spor sonuçlarını ve hisse senedi fiyatlarını alır
- Abone 2, film incelemelerini alır
- Abone 3, spor sonuçlarını alır

Abonelerin hiçbiri televizyon listelerine ilgi göstermedi, bu yüzden bunlar dağıtılmadı.



Şekil 4. Tek kuyruk yöneticisi yayınlama/abone olma örneği

Yayıncılar ve yayınlar

WebSphere MQ yayınlama/abone olma, belirli bir konu hakkında bilgi veren bir uygulamadır. Bu uygulama, bir yayınlama adı verilen standart WebSphere MQ iletisi biçiminde, kuyruk yöneticisi tarafından kullanılabilir. Bir yayınlama, birden fazla konu hakkında bilgi yayınlatabilir.

Yayıncılar, daha önce açılmış bir konuya ileti koymak için MQPUT komutunu kullanır, bu ileti bir yayındır. Daha sonra, yerel kuyruk yöneticisi yayını, yayın konusu olan abonelere sahip olan abonelere yönlendirir. Yayınlanan bir ileti birden çok abone tarafından tüketilebilir.

Bir kuyruk yöneticisi, yayınları uygun aboneliklere sahip tüm yerel abonelere dağıtmanın yanı sıra, bu yayını doğrudan ya da abone olan bir kuyruk yöneticisi ağı aracılığıyla bu konuya bağlı olan diğer kuyruk yöneticilerine de dağıtabilir.

Bir WebSphere MQ yayınlama/abone olma ağlarında, bir yayınlama uygulaması da aboneli olabilir.

Syncpoint altındaki yayınlar

Yayıncılar, bir iş birimindeki abonelere teslim edilen tüm iletileri içermek için, MQPUT ya da MQPUT1 çağrılarını syncpoint 'te yayınlatabilir. MQPMO_RETAIN seçeneği ya da ALL ya da ALLDUR değerleri içeren NPMGDLV ya da PMSGDLV konu teslim seçenekleri belirtilirse, kuyruk yöneticisi, yayınlama MQPUT ya da MQPUT1 çağrısı kapsamında, syncpoint 'te iç MQPUT ya da MQPUT1 çağrılarını kullanır.

Durum ve olay bilgileri

Yayınlama, hisse senetlerinin geçerli fiyatı gibi devlet yayınları ya da söz konusu stoktaki bir ticaret gibi etkinlik yayınları kategorilendirilebilir.

Durum yayınları

Durum yayınları , hisse senedi fiyatı ya da bir futbol karşılaşmasında geçerli puan gibi bir şeyin geçerli durumuna ilişkin bilgileri içerir. Bir şey olduğunda (örneğin, hisse senedi fiyatı değişiklikleri ya da futbol puanı değişirse), önceki durum bilgileri artık gerekli değildir, çünkü bu bilgiler yeni bilgiler tarafından yerine konmaktadır.

Bir abone başlatıldığında durum bilgilerinin geçerli sürümünü almak ve durum her değiştiğinde yeni bilgiler göndermek isteyecektir.

Bir yayında durum bilgisi varsa, bu yayın genellikle alıkonan bir yayın olarak yayınlanır. Yeni bir abone tipik olarak yürürlükteki durum bilgilerinin hemen olmasını ister; abonenin bilgilerin yeniden yayınlanmasına neden olan bir olayın beklenmesini beklemeyiz. Abone MQSO_PUBLICATIONS_ON_REQUEST ya da MQSO_NEW_PATICATIONS_ONLY seçeneklerini kullanmadığı sürece abone, abone olduğunda otomatik olarak bir konunun alıkonduğu yayınını alır.

Olay yayınları

Olay yayınları , bir hisse senedi içindeki ticaret ya da belirli bir hedefin puanlaması gibi, ortaya çıkan her bir olaya ilişkin bilgileri içerir. Her olay, diğer olaylardan bağımsızdır.

Bir abone, olaylar hakkında bilgi almak isteyecektir.

Alıkonan yayınlar

Varsayılan olarak, ilgili tüm abonelere bir yayın gönderildikten sonra atılır. Ancak, bir yayıncı, bir yayının kopyasının, konuya ilgi duyan gelecekteki abonelere gönderilebilmesi için alıkonabileceğini belirtebilir.

Tüm ilgili abonelere gönderildikten sonra yayınların silinmesi, etkinlik bilgileri için uygundur, ancak her zaman durum bilgileri için uygun değildir. Bir iletiyi koruyarak, yeni abonelerin ilk durum bilgilerini almadan önce bilgilerin yeniden yayınlanabilmesi için beklemek zorunda kalmazlar. Örneğin, hisse senedi aboneliğine aboneliği olan bir abone, hisse senedi fiyatının değişmesini beklemeyiz (ve dolayısıyla yeniden yayınlanmak üzere) geçerli fiyatı doğrudan alır.

Kuyruk yöneticisi her konu için yalnızca bir yayını tutabilir; böylece, kuyruk yöneticisinde yeni bir alıkonan yayın geldiğinde, bir konuya ilişkin var olan tutulan yayın yayını silinir. Ancak, var olan yayının silinmesi, yeni alıkonan yayının gelmesiyle zamanuyumlu olarak gerçekleşmeyebilir. Bu nedenle, mümkün olan her yerde, herhangi bir konuyla ilgili olarak saklanan yayınlardan birden fazla yayınlayıcı gönderemez.

Aboneler, MQSO_NEW_YAYINICATIONS_ONLY abonelik seçeneğini kullanarak alıkonan yayınlar almak istemediklerini belirtebilir. Var olan aboneler, alıkonan yayınların kopyalarını çoğaltmak için kendilerine bilgi isteyebilir.

Durum bilgileri için bile, yayınları tutmak istemeyebileceğinizde bazı zamanlar vardır:

- Bir konuya ilişkin tüm abonelikler o konuda herhangi bir yayın yapılmadan önce yapılıyorsa ve bu konuda herhangi bir yayın yapılmadan önce, yeni aboneliklere izin vermezseniz ya da izin vermezseniz, bu yayınlar ilk yayınlandıklarında tüm aboneler kümesine teslim edildiğinden, yayınların alıkonmasına gerek yoktur.
- Yayınların her saniye gibi sık ortaya çıkması durumunda, yeni bir abone (ya da bir hatadan kurtarma bir abone), ilk aboneliklerinden hemen sonra geçerli durumu alır; bu nedenle, bu yayınları alıkoymaya gerek yoktur.
- Yayınlar büyükse, her konu için alıkonan yayını saklamak için çok miktarda depolama alanına gereksinim duyabilirsiniz. Birden çok kuyruk yöneticisi ortamında, tutulan yayınlar, eşleşen bir aboneliğe sahip olan ağdaki tüm kuyruk yöneticileri tarafından saklanır.

Tutulan yayınların kullanımına karar verilirken, uygulamaların bir hatadan nasıl kurtarılacağını göz önünde bulundurun. Yayınlayıcı saklan yayınlarını kullanmıyorsa, abone uygulamasının yürürlükteki durumunu yerel olarak saklamaya gerek olabilir.

Bir yayının alıkonmasını sağlamak için, MQPMO_RETAIN put-message seçeneğini kullanın. Bu seçenek kullanılırsa ve yayın saklanamazsa, ileti yayınlanmaz ve çağrı MQRC_PUT_NOT_ALIMARM ile başarısız olur.

Bir ileti alıkonan bir yayınsa, bu ileti MQIsRetained ileti özelliği ile gösterilir. Bir iletinin devamlılığı, ilk olarak yayınlandığı sırada olduğu gibidir.

Syncpoint altındaki yayınlar

IBM WebSphere MQ publish/subscede içinde, syncpoint yayıncılar tarafından ya da kuyruk yöneticisi tarafından dahili olarak kullanılabilir.

Yayıncılar, MQPMO_SYNCPOINT seçeneğiyle MQPUT/MQPUT1 çağrılarını yayınladığında syncpoint 'i kullanır. Abonelere teslim edilen tüm iletiler, work.The MAXUMSGS kuyruk yöneticisi özneteliğinde, kesinleştirilmemiş ileti sayısı üst sınırına kadar sayılır. Bu sınır, bu sınırı belirler. Sınıra ulaşırsa, yayıncı 2024 (07E8) (RC2024): MQRC_SYNCPOINT_LIMIT_UVARD neden kodlarına ulaşır.

Bir yayıncı, MQPMO_RETAIN seçeneğiyle MQPMO_NO_SYNCPOINT işlevini ya da ALL ya da ALLDR değerleri içeren NPMGDLV/PMSGDLV konu teslim seçeneklerini kullanarak MQPUT/MQPUT1 ' i çağırıldığında, kuyruk yöneticisi iletilerin istenildiği gibi teslim edildiğini garanti etmek için iç eşitleme noktaları kullanır. Yayıncı MQPUT/MQPUT1 çağrısı kapsamında sınırı ulaşırsa, yayıncı 2024 (07E8) (RC2024): MQRC_SYNCPOINT_LIMIT_UVARD neden kodunu alabilir.

Aboneler ve abonelikler

In WebSphere MQ publish/subscribe, a subscriber is an application that requests information about a specific topic from a queue manager in a publish/subscribe network. Bir abone, aynı ya da farklı konular hakkında, birden çok yayıncıdan ileti alabilir.

Abonelikler, bir MQSC komutu ya da uygulamalar kullanılarak el ile yaratılabilir. Bu abonelikler yerel kuyruk yöneticisine verilir ve abonenin almak istediği yayınlarla ilgili bilgileri içerir:

- Abonenin ilgilendiği konu; genel arama karakterlerinin kullanılması durumunda birden çok konuya çözümlenebilir.
- Yayınlanan iletilere uygulanacak isteğe bağlı bir seçim dizgisi.
- Seçilen yayınların yerleştirileceği ve isteğe bağlı CorrelIdolan bir kuyruğa ilişkin tanıtıcı (*abone kuyruğu* olarak bilinir).

Yerel kuyruk yöneticisi, abonelik bilgilerini saklar ve bir yayın aldığı anda, yayının konu ve seçim dizgisiyle eşleşen bir abonelik olup olmadığını saptamak için bilgileri tarar. Eşleşen her abonelik için, kuyruk yöneticisi yayını abonenin abone kuyruğuna yönlendirir. Bir kuyruk yöneticisi abonelikleri hakkında sakladığı bilgiler, DIS SUB ve DIS SBSTATUS komutları kullanılarak görüntülenebilir.

Bir abonelik, yalnızca aşağıdaki olaylardan biri gerçekleştiğinde silinir:

- MQCLOSE çağrısını kullanarak abone olmayan abone (abonelik kalıcı olmayan bir şekilde yapıldıysa).
- Aboneliğin süresi dolacak.
- Abonelik, DELETE SUB komutunu kullanarak sistem denetimcisi tarafından silinir.
- Abone uygulaması sona erer (abonelik sürgüsüz olarak yapıldıysa).
- Kuyruk yöneticisi durdurulmuş ya da yeniden başlatıldıysa (abonelik durdurulamaz bir şekilde yapıldıysa).

İletiler alınırken, MQGET çağrısında uygun seçenekleri kullanın. If your application processes only messages for one subscription then, as a minimum, you should use `get-by-correlid`, as demonstrated in the C sample program `amqssbxa.c` and at [yönetilmeyen MQ abonesi](#). Kullanılacak **CorrelId**, MQSD 'deki MQSUB' dan döndürülür. **SubCorrelId** alanı.

Yönetilen kuyruklar ve yayınlama/abone olma

Bir abonelik oluşturduğunuzda, yönetilen kuyruğa alma seçeneğini kullanmayı seçebilirsiniz. Yönetilen kuyruğa alma, abonelik yarattığınızda otomatik olarak bir abonelik kuyruğu oluşturulabilir. Yönetilen kuyruklar, aboneliğin dayanaklına uygun olarak otomatik olarak döşenir. Yönetilen kuyrukların kullanılması, yayınları almak için kuyruklar yaratma konusunda endişelenmenize gerek olmadığı ve sürekli olmayan bir abonelik bağlantısı kapatılmışsa, tüketilmeyen yayınlar otomatik olarak abone kuyruklarından kaldırılır.

If an application has no need to use a particular queue as its subscriber queue, the destination for the publications it receives, it can make use of the *yönetilen abonelikler* using the MQSO_MANAGED subscription option. Yönetilen bir abonelik yaratırsanız, kuyruk yöneticisi, kuyruk yöneticisinin yayınların alınacağı yerde yarattığı bir abone kuyruğuna ilişkin aboneye bir nesne tanıtıcısı döndürür. Kuyruğa göz atmanıza, kuyruğa girmenize ya da kuyruğa sormanıza olanak tanıyan kuyruğun nesne tanıtıcısı döndürülür (geçici dinamik kuyruklara belirtik olarak erişim verilmediyse, yönetilen bir kuyruğun özniteliklerini koymak ya da ayarlamak olanaklı değildir).

Aboneliğin dayanıklılığı, kuyruk yöneticisine abone olunan uygulamanın bağlantısı kesildiğinde yönetilen kuyruğun mı kalacağını belirler.

Yönetilen abonelikler özellikle, kalıcı olmayan aboneliklerle kullanıldığında yararlı olur; çünkü uygulamanın bağlantısı sona erdiğinde, tüketilmeyen iletiler, kuyruk yöneticinizde süresiz olarak yer alan abone kuyruğunda başka bir şekilde kalır. Yönetilen bir abonelik kullanıyorsanız, yönetilen kuyruk geçici bir dinamik kuyruk olur ve aşağıdaki nedenlerden herhangi biri için bağlantı kesildiğinde, tüketilmeyen iletilerle birlikte silinecektir.

- MQCO_REMOVE_SUB ile MQCLOSE kullanılır ve yönetilen Hobj kapatılır.
- Kalıcı olmayan bir abonelik (MQSO_NON_DAYANIKLI) kullanılarak bir uygulama bağlantısı kaybedilir.
- Abonelik süresi dolduğu ve yönetilen Hobj kapatıldığı için bir abonelik kaldırıldı.

Yönetilen abonelikler, sürekli abonelikler ile de kullanılabilir, ancak bağlantı yeniden açıldığında bunların alınabilmesi için, tüketilmeyen iletileri abone kuyruğunda bırakmak isteyebilirsiniz. Bu nedenle, dayanıklı abonelikler için yönetilen kuyruklar kalıcı bir dinamik kuyruk şeklini alır ve abone olan uygulamanın kuyruk yöneticisine bağlantısı kesildiğinde kalır.

Kalıcı dinamik yönetilen kuyruk kullanmak istiyorsanız, aboneliğiniz için süre bitimi ayarlayabilirsiniz; böylece, bağlantı kesildikten sonra kuyruk hala var olmaya devam eder, süresiz olarak var olmaya devam etmeyecektir.

Yönetilen kuyruğu silerseniz, bir hata iletisi alırsınız.

Yaratılan yönetilen kuyruklar, her birinin benzersiz olması için, sonundaki sayılarla (zaman damgaları) adlandırılır.

Abonelik dayanıklılığı

Abonelikler, dayanıklı ya da dayanıklı olmayan bir şekilde yapılandırılabilir. Abonelik dayanıklılığı, uygulamaların kuyruk yöneticisinden bağlantılarını kestiğinde aboneliklere ne olacağını belirler.

Sürekli abonelikler

Abone olunan bir uygulamanın kuyruk yöneticisine yönelik bir bağlantısı kapatıldığında, sürekli abonelikler var olmaya devam eder. Bir abonelik dayanıklıysa, abone olma uygulaması bağlantısını kestiğinde abonelik yerinde kalır ve abonelik, abonelik yaratıldığında döndürülen SubName 'ı (SubName) kullanarak aboneliği yeniden bağlarken abone olunan uygulama tarafından kullanılabilir.

Süreye abone olduğunda, bir abonelik adı (SubName) gereklidir. Abonelik tanımlamak için abonelik adları, bir kuyruk yöneticisi içinde benzersiz olmalıdır. Böylece, abonelik adları kullanılabilir. This means of identification is necessary when specifying a subscription you want to resume, if you have either deliberately closed the handle to the subscription (using the MQCO_KEEP_SUB option) or have been disconnected from the queue manager. MQSO_RESUME seçeneği ile MQSUB çağrısını kullanarak var olan bir aboneliği sürdürebilirsiniz. DISPLAY SBSTATUS komutunu SUBTYPE ALL ya da ADMIN ile birlikte kullanırsanız, abonelik adları da görüntülenir.

Bir uygulama artık dayanıklı bir abonelik gerektirmediği zaman, MQCO_REMOVE_SUB seçeneğiyle MQCLOSE işlev çağrısı kullanılarak kaldırılabilir ya da MQSC komutu DELETE SUB kullanılarak el ile silinebilir.

Bir konuya dayanıklı aboneliklerin yapılabilmesi için **DURSUB** konu özneliği kullanılarak kontrol edilebilir.

MQSO_RESUME seçeneğini kullanarak bir MQSUB çağrısından dönüşte, abonelik süre bitimi aboneliğin özgün süre bitimi olarak ayarlanır ve kalan süre bitimi olmaz.

Bir kuyruk yöneticisi, abone uygulaması bağlı olmasa da, kalıcı bir aboneliği karşılamak için yayınları göndermeye devam eder. Bu, abone kuyruğunda iletilerin bir oluşturmasına neden olur. Bu sorunu önlemenin en kolay yolu, uygun her yerde dayanıklı olmayan bir abonelik kullanmandır. However, where it is necessary to use durable subscriptions, a build up of messages can be avoided if the subscriber subscribes using the Alıkonan yayınlar option. Bir abone, MQSUBRQ çağrısını kullanarak yayınları aldığında denetleyebilirler.

Kalıcı olmayan abonelikler

Kalıcı olmayan abonelikler yalnızca, abone olunan uygulamanın kuyruk yöneticisine bağlantısı açık kaldığı sürece var olur. Abone olunan uygulama kuyruk yöneticisinden kasıtlı olarak ya da bağlantı kaybindan kesildiğinde abonelik kaldırılır. Bağlantı kapatıldığında, abonelikle ilgili bilgiler kuyruk yöneticisinden kaldırılır ve DISPLAY SBSTATUS komutunu kullanarak abonelikleri görüntülediğinizde gösterilmez. Abone kuyruğuna başka ileti konmaz.

Kalıcı olmayan abonelikler için abone kuyruğunda tüketilmeyen yayınlara ne olursa olsun, aşağıdaki gibi belirlenir.

- Abone olunan bir uygulama yönetilen hedef kullanıyorsa, tüketilmeyen yayınlar otomatik olarak kaldırılır.
- Abone olunan uygulama, abone olduğunda kendi abone kuyruğuna bir tanıtıcı sağlıyorsa, tüketilmeyen iletiler otomatik olarak kaldırılmaz. Uygunsa, kuyruğun temizlenmesi uygulamanın sorumluluğundadır. Kuyruk birden çok abone ya da noktadan noktaya iletişim uygulamaları tarafından paylaşılıyorsa, kuyruğu tamamen temizlemek uygun olmayabilir.

Kalıcı olmayan abonelikler için gerekme de, varsa bir abonelik adı kuyruk yöneticisi tarafından kullanılır. Abonelik tanımlamak için abonelik adları, kuyruk yöneticisi içinde benzersiz olmalıdır; böylece, abonelik adı kullanılabilir.

Seçim dizgileri

Seçim dizgisi , bir yayınlı eşleşip eşleşmediğini belirlemek için yayına uygulanan bir ifadedir. Seçim dizeleri genel arama karakterleri içerebilir.

Abone olduğunuzda, bir konu belirtmenin yanı sıra, bu yayınları ileti özelliklerine göre seçmek için bir seçim dizgisi belirtebilirsiniz.

Konular

Konu, yayınlama/abone olma iletisinde yayınlanan bilgilerin konusu.

Noktadan noktaya iletişim sistemlerindeki iletiler belirli bir hedef adrese gönderilir. Konuya dayalı yayınlama/abone olma sistemlerindeki iletiler, iletinin içeriğini açıklayan konuya dayalı olarak abonelere gönderilir. İçeriğe dayalı sistemlerde, iletiler iletinin içeriğini temel alan abonelere gönderilir.

IBM WebSphere MQ yayınlama/abone olma sistemi, konuya dayalı bir yayınlama/abone olma sistemidir. Yayınlayıcı bir ileti oluşturur ve bunu, yayının konusuna en uygun olan bir konu dizesiyle yayınlar. Yayınları almak için, bir abone, yayın konularını seçmek için bir kalıp eşleştirmesi konu dizgisiyle bir abonelik oluşturur. Kuyruk yöneticisi, yayınlara, yayın konularıyla eşleşen abonelikleri olan ve yayınları alma yetkisine sahip abonelere teslim eder. “Konu dizgileri” sayfa 26 makalesi, bir yayının konusunu tanımlayan konu dizgilerinin sözdizimini açıklar. Aboneler, hangi konuların alınacağı seçmek için konu dizgileri de oluşturur. Abone yaratılacak konu dizgileri, yayınlardaki konu dizgileriyle eşleşen örüntü eşleştirmesi için iki alternatif genel arama şemasını içerebilir. Kalıp eşleştirme, “Joker şemalar” sayfa 27 içinde açıklanmıştır.

Konu tabanlı yayınlama/abone olma, yayıncılar ya da denetimciler, konuları konu başlıklarına sınıflandırmaktan sorumludur. Genellikle konular, konu dizgisinde alt konular yaratmak için ' / ' karakteri kullanılarak, konu ağaçlarına sıradüzensel olarak sıralanır. Konu ağaçlarına ilişkin örnekler için bkz. “Konu ağaçları” sayfa 33 . Konular, konu ağacındaki düğümlerdir. Konular, başka alt konuları olmayan ya da alt konuları olan ara düğümler olan yaprak düğümler olabilir.

Konuları sıradüzenli bir konu ağacına düzenlemeyle paralel olarak, konuları yönetici konu nesnelileriyle ilişkilendirebilirsiniz. Bir konuya, konunun bir yönetim konusu nesnesiyle ilişkilendirilerek, konunun

bir kümede dağıtılıp dağıtılmayacağı gibi, bir konuya öznitelik atayabilirsiniz. İlişkilendirme, yönetimle ilgili konu nesnesinin TOPICSTR özniteliğini kullanarak konuyu adlandırılarak yapılır. Bir yönetim konusu nesnesini belirttik olarak bir konu ile ilişkilendirmezseniz, konu, bir denetim konusu nesnesiyle ilişkilendirdiğiniz *sahip* konu ağacındaki en yakın atasının özniteliklerini devralır. Herhangi bir üst başlık tanımlamadıysanız, bu başlık SYSTEM. BASE. TOPIC. Yönetimle ilgili konu nesnelere “Yönetimle ilgili konu nesnelere” sayfa 35’inde açıklanmıştır.

Not: Bir konunun tüm özniteliklerini SYSTEM. BASE. TOPIC, doğrudan SYSTEM. BASE. TOPIC. For example, in the topic space of US states, USA/Alabama USA/Alaska, and so on, USA is the root topic. Kök konunun ana amacı, yanlış aboneliklerle eşleşen yayınlardan kaçınmak için çakışmayan, çakışmayan konu alanları oluşturmaktır. Ayrıca, tüm konu alanınızı etkileyecek şekilde, kök konunun özniteliklerini değiştirebilirsiniz. Örneğin, **CLUSTER** özniteliğinin adını ayarlayabilirsiniz.

Bir konuyu yayınlıyıcı ya da abone olarak belirttiğinizde, bir konu nesnesine gönderme yapan, bir konu dizgisi belirtme seçeneğiniz var ya da her ikisini de yapabilirsiniz. Bu durumda, kaynağınız, konu nesnesinin bir alt konusunu tanımlamanız için konu dizilimini içerir. Kuyruk yöneticisi konuyu, konu nesnesinde belirtilen konu dizgisi önekinde ekleyerek, iki konu dizgisinin arasına ek bir ' / ' ekleme (örneğin, *konu dizilimi/nesne dizgisi*) ile konuyu tanıtır. “Konu dizgileri birleştirilmesi” sayfa 31 , bunu daha ayrıntılı olarak açıklar. Sonuçta ortaya çıkan konu dizesi, konuyu tanımlamak ve bunu bir yönetici konu nesnesiyle ilişkilendirmek için kullanılır. Denetim konusu nesnesi, ana konuya karşılık gelen konu nesnesiyle aynı konu nesnesiyle aynı olmak zorunda değildir.

İçerik tabanlı yayınlama/abone olma içinde, almak istediğiniz iletileri, her iletinin içeriğini arayan seçim dizgileri sağlayarak tanımladınız. WebSphere MQ , iletinin tüm içeriğini değil, ileti özelliklerini tarayan ileti seçicileri kullanarak bir ara içerik tabanlı içerik tabanlı yayınlama/abone olma özelliğini sağlar; bkz. Seçiciler. İleti seçicilerinin oksal kullanımı bir konuya abone olmak ve sonra seçimi sayısal bir özelliklikle nitelendirmek. Seçici, yalnızca belirli bir aralıktaki değerlerle ilgilendiğinizi belirtmenizi sağlar; karakter ya da konu tabanlı joker karakter kullanarak yapmadığınız bir şey. İletinin tam içeriğine dayalı olarak süzgeç uygulamaya gerek duyarsanız, WebSphere Message Broker olanağını kullanmanız gerekir.

Konu dizgileri

Bir konu dizesini kullanarak konu olarak yayınladığınız etiket bilgileri. Karakter ya da konu tabanlı genel arama karakteri konu dizgilerini kullanarak konu gruplarına abone olun.

Konular

Konu dizesi bir yayınlama/abone olma iletisinin konusunu tanımlayan bir karakter dizisidir. Konu dizesini oluştururken beğendiğiniz karakterleri kullanabilirsiniz.



Sürüm 7 yayınlama/abone olma özelinde üç karakterin özel anlamı vardır. Bunlar, bir konu dizisinde herhangi bir yere izin verilir, ancak bunları dikkatli bir şekilde kullanın. Özel karakterlerin kullanımı “Konu tabanlı genel arama karakteri şeması” sayfa 27’inde açıklanmıştır.

Eğik çizgi (/)

Konu düzeyi ayırıcısı. Konuyu bir konu ağacına yapılandırmak için ' / ' karakterini kullanın.

Avoid empty topic levels, ' / / ', if you can. Bu, konu sıradüzenindeki konu dizisindeki düğümlere karşılık gelir. Bir konu dizisindeki baştaki ya da sondaki ' / ' , baştaki ya da sondaki boş bir düğüme karşılık gelir ve önlenmelidir.

Hash işareti (#)

Aboneliklerde çok düzeyli bir joker karakter oluşturmak için ' / ' ile birlikte kullanılır. Yayınlanan konuları adlamak için kullanılan konu dizilerinde ' / ' bitişiğindeki '# ' komutunu kullanarak dikkatli olun. “Konu dizgileri örnekleri” sayfa 27 '# ' in mantıklı bir kullanımını gösterir.

The strings '.../#/...', '#/...' and '.../#' have a special meaning in subscription topic strings. Dizgiler, konu sıradüzenindeki bir ya da daha fazla düzeydeki tüm konuları eşleştirir. Bu nedenle, bu sıralardan biriyle bir konu oluşturduysa, konu sıradüzeninde birden çok düzeyde bulunan tüm konulara da abone olmadan, bu sıralara abone olamazsınız.

Artı işareti (+)

Aboneliklerde tek düzeyli joker karakter oluşturmak için '/' ile birlikte kullanılır. Yayınlanan konuları adlamak için kullanılan konu dizgilerinde '/' bitişiğindeki '+' komutunu kullanarak dikkatli olun.

The strings '.../+/...', '+/...' and '.../+' have a special meaning in subscription topic strings. Dizgiler, konu sıradüzeninde bir düzey altındaki tüm konuları eşleştirir. Bu nedenle, bu sıralardan biriyle bir konu oluşturursanız, konu sıradüzenindeki bir düzeydeki tüm konulara da abone olmadan, bu diziyeye abone olamazsınız.

Konu dizgileri örnekleri

```
IBM/Business Area#/Results
IBM/Diversity/%African American
```

Joker şemalar

Birden çok konuya abone olmak için kullanılan iki genel arama tipi şeması vardır. Şema seçimi bir abonelik seçeneğidir.

MQSO_WILDCARD_KONUSU

Konu tabanlı genel arama şemasını kullanarak abone olmak için konuları seçin.

Genel arama karakteri belirttik olarak seçilmezse, bu varsayılan değer olarak varsayılan değer olur.

MQSO_WILDCARD_CHAR

Karakter tabanlı genel arama şemasını kullanmak için abone olunması gereken konuları seçin.

DEFINE SUB komutunda **wschema** parametresini belirterek şemayı ayarlayın. Ek bilgi için bkz. [DEFINE ALT](#).

Not: Subscriptions that were created before WebSphere MQ Version 7.0 always use the character-based wildcard scheme.

Örnekler

```
IBM+/Results
#/Results
IBM/Software/Results
IBM/*ware/Results
```

Konu tabanlı genel arama karakteri şeması

Konu tabanlı genel arama karakterleri, abonelerin bir kerede birden çok konuya abone olmalarına olanak sağlar.

Topic-based wildcards are a powerful feature of the topic system in WebSphere MQ publish/subscribe. Abonelikler için çok düzeyli genel arama karakteri ve tek düzeyli genel arama karakteri kullanılabilir, ancak bir iletinin yayıncısı tarafından bir konu içinde kullanılamaz.

Konu tabanlı genel arama şeması, konu düzeyine göre gruplanmış yayınları seçmenizi sağlar. *Konu sıradüzenindeki her düzeyi* için, söz konusu konu düzeyindeki aboneliğin, yayındaki dizgiyle tam olarak eşleşmesi gerekip gerekmediği içinchooseher düzey düzeyi için seçim yapabilirsiniz. Örneğin, abonelik IBM+/Results tüm konuları seçer.

```
IBM/Software/Results
IBM/Services/Results
IBM/Hardware/Results
```

İki tip genel arama karakteri vardır.

Çok düzeyli genel arama karakteri

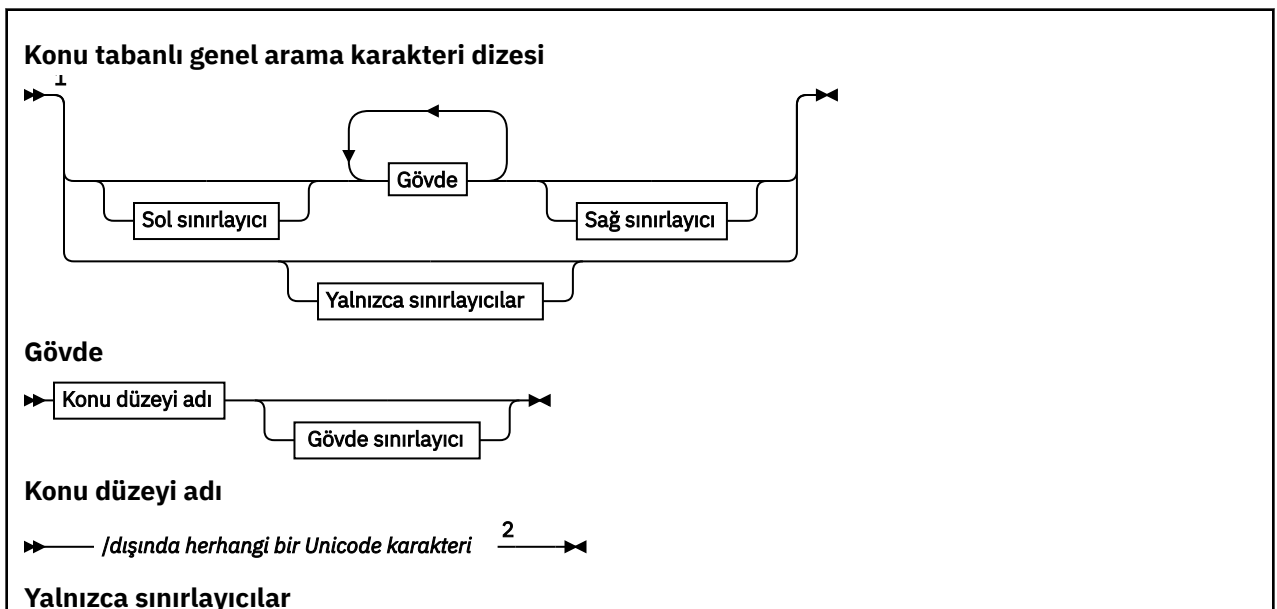
- Çok düzeyli genel arama karakteri aboneliklerde kullanılır. Bir yayında kullanıldığında, hazır bilgi olarak kabul edilir.
- The multilevel wildcard character '#' is used to match any number of levels within a topic. For example, using the example topic tree, if you subscribe to 'USA/Alaska/#', you receive messages on topics 'USA/Alaska' and 'USA/Alaska/Juneau'.
- Çok düzeyli genel arama karakteri sıfır ya da daha fazla düzeyi gösterebilir. Bu nedenle, 'USA/#' tekil 'USA' ile de eşleşebilir; burada '#' , sıfır düzeylerini gösterir. Aynı bir düzey olmadığı için, konu düzeyi ayırıcısı bu bağlamda anlamsızdır.
- Çok düzeyli genel arama karakteri, yalnızca kendi başına ya da konu düzeyi ayırıcısı karakterinin yanında belirtildiğinde etkili olur. Bu nedenle, '#' ve 'USA/#' , '#' karakterinin genel arama karakteri olarak işlem gördüğü geçerli konulardır. Ancak, 'USA#' aynı zamanda geçerli bir konu dizgisiyse, '#' karakteri genel arama karakteri olarak kabul edilmez ve özel bir anlamı yoktur. Daha fazla bilgi için bkz. [“Konu tabanlı genel arama karakterleri genel arama karakteri değilse” sayfa 30](#) .

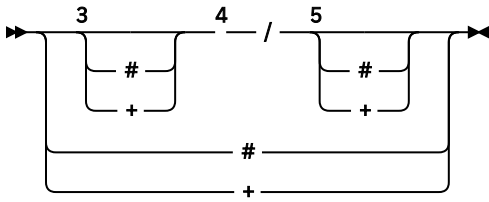
Tek düzeyli genel arama karakteri

- Aboneliklerde tek genel arama karakteri kullanılır. Bir yayında kullanıldığında, hazır bilgi olarak kabul edilir.
- Tek düzeyli genel arama karakteri '+', bir ve konu düzeyinde tek bir genel arama karakteriyle eşleşir. For example, 'USA/+' matches 'USA/Alabama', but not 'USA/Alabama/Auburn'. Tek düzeyli genel arama karakteri yalnızca tek bir düzeyle eşleştiğinden 'USA/+', 'USA' ile eşleşmez.
- Tek düzeyli joker karakter, konu ağacındaki herhangi bir düzeyde ve çok düzeyli genel arama karakteriyle birlikte kullanılabilir. Tek düzeyli genel arama karakteri, konu düzeyi ayırıcısının yanında belirtilmelidir; ancak, bu ayırıcı kendi başına belirtildiğinde belirtilmelidir. Bu nedenle, '+' ve 'USA/+', '+' karakterinin genel arama karakteri olarak işlem gördüğü geçerli konulardır. Ancak, 'USA+' aynı zamanda geçerli bir konu dizgisiyse, '+' karakteri genel arama karakteri olarak kabul edilmez ve özel bir anlamı yoktur. Daha fazla bilgi için bkz. [“Konu tabanlı genel arama karakterleri genel arama karakteri değilse” sayfa 30](#) .

Konuya dayalı genel arama karakterinin sözdiziminin çıkış karakteri yok. '#' ve '+' in genel arama karakteri olarak mı değerlendirileceğini, yoksa bunların bağlamına bağlı olup olmadığını belirleyin. Daha fazla bilgi için bkz. [“Konu tabanlı genel arama karakterleri genel arama karakteri değilse” sayfa 30](#) .

Not: Bir konu dizgisinin başı ve sonu özel bir şekilde işlem görür. Dizginin sonunu belirtmek için '\$' u kullanarak, '\$#/...' çok düzeyli bir genel arama karakteridir ve '\$/#/...' kökteki boş bir düğümdür ve çok düzeyli bir genel arama karakteridir.

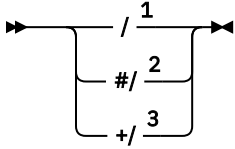




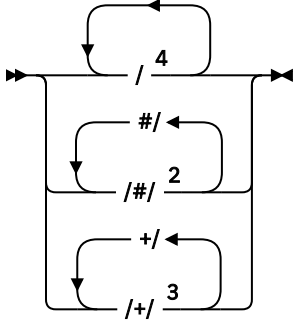
Notlar:

- 1 Boş değerli ya da sıfır uzunluklu bir konu dizgisi geçersiz
- 2 Char tabanlı ve konu tabanlı genel arama karakterleri arasında uyumluluk sağlamak için düzey adı dizgilerinde *, ?, % hiçbirini kullanmamanız önerilir.
- 3 Bu durumlar, *sol sınırlayıcı* kalıbına eşdeğerdir.
- 4 Genel arama karakteri olmayan/, tek boş bir konu ile eşleşmez.
- 5 Bu durumlar *sağ sınırlayıcı* örüntüye eşdeğerdir.
- 6 Her konuyu eşleştirin.
- 7 Sadece bir seviyenin olduğu her konuyu eşleştirin.

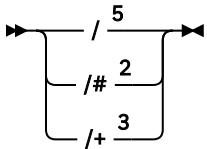
Sol sınırlayıcı



Gövde sınırlayıcı



Sağ sınırlayıcı



Notlar:

- 1 Konu dizgisi boş bir konu ile başlıyor
- 2 Sıfır ya da daha fazla düzeye eşleştirir. Çok düzeyli birden çok eşleşme dizisi, tek bir çok düzeyli eşleşme dizisiyle aynı etkiye sahiptir.
- 3 Tam olarak bir düzey ile eşleşir.
- 4 // boş bir konudur-konu dizgisi olmayan bir konu nesnesidir.
- 5 Konu dizisi boş bir konu ile bitiyor

Konu tabanlı genel arama karakterleri genel arama karakteri değilse

'+' ve '#' genel arama karakterleri, bir konu düzeyinde diğer karakterlerle (kendileri de içinde olmak üzere) karıştırıldığında özel bir anlam ifade etmiyorlardı.

Bu, '+' ya da '#' sözcüğünü içeren konuların bir konu düzeyindeki diğer karakterlerle birlikte yayınlanabileceği anlamına gelir.

Örneğin, aşağıdaki iki konuyu göz önünde bulundurun:

1. level0/level1+/level4/#
2. level0/level1/#+/level4/level#

İlk örnekte, '+' ve '#' karakterleri joker karakterler olarak ele alınır ve bu nedenle, yayınlanacak olan ancak bir abonelikte geçerli olan bir konu dizisinde geçerli değildir.

İkinci örnekte, '+' ve '#' karakterleri joker karakterler olarak işlenmez ve dolayısıyla konu dizisi hem yayınlanabilir hem de abone olunabilir.

Örnekler

```
IBM/+/Results
#/Results
IBM/Software/Results
```

Karaktere dayalı genel arama karakteri şeması

Karakter tabanlı genel arama şeması, geleneksel karakter eşleştirmesine dayalı konuları seçmenize olanak tanır.

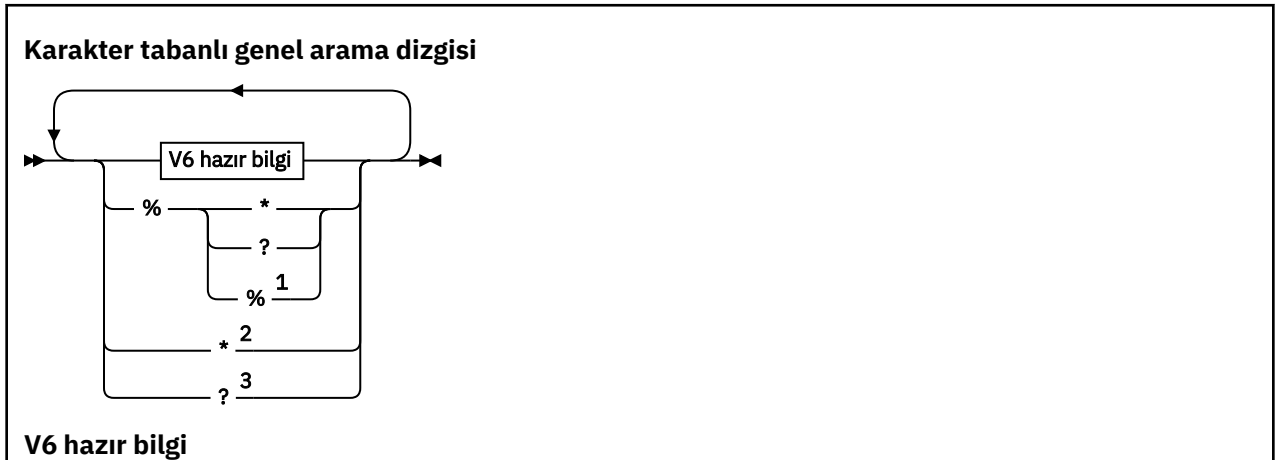
'*' dizgisini kullanarak bir konu sıradüzeninde birden çok düzeyde tüm konuları seçebilirsiniz. Karakter tabanlı genel arama şemasındaki '*' kullanılması, topickonu tabanlı genel arama karakteri dizgisinin kullanılmasının eşdeğeridir. '#'

'x/*y' is equivalent to 'x/#y' in the topic-based scheme, and selects all topics in the topic hierarchy between levels 'x' and 'y', where 'x' and 'y' are topic names that are not in the set of levels returned by the wildcard.

Konu tabanlı şemada '/+/' , karakter tabanlı şemada tam olarak eşdeğer bir değer içermiyor. 'IBM/*/Results' , 'IBM/Patents/Software/Results' ögesini de seçerdi. Yalnızca, sıradüzeninin her düzeyindeki konu adları kümesi benzersizse, her zaman aynı eşleşmeleri veren iki şemaya sahip sorgular oluşturabilir misiniz?

Karakter tabanlı şemada genel bir şekilde kullanılan '*' ve '?', konu tabanlı şemada eşdeğerleri yoktur. Konu tabanlı şema, joker karakterler kullanılarak kısmi eşleştirme gerçekleştiriyor. The character based wildcard subscription 'IBM/*ware/Results' has no topic-based equivalent.

Not: Karakter genel arama karakteri abonelikleri kullanılarak eşleşmeler, konu tabanlı abonelikler kullanılarak eşleşmelere göre daha düşüktür.



► *;dışında herhangi bir Unicode karakteri ve % ►

Notlar:

- 1 Bir hazır bilgi olarak işlem görebilmesi için "Aşağıdaki karakterden kaçış" anlamına gelir. '%' must be followed by either '*', '?' or '%'. Bkz. "Konu dizgileri örnekleri" sayfa 27.
- 2 Bir abonelikte "Sıfır ya da daha fazla karakter eşleştir" anlamına gelir.
- 3 Bir abonelikte "Tam olarak bir karakter eşleştir" anlamına gelir.

Örnekler

IBM/*/Results
IBM/*ware/Results

Konu dizgileri birleştirilmesi

Abonelikleri oluştururken ya da konuları yayınlatabilmeniz için konuları açarken, konu dizgisi iki ayrı alt konu dizesiyle ya da "alt konular"birleştirilerek oluşturulabilir. Uygulama ya da denetim komutu bir konu dizgisi olarak bir alt konu, diğeri bir konu nesnesiyle ilişkilendirilmiş konu dizgisiyle sağlanır. Alt konuyu kendi başına konu dizesi olarak kullanabilir ya da yeni bir konu adı oluşturmak için birleştirebilirsiniz.

For example, when you define a subscription using the MQSC command **DEFINE SUB**, the command can take either **TOPICSTR** (topic string) or **TOPICOBJ** (topic object) as an attribute, or both together. Yalnızca **TOPICOBJ** sağlandıysa, o konu nesnesiyle ilişkilendirilmiş konu dizgisi konu dizgisi olarak kullanılır. Yalnızca **TOPICSTR** sağlandıysa, konu dizgisi olarak kullanılır. If both are provided, they are concatenated to form a single topic string in the form **TOPICOBJ/TOPICSTR**, where the **TOPICOBJ** configured topic string is always first and the two parts of the string are always separated by a "/" character.

Similarly, in an MQI program the full topic name is created by MQOPEN. Yayınlama/abone olma MQI çağrılarında kullanılan iki alandan oluşur ve listelenen sırada:

1. **ObjectName** alanında adlandırılan konu nesnesinin **TOPICSTR** özneliği.
2. Uygulama tarafından sağlanan alt konuyu tanımlayan **ObjectString** parametresi.

Sonuçtaki konu dizgisi, **ResObjectString** parametresine döndürülür.

Her alanın ilk karakteri boş değerli ya da boş bir karakter değilse ve alan uzunluğu sıfırdan büyükse, bu alanlar var olarak kabul edilir. Alanlardan yalnızca biri varsa, bu alan konu adı olarak değiştirilmeden kullanılır. Her iki alanda bir değer yoksa, çağrı neden kodu MQRC_UNKOWN_OBJECT_NAME ile başarısız olur ya da tam konu adı geçersiz ise MQRC_TOPIC_STRING_ERROR.

Her iki alan da mevcutsa, sonuçtaki birleşik konu adının iki ögesi arasına bir "/" karakteri eklenir.

Çizelge 2 sayfa 31 , konu dizgi bitişirme örneklerini gösterir:

Konu nesnesininTOPICSTR	Uygulama ya da DEFINE SUB komutu tarafından sağlanan konu dizesi	Tam konu adı	Açıklama
Futbol/İskoçya	' '	Futbol/İskoçya	Konu nesnesinin TOPICSTR tek başına kullanılır.
' '	Futbol/İskoçya	Futbol/İskoçya	ObjectString/TOPICSTR yalnız kullanılır.
Futbol	Puanlar	Futbol/İskoçya	Bitişirme noktasına bir "/" karakteri eklenir.

Çizelge 2. Konu dizgi bitişirme örnekleri (devamı var)

Konu nesnesinin TOPICSTR	Uygulama ya da DEFINE SUB komutu tarafından sağlanan konu dizisi	Tam konu adı	Açıklama
Futbol	/İskolar	Futbol/İskoçya	İki dizgi arasında 'boş düğüm' üretildi. Bu, "Futbol/Scores" a farklıdır.
/Futbol	Puanlar	/Futbol/Scores	Konu, 'boş düğüm' ile başlar. Bu, "Futbol/Scores" a farklıdır.

The "/" character is considered as a special character, providing structure to the full topic name in "Konu ağaçları" sayfa 33. Konu ağacının yapısı etkilendiği için, "/" karakteri başka bir nedenle kullanılmamalıdır. The topic "/Football" is not the same as the topic "Football".

Not: Abonelik yaratırken bir konu nesnesi kullanırsanız, konu nesnesi konu dizgisinin değeri, tanımlama sırasında aboneliğe sabittir. Konu nesnesindeki sonraki değişiklikler, aboneliğin tanımlandığı konu dizisini etkilemez.

Konu dizgilerinde genel arama karakterleri

Aşağıdaki genel arama karakterleri özel karakterlerdir:

- artı işareti (+)
- sayı işareti (#)
- yıldız işareti (*)
- soru işareti (?)

Joker karakterler, abonelik tarafından kullanıldığında özel anlamlara sahiptir. Bu karakterler başka bir yerde kullanıldığında geçersiz olarak kabul edilmiyor, ancak bunların nasıl kullanıldığını anladığınızdan emin olmanız ve konu nesnelerini yayınlarken ya da tanımlarken bu karakterleri konu dizilimlerinizde kullanmamayı tercih etmeniz gerekir.

If you publish on a topic string with # or + mixed in with other characters (including themselves) within a topic level, the topic string can be subscribed to with either wildcard scheme.

Bir konu dizgisinde, iki / karakteri arasındaki tek karakter olarak # ya da + ile bir konu dizgisinde yayınlanırsa, MQSO_WILDCARD_TOPIC genel arama karakteri kullanılarak bir uygulama tarafından belirtik olarak bir uygulama abone olunamaz. Bu durum, uygulamanın beklenenden daha fazla yayın elde etmesi sonucunu elde eder.

Tanımlı bir konu nesnesinin konu diziliminde genel arama karakteri kullanmamalısınız. Bunu yapmazsanız, nesne bir yayıncı tarafından kullanıldığında ve abonelik tarafından kullanıldığında genel arama karakteri olarak karakter olarak kabul edilir. Bu karışıklığa yol açabilir.

Örnek kod parçacığı

Örnek 2: Bir değişken konusuna yayıncıörnek programından çıkarılan bu kod parçacığı, bir konu nesnesini bir değişken konu dizgisiyle birleştirir.

```
MQOD      td = {MQOD_DEFAULT}; /* Object Descriptor          */
td.ObjectType = MQOT_TOPIC; /* Object is a topic    */
td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
strcpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
td.ObjectString.VSPtr = topicString;
td.ObjectString.VSLength = (MQLONG)strlen(topicString);
td.ResObjectString.VSPtr = resTopicStr;
```



```
td.ResObjectString.VSBufSize = sizeof(resTopicStr)-1;  
MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF_QUIESCING, &Hobj, &CompCode, &Reason);
```

Konu ağaçları

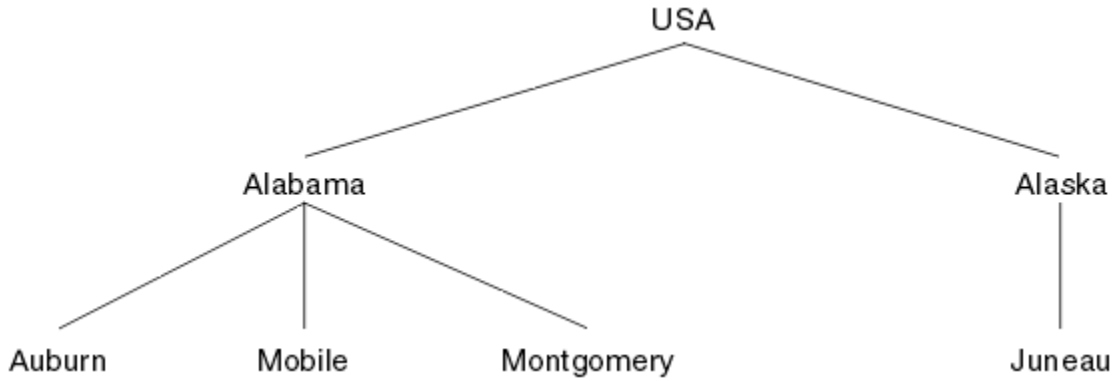
Tanımladığınız her konu, konu ağacında bir ögedir ya da düğümdür. Konu ağacı, önceden MQSC ya da PCF komutlarını kullanarak önceden tanımlanmış konuları içeren ya da içerebilecek boş olabilir. Yeni bir konuyu, konu yaratma komutlarını kullanarak ya da bir yayınlama ya da abonelikte ilk kez konuyu belirterek tanımlayabilirsiniz.

Bir konunun konu dizgisini tanımlamak için herhangi bir karakter dizgisi kullanabilmenize rağmen, sıradüzensel bir ağaç yapısına uygun bir konu dizgisi seçmeniz önerilir. Konu sokmaları ve konu ağaçlarının düşünceli tasarımı aşağıdaki işlemleri yapabilmenize yardımcı olabilir:

- Birden çok konuya abone olma.
- Güvenlik ilkeleri oluşturuluyor.

Düz, doğrusal bir yapı olarak bir konu ağacı oluşturabilmenize rağmen, bir ya da daha fazla kök konumla sıradüzensel bir yapıda bir konu ağacı oluşturmak daha iyi olur. Güvenlik planlaması ve konularıyla ilgili daha fazla bilgi için [Güvenlik yayınlama/abone olmabaşlıklı konuya](#) bakın.

[Şekil 5 sayfa 33](#) , bir kök konusu olan bir konu ağacına ilişkin bir örneği gösterir.



Şekil 5. Konu ağacı örneği

Şekildeki her karakter dizgisi, konu ağacındaki bir düğümü temsil eder. Genel bir konu dizgesi, konu ağacındaki bir ya da daha fazla düzeyden düğümler toplayarak yaratılır. Düzeyler "/" karakteriyle ayrılır. Tam olarak belirtilmiş bir konu dizgisinin biçimi şöyledir: "root/level2/level3".

[Şekil 5 sayfa 33](#) içinde gösterilen konu ağacındaki geçerli konular şunlardır:

```
"ABD"  
"ABD/Alabama"  
"USA/Alaska"  
"USA/Alabama/Auburn"  
"USA/Alabama/Mobile"  
"USA/Alabama/Montgomery"  
"USA/Alaska/Juneau"
```

Konu dizgileri ve konu ağaçlarını tasarladığınızda, kuyruk yöneticisinin yorumlamadığını ya da konu dizgisinin kendisinden anlam türetmeyi denediğini unutmayın. Bu konu, seçilen iletileri o konunun abonelerine göndermek için konu dizgisini kullanır.

Bir konu ağacının yapısı ve içeriği için aşağıdaki ilkeler geçerlidir:

- Bir konu ağacındaki düzeylerin sayısı için bir sınır yoktur.
- Bir konu ağacında bir düzeyin adının uzunluğuna ilişkin bir sınır yoktur.
- Herhangi bir sayıda "kök" düğüm olabilir; yani, herhangi bir sayıda konu ağacı olabilir.

Konu ağacındaki istenmeyen konuların sayısını azaltma

Konu ağacındaki istenmeyen konuların sayısı azaltılarak yayınlama/abone olma sisteminin performansı iyileştirilir. İstenmeyen bir konu nedir ve bunları nasıl kaldırılıyorsunuz?

Performansı olumsuz etkilemeden çok sayıda konu oluşturabilirsiniz. Ancak, sürekli olarak konu ağaçlarının genişletilmesi için yayınlama/abone olma sonuçlarını kullanmanın yolları vardır. Son derece büyük sayıda konu bir kez oluşturulur ve bir daha kullanılmaz. Artan konu sayısı bir performans sorunu olabilir.

Büyük ve artan sayıda istenmeyen konuya yol açan tasarımlardan nasıl kaçınabilirsiniz? Kuyruk yöneticisinden istenmeyen konuları konu ağacından kaldırmasına yardımcı olmak için neler yapabilirsiniz?

Kuyruk yöneticisi, 30 dakika boyunca kullanılmadığı için istenmeyen bir konuyu tanıır. Kuyruk yöneticisi, konu ağacından sizin için kullanılmayan konuları kaldırır. Kuyruk yöneticisi özneteliği **TREELIFE** ile değiştirilerek 30 dakikalık süre değiştirilebilir. Kuyruk yöneticisinin, konunun kuyruk yöneticisinin kullanılmaması için kuyruğun yöneticisine görüneceğinden emin olarak, istenmeyen konuları kaldırmasına yardımcı olabilirsiniz. "Kullanılmamış bir konu nedir?" sayfa 34 bölümünde, kullanılmamış bir konunun ne olduğunu açıklar.

Bir programcı, herhangi bir uygulama tasarlama ve özellikle de uzun süre çalışan bir uygulama tasarlama, kaynak kullanımını dikkate alır: Programın ne kadar kaynak gerektirdiğini, sınırsız talep ve kaynak sızıntıları olduğunu mu göz önünde bulundurur? Konular, programların kullanımını yayınlayan/abone olan bir kaynaktır. Bir program tarafından kullanılan diğer kaynaklar gibi konuların kullanımını önlendirin.

Kullanılmamış bir konu nedir?

Kullanılmamış bir konunun ne olduğunu tanımlamadan önce, konu olarak tam olarak ne sayılır?

Bir konu dizgisi (USA/Alabama/Auburn gibi) bir konuya dönüştürüldüğünde konu, konu ağacına eklenir. Gerekirse, ek konu düğümleri ve bunlara karşılık gelen konular ağaçta yaratılır. The topic string USA/Alabama/Auburn is converted into a tree with three topics.

- USA
- USA/Alabama
- USA/Alabama/Auburn

To display all the topics in the topic tree, use the **runmqsc** command `DISPLAY TPSTATUS('# ') TYPE(TOPIC)`.

Konu ağacındaki kullanılmayan bir konu aşağıdaki özelliklere sahiptir.

Bir konu nesnesiyle ilişkilendirilmemiş

Bir denetim konusu nesnesi, konuyu bir konu ile ilişkilendiren bir konu dizgisine sahiptir. When you define the topic object Alabama, if the topic, USA/Alabama, it is to be associated with does not exist, the topic is created from the topic string. Konu varsa, konu nesnesi ve konu, konu dizisini kullanarak ilişkilendirilir.

Alıkonan bir yayını yok

Alıkonan yayının sonuçlarına sahip bir konu, yayıncıdan MQPMO_RETAIN seçeneği ile bir konuya ileti yerleştiriyor.

Use the **runmqsc** command `DISPLAY TPSTATUS(' USA/Alabama ') RETAINED` to check if USA/Alabama has a retained publication. Yanıt YES ya da NO' dir.

Alıkonan bir yayını USA/Alabama ' den kaldırmak için **runmqsc** komutunu `CLEAR TOPICSTR(' USA/Alabama ') CLRTYPE(RETAINED)` komutunu kullanın.

Alt konuları yok

USA/Alabama/Auburn, alt konuları olmayan bir konudur. USA/Alabama/Auburn, USA/Alabama' in doğrudan alt konudur.

USA/Alabama ' un doğrudan alt öğelerini **runmqsc** komutu `DISPLAY TPSTATUS('USA/Alabama/+')` ile görüntüleyin.

Düğümdeki etkin yayıncılar yok

Bir düğüm için etkin bir yayıncı, çıkışa açık olan bir uygulamadır.

For example, an application opens the topic object named **Alabama** with open options `MQOO_OUTPUT`.

To display active publishers to USA/Alabama and all its children, use the **runmqsc** command `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(PUB) ACTCONN`.

Düğümde etkin abone yok

Etkin bir abone, kalıcı bir abonelik olabilir ya da `MQSUB` ile bir konuya ilişkin aboneliği kaydettirmiş ve kapatmamış bir uygulama olabilir.

Etkin abonelikleri USA/Alabama' e görüntülemek için **runmqsc** komutunu `DISPLAY TPSTATUS('USA/Alabama') TYPE(SUB) ACTCONN` kullanın.

Etkin abonelikleri USA/Alabama ve tüm alt öğeleri için görüntülemek için **runmqsc** komutunu `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(SUB) ACTCONN` kullanın.

Bir konu ağacındaki konu sayısını azaltma

Özette, bir konu ağacındaki konu sayısını azaltmanın birçok yolu vardır.

TREELIFE öğesini değiştir

Kullanılmayan bir konunun varsayılan değer olarak ömrü 30 dakika olur. Kullanılmayan bir konunun ömrünün daha küçük olmasını sağlayabilirsiniz.

Örneğin, **runmqsc** komutu, `ALTER QMGR TREELIFE(900)`, kullanılmayan bir konunun ömrünü 30 dakika ile 15 dakikaya indirir.

Kural dışı olarak kuyruk yöneticisini yeniden başlatın.

Kuyruk yöneticisi yeniden başlatıldığında, konu ağacında konu nesnelere, alıkonan yayınlara sahip düğümlerden ve sürekli aboneliklerden yeniden başlatılır. Yayıncı ve abone programlarının işleyişi ile oluşturulan konular ortadan kaldırılmıştır.

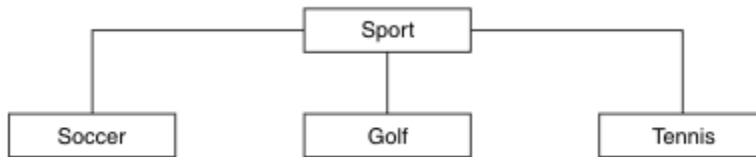
Use the **runmqsc** command `DISPLAY TPSTATUS('#') TYPE(TOPIC)` periodically to list all topics and check if the number is growing.

Son çare olarak, istenmeyen konulardaki büyüme geçmişteki performans sorunlarının nedeni olarak ortaya çıktı ve kuyruk yöneticisini yeniden başlatın.

Yönetimle ilgili konu nesneleri

Yönetimle ilgili bir konu nesnesini kullanarak, konulara varsayılan, varsayılan olmayan öznitelikler atayabilirsiniz.

Şekil 6 sayfa 35 , Sport üst düzey bir konunun farklı sporları kapsayan ayrı konulara nasıl bölüneceğini, konu ağacı olarak nasıl görselleştirdiğini gösterir:



Şekil 6. Konu ağacını görselleştirme

Şekil 7 sayfa 36 , konu ağacının her bir sporla ilgili farklı bilgi tiplerini birbirinden ayırmak için nasıl daha fazla bölünebileceğini gösterir:



Şekil 7. Ek konu ağacı

Konu ağacını gösterebilmek için, hiçbir denetim konusu nesnesi tanımlanmalı. Bu ağaçtaki düğümlerin her biri, yayınlama ya da abone olma işleminde yaratılan bir konu dizisiyle tanımlanır. Ağaçtaki her konu, özniteliklerini üst öğelerinden devralır. Varsayılan olarak tüm öznitelikler ASPARENT olarak ayarlandığından, öznitelikler üst konu nesnesinden devralınır. Bu örnekte, her konu Sport ile aynı özniteliklere sahiptir. Sport konularının denetim konusu nesnesi yoktur ve öznitelikleri `SYSTEM.BASE.TOPIC`.

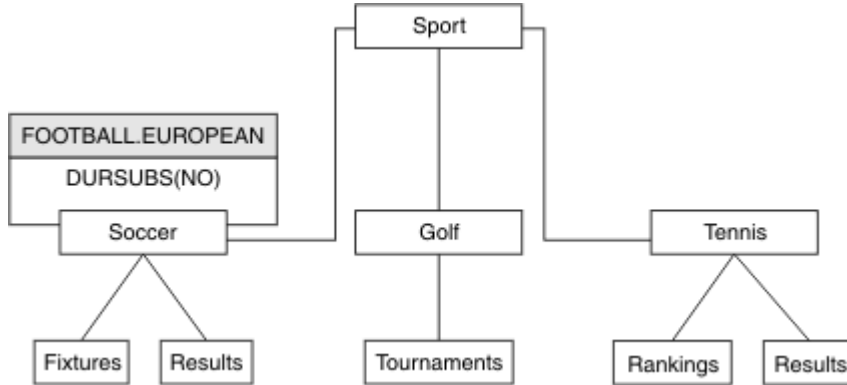
Note, that it is not good practice to give authorities for non-mqm users at the root node of the topic tree, which is `SYSTEM.BASE.TOPIC`, because the authorities are inherited but cannot be restricted. bu nedenle otoriteleri bu seviyede vererek, yetkililere tüm ağacı verebiliyorsun. Hiyerarşide daha düşük bir konu düzeyinde yetki vermelisiniz.

Denetim konusu nesnelere, konu ağacındaki belirli düğümlere ilişkin belirli öznitelikleri tanımlamak için kullanılabilir. Aşağıdaki örnekte, yönetici konu nesnesi, soccer konularının DURABLE özelliğini NO değerine ayarlamak için tanımlıdır:

```

DEFINE TOPIC(FOOTBALL.EUROPEAN)
TOPICSTR('Sport/Soccer')
DURSUB(NO)
DESCR('Administrative topic object to disallow durable subscriptions')
  
```

Konu ağacı şu şekilde görselleştirilebilir:



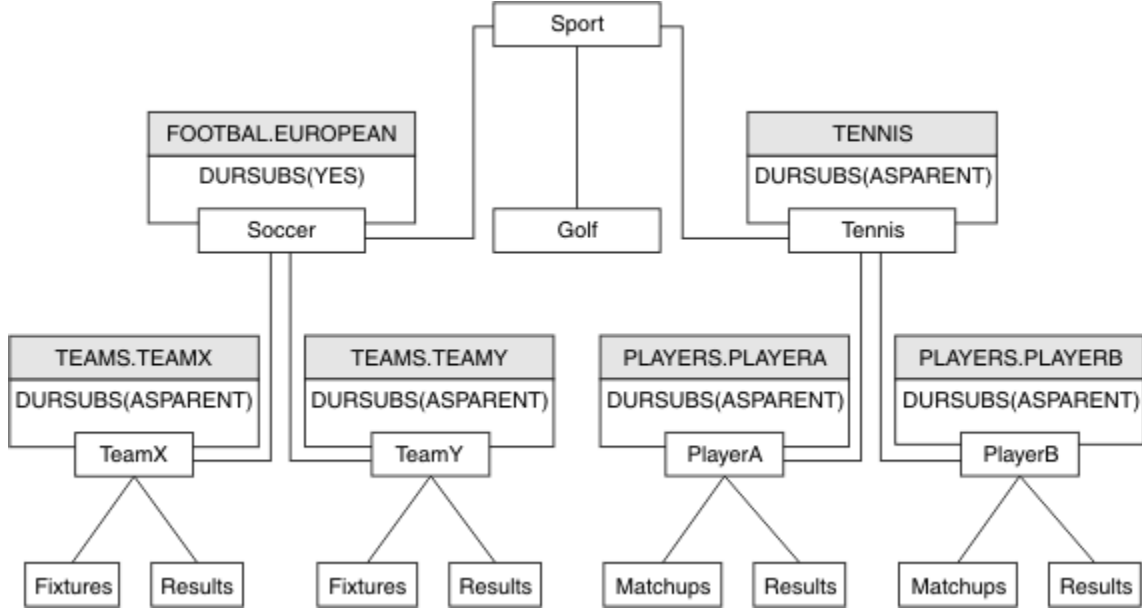
Şekil 8. Sport/Soccer konusuyla ilişkili bir yönetici konu nesnesinin görselleştirilmesi

Ağaçtaki Soccer altındaki konulara abone olan uygulamalar, denetim konusu nesnesi eklenmeden önce kullandıkları konu dizelerini kullanmaya devam edebilir. However, an application can now be written to subscribe using the object name `FOOTBALL.EUROPEAN`, instead of the string `/Sport/Soccer`. For example, to subscribe to `/Sport/Soccer/Results`, an application can specify MQSD.ObjectName as `FOOTBALL.EUROPEAN` and MQSD.ObjectString as `Results`.

Bu özellik sayesinde, uygulama geliştiricilerinden konu ağacının bir kısmını gizleyebilirsiniz. Konu ağacındaki belirli bir düğümden bir denetim konusu nesnesi tanımlayın ve uygulama geliştiricileri, kendi konularını düğümün alt öğeleri olarak tanımlayabilir. Geliştiricilerin üst konu hakkında bilgi sahibi olması gerekir, ancak üst ağaçtaki diğer düğümlerle ilgili bilgi sahibi olmamalıdır.

Öznitelikleri edinir

Bir konu ağacında birçok yönetici konu nesnesi varsa, her bir denetim konusu nesnesi varsayılan olarak, özniteliklerini en yakın üst düzey yönetici konusundan edinir. Önceki örnek, Şekil 9 sayfa 37’inde genişletilmiştir:



Şekil 9. Çok sayıda denetim konusu nesnesi olan konu ağacı

For example use inheritance to give all the child topics of /Sport/Soccer the property that subscriptions are non-durable. FOOTBALL . EUROPEAN ' nin DURSUB özneliğini NOolarak değiştirin.

Bu öznitelik aşağıdaki komutu kullanarak ayarlanabilir:

```
ALTER TOPIC (FOOTBALL . EUROPEAN) DURSUB (NO)
```

All the administrative topic objects of child topics of Sport/Soccer have the property DURSUB set to the default value ASPARENT. FOOTBALL . EUROPEAN , DURSUB özellik değerini NOolarak değiştirdikten sonra, Sport/Soccer alt konuları, NO DURSUB özellik değerini devralır. All child topics of Sport/Tennis inherit the value of DURSUB from SYSTEM . BASE . TOPIC object. SYSTEM . BASE . TOPIC , YESdeğerine sahiptir.

Sport/Soccer/TeamX/Results konusuna dayanlı bir abonelik yapmaya çalışmak artık başarısız olacaktır; ancak, Sport/Tennis/PlayerB/Results ' e kalıcı bir abonelik yapmaya çalışmak başarılı olur.

Controlling wildcard usage with the JOKER property

Yayınların genel arama karakteri konu dizgisi adlarını kullanan abone uygulamalarına teslim edilmesini denetlemek için MQSC **Topic JOKER** özelliğini ya da eşdeğer PCF Topic WildcardOperation özelliğini kullanın. JOKER özelliği, iki olası değerden birine sahip olabilir:

Genel arama karakteri

Bu konuya ilişkin genel arama karakteri abonelikleri davranışı.

Passthru

Bu konu nesnesindeki konu dizgisinden daha az belirli bir genel konuya özgü olan abonelikler, bu konuda yapılan yayınları ve bu konuya ilişkin dizgileri daha belirli bir konuya getirmektedir.

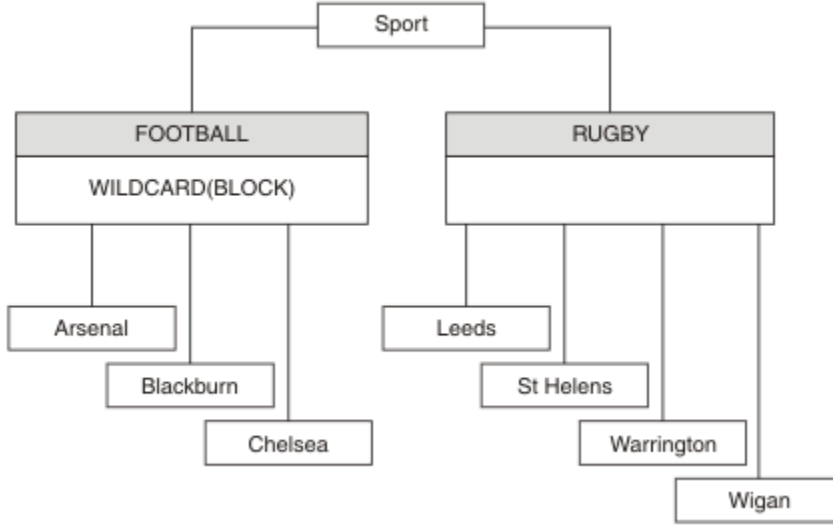
Öbek

Bu konu nesnesindeki konu dizgisinden daha az belirli bir genel konuya özgü olan abonelikler, bu konuya ya da bu konudan daha belirli bir konuya ilişkin dizgileri elde edemeyecek şekilde yayınlar.

Abonelikler tanımlandığında bu özneliğin değeri kullanılır. Bu özneliği değiştirirseniz, var olan aboneliklerle kapsanan konu kümesi değişiklikten etkilenmez. Bu senaryo, konu nesneleri

yaratıldığında ya da silindiğinde topoloji değişirse de geçerli olur; JOKER özneliğinin değiştirilmesinin ardından oluşturulan aboneliklerle eşleşen konu kümesi, değiştirilen topoloji kullanılarak oluşturulur. Var olan abonelikler için eşleşen konular kümesini yeniden değerlendirmeye zorlamak istiyorsanız, kuyruk yöneticisini yeniden başlatmanız gerekir.

In the example, “[Örnek: Sport yayınlama/abone olma kümesini yaratın](#)” sayfa 41, you can follow the steps to create the topic tree structure shown in [Şekil 10 sayfa 38](#).



Şekil 10. JOKER özelliğini kullanan bir konu ağacı, BLOCK

A subscriber using the wildcard topic string # receives all publications to the Sport topic and the Sport/Rugby subtree. The subscriber receives no publications to the Sport/Football subtree, because the JOKER property value of the Sport/Football topic is BLOCK.

Varsayılan değer PASSTHRU ' dir. Sport ağacındaki düğümlere JOKER özellik değerini PASSTHRU olarak ayarlayabilirsiniz. Düğümlerde BLOCK JOKER özellik değeri yoksa, PASSTHRU ayarı abonelerin Sports ağacındaki düğümlere göre gözlemlediği davranışı değiştirmez.

Örnekte, genel arama karakterinin, teslim edilen yayınları nasıl etkilediğini görmek için abonelikler oluşturun; bkz. [Şekil 14 sayfa 43](#). Bazı yayınlar yaratmak için [Şekil 17 sayfa 44](#) içinde yayınlama komutunu çalıştırın.

```
pub QMA
```

Şekil 11. QMA ' ya Yayınla

Sonuçlar, Çizelge 3 sayfa 38 içinde gösterilir. JOKER property özellik değerinin BLOCK' in genel arama karakteri kapsamında, genel arama karakterlerinin genel arama karakterlerine sahip aboneliklerini nasıl önlediğini fark edin.

Çizelge 3. QMA tarihinde alınan yayınlar			
Abonelik	Konu dizisi	Alınan yayınlar	Notlar
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	All publications to Football subtree blocked by WILDCARD (BLOCK) on Sports/Football
SARSENAL	Sports/#/Arsenal	-	Sports/Football üzerinde WILDCARD (BLOCK) , Arsenal üzerinde genel arama karakteri aboneliğini önler

Çizelge 3. QMA tarihinde alınan yayınlar (devamı var)			
Abonelik	Konu dizisi	Alınan yayınlar	Notlar
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby için varsayılan WILDCARD , Leeds üzerinde genel arama karakteri aboneliğini engellemektedir.

Not:

Suppose a subscription has a wildcard that matches a topic object with the JOKER property value BLOCK. Aboneliğin aynı zamanda eşleşen genel arama karakterinin sağında bir konu dizisi varsa, abonelik hiçbir zaman bir yayını almaz. Engellenen yayınların kümesi, engellenen genel arama karakterinin ebeveynleri olan konular için yayınlardır. Publications to topics that are children of the topic with the BLOCK property value are blocked by the wildcard. Bu nedenle, genel arama karakterinin sağına bir konu içeren abonelik konu dizileri, hiçbir yayın için hiçbir yayın almayacak.

WILDCARD özellik değerinin BLOCK olarak ayarlanması, genel arama karakterleri içeren bir konu dizisini kullanarak abone olamayacağınız anlamına gelmez. Böyle bir abonelik normal. The subscription has an explicit topic that matches the topic with a topic object having a JOKER property value BLOCK. It uses wildcards for topics that are parents or children of the topic with the JOKER property value BLOCK. [Şekil 10 sayfa 38](#)örneğinde, Sports/Football/# gibi bir abonelik yayınları alabilir.

Genel arama karakterleri ve küme konuları

Küme konu tanımları, bir kümedeki her kuyruk yöneticisine dağıtılır. Kuyruk yöneticisinde bir küme yöneticisinde bulunan bir küme konusuna abonelik, yetkili abonelikleri oluşturan kuyruk yöneticisinde yer alan kuyruk yöneticisinde yer alan bir sonuçla sonuçlanır. Bir yetkili sunucu aboneliği, kümedeki diğer her kuyruk yöneticisinde yaratılır. Genel arama karakterleri içeren, küme konularıyla birleştirilen konuları kullanan abonelikler, davranışı tahmin etmek için zor bir yol gösterebilirler. Bu davranış, aşağıdaki örnekteki açıklamadır.

In the cluster set up for the example, “[Örnek: Sport yayınlama/abone olma kümesini yaratın](#)” sayfa 41, QMB has the same set of subscriptions as QMA, yet QMB received no publications after the publisher published to QMA, see [Şekil 11 sayfa 38](#). Sports/Football ve Sports/Rugby konuları küme başlıklarına rağmen, fullsubs.tst içinde tanımlanan abonelikler bir küme konusuna gönderme yapmamış olur. Yetkili sunucu aboneliği, QMB 'tan QMA' a yayılmaz. Yetkili abonelikler olmadan, QMA hiçbir yayın QMB ' a iletilmedi.

Some of the subscriptions, such as Sports/#/Leeds , might seem to reference a cluster topic, in this case Sports/Rugby . The Sports/#/Leeds subscription actually resolves to the topic object SYSTEM.BASE.TOPIC.

Sports/#/Leeds gibi bir aboneliğin gönderme yaptığı konu nesnesinin çözülmesine ilişkin kural şu şekildedir. Konu dizisini ilk genel arama karakterine kes. İlişkili bir denetim konusu nesnesi olan ilk konuyu aramak için konu dizisinden sola doğru tarayın. Konu nesnesi bir küme adı belirtebilir ya da bir yerel konu nesnesi tanımlayabilir. In the example, Sports/#/Leeds, the topic string after truncation is Sports, which has no topic object, and so Sports/#/Leeds inherits from SYSTEM.BASE.TOPIC, which is a local topic object.

Kümelenmiş konulara abone olunmanın, genel arama karakterinin yayılmasını nasıl değiştirebileceğini görmek için, [upsubs.batto](#)plu iş komut dosyasını çalıştırın. Komut dosyası abonelik kuyruklarını temizler ve fullsubs.tst içinde küme başlığı aboneliklerini ekler. Bir dizi yayın yaratmak için [puba.bat](#) komutunu yeniden çalıştırın; bkz. [Şekil 11 sayfa 38](#).

[Çizelge 4 sayfa 40](#) , yayınların yayımlandığı kuyruk yöneticisine iki yeni abonelik eklenmesinin sonucunu gösterir. Sonuç beklendiği gibi, yeni abonelikler her biri bir yayın alır ve diğer abonelikler tarafından alınan yayınların sayıları değişmez. Diğer küme kuyruk yöneticisinde beklenmeyen sonuçlar ortaya çıkar; bkz. [Çizelge 5 sayfa 40](#).

Çizelge 4. QMA tarihinde alınan yayınlar			
Abonelik	Konu dizisi	Alınan yayınlar	Notlar
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	All publications to Football subtree blocked by WILDCARD (BLOCK) on Sports/Football
SARSENAL	Sports/#/Arsenal	-	Sports/Football üzerinde WILDCARD (BLOCK) , Arsenal üzerinde genel arama karakteri aboneliğini önler
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby için varsayılan WILDCARD , Leeds üzerinde genel arama karakteri aboneliğini engellemektedir.
FARSENAL	Sports/Football/ Arsenal	Sports/Football/ Arsenal	Arsenal , aboneliğin bir genel arama karakterinin olmadığı için bir yayın alır.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds , herhangi bir etkinlikte bir yayın alır.

Çizelge 5 sayfa 40 shows the results of adding the two new subscriptions on QMB and publishing on QMA. QMB ' in bu iki yeni aboneliği olmadan hiçbir yayın almadığını geri çağırın. As expected, the two new subscriptions receive publications, because Sports/FootBall and Sports/Rugby are both cluster topics. QMB forwarded proxy subscriptions for Sports/Football/Arsenal and Sports/Rugby/Leeds to QMA, which then sent the publications to QMB.

Beklenmeyen sonuç, daha önce yayın içermeyen Sports/# ve Sports/#/Leeds adlı aboneliklerin artık yayınları aldığından kaynaklanır. Bunun nedeni, diğer abonelikler için QMB 'e iletilen Sports/Football/Arsenal ve Sports/Rugby/Leeds yayınlarının artık QMB ' e bağlı herhangi bir abone için kullanılabilmesinin nedeni. Consequently the subscriptions to the local topics Sports/# and Sports/#/Leeds receive the Sports/Rugby/Leeds publication. Spor/Futbol, JOKER özellik değeri BLOCK olarak ayarlanmış olduğundan, Sports/#/Arsenal bir yayın almamaya devam eder.

Çizelge 5. QMB tarihinde alınan yayınlar			
Abonelik	Konu dizisi	Alınan yayınlar	Notlar
SPORTS	Sports/#	Sports/Rugby/Leeds	All publications to Football subtree blocked by WILDCARD (BLOCK) on Sports/Football
SARSENAL	Sports/#/Arsenal	-	Sports/Football üzerinde WILDCARD (BLOCK) , Arsenal üzerinde genel arama karakteri aboneliğini önler
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby için varsayılan WILDCARD , Leeds üzerinde genel arama karakteri aboneliğini engellemektedir.
FARSENAL	Sports/Football/ Arsenal	Sports/Football/ Arsenal	Arsenal , aboneliğin bir genel arama karakterinin olmadığı için bir yayın alır.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds , herhangi bir etkinlikte bir yayın alır.

Çoğu uygulamada, bir aboneliğin başka bir aboneliğin davranışını etkilemesi istenmeyen bir davranışa neden olur. One important use of the JOKER property with the value BLOCK is to make the subscriptions to the same topic string containing wildcards behave uniformly. Aboneliğin yayıncı ile aynı kuyruk yöneticisine mi, yoksa farklı bir aboneliğe mi ait olduğunu, aboneliğin sonuçlarının aynı olduğunu mu?

Joker karakterler ve akışlar

WebSphere MQ Sürüm 6 akışları, konulara WebSphere MQ Sürüm 7 ile eşlenir; bkz. “Akışlar ve konular” sayfa 44 . In the default mapping, which is performed by **strmqbrk** in Version 7, all the topics in the stream Sports would be mapped to the topic Sports. Business akışındaki tüm konular Businesskonuyla eşlenecek.

Sports akışındaki WebSphere MQ Sürüm 6 ile * arasındaki bir abonelik, Sports ağacındaki tüm yayınları alır ve Business ağacındaki hiçbir yayın almaz. Sürüm 7 'de aynı abonelik, Sports ağacındaki tüm yayınları ve Business ağacındaki tüm yayınları alır. Akışlar sürüm 7 'ye geçirildiğinde bu davranışı engellemek için **strmqbrk** , JOKER özelliğini ayarlar. Bir akıştan geçirilen en üst düzey konuların her biri için bu değeri BLOCK değerine ayarlar. Sports ve Business için JOKER özelliği, Sports ve Businessadlı sürüm 6 akışlarından dönüştürme tarafından BLOCK değerine ayarlıdır.

Yayınlama/abone olma API ' ya yazılan yeni bir uygulama için geçerli olan etki, * aboneliğinin yayınsız olarak alınmamasını sağlar. To receive all the Sports publications you must subscribe to Sports/*, or Sports/# , and similarly for Business publications.

Yayınlama/abone olma aracısı WebSphere MQ Sürüm 7 'ye geçirildiğinde, kuyruğa alınan var olan bir yayınlama/abone olma uygulamasının davranışı değişmez. The **StreamName** property in the **Publish**, **Register Publisher**, or **Subscriber** commands is mapped to the name of the topic the stream has been migrated to.

Genel arama karakterleri ve abonelik noktaları

WebSphere Message Broker abonelikleri noktaları, WebSphere MQ Sürüm 7 ile konulara eşlenir; bkz. “Abonelik noktaları ve konular” sayfa 46. Sürüm 7 'de **migmqbrk** tarafından gerçekleştirilen varsayılan eşlemede, Sports abonelik noktasındaki tüm konular Sportskonusuna eşlenecektir. Business abonelik noktasındaki tüm konular, Businesskonuyla eşlenecektir.

Sports abonelik noktasındaki WebSphere Message Broker Sürüm 6 ile * arasındaki bir abonelik, Sports ağacındaki tüm yayınları alır ve Business ağacındaki hiçbir yayın yayınlanmaz. Sürüm 7 'de aynı abonelik, Sports ağacındaki tüm yayınları ve Business ağacındaki tüm yayınları alır. Bu davranışı engellemek için, abonelik noktaları sürüm 7 'ye geçirildiğinde, **migmqbrk** JOKER özelliğini ayarlar. Bir abonelik noktasından geçirilen en üst düzey konuların her biri için bu değeri BLOCK değerine ayarlar. The JOKER property of Sports and Business is set to the value BLOCK by the conversion from the WebSphere Message Broker subscription points called Sports and Business .

Yayınlama/abone olma API ' ya yazılan yeni bir uygulama için, geçişin etkisi, * aboneliğinin yayınsız olarak alınmamasını sağlar. To receive all the Sports publications you must subscribe to Sports/*, or Sports/# , and similarly for Business publications.

Yayınlama/abone olma aracısı WebSphere MQ Sürüm 7 'ye geçirildiğinde, kuyruğa alınan var olan bir yayınlama/abone olma uygulamasının davranışı değişmez. **Publish**, **Register Publisher** ya da **Subscriber** komutlarındaki **SubPoint** özelliği, aboneliğin yeni düzeye geçirilmiş olduğu konunun adıyla eşlenir.

Örnek: Sport yayınlama/abone olma kümesini yaratın

Aşağıdaki adımlar, dört kuyruk yöneticisi içeren bir küme (CL1) oluşturur: iki tam havuz, CL1A ve CL1B , ve iki kısmi havuz, QMA ve QMB. Tüm havuzlar yalnızca küme tanımlamalarını tutmak için kullanılır. QMA , küme konusu anasistemine atanır. Sürekli abonelikler hem QMA hem de QMBüzerinde tanımlanır.

Not: Örnek, Windowsiçin kodlanmıştır. Örneğin, diğer platformlardaki örneği yapılandırmak ve sınamak için [Create qmgrs.bat](#) ve [pub.bat](#)oluşturma kodunu yeniden kodlamanız gerekir.

1. Komut dosyalarını oluşturun.

- a. [topics.tst](#) yarat
- b. [Create wildsubs.tst](#)
- c. [Create fullsubs.tst](#)
- d. [Create qmgrs.bat](#)
- e. [oluştur pub.bat](#)

2. Yapılandırmayı oluşturmak için [Create qmgrs.bat](#) komutunu çalıştırın.

```
qmgrs
```

Şekil 10 sayfa 38’inde konuları oluşturun. The script in figure 5 creates the cluster topics Sports/ Football and Sports/Rugby.

Not: REPLACE seçeneği, bir konunun TOPICSTR özelliklerini değiştirmez. TOPICSTR , örnekte farklı konu ağaçlarını test etmek için kullanılan bir özeldir. Konuları değiştirmek için önce konuyu silin.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

Şekil 12. Konuları silin ve yaratın: [topics.tst](#)

Not: REPLACE konu dizgilerinin yerini değiştirmedeği için konuları silin.

Joker karakterler içeren abonelikler oluşturun. Genel arama karakterleri, Şekil 10 sayfa 38’indeki konu nesnelileri ilgili konulara karşılık gelir. Her abonelik için bir kuyruk yaratın. Kuyruklar temizlenir ve komut dosyası çalıştırıldığında ya da yeniden çalıştırıldığında abonelikler silinir.

Not: REPLACE seçeneği, bir aboneliğin TOPICOBJ ya da TOPICSTR özelliklerini değiştirmez. TOPICOBJ or TOPICSTR are the properties that are usefully varied in the example to test different subscriptions. Bunları değiştirmek için önce aboneliği silin.

```
DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)
```

Şekil 13. Genel arama karakteri abonelikleri yarat: [wildsubs.tst](#)

Küme konusu nesnelere başvuruda bulunan abonelikler oluşturun.

Not:

The delimiter, /, is automatically inserted between the topic string referenced by TOPICOBJ, and the topic string defined by TOPICSTR.

DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL), aynı aboneliği oluşturur. TOPICOBJ, önceden tanımladığınız konu dizgisine hızlı bir şekilde gönderme yapmak için kullanılır. Abonelik, yaratıldığında, artık konu nesnesini ifade etmemez.

```
DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)
```

Şekil 14. Abonelikleri sil ve yarat: fullsubs.tst

İki havuzu olan bir küme oluşturun. Yayınlama ve abone olma için iki kısmi havuz oluşturun. Her şeyi silmek ve yeniden başlamak için komut dosyasını yeniden çalıştırın. Komut dosyası aynı zamanda konu sıradüzenini ve ilk genel arama karakteri aboneliklerini de yaratır.

Not:

Diğer platformlarda, benzer bir komut dosyası yazın ya da tüm komutları yazın. Bir komut dosyasının kullanılması, her şeyi silmesini ve aynı bir yapılandırma ile yeniden başlatılmasını hızlı bir şekilde sağlar.

```
@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof
```

Şekil 15. Kuyruk yöneticileri yarat: qmgrs.bat

Küme konularına abonelikleri ekleyerek yapılandırmayı güncelleyin.

```
@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst
```

Şekil 16. Abonelikleri güncelleştir: upsubs.bat

Yayın konusu dizgisini içeren iletileri yayınlamak için, parametre olarak kuyruk yöneticisiyle pub.bat komutunu çalıştırın. Pub.bat uses the sample program **amqspub**.

```
@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1
```

Şekil 17. Yayınla: pub.bat

Akışlar ve konular

Kuyruğa alınan yayınlama/abone olma, tümleştirilmiş yayınlama/abone olma modelinde var olmayan bir yayın akışı kavramına sahip. Kuyruklanan yayınlama/abone olma yolunda akışlar, farklı konulara ilişkin bilgi akışını birbirinden ayırmanın bir yolunu sağlar. IBM WebSphere MQ Version 6.0 uygulamasında, akış, akışı destekleyen her bir araçta tanımlı bir kuyruk olarak uygulanır. Her kuyruk aynı ada sahiptir (akışın adı). From IBM WebSphere MQ Version 7.0 onwards, a stream is implemented as a top-level topic that can be mapped to a different topic identifier administratively.

Varsayılan akış SYSTEM.BROKER.DEFAULT.STREAM, bir ağdaki tüm araçlar ve kuyruk yöneticileri için otomatik olarak ayarlanır ve varsayılan akışın kullanılması için ek bir yapılandırma gerekmez. Varsayılan akışı, adlandırılmamış bir varsayılan konu alanı olarak düşünün. Topics published to the default stream are immediately available to all connected Version 6.0 brokers, and all queue managers from Version 7.0 onwards, with queued publish/subscribe enabled. Adlandırılan akışlar, adlandırılmış, konu alanları gibi ayrı bir alanlardır. Adı belirtilen akış, kullanıldığı her bir araçta tanımlanmalıdır.

When you define a topic, the topic is available to Version 6.0 publish/subscribe brokers, and to publishers and subscribers running on a later version of IBM WebSphere MQ, with no special configuration.

Yayıncılar ve aboneler farklı kuyruk yöneticileriye, araçlar aynı aracı sıradüzenine bağlandıktan sonra yayınlar için başka bir yapılandırma gerekmez ve bunlar arasında akışa ilişkin abonelikler de yoktur. Aynı birlikte çalışabilirlik de aynı şekilde devam ediyor.

Adlandırılan akışlar

Kuyruğa alınmış yayınlama/abone olma programlama modeli ile çalışan bir çözüm tasarımcısı, tüm spor yayınlarını Sport adlı bir adlandırılan akışa yerleştirmeye karar verebilir. Version 6.0 içinde bir akış, genellikle model kuyruğunu kullanan diğer araçlarda otomatik olarak eşlenir, SYSTEM.BROKER.MODEL.STREAM. Ancak, akışın Version 7.0 üzerinde kuyruğa alınan yayınlama/abone olma etkinleştirilmiş olarak çalıştırıldığı bir kuyruk yöneticisine kullanılabilir olması için, akışın el ile eklenmesi gerekir.

If you migrate a queue manager from Version 6.0, running the command **strmqbrk** migrates Version 6.0 named streams to topics. Sport akışı Sportkonuyla eşlendi. Bu, z/OS için geçerli değildir.

Queued publish/subscribe applications that subscribe to Soccer/Results on stream Sport work without change. Integrated publish/subscribe applications that subscribe to the topic Sport using MQSUB, and supplying the topic string Soccer/Results receive the same publications too.

When the topic Soccer/Result is created by **strmqbrk**, it is defined as a child of the topic Sport, with the topic string Sport. A subscription to Soccer/Results is realized as a subscription to Sport/Soccer/Results, and so publications to the Sport stream are mapped to different place in topic space to publications to a different stream, such as Business.

strmqbrk tarafından gerçekleştirilen otomatik geçişin yanıt olmadığı senaryolar vardır ve akışları el ile eklemeniz gerekir. Akış ekleme görevi, [Akış eklenmesibaşlıklı](#) konuda açıklanmaktadır. Üç nedenden dolayı el ile akışlar eklemeniz gerekebilir.

1. Daha sonraki kuyruk yöneticilerine çalışan yeni yazılan yayınlama/abone olma uygulamalarıyla birlikte çalışan sürüm 6 kuyruk yöneticilerindeki yayınlama/abone olma uygulamalarını sürdürmeye devam edebilirsiniz.
2. Uygulamaları tümleşik yayınlama/abone olma MQI arabirimine geçirmek yerine, daha sonraki sürüm kuyruğu yöneticilerindeki kuyruklanmış yayınlama/abone olma uygulamalarınızı geliştirmeye devam edebilirsiniz.

3. Akışların konuları varsayılan olarak eşleştirmesi, konu alanında "çarpışma" a yol açar ve bir akıştaki yayınların konu dizgisinin başka bir yerden yayınlarla aynı olması gerekir.

Yetkiler

Varsayılan olarak, konu ağacının kökünde birden çok konu nesnesi vardır: SYSTEM.BASE.TOPIC, SYSTEM.BROKER.DEFAULT.STREAM ve SYSTEM.BROKER.DEFAULT.SUBPOINT. Yetkililer (örneğin, yayınlama ya da abone olma için) SYSTEM.BASE.TOPIC üzerindeki yetkiler tarafından belirlenir; SYSTEM.BROKER.DEFAULT.STREAM ya da SYSTEM.BROKER.DEFAULT.SUBPOINT üzerindeki tüm yetkiler yoksayılar. SYSTEM.BROKER.DEFAULT.STREAM ya da SYSTEM.BROKER.DEFAULT.SUBPOINT silinir ve boş olmayan bir konu dizgisiyle yeniden yaratılırsa, bu nesnelere üzerinde tanımlanan yetkiler, normal bir konu nesnesiyle aynı şekilde kullanılır.

Akışlar ve konular arasında eşleme

Kuyruğa alınmış bir yayınlama/abone olma akışı, Version 7.0 ' ta bir kuyruk yaratılarak ve akıyla aynı adı verilerek taklit edilir. Bazen kuyruğun adı akış kuyruğu olarak adlandırılır, çünkü bu şekilde kuyruğa yollanan yayınlama/abone olma uygulamaları söz edilir. The queue is identified to the publish/subscribe engine by adding it to the special namelist called SYSTEM.QPUBSUB.QUEUE.NAMELIST. Ad listesine ek özel kuyruklar ekleyerek, gereksinim duyarken istediğiniz sayıda akış ekleyebilirsiniz. Son olarak, akışlar ve akış adıyla aynı konu dizgileriyle aynı adı taşıyan konular eklemeli ve bu konuları yayınlatabilmeniz ve konulara abone olmanız gerekir.

Ancak, istisnai durumlarda, konuları tanımlarken seçtiğiniz konulara ilişkin konulara karşılık gelen konuları da verebilirsiniz. Konu dizgisinin amacı, konuya konu alanında benzersiz bir ad vermeğidir. Genellikle akış adı bu amaca mükemmel bir şekilde hizmet eder. Bazen, bir akış adı ve var olan bir konu adı çarpıştırır. Sorunu çözmek için akımla ilişkili konu için başka bir konu dizgisi seçebilirsiniz. Herhangi bir konu dizgisi seçin; bunun benzersiz olmasını sağlayın.

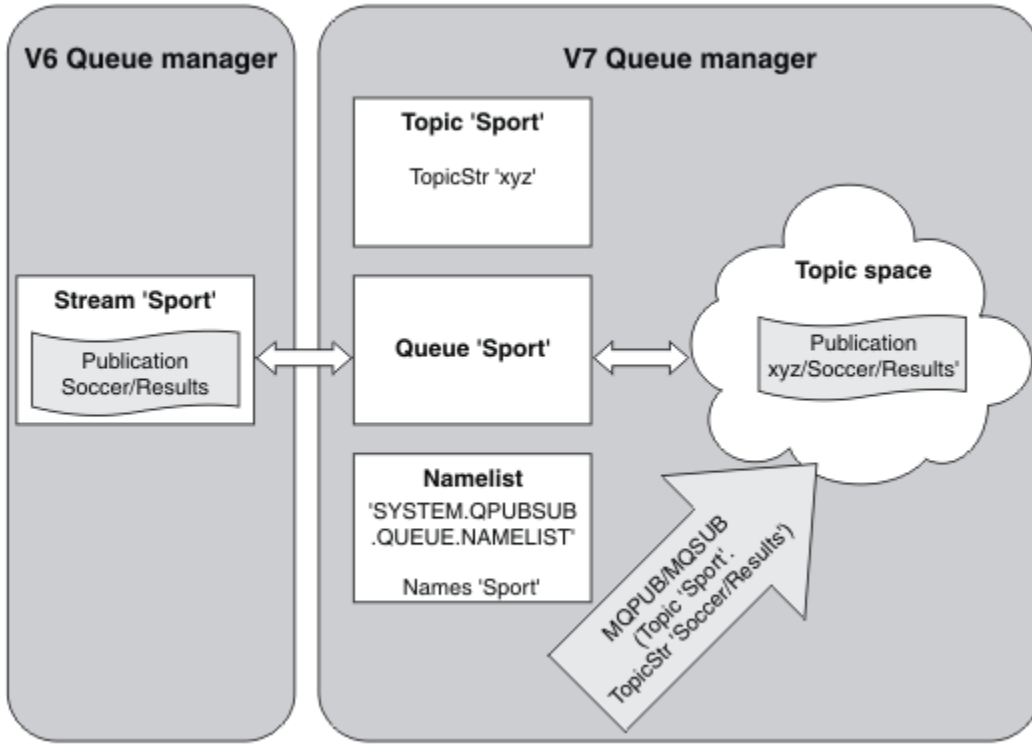
Konu tanımında tanımlanan konu dizgisinin MQOPEN ya da MQSUB MQI çağrılarını kullanılarak yayıncılar ve aboneler tarafından sağlanan konu dizgisine olağan şekilde önekli olması gerekir. Konu nesnelerini kullanan konulara başvuran uygulamalar, örnek konusu dizgisinin seçiminden etkilenmez; bu nedenle, yayınların konu alanında benzersiz olmasını sağlayan herhangi bir konu dizgisini seçebilirsiniz.

Farklı konulara ilişkin farklı akışların yeniden eşlenmesi, konu dizgileri için kullanılan örnekleri, bir konu kümesini diğerinden ayırmak için kullanılan örnekleri kullanır. Eşlemenin çalışması için rivayla uyulması gereken evrensel bir konu adlandırma kuralı tanımlamanız gerekir. Version 7.0' da, konu dizgileri çarpıştıysa, konu alanlarını ayırmak için akışlar kullanabilirsiniz. Version 7.0 olanağından başlayarak, bir konu dizgisini konu alanındaki başka bir yere yeniden eşlemek için önceden düzeltme düzeneğini kullanıyorsanız.

Not: Bir akışı sildiğinizde, önce akıştaki tüm abonelikleri silin. Aboneliklerin herhangi biri, aracı sıradüzenindeki diğer araçlardan kaynaklanırsa, bu işlem en önemlidir.

Örnek

In Şekil 18 sayfa 46, topic 'Sport' has the topic string 'xyz' resulting in publications that originate from stream 'Sport' being prefixed with the string 'xyz' in the version 7 queue manager topic space. Publishing or subscribing in version 7 to the topic 'Sport' prefixes 'xyz' to the topic string. Yayın bir sürüm 6 abonesine akıyorsa, 'xyz' öneki yayından kaldırılır ve 'Sport' akımında yerleştirilir. Ters durumda, bir yayın düzeyi 6 sürümünden 7 sürümüne, 'Sport' akışından 'Sport' konusuna, 'xyz' öneki konu dizgisine eklenir.



Şekil 18. Sürüm 7 konularıyla birlikte var olan sürüm 6 akışları

Abonelik noktaları ve konular

WebSphere MQ Event Broker ve Message Broker 'da belirli bir yayın düğümleri kümesinden yayınları istemek için kullanılan bir abonelik noktası. Adlandırılmış abonelik noktaları, konular ve konu nesnelere tarafından öykünülmektedir.

WebSphere MQ Event Broker V6.0 - WebSphere MQ V7.0.1 geçiş yordamı (**migmbbrk**), adlandırılmış abonelik noktalarını konulara ve konu nesnelere dönüştürür. Bir abonelik noktası, alıkonan bir yayın ya da kayıtlı bir aboneye sahipse otomatik olarak geçirilir. **migmbbrk**, adlandırılmış abonelik noktalarından konu nesnelere yaratır. Abonelik noktasının adı, konu nesnesinin adı ve konu dizgisinin kendisi olur. Konu nesnesi 'SYSTEM.QPUBSUB.SUBPOINT.NAMELIST' e eklenir.

Aynı adı taşıyan bir konu nesnesi varsa, **migmbbrk** iki şeyden birini gerçekleştirir.

1. Konu nesnede farklı bir konu dizgisi varsa ya da abonelik noktası adı bir nesne adından daha uzunsa, **migmbbrk**, üretilmiş bir adla bir konu nesnesi yaratır.
2. Konu nesnesi aynı konu dizgisine sahipse, **migmbbrk** var olan nesneyi ad listesine ekler.

Abonelik noktalarını el ile eklemek için [Abonelik noktası eklenmesibaşlıklı konuya](#) bakın.

WebSphere MQ Event Broker 'daki abonelik noktaları

Yayınlama düğümleri, iletileri süzmek ve abonelere iletmek için bir WebSphere MQ Event and Message Broker ileti akışında kullanılır. Yayıncılar genellikle yayın düğümlerinde abonelik noktaları belirlemez. Aboneler belirli bir konu kümesine ilgi gösterir ve genellikle abonelik puanlarını belirtmez.

Abonelik noktası, hangi yayın düğümlerinin bir aboneliğe iletileceğini seçmenin bir yoludur. Abone, bir abonelik noktası adı ile ilgili bir dizi konu ile ilgilerini niteliyor.

Abonelik noktası adını ayarlamak için, yayın düğümünün **Subscription point** özelliğine bir ad atayın.

Abonelik noktası özelliği, bir konuya ilişkin bir yayının abonelere aynı konuya iletilip iletilmediğini denetler. Adlandırılmış bir abonelik noktasına sahip yayın düğümlerinden gelen yayınlar, yalnızca aynı abonelik

noktasına aboneler için iletilir. Adlandırılmış bir abonelik noktası olmayan yayın düğümlerinden gelen yayınlar, varsayılan olarak yalnızca, bir abonelik noktası adı vermemiş abonelere iletilir.

Nodes with a named subscription point send Publish command messages in MQRFH2 format, with the **SubPoint** property set. Adlandırılmış bir abonelik noktasına yönelik abonelikler, MQRFH2 Register subscriber komut iletilisinde **SubPoint** özelliğini ayarlamalıdır.

Subscription points in WebSphere MQ

WebSphere MQ , abonelik noktalarını WebSphere MQ konu ağacında farklı konu alanları ile eşler. Topics in command messages without a subscription point are mapped unchanged to the root of the WebSphere MQ topic tree and inherit properties from SYSTEM.BASE.TOPIC.

Bir abonelik noktası içeren komut iletileri, SYSTEM.QPUBSUB.SUBPOINT.NAMELISTiçindeki konu nesnelere listesini kullanarak işleme devam eder. Komut iletilisinde abonelik noktası adı, listedeki konu nesnelere her biri için konu dizgisine göre eşleştirilir. Bir eşleşme bulunursa, bir konu düğümü olarak, abonelik noktası adı konu dizgisine önceden işaretlenmiş olur. Konu, özelliklerini SYSTEM.QPUBSUB.SUBPOINT.NAMELISTiçinde bulunan ilişkili konu nesnesinden edinir.

Abonelik noktalarını kullanmanın etkisi, her bir abonelik noktası için ayrı bir konu alanı yaratmasıdır. Konu alanı, abonelik noktasıyla aynı adı taşıyan bir konuya kök saldı. Her konu alanındaki konular, özelliklerini, abonelik noktasıyla aynı adı taşıyan konu nesnesinden devralır.

Any properties not set in the matching topic object are inherited, in the normal fashion, from SYSTEM.BASE.TOPIC.

Var olan kuyruğa alınmış yayınlama/abone olma uygulamaları, MQRFH2 ileti üstbilgilerini kullanarak, Publish ya da Register subscriber komut iletilerinde **SubPoint** özelliğini ayarlayarak çalışmaya devam eder. Abonelik noktası, komut iletilisinde konu dizgisiyle birleştirilir ve sonuçta ortaya çıkan konu herhangi bir diğer konu gibi işlenir.

Yeni bir WebSphere MQ V7 uygulaması abonelik noktalarından etkilenmez. Eşleşen konu nesnelere birinden edinen bir konu kullanılırsa, bu nesne, eşleşen abonelik noktasını kullanarak kuyruğa alınan bir uygulamayla birlikte çalışır.

Örnek

Var olan bir WebSphere MQ Event Broker yayınlama/abone olma uygulaması, farklı para birimlerindeki paylaşım fiyatlarını yayınlamak için abonelik puanlarını kullanır. The dollar spot price of the IBM stock is published using the subscription point USD, and the topic NYSE/IBM/SPOT. The sterling price is published using the same topic, and the subscription point GBP.

The migration procedure on WebSphere MQ creates two topic objects, GBP and USD, with the corresponding topic strings 'GBP' and 'USD'.

Existing publishers to the topic NYSE/IBM/SPOT, migrated to run on WebSphere MQ, that use the subscription point USD create publications on the topic USD/NYSE/IBM/SPOT. Similarly existing subscribers to NYSE/IBM/SPOT, using the subscription point USD create subscriptions to USD/NYSE/IBM/SPOT.

Subscribe to the dollar spot price in a version 7 publish/subscribe program by calling MQSUB. Create a subscription using the USD topic object and the topic string 'NYSE/IBM/SPOT', as illustrated in the 'C' code fragment.

```
stncpy(sd.ObjectName, "USD", MQ_TOPIC_NAME_LENGTH);
sd.ObjectString.VSPtr = "NYSE/IBM/SPOT";
sd.ObjectString.VSLength = MQVS_NULL_TERMINATED;
MQSUB(Hconn, &sd, &Hobj, &Hsub, &CompCode, &Reason);
```

Consider whether your WebSphere MQ Event Broker applications in the collective always used the subscriptions points USD and GBP. Bunu yapsalar, küme konusu anasistemdeki küme konuları olarak yalnızca bir kez USD ve GBP konu nesnelere yaratın. You do not need to carry out step [../com.ibm.mq.mig.doc/q007670_.dita#q007670_/clusterstep](http://com.ibm.mq.mig.doc/q007670_.dita#q007670_/clusterstep) of the migration procedure to change

SYSTEM.BASE.TOPIC, on every queue manager in the cluster, to a cluster topic. Bunun yerine, aşağıdaki adımları gerçekleştirin:

1. Küme konusu ana makinesinde USD ve GBP konu nesnelerinin CLUSTER özniteliğini ayarlayın.
2. Kümedeki diğer kuyruk yöneticilerindeki USD ve GBP konu nesnelerinin tüm kopyalarını silin.
3. Make sure that USD and GBP are defined in SYSTEM.QPUBSUB.SUBPOINT.NAMELIST on every queue manager in the cluster.

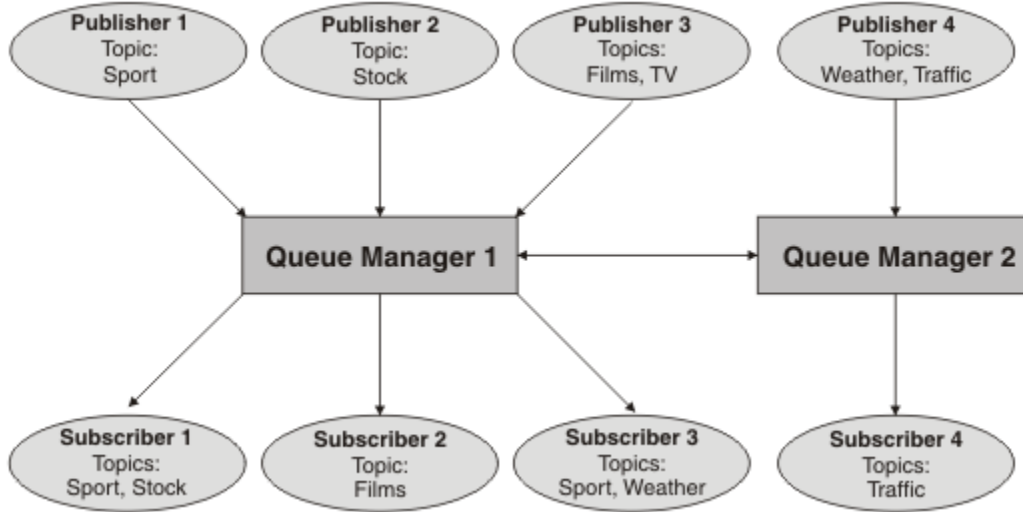
Dağıtılmış yayınlama/abone olma

Bu bölümde, kuyruk yöneticileri, kümeler ve sıradüzenler arasında bağlantı kurmak için kullanılabilir iki farklı kuyruk yöneticisi topolojisini ve kuyruk yöneticileri arasında yayınlama/abone olma iletilerinin nasıl gerçekleştirilebileceği ile ilgili bilgiler bulunur.

Queue managers can communicate with other queue managers in your WebSphere MQ publish/subscribe system, so that subscribers can subscribe to one queue manager and receive messages that were initially published to another queue manager. Bu, [Şekil 19 sayfa 48](#)' ta gösterilmektedir.

[Şekil 19 sayfa 48](#) , iki kuyruk yöneticisine sahip bir yayınlama/abone olma sistemi gösterir.

- Kuyruk yöneticisi 2, yayınlayıcı 4 tarafından hava durumu tahmini bilgilerini yayınlamak için, Hava durumu konusunu ve trafik durumu hakkında trafik durumunu kullanarak, trafik durumu hakkında bilgi yayınlatabilirsiniz.
- Abone 4, bu kuyruk yöneticisini kullanır ve trafik koşulları ile ilgili bilgileri konu trafiğini kullanarak bilgilendirir.
- Abone 3, yayınlayıcıdan farklı bir kuyruk yöneticisi kullansa da hava durumu koşullarına ilişkin bilgileri de abone eder. Bunun nedeni, kuyruk yöneticilerinin birbiriyle bağlantılı olması olabilir.



Şekil 19. İki kuyruk yöneticisi içeren yayınlama/abone olma örneği

Dağıtılmış yayınlama/abone olma işi nasıl çalışır?

WebSphere MQ yayınlama/abone olma, abonelerin uzak kuyruk yöneticilerine yayınlanan iletileri alabilmesi için yetkili abonelikleri kullanır.

Dağıtılmış yayınlama/abone olma, kuyruk yöneticilerine ilişkin ağları bağlamak ve bunun sonucunda kuyruk yöneticilerine bağlanan uygulamalar için dağıtılmış kuyruğa alma işlemi ile aynı bileşenleri kullanır. Kuyruk yöneticileri ile kuyruk yöneticileri arasında bağlantı oluşturma arasındaki bileşenler arasında ileti alışverişi hakkında daha fazla bilgi almak için *Intercommunication* (İletişim) belgelerine bakın.

Abonelerin, dağıtılmış bir yayınlama/abone olma sisteminde standart abonelik işlemi dışında hiçbir şey yapmaması gerekir. Bir kuyruk yöneticisinde bir abonelik yapıldığında, kuyruk yöneticisi, aboneliğin bağlı kuyruk yöneticilerine dağıtıldığı işlemi yönetir. Yetkili abonelikler, ağdaki tüm kuyruk yöneticilerine veri

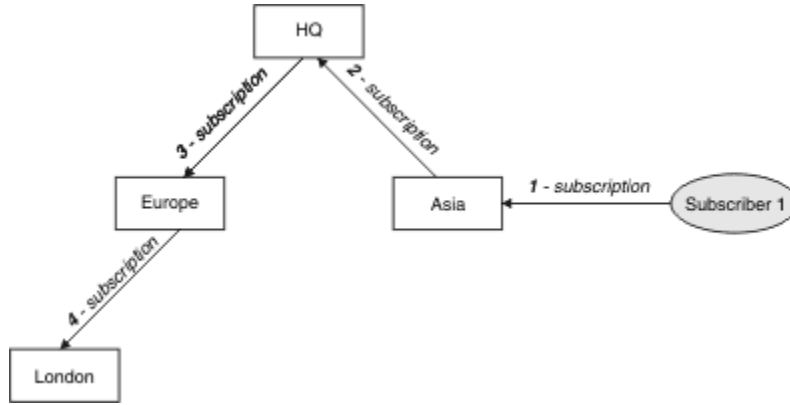
akışı sağlar. Bunlar, yayınların özgün aboneliğin yaratıldığı kuyruk yöneticisine geri yönlendirildiğinden emin olmak için yaratılırlar; bkz. Şekil 20 sayfa 49.

Bir yayın, uzak kuyruk yöneticisinde o uzak kuyruk yöneticisinde varsa, yayın uzak bir kuyruk yöneticisine dağıtılır.

Kuyruk yöneticisi, yerel uygulamalardan ya da uzak kuyruk yöneticilerinden gelen tüm abonelikleri birleştiriyor. Bir abonelik var olmadığı sürece, komşuları ile aboneliklerin konuları için yetkili abonelikler oluşturur; bkz. Şekil 21 sayfa 49.

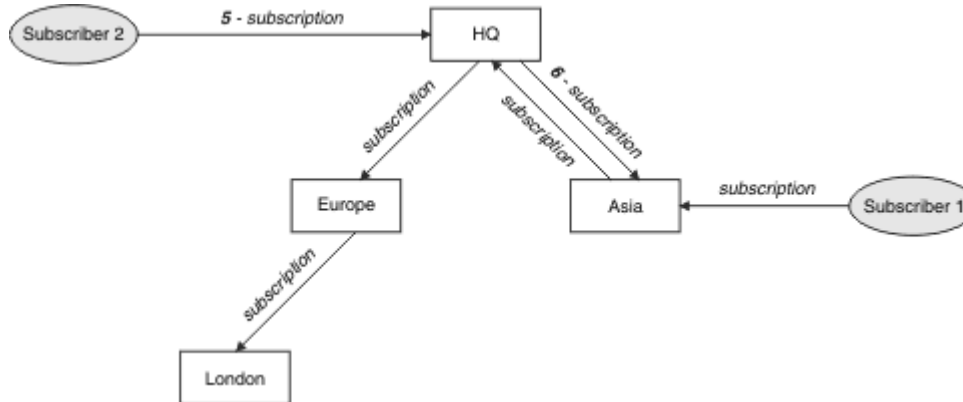
Bir uygulama bilgileri yayınlarken, alan kuyruk yöneticisi bunu uzak kuyruk yöneticilerindeki geçerli abonelikleri olan uygulamalara iletir. Bunu bir ya da daha çok ara kuyruk yöneticisi aracılığıyla iletebilir; bkz. Şekil 22 sayfa 50.

Abone 1, Asya kuyruk yöneticisinde (1) belirli bir konuya ilişkin aboneliği kaydeder. Bu konuya ilişkin abonelik, ağdaki diğer tüm kuyruk yöneticilerine iletilir (2,3, 4).



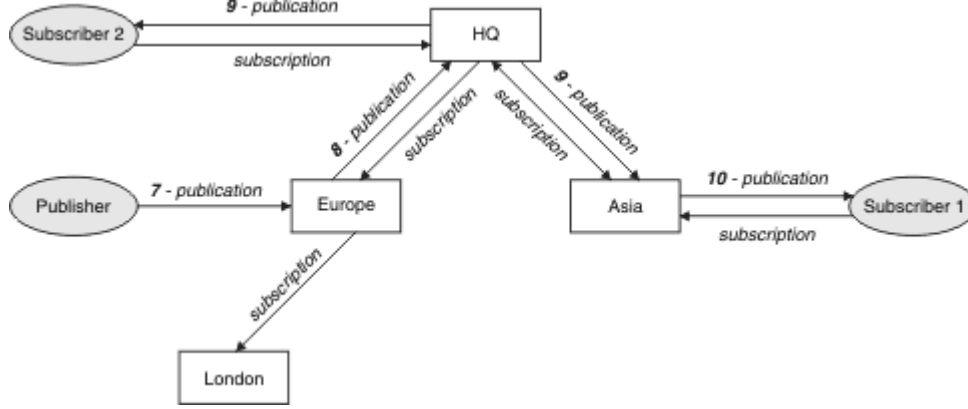
Şekil 20. Bir kuyruk yöneticisi ağı aracılığıyla aboneliklerin yayılması

Subscriber 2 registers a subscription, to the same topic as in Şekil 20 sayfa 49, on the HQ queue manager (5). Bu konuya ilişkin abonelik, Asya kuyruk yöneticisine iletilir; böylece, aboneliklerin ağ üzerinde başka bir yerde (6) yer aldığından haberdar olur. Abonelik, bu konuya ilişkin bir abonelik zaten kayıtlı olduğundan, Avrupa kuyruk yöneticisine iletilmedi; Şekil 20 sayfa 49 içindeki 3. adıma bakın.



Şekil 21. Birden çok abonelik

Yayıncı, Şekil 21 sayfa 49 ile aynı konu üzerinde, Avrupa kuyruk yöneticisine (7) bir yayın gönderir. Bu konuya ilişkin abonelik, HQ 'dan Avrupa 'ya varolduğundan, yayın, HQ kuyruk yöneticisine iletilir (8). Ancak, Londra 'dan Avrupa 'ya (yalnızca Avrupa 'dan Londra 'ya) abonelik yoktur, bu nedenle yayın, Londra kuyruk yöneticisine iletilmez. HQ kuyruk yöneticisi yayını doğrudan abone 2 'ye ve Asya kuyruk yöneticisine (9) gönderir. Yayın, Asya 'dan (10) abone 1 'e iletilir.



Şekil 22. Bir kuyruk yöneticisi ağı aracılığıyla yayınların yayılması

Bir kuyruk yöneticisi başka bir kuyruk yöneticisine herhangi bir yayın ya da abonelik gönderdiğinde, ileti içinde kendi kullanıcı kimliğini ayarlar. Bir yayınlama/abone olma sıradüzeni kullanıyorsanız ve gelen kanal, iletilerde kullanıcı kimliği yetkisi olan iletileri koymak üzere ayarlandıysa, gönderen kuyruk yöneticisinin kullanıcı kimliğini yetkilendirmeniz gerekir; bkz. [“Kuyruk yöneticisi sıradüzeniyle varsayılan kullanıcı kimliklerini kullanma” sayfa 103](#). Yayınlama/abone olma kümeleri kullanıyorsanız, yetkilendirme küme tarafından işlenir.

Yayınlama/abone olma kuyruğu yöneticilerinin birbiriyle bağlantılı niteliği nedeniyle, yetkili aboneliklerin ağdaki tüm düğümlere yayılması zaman alır. Uzak yayınların hemen hemen abone olunması gerekmez. Abonelik gecikmesini, [“Yönelme mekanizmalarında daha fazla” sayfa 51](#) içinde açıklandığı şekilde FORCE değeriyle PROXYSUB **Topic** özniteliğini kullanarak ortadan kaldırebilirsiniz.

Abonelik işlemi, her bir doğrudan bağlı kuyruk yöneticisi için uygun iletim kuyruğuna yetkili sunucu aboneliği konduğunda sonlandırır. Abonelik işlemi, yetkili sunucu aboneliğinin topolojinin geri kalanına yayılmasını beklemeyiz.

Yetkili sunucu abonelikleri, bunları yaratan kuyruk yöneticisi adıyla ilişkilendirilir. Sıradüzendeki kuyruk yöneticileri aynı ada sahip olursa, yayınların erişememesine yol açabilir. Noktadan noktaya ileti alışverişi gibi bu sorunu önlemek için, özellikle doğrudan ya da dolaylı olarak bir WebSphere MQ ağına bağlanılmışsa, kuyruk yöneticilerine benzersiz adlar verin.

Dağıtılmış bir yayınlama/abone olma ağı içinde, yayınların ve aboneliklerin akışı denetlenebilir ve uygun, sınırlı, yayın ve abonelik kapsamı kullanılarak sınırlandırılabilir.

Yetkili abonelik toplama ve yayın toplama

Yayınlama/abone olma kuyruk yöneticileri arasında geçen ileti miktarını en aza indirmek için dağıtılmış yayınlama/abone olma yayınları ve yetkili sunucu abonelikleri toplanır.

Yetkili sunucu aboneliği, başka bir kuyruk yöneticisinde yayınlanan konular için bir kuyruk yöneticisi tarafından yapılan bir aboneliklerdir. Yetkili sunucu aboneliklerini belirttik olarak yaratmadığınızda, kuyruk yöneticisi sizin adınıza bunu yapar; bkz. [“Dağıtılmış yayınlama/abone olma işi nasıl çalışır?” sayfa 48](#).

Kuyruk yöneticilerini bir yayınlama/abone olma sıradüzenine ya da bir yayınlama/abone olma kümesine bağlayabilirsiniz. Yetkili sunucu abonelikleri, bağlı kuyruk yöneticileri arasında akar. Yetkili abonelikler, abonelerin diğer kuyruk yöneticilerine bağlı olan bu konuya abone olarak bir kuyruk yöneticisine bağlı bir yayıncı tarafından oluşturulan bir konuya ilişkin yayınlara neden olur; bkz. [“Yayınlama/abone olma topolojisi” sayfa 60](#).

Yetkili abonelik, bir abonelik tarafından abone olunan her bir konu dizisine ilişkin kuyruk yöneticileri arasında akışları içerir.

You can restrict the flow of proxy subscriptions and publications between connected queue managers using the **Topic** attributes `YAYINLAMA` and `ALT_KAPSAM`. You can also restrict the flow of proxy subscriptions containing wildcards by setting the **Topic** attribute `JOKER` to `BLOCK`; see [“Joker kuralları” sayfa 53](#).

Yetkili sunucu abonelikleri, aboneliklerin yaratılmasına zamanuyumsuz olarak kuyruk yöneticileri arasında akıtılır. You can reduce the latency of waiting for a proxy subscription to be propagated to all the connected queue managers, by setting the **Topic** attribute `PROXYSUB` to `FORCE` on the topic, or a parent of a topic that will be subscribed to; see [“Yönelme mekanizmalarında daha fazla” sayfa 51](#).

Yetkili sunucu aboneliği toplaması

Yetkili sunucu abonelikleri, yinelenen bir eleme sistemi kullanılarak toplanır. Çözömlenen bir konu dizisi için, ilk yerel abonelik ya da alınan yetkili abonelik için bir yetkili sunucu aboneliği gönderilir. Aynı konu dizisine ilişkin sonraki abonelikler, bu var olan yetkili aboneliğin kullanımını sağlar.

Son yerel abonelik ya da alınan yetkili abonelik iptal edildikten sonra yetkili sunucu aboneliği iptal edilir.

Bireysel konu dizilerine binlerce abonelik içeren ya da bu aboneliklerin varlığının hızla değişebileceği, yayınlama/abone olma/abone olma topolojilerinde, yetkili sunucu aboneliği yayılımın ek yükü göz önünde bulundurulmalıdır. The individual proxy subscriptions can be consolidated through the use of the topic attribute **PROXYSUB** being set to `FORCE`. Yönelme mekanizmaları ve küme konusu performansıyla ilgili daha fazla ayrıntı için bkz. [“Yönelme mekanizmalarında daha fazla” sayfa 51](#).

Yayın toplaması

Bir kuyruk yöneticisinde aynı konu dizisine birden fazla abonelik varsa, yayınlama/abone olma topolojisindeki diğer kuyruk yöneticilerinden bu konu dizisiyle eşleşen her bir yayın ile eşleşen her bir yayının tek bir kopyası gönderilir. İletinin gelmesiyle, yerel kuyruk yöneticisi iletinin bir kopyasını her bir eşleşen aboneliğe aktarır.

Yetkili abonelikler joker karakterler içerdiğinde tek bir yayının konu dizisiyle eşleşmesi için birden çok yetkili sunucu aboneliği olabilir. Bir ileti, bağlı tek bir kuyruk yöneticisi tarafından yaratılan iki ya da daha fazla yetkili sunucu aboneliğine uyan bir kuyruk yöneticisinde yayınlandıysa, birden çok yetkili sunucu aboneliğini karşılamak için uzak kuyruk yöneticisine yayının yalnızca bir kopyası uzak kuyruk yöneticisine iletilir.

Yönelme mekanizmalarında daha fazla

Her yerde yayınla , tek tek yetkili abonelik iletimi için alternatif bir yönlendirme mekanizmasıdır. Bireysel yetkili sunucu aboneliği iletimi, yalnızca konu dizisinde eşleşen bir aboneliğin olduğu yayınların uzak bir ileti alışverişi sunucusuna gönderileceği anlamına gelir. Bir ileti alışverişi sunucusunda yayınlanan tüm yayınları, dağıtılmış bir yayınlama/abone olma ağındaki diğer tüm ileti sistemi sunucularına ileterek her yerde yayınlayın ya da yayınlayın, bu yayınların tümünü yayınlayın. Alıcı ileti sistemi sunucuları, yerel aboneliklerle eşleşen yayınları sağlar.

Her mekanizmanın fazları vardır, ama aynı zamanda sınırlandırmalar da vardır.

Bireysel yetkili sunucu aboneliği iletilme

Bu mekanizma, yalnızca bir kuyruk yöneticisinde aboneliklerle eşleşen yayınlar gönderildiğinden, kuyruk yöneticisi yayın trafiğindeki en az miktarda sonuçlar elde eder.

Ancak:

- Yayınlama/abone olma topolojisindeki diğer tüm kuyruk yöneticilerine gönderilen bir yetkili sunucu aboneliğine abone olunan her bir konu dizisidir. Yaratmak ya da silmek için (örneğin, bir kuyruk yöneticisinin yeniden başlatılmasından sonra tüm kalıcı olmayan abonelikler) ya da abonelikler kümesi hızla değişiyorsa ve her biri farklı bir konu dizisine sahip olduğunda, bu ileti alışverişi genel giderleri önemli olabilir.

- Yetkili sunucu abonelikleri, zamanuyumsuz ileti sistemi kullanan diğer kuyruk yöneticilerine aktarıldığından, diğer kuyruk yöneticileri tarafından abonelik oluşturma ve yetkili abonelik oluşturma, teslim alma ve işleme arasında bir gecikme vardır. Bu aralıktaki kuyruk yöneticilerinde yayınlanan iletiler, uzak aboneliğe teslim edilmez.

Her yerde yayınlama

Bu düzenek ile:

- Sistem üzerinde hızlı abonelik yaratma, silme ya da değiştirme anlamına gelen, sistem üzerinde herhangi bir konu dizgi yetkili sunucusu abonelik ek yükü yoktur; bu, ağ yükü ve işlemenin artırılmasıyla sonuçlanmaz.
- Bir kuyruk yöneticisine her zaman tüm kuyruk yöneticilerine aktarıldığı için, abonelik ve yayınların kuyruk yöneticisine akıtılması arasında gecikme yoktur. Bu nedenle, yayınların yeni oluşturulan uzak aboneliklere teslim edilmediği bir pencere yoktur.

Ancak:

- Tüm yayınlar yayınlama/abone olma topolojisindeki tüm kuyruk yöneticilerine gönderilir. Bu, yayınların her bir kuyruk yöneticisinde eşleşen aboneliklere sahip olmadığı aşırı ağ trafiğiyle sonuçlanabilir.

Bir yayının, küme ya da sıradüzendeki kuyruk yöneticilerinizin önemli bir oranından abone olmasını beklediğinizde ya da abonelik değişikliklerinin sıklığından dolayı yetkili sunucu aboneliği üst başlarının çok büyük olduğu bir yayının abone olmasını istiyorsanız, her yerde yayınlama mekanizmasını kullanmak isteyebilirsiniz. Bu çalışma yöntemi, abone abonelikleri olan kuyruk yöneticilerine değil, tüm kuyruk yöneticilerine gönderildiklerinde, tüm kuyruk yöneticilerine gönderildiklerinde, ileti trafiğini artırdığınız diğer kişilere göre bu eşgörünümde daha etkili bir çalışma yöntemi olabilir.

A publish everywhere mechanism can be enabled in IBM WebSphere MQ distributed publish/subscribe topologies by setting the **PROXYSUB** attribute to FORCE for a high-level topic object.

Bireysel yetkili aboneliklerin devre dışı bırakılmasına ilişkin ayrıntılar için bkz. [“Yetkili sunucu aboneliklerinin geçersiz kılınması” sayfa 73.](#)

Bu zorlamalı yetkili sunucu aboneliği topoloji boyunca yayıldığında, yeni abonelikler hemen bağlı olan diğer kuyruk yöneticilerinden herhangi bir yayın almasına izin verilmeden, gecikme süresi dolmadan, bu abonelikleri alır.

Böyle bir sistemi yapılandırırken dikkatli olmanız gerekir. No topic objects below a topic with **PROXYSUB** set to FORCE must be in a different cluster or hierarchy stream to the node where **PROXYSUB** is set to FORCE. Benzer şekilde, alt konu nesnelere **WILDCARD** özniteliğini BLOCK olarak ayarlamaması gerekir. Her iki durumda da, yayınlanan iletiler bir kuyruk yöneticisinden başka bir doğru olarak doğru akıp gitmeyebilir.

PROXYSUB FORCE değerine ayarlandığında bile, abone olunan her bir konu dizisine ilişkin bir yetkili abonelik yayılmaya devam eder. Aboneliklerin sayısı ve sıklığı, sistemdeki önemli giderlere yol açabilecek kadar yüksekse, bir kuyruk yöneticisindeki tüm konular için devre dışı bırakılabilir. Bireysel yetkili aboneliklerin devre dışı bırakılmasına ilişkin ayrıntılar için bkz. [“Yetkili sunucu aboneliklerinin geçersiz kılınması” sayfa 73.](#)

Çok noktaya gönderim ve abonelik gecikme süresi

Bir yetkili aboneliği korumak için abonelik gecikme süresi ve PROXYSUB (FORCE) seçeneği kullanılabilir.

Örneğin, tüm abonelerin bağlantısı kesildikten sonra, QM_B 'den QM_A' ya bir yetkili sunucu aboneliği olası sorunu olabilir. Kuyruk yöneticisine tek hedefli bağlantı sonlandığında bile çoklu yayın trafiğinin devam etmesini istiyorsanız, bu durum istenmeyebilir. WebSphere MQ çoklu yayını, yeni bir abone bağlandığında, her yetkili sunucu aboneliğine birkaç dakika gecikme süresi ekleyerek, son abonenin son abonesinin sona erdirileceği anda yapılmaması durumunda, kısa bir süre için yetkili aboneliği sürdürür.

Olağanüstü bir yetkili sunucu aboneliğinin her zaman olağanüstü olmasına dikkat etmek için, konuyla ilgili PROXYSUB (FORCE) seçeneğini de kullanabilirsiniz. Kuyruklar boyunca akan iletilerin, aboneliğin etkin olduğu sürenin büyük kısmı için en az bir abone gerekli olduğundan emin olmalısınız. PROXYSUB

(FORCE) değeri belirlendiyse, bir yetkili sunucu aboneliği ilk yerel abonelik ya da alınan yedek sunucu aboneliğinden önce gönderilebilir; son yerel abonelik ya da alınan yetkili abonelik iptal edildikten sonra da iptal edilmeyecektir.

Abonelik hala yapılmazsa, ileti aktarımlarının devam etmesini sağlamak için eşdüzeyler arası iletişim kullanılabilir; daha fazla bilgi için bkz. [Çok noktaya gönderim için yüksek kullanılabilirlik](#).

Joker kuralları

Yetkili sunucu aboneliklerinde genel arama karakterleri, konu genel arama karakterlerini kullanacak şekilde dönüştürülür.

If a subscription for a wildcard is received, it can be a character, as used by WebSphere MQ Version 6.0. Ayrıca, WebSphere Message Broker Sürüm 6.0 ve WebSphere MQ Sürüm 7.0 tarafından kullanılan bir konu da olabilir.

- Karakter genel arama karakterleri, / ile birlikte herhangi bir karakteri temsil etmek için * kullanır.
- Topic wildcards use # to represent a portion of the topic space between / characters.

WebSphere MQ Sürüm 7.0' da, tüm yetkili abonelikler konu genel arama karakterlerini kullanacak şekilde dönüştürülür. Bir karakter genel arama karakteri bulunursa, bu genel arama karakteri bir # karakteriyle değiştirilir; en yakın / karakterine geri döner. Örneğin, /aaa/bbb/c*d , /aaa/bbb/#olarak dönüştürülür. Uzak kuyruk yöneticilerindeki dönüştürme sonuçları, belirtik olarak abone olunandan biraz daha fazla yayın göndermektedir. Bu ek yayınlar, yayınları yerel abonelerine teslim ettiğinde, yerel kuyruk yöneticisi tarafından dışarı süzölmüş olur.

Controlling wildcard usage with the JOKER property

Yayınların genel arama karakteri konu dizgisi adlarını kullanan abone uygulamalarına teslim edilmesini denetlemek için MQSC **Topic JOKER** özelliğini ya da eşdeğer PCF Topic WildcardOperation özelliğini kullanın. JOKER özelliği, iki olası değerden birine sahip olabilir:

Genel arama karakteri

Bu konuya ilişkin genel arama karakteri abonelikleri davranışı.

Passthru

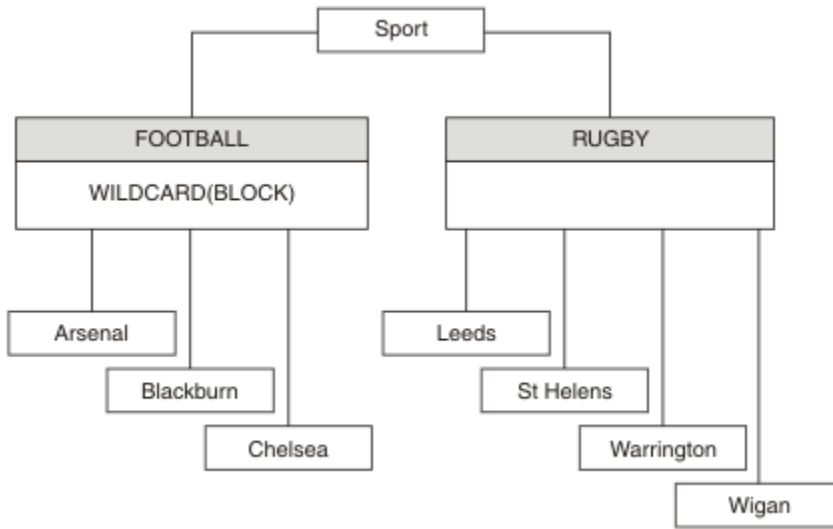
Bu konu nesnesindeki konu dizgisinden daha az belirli bir genel konuya özgü olan abonelikler, bu konuda yapılan yayınları ve bu konuya ilişkin dizgileri daha belirli bir konuya getirmektedir.

Öbek

Bu konu nesnesindeki konu dizgisinden daha az belirli bir genel konuya özgü olan abonelikler, bu konuya ya da bu konudan daha belirli bir konuya ilişkin dizgileri elde edemeyecek şekilde yayınlar.

Abonelikler tanımlandığında bu özniteliğin değeri kullanılır. Bu özniteliği değiştirirseniz, var olan aboneliklerle kapsanan konu kümesi değişiklikten etkilenmez. Bu senaryo, konu nesnelere yaratıldığında ya da silindiğinde topoloji değişirse de geçerli olur; JOKER özniteliğinin değiştirilmesinin ardından oluşturulan aboneliklerle eşleşen konu kümesi, değiştirilen topoloji kullanılarak oluşturulur. Var olan abonelikler için eşleşen konular kümesini yeniden değerlendirmeye zorlamak istiyorsanız, kuyruk yöneticisini yeniden başlatmanız gerekir.

In the example, "[Örnek: Sport yayınlama/abone olma kümesini yaratın](#)" sayfa 41, you can follow the steps to create the topic tree structure shown in [Şekil 10 sayfa 38](#).



Şekil 23. JOKER özelliğini kullanan bir konu ağacı, BLOCK

A subscriber using the wildcard topic string # receives all publications to the Sport topic and the Sport/Rugby subtree. The subscriber receives no publications to the Sport/Football subtree, because the JOKER property value of the Sport/Football topic is BLOCK.

Varsayılan değer PASSTHRU ' dir. Sport ağacındaki düğümlere JOKER özellik değerini PASSTHRU olarak ayarlayabilirsiniz. Düğümlerde BLOCK JOKER özellik değeri yoksa, PASSTHRU ayarı abonelerin Sports ağacındaki düğümlere göre gözlemlendiği davranışı değiştirmez.

Örnekte, genel arama karakterinin, teslim edilen yayınları nasıl etkilediğini görmek için abonelikler oluşturun; bkz. Şekil 14 sayfa 43. Bazı yayınlar yaratmak için Şekil 17 sayfa 44 içinde yayınlama komutunu çalıştırın.

pub QMA

Şekil 24. QMA ' ya Yayınla

Sonuçlar, Çizelge 3 sayfa 38 içinde gösterilir. JOKER property özellik değerinin BLOCK' in genel arama karakteri kapsamında, genel arama karakterlerinin genel arama karakterlerine sahip aboneliklerini nasıl önlediğini fark edin.

Çizelge 6. QMA tarihinde alınan yayınlar			
Abonelik	Konu dizisi	Alınan yayınlar	Notlar
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	All publications to Football subtree blocked by WILDCARD (BLOCK) on Sports/Football
SARSENAL	Sports/#/Arsenal	-	Sports/Football üzerinde WILDCARD (BLOCK) , Arsenal üzerinde genel arama karakteri aboneliğini önler
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby için varsayılan WILDCARD , Leeds üzerinde genel arama karakteri aboneliğini engellememektedir.

Not:

Suppose a subscription has a wildcard that matches a topic object with the JOKER property value BLOCK. Aboneliğin aynı zamanda eşleşen genel arama karakterinin sağında bir konu dizgisi varsa, abonelik hiçbir zaman bir yayını almaz. Engellenen yayınların kümesi, engellenen genel arama karakterinin ebeveynleri olan konular için yayınlardır. Publications to topics that are children of the topic with the BLOCK property value are blocked by the wildcard. Bu nedenle, genel arama karakterinin sağına bir konu içeren abonelik konu dizgileri, hiçbir yayın için hiçbir yayın almayacak.

WILDCARD özellik değerinin BLOCK olarak ayarlanması, genel arama karakterleri içeren bir konu dizgisini kullanarak abone olamayacağınız anlamına gelmez. Böyle bir abonelik normal. The subscription has an explicit topic that matches the topic with a topic object having a JOKER property value BLOCK. It uses wildcards for topics that are parents or children of the topic with the JOKER property value BLOCK. Şekil 10 sayfa 38örneğinde, Sports/Football/# gibi bir abonelik yayınları alabilir.

Genel arama karakterleri ve küme konuları

Küme konu tanımları, bir kümedeki her kuyruk yöneticisine dağıtılır. Kuyruk yöneticisinde bir küme yöneticisinde bulunan bir küme konusuna abonelik, yetkili abonelikleri oluşturan kuyruk yöneticisinde yer alan kuyruk yöneticisinde yer alan bir sonuçla sonuçlanır. Bir yetkili sunucu aboneliği, kümedeki diğer her kuyruk yöneticisinde yaratılır. Genel arama karakterleri içeren, küme konularıyla birleştirilen konuları kullanan abonelikler, davranışı tahmin etmek için zor bir yol gösterebilirler. Bu davranış, aşağıdaki örnekteki açıklamadır.

In the cluster set up for the example, “Örnek: Sport yayınlama/abone olma kümesini yaratın” sayfa 41, QMB has the same set of subscriptions as QMA, yet QMB received no publications after the publisher published to QMA, see Şekil 11 sayfa 38. Sports/Football ve Sports/Rugby konuları küme başlıklarına rağmen, fullsubs.tst içinde tanımlanan abonelikler bir küme konusuna gönderme yapmamış olur. Yetkili sunucu aboneliği, QMB 'tan QMA' a yayılmaz. Yetkili abonelikler olmadan, QMA hiçbir yayın QMB ' a iletilmedi.

Some of the subscriptions, such as Sports/#/Leeds , might seem to reference a cluster topic, in this case Sports/Rugby . The Sports/#/Leeds subscription actually resolves to the topic object SYSTEM.BASE.TOPIC.

Sports/#/Leeds gibi bir aboneliğin gönderme yaptığı konu nesnesinin çözülmesine ilişkin kural şu şekildedir. Konu dizisini ilk genel arama karakterine kes. İlişkili bir denetim konusu nesnesi olan ilk konuyu aramak için konu dizgisinden sola doğru tarayın. Konu nesnesi bir küme adı belirtebilir ya da bir yerel konu nesnesi tanımlayabilir. In the example, Sports/#/Leeds, the topic string after truncation is Sports, which has no topic object, and so Sports/#/Leeds inherits from SYSTEM.BASE.TOPIC, which is a local topic object.

Kümelenmiş konulara abone olunmanın, genel arama karakterinin yayılmasını nasıl değiştirebileceğini görmek için, upsubs.battoplu iş komut dosyasını çalıştırın. Komut dosyası abonelik kuyruklarını temizler ve fullsubs.tst içinde küme başlığı aboneliklerini ekler. Bir dizi yayın yaratmak için puba.bat komutunu yeniden çalıştırın; bkz. Şekil 11 sayfa 38.

Çizelge 4 sayfa 40 , yayınların yayınlandığı kuyruk yöneticisine iki yeni abonelik eklenmesinin sonucunu gösterir. Sonuç beklendiği gibi, yeni abonelikler her biri bir yayın alır ve diğer abonelikler tarafından alınan yayınların sayıları değişmez. Diğer küme kuyruk yöneticisinde beklenmeyen sonuçlar ortaya çıkar; bkz. Çizelge 5 sayfa 40.

Çizelge 7. QMA tarihinde alınan yayınlar			
Abonelik	Konu dizisi	Alınan yayınlar	Notlar
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	All publications to Football subtree blocked by WILDCARD (BLOCK) on Sports/Football

Çizelge 7. QMA tarihinde alınan yayınlar (devamı var)			
Abonelik	Konu dizisi	Alınan yayınlar	Notlar
SARSENAL	Sports/#/Arsenal	-	Sports/Football üzerinde WILDCARD (BLOCK) , Arsenal üzerinde genel arama karakteri aboneliğini önler
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby için varsayılan WILDCARD , Leeds üzerinde genel arama karakteri aboneliğini engellemektedir.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal , aboneliğin bir genel arama karakterinin olmadığı için bir yayın alır.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds , herhangi bir etkinlikte bir yayın alır.

Çizelge 5 sayfa 40 shows the results of adding the two new subscriptions on QMB and publishing on QMA. QMB ' in bu iki yeni aboneliği olmadan hiçbir yayın almadığını geri çağırın. As expected, the two new subscriptions receive publications, because Sports/FootBall and Sports/Rugby are both cluster topics. QMB forwarded proxy subscriptions for Sports/Football/Arsenal and Sports/Rugby/Leeds to QMA, which then sent the publications to QMB.

Beklenmeyen sonuç, daha önce yayın içermeyen Sports/# ve Sports/#/Leeds adlı aboneliklerin artık yayınları aldığından kaynaklanır. Bunun nedeni, diğer abonelikler için QMB 'e iletilen Sports/Football/Arsenal ve Sports/Rugby/Leeds yayınlarının artık QMB ' e bağlı herhangi bir abone için kullanılabilmesinin nedeni. Consequently the subscriptions to the local topics Sports/# and Sports/#/Leeds receive the Sports/Rugby/Leeds publication. Spor/Futbol, JOKER özellik değeri BLOCK olarak ayarlanmış olduğundan, Sports/#/Arsenal bir yayın almamaya devam eder.

Çizelge 8. QMB tarihinde alınan yayınlar			
Abonelik	Konu dizisi	Alınan yayınlar	Notlar
SPORTS	Sports/#	Sports/Rugby/Leeds	All publications to Football subtree blocked by WILDCARD (BLOCK) on Sports/Football
SARSENAL	Sports/#/Arsenal	-	Sports/Football üzerinde WILDCARD (BLOCK) , Arsenal üzerinde genel arama karakteri aboneliğini önler
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby için varsayılan WILDCARD , Leeds üzerinde genel arama karakteri aboneliğini engellemektedir.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal , aboneliğin bir genel arama karakterinin olmadığı için bir yayın alır.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds , herhangi bir etkinlikte bir yayın alır.

Çoğu uygulamada, bir aboneliğin başka bir aboneliğin davranışını etkilemesi istenmeyen bir davranışa neden olur. One important use of the JOKER property with the value BLOCK is to make the subscriptions

to the same topic string containing wildcards behave uniformly. Aboneliğin yayıncı ile aynı kuyruk yöneticisine mi, yoksa farklı bir aboneliğe mi ait olduğunu, aboneliğin sonuçlarının aynı olduğunu mu?

Joker karakterler ve akışlar

WebSphere MQ Sürüm 6 akışları, konulara WebSphere MQ Sürüm 7 ile eşlenir; bkz. “Akışlar ve konular” sayfa 44 . In the default mapping, which is performed by **strmqbrk** in Version 7, all the topics in the stream Sports would be mapped to the topic Sports. Business akışındaki tüm konular Businesskonuyla eşlenecek.

Sports akışındaki WebSphere MQ Sürüm 6 ile * arasındaki bir abonelik, Sports ağacındaki tüm yayınları alır ve Business ağacındaki hiçbir yayın almaz. Sürüm 7 'de aynı abonelik, Sports ağacındaki tüm yayınları ve Business ağacındaki tüm yayınları alır. Akışlar sürüm 7 'ye geçirildiğinde bu davranışı engellemek için **strmqbrk** , JOKER özelliğini ayarlar. Bir akıştan geçirilen en üst düzey konuların her biri için bu değeri BLOCK değerine ayarlar. Sports ve Business için JOKER özelliği, Sports ve Businessadlı sürüm 6 akışlarından dönüştürme tarafından BLOCK değerine ayarlıdır.

Yayınlama/abone olma API ' ya yazılan yeni bir uygulama için geçerli olan etki, * aboneliğinin yayınsız olarak alınmamasını sağlar. To receive all the Sports publications you must subscribe to Sports/*, or Sports/# , and similarly for Business publications.

Yayınlama/abone olma aracısı WebSphere MQ Sürüm 7 'ye geçirildiğinde, kuyruğa alınan var olan bir yayınlama/abone olma uygulamasının davranışı değişmez. The **StreamName** property in the **Publish**, **Register Publisher**, or **Subscriber** commands is mapped to the name of the topic the stream has been migrated to.

Genel arama karakterleri ve abonelik noktaları

WebSphere Message Broker abonelikleri noktaları, WebSphere MQ Sürüm 7 ile konulara eşlenir; bkz. “Abonelik noktaları ve konular” sayfa 46. Sürüm 7 'de **migmqbrk** tarafından gerçekleştirilen varsayılan eşlemede, Sports abonelik noktasındaki tüm konular Sportskonusuna eşlenecektir. Business abonelik noktasındaki tüm konular, Businesskonuyla eşlenecektir.

Sports abonelik noktasındaki WebSphere Message Broker Sürüm 6 ile * arasındaki bir abonelik, Sports ağacındaki tüm yayınları alır ve Business ağacındaki hiçbir yayın yayınlanmaz. Sürüm 7 'de aynı abonelik, Sports ağacındaki tüm yayınları ve Business ağacındaki tüm yayınları alır. Bu davranışı engellemek için, abonelik noktaları sürüm 7 'ye geçirildiğinde, **migmqbrk** JOKER özelliğini ayarlar. Bir abonelik noktasından geçirilen en üst düzey konuların her biri için bu değeri BLOCK değerine ayarlar. The JOKER property of Sports and Business is set to the value BLOCK by the conversion from the WebSphere Message Broker subscription points called Sports and Business .

Yayınlama/abone olma API ' ya yazılan yeni bir uygulama için, geçişin etkisi, * aboneliğinin yayınsız olarak alınmamasını sağlar. To receive all the Sports publications you must subscribe to Sports/*, or Sports/# , and similarly for Business publications.

Yayınlama/abone olma aracısı WebSphere MQ Sürüm 7 'ye geçirildiğinde, kuyruğa alınan var olan bir yayınlama/abone olma uygulamasının davranışı değişmez. **Publish**, **Register Publisher** ya da **Subscriber** komutlarındaki **SubPoint** özelliği, aboneliğin yeni düzeye geçirilmiş olduğu konunun adıyla eşlenir.

Örnek: Sport yayınlama/abone olma kümesini yaratın

Aşağıdaki adımlar, dört kuyruk yöneticisi içeren bir küme (CL1) oluşturur: iki tam havuz, CL1A ve CL1B , ve iki kısmi havuz, QMA ve QMB. Tüm havuzlar yalnızca küme tanımlamalarını tutmak için kullanılır. QMA , küme konusu anasistemine atanır. Sürekli abonelikler hem QMA hem de QMBüzerinde tanımlanır.

Not: Örnek, Windowsiçin kodlanmıştır. Örneğin, diğer platformlardaki örneği yapılandırmak ve sınamak için [Create qmgrs.bat](#) ve [pub.bat](#)oluşturma kodunu yeniden kodlamanız gerekir.

1. Komut dosyalarını oluşturun.

a. [topics.tstyarat](#)

- b. [Create wildsubs.tst](#)
- c. [Create fullsubs.tst](#)
- d. [Create qmgrs.bat](#)
- e. [oluştur pub.bat](#)

2. Yapılandırmayı oluşturmak için [Create qmgrs.bat](#) komutunu çalıştırın.

```
qmgrs
```

Şekil 10 sayfa 38 içinde konuları oluşturun. The script in figure 5 creates the cluster topics Sports/ Football and Sports/Rugby.

Not: REPLACE seçeneği, bir konunun TOPICSTR özelliklerini değiştirmez. TOPICSTR , örnekte farklı konu ağaçlarını test etmek için kullanılan bir özeldir. Konuları değiştirmek için önce konuyu silin.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

Şekil 25. Konuları silin ve yaratın: topics.tst

Not: REPLACE konu dizgilerinin yerini değiştirmediği için konuları silin.

Joker karakterler içeren abonelikler oluşturun. Genel arama karakterleri, Şekil 10 sayfa 38 içindeki konu nesnelileri ilgili konulara karşılık gelir. Her abonelik için bir kuyruk yaratın. Kuyruklar temizlenir ve komut dosyası çalıştırıldığında ya da yeniden çalıştırıldığında abonelikler silinir.

Not: REPLACE seçeneği, bir aboneliğin TOPICOBJ ya da TOPICSTR özelliklerini değiştirmez. TOPICOBJ or TOPICSTR are the properties that are usefully varied in the example to test different subscriptions. Bunları değiştirmek için önce aboneliği silin.

```
DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)
```

Şekil 26. Genel arama karakteri abonelikleri yarat: wildsubs.tst

Küme konusu nesnelere başvuruda bulunan abonelikler oluşturun.

Not:

The delimiter, /, is automatically inserted between the topic string referenced by TOPICOBJ, and the topic string defined by TOPICSTR.

DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) , aynı aboneliği oluşturur. TOPICOBJ , önceden tanımladığınız konu dizgisine hızlı bir şekilde gönderme yapmak için kullanılır. Abonelik, yaratıldığında, artık konu nesnesini ifade etmemez.

```
DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)
```

Şekil 27. Abonelikleri sil ve yarat: fullsubs.tst

İki havuzu olan bir küme oluşturun. Yayınlama ve abone olma için iki kısmi havuz oluşturun. Her şeyi silmek ve yeniden başlamak için komut dosyasını yeniden çalıştırın. Komut dosyası aynı zamanda konu sıradüzenini ve ilk genel arama karakteri aboneliklerini de yaratır.

Not:

Diğer platformlarda, benzer bir komut dosyası yazın ya da tüm komutları yazın. Bir komut dosyasının kullanılması, her şeyi silmesini ve aynı bir yapılandırmayla yeniden başlatılmasını hızlı bir şekilde sağlar.

```
@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof
```

Şekil 28. Kuyruk yöneticileri yarat: qmgrs.bat

Küme konularına abonelikleri ekleyerek yapılandırmayı güncelleyin.

```
@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst
```

Şekil 29. Abonelikleri güncelleştir: upsubs.bat

Yayın konusu dizgisini içeren iletileri yayınlamak için, parametre olarak kuyruk yöneticisiyle pub.bat komutunu çalıştırın. Pub.bat uses the sample program **amqspub**.

```
@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1
```

Şekil 30. Yayınla: pub.bat

Yayınlama/abone olma topolojisi

Yayınlama/abone olma topolojisi , yayınlama/abone olma uygulamalarını destekleyen kuyruk yöneticilerinden ve bu öğelerden oluşan bağlantılardan oluşur.

Bir yayınlama/abone olma uygulaması, birbirine bağlı bir kuyruk yöneticilerinden oluşan bir ağdan oluşabilir. Kuyruk yöneticilerinin tümü aynı fiziksel sistemde olabilir ya da birkaç fiziksel sistem üzerinden dağıtılabılır. Kuyruk yöneticilerini bir araya bağlayarak, yayınların ağ içindeki herhangi bir kuyruk yöneticisini kullanan bir uygulama tarafından alınabileceği ifade edilebilir.

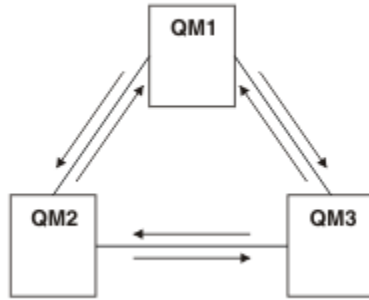
Bu, aşağıdaki yararları sağlar:

- İstemci uygulamaları, uzak bir kuyruk yöneticisiyle değil, yakındaki bir kuyruk yöneticisiyle iletişim kurabilir, böylece daha iyi yanıt süreleri elde edilir.
- Birden çok kuyruk yöneticisi kullanarak daha fazla abone desteklenebilir.

Yayınlama/abone olma ileti alışverişi yapan kuyruk yöneticilerini farklı iki yolla, kümelere ve sıradüzenlere yerleştirebilirsiniz. Basit bir kümeye ve basit bir sıradüzene ilişkin örnekler için bkz. Şekil 31 sayfa 60 ve Şekil 32 sayfa 61. Bu iki topolojiye ilişkin ek bilgi edinmek ve sizin için en uygun olan bilgileri öğrenmek için ürün belgelerinin bu bölümündeki bilgilere bakın.

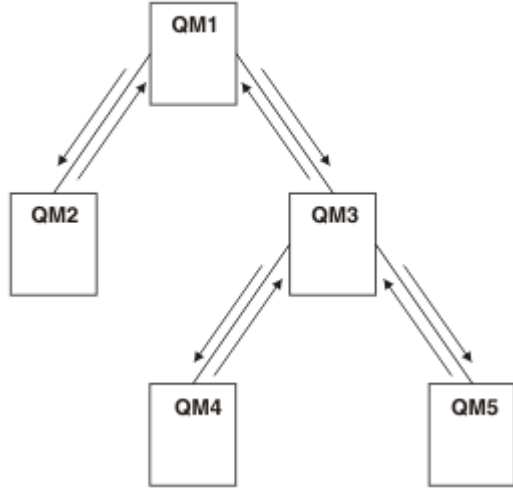
Bir sıradüzende kümeleri birleştirerek her iki topolojiyi de birlikte kullanmak mümkündür.

Cluster



Şekil 31. Basit yayınlama/abone olma kümesi

Hierarchy



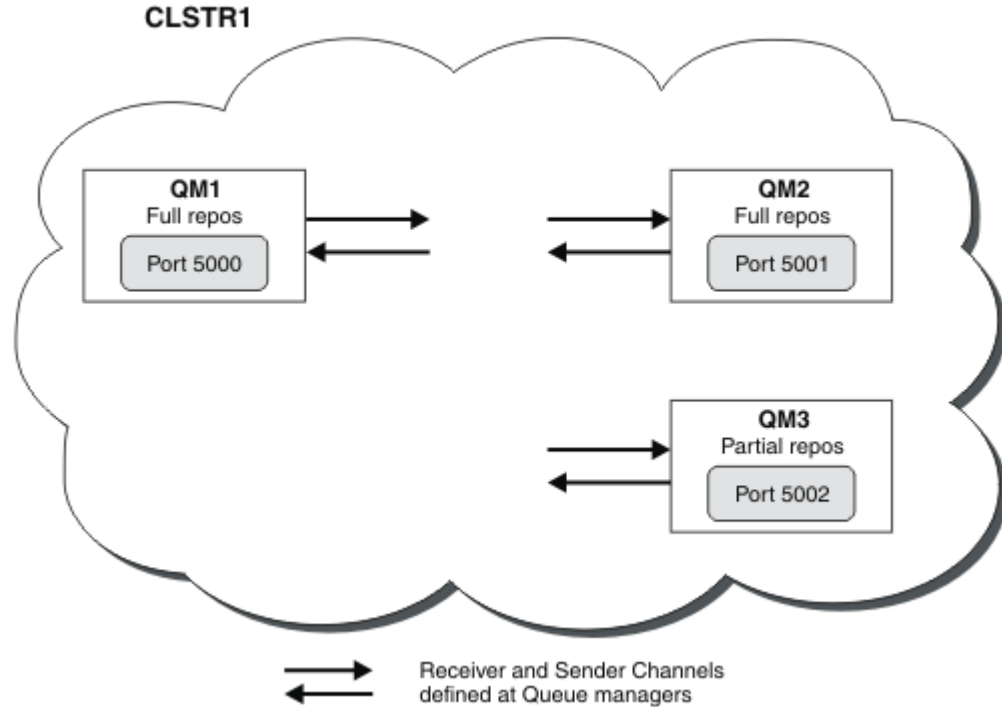
Şekil 32. Basit yayınlama/abone olma sıradüzeni

Yayınlama/abone olma kümesi ayarlanıyor: Senaryo 1

İki kuyruk yöneticisini bir kümeye tam havuz olarak ekleyin ve bu kanalları arasındaki kanalları tanımlayın.

Bu görev hakkında

Aşağıdaki şemada üç kuyruk yöneticisi vardır: QM1 , QM2ve QM3: Üç kuyruk



yöneticisinin

QM1 ve QM2 , bir kümedeki tam havuzlardır ve QM3 kısmi bir havuzdur.

Scenario 1 adds QM1 and QM2 to the cluster DEMO as full repositories.

2. senaryo adds QM3 to the cluster DEMO as a partial repository.

Bu görevler en az bir komut penceresi gerektirir.

Yordam

1. QM1 ve QM2 ' yi DEMO kümesinin tam havuzları olarak ayarlayın:

```
alter QMGR REPOS(DEMO)
```

2. QM1 için bir dinleyici tanımlayın ve başlatın:

```
define listener(QM1_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5000)
start listener(QM1_LS)
```

3. QM2 için bir dinleyici tanımlayın ve başlatın:

```
define listener(QM2_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5001)
start listener(QM2_LS)
```

4. QM1 için bir alıcı kanalı tanımlayın:

```
DEFINE CHANNEL(DEMO.QM1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')
CLUSTER(DEMO) DESCR('TCP Cluster-receiver channel for queue manager QM1')
```

5. Define a sender channel from QM1 to QM2:

```
DEFINE CHANNEL(DEMO.QM2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5001)')
CLUSTER(DEMO) DESCR('TCP Cluster-sender channel from QM1 to queue manager QM2')
```

6. QM2 için bir alıcı kanalı tanımlayın:

```
DEFINE CHANNEL(DEMO.QM2) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5001)')
CLUSTER(DEMO) DESCR('TCP Cluster-receiver channel for queue manager QM2')
```

7. Define a sender channel from QM2 to QM1:

```
DEFINE CHANNEL(DEMO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')
CLUSTER(DEMO) DESCR('TCP Cluster-sender channel from qm2 to qm1')
```

8. QM1 üzerindeki scores küme konusunu tanımlayın:

```
define topic(scores) TOPICSTR(/football) CLUSTER(DEMO)
```

9. Aşağıdaki komutlara sahip kuruluşu doğrulayın:

```
display topic(scores) type(all) clusinfo
display clusqmgr(*)
display chstatus(*)
```

10. İki komut penceresi kullanarak kurulumu test edin:

- a. Bu komutu ilk komut penceresinde girin:

```
/opt/mqm/samp/bin/amqspub /FOOTBALL/scores QM1
```

- b. İkinci komut penceresinde bu komutu girin:

```
/opt/mqm/samp/bin/amqssub /FOOTBALL/scores QM2
```

İlgili görevler

[WebSphere MQ kümelerini yönetme](#)

[Yeni bir küme ayarlanıyor](#)

Yayınlama/abone olma kümesi ayarlanıyor: Senaryo 2

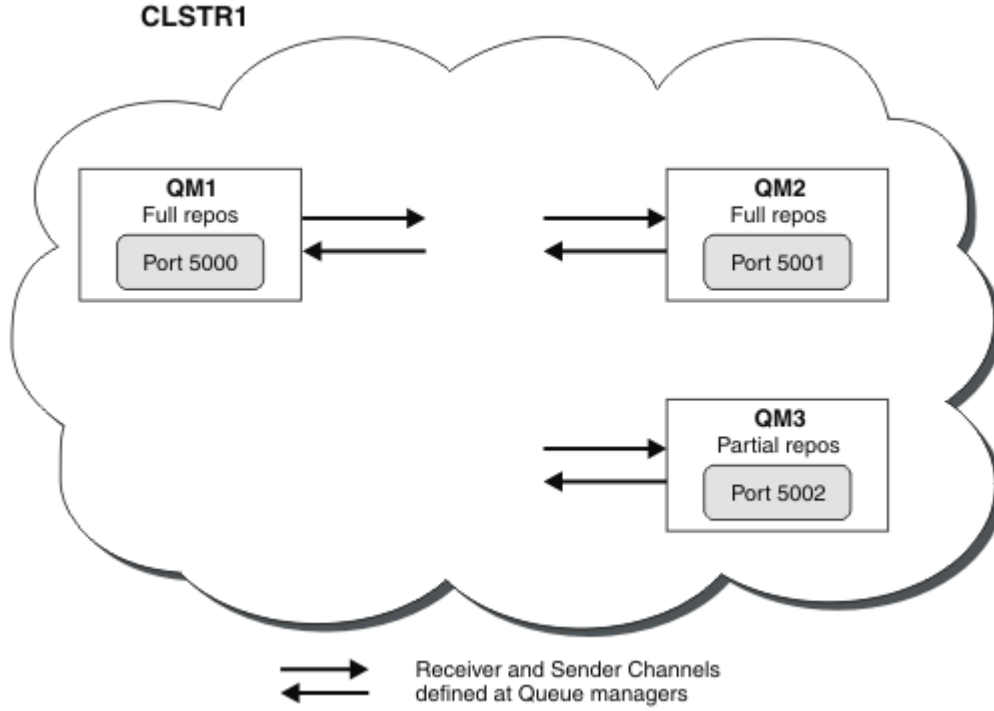
Kümeye kısmi bir havuz olarak üçüncü bir kuyruk yöneticisi ekleyin.

Başlamadan önce

You must have completed the task in [“Yayınlama/abone olma kümesi ayarlanıyor: Senaryo 1”](#) sayfa 61 before completing this task.

Bu görev hakkında

Aşağıdaki şemada 3 kuyruk yöneticisi vardır; QM1, QM2ve



QM3:

QM1 ve QM2 , bir kümedeki tam havuzlardır ve QM3 kısmi bir havuzdur.

1. senaryo adds QM1 and QM2 to the cluster DEMO as full repositories.

Scenario 2 adds QM3 to the cluster DEMO as a partial repository.

Bu görevler en az 1 komut penceresi gerektirir.

Yordam

1. QM3 için bir dinleyici tanımlayın ve başlatın:

```
define listener(QM3_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5002)
start listener(QM3_LS)
```

2. QM3 için bir alıcı kanalı tanımlayın:

```
DEFINE CHANNEL(DEMO.QM3) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5002)') CLUSTER
(DEMO) DESCR('TCP Cluster-receiver channel for queue manager QM3')
```

3. Define a sender channel from QM3 to QM1:

```
DEFINE CHANNEL(DEMO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')
CLUSTER(DEMO) DESCR('TCP Cluster-sender channel from qm3 to qm1')
```

4. Aşağıdaki komutlara sahip kuruluşu doğrulayın:

```
display topic(scores) type(all) clusinfo
display clusqmgr(*)
display chstatus(*)
```

5. 2 komut penceresini kullanarak kurulumu test edin:

a. Bu komutu ilk komut penceresinde girin:

```
/opt/mqm/samp/bin/amqspub /FOOTBALL/scores QM2
```

b. İkinci komut penceresinde bu komutu girin:

Kümeleri yayınla/abone ol

Bir yayınlama/abone olma kümesi, yayınların otomatik olarak yayınlama uygulamalarından kümedeki herhangi bir kuyruk yöneticisinde bulunan aboneliklere otomatik olarak taşındığı, birbirine bağlı kuyruk yöneticilerinin standart bir IBM WebSphere MQ kümesidir.

Yayınlama/abone olma ileti sistemi için kullanılan bir küme, standart bir IBM WebSphere MQ kümesinden farklı değildir. Bu nedenle, yayınlama/abone olma kümesi içindeki kuyruk yöneticileri fiziksel olarak ayrı bilgisayarlarda var olabilir ve her kuyruk yöneticisi çifti gerektiğinde otomatik olarak küme kanallarıyla birbirine bağlanır. Bir IBM WebSphere MQ kümesinin planlamaya ve yapılandırılmasına ilişkin bilgi için bkz. Kümelerin çalışma şekli.

Kümelenmiş bir konu nesnesi tanımlandığında bir yayınlama/abone olma kümesi yaratılır; bu nesne, kümedeki herhangi bir kuyruk yöneticisi tarafından yapılandırılan konu üzerindeki **CLUSTER** özneliğini ayarlanarak yaratılır. Konu tanımlaması, kümenin tüm üyelerine dağıtılır. Konu ağacında, kümedeki herhangi bir kuyruk yöneticisinde bu konuyu ve konu ağacındaki herhangi bir konu dizisini yayınlatabilir ve bu konuya abone olabilirsiniz. Yayınlar, kümedeki diğer kuyruk yöneticilerine bağlı abonelere otomatik olarak yayılır.

Kümeli olmayan yayınlama/abone olma etkinliği, kümelenmiş bir konu nesnesinin altına düşmeyen konu dizileriyle çalışarak yayınlama/abone olma kümesinde de yer alabilir. Bu düzenleme, tüm aboneliklerin tüm sıradüzenine dağıtıldığı bir yayınlama/abone olma sıradüzeninden farklıdır. Her iki durumda da, Abonelik ve Yayın kapsamı kullanılarak soner denetimi kullanılabilir.

Yayınlama/abone olma topolojisindeki kümelerin kullanılması aşağıdaki yararları sağlar:

- Aynı kümedeki belirli bir kuyruk yöneticisinde bulunan bir aboneliğin gönderileceği iletiler doğrudan o kuyruk yöneticisine taşınır ve ara kuyruk yöneticisinde geçişe gerek yoktur. Bu, performansı iyileştirir ve sıradüzenli bir topolojiyle karşılaştırıldığında, kuyruk yöneticisi yayınlama/abone olma trafiğini eniyiler.
- Tüm kuyruk yöneticileri doğrudan birbirine bağlıken, bu topolojide tek bir hata noktası yoktur. Bir kuyruk yöneticisi yoksa, kümedeki diğer kuyruk yöneticilerindeki abonelikler hala var olan kuyruk yöneticilerindeki yayıncılardan ileti alabilirler.
- Birden çok ayrı küme içeren bir sistemde, örneğin kümelerin coğrafi olarak dağıldığı durumlarda, kümeleri küme sıradüzenine bağlamak mümkündür. Bu bağlantı, yayınların ve aboneliklerin ağ üzerinden akışını etkinleştirmek için her kümede tek bir kuyruk yöneticisine katılarak yaratılır; bkz. “Birden çok kümenin konu alanlarını birleştir” sayfa 95. Ayrıca, bir kümeden diğerine hangi yayın akışını denetleyebileceğinin de denetleyebileceğiniz; bkz. “Birden çok kümede konu alanlarını birleştirin ve yalıtın” sayfa 96.
- Abone olunan bir uygulama, en yakın kuyruk yöneticisine, kendi başarımını artırmak için bağlanabilir. Kuyruk yöneticisi, küme içindeki tüm kuyruk yöneticilerinden istemcinin abonelik kaydı ile eşleşen tüm iletileri alır.

Bir istemci uygulamasının başarımı, bu kuyruk yöneticisinden istenen diğer hizmetler için de iyileştirilir. Bir istemci uygulaması hem yayınlama/abone olma, hem de noktadan noktaya ileti sistemini kullanabilir.

- İş yükünü paylaşmak üzere kümeye daha fazla kuyruk yöneticisi eklenerek, her kuyruk yöneticisine ilişkin istemci sayısı ve abonelikler azaltılabilir. Yayınlar otomatik olarak yeni kuyruk yöneticilerindeki istemcilere dağıtır. Bazı kullanım örüntüleri için bu işlem bir yayınlama/abone olma küme topolojisini yüksek düzeyde ölçeklenebilir yapabilir.

Yayınlama/abone olma içindeki kümeler kullanılırken dikkate alınacak şeyler:

- Bir yayınlama/abone olma kümesindeki tüm kuyruk yöneticileri, kümedeki diğer tüm kuyruk yöneticilerinden otomatik olarak haberdar edilir. Bu işlem, yalnızca kuyruk yöneticisine ilgi çeken kuyruk yöneticilerinin bilindiği noktadan noktaya bir küme için farklıdır.
- Kümelenmiş bir konuya bir ya da daha fazla aboneliği barındıran yayınlama/abone olma kümesindeki kuyruk yöneticileri, kümedeki tüm diğer kuyruk yöneticilerine otomatik olarak küme gönderen kanalları yaratır. Kuyruk yöneticileri, alma kuyruğu yöneticileri kümelenmiş herhangi bir konu üzerinde ileti yayınlamasa da, her birine abonelikleriyle ilgili bilgiler de gönderir.

- Kümelenmiş bir konu altındaki bir konu dizgisine kuyruk yöneticisinin ilk aboneliği, kümedeki diğer tüm kuyruk yöneticisine gönderilen bir iletiyle sonuçlanır. Benzer şekilde, bir konu dizgisinin de silineceği son abonelik, bir iletiyle sonuçlanabilir. Kümelenmiş bir konu altında kullanılmakta olan tek tek konu dizgileri, kuyruk yöneticisi iletişiminin daha fazla ortaya çıkması.



DİKKAT:

Bu konuda daha önce listelenen nedenlerden dolayı, kümelenmiş bir konunun büyük bir IBM WebSphere MQ kümesine (yani, birçok kuyruk yöneticisi içeren bir küme) getirilmesi, kümedeki her kuyruk yöneticisinde ek yük ile sonuçlanabilir ve bazı durumlarda performansın azaltılmasına neden olabilir. Daha fazla bilgi için "[Küme konusu performansı](#)" sayfa 71 başlıklı konuya bakın.

Özellikle mevcut bir küme olan bir kuyruk yöneticilerine yayınlama/abone olma, bu düşüşleri performansında barındırmak için dikkatli bir şekilde planlanmalıdır.

Where it is known that a cluster cannot accommodate the reduction in performance of publish/subscribe, it is possible to disable the clustered publish/subscribe functionality in queue managers using the **PSCLUS** parameter. **PSCLUS** parametresi öncelikle, kümelenmiş bir konuyu yanlışlıkla ya da yanlış tanımlayarak yayınlama/abone olma kümesi yaratılmasıyla oluşabilecek ciddi sorunları durdurmak için. Bu işlevin geçersiz kılınmasıyla ilgili daha fazla bilgi için bkz. "[Bir kümede kümelenmiş yayınlama/abone olma engelini engelleme](#)" sayfa 70.

Kümelemeyi yayınlama/abone olma: En iyi uygulamalar

Bu konuda, IBM WebSphere MQ Yayınlama/abone olma kümelerinin planlanması ve yönetilmesi için kılavuz bilgiler yer alır. Bilgiler, müşterilerden gelen test ve geri bildirimlere dayanır.

The following information assumes that the user has a basic understanding of IBM WebSphere MQ clusters, Publish/subscribe, and is familiar with the topics in "[Dağıtım yayınlama/abone olma](#)" sayfa 48. Bu bilgiler, "tek bir boyut tümü" çözümü olarak tasarlanmamaktadır, ancak bunun yerine ortak sorunlara ortak yaklaşımlar paylaşılmaya çalışılır.

Yayınlama/Abone Olma kümeleri

Bir küme ile, gerektiğinde kümedeki kuyruk yöneticileri arasında "her türlü" doğrudan bağlantılığa sahip olur. Bir küme noktadan noktaya ileti sistemi için kullanıldığında, kümedeki her bir kuyruk yöneticisi yalnızca, kümedeki ve kümelenmiş kuyruklardaki diğer kuyruk yöneticileri gibi diğer küme kaynaklarıyla ilgili bilgileri bilir. Bu bilgiler, uygulamaların bunları kullanma isteğini bağlarken; bunlar, bilmesi gereken bir temel üzerinde çalışırlar.

Yayınlama/abone olma kümesi, olağan [CLUSSDR](#) ve [CLUSTRVR](#) kanal tanımlamalarıyla birlikte bir kuyruk yöneticilerinden oluşan bir kümedir. Ancak bir yayınlama/abone olma kümesi, bir küme adını saptadığı kümede en az bir kuyruk yöneticisinde tanımlanan en az bir [KONU](#) nesnesi de içerir.

Kümede tanımlı bir konu nesnesiyle, kümedeki bir kuyruk yöneticisine bağlı bir uygulama o konuya ya da bu konunun altındaki konu ağacındaki herhangi bir düğüme abone olabilir ve kümedeki diğer kuyruk yöneticilerinden bu konuyla ilgili yayınları alır. Bu işlem, aboneliğin bulunduğu kuyruk yöneticisini tanımlayan kümedeki diğer tüm kuyruk yöneticilerine yetkili sunucu abonelikleri yaratılmasıyla gerçekleştirilir. Bu nedenle, söz konusu konuya ilişkin bir yayın kuyruk yöneticisinde gerçekleştiğinde, bunu kümenin diğer uygun üyelerine iletmeyi bilirler ve oradan da bunu tek tek uygulama aboneliklerine ulaştırmayı bilirler.

Bu teslimata ulaşmak için, kümedeki her kuyruk yöneticisinin bir kümeye bir konu eklendiği anda kümedeki diğer tüm kuyruk yöneticilerinin kimliğini bilmesi gerekir. Bu bilgi, kümenin tam havuz kuyruğu yöneticilerine göre yayılır. Bir kuyruk yöneticisinde yayınlanan iletiler yalnızca, kümedeki diğer kuyruk yöneticilerine, aynı konuya abone olan abonelikler tarafından bilinen diğer kuyruk yöneticilerine gönderilir. Bu işlemi gerçekleştirmek için, bir uygulama kümelenmiş bir konuya ilişkin abonelik yarattığında, o kuyruk yöneticisi yetkili abonelikleri yaymak için küme gönderici kanalları yoluyla, kümedeki diğer tüm kuyruk yöneticileriyle doğrudan iletişim kurmalıdır.

Bu işlem, noktadan noktaya teslimi için bir küme kullanırken gerekli olan, gerekli olan sınırlı bilgi ve bağlantıdan büyük ölçüde farklılık gösterir. Bu nedenle, yayınlama/abone olma kümelerindeki

gereksinimler, bir noktadan noktaya küme üzerindeki gereksinimlerin (kümelenmiş konuları olmayan bir küme) farklıdır.

Kümelenmiş konuların kullanılması, yayınlama/abone olma etki alanının kuyruk yöneticileri arasında genişlemesini kolaylaştırır, ancak mekanik ve çıkarımlar anlaşılmadığında ve yayınlama/abone olma için kullanılan kümeye göre dikkate alınırsa sorunlara yol açabilir. Aşağıdaki en iyi uygulamalar, bu anlayış ve hazırlık konusunda yardımcı olmak üzere tasarlanmıştır.

Özet olarak, kümelenmiş yayınlama/abone olma performansının etkileri büyük bir kümeye zarar verebilir ve var olan bir kümede yayınlama/abone olma işlemi kullanma girişiminde bulunmadan önce dikkatle düşünülmesi ve anlaşılması gerekir. Örneğin, kümelenmiş bir konu nesnesinin basit bir yaratımı bile. Yayınlama/abone olma etkinliği için özel olarak ayrılmış küçük bir yeni kümeyle başlamak ve kümeyi oradan büyütme daha iyi olabilir.

Yayınlama/abone olma topolojisi tasarlama

Daha önce açıklandığı gibi, bir kümede yayınlama/abone olma özelliğini kullanırken, kapasite ve başarımlar konuları da vardır. Bu nedenle, kuyruk yöneticilerine yayınlama/abone olma gereksinmesini dikkatli bir şekilde göz önünde bulundurmanız ve bunu yalnızca gerekli olan kuyruk yöneticisi sayısına sınırlandırmak en iyi uygulamadır. Bir konu kümesine yayınlama ve abone olma gereksinimi olan kuyruk yöneticisi sayısı alt sınırı belirlendikten sonra, bunlar yalnızca bunları içeren bir kümenin üyeleri ve başka kuyruk yöneticisi bulunmayabilir.

Bu, özellikle noktadan noktaya ileti sistemi için düzgün çalışan bir küme içinde yer alan bir kümedir. Bu nedenle, var olan büyük bir kümeyi bir yayınlama/abone olma kümesine çevirirken, uygulamanın yürürlükteki kümeyi kullanmak yerine, uygulamaların denenebileceği yayınlama/abone olma çalışmaları için ayrı bir küme yaratmanız daha iyi bir uygulamadır. Bir ya da daha fazla noktadan noktaya kümede var olan kuyruk yöneticilerini kullanmaya devam etmek mümkündür; bu kuyruk yöneticilerinin alt kümesinin yeni yayınlama/abone olma kümesinin üyeleri olması gerekir. Ancak bu yeni kümenin, var olan küme tam havuzlarından ek yükü yalıtım için tam havuz olarak yapılandırılan ayrı kuyruk yöneticilerine sahip olması gerekir.

Bir kümenin boyutu ya da güncel yükü nedeniyle yayınlama/abone olma için kullanılmaması gereken bir yerde, kümedeki herhangi bir kuyruk yöneticisinde kümelenmiş bir konunun basit yaratılmasıyla beklenmeyen bir şekilde bu kümenin bir yayınlama/abone olma kümesi içinde olmasını önlemek iyi bir uygulamadır. Bu tasarımı gerçekleştirmek için **PSCLUS** kuyruk yöneticisi özelliğini kullanın, ayrıntılar için bkz. [Bir kümede kümelenmiş yayınlama/abone olma engelini engelleme](#).

Ayrıca, kümeye hangi konuların ekleneceği konusunda dikkatli olmanız da önemlidir: Konu ağacında bu konuların daha yüksek olması, daha yaygın hale gelmeleri demektir. Bu nedenle, görülen davranışı göz önünde bulunmadan, konu kök düğümünün kümeye konması önerilmez. Make global topics obvious where possible, for example by using a high-level qualifier in the topic string: /global or /cluster.

Sistemler nasıl boyutlanacak?

Yayınlama/abone olma kümeleri, modelin noktadan noktaya ileti alışverişi için farklı olması nedeniyle birçok kanal gerektirir: Her kuyruk yöneticisinin o kümedeki diğer tüm kuyruk yöneticileriyle konuşması gereken bir gereksinim vardır. Noktadan noktaya iletişim modeli 'opt in' biridir, ancak yayınlama/abone olma kümelerinde abonelik fanı çıkılmaz bir ayırt etme niteliği vardır. Bu nedenle, tam havuz kuyruğu yöneticileri ve bir yayınlama/abone olma kümesinde yerel abonelikleri barındıran herhangi bir kuyruk yöneticisinin, aynı anda kümenin her üyesinde kanal oluşturma kapasitesine sahip olması gerekir.

Yayınlama/abone olma kümesindeki her kuyruk yöneticisinin bu kapasiteyi gerçekleştirmelerini sağlamak en iyilerdir, ancak hiçbir zaman anasistem aboneliklerine ev sahipliği yapmadığı bilinen kuyruk yöneticilerinin, diğer kuyruk yöneticileriyle kanal oluşturmak zorunda kalmaması ve bu nedenle bu kapasite düzeyini gerektirmemesi kabul edilir.

Ancak, bu tür bir kuyruk yöneticisinde oluşturulmuş yanlışlıkla bir aboneliğin olması ya da bu kuyruk yöneticisini kümedeki diğer kişilerle el ile yeniden eşzamanlamak için herhangi bir girişimde bulunulması ya da tüm kanalların eşzamanlı olarak başlatılmakta olan sonuçlarla sonuçlanması için dikkatli olmanız

gerekir. Ek bilgi için “Yetkili sunucu aboneliklerinin yeniden eşzamanlanması” sayfa 67 başlıklı konuya bakın.

Kümelenmiş yayınlama/abone olma, diğer kuyruk yöneticilerindeki aboneliklere teslim edilmesi için bir kuyruk yöneticisine yayınlanan iletilerin sağlanmasına olanak sağlar. Ancak, noktadan noktaya ileti alışverişlerine ilişkin olarak, kuyruk yöneticileri arasındaki iletilerin iletilmesinin maliyeti performansa zarar verebilir. Bu nedenle, iletilerin yayınlanmakta olduğu kuyruk yöneticilerine aynı kuyruk yöneticilerindeki konulara abonelikler oluşturmak için mümkün olan her yerde girişimde bulunmalı.

Başka bir dikkat edilmesi de, yetkili sunucu aboneliklerinin yayılması sistemindeki performansa ilişkin etkilidir. Tipik olarak, kuyruk yöneticisi belirli bir kümelenmiş konu dizisine ilişkin ilk abonelik (yalnızca yapılandırılmış bir konu nesnesi değil) yaratıldığında, kümedeki diğer her kuyruk yöneticisine bir yetkili abonelik ileti gönderir. Bir yayınlama/abone olma çözümü, abone olunmakta olan birçok benzersiz konu dizisinden oluşuyorsa ya da konular sık sık abone olunacaksa ve abone olunmayan konular arasında, bir kümedeki tüm kuyruk yöneticileri arasında önemli miktarda yetkili abonelik trafiği oluşturulabilir ve sistemin genel performansını olumsuz yönde etkileyebilirsiniz. Yetkili abonelik ek yükünü azaltmanın yolları hakkında bilgi için bkz. “Küme konusu performansı” sayfa 71 .

Yetkili sunucu aboneliklerinin yeniden eşzamanlanması

Olağan koşullarda, kuyruk yöneticileri, sistemdeki yetkili sunucu aboneliklerinin, kümedeki her kuyruk yöneticisinde bulunan abonelikleri doğru şekilde yansıttığından otomatik olarak emin olur.

Ancak, gereksinim ortaya çıkarsa, bir kuyruk yöneticisinin yerel aboneliklerini, **QMGR TIPINI YENILE (PROXYSUB)** komutunu kullanarak kümede yaydığı yetkili sunucu abonelikleriyle el ile yeniden eşitleyebilirsiniz.

Not: Yeniden eşzamanlama geçici olarak, komutun yayınlandığı kuyruk yöneticisinden kaynaklanan, kümede ani bir ek yetkili sunucu aboneliği yükü yaratır. Bu nedenle, IBM WebSphere MQ hizmeti, IBM WebSphere MQ belgeleri ya da hata günlüğü kaydı size yöneltmedikçe, bu nedenle kullanmayın.

Bir kuyruk yöneticisi bir kanal durdurulduğundan ve tüm iletiler iletim için kuyruğa alınmadığından ya da işletmen hatası iletilerin SYSTEM.CLUSTER.TRANSMIT.QUEUE kuyruğundan yanlış bir şekilde silinmesine neden olduğundan, bir kuyruk yöneticisi yetkili aboneliklerini doğru şekilde yatamayacağından, yeniden eşzamanlamanın gerekli olduğu bir örnek. Bu durumda, önce özgün sorunu yeniden düzeltin (örneğin, kanalı yeniden başlatarak), kuyruk yöneticisinde **REFRESH QMGR TYPE (PROXYSUB)** komutunu verin. Etkilenen abonelikler için, yetkili sunucu abonelikleri yerine geçmeyen yayınların kurtarılmadığına dikkat edin. Bu sakıncalı hesaba dahil edilmelidir.

Yeniden eşzamanlama, kuyruk yöneticisinin, kanalları kümedeki diğer tüm kuyruk yöneticilerine başlatmasını gerektirir. Bu nedenle, yenilemediğiniz kuyruk yöneticisinin, kümedeki diğer tüm kuyruk yöneticileriyle iletişim kurmasıyla başa çıkmak için yeterli yetenesahip olması gerekir.

Küme konuları

Cluster topics are administrative topics with the **cluster** attribute defined. Küme konularıyla ilgili bilgiler bir kümenin tüm üyelerine yönlendirilir ve her kuyruk yöneticisinde farklı bir konu alanı yaratmak için yerel konular ile birleştirilir.

Bir kuyruk yöneticisi üzerinde bir küme konusu tanımladığınızda, küme konu tanımlaması tüm havuz kuyruğu yöneticilerine gönderilir. Tüm havuzlar, küme konusu tanımlamasını kümedeki tüm kuyruk yöneticilerine yayın ve küme içindeki herhangi bir kuyruk yöneticisinde yayıncılar ve aboneler için aynı küme konusunu kullanıma sağlar. Küme konusu yarattığınız kuyruk yöneticisi, küme konusu anasistemi olarak bilinir. Küme konusu, kümedeki herhangi bir kuyruk yöneticisi tarafından kullanılabilir, ancak küme konusunda herhangi bir değişiklik, o konunun tanımlandığı (anasistem) kuyruk yöneticisinde yapılmalıdır. Bu noktada, değişiklik, tüm havuzlarla kümenin tüm üyelerine dağıtılır.

Her kuyruk yöneticisinde, tek bir konu adı alanı, bunun farkında olduğu yerel ve küme konu tanımlarından oluşturulur. Bir uygulama, kümelenmiş bir konuya çözülen bir konuya abone olduğunda, IBM WebSphere MQ bir yetkili abonelik yaratır ve bu aboneliği doğrudan, aboneliğin oluşturulduğu kuyruk yöneticisinden kümenin diğer tüm üyelerine gönderir. Kümelenmiş konunun kendisinden farklı olarak, yetkili sunucu abonelikleri, tam havuz kuyruğu yöneticileri aracılığıyla gerçekleşmez.

Bir konuda yayınlanan iletiler, yayıncısının bağlı olduğu kuyruk yöneticisiyle bilinen tüm aboneliklere gönderilir. Bu aboneliklerden herhangi biri yetkili sunucu abonelikse, yayınlanan iletinin bir kopyası, yetkili aboneliği oluşturan kuyruk yöneticisine gönderilir. Alıcı kuyruk yöneticisi, iletinin bir kopyasını her yerel aboneliğe gönderir. Bu işlem, kümelenmiş bir konuya abonenin, kümedeki kuyruk yöneticilerine bağlı yayıncılardan yayınların yayınlanmasını sağlar ve en az sayıda yayınlanmış ileti, küme üzerinden yayılır.

Kümelenmiş bir konusunuz ve yerel bir konu nesnesiniz varsa, yerel konu öncelik kazanır. Ek bilgi için [“Birden çok küme konu tanımlaması” sayfa 69](#) başlıklı konuya bakın.

Küme konularını görüntülemek için kullanılacak komutlarla ilgili daha fazla bilgi için aşağıdaki ilgili bağlantılara bakın:

Joker abonelikler

Yetkili sunucu abonelikleri, yerel abonelikler, kümelenmiş bir konu nesnesinin üzerinde ya da altında çözülen bir konu dizgisine yapıldığında oluşturulur. Konu sıradüzeninde herhangi bir küme konusundan daha yüksek bir genel arama aboneliği varsa, bu abonelik, eşleşen küme konusu için kümenin etrafında yetkili sunucu abonelikleri gönderilmez ve bu nedenle kümenin diğer üyelerinden herhangi bir yayın almaz. Ancak, yerel kuyruk yöneticisinden yayın alma işlemi gerçekleştirir.

Ancak, başka bir uygulama küme konusuna ya da küme konusuna çözülen bir konu dizgisine abone olursa, yetkili sunucu abonelikleri oluşturulur ve yayınlar bu kuyruk yöneticisine dağıtılır. Özgün, daha yüksek genel arama karakteri aboneliği, bu yayınların geçerli bir alıcısı olarak kabul edilir ve bir kopya alır.

Bu davranış, aynı konu başlıklarına yerel olarak yayınlanan iletilerden farklıdır. Bu davranış gerekli değilse, kümelenmiş konu üzerinde **WILDCARD (BLOCK)** ayarı, özgün genel arama karakterinin geçerli bir abonelik olarak kabul edilmemesi ve küme konusunda ya da alt konularında herhangi bir yayın (yerel ya da kümenin başka bir yerindeki) almaması gerekir.

İlgili kavramlar

[Yönetimle ilgili konularla çalışma](#)

[Aboneliklerle çalışma](#)

İlgili başvurular

[KONUYU GÖRÜNTÜLE](#)

[TANITIM](#)

[GÖRÜNTÜLE](#)

Küme konusu öznitelikleri

Yayınlama/abone olma kümelerini tasarlamak ve yönetmek için küme konusu özniteliklerinin iyi bir şekilde anlaşılması gerekir.

Bir konu nesnesinin, çok kuyruk yöneticisi yayınlama/abone olma topolojileri için geçerli olan bir dizi özniteliği vardır. Böyle bir topoloji oluşturmak için bir IBM WebSphere MQ kümesi kullanıyorsanız, bu öznitelikler aşağıdaki davranışa sahiptir.

PROXYSUB

- **PROXYSUB**, yetkili sunucu aboneliklerinin ne zaman yapıldığını denetleyen bir öznitedir. Bu özniteliği FIRSTUSE varsayılan değerinden neden değiştirmek isteyebileceğiyle ilgili ayrıntılar için bkz. [“Yönelme mekanizmalarında daha fazla” sayfa 51](#).
- Kümelenmiş bir konunun diğer öznitelikleriyle aynı şekilde, **PROXYSUB** özniteliği, yalnızca konunun tanımlandığı kuyruk yöneticisi değil, kümedeki her kuyruk yöneticisine dağıtılır. Bu, kümedeki her kuyruk yöneticisinde, diğer tüm kuyruk yöneticilerinde genel arama aboneliği oluşturan her kuyruk yöneticisinde anında sonuç elde eder. Bu işlemin sonucu, her kuyruk yöneticisinin diğer tüm kuyruk yöneticilerine küme gönderen kanalları yaratması ve yayınlanan tüm iletiler her kuyruk yöneticisine gönderilir.

PUBCOBE ve SUBSCOPE

PUBSCOPE ve **SUBSCOPE**, bu kuyruk yöneticisinin yayınları topolojide (yayınlama/abone olma kümesi ya da sıradüzeni) kuyruk yöneticilerine mi geçireceğini, yoksa kapsamı yalnızca

yerel kuyruk yöneticisiyle sınırlayıp geçirmedigini belirler. Eşdeğer işi, MQPMO_SCOPE_QMGR/MQSO_SCOPE_QMGR ' yi kullanarak programsal olarak yapabilirsiniz.

- **PUBSCOPE** Bir küme konusu nesnesi **PUBSCOPE (QMGR)** ile tanımlandıysa, tanım kümeyle paylaşılır, ancak o konuya dayalı yayınların kapsamı yalnızca yereldir ve kümedeki diğer kuyruk yöneticilerine gönderilmez.
- **SUBSCOPE** Bir küme konusu nesnesi **SUBSCOPE (QMGR)** ile tanımlandıysa, tanım kümeyle paylaşılır, ancak o konuya dayalı aboneliklerin kapsamı yalnızca yereldir; dolayısıyla, kümedeki diğer kuyruk yöneticilerine hiçbir yetkili abonelik gönderilmez.

Bu iki öznelik, bir kuyruk yöneticisini belirli konu başlıklarına ilişkin kümenin diğer üyeleriyle etkileşimde bulunan bir kuyruk yöneticisini yalıtım için genellikle birlikte kullanılır. Kuyruk yöneticisi, kümedeki diğer üyelerden ve diğer üyelerden yayınların yayınlarını ya da yayınlarını yayınlamayı ya da almalarını sağlar. Konu nesneleri alt konularda tanımlandıysa, bu durum yayınlanmasını ya da aboneliği önlememektedir.

Setting **SUBSCOPE** to MMGR on a local definition of a topic does not prevent other queue managers in the cluster from propagating their proxy subscriptions to the queue manager if they are using a clustered version of the topic, with **SUBSCOPE (ALL)**. However, if the local definition also sets **PUBSCOPE** to MMGR those proxy subscriptions are not sent publications from this queue manager.

Birden çok küme konu tanımlaması

Yerel bir konu tanımlaması, aynı adı içeren uzaktan tanımlanmış bir küme konu tanımlamasını geçersiz kılar. Bir kümedeki farklı kuyruk yöneticilerindeki aynı küme konusuna ilişkin birden çok tanımlama yaratılması da mümkündür. Bu senaryoların her ikisi de biraz dikkat gerektiriyor, ancak bu konuda nedenler açıklanıyor.

Kümelenmiş kuyruklar için olduğu gibi, bir kümede aynı küme konusu nesnesinin birden çok tanımı olması, her birinde farklı özelliklerin tanımlanması sağlar. Konu tanımlamasının hangi sürümünün kümede her kuyruk yöneticisi tarafından görüldüğünü belirlemek kolay değildir ve bu nedenle beklenen davranışı belirlemek oldukça zordur.

İki ya da daha fazla küme konu tanımlaması, tek bir konu dizgisi için, farklı özneliklere sahip ya da birden çok kümede var olan iletiler ([AMQ5465](#) & [AMQ5466](#)) hata günlüğüne yazılır ve en son alınan küme konu tanımı kullanılır.

Küme konusu anasistem kuyruk yöneticisi konu tanımını silmemeli ve kümelenmiş konunun kümenin tüm üyeleri tarafından bilinmesini sağlamak için kümede kalır. Küme konusu tanımı, tam havuz kuyruğu yöneticileri tarafından ve kısmi küme havuzlarındaki diğer tüm kuyruk yöneticileri tarafından önbelleğe alındığından, bu anasistem kuyruk yöneticisinin sürekli olarak kullanılabilir olması zorunlu değildir. Bu önbelleğe alma olanağı, anasistem kuyruk yöneticisi kullanılmazken en az 60 gün kullanılabilir durumda olur. Bu konuyla ilgili daha fazla bilgi için bkz. "[Yayınlama/abone olma küme kuyruğu yöneticilerine ilişkin temel roller](#)" sayfa 75.

Bir küme konusu tanımlamasının yerel olarak geçersiz kılınması

Kümedeki belirli kuyruk yöneticilerindeki kümelenmiş bir konunun davranışını geçersiz kılmak gerekebilir. Bu geçersiz kılma işlemi, bir yerel konu nesnesini tanımlayarak, aynı konu dizgisiyle bir küme konusu nesnesini geçersiz kılacak ve yalnızca yerel olarak bağlı abonelere yayınlamak için bu işlemi gerçekleştirilebilir.

Bir konunun yerel tanımlaması bir kuyruk yöneticisinden kümelenmiş bir konuyu geçersiz kılmak için yaratılırsa bile, kuyruk yöneticisi kümelenmiş konu tanımlamasını kullanarak kümenin diğer üyelerinden yetkili sunucu abonelikleri almaya devam eder. Varsayılan olarak, yerel olarak yayınlanan iletiler uzak kuyruk yöneticilerine, yetkili sunucu aboneliklerini onurlandırmak için gönderilmeye devam eder. Bu düzenleme gerekli değilse, bu kuyruk yöneticisine bağlı yayınlamaları için uygulamalarının yalnızca yerel aboneler tarafından yayınlanmasını sağlamak için yerel konu nesnesindeki **PUBSCOPE (QMGR)** değerini belirtin.

Küme konusu tanımlamasının değiştirilmesi

Bir küme konusu tanımlamasını değiştirmeniz gerekirse, bu tanımlama, küme konusu anasisteminde tanımlanmış olan aynı kuyruk yöneticisinde değiştirin. Kümedeki farklı bir kuyruk yöneticisinde aynı küme konusuna ilişkin bir tanım yaratmayın. Konu, aynı küme konusu için iki küme konusu ana makinesinde yeniden tanımlanmasını sağlar.

Bir küme konusu birden çok kez tanımlanması, çakışan tanımlamaları yaratır ve farklı kuyruk yöneticilerinin farklı tanımları farklı zamanlarda kullanması olasılığını oluşturur.

Küme konusu tanımlamasının kümede farklı bir kuyruk yöneticisine taşınması

Örneğin, bir küme konusu tanımlamasını kümedeki bir kuyruk yöneticisinden diğerine taşımanız gerekebilir; örneğin, bir kuyruk yöneticisini kümeden devreden çıkarma işlemi. Bir küme konusu tanımlamasını, yayın akışını kesintiye uğratmadan kümede farklı bir kuyruk yöneticisine taşımak için aşağıdaki adımları izlemeniz gerekir. Bu örnek, bir tanımlamayı QM1 'den QM2' a taşır.

1. QM2 üzerindeki küme konusu tanımlamasının bir kopyasını, QM1' un tanımlamasıyla aynı özniteliklerle birlikte yaratın.
2. Yeni tanımlamanın, tüm havuz kuyruğu yöneticilerine göre küme boyunca yayılmasını bekleyin. Yayma işlemi, (**DISPLAY CLUSTER**) komutunu kullanarak her küme üyesinde küme konuları görüntüleyerek ve QM2' den kaynaklanan bir tanımlamayı denetleyerek saptlanabilir.
3. Küme konu tanımını QM1' den silin.

Özgün tanım QM1 ' den silindikten sonra, özelliklerde çakışma olmadan, gerekirse QM2 üzerindeki tanımlamayı değiştirmek mümkündür.

Başarısız bir kuyruk yöneticisinde bir küme konusu tanımlamasının değiştirilmesi

In the previous [senaryo](#), it might not be possible to delete the definition from QM1 if QM1 is not available for a period of time. Bu senaryoda, varoluş olarak her iki tanımda da çalıştırılabiliriz kabul edilir.

If it then becomes a requirement to modify the clustered topic definition, it is possible to modify the version on QM2 in the knowledge that the QM2 definition is newer than the QM1 definition, and therefore prevails. Ancak, bu süre boyunca, çakışan bir küme konusu tanımlaması olduğu için, hatalar kuyruk yöneticilerinin hata günlüklerine yazılır. Yenilenen küme konusu tanımı yeniden başlatılabildiğinde QM1 ' den yenilenen küme konu tanımlamasını kaldırarak hatayı çözebilirsiniz.

Diğer bir seçenek olarak, QM1 hiçbir zaman kümeye geri dönmeyecekse (örneğin, yıkıcı bir donanım hatası sonrasında beklenmeyen bir kullanımdan çıkarma işlemi), kuyruk yöneticisini zorla çıkarmak için **RESET CLUSTER** komutu kullanılabilir. **RESET CLUSTER** , hedef kuyruk yöneticisinde barındırılan tüm konu nesnelerini otomatik olarak siler.

Bir kümede kümelenmiş yayınlama/abone olma engelini engelleme

Özellikle mevcut bir küme olan bir kuyruk yöneticilerine yayınlama/abone olma, performansın herhangi bir düşüşü için dikkatli bir şekilde planlanmalıdır.

Büyük bir IBM WebSphere MQ kümesine kümelenmiş bir konunun tanıtımı (birçok kuyruk yöneticisi içeren bir küme), kümedeki her bir kuyruk yöneticisinde ve bazı durumlarda performansın azaltılması ile ilgili ek yüklerin hemen sonuçlanabileceğini sağlar. Bu nedenle, yayınlama/abone olma aboneliğinin getirilmesi dikkatle planlanmalıdır. Ek bilgi için "[Küme konusu performansı](#)" sayfa 71 başlıklı konuya bakın.

Bir kümenin yayınlama/abone olma üst başlarını barındıramadığı bilinmektedir; **PSCLUS** kuyruk yöneticisi öznitelikli DEVRE Dışı olarak ayarlayarak, kuyruk yöneticilerinde kümelenmiş yayınlama/abone olma işlevselliğini devre dışı bırakmak mümkün olabilir.

PSCLUS ayarının DISABLE olarak ayarlanması, kuyruk yöneticisi işlevinin üç yönünün değiştirilmesini sağlar:

- Bu kuyruk yöneticisinin denetimcisi, artık kümelenmiş olarak bir konu nesnesini tanımlayamaz.
- Diğer kuyruk yöneticilerinden gelen konu tanımları ya da yetkili sunucu abonelikleri reddedilir (yanlış yapılandırmanın yöneticisine bilgi vermek için bir uyarı iletisi günlüğe kaydedilir).

- Tam havuzlar, bir konu tanımlaması aldıklarında, diğer tüm kısmi havuzlarla ilgili olarak her kuyruk yöneticisiyle ilgili bilgileri artık paylaşmaz.

PSCLUS , bir kümedeki her bir kuyruk yöneticisinin bir parametresi olsa da, kümedeki bir kuyruk yöneticisi alt kümesinde yayınlama/abone olma işlemini seçmeli olarak devre dışı bırakması amaçlanmaz. Başka bir şey dışında, bu yöntem sık sık hata iletilerinin, yetkili sunucu abonelikleri ve konu tanımlarının sürekli olarak görülmesine ve reddedilmesine neden olur. İdeal olarak, bu seçeneği kullanırken, kümedeki tüm kuyruk yöneticilerini devre dışı bırakılacak şekilde tutarlı bir şekilde ayarlayın. Bir kuyruk yöneticisinin bir ya da daha fazla yayınlama abone olma kümesine ya da kümesine katılacağı ve aynı zamanda bir ya da daha çok geleneksel küme ya da küme olduğu durumlarda, o kuyruk yöneticisinde **PSCLUS** ENABLE olarak ayarlanmış olmalıdır. Tüm havuzlarda devre dışı bırakma hakkında aşağıdaki bilgilere bakın.

Daha da önemlisi, kümedeki tüm havuz kuyruğu yöneticilerinde **PSCLUS** ' in DEVRE Dışı olarak ayarlanması, kümelenmiş bir kısmi havuzda kümelenmiş herhangi bir konu tanımlamasının, kümedeki diğer kuyruk yöneticilerini etkilemişlerinden etkilenmesini önler. Bu tür senaryolarda tutarsızlık, tam havuz kuyruğu yöneticilerinin hata günlüklerinde bildirilir.

Bir yayınlama abone olma kümesiyle geleneksel bir noktadan nokta kümesi çakışırken, her birinde ayrı bir havuzun ayrı bir kümesi kullanılması önemlidir. Bu düzenleme, konu tanımlamalarının ve 'tüm kuyruk yöneticisi' bilgilerinin yalnızca yayınlama/abone olma kümesinde akmasına izin verir.

Bu parametrenin kullanımı sırasında tutarsız yapılandırmalardan kaçınmaya yardımcı olan bazı yol gösterileri vardır. Bu kuyruk yöneticisinin üyesi olduğu herhangi bir kümede kümelenmiş konu nesnesi yok, ETKINED ' den DISABLEolarak değiştirilirken bu kuyruk yöneticisi üyesi olamaz. Bu işlevi devre dışı bırakmadan önce, bu tür konuların (uzaktan tanımlı olanlar bile) silinmelidir

PSCLUS ile ilgili daha fazla bilgi için bkz. [ALTER QMGR \(PSCLUS\)](#).

Küme konusu performansı

Küme başlıklarına ilişkin performans özellikleri, küme kuyruklarının performans özelliklerinden farklı olduğu için özel önem verilmelidir ve düşük kabul edilen kullanım, büyük ya da dengesiz kümelerde performans sorunları kaynağı olabilir.

Yayınlama/abone olma performansının etkisini azaltma

Bir kümedeki bir kuyruk yöneticisinde iki iş yükü kaynağı vardır: uygulama programları için doğrudan iletilerin işlenmesi ve kümeyi yönetmek için gerekli olan iletilerin ve kanalların işlenmesi. Tipik bir noktadan noktaya küme içinde, küme sistemi iş yükü büyük ölçüde kümenin üyeleri tarafından gerektiği şekilde açık bir şekilde sınırlanmıştır ("[Yayınlama/abone olma kümelerinin performans özellikleri](#)" sayfa 74' ta bir karşılaştırmaya bakın). Bu nedenle, çok büyük bir küme dışında, örneğin, binlerce kuyruk yöneticisi içeren bir küme dışında, kuyruk yöneticisi performansını dikkate aldığımızda kümenin yönetiminin performans etkisini büyük oranda indirebilirsiniz.

Bir yayınlama/abone olma kümesinde, kümelenmiş konular ve yetkili sunucu abonelikleri gibi bilgiler, tüm küme kuyruğu yöneticilerinin yayınlama/abone olma ileti alışverişlerine etkin olarak katılıp katılmayabilmesinden bağımsız olarak, bir kümenin tüm üyelerine yönlendirilir. Bu işlem, sistemde önemli bir ek yük oluşturabiliyor. Bu nedenle, küme yönetiminin kuyruk yöneticisi başarımını, hem zamanlamasındaki, hem de büyüklüğünün etkisini göz önünde bulundurmanız gerekir.

Bir kümenin performansına ilişkin yayınlama/abone olma kümesi yönetiminin etkisini azaltmak için aşağıdaki iki öneriyi dikkate alın:

1. Günün en yoğun zamanlarında küme, konu ve abonelik güncellemelerini gerçekleştirin.
2. Yalnızca küme zaten var olduğundan, var olan büyük bir kümeye yayınlama/abone olma konuları eklemeyi düşünüyorsanız, yayınlama/abone olma içinde yer alan çok daha küçük bir kuyruk yöneticisi altkümesi tanımlayıp tanımlamayacağınızı düşünün ve "çakışan" bir küme yapın. Bu küme, daha sonra küme konularının tanımlandığı kümedir. Bazı kuyruk yöneticileri şu anda iki kümeden geçse de, yayınlama/abone olma genel etkisi azaltılır:
 - a. Yayınlama/abone olma kümesinin boyutu küçüktür.
 - b. Yayınlama/abone olma kümesinde olmayan kuyruk yöneticileri, küme yönetimi trafiğindeki etkilerden çok daha az etkilenir.

Üreticileri ve tüketicileri dengeleme

Zamanuyumsuz ileti sistemi performansında önemli bir kavram *bakiye*dir. İleti tüketicileri ileti üreticileriyle dengelenmediği sürece, tüketilmeyen iletilerin arka günlüğü birden çok uygulamanın başarımını yükseltebilecek ve ciddi şekilde etkileyebilecek bir tehlike söz etmiyedir.

Bir noktadan noktaya ileti sistemi topolojisinde, ileti tüketicileri ve ileti üreticileri arasındaki ilişki kolayca anlaşılır. İleti üretimi ve tüketimi, kuyruğa göre kuyruk, kanala göre kanal gibi tahminler elde edebilirsiniz. Dengede bir eksiklik varsa, darboğazlar kolayca tespit edilir ve sonra giderilir.

Yayıncıların ve abonelerin bir yayınlama/abone olma topolojisinde dengelenip dengelendiğini çözmesi daha zordur. Kümelenmiş bir konuya çözülen her abonelikten başlayın ve konu üzerinde yayıncılar olan kuyruk yöneticilerine geri dönün. Her bir kuyruk yöneticisinden her aboneye akan yayınların sayısını hesaplar.

Kümedeki bir uzak kuyruk yöneticisinde bulunan bir abonelikte eşleşen her yayın (yetkili aboneliklere dayalı olarak) SYSTEM . CLUSTER . TRANSMIT . QUEUE ' e konmaktadır. Birden çok uzak kuyruk yöneticisinin bu yayın için yetkili sunucu aboneliği varsa, iletinin birden çok kopyası iletim kuyruğuna konursa, farklı küme gönderen kanalları için hedeflenmiş olur.

Bu yayınlar, uzak kuyruk yöneticilerindeki SYSTEM . INTER . QMGR . PUBS kuyruğunda hedeflenir. Her kuyruk yöneticisi, kuyruğa gelen iletileri işler ve bunları kuyruk yöneticisinde doğru aboneliklere teslim eder.

Bu nedenle, darboğazların ortaya çıkabileceği aşağıdaki noktalarda yükü izleyin:

- Bireysel abonelik kuyrukları kendileri:
 - Bu darboğaz, abone olunan uygulamanın yayınlanmakta olduğu kadar hızlı bir şekilde yayınlarını tüketmediğini ima eder.
- SYSTEM . INTER . QMGR . PUBS kuyruğu:
 - Kuyruk yöneticisi, bir ya da daha çok uzak kuyruk yöneticisinden, bu yayınları yerel aboneliklere dağıtabileceğinden daha hızlı bir şekilde alıyor.
- Yayınlama kuyruk yöneticisi, yayınlama kuyruğu yöneticileri ve küme iletim kuyrukları arasındaki küme kanalları (varsayılan olarak SYSTEM . CLUSTER . TRANSMIT . QUEUE):
 - Bir ya da daha fazla küme kanalı çalışmıyor ya da yerel kuyruk yöneticisinde, kanallardan daha hızlı bir şekilde yayınlanmakta olan iletiler uzak kuyruk yöneticisine teslim edilebiliyor.
- Yayınlama uygulaması kuyruğa alınmış bir yayınlama/abone olma arabirimi kullanıyorsa, SYSTEM . BROKER . DEFAULT . STREAM kuyruğu ve SYSTEM . QPUBSUB . QUEUE . NAMELIST içinde listelenen diğer akış kuyrukları ve SYSTEM . BROKER . DEFAULT . SUBPOINT kuyruğu ve diğer tüm alt nokta kuyrukları SYSTEM . QPUBSUB . SUBPOINT . NAMELIST' te listelendiği gibi, ayrıca dikkat edilmesi gereken noktalar:
 - İletiler yerel yayınlama uygulamaları tarafından yerel kuyruk yöneticisinden daha hızlı yerleştiriliyor, iletileri işleyebilir.

Abonelik performansına ilişkin dikkat

Daha önce açıklandığı gibi, bir kuyruk yöneticisinde, kümelenmiş bir konuya çözülen bir konu dizgisi için bir abonelik yapıldığında, o kuyruk yöneticisinin kümedeki diğer her kuyruk yöneticisinin konuya ilişkin bir yetkili aboneliği olduğundan emin olması gerekir. Bu sonucu elde etmek için kuyruk yöneticisi, kümedeki diğer tüm kuyruk yöneticisine bir yetkili abonelik iletileri yaratır ve gönderir.

Varsayılan yapılandırmanın, kümelenmiş bir konuya abonelik yaratmanın yeni yetkili sunucu abonelikleriyle sonuçlanmaması, yerel kuyruk yöneticisinde zaten aynı konu dizgisinin olması nedeniyle, varsayılan yapılandırmaların gönderilmesine neden olmaz. Bu durumda, gelen yayınlar yalnızca konu dizgisine ilişkin özgün aboneliğin değil, eşleşen tüm aboneliklere teslim edildiğinden, ek yetkili abonelik aboneliği gerekmez.

Varsayılan yapılandırmaya alternatif olarak bkz. [“Yetkili sunucu aboneliklerinin geçersiz kılınması” sayfa 73.](#)

Abonelik seçicileri dikkate alınmaz, bu nedenle aynı konu dizgisine iki abonelik var, ancak farklı seçicilerle birlikte hala yetkili abonelikleri paylaşıyorsunuz. Bu durum ayrıca, yayının aboneliğin seçiciyle eşleşme de, konu dizgisiyle eşleşen yayınların abone kuyruk yöneticisine geçirilmiş olduğu anlamına da gelebilir.

Yetkili abonelik iletisine eşdeğer bir ileti yaratılır ve bir konu dizgisine son abonelik kuyruk yöneticisinden silindiğinde tüm kuyruk yöneticilerine gönderilir. Bu işlem, uzak kuyruk yöneticilerinden yetkili sunucu aboneliklerini kaldırır.

Bu nedenlerden dolayı, küme ve abonelik sıklığı farklı konu dizgilerinin boyutu, kümenin kendisi üzerinde önemli bir yük verebilir ve kümenin planlanması ve yayınlama/abone olma uygulamaları tarafından kullanılacak konular düşünüldüğünde dikkate alınması gerekir.

Yetkili abonelik trafiğinden sistemdeki yükü dikkate aldığımızda, “Üreticileri ve tüketicileri dengeleme” sayfa 72 bölümünde listelenen kuyrukların izlenmesine ek olarak, aşağıdaki kuyrukları izleyin.

- SYSTEM.INTER.QMGR.FANREQ kuyruğu, abone kuyruğu yöneticisinde kuyruğa girmektedir.
- SYSTEM.INTER.QMGR.CONTROL (CONTROL)cluster, kümedeki diğer tüm kuyruk yöneticilerindeki kuyrukta.

Bu kuyruklardaki önemli ileti birikimlerinden biri, abonelik değişikliği hızının sistem için çok büyük olduğunu ya da bir kuyruk yöneticisinin kümede düzgün bir şekilde çalışmadığını gösterir. Yayınlama/abone olma desteği devre dışı bırakıldığı için (ALTER QMGR içindeki **PSMODE** konusuna bakın) ya da daha fazla araştırma gerektiren bir sorun ortaya çıktı. Bu noktada kuyruk yöneticisi hata günlüklerini denetleyin.

Yetkili sunucu aboneliği trafiğini azaltma

Yetkili sunucu aboneliği ek yükü yüksekse, bunu azaltmak için adımlar atılmalıdır. Bu, genel konu birleştirmesi ya da kuyruklar arası yönetici yayınları için bir yayın modeline geçilmesi yoluyla mümkün olabilir.

Sistem için kaynakların genel yükünü azaltmak için bir şekilde birleştirilip birleştirilmeyeceklerini görmek için konu dizgilerinin kullanımının değerlendirilmiş olduğu genel bir yayınlama/abone olma öneridir. Pek çok ayrı, geçici konu dizgisinin kullanımı, yayıncıların ya da aboneliklerin bağlı olduğu sistemdeki her kuyruk yöneticisinde bazı yönetim ek yükleri sağlar. Konu dizgilerinin sayısını ve geçici niteliyi azaltmak ve bu nedenle yayıncılar ve abonelikler sistem üzerindeki etkiyi azaltır.

Yetkili abonelik trafiğini azaltmaya yönelik bir yöntem, aynı kuyruk yöneticisindeki aynı konu dizgisine aboneliklerin yerini bulmaktan başka bir yöntemdir. Bu yöntem, bu kuyruk yöneticisinin, her biri aynı konu dizgisindeki kendi abonelikleri için, birden çok kuyruk yöneticisi gönderme yapan birden çok kuyruk yöneticisine sahip olmak yerine, diğer kuyruk yöneticilerine tek bir yetkili sunucu aboneliği göndermesini sağlar. Bu uygulama, küme genelinde yayın yönlendirmesini de optimize eder.

Yetkili sunucu aboneliklerinin geçersiz kılınması

Bazı durumlarda, bir kümenin içinde abone olunan ayrı konu dizgileri büyük ve sürekli olarak değişirse, abonelik yayılımı modelinden bir yayın yayın modeline geçilmesi daha iyi olabilir. Bu tercih edilen model, herhangi bir küme konusundaki her yayının, bu kuyruk yöneticilerindeki aboneliklerin varlığından bağımsız olarak, kümedeki her kuyruk yöneticisine otomatik olarak gönderileceği yer olur.

Alıcı kuyruk yöneticileri iletileri var olan yerel aboneliklere iletebilirler ya da iletiyi atar. Bu modelde, aboneliklerin varlığına dayalı olarak tek tek yetkili sunucu aboneliklerine gerek yoktur ve bu aboneliklerin silinmesine gerek yoktur. Bu kipte çalışırken, tüm yayınlar tüm kuyruk yöneticilerine gönderildiğinden, yayınlanan ileti kaynağı yükünün de yükselmesi olası olur. Bu nedenle, kümedeki kuyruk yöneticilerinin bu ek yükü işleyebilmek için gereken kapasiteye sahip olması gerekir.

Aşağıdaki yapılandırma adımlarını kullanarak bir yayın modelini etkinleştirin:

1. Abonelikleri barındıran her kuyruk yöneticisi, kümelenebilir konularla yerel aboneliklerle eşleşen yetkili sunucu aboneliklerini göndermeyecek şekilde yapılandırılmalıdır. Bu yapılanış, küme başlıklarını

tanımlamadan ya da kümedeki aboneliklerin yaratılmasından önce, her kuyruk yöneticisi qm.ini dosyasında aşağıdaki ayar değiştirgenin ayarlanmasını gerektirir:

```
TuningParameters:  
psscProxySubFlags=1
```

2. Ayarlama parametresi ayarlandıktan sonra, tüm kuyruk yöneticilerinin yeniden başlatılması gerekir.
3. Kuyruk yöneticileri yeniden başlatıldıktan sonra kümelenecek konu/konular tanımlanabilir. Her küme konusu **PROXYSUB** için FORCEolarak ayarlanmalıdır.

Davranışı tersine çevirme

“Yetkili sunucu aboneliklerinin geçersiz kılınması” sayfa 73’ ta daha önce açıklanan işlem kipini tersine çevirmek için aşağıdaki adımları kullanın:

1. Her kuyruk yöneticisi için, ayarlama parametresini qm.ini dosyasından kaldırın.
2. Her kuyruk yöneticisini yeniden başlatın.
3. Abonelikleri barındıran her kuyruk yöneticisi için **REFRESH QMGR TYPE (PROXYSUB)** komutunu verin.
4. Kümelenecek konu ya da konular üzerinde **PROXYSUB** ögesini FIRSTUSE olarak ayarlayın.



DİKKAT: Bu davranışın etkinleştirilmesi ve tersine çevrilmesinde, tüm adımlar belgelenmiş sırayla tamamlanmazsa, aboneliklere ilişkin doğru yayın akışı gerçekleşmeyebilir.

Not: PROXYSUB ayarının etkileri (FORCE için)

Bu konuda daha önce açıklandığı gibi, **PROXYSUB (FORCE)** konu özniteliği yetkili abonelik trafiğini azaltabilir, ancak bu, dikkatli kullanılması gerektiğini ifade eder. **PROXYSUB (FORCE)** özniteliği, yalnızca konunun tanımlı olduğu kuyruk yöneticisi değil, kümedeki her kuyruk yöneticisine dağıtılır. Bu, kümedeki her kuyruk yöneticisinde, diğer tüm kuyruk yöneticilerinde genel arama aboneliği oluşturan her kuyruk yöneticisinde anında sonuç elde eder. Bu işlemin sonucu, her kuyruk yöneticisinin diğer tüm kuyruk yöneticilerine küme gönderen kanalları yaratması ve yayınlanan iletilerin her kuyruk yöneticisine gönderilmesi olur.

Bu özelliğin büyük ya da yoğun bir kümede ayarlanması, sistem kaynaklarında ek yük ile sonuçlanabilir.

Yayınlama/abone olma kümelerinin performans özellikleri

Bir yayınlama/abone olma kümesinin (kuyruk yöneticisi, konu ya da abonelik eklenmesi gibi), kümede çalışan uygulamaların performansını etkilediği gibi, bir yayınlama/abone olma kümesinin değişen özniteliklerinin nasıl değiştirileceğini göz önünde bulundurmanız önemlidir.

İki yönetim görevi bakımından bir yayınlama/abone olma kümesiyle bir noktadan noktaya bir kümeyi karşılaştır.

Öncelikle, bir nokta kümesi noktası:

1. Yeni bir küme kuyruğu tanımlandığında, hedef bilgiler tam havuz kuyruğu yöneticilerine gönderilir ve yalnızca bir küme kuyruğuna ilk başvurduğunda diğer küme üyelerine gönderilir (örneğin, bir uygulama açmayı denediğinde). Bu bilgiler, kuyruk yöneticisi tarafından, kuyruğa her erişildiğinde uzaktan bilgi alma gereksinimini ortadan kaldırmak için yerel olarak önbelleğe alınır.
2. Bir kümeye kuyruk yöneticisi eklenmesi, diğer kuyruk yöneticilerindeki yükü doğrudan etkilemez. Yeni kuyruk yöneticisine ilişkin bilgiler tüm havuzlara yönlendirilir, ancak kümedeki diğer kuyruk yöneticilerinden yeni kuyruk yöneticisine yönelik kanallar yaratılır ve yeni kuyruk yöneticisinden ya da yeni kuyruk yöneticisinden trafik akışı başlatıldığında başlatılır.

Kısaca, bir noktadan noktaya kümedeki bir kuyruk yöneticisinde bulunan yük, uygulama programları için işleyeceği ileti trafiğiyle ilgilidir ve doğrudan kümenin boyuyla ilgili değildir.

İkinci olarak, bir yayınlama/abone olma kümesi:

1. Yeni bir küme konusu tanımlandığında, bilgiler tam havuz kuyruğu yöneticilerine, oradan da doğrudan kümenin tüm üyelerine hemen gönderilir ve önceden başlatılmamışsa, tüm havuzlardan kümenin her bir üyeye kanalların başlatılmasına neden olur.
2. Bir abonelik yeni bir konu dizgisindeki bir küme konusuna oluşturulduğunda, bilgiler doğrudan o kuyruk yöneticisinden kümenin diğer tüm üyelerine hemen yönlendirilir ve önceden başlatılmamışsa, bu kuyruk yöneticisinden kümenin her bir üyesine kanalların başlatılmasına neden olur.
3. Yeni bir kuyruk yöneticisi var olan bir kümeye katıldığında, tüm kümelenmiş konularla ilgili bilgiler, havuz kuyruğu yöneticilerinden ona yönlendirilir. Yeni kuyruk yöneticisi, kümedeki tüm aboneliklerin bilgilerini kümenin tüm üyeleriyle birlikte kümedeki tüm küme konularına eşitler ve yeni kuyruk yöneticisinden kümenin her bir üyesi için kanalların yaratılmasına ve her bir üyeye başlatılmasına neden olur.

Özette, kümedeki herhangi bir kuyruk yöneticisinde küme yönetimi yükü, her kuyruk yöneticisinde bu küme konularının yerel kullanılmasından bağımsız olarak, küme içindeki kuyruk yöneticisi sayısı, kümelenmiş konular ve yetkili sunucu abonelikleriyle birlikte büyür.

Yayınlama/abone olma küme kuyruğu yöneticilerine ilişkin temel roller

Noktadan noktaya iletişim kümelerine benzer şekilde, bir yayınlama/abone olma kümesinde kuyruk yöneticilerine ilişkin iki temel rol vardır; tam havuz kuyruğu yöneticileri ve küme konusu anasistemleri olarak.

Tam havuz

Tam bir havuz kuyruk yöneticisi, nesne tanımlamalarını bir kümenin diğer üyelerine zorlama rolüne sahiptir; yayınlama/abone olma kümelerinde, kümelenmiş konu nesnesi tanımlamalarını kümenin diğer üyelerine zorlar.

Küme konusu anasistemi

Küme konusu anasistem, kümelenmiş bir konu nesnesinin tanımlandığı bir kuyruk yöneticidir. Yayınlama/abone olma kümesindeki herhangi bir kuyruk yöneticisinde kümelenmiş konu nesneleri tanımlayabilirsiniz. Küme konusu nesnesi, tam havuz kuyruğu yöneticilerine yönlendirilir; daha sonra, bunu, kümedeki diğer kuyruk yöneticilerine ve kümedeki herhangi bir kuyruk yöneticisinde çalışan yayıncılar ve aboneler tarafından kullanılmak üzere önbelleğe alınmış olan tüm kuyruk yöneticilerine yönlendirir.

Kullanılabilirlik ve yönetim

Kümede küme konusu tanımlarının kullanılabilirliğini en üst düzeye çıkarmak için bir kümede iki tam havuz tanımlamanız gerekir.

Kuyruğa alınan ileti sistemi kümelerinde olduğu gibi, birçok bilgisayar arasında yalnızca iki adet son derece kullanılabilir bilgisayar içeren yayınlama/abone olma kümelerinde, yüksek düzeyde kullanılabilir bilgisayarları tam havuz olarak tanımlamak iyi bir uygulamadır.

Kuyruğa alınan kümelerde, kümedeki birden çok kuyruk yöneticisinde aynı küme kuyruğunu *tanımlayarak* aynı küme kuyruğunu kullanarak küme kuyruğunun kullanılabilirliğini ve verimini artırabilirsiniz. Bundan sonra iletiler, bunlar arasında dengeli bir iş yükü olur. Buna karşılık, yayınlama/abone olma kümelerinde, kümelenmiş bir konu kümedeki tüm kuyruk yöneticilerinde *kullanılabilir*, ancak yayınlama/abone olma trafiğindeki iş yükü dengelemesi gerçekleştirilmez. Bunun yerine, ayrı abonelikler ve yayıncılar, yayınlama/abone olma yükünü yaymak için farklı kuyruk yöneticilerine dağıtılmalıdır. Küme konusunu tanımladığınız kuyruk yöneticisi kullanılamaz hale gelirse, diğer kuyruk yöneticileri konu için yayınlama/abone olma isteklerini işlemeye devam eder.

Ancak, küme konusu nesnesini tanımladığınız kuyruk yöneticisi bir daha kullanılabilir kılınmadıysa, sonunda diğer kuyruk yöneticilerindeki önbelleğe alınan konu nesneleri silinir ve konu kullanılamaz olur. Bu işlem, konu tanımlamasının kullanılmadığı zamandan itibaren en az 60 gün (konu tanımlamasının son yenilenme durumuna bağlı olarak) sonra gerçekleşir.

Küme konu nesnelere tanımladığınız kuyruk yöneticisini kurtarmak için 60 günlük süre ile, bir küme konusu anasistemi yüksek kullanılabilirlikli kılmak için özel ölçümler almaya gerek yoktur. 60 günlük süre teknik sorunlar için yeterli olacaktır; 60 günlük süre sadece idari hatalar nedeniyle aşılabilir. Bu olasılığı en aza indirmek için, küme konusu anasistemi kullanılamıyorsa, kümenin tüm üyeleri, önbelleğe alınan küme

konu nesnesinin yenilenmemiş olduğu hata günlüğü iletilerini saat başına yazar. Bu iletiye, küme konusu nesnesinin tanımlı olduğu kuyruk yöneticisinin çalışmakta olduğundan emin olun.

Diğer kuyruk yöneticilerindeki aynı küme konusu nesnesinin tanımlanması uygulamasını benimseyebilirsiniz. Her tanım, diğer küme konusu anasistemleri de dahil olmak üzere kümedeki diğer kuyruk yöneticilerine itilmekte olan ek bir küme konusu nesnesiyle sonuçlanır. Şimdi, 60 gün boyunca bir küme konusu anasistemi kullanılamaz hale gelirse, diğer anasistemlerden yalnızca küme konusu nesnesinin sürümü kaldırılır. Küme konusu nesnesinin diğer sürümleri kalır. Bir kümedeki belirli bir konuya ilişkin tüm tanımların aynı olması gereklidir; tersi durumda, bir kuyruk yöneticisi tarafından hangi konu tanımlamasının kullanıldığını kesin olarak kabul etmek zor olur. Herhangi bir anasistemdeki en son kopya, her zaman kullanılan küme konusu nesnesidir.

Artan idari karmaşıklığa karşı, birden çok küme konusu tanımının eklenen korumasını tart: karmaşıklığın artması, daha büyük bir insan hatası olasılığı daha ortaya çıkar.

Kümelenmiş bir kuyruğun barındırılmasına farklı olarak, kümelenmiş bir konu tanımlaması için anasistem kuyruk yöneticisi ek uygulama iletileri trafiği tanımaz. Bu trafik, aboneliklerin yaratıldığı kuyruk yöneticileriyle ve yayınlanan iletiler ile sınırlanır. Bu kümelenmiş konuyu, ikisini de yapmayan bir kuyruk yöneticisine de anasistem ile birlikte barınmak mümkündür. Bu durum, zorunlu olmamakla birlikte, genellikle bu kuyruk yöneticileri daha yüksek kullanılabilirlik düzeyleriyle yetkilendirilebilecek ve üzerinde daha sıkı denetim denetime sahip olabileceği için, küme için tam havuz kuyruğu yöneticilerine ilişkin kümelenmiş konulara ev sahipliği yapmak genellikle mantıklı olur. Bu düzenleme, tanımların yanlış değiştirilmesi ya da silinmesi olasılığını azaltır veya kuyruk yöneticisini bile azaltır.

Çakışan küme desteği ve yayınlama/abone olma

IBM WebSphere MQ kümeleriyle tek bir kuyruk yöneticisi birden çok kümenin üyesi olabilir. Bu düzenleme çakışan kümeler olarak bilinir. Bir yayınlama/abone olma kümelerindeki kümelenmiş konular, kümeler bir kuyruk yöneticisinde çakışınca kuyruklara farklı davranır. Kümelenmiş kümelerle kümelenmiş yayınlama/abone olma özelliğini kullanırken bu davranış açıkça anlaşılmalıdır.

Kuyruk için farklı olarak, bir konu tanımlamasını birden çok kümeyle ilişkilendirebilme yeteneği yoktur. Bu nedenle, bir kümede yaratılan yetkili sunucu abonelikleri kapsamı, kümelenmiş konunun tanımlandığı tek kümeyle sınırlıdır. Ancak, her kuyruk yöneticisinin, üyesi oldukları herhangi bir kümeden tüm yerel konuları ve bilinen kümelenmiş konuları içeren tek bir konu ağacı vardır. Bu nedenle, yayınlama/abone olma davranışlarının anlaşılması zor bir sistem gibi bir sistemin mimarı için bir neden olabilir.

Birden çok yayınlama/abone olma kümesini bütünleştirme

Noktadan noktaya iletişim iletileri için, tek bir kuyruk yöneticisi yapmanın bir nedeni, birden çok kümenin bir üyesinin iki küme arasında bir küme ağ geçidi yaratmasıdır. Bu konuyla ilgili daha fazla bilgi için [Çakışan kümeler](#) başlıklı konuya bakın. Bu küme ağ geçidi, bir kümedeki noktadan noktaya iletilerin başka bir kümedeki sorgulara yöneltilmesini sağlar. Yayınlama/abone olma kümeleri, geleneksel kuyruk yöneticisi kümelerinden örtüşme yeteneğini devralır. Ancak, yayınları ve abonelikleri bir kümeden diğerine yöneltilmek için bu mekanizmayı kullanamazsınız.

Bunun yerine, bir kümedeki kuyruk yöneticilerinden diğerine yayınları ve abonelikleri geçirmek için, bir yayınlama/abone olma sıradüzeni kullanarak kuyruk yöneticilerini birbirine bağlamanız gerekir. Bu düzenleme, diğer kümedeki başka bir kuyruk yöneticisiyle bir kümedeki bir kuyruk yöneticisi arasında belirtik olarak bir üst-alt sıradüzensel ilişki yaratılarak elde edilebilir. Bu ilişki, kümeler arası tüm yetkili sunucu abonelikleri ve dolayısıyla eşleşen yayınlar arasındaki akışı etkinleştirir. Bu ilişkiyle ilgili daha fazla bilgi için bkz. [“Sıradüzenleri yayınlama/abone olma” sayfa 79.](#)

Kümeler arasındaki yayınların ve aboneliklerin akışını sınırlandırmak için, her iki kümede de bulunan bir ağ geçidi kuyruk yöneticisi kullanılmaktadır; bkz. [“Birden çok kümede konu alanlarını birleştirin ve yalıtın” sayfa 96.](#)

Çakışan kümeler, tek konu ağacı

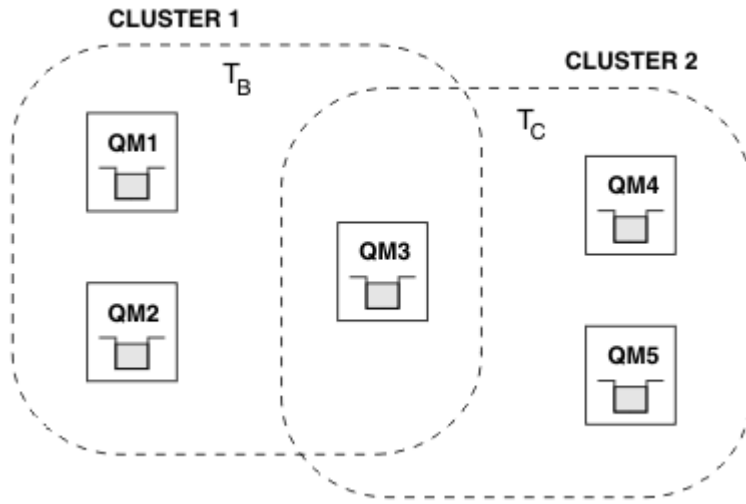
Her kuyruk yöneticisinin, yerel konuları ve bilinen tüm kümelenmiş konuları içeren tek bir [Konu ağacı](#) vardır. Her ikisi de yayınlama/abone olma özelliğini kullanarak, çakışan iki kümeyle ilgili daha fazla dikkat edilmesi, her bir kümedeki bir kuyruk yöneticisinin aynı ada sahip kümelenmiş bir konuyu tanımlamasına ya da aynı konu dizgisine sahip farklı adlandırılmış kümelenmiş konular tanımlamasının olanaklı olduğunu ifade eder. Her iki kümenin üyesi olan kuyruk yöneticilerine, her küme için bir tane olmak üzere birden çok küme konusu tanımından haberdar oldukları zaman çakışmalar ortaya çıkar.

Bir sorun kuyruk yöneticisi tarafından raporlanır, ancak kuyruk yöneticisi yalnızca en son kümelenmiş konu tanımlamasını kullanarak çalışmaya devam eder. Bu nedenle, davranış belirleyici olmayan bir davranışa neden olur ve bu davranışa güvenilemez.

Bu nedenle, kümelenmiş yayınlama/abone olma özelliğini kullanan çakışan kümeler, tüm kümeleri kapsayacak şekilde konu tanımlaması ad alanını dikkate almalıdır ve konu nesnelere adını ve konu dizgilerini buna göre yapılarını dikkate almalıdır. Daha sonra her iki kümeyi de yayınlamak ve her iki kümeye abone olmak için çakışma üst üste kuyruk yöneticilerini kullanabilirsiniz.

Şekil 33 sayfa 77, T_B ve T_C , çakışmayan konu tanımlamalarıdır. Küme çakışması içinde QM3'e bağlı bir yayınlama, kendi kümelerindeki her iki konuyu da yayınlatabilmektedir. Çakışan QM3'e bağlı bir abone, her iki kümeden de konulara abone olabilir.

Şekil 33 sayfa 77'i düşünmenin alternatif bir yolu, yetkili sunucu aboneliklerini dikkate almasıdır. An application connected to queue manager QM3, subscribing on a topic that resolves to topic object T_E (which exists only in CLUSTER 1) results in proxy subscriptions being sent from queue manager QM3 to queue managers QM1 and QM2 only. An application connected to queue manager QM3 subscribes on a topic that resolves to topic object T_C (which exists only in the CLUSTER 2). The subscription results in proxy subscriptions being sent from queue manager QM3 to queue managers QM4 and QM5 only.



Şekil 33. Çakışan kümeler: her biri farklı başlıklara abone olan iki küme

Yayıncılar ve aboneler, çakışmada yer almayan kuyruk yöneticilerine, yalnızca kümelerindeki konuları yayınlatabilirler ve abone olabilirler; örneğin, QM2 üzerindeki bir konu dizgisine ilişkin bir abone, konuları kümelemeyen bağımsız olarak QM5'ten yayınlanan aynı konu dizgisine yayınlanan iletiler almaz. Bu düzenlemeye ulaşmak için bir yayınlama/abone olma hiyerarşisi gereklidir.

Çakışan kümeler, genel arama abonelikleri

Bu konunun önceki bölümünde yer alan nedenlerden dolayı, birden çok kümenin üyesi olan bir kuyruk yöneticisinde bulunan konulara abone olmak için joker karakterler kullanılırken dikkatli olunmalıdır.

Önceki örnekte, iki konu nesnesinin şu şekilde yapılandırıldığını varsayın:

- T_B : Konu adı 'Football', küme 'CLUSTER1'. konu dizisi '/Sport/Football'
- T_C : Konu adı 'Tennis', küme 'CLUSTER2'. konu dizisi '/Sport/Tennis'

Bu senaryoda, kümelenmiş iki konu, konu ya da konu dizgisinde örtüşme olmadan ayrı olarak ayrı ayrı olur.

QM3'e bağlı bir uygulama, '/Sport/Football' aboneliği ve '/Sport/Tennis' aboneliği oluşturabilir. Daha sonra bu iki kümeden herhangi bir yayın alırdı. However, as described in "Yönetimle ilgili konu nesnelere" sayfa 35, if they were to subscribe to '/Sport/#', with the intent of receiving publications on both '/Sport/Football' and '/Sport/Tennis', this model is not recognized as a clustered topic in either cluster and therefore no proxy subscriptions would be created. Daha sonra, her iki kümede bulunan diğer kuyruk yöneticilerinden gelen yayınları kaçırmazlar.

As already described, it is not valid to create a clustered topic for '/Sport/#' in both CLUSTER 1 and CLUSTER 2, because these clustered topics would conflict and informational messages are written to the error logs to indicate this. However, it is 'allowed' to create such a topic in just one of the clusters, say CLUSTER 1. Now a subscription to '/Sport/#' in QM3 would result in proxy subscriptions being sent to the queue managers in CLUSTER 1 only, so still, publications to '/Sport/Tennis' from QM4 or QM5 would still fail to be received.

Bu senaryodaki tek çözüm, iki ayrı abonelik oluşturmaya devam etmek.

Yayınlama/abone olma kümeleri için Kümeyi Yenile

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

The time taken from issuing the **REFRESH CLUSTER** command to the point that the queue manager regains a full knowledge of the necessary information for clustered publish/subscribe depends on the size of the cluster, the availability, and the responsiveness of the full repository queue managers.

Yenileme işlemi sırasında, yayınlama/abone olma kümesinde yayınlama/abone olma trafiğindeki kesinti gerçekleşir. Büyük kümeler için, **REFRESH CLUSTER** komutunun kullanımı devam ederken kümeyi bozabilir ve daha sonra, küme nesnelere tüm ilgili kuyruk yöneticilerine otomatik olarak durum güncellemeleri gönderdiğinde bundan sonra 27 gün aralıklarla küme bozabilir. Bkz. Büyük bir kümede yenilenme, kümenin performansını ve kullanılabilirliğini etkileyebilir. Bu nedenlerden dolayı, **REFRESH CLUSTER** komutu yalnızca IBM Destek Merkezi 'nin rehberliğinde bir yayınlama/abone olma kümesinde kullanılmalıdır.

Kümede kesinti, aşağıdaki belirtiler olarak dışarıdan görülebilir:

- Bu kuyruk yöneticisinde küme konularına ilişkin abonelikler, kümedeki diğer kuyruk yöneticilerine bağlı yayıncılardan gelen yayınları almamakta.
- Bu kuyruk yöneticisindeki küme konularına yayınlanan iletiler, diğer kuyruk yöneticilerindeki aboneliklere geçirilmiyor.
- Bu dönem içinde yaratılan bu kuyruk yöneticisinde bulunan küme konularına ilişkin abonelikler, kümenin diğer üyelerine tutarlı bir şekilde yetkili sunucu aboneliği göndermiyor.
- Bu süre içinde silinen bu kuyruk yöneticisinde bulunan küme konularına ilişkin abonelikler, yedek sunucu aboneliklerini kümenin diğer üyelerinden tutarlı bir şekilde kaldırmaz.
- İleti tesliminde 10 saniyelik duraksamalar ya da daha uzun süre.
- **MQP** hataları; örneğin, **MQRC_PUBLICATION_FAILURE**.
- Publications placed on the dead-letter queue with a reason of **MQRC_UNKNOWN_REMOTE_Q_MGR**

For these reasons publish/subscribe applications need to be quiesced before issuing the **REFRESH CLUSTER** command.

Ayrıca bkz. **REFRESH CLUSTER** için Usage Notes ve Clustering: Using REFRESH CLUSTER BEST PRACTICES.

Yayınlama/abone olma kümesindeki bir kuyruk yöneticisinde bir **REFRESH CLUSTER** komutu verildikten sonra, tüm küme kuyruğu yöneticileri ve küme konuları başarıyla yenileninceye kadar bekleyin, daha sonra yetkili sunucu aboneliklerini "Yetkili sunucu aboneliklerinin yeniden eşzamanlanması" sayfa 67 içinde açıklandığı biçimde yeniden eşitleyin. Bu düzenleme, küme gönderen kanallarının bu kuyruk yöneticisinden kümedeki diğer tüm kuyruk yöneticilerine başlatılmasını gerektirir. Tüm yetkili abonelikler doğru şekilde yeniden eşitlenmiş olduğunda, yayınlama/abone olma uygulamalarınızı yeniden başlatın.

If a **REFRESH CLUSTER** command is taking a long time to complete, monitor it by looking at the CURDEPTH of SYSTEM.CLUSTER.COMMAND.QUEUE .

İlgili kavramlar

REFRESH CLUSTER çalıştırılırken uygulama sorunları görüldü

Kümeleme: REFRESH CLUSTER en iyi uygulamaları kullanma

İlgili başvurular

MQSC Commands başvurusu: REFRESH CLUSTER

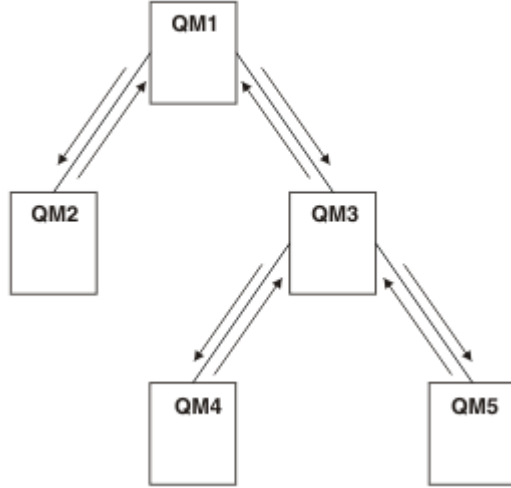
Sıradüzenleri yayınlama/abone olma

Kuyruk yöneticileri bir sıradüzende birlikte gruplandırılabilir; burada sıradüzen, doğrudan bağlı bir ya da daha fazla kuyruk yöneticisi içerir. Kuyruk yöneticileri, bir bağlantı süresi üst ve alt ilişkisi kullanılarak birbirine bağlanır. İki kuyruk yöneticisi ilk kez bir araya bağlandığında, alt kuyruk yöneticisi üst kuyruk yöneticisine bağlanır.

Üst ve alt kuyruk yöneticileri bir sıradüzene bağlandığında, kuyruk yöneticilerini sıradüzenden çıkarıncaya kadar bunlar arasında işlevsel bir farklılık yoktur.

Not: IBM WebSphere MQ sıradüzensel bağlantılar, PSMODE kuyruk yöneticisi özneliğinin ENABLED olarak ayarlanmasını gerektirir.

Hierarchy



Şekil 34. Basit yayınlama/abone olma sıradüzeni

Kuyruk yöneticisini aracı sıradüzenine bağla

Bir aracı sıradüzenini değiştirmek için yerel bir kuyruk yöneticisini üst kuyruk yöneticisine bağlayabilirsiniz.

Başlamadan önce

1. Kuyruğa alınmış yayınlama/abone olma kipini etkinleştirin. Bkz. [Kuyruktaki yayınlama/abone olma işlemini başlatma](#).
2. Bu değişiklik, bir IBM WebSphere MQ bağlantısı kullanılarak üst kuyruk yöneticisine yapılır. Bağlantıyı kurmanın iki yolu vardır.
 - Kuyruk yöneticilerinin bir IBM WebSphere MQ kümesine bağlanması için [Kümeyle kuyruk yöneticisi eklenmesi](#) başlıklı konuya bakın.
 - Üst kuyruk yöneticisiyle aynı adı taşıyan bir iletim kuyruğunu ya da kuyruk yöneticisi diğer adını kullanarak noktadan noktaya iletişim kanalı bağlantısı kurun. Noktadan noktaya iletişim kanalı bağlantısı kurulmasına ilişkin ek bilgi için [WebSphere MQ distributed-messaging \(dağıtılmış ileti sistemi\) teknikleri](#) başlıklı konuya bakın.

Bu görev hakkında

Alt öğeleri üst öğelere bağlamak için ALTER QMGR PARENT (PARENT_NAME) runmqsc komutunu kullanın.

Dağıtılmış yayınlama/abone olma, kuyruk yöneticisi kümeleri ve kümelenmiş konu tanımlamaları kullanılarak gerçekleştirilir. IBM WebSphere MQ Version 6.0 ve WebSphere Message Broker Version 6.1 ve WebSphere Event Broker Version 6.1 ve önceki sürümleriyle birlikte çalışabilirlik için, kuyruğa alınmış

yayınlama/abone olma kipi etkinleştirildiği sürece Version 7.1 ya da sonraki kuyruk yöneticilerini bir aracı sıradüzenine bağlayabilirsiniz.

Yordam

QMGR ÜST ÖĞESİNİ DEĞİŞTİR (ÜST)

Örnek

İlk örnekte, QM2 'nin QM1alt ögesi olarak nasıl bağlanacağı ve daha sonra, bağlantısı için QM2 ' nin nasıl sorgulanacağı gösterilmektedir:

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2
alter qmgr parent(QM1)
  1 : alter qmgr parent(QM1)
AMQ8005: WebSphere MQ queue manager changed.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.
      QMNAME(QM2)                TYPE(LOCAL)
      STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
      QMNAME(QM1)                TYPE(PARENT)
      STATUS(ACTIVE)
```

Sonraki örnekte, bağlantıları için QM1 sorgulanırken elde edilen sonuç gösterilmektedir:

```
C:\Documents and Settings\Admin>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.
      QMNAME(QM1)                TYPE(LOCAL)
      STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
      QMNAME(QM2)                TYPE(CHILD)
      STATUS(ACTIVE)
```

Sonraki adım

Bir aracıda ya da kuyruk yöneticisinde, bağlı kuyruk yöneticilerindeki yayıncıların ve abonelerin kullanabileceği konular tanımlayabilirsiniz. Daha fazla bilgi için [Yönetim konusunu tanımlama](#) başlıklı konuya bakın.

İlgili kavramlar

[Akışlar ve konular](#)

[WebSphere MQ yayınlama/abone olma ileti sistemine giriş](#)

İlgili başvurular

[PUBSUB 'I GÖRÜNTÜLE](#)

Bir kuyruk yöneticisinin aracı sıradüzeninden bağlantısını kes

Aracı sıradüzenindeki bir üst kuyruk yöneticisiyle alt kuyruk yöneticisinin bağlantısını kesin.

Bu görev hakkında

Bir kuyruk yöneticisinin aracı sıradüzeniyle bağlantısını kesmek için **ALTER QMGR** komutunu kullanın. İstedığınız zaman bir kuyruk yöneticisinin bağlantısını kesebilirsiniz.

Kuyruk yöneticileri arasındaki bağlantı çalışırken üst ögeyi güncelleme isteği gönderilir.

Yordam

ALTER QMGR PARENT('')

Örnek

```
C:\Documents and Settings\Admin>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2.
  1 : alter qmgr parent('')
AMQ8005: WebSphere MQ queue manager changed.
  2 : display pubsub type(child)
AMQ8147: WebSphere MQ object not found.
display pubsub type(parent)
  3 : display pubsub type(parent)
AMQ8147: WebSphere MQ object not found.
```

Sonraki adım

Artık gerekli olmayan akışları, kuyukları ve el ile tanımlanan kanalları silebilirsiniz.

Yayınlama/abone olma sıradüzeni örneği: Senaryo 1

Kuyruk yöneticisi adı diğer adı olan noktadan noktaya iletişim kanallarını kullanarak bir yayınlama/abone olma sıradüzeni topolojisi ayarlayın.

Bu görev hakkında

Bu senaryolar, kuyruk yöneticileri arasındaki bağlantıyı kurmak için farklı şekillerde bir yayınlama/abone olma hiyerarşisi oluşturur. Bu senaryolar, QM1adlı bir üst kuyruk yöneticisini ve QM2adlı iki alt kuyruk yöneticisini ve QM3' yi kullanır.

1. senaryo, sürecin daha kolay izlenmesini sağlamak için daha küçük bölümlere ayrılır.

1. senaryo: 1. bölüm: Kuyruk yöneticilerinin yaratılması

Yordam

1. Aşağıdaki komutları kullanarak QM1, QM2ve QM3 adlı üç kuyruk yöneticisi yaratın ve başlatın:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
strmqm QM1

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
strmqm QM2

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
strmqm QM3
```

2. Tüm üç kuyruk yöneticilerindeki aşağıdaki komutu kullanarak kuyruk yöneticisi yayınlama/abone olma kipini etkinleştirin:

```
ALTER QMGR PSMODE(ENABLED)
```

Senaryo 1 bölüm 2: Noktadan noktaya kanal bağlantıları

Bu görev hakkında

Kuyruk yöneticisi diğer adını, üst kuyruk yöneticisiyle aynı adı taşıyan bir kuyruk yöneticisi diğer adını kullanarak, noktadan noktaya iletişim kanalı bağlantıları kurun.

Yordam

1. Define a transmission queue and queue manager alias on QM2 to QM1. Define a sender channel to QM1 and a receiver channel for the sender channel created on QM1 for QM2:

```
DEFINE QLOCAL(QM1.XMITQ) USAGE(XMITQ)

DEFINE QREMOTE(QM1) RNAME('') RQMNAME(QM1) XMITQ(QM1.XMITQ)

DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1.XMITQ)
```

```
TRPTYPE(TCP)
DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(RCVR) TRPTYPE(TCP)
```

2. Define a transmission queue and queue manager alias on QM3 to QM1. Define sender channel to QM1 and a receiver channel for the sender channel created on QM1 for QM3:

```
DEFINE QLOCAL(QM1.XMITQ) USAGE(XMITQ)
DEFINE QREMOTE (QM1) RNAME('') RQMNAME(QM1) XMITQ(QM1.XMITQ)
DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1.XMITQ)
TRPTYPE(TCP)
DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(RCVR) TRPTYPE(TCP)
```

3. Define a transmission queue and queue manager alias on QM1 to QM2 and QM3. Define sender channel to QM2 and QM3, and a receiver channel for the sender channels created on QM2 and QM3 for QM1:

```
DEFINE QLOCAL(QM2.XMITQ) USAGE(XMITQ)
DEFINE QREMOTE (QM2) RNAME('') RQMNAME(QM2) XMITQ(QM2.XMITQ)
DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(SDR) CONNAME('localhost(7777)') XMITQ(QM2.XMITQ)
TRPTYPE(TCP)
DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)
DEFINE QLOCAL(QM3.XMITQ) USAGE(XMITQ)
DEFINE QREMOTE (QM3) RNAME('') RQMNAME(QM3) XMITQ(QM3.XMITQ)
DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(SDR) CONNAME('localhost(8888)') XMITQ(QM3.XMITQ)
TRPTYPE(TCP)
DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)
```

4. Kuyruk yöneticilerindeki uygun dinleyicileri başlatın:

```
runmqclsr -m QM1 -t TCP -p 9999 &
runmqclsr -m QM2 -t TCP -p 7777 &
runmqclsr -m QM3 -t TCP -p 8888 &
```

5. Aşağıdaki kanalları başlatın:

- a. QM1üzerinde:

```
START CHANNEL('QM1.TO.QM2')
START CHANNEL('QM1.TO.QM3')
```

- b. QM2üzerinde:

```
START CHANNEL('QM2.TO.QM1')
```

- c. QM3üzerinde:

```
START CHANNEL('QM3.TO.QM1')
```

6. Tüm kanalların başlatıldığını doğrulayın:

```
DISPLAY CHSTATUS('QM1.TO.QM2')
DISPLAY CHSTATUS('QM1.TO.QM3')
DISPLAY CHSTATUS('QM2.TO.QM1')
DISPLAY CHSTATUS('QM3.TO.QM1')
```

Senaryo 1 bölüm 3: Bağlantı kuyruğu yöneticileri ve bir konu tanımlayın

Bu görev hakkında

Alt kuyruk yöneticilerini QM2 ve QM3 üst kuyruk yöneticisine bağlayın QM1.

Yordam

1. QM2 ve QM3 üzerinde, üst kuyruk yöneticisini QM1 olarak ayarlayın:

```
ALTER QMGR PARENT (QM1)
```

2. Alt kuyruk yöneticilerinin üst kuyruk yöneticisine bağlı olup olmadığını denetlemek için, tüm kuyruk yöneticilerindeki şu komutu çalıştırın:

```
DISPLAY PUBSUB TYPE(ALL)
```

3. Bir konu nesnesi tanımlayın:

```
define topic(FOOTBALL) TOPICSTR('Sport/Soccer')
```

1. senaryo bölüm 4: Konuyu yayınlayın ve abone olun

Bu görev hakkında

Konuyu yayınlamak ve abone olmak için amqsub . exe ve amqssub . exe uygulamalarını kullanın.

Yordam

1. Bu komutu ilk komut penceresinde çalıştırın:

```
amqsub Sport/Soccer QM2
```

2. Bu komutu ikinci komut penceresinde çalıştırın:

```
amqssub Sport/Soccer QM1
```

3. Bu komutu üçüncü komut penceresinde çalıştırın:

```
amqssub Sport/Soccer QM3
```

Sonuçlar

İkinci ve üçüncü komut pencerelerinde bulunan amqssub . exe uygulamaları, ilk komut penceresinde yayınlanan iletileri alır.

İlgili görevler

[“Yayınlama/abone olma sıradüzeni örneği: Senaryo 2” sayfa 83](#)

İletim kuyruğu adı uzak kuyruk yöneticisiyle aynı olan noktadan noktaya iletişim kanallarını kullanarak bir yayınlama/abone olma sıradüzeni topolojisi ayarlayın.

[“Yayınlama/abone olma sıradüzeni örneği: Senaryo 3” sayfa 86](#)

Bir küme kanalını kullanarak bir sıradüzen topolojisine kuyruk yöneticisi ekleyin.

Yayınlama/abone olma sıradüzeni örneği: Senaryo 2

İletim kuyruğu adı uzak kuyruk yöneticisiyle aynı olan noktadan noktaya iletişim kanallarını kullanarak bir yayınlama/abone olma sıradüzeni topolojisi ayarlayın.

Bu görev hakkında

Bu senaryolar, kuyruk yöneticileri arasındaki bağlantıyı kurmak için farklı şekillerde bir yayınlama/abone olma hiyerarşisi oluşturur. Bu senaryolar, QM1adlı bir üst kuyruk yöneticisini ve QM2adlı iki alt kuyruk yöneticisini ve QM3' yi kullanır.

2. senaryo, sürecin daha kolay izlenmesini sağlamak için daha küçük bölümlere ayrılır. This scenario reuses Scenario 1 part 1, Scenario 1 part 3, and Scenario 1 part 4 from [“Yayınlama/abone olma sıradüzeni örneği: Senaryo 1” sayfa 81.](#)

Senaryo 2 bölüm 1: Kuyruk yöneticisi yarat ve PSMODE ayarla

Yordam

1. Aşağıdaki komutları kullanarak QM1, QM2ve QM3 adlı üç kuyruk yöneticisi yaratın ve başlatın:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
strmqm QM1

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
strmqm QM2

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
strmqm QM3
```

2. Tüm üç kuyruk yöneticilerindeki aşağıdaki komutu kullanarak kuyruk yöneticisi yayınlama/abone olma kipini etkinleştirin:

```
ALTER QMGR PSMODE(ENABLED)
```

Senaryo 2 bölüm 2: Noktadan noktaya kanal bağlantıları

Bu görev hakkında

Üst kuyruk yöneticisiyle aynı adı taşıyan bir iletim kuyruğunu kullanarak, kuyruk yöneticisi arasında noktadan noktaya iletişim kanalı bağlantıları kurun.

Yordam

1. QM2 ile QM1arasında bir iletim kuyruğu tanımlayın. Define a sender channel to QM1 and a receiver channel for the sender channel for QM2 created on QM1:

```
DEFINE QLOCAL(QM1) USAGE(XMITQ)
DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1) TRPTYPE(TCP)
DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(RCVR) TRPTYPE(TCP)
```

2. QM3 ile QM1arasında bir iletim kuyruğu tanımlayın. Define sender channel to QM1 and a receiver channel for the sender channel created on QM1 for QM3:

```
DEFINE QLOCAL(QM1) USAGE(XMITQ)
DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1) TRPTYPE(TCP)
DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(RCVR) TRPTYPE(TCP)
```

3. Define transmission queues on QM1 to QM2 and QM3. Define sender channels to QM2 and QM3, and a receiver channel for the sender channels created on QM2 and QM3 for QM1:

```
DEFINE QLOCAL(QM2) USAGE(XMITQ)
DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(SDR) CONNAME('localhost(7777)') XMITQ(QM2) TRPTYPE(TCP)
DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)
DEFINE QLOCAL(QM3) USAGE(XMITQ)
```

```
DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(SDR) CONNAME('localhost(8888)') XMITQ(QM3) TRPTYPE(TCP)
DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)
```

4. Kuyruk yöneticilerindeki uygun dinleyicileri başlatın:

```
runmqclsr -m QM1 -t TCP -p 9999 &
runmqclsr -m QM2 -t TCP -p 7777 &
runmqclsr -m QM3 -t TCP -p 8888 &
```

5. Aşağıdaki kanalları başlatın:

a. QM1üzerinde:

```
START CHANNEL('QM1.TO.QM2')
START CHANNEL('QM1.TO.QM3')
```

b. QM2üzerinde:

```
START CHANNEL('QM2.TO.QM1')
```

c. QM3üzerinde:

```
START CHANNEL('QM3.TO.QM1')
```

6. Tüm kanalların başlatıldığını doğrulayın:

```
DISPLAY CHSTATUS('QM1.TO.QM2')
DISPLAY CHSTATUS('QM1.TO.QM3')
DISPLAY CHSTATUS('QM2.TO.QM1')
DISPLAY CHSTATUS('QM3.TO.QM1')
```

Senaryo 2 bölüm 3: Bağlantı kuyruğu yöneticileri ve bir konu tanımlayın

Bu görev hakkında

Alt kuyruk yöneticilerini QM2 ve QM3 üst kuyruk yöneticisine bağlayın QM1.

Yordam

1. QM2 ve QM3üzerinde, üst kuyruk yöneticisini QM1olarak ayarlayın:

```
ALTER QMGR PARENT (QM1)
```

2. Alt kuyruk yöneticilerinin üst kuyruk yöneticisine bağlı olup olmadığını denetlemek için, tüm kuyruk yöneticilerindeki şu komutu çalıştırın:

```
DISPLAY PUBSUB TYPE(ALL)
```

3. Bir konu nesnesi tanımlayın:

```
define topic(FOOTBALL) TOPICSTR('Sport/Soccer')
```

2. senaryo bölüm 4: Konuyu yayınlayın ve abone olun

Bu görev hakkında

Konuyu yayınlamak ve abone olmak için amqspub . exe ve amqssub . exe uygulamalarını kullanın.

Yordam

1. Bu komutu ilk komut penceresinde çalıştırın:

```
amqspub Sport/Soccer QM2
```

2. Bu komutu ikinci komut penceresinde çalıştırın:

```
amqssub Sport/Soccer QM1
```

3. Bu komutu üçüncü komut penceresinde çalıştırın:

```
amqssub Sport/Soccer QM3
```

Sonuçlar

İkinci ve üçüncü komut pencerelerinde bulunan amqssub . exe uygulamaları, ilk komut penceresinde yayınlanan iletileri alır.

İlgili görevler

“Yayınlama/abone olma sıradüzeni örneği: Senaryo 1” sayfa 81

Kuyruk yöneticisi adı diğer adı olan noktadan noktaya iletişim kanallarını kullanarak bir yayınlama/abone olma sıradüzeni topolojisi ayarlayın.

“Yayınlama/abone olma sıradüzeni örneği: Senaryo 3” sayfa 86

Bir küme kanalını kullanarak bir sıradüzen topolojisine kuyruk yöneticisi ekleyin.

Yayınlama/abone olma sıradüzeni örneği: Senaryo 3

Bir küme kanalını kullanarak bir sıradüzen topolojisine kuyruk yöneticisi ekleyin.

Bu görev hakkında

Bu senaryolar, kuyruk yöneticileri arasındaki bağlantıyı kurmak için farklı şekillerde bir yayınlama/abone olma hiyerarşisi oluşturur. Bu senaryolar, QM1adlı bir üst kuyruk yöneticisini ve QM2adlı iki alt kuyruk yöneticisini ve QM3' yi kullanır.

3. senaryo, sürecin daha kolay izlenmesini sağlamak için daha küçük bölümlere ayrılır. This scenario reuses Scenario 1 part 1, Scenario 1 part 3, and Scenario 1 part 4 from “Yayınlama/abone olma sıradüzeni örneği: Senaryo 1” sayfa 81.

Bu senaryo, DEM0 adlı bir küme yaratır; burada QM1 ve QM2 tam havuzlardır ve QM3 kısmi bir havuzdur. Queue manager QM1 is the parent of queue managers QM2 and QM3.

Senaryo 2 bölüm 1: Kuyruk yöneticisi yarat ve PSMODE ayarla

Yordam

1. Aşağıdaki komutları kullanarak QM1, QM2ve QM3 adlı üç kuyruk yöneticisi yaratın ve başlatın:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
strmqm QM1

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
strmqm QM2

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
strmqm QM3
```

2. Tüm üç kuyruk yöneticilerindeki aşağıdaki komutu kullanarak kuyruk yöneticisi yayınlama/abone olma kipini etkinleştirin:

```
ALTER QMGR PSMODE(ENABLED)
```

Bu görev hakkında

Kuyruk yöneticileri arasında bir küme için noktadan noktaya kanal bağlantıları kurun.

Yordam

1. QM1 ve QM2 üzerinde, **REPOS** parametresini DEMO kümesinin adına ayarlayın:

```
ALTER QMGR REPOS(DEMO)
```

2. Kuyruk yöneticilerindeki uygun dinleyicileri başlatın:

```
runmqclsr -m QM1 -t TCP -p 9999 &  
runmqclsr -m QM2 -t TCP -p 7777 &  
runmqclsr -m QM3 -t TCP -p 8888 &
```

3. Her kuyruk yöneticisi için küme alıcı kanalını tanımlayın:

- a. QM1 üzerinde:

```
DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(9999)')  
CLUSTER(DEMO)
```

- b. QM2 üzerinde:

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(7777)')  
CLUSTER(DEMO)
```

- c. QM3 üzerinde:

```
DEFINE CHANNEL(TO.QM3) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(8888)')  
CLUSTER(DEMO)
```

4. Küme içindeki her bir kuyruk yöneticisinde tam bir havuz için bir küme gönderen kanalı tanımlayın:

- a. QM1 üzerinde:

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(7777)')  
CLUSTER(DEMO)
```

- b. QM2 üzerinde:

```
DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(9999)')  
CLUSTER(DEMO)
```

- c. QM3 , QM1 ya da QM2 üzerinde tam havuza bir küme gönderen kanalı olabilir. This example defines the channel to QM1:

```
DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(9999)')  
CLUSTER(DEMO)
```

Bu görev hakkında

Alt kuyruk yöneticilerini QM2 ve QM3 üst kuyruk yöneticisine bağlayın QM1.

Yordam

1. QM2 ve QM3 üzerinde, üst kuyruk yöneticisini QM1 olarak ayarlayın:

```
ALTER QMGR PARENT (QM1)
```

2. Alt kuyruk yöneticilerinin üst kuyruk yöneticisine bağlı olup olmadığını denetlemek için, tüm kuyruk yöneticilerindeki şu komutu çalıştırın:

```
DISPLAY PUBSUB TYPE(ALL)
```

3. Bir konu nesnesi tanımlayın:

```
define topic(FOOTBALL) TOPICSTR('Sport/Soccer')
```

2. senaryo bölüm 4: Konuyu yayınlayın ve abone olun

Bu görev hakkında

Konuyu yayınlamak ve abone olmak için amqsub . exe ve amqssub . exe uygulamalarını kullanın.

Yordam

1. Bu komutu ilk komut penceresinde çalıştırın:

```
amqsub Sport/Soccer QM2
```

2. Bu komutu ikinci komut penceresinde çalıştırın:

```
amqssub Sport/Soccer QM1
```

3. Bu komutu üçüncü komut penceresinde çalıştırın:

```
amqssub Sport/Soccer QM3
```

Sonuçlar

İkinci ve üçüncü komut pencerelerinde bulunan amqssub . exe uygulamaları, ilk komut penceresinde yayınlanan iletileri alır.

İlgili görevler

“Yayınlama/abone olma sıradüzeni örneği: Senaryo 1” sayfa 81

Kuyruk yöneticisi adı diğer adı olan noktadan noktaya iletişim kanallarını kullanarak bir yayınlama/abone olma sıradüzeni topolojisi ayarlayın.

“Yayınlama/abone olma sıradüzeni örneği: Senaryo 2” sayfa 83

İletim kuyruğu adı uzak kuyruk yöneticisiyle aynı olan noktadan noktaya iletişim kanallarını kullanarak bir yayınlama/abone olma sıradüzeni topolojisi ayarlayın.

Yayınlamaların ve aboneliklerin akışını denetleme

Dağıtık bir yayınlama/abone olma topolojisine bir araya gelen kuyruk yöneticileri ortak bir birleşik konu alanını paylaşır. Her yayının ve aboneliğin yerel mi, yoksa genel mi olduğunu seçerek topoloji içinde yayın akışını ve abonelikleri denetleyebilirsiniz.

Yerel yayınlar ve abonelikler, yayınlayıcının ya da abonenin bağlı olduğu kuyruk yöneticisinin ötesine yayılmaz.

Kümeler ya da sıradüzenlerde kuyruk yöneticilerini birbirine bağlayarak yaratılan konu alanlarının kapsamını denetleyebilirsiniz. Bir yayınlama/abone olma kümesinde, konu nesnesi 'clustered' olmalıdır ya da tüm öğeler yerel olarak kalır ve yayın ya da abonelik hiçbir etkisine sahip değildir.

Bir abonelik, farklı yayınlardaki konu dizgileriyle eşleştiğinde, farklı konu nesnelere çözümleyebilirler. Bunlara çakışan konular denir. Belirli bir eşleşme için bir yayınlı ilişkili konu nesnesi, konu özniteliklerini sağlar ve abonenin yayını alacaksa, örneğin, örneğin, bu öznitelikleri belirler.

Yayın kapsamı

Bir yayının kapsamı, kuyruk yöneticilerinin bir yayını uzak kuyruk yöneticilerine iletip iletmeyeceğini denetler. Yayınların kapsamını yönetmek için **PUBSCOPE** konu öznitelikliğini kullanın.

Bir yayın uzak kuyruk yöneticilerine iletilmezse, yayını yalnızca yerel aboneler alır.

PUBSCOPE konu özniteliği, belirli bir konuda yapılan yayınların kapsamını belirlemek için kullanılır. Özniteliği aşağıdaki değerlerden birine ayarlayabilirsiniz:

MMGR

Yayın yalnızca yerel abonelere teslim edilir. Bu yayınlara *yerel yayınlar* adı verilir. Yerel yayınlar, uzak kuyruk yöneticilerine iletilmez ve bu nedenle uzak kuyruk yöneticilerine bağlı olan aboneler tarafından alınmaz.

TÜMÜ

Yayın, yerel abonelere ve uzak kuyruk yöneticilerine bağlı abonelere teslim edilir. Bu yayınlara *genel yayınlar* adı verilir.

ASPARENT

Üst öğenin **PUBSCOPE** ayarını kullanın.

Yayıncılar, MQPMO_SCOPE_QMGR put iletisi seçeneğini kullanarak bir yayının yerel mi, yoksa genel mi olduğunu da belirleyebilir. Bu seçenek kullanılırsa, **PUBSCOPE** konu özniteliği kullanılarak ayarlanan herhangi bir davranışı geçersiz kılar.

Abonelik kapsamı

Abonelik kapsamındaki bir aboneliğin kapsamı, bir kuyruk yöneticisinde aboneliğin olup olmadığını, yayınlama/abone olma kümesi ya da sıradüzeninde başka bir kuyruk yöneticisinde yayınlanan yayınların mı, yoksa yalnızca yerel yayıncılardaki yayınların yayınlanıp alınmayacağını denetler.

Abonelik kapsamını bir kuyruk yöneticisiyle sınırlandırma, yetkili sunucu aboneliklerini yayınlama/abone olma topolojisindeki diğer kuyruk yöneticilerine iletmekten durdurur. Bu, kuyruklar arası yönetici yayınlama/abone olma ileti alışverişi trafiğini azaltır.

SUBSCOPE konu özniteliği, belirli bir konuda yapılan aboneliklerin kapsamını belirlemek için kullanılır. Özniteliği aşağıdaki değerlerden birine ayarlayabilirsiniz:

MMGR

Bir abonelik yalnızca yerel yayınları alır ve yetkili sunucu abonelikleri uzak kuyruk yöneticilerine yayılmaz.

TÜMÜ

Yetkili abonelik, uzak kuyruk yöneticilerine yayılır ve abonenin yerel ve uzak yayınları alır.

ASPARENT

Üst öğenin **SUBSCOPE** ayarını kullanın.

Bireysel aboneler, abonelik oluştururken MQSO_SCOPE_QMGR abonelik seçeneğini belirterek Tüm ' un **SUBSCOPE** ayarını geçersiz kılabilir. Bir abonelik, bir konunun **SUBSCOPE** ayarını ALLolarak geçersiz kılabilir.

Not: Bireysel aboneler yalnızca konunun **SUBSCOPE** ' unu kısıtlayabilir. When an individual subscription has **SUBSCOPE** set to Tüm, the subscription honors the matching topics' **SUBSCOPE** setting.

Yayın ve abonelik kapsamlarının birleştirilmesi

WebSphere MQ sürüm 7, yayın ve abonelik kapsamı, kuyruk yöneticileri arasındaki yayınların akışını belirlemek için bağımsız olarak çalışır.

Yayınlara, yayınlama/abone olma topolojisine bağlı olan tüm kuyruk yöneticilerine ya da yalnızca yerel kuyruk yöneticisine (ya da yerel kuyruk yöneticisiyle) akıtılabilir. Yetkili abonelikler için de benzer şekilde. Bir abonelikte eşleşen yayınlar, bu iki akımın birleşimiyle yönetilir.

Yayınlara ve aboneliklerin her ikisi de QMGR ya da ALLolarak kapsamlanabilir. Bir yayınlama ve bir abonenin aynı kuyruk yöneticisine bağlı olması durumunda, kapsam ayarları, abonenin aldığı abonelerin yayınlara etkilemez.

Yayınlama ve abone farklı kuyruk yöneticilerine bağlıysa, uzak yayınları almak için her iki ayarın da ALL (TÜMÜ) olması gerekir.

Yayıncıların farklı kuyruk yöneticilerine bağlı olduğunu varsayın. Bir abonenin herhangi bir yayıncıdan yayınlarını almasını istiyorsanız, abonelik kapsamını ALLolarak ayarlayın. Daha sonra her yayıncı için, yayınlarının kapsamını yerel abonelere yayıncı yerel olarak yayınlayıp sınırlamayacağınıza karar verebilirsiniz.

Abonelerin farklı kuyruk yöneticilerine bağlı olduğunu varsayın. Yayınların tüm abonelere göndermesini istiyorsanız, yayın kapsamını ALLolarak ayarlayın. Bir abonenin yalnızca aynı kuyruk yöneticisine bağlı bir yayıncıdan yayınlarını almasını istiyorsanız, abonelik kapsamını QMGRolarak ayarlayın.

Sürüm 6 'da ve daha önceki sürümlerde, yayın ve abonelik kapsamı yalnızca bu yayınları yönetmez. Ayrıca, yayının kapsamı, abonelik kapsamının eşleştirmek zorunda kaldı.

Örnek: futbol sonuçları hizmeti

Diyelim ki futbol liginde bir üye takımı olduğunuzu varsayalım. Her ekibin bir yayınlama/abone olma kümesindeki diğer tüm ekiplere bağlı bir kuyruk yöneticisi vardır.

The teams publish the results of all the games played on their home ground using the topic, *Football/result/Home team name/Away team name*. İtalik dizgiler, değişken konu adlarıdır ve yayın, eşleşmenin sonucunun sonudur.

Each club also republishes the results just for the club using the topic string *Football/myteam/Home team name/Away team name*.

Her iki konu da tüm küme içinde yayınlanır.

Aşağıdaki abonelikler lige göre ayarlanmıştır, böylece herhangi bir ekibin taraftarları sonuçlara üç ilginç şekilde abone olabilirler.

Notice that you can set up cluster topics with SUBSCOPE (QMGR). Konu tanımlamaları, kümenin her bir üyeye dağıtılır, ancak aboneliğin kapsamı yalnızca yerel kuyruk yöneticisidir. Bu nedenle, her kuyruk yöneticisinde aboneler aynı abonelikten farklı yayınlar alırlar.

Tüm sonuçları al

```
DEFINE TOPIC(A) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(ALL)
```

Tüm ana sonuçları al

```
DEFINE TOPIC(B) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(QMGR)
```

Aboneliğin QMGR kapsamı olduğundan, yalnızca ana zeminde yayınlanan sonuçlar eşleştirilir.

Tüm ekip sonuçlarını al

```
DEFINE TOPIC(C) TOPICSTR('Football/myteam/') CLUSTER SUBSCOPE(QMGR)
```

Aboneliğin QMGR kapsamına sahip olduğu için, yerel olarak yeniden yayınlanan yerel ekip sonuçları eşleştirilir.

Konu alanları

Konu alanı, abone olabileceğiniz konular kümesidir. Dağıtılmış yayınlama/abone olma topolojisindeki bir kuyruk yöneticisine bağlı bir abonenin, bağlı kuyruk yöneticilerinde tanımlı konuları içeren bir konu alanına sahip olması gerekir.

Konular başlangıçta, bir konu nesnesini ya da kalıcı aboneliği tanımladığınızda ya da bir uygulama dinamik olarak bir yayın ya da abonelik yarattığında dinamik olarak yaratılır.

Konular diğer kuyruk yöneticilerine hem yetkili sunucu abonelikleri yoluyla, hem de denetim kümesi konu nesnelere yaratılarak yayılır. Yetkili sunucu abonelikleri, yayıncının bağlı olduğu kuyruk yöneticisinden, abonelerin kuyruk yöneticilerine iletilmekte olan yayınlarla sonuçlanır. Yetkili sunucu abonelikleri, farklı kuyruk yöneticilerinde tanımlanan konuların ortak bir konu alanı içinde birleştirilerek mekanizmasıdır.

Yetkili sunucu abonelikleri, bir kuyruk yöneticisi sıradüzenindeki üst-alt öge ilişkileri tarafından birbirine bağlanan tüm kuyruk yöneticileri arasında geçirilir. Sonuç olarak, bir kuyruk yöneticisinde, sıradüzenindeki

başka bir kuyruk yöneticisinde tanımlanan bir konuya abone olabilirsiniz. Kuyruk yöneticileri arasında bağlı bir yol olduğu sürece, kuyruk yöneticilerinin nasıl bağlandıklarının bir önemi yoktur.

Yetkili sunucu abonelikleri, bir kümenin tüm üyeleri arasında *küme* konuları için de yayılır. Küme konusu, **CLUSTER** özneliğine sahip bir konu nesnesine eklenen ya da üst ögesinden özneliği devralan bir konudur. Küme konuları olmayan konular yerel konular olarak bilinir ve kümede eşlenmez. Bir yetkili abonelik aboneliği, yerel konulara aboneliklerden kümeye geçirilir.

Özetlemek için, iki durumda aboneler için yetkili sunucu abonelikleri oluşturulur.

1. Kuyruk yöneticisi, bir sıradüzeninin üyesidir ve bir yetkili sunucu aboneliği kuyruk yöneticisinin üst ve alt öğelerinden iletilir.
2. Kuyruk yöneticisi bir kümenin üyesidir ve abonelik konu dizgisi, bir küme konusu nesnesiyle ilişkili bir konuya çözülüyor. Yetkili sunucu abonelikleri, kümenin tüm üyelerine iletilir. Karmaşıklıklar hakkında daha fazla bilgi için bkz. "[Çakışan konular](#)" sayfa 99 .

Bir kuyruk yöneticisi, bir kümenin ve sıradüzeninin üyesiye, yetkili abonelikler her iki mekanizma tarafından da aboneye yinelenen yayınlar sağlanmadan yayılır.

Bir küme konusu nesnesi yaratımın etkisi yaşıdır. Bir abonelik bir küme konusuna çözüldüğünde, bir konuya ilişkin yetkili sunucu abonelikleri kümenin diğer üyelerine gönderilir. Ayrıca, konu nesnesinin bir kopyasını kümenin diğer üyelerine de gönderir. Küme konusu nesnelere iletilmesi, konuların denetlenmesini basitleştirmekten etkilenir. Tipik olarak, küme konusu nesnelere küme konusu anasistem adı verilen kümedeki tek bir kuyruk yöneticisinde tanımlanır.

Üç yayınlama/abone olma topolojisine ilişkin konu alanları aşağıdaki listede açıklanmıştır:

- "[Dava 1. Kümeleri yayınla/abone ol](#)" sayfa 91.
- "[Dava 2. Sürüm 7 'de sıradüzenleri yayınla/abone ol](#)" sayfa 92.
- "[Dava 3. Sürüm 6 'daki sıradüzenleri ve akışları yayınlama/abone olma](#)" sayfa 93.

Ayrı konularda, aşağıdaki görevler konu alanlarının nasıl birleştirileceğini açıklamalı.

- "[Yayınlama/abone olma kümesinde tek bir konu alanı yarat](#)" sayfa 93.
- "[Var olan sürüm 6 konu alanlarına bir sürüm 7 kuyruk yöneticisi ekler](#)" sayfa 94.
- "[Birden çok kümenin konu alanlarını birleştir](#)" sayfa 95.
- "[Birden çok kümede konu alanlarını birleştirin ve yalıtın](#)" sayfa 96
- "[Birden çok kümede konu boşluklara abone olma ve abone olma](#)" sayfa 98

Dava 1. Kümeleri yayınla/abone ol

Örnekte, kuyruk yöneticisinin bir yayınlama/abone olma sıradüzenineconnectedbağlı olmadığını varsayın.

Bir kuyruk yöneticisi yayınlama/abone olma kümesinin üyesiye, konu alanı yerel konulardan ve küme konularından oluşur. Yerel konular, **CLUSTER** özneliği olmayan konu nesnelere ilişkilendirilir. Bir kuyruk yöneticisinin yerel konu nesnesi tanımlamaları varsa, bu alanın konu alanı, yerel olarak tanımlanmış bir konu nesnelere de sahip olan kümedeki başka bir kuyruk yöneticisinden farklıdır.

Bir yayınlama/abone olma kümesinde, abone olduğunuz konu bir küme konusu nesnesine çözümleniyorsa, başka bir kuyruk yöneticisinde tanımlanan bir konuya abone olamazsınız.

Kümenin başka bir yerinde tanımlanan bir küme konularının çakışan tanımlamaları, en son tanım lehine çözümlenir. Herhangi bir zamanda, bir küme konularının tanımlanmış olması durumunda, farklı kuyruk yöneticilerindeki küme konusu tanımlaması farklı olabilir.

Tanımın bir küme konusu ya da yerel konu için olup olmadığı, bir konu nesnesinin yerel tanımlaması, kümenin başka bir yerinde tanımlanan aynı konu nesnesinden önceliklidir. Yerel olarak tanımlanmış konu, başka bir yerde tanımlanan nesne daha yeni olsa bile kullanılır.

PUBSCOPE ve **SUBSCOPE** seçeneklerinden birini QMGRolarak ayarlayın; bir yayının ya da bir küme konularındaki aboneliğin, kümedeki farklı kuyruk yöneticilerine akmasını önleyin.

Küme konu anasistemadaki USA/Alabama konu dizgisiyle Alabama bir küme konu nesnesi tanımladığınızı varsayalım. Sonuç aşağıdaki gibidir:

1. The topic space at the cluster topic host now includes the cluster topic object Alabama and the topic USA/Alabama.
2. The cluster topic object Alabama is replicated to all queue managers in the cluster where it is combined with the topic space at each queue manager. What happens at each queue manager in the cluster depends on whether the topic object Alabama exists at a queue manager.
 - If Alabama is a new topic object, the queue manager adds the cluster topic object Alabama, and the topic USA/Alabama, to its topic space.
 - If Alabama is a local definition, the cluster topic object Alabama is added. Yerel tanımlama silinmedikçe, uzaktan tanımlanan küme konusu nesnesi yoksayılr. Kuyruk yöneticisi her iki tanımlama da korur.
 - Alabama başka bir yerde tanımlanmış daha eski bir küme konusu nesnesiyse, yeni küme konu nesnesiyle değiştirilir.
3. Bir uygulama ya da yönetici, kümenin herhangi bir yerinde, Alabama konu nesnesine başvurarak USA/Alabama ' e abonelik oluşturabilir.
4. An application, anywhere in the cluster, using the topic string USA/Alabama directly can create a subscription that inherits the attributes of the topic object Alabama. Alabama konu nesnesi, USA/Alabama ile başlayan herhangi bir konu dizgisinden oluşturulmuş bir abonelik edinilir.

Diğer kuyruk yöneticilerinden birindeki Alabama konu nesnesinin başka bir tanımlaması varsa, küme konu anasistemdeki tanımdan daha öncelikli olur. Yerel nesnenin bir küme özniteliği olabilir ya da bu öznitelik olmayabilir. Küme özniteliği aynı kümeye ya da başka bir kümeye başvurabilir. Bu birden çok tanımlama vakasını önlemekten kaçının. Davranış farklılıklarına yol açarlar.

5. If the topic object Alabama has the **PUBSCOPE** attribute Tümü, subscriptions that resolve to Alabama are sent to all the other queue managers in the cluster.

Yayıncılardan farklı kuyruk yöneticilerine bağlı yayınlardan abonelere akan yayınları önlemek için, Alabama **PUBSCOPE** özniteliğini QMGR olarak ayarlayın.

Alabama konu nesnesi kümedeki her kuyruk yöneticisine eşlenir; bu nedenle **PUBSCOPE** ve **PUBSCOPE** öznitelikleri kümedeki tüm kuyruk yöneticilerine uygulanır.

Küme konusu nesnesinin, kümedeki her yerde aynı konu dizgisiyle ilişkilendirilmesi önemlidir. Bir konu nesnesinin ilişkilendirildiği konu dizisini değiştiremezsiniz. Aynı konu nesnesini farklı bir konu dizgisiyle ilişkilendirmek için, konu nesnesini silmeniz ve yeni konu dizisiyle yeniden yaratmanız gerekir. Konu kümelenmişse, etki, kümenin diğer üyelerinde saklanan konu nesnesinin kopyalarını silmek ve daha sonra, yeni konu nesnesinin kopyalarının küme içinde her yerde yaratılabilmesinden etkilenir. Konu nesnesinin kopyalarının tümü aynı konu dizgisine gönderme yapıyor.

Ancak, kümede başka bir kuyruk yöneticisinde, farklı bir konu dizgisiyle bir konu nesnesinin yinelenen bir tanımlamasını yaratabilirsiniz. Her zaman, bir kuyruk yöneticisindeki küme konusu anasistemleri yöneterek yinelenmeleri önlemekten kaçının. Bu önemli noktaya ilişkin daha fazla bilgi için bkz. "[Birden çok küme konu tanımlaması](#)" sayfa 69 . Aynı konu nesnesinin farklı konu dizgilerine sahip birden çok tanımlama, konuya ve konuya gönderme yapılan yere bağlı olarak farklı sonuçlar üretebilir.

Dava 2. Sürüm 7 'de sıradüzenleri yayınlama/abone ol

Örnekte, kuyruk yöneticisinin bir yayınlama/abone olma kümesinin üyesi *değil* olduğunu varsayın.

Sürüm 7 'de, bir kuyruk yöneticisi yayınlama/abone olma sıradüzeninin bir üyesiyse, konu alanı yerel olarak ve bağlı kuyruk yöneticilerindeki tüm konulardan oluşur. Bir sıradüzendeki tüm kuyruk yöneticilerinin konu alanı aynı. Yerel konular ve küme konularına ilişkin bir bölüm bölünmesi yoktur.

Bir yayıncıdan, sıradüzendeki farklı kuyruk yöneticilerine bağlı bir aboneye akan bir konuda bir yayının yayınlanmasını önlemek için, **PUBSCOPE** ve **SUBSCOPE** seçeneklerinden birini QMGR olarak ayarlayın.

Kuyruk yöneticisi QMA' daki USA/Alabama konu dizgisiyle Alabama bir konu nesnesi tanımladığınızı varsayalım. Sonuç aşağıdaki gibidir:

1. Artık QMA konumundaki konu alanı Alabama konu nesnesini ve USA/Alabama konu dizgisini içerir.
2. An application or administrator can create a subscription at QMA using the topic object name Alabama.
3. Bir uygulama, sıradüzendeki herhangi bir kuyruk yöneticisinde USA/Alabama dahil olmak üzere herhangi bir konuya abonelik oluşturabilir. If QMA has not been defined locally, the topic USA/Alabama resolves to the topic object SYSTEM.BASE.TOPIC.

Dava 3. Sürüm 6 'daki sıradüzenleri ve akışları yayınlama/abone olma

Sürüm 7 'den önce, konu alanı, tüm kuyruk yöneticilerindeki varsayılan akışın yer aldığı ayrı akışlar olarak bölündü. Yayınlar, farklı akışlar arasında akışa geçemez. Adlandırılmış akışlar kullanılırsa, farklı kuyruk yöneticilerindeki konu alanları farklı olabilir. Konular, varsayılan akıştaki konulara ve farklı adlandırılmış akışlardaki konulara bölünmektedir.

Not: Her adlandırılmış akış ayrı bir konu alanı oluşturur. Bağlı bir topoloji oluşturmak için bağlı kuyruk yöneticilerindeki her bir adlandırılan akış var olmalıdır. X akımı X, QMA ve QMC' de tanımlıdır, ancak QMB üzerinde tanımlanmaz. QMA , QMB'nin üst ögesi ise ve QMB , QMC' nin üst ögesi ise, XX akışındaki hiçbir konu QMA ile QMC arasında akamaz.

Hem **PUBSCOPE** hem de **SUBSCOPE** seçeneklerinin QMGR ya da ALL olarak ayarlanması, yalnızca yerel tüketime ilişkin yayınların değiş tokuş edilmesi ya da yalnızca genel kullanım için yayınların değiş tokuş edilmesi için bir yayıncı ve abone olmasını gerektirir.

Sürüm 7 'den, yayınlama/abone olma API 'si kullanılarak akışlar kullanılamaz. Sürüm 7 kuyruk yöneticisine kuyruklanmış yayınlama/abone olma olanağını kullanırsanız, akışlar akışların etkisinin benzetimini yapan farklı konu nesnelere eşlenir. Akıştaki tüm konular için kök konu olan bir konu nesnesi yaratılarak bir akış benzetimi oluşturur. Kuyruk yöneticisi, akışın ve her bir ağacın ilgili kök konusu arasındaki yayınları ve abonelikleri eşliyor.

Konu alanlarının birleştirilmesi

Bir kuyruk yöneticisinin konu alanını, yayınlama/abone olma kümesindeki diğer kuyruk yöneticileriyle ya da sıradüzendeki diğer kuyruk yöneticileriyle birleştirin. Yayınlama/abone olma kümelerini birleştirin ve sıradüzenleri içeren kümeleri yayınlayın/abone olun.

CLUSTER, PUBSCOPE ve **SUBSCOPE** özniteliklerinin oluşturma öbeklerini, yayınlama/abone olma kümelerini ve yayınlama/abone olma sıradüzenlerini kullanarak farklı yayınlama/abone olma konuları alanları oluşturabilirsiniz.

Tek bir kuyruk yöneticisinden bir yayınlama/abone olma kümesine ölçekleme örneğinden başlayarak, aşağıdaki senaryolar farklı yayınlama/abone olma topolojilerini gösterir.

Yayınlama/abone olma kümesinde tek bir konu alanı yarat

Birden çok kuyruk yöneticisi üzerinde çalışmak için yayınlama/abone olma sistemini ölçeklendirin. Tek bir özdeş konu alanı içeren her yayıncı ve abone sağlamak için bir yayınlama/abone olma kümesi kullanın.

Başlamadan önce

Tek bir sürüm 7 kuyruk yöneticisi üzerinde bir yayınlama/abone olma sistemi uyguladınız.

SYSTEM.BASE.TOPIC özniteliklerini devralabilmeye güvenmek yerine, her zaman kendi kök konularıyla konu alanları yaratın. Yayınlama/abone olma sisteminizi bir kümeye yükseltiyorsanız, kök başlıklarınızı küme konuları olarak tanımlayabilir, küme konusu ana makinesinde ve daha sonra tüm konularınız küme boyunca paylaşılır.

Bu görev hakkında

Artık sistemi daha fazla yayıncı ve aboneyi destekleyecek şekilde ölçeklemek ve her konunun küme boyunca görünmesini istiyorsanız, sistemi ölçeklendirin.

Yordam

1. Yayınlama/abone olma sistemiyle kullanmak için bir küme yaratın.

Performans nedenlerinden dolayı, var olan geleneksel bir kümeniz varsa, yeni yayınlama abone olma sistemi için yeni bir küme kurmanın daha iyi olması gerekir. Her iki kümenin küme havuzları için de aynı sunucuları kullanabilirsiniz

2. Küme konusu anasistem olmak üzere, havuzlardan biri olan bir kuyruk yöneticisi seçin.
3. Yayınlama/abone olma kümesi boyunca görünür olacak her konunun bir denetim konusu nesnesine çözülmesini sağlayın.
Yayınlama/abone olma kümesini adlandırma **CLUSTER** özneteliğini ayarlayın.

Sonraki adım

Yayınlayıcıyı ve abone uygulamalarını kümedeki kuyruk yöneticilerine bağlayın.

CLUSTER özneteliğine sahip yönetim konusu nesnelere oluşturun. Konular küme boyunca da yayılır. Yayınlayıcı ve abone programları, yönetimle ilgili konuları, davranışlarının kümedeki farklı kuyruk yöneticilerine bağlanarak değiştirilmemesi için kullanır.

Her kuyruk yöneticisinde bir küme konusu gibi işlem yapmak için SYSTEM.BASE.TOPIC ' e gerek duyarsanız, her kuyruk yöneticisinde bunu değiştirmeniz gerekir.

Var olan sürüm 6 konu alanlarına bir sürüm 7 kuyruk yöneticisi ekler

Var olan bir sürüm 6 yayınlama/abone olma sistemini, bir sürüm 7 kuyruk yöneticisiyle birlikte çalışmak üzere genişletir ve aynı konu alanlarını paylaşır.

Başlamadan önce

Var olan bir sürüm 6 yayınlama/abone olma sisteminiz var.

WebSphere MQ sürüm 7 'i yeni bir sunucuya kurdun ve bir kuyruk yöneticisi yapılandırdınız.

Bu görev hakkında

Sürüm 7 kuyruk yöneticileriyle çalışmak için var olan sürüm 6 yayınlama/abone olma sisteminizi genişletmek istiyorsunuz.

Kuyruğa alınan yayınlama/abone olma arabirimini kullanan sürüm 6 yayınlama/abone olma sisteminin geliştirilmesini kararlaştıran bir karar verdiniz. Sürüm 7 MQI olanağını kullanarak sisteme uzantı eklemeyi amaçladınız. Kuyruğa alınan yayınlama/abone olma uygulamalarını yeniden yazmak için artık bir planınız yok.

Sürüm 6 kuyruk yöneticilerini ileride sürüm 7 'ye yükseltmeyi amaçlıyorsunuz. Şu an için, sürüm 7 kuyruk yöneticilerindeki var olan kuyruğa alınmış yayınlama/abone olma uygulamalarını çalıştırmaya devam ediyor olabilirsiniz.

Yordam

1. Sürüm 7 kuyruk yöneticisini her iki yöndeki sürüm 6 kuyruk yöneticilerinden biriyle bağlamak için bir gönderen alıcılı kanal kümesi yaratın.
2. Hedef kuyruk yöneticilerinin adlarıyla iki iletim kuyruğu yaratın. Bir nedenden dolayı hedef kuyruk yöneticisinin adını iletim kuyruğu adı olarak kullanamazsanız, kuyruk yöneticisi diğer adlarını kullanın.
3. İletim kuyruklarını, gönderen kanallarını tetikleyecek şekilde yapılandırın.
4. Sürüm 6 yayınlama/abone olma sistemi akışları kullanıyorsa, akımları sürüm 7 kuyruk yöneticisine Akış eklenmesi içinde açıklandığı biçimde ekleyin.
5. 7. sürüm kuyruk yöneticisi **PSMODE** ' nin ETKIolarak ayarlananını denetleyin.
6. **PARENT** özneteliğini, sürüm 6 kuyruk yöneticilerinden birine gönderme yapmak için değiştirin.
7. Her iki yönde de etkin olan kuyruk yöneticileri arasındaki üst-alt ilişkisinin durumunu denetleyin.

Sonraki adım

Görevi tamamladıktan sonra, hem sürüm 6, hem de sürüm 7 kuyruk yöneticisi aynı konu alanlarını paylaşır. Örneğin, aşağıdaki tüm görevleri yapabilirsiniz.

- Sürüm 6 ve sürüm 7 kuyruk yöneticileri arasındaki yayınları ve abonelikleri değiştirin.
- Sürüm 7 kuyruk yöneticisinde varolan sürüm 6 yayınlama/abone olma programlarınızı çalıştırın.
- Sürüm 6 ya da sürüm 7 kuyruk yöneticisinde konu alanını görüntüleyin ve değiştirin.
- Sürüm 7 yayınlama/abone olma uygulamaları yazın ve bunları sürüm 7 kuyruk yöneticisi üzerinde çalıştırın.
- Sürüm 7 uygulamalarıyla yeni yayınlar ve abonelikler oluşturun ve bunları sürüm 6 uygulamalarıyla değiştirin.

Birden çok kümenin konu alanlarını birleştir

Birden çok kümeyi kapsayan konu alanları yaratın. Bir kümede bir konuya yayınlayın ve başka bir kümede bu konuya abone olun.

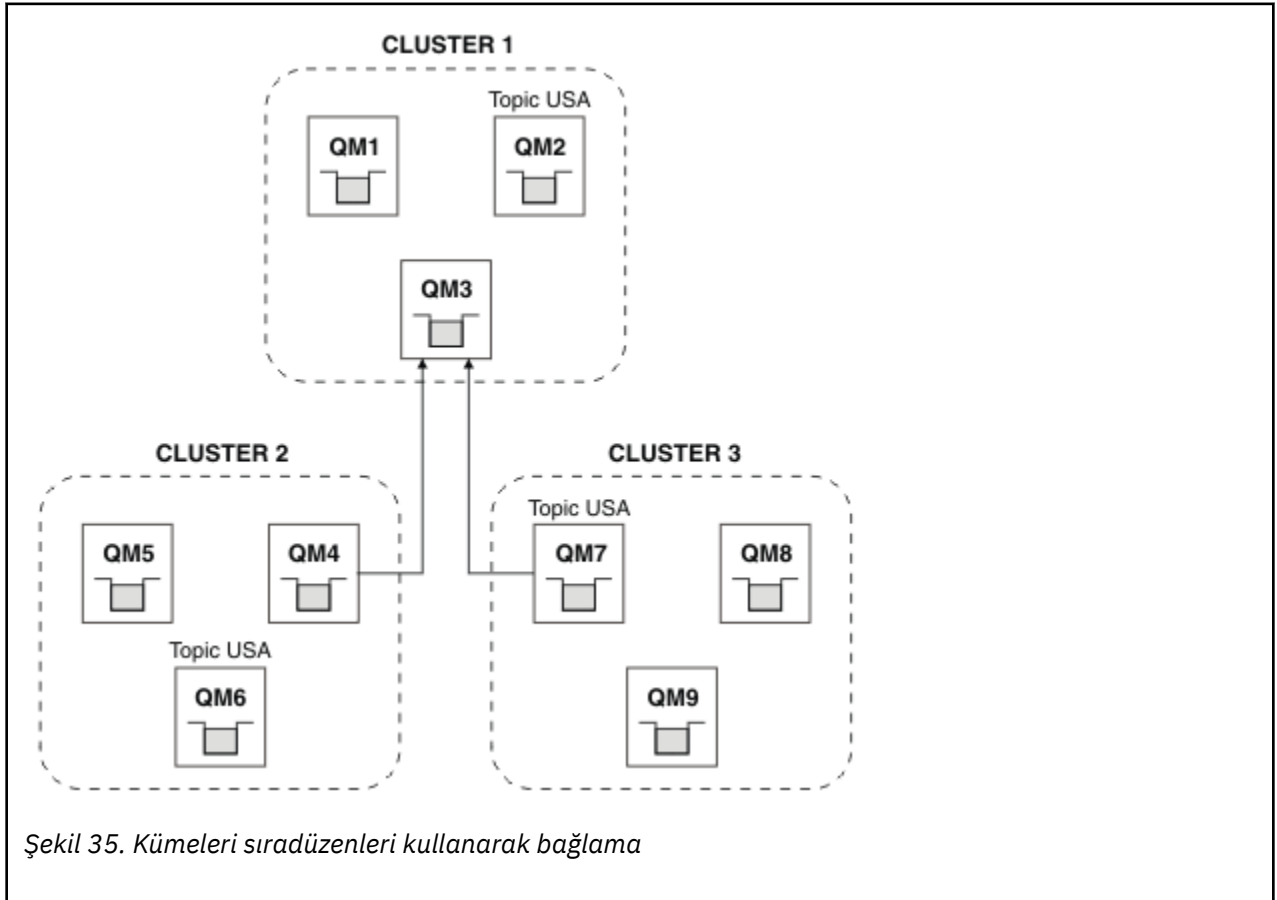
Başlamadan önce

Var olan yayınlama/abone olma kümeleriniz var ve bazı küme konularını tüm kümelere yaymak istiyorsunuz.

Bu görev hakkında

Yayınları bir kümeden başka bir kümeye yaymak için, kümeleri bir sıradüzenle birlikte birleştirmeniz gerekir; bkz. [Şekil 35 sayfa 95](#). Sıradüzensel bağlantılar, bağlı kuyruk yöneticileri arasındaki abonelikleri ve yayınları yayır ve kümeler küme konularını her küme içinde yayır, ancak kümeler arasında yayılmaz.

Bu iki mekanizmanın birleşimi, küme konularını tüm kümeler arasında yayılır. Küme konusu tanımlamalarını her kümede yinelemeniz gerekir.



Şekil 35. Kümeleri sıradüzenleri kullanarak bağlama

Aşağıdaki adımlar kümeleri bir sıradüzene bağlar.

Yordam

1. QM3 ve QM4 ve QM3 ve QM7 bağlantısını her iki yönde de bağlamak için iki adet gönderici alıcı kanalı kümesi oluşturun. Bir sıradüzeni bağlamak için, bir küme yerine geleneksel gönderen alıcı kanallarını ve iletim kuyruklarını kullanmanız gerekir.
2. Hedef kuyruk yöneticilerinin adlarıyla birlikte üç iletim kuyruğu yaratın. Bir nedenden dolayı hedef kuyruk yöneticisinin adını iletim kuyruğu adı olarak kullanamazsanız, kuyruk yöneticisi diğer adlarını kullanın.
3. İletim kuyruklarını, gönderen kanallarını tetikleyecek şekilde yapılandırın.
4. QM3, QM4 ve QM7 'in **PSMODE** ' i ETKI olarak ayarlann.
5. QM4 ve QM7 ' nin **PARENT** özniteliğini QM3 olarak değiştirin.
6. Her iki yönde de etkin olan kuyruk yöneticileri arasındaki üst-alt ilişkisinin durumunu denetleyin.
7. Create the administrative topic USA with the attribute **CLUSTER**(' KÜME 1 '), **CLUSTER**(' KÜME 2 '), and **CLUSTER**(' KÜME 3 ') on each of the three cluster topic hosts in clusters 1, 2 and 3. Küme konusu anasisteminin sıradüzensel olarak bağlı bir kuyruk yöneticisi olması gerekmez.

Sonraki adım

Artık [Şekil 35 sayfa 95](#) içindeki USA küme konusunu yayınlayabilir ya da bu kümeye abone olabilirsiniz. Yayın abonelikleri yayıncılara ve abonelere her üç kümeden de akış sağlar.

Suppose that you did not create USA as a cluster topic in the other clusters. If USA is only defined on QM7, then publications and subscriptions to USA are exchanged between QM7, QM8, QM9, and QM3. Publishers and subscribers running on QM7, QM8, QM9 inherit the attributes of the administrative topic USA. QM3 üzerindeki yayıncılar ve aboneler, QM3 üzerindeki SYSTEM . BASE . TOPIC özniteliklerini devralır.

Birden çok kümede konu alanlarını birleştirin ve yalıtın

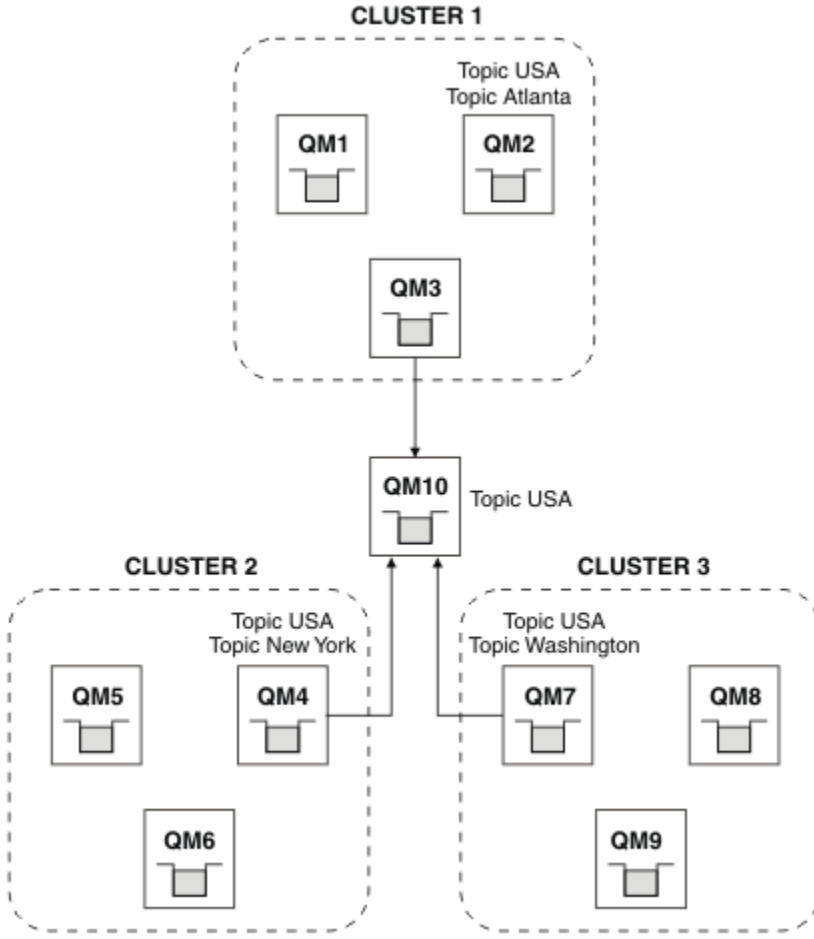
Bazı konu alanlarını belirli bir kümede yalıtın ve bunları bağlı tüm kümelerde erişilebilir kılmak için diğer konu alanlarını birleştirin.

Başlamadan önce

“Birden çok kümenin konu alanlarını birleştir” sayfa 95 konusunu inceleyin. Köprü olarak ek kuyruk yöneticisi eklemeksizin, gereksinimleriniz için yeterli olabilir.

Bu görev hakkında

“Birden çok kümenin konu alanlarını birleştir” sayfa 95 içinde [Şekil 35 sayfa 95](#) içinde gösterilen topolojide olası bir gelişme, tüm kümelerde paylaşılmayan küme konularını yalıtılmamaktadır. Kümelerin hiçbirinde olmayan bir köprü kuyruk yöneticisi yaratarak kümeleri yalıtın; bkz. [Şekil 36 sayfa 97](#). Hangi yayınlar ve aboneliklerin bir kümeden diğerine akabileceğini süzmek için köprü oluşturma kuyruk yöneticisini kullanın.



Şekil 36. Köprülü kümeler

Diğer kümelerde köprü'nün üzerinde gösterilmesini istemediğiniz küme konularını yalıtılmak için köprüyü kullanın. Şekil 36 sayfa 97'inde, USA tüm kümelerde paylaşılan bir küme konudur ve Atlanta, New York ve Washington, yalnızca tek bir kümede paylaşılan küme başlıklarına sahip olur.

Aşağıdaki yordamı kullanarak yapılandırmanızı modelleyin:

Yordam

1. Tüm kuyruk yöneticilerine **SUBSCOPE(QMGR)** ve **PUBSCOPE(QMGR)** sahip olacak şekilde tüm SYSTEM.BASE.TOPIC konu nesnelerini değiştirin.
Küme konularınızın kök konusu üzerinde belirttik olarak **SUBSCOPE(ALL)** ve **PUBSCOPE(ALL)** ayarlamadıkça, hiçbir konu (hatta küme konuları bile) diğer kuyruk yöneticilerine yayılır.
2. Define the topics on the three cluster topic hosts that you want to be shared in each cluster with the attributes **CLUSTER(küme adı)**, **SUBSCOPE(Tüm)** and **PUBSCOPE(Tüm)**.
Tüm kümeler arasında paylaşılan bazı küme konuları olmasını istiyorsanız, her bir küme için aynı konuyu tanımlayın. Küme özniteliği olarak her bir kümenin küme adını kullanın.
3. Tüm kümeler arasında paylaşılan olarak istediğiniz küme konuları için, köprü kuyruk yöneticisiyle (QM10), **SUBSCOPE(ALL)** ve **PUBSCOPE(ALL)** öznitelikleriyle konuları yeniden tanımlayın.

Örnek

Şekil 36 sayfa 97'ta örnekte, yalnızca USA 'den devralan konular, tüm üç küme arasında yayılırdır.

Sonraki adım

Subscriptions for topics defined on the bridge queue manager with **SUBSCOPE(Tüm)** and **PUBSCOPE(Tüm)** are propagated between the clusters.

Subscriptions for topics defined within each cluster with attributes **CLUSTER(küme adı)**, **SUBSCOPE(Tüm)** and **PUBSCOPE(Tüm)** are propagated within each cluster.

Diğer abonelikler yerel bir kuyruk yöneticisindir.

Birden çok kümede konu boşluklara abone olma ve abone olma

Örtüşme kümeleri kullanarak birden çok kümeden konulara abone olma ve bu konulara abone olma. Bu tekniği, kümelerdeki konu alanları örtüşmediği sürece kullanabilirsiniz.

Başlamadan önce

Kümeler arasındaki kesişimde bazı kuyruk yöneticileriyle birden çok geleneksel küme yaratın.

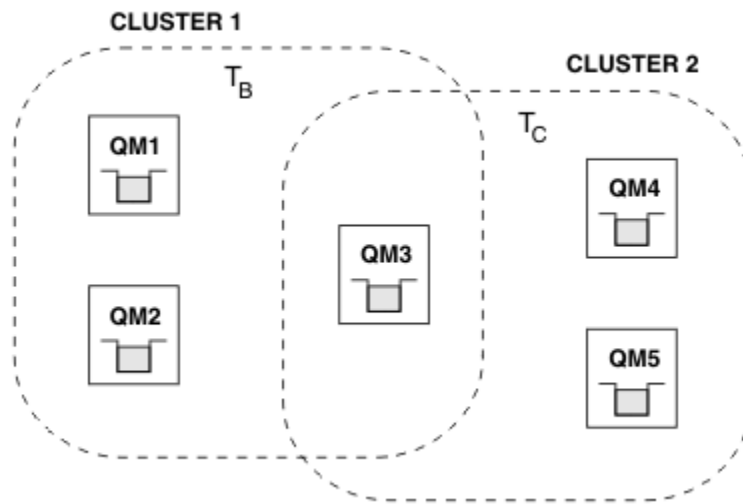
Bu görev hakkında

Farklı nedenlerden dolayı kümeleri örtüşmeyi seçmiş olabilirsiniz.

1. Sınırlı sayıda yüksek kullanılabilirlik sunucunuz ya da kuyruk yöneticisi var. Tüm küme havuzlarını ve küme konusu anasistemleri konuşlandırmanıza karar verirsiniz.
2. Ağ geçidi kuyruk yöneticilerini kullanarak bağlı var olan geleneksel kuyruk yöneticisi kümeniniz var. Yayınlama/abone olma uygulamalarını aynı küme topolojisine konuşlandırmak istiyorsunuz.
3. Birden çok kendi kendine sahip yayınlama/abone olma uygulamaları var. Performans nedenlerinden dolayı, yayınlama/abone olma kümelerini küçük ve geleneksel kümelerden ayrı tutmak daha iyi olur. Uygulamaları farklı kümelere yerleştirmeye karar verdiniz. Ancak, izleme uygulamasının yalnızca bir kopyasını lisansladığınızda, tek bir kuyruk yöneticisine ilişkin tüm yayınlama/abone olma uygulamalarını da izlemek isteyebilirsiniz. Bu kuyruk yöneticisinin, tüm kümelerdeki küme konularına ilişkin yayınlara erişimi olması gerekir.

Konularınızın örtüşmeyen konu alanlarında tanımlanmasını sağlayarak, konuları çakışan yayınlama/abone olma kümelerinde konuşlandırabilirsiniz, bkz. [Şekil 37 sayfa 98](#). Konu alanları örtüşerse, çakışan kümelere konuşlandırma sorunlara yol açar.

Yayınlama/abone olma kümeleri çakıştığından, çakışan kuyruk yöneticilerini kullanarak herhangi bir konu alanını yayınlatabilir ve herhangi bir konu alanını abone edebilirsiniz.



Şekil 37. Çakışan kümeler, çakışmayan konu alanları

Yordam

Konu alanlarının çakışmamasını sağlamak için bir yöntem oluşturun.

Örneğin, konu alanlarının her biri için benzersiz bir kök konu tanımlayın. Kök konuları küme konularını yapın.

- a) DEFINE TOPIC(B) TOPICSTR('B') CLUSTER('CLUSTER 1') ...
- b) DEFINE TOPIC(C) TOPICSTR('C') CLUSTER('CLUSTER 2') ...

Örnek

Şekil 37 sayfa 98 yayıncıları ve aboneleri QM3 'a bağlı olarak T_B ya da T_C 'a abone olabilir ya da abone olabilir.

Sonraki adım

Her iki kümede de konuları kullanan yayıncıları ve aboneleri, çakışan yöneticileri kuyruğa almak için bağlayın.

Yalnızca belirli bir kümedeki konuları, çakışma içinde olmayan kuyruk yöneticilerine kullanması gereken yayıncıları ve aboneleri bağlayın.

Çakışan konular

Çakışan konular, dağıtımli yayınlama/abone olma topolojisi, yayınlama ve abonelik konu dizgilerine bağlı olarak farklı konu nesnelere ilişkilendirilebildiğinde oluşur.

Bir konunun birden çok konu nesnesine çözülebileceği durumlarda, konular arasındaki örtüşmeler dikkate alınmaz.

Bir kümedeki yerel konular

Bir konu, bir küme içindeki herhangi bir kuyruk yöneticisinde tanımlanabilir. Konu yerel olarak tanımlanırsa, başka bir yerde tanımlı olan ve aynı konu dizgisine çözülen bir küme konusuna göre öncelik kazanır.

Kümedeki konuları kümeleme

Bir konu, bir küme içindeki herhangi bir kuyruk yöneticisinde tanımlanabilir. Konu kümelenmişse, kümenin diğer üyelerine kopyalanır. Konu, kümedeki başka bir kuyruk yöneticisinde küme konusu olarak tanımlandıysa, bu bir hatadır. Kuyruk yöneticisinin hata günlüğüne, varolan bir küme tanımlamasına sahip bir hata iletisi yazıldı.

Bir kural olarak, küme konularının yalnızca tek bir kuyruk yöneticisinde tanımlanmasını sağlamak için, küme başlıklarını içeren "küme konusu anasistemi" olan tek bir kuyruk yöneticisinde tanımlayın.

Bir küme konusunu yeniden tanımlarsanız, değişiklik her kuyruk yöneticisine ulaşılması zaman alır. En sonunda, en son tanım, küme olmayan konu anasistemlerine eşlenen önceki küme konusu tanımlamalarını geçersiz kılar.

Farklı önceliklere sahip kümede birden çok kuyruk yöneticisinde bir küme konusu tanımlarsanız, en son tanım önceki yerel tanımların hiçbirini geçersiz kılmaz.

Joker abonelikleri birden çok konu dizesini çözer

Bir abonelik joker karakterler içerdiğinde, bir konu alanındaki potansiyel olarak farklı konular, abonelikte eşleşebilir ve farklı konu nesnelere ilişkin abonelikte sonuçlanabilir.

Örneğin, SPORTSkümesinde aşağıdaki konu tanımlarını dikkate alın.

```
DEFINE TOPIC(A) TOPICSTR('Football/result/#') SUBSCOPE(QMGR) CLUSTER(SPORTS)
DEFINE TOPIC(B) TOPICSTR('Football/#') SUBSCOPE(ALL) CLUSTER(SPORTS)
DEFINE TOPIC(C) TOPICSTR('Football/result/Newport/Cardiff') PUBSCOPE(ALL) SUBSCOPE(ALL)
CLUSTER(SPORTS)
DEFINE TOPIC(D) TOPICSTR('Football/matches/Newport/Cardiff') PUBSCOPE(ALL) SUBSCOPE(QMGR)
CLUSTER(SPORTS)
```

Kümede iki kuyruk yöneticisi (QM1 ve QM2) olduğunu varsayın. Topics C and D are published on QM1.

Bu abonelikler gruplandırılmıyorsa, QM2 üzerindeki bir abonenin aldığı öğeyi göz önünde bulundurun.

- A konusuna abonelik hiçbir şey almayacak.
 - SUBSCOPE (QMGR) ve yayın, diğer kuyruk yöneticisinde yer alıyor.
- B konusuna abonelik her iki yayını da alır.
 - Her iki durumda da SUBSCOPE (ALL) ve PUBSCOPE (ALL) .
- C konusuna abonelik, bir yayın alır.
 - SUBSCOPE (ALL) ve PUBSCOPE (ALL) , ve C ile ilgili yayınla eşleşir.
- D konusuna abonelik hiçbir şey almayacak.
 - SUBSCOPE (QMGR) ve yayın, diğer kuyruk yöneticisinde yer alıyor.

Bu abonelikler grupsana, QM2 üzerindeki bir abonenin aldığı öğeyi göz önünde bulundurun.

- Abone, C başlıklı konuda bir yayın alır.
 - The matching subscription on topic A with SUBSCOPE (QMGR) is overridden by the matching subscription on topic C with SUBSCOPE (ALL) . Daha belirli abonelikleri kazanır ve yayın alınır.
 - The matching subscription on topic B is rejected in favor of the matching subscription on topic C, because the subscriptions are grouped, and C is more specific. Yinelenen yayın atılır.
- Abone, D başlıklı konuda bir yayın aldı.
 - The matching subscription on topic B with SUBSCOPE (ALL) is overridden by the matching subscription on topic D with SUBSCOPE (QMGR) . Daha belirli abonelik kazanır ve yayın atılır.

Döngü saptama işleri

Dağıtılmış bir yayınlama/abone olma ağlarında, yayınların ve yetkili aboneliklerin aynı özgün yayının birden çok kopyasını alan bağlı abonelerle dolu bir ağa yol göstereceği için, yayınların ve yetkili sunucu aboneliklerinin döngü yapamaması önemlidir.

“Yetkili abonelik toplama ve yayın toplaması” sayfa 50 içinde açıklanan yetkili abonelik toplama sistemi, yetkili aboneliklerin sürekli olarak döngülenmesini önlese de, bir döngünün oluşumunu engellememektedir. Yayınların yayılımı, yetkili sunucu aboneliklerinin varlığı tarafından belirlendiğinden, sürekli bir döngü içine girebilirler. Websphere MQ V7.0 , yayınları sürekli olarak döngüden önlemek için aşağıdaki tekniği kullanır:

Yayınlar yayınlama/abone olma topolojisi etrafında dolaşırken, her kuyruk yöneticisi ileti üstbilgisine benzersiz bir parmak izi ekler. Bir yayınlama/abone olma kuyruk yöneticisi başka bir yayınlama/abone olma kuyruk yöneticisinden bir yayın aldığı anda, ileti üstbilgisinde tutulan parmak izleri denetlenir. Kendi parmak izi önceden varsa, yayın bir döngü etrafında tam olarak dolaştırılır, böylece kuyruk yöneticisi iletiyi atar ve hata günlüğüne bir giriş ekler.

Not: Bir döngü içinde, yayınlar döngüdeki her iki yöne yayılır ve döngü içindeki her bir kuyruk yöneticisi, kaynak kuyruk yöneticisi döngülü yayınları atmadan önce her iki yayını da alır. Bu, döngüye bölününceye kadar yinelenen yayınların kopyalarını alan uygulamalara abone olma ile sonuçlanır.

Döngü algılama parmak izi biçimi

Döngü algılama parmak izleri, V7.0 protokolünün bir parçası olarak bir RFH2 üstbilgisine eklenir ya da bir akışa neden olur. Bir RFH2 programcısının, üstbilgiyi anması ve parmak izi bilgilerini sağlam bir şekilde aktarması gerekir. WebSphere MessageBroker , parmak izi bilgilerini içermeyecek olan RFH1 üstbilgilerini kullanır.

```
<ibm>
  <Rfp>uuid1</Rfp>
  <Rfp>uuid2</Rfp>
  <Rfp>uuid3</Rfp>
</ibm>
```

< ibm>, ziyaret edilen her bir kuyruk yöneticisinin benzersiz kullanıcı tanıtıcısını (uuid) içeren yönlendirme parmak izleri listesini bulunduran klasörün adıdır.

Bir kuyruk yöneticisi tarafından her ileti yayınlanırsa, < Rfp> (yönlendirme parmak izi) etiketini kullanarak uuid değerini < ibm> klasöre ekler. Bir yayın alındığında, WebSphere MQ , belirli bir uuid değerinin mevcut olup olmadığını görmek için < Rfp > etiketlerinde yineleme yapmak için ileti özellikleri API 'sını kullanır. Because of the way that the WebSphere Platform Messaging component of WebSphere MQ attaches to Websphere Message Broker through a channel and RFH2 subscription when using the queued publish/subscribe interface, WebSphere MQ also creates a fingerprint when it receives a publication by that route.

Bu amaç, herhangi bir RFH2 ' yi, parmak izi bilgilerimize eklediğimiz için herhangi bir beklenen uygulama için teslim etmemek.

Bir RFH2 , ileti özelliklerine dönüştürüldüğünde, < ibm> klasörünü dönüştürmek için de gerekli olacaktır; bu işlem, geçirilen ya da Websphere MQ V7.0 API 'yı kullanan uygulamalara teslim edilen RFH2 ' den parmak izi bilgilerini kaldırır.

Parmak izi bilgisi olan bir ileti bir RFH1 abonesine teslim edildiyse ya da Websphere Message Broker V6.0'a geçirilirse, parmak izi bilgileri bir RFH1' e dönüştürülür.

Websphere Message Broker V6.0 bu iletiyi, SIB gibi bir RFH2 abonesine ilettiğinde, parmak izi bilgilerini bir RFH2 biçimine dönüştürmesi gerekir.

JMS uygulamaları, parmak izi bilgilerini görmezse, JMS arabirimi bu bilgileri RFH2' den çıkarmaz ve dolayısıyla bu bilgileri uygulamalarına teslim etmez.

Rfp ileti özellikleri propDesc.CopyOptions = MQCOPY_FORWARD and MQCOPY_PUBLISH ile yaratılır. Bu, uygulamaların aldığı uygulamaların sonuçlarını içerir ve aynı iletiyi yeniden yayınlamayı sağlar. Bu, böyle bir uygulamanın PutMsgOpts.Action = MQACTP_FORWARD komutunu kullanarak parmak izi yöneltme zincirine devam edebilmesi, ancak zincirden kendi parmak izini çıkarmak için uygun bir şekilde kodlanmalıdır. Varsayılan olarak uygulama PutMsgOpts.Action = MQACTP_NEW ' ı kullanır ve yeni bir zincir başlatır.

Dağıtılmış bir yayınlama/abone olma topolojisindeki yayınları korudu

Alıkonan yayınları dağıtılmış bir yayınlama/abone olma topolojisinde kullanırken, yalnızca topolojideki tek bir kuyruk yöneticisinden aynı konuya ilişkin tutulan yayınları yayınlamak en iyi uygulamadır.

Tersi durumda, aynı konu için farklı kuyruk yöneticilerindeki farklı alıkonan yayınların etkin olabileceği, beklenmeyen davranışlara yol açan bir başka yayınlar olabilir. Birden çok yetkili sunucu aboneliği dağıtıldığı için, alıkonan birden çok yayın alınabilir.

Kuyruk yöneticileri arasında güvenliği yayınlama/abone ol

Yetkili abonelik abonelikleri ve yayınlar gibi iç iletileri yayınlama/abone olma, olağan kanal güvenlik kurallarını kullanarak sistem kuyruklarını yayınlamaya/abone olma konumuna getirmektedir. Bu konudaki bilgi ve çizgeler, bu iletilerin tesliminde yer alan çeşitli süreçleri ve kullanıcı kimliklerini vurgular.

Yerel erişim denetimi

Yayın ve aboneliklere ilişkin konulara erişim, Yayınlama/abone olma güvenliğinde açıklanan yerel güvenlik tanımlarıyla ve kurallarıyla yönetilir. z/OS' ta, erişim denetimi oluşturmak için herhangi bir yerel konu nesnesi gerekmez. Diğer altyapılarda da erişim denetimi için yerel bir konu gerekmez. Yöneticiler, kümede var olup olmamaları bağımsız olarak, kümelenmiş konu nesnelere erişim denetimi uygulamayı tercih edebilir.

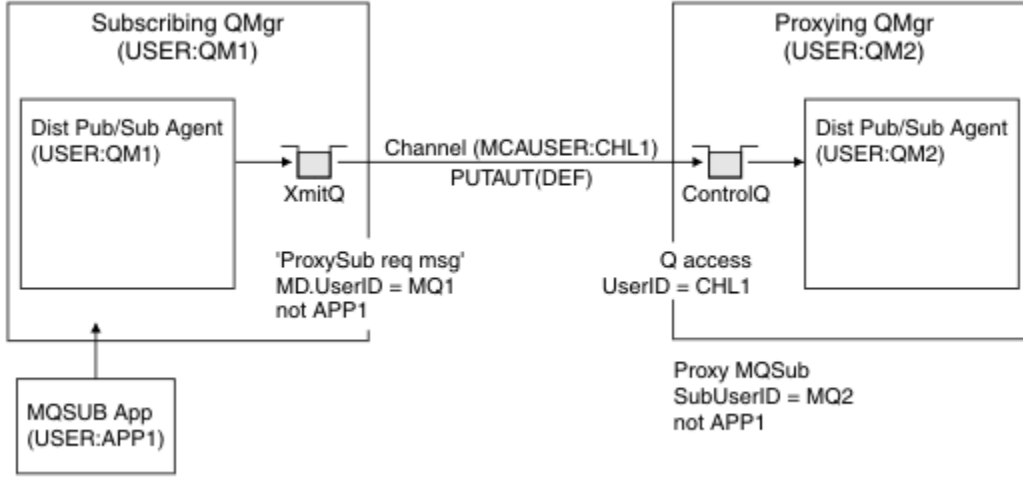
Sistem yöneticileri, yerel sistemlerindeki erişim denetiminden sorumludur. Erişim denetimi ilkesinden sorumlu olmak için, sıradüzeninin diğer üyelerinin yöneticilerine ya da küme kolektiflerine güvenmeleri gerekir. Erişim denetimi her bir ayrı makine için tanımlandığından, yüksek düzeyde denetime gerek duyulması durumunda da bu denetim ögesi büyük olasılıkla gömüledir. Erişim denetimi uygulanması gerekli olmayabilir ya da erişim denetimi, konu ağacındaki üst düzey nesnelere üzerinde tanımlanabilir. Önemli düzey erişim denetimi, konu ad alanının her bir alt bölümü için tanımlanabilir.

Yetkili abonelik oluşturma

Bir kuruluşun kuyruk yöneticisini kuyruk yöneticinize bağlayacak bir kuruluş için güven, olağan kanal kimlik doğrulaması tarafından onaylanır. Bu güvenilir kuruluşun dağıtılmış yayınlama/abone olma izni de

veriliyorsa, bir yetki denetimi yapılır. Bu denetim, kanal, dağıtılmış yayınlama/abone olma kuyruğuna bir ileti yerleştirdiğinde yapılır. Örneğin, SYSTEM . INTER . QMGR . CONTROL kuyruğuna bir ileti konursa. Kuyruk yetkisi denetiminin kullanıcı kimliği, alan kanalının PUTAUT değerlerine bağlıdır. Örneğin, kanalın kullanıcı kimliği (MCAUSER), değer ve altyapıya bağlı olarak ileti bağlamı. Kanal güvenliğiyle ilgili ek bilgi için [Kanal güvenliği](#) başlıklı konuya bakın.

Yetkili sunucu abonelikleri, uzak kuyruk yöneticilikteki dağıtılmış yayınlama/abone olma aracısının kullanıcı kimliğiyle yapılır. Örneğin, Şekil 38 sayfa 102’inde QM2 . Kullanıcı kimliği sistemde tanımlı olduğundan ve dolayısıyla etki alanı çakışmaları olmadığından, kullanıcı daha sonra yerel konu nesne profillerine kolayca erişim izni verilir.



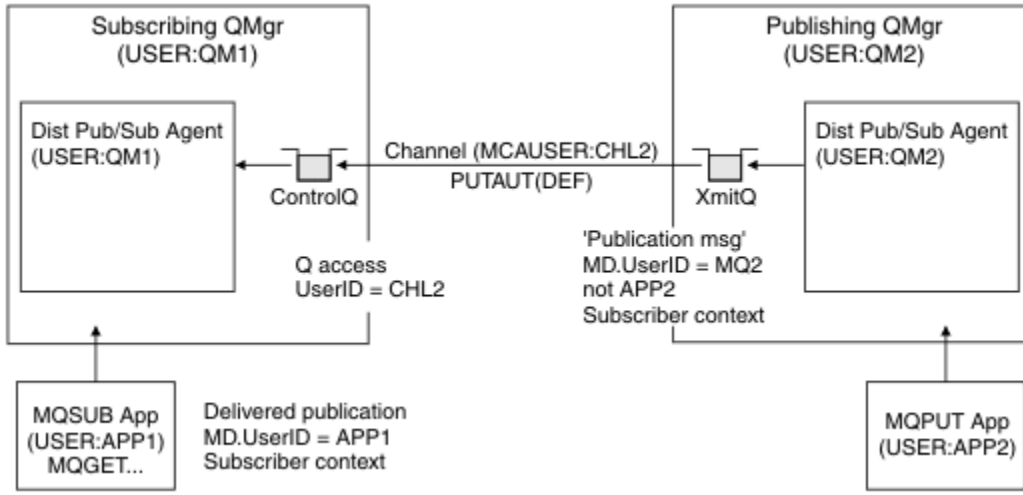
Şekil 38. Yetkili abonelik güvenliği, abonelik yapma

Uzak yayınları gönderme

Yayınlama kuyruk yöneticisince bir yayın oluşturulduğunda, herhangi bir yetkili abonelik için yayının bir kopyası oluşturulur. Kopyalanan yayının bağlamı, aboneliği yapan kullanıcı kimliğinin bağlamını içerir; Şekil 39 sayfa 103’ünde QM2 . Yetkili abonelik, uzak kuyruk olan bir hedef kuyrukla yaratılır; bu nedenle, yayının iletisi bir iletim kuyruğuna çözülür.

Bir kuruluşun kuyruk yöneticisini (QM2), başka bir kuyruk yöneticisine (QM1) bağlamak için güvenli bir kuruluşa güven, normal kanal kimlik doğrulaması tarafından onaylanır. Bu güvenilir kuruluşun dağıtılmış yayınlama/abone olma izni verilmesine izin verilirse, kanal, yayın iletisini dağıtılmış yayınlama/abone olma yayın kuyruğuna SYSTEM . INTER . QMGR . PUBS' ye koyduğunda bir yetki denetimi yapılır. Kuyruk yetkisi denetiminin kullanıcı kimliği, giriş kanalının PUTAUT değerine bağlıdır (örneğin, kanala ilişkin kullanıcı kimliği, MCAUSER, ileti bağlamı ve diğerleri, değere ve platforma bağlı olarak). Kanal güvenliğiyle ilgili ek bilgi için [Kanal güvenliği](#) başlıklı konuya bakın.

Yayın iletisi abone olunan kuyruk yöneticisine ulaştığında, konuya ilişkin başka bir MQPUT o kuyruk yöneticisinin yetkisi altında yapılır ve iletiyle bağlam, her bir yerel abonenin her biri ileti verilen her bir yerel abonenin bağlamıyla değiştirilir.



Şekil 39. Yetkili abonelik güvenliği, yayın yayınları

Güvenlikle ilgili olarak küçük sayılan bir sistemde, dağıtılmış yayınlama/abone olma işlemlerinin büyük olasılıkla mqm grubundaki bir kullanıcı kimliği altında çalıştırılması, bir kanaldaki MCAUSER parametresinin boş olması (varsayılan değer) ve iletilerin gerektiği şekilde çeşitli sistem kuyruklarına teslim edilmeleri olabilir. Güvenli olmayan sistem, dağıtımlı yayınlama/abone olma/abone olma gibi bir kavramın kanıtını ortaya koymayı kolaylaştırıyor.

güvenliğin daha ciddi olarak düşünüldüğü bir sistemde, bu iç mesajlar, kanaldan geçen herhangi bir mesajla aynı güvenlik denetimlerine tabi olur.

If the channel is set up with a non-blank MCAUSER and a PUTAUT value specifying that MCAUSER must be checked, then the MCAUSER in question must be granted access to SYSTEM. INTER. QMGR. * queues. Farklı MCAUSER kimlikleri altında çalışan kanallarla birden çok farklı uzak kuyruk yöneticisi varsa, tüm bu kullanıcı kimliklerinin SYSTEM. INTER. QMGR. * kuyruklarına erişim izni verilmesi gerekir. Farklı MCAUSER tanıtıcıları altında çalışan kanallar olabilir; örneğin, tek bir kuyruk yöneticisinde birden çok sıradüzensel bağlantı yapılandırıldığına.

Kanal, ileti bağlamının kullanıldığını belirten bir PUTAUT değeriyle ayarlandıysa, iç iletinin içindeki kullanıcı kimliğine dayalı olarak SYSTEM. INTER. QMGR. * kuyruklarına erişim denetlenir. Tüm bu iletiler, iç iletiyi ya da yayın iletisini gönderen kuyruk yöneticisinden dağıtılmış yayınlama/abone olma aracısının kullanıcı kimliği ile konduğu için (bkz. Şekil 39 sayfa 103), dağıtımlı yayınlama/abone olma güvenliğini bu şekilde ayarlamak istiyorsanız, çeşitli sistem kuyruklarına (uzak kuyruk yöneticisi başına bir tanesi) erişim izni vermek için çok büyük bir kullanıcı kimliği kümesi değil. Kanal bağlamı güvenliğinin her zaman her zaman sahip olduğu tüm sorunlar vardır; farklı kullanıcı kimliği etki alanları ve iletteki kullanıcı kimliğinin, giriş sisteminde tanımlanmaması gerekir. Ancak gerekirse, bu, gerektiğinde çalıştırılabilmenin son derece kabul edilebilir bir yoldur.

Olağan kanal güvenliği kullanılarak, dağıtılmış yayınlama/abone olma amaçları için tüm kuyruk yöneticisi ileti alışverişi yürütülür.

Konu düzeyinde yayınları ve yetkili sunucu aboneliklerini kısıtlamakla ilgili bilgi edinmek için [Yayınlama/abone olma güvenliği](#) konusuna bakın.

Kuyruk yöneticisi sıradüzeniyle varsayılan kullanıcı kimliklerini kullanma

Farklı platformlarda çalışan ve varsayılan kullanıcı kimliklerini kullanan bir kuyruk yöneticisi sıradüzeniniz varsa, bu varsayılan kullanıcı kimliklerinin altyapılar arasında farklılık gösterdiğine ve hedef altyapıda bilinmeyebileceğinin unutulmamasını unutmayın. Sonuç olarak, bir platformda çalışan bir kuyruk yöneticisi, kuyruk yöneticilerinden alınan iletileri, MQRC_NOT_AUTHORIZEDneden koduyla birlikte diğer platformlarda reddeder.

Reddedilmekte olan iletileri minimum olarak önlemek için, aşağıdaki yetkilerin diğer platformlarda kullanılan varsayılan kullanıcı kimliklerine eklenmesi gerekir:

- *PUT *GET yetkisi SYSTEM.BROKER. Kuyruklar
- *PUB *SUB authority on the SYSTEM.BROKER. Konular
- *ADMCR *ADMILT *ADMCHG authority on the SYSTEM.BROKER.CONTROL.QUEUE queue.

Varsayılan kullanıcı kimlikleri aşağıdaki gibidir:

Altyapı	Varsayılan kullanıcı kimliği
Pencereler	MUSR_MQADMIN Not: MUSR_MQADMIN , yalnızca ilk kuruluş için varsayılan kullanıcı kimliğidir. Sonraki kuruluşlarda, Prepare IBM WebSphere MQ Wizard,MUSR_MQADMINiçin adlandırılan bir kullanıcı hesabı yaratır; burada x, var olmayan bir kullanıcı kimliğini temsil eden sonraki kullanılabilir sayıdır.
UNIX and Linux sistemleri	mqm
IBM i	QMQM
z/OS	Kanal başlatıcı adres alanı kullanıcı kimliği

Windows, UNIX, Linuxve z/OS platformlarında Kuyruk Yöneticileri için IBM i 'de bir kuyruk yöneticisine hierarchy olarak eklenirse,' qmqm ' kullanıcı kimliği için erişim yaratın ve bu kullanıcı kimliğine erişim verin.

IBM i ve z/OS platformlarındaki Kuyruk Yöneticileri için Windows, UNIXya da Linux üzerinde bir kuyruk yöneticisine hierarchy olarak bağlandıysa, 'mqm' kullanıcı kimliği oluşturun ve bu kimlik için erişim izni verin.

Windows, UNIX, Linuxve IBM i platformlarındaki Kuyruk Yöneticileri için z/OS üzerindeki bir kuyruk yöneticisine sıradüzensel olarak bağlı olduğunda, z/OS kanal başlatıcı adres alanı kullanıcı kimliğine kullanıcı erişimi yaratın ve bu kullanıcı kimliğini verin.

Kullanıcı kimlikleri büyük ve küçük harfe duyarlı olabilir. Kaynak kuyruk yöneticisi (IBM i, Windows, UNIXya da Linux sistemleri ise), kullanıcı kimliğini tüm büyük harflere sahip olacak şekilde zorlar. Alıcı kuyruk yöneticisi (Windows, UNIX ya da Linux sistemleri ise), kullanıcı kimliğini tüm küçük harf olarak kabul eder. Bu nedenle, UNIX and Linux sistemlerinde yaratılan tüm kullanıcı kimlikleri küçük harfli biçimlerinde yaratılmalıdır. Bir ileti çıkışı kurulduysa, kullanıcı kimliğini büyük harfli ya da küçük harfe zorlamak yerine getirmez. İleti çıkışının kullanıcı kimliğini nasıl işlediğini anlamak için dikkatli olmanız gerekir.

Kullanıcı kimliklerinin dönüştürülmesiyle ilgili olası sorunları önlemek için:

- UNIX, Linux ve Windows sistemlerinde kullanıcı kimliklerinin küçük harfli olarak belirtildiğinden emin olun.
- IBM i ve z/OSüzerinde, kullanıcı kimliklerinin büyük harfli olarak belirtildiğinden emin olun.

Dağıtılmış yayınlama/abone olma sistem kuyrukları

Yayınlama/abone olma ileti alışverişi için kuyruk yöneticileri tarafından dört sistem kuyruğu kullanılır. Varlıklarının yalnızca sorun belirleme ya da kapasite planlama amaçlarıyla farkında olmanız gerekir.

<i>Çizelge 9. Dağıtılmış altyapılarda yayınlama/abone olma sistem kuyrukları</i>	
Sistem kuyruğu	Amaç
SYSTEM.INTER.QMGR.CONTROL	WebSphere MQ dağıtılmış yayınlama/abone olma denetim kuyruğu
SYSTEM.INTER.QMGR.FANREQ	WebSphere MQ dağıtılmış yayınlama/abone olma iç yetkili aboneliği fan-out süreci giriş kuyruğu
SYSTEM.INTER.QMGR.PUBS	WebSphere MQ dağıtılmış yayınlama/abone olma yayınları

<i>Çizelge 9. Dağıtılmış altyapılarda yayınlama/abone olma sistem kuyrukları (devamı var)</i>	
Sistem kuyruğu	Amaç
SYSTEM.HIERARCHY.STATE	WebSphere MQ dağıtılmış yayınlama/abone olma sıradüzeni ilişkisi durumu

Yayınlama/abone olma sistem kuyruklarının öznitelikleri [Çizelge 10 sayfa 105](#) içinde gösterilir.

<i>Çizelge 10. Yayınlama/abone olma sistem kuyruklarının öznitelikleri</i>	
Öznitelik	Varsayılan değer
DEĞERLERI	Evet
DEFSOPT	EXC
MAXMSGL	AIX, HP-UX, Linux, IBM i, Solaris ve Pencereleler platformlarında: ALTER QMGR komutunun MAXMSGL parametresindeki değer
MAXDEPTH	999999999
Paylaş	Uyglnmz
STGCLASS	Bu öznitelik yalnızca z/OS platformlarında kullanılır.

Yayınlama/abone olma sistem kuyruğu hataları

Dağıtılmış yayınlama/abone olma kuyruk yöneticisi kuyrukları kullanılmadığında hatalar oluşabilir.

Fan çıkış isteği kuyruğu SYSTEM.INTER.QMGR.FANREQ kullanılmıyor, MQSUB API ' nın aldığı neden kodları ve hata iletileri, yetkili sunucu aboneliklerinin doğrudan bağlı kuyruk yöneticilerine teslim edilmesi gereken durumlarda, hata günlüğüne yazılır.

Sıradüzen ilişkisi durum kuyruğu SYSTEM.HIERARCHY.STATE kullanılmıyor, hata günlüğüne hata iletileri yazıldı ve yayınlama/abone olma motoru COMPAT kipine sokuyor.

SYSTEM.INTER.QMGR kuyrukları kullanılamaz, hata günlüğüne bir hata iletileri yazılır ve işlev geçersiz kılınmasa da, yayınlama/abone olma iletilerinin uzak kuyruk yöneticilerindeki kuyruklarda oluşturulacağı olasılığı vardır.

Bir üst öge, alt öge ya da yayınlama/abone olma kümesi kuyruk yöneticisi için iletim kuyruğu kullanılamaz durumda olduğunda:

1. MQPUT API, neden kodları alır ve yayınlar teslim edilmez.
2. Alınan kuyruk yöneticisi yayınları, giriş kuyruğuna geriletir ve geriletme eşliğine ulaşırsa, ileti kuyruğunda yeniden denenmeye çalışılır.
3. Yedek sunucu abonelikleri fanout istek kuyruğuna yedeklenir ve daha sonra, geriletme eşliğine ulaşırsa, çıkmaz ileti kuyruğuna yerleştirilmek üzere yeniden denenir; bu durumda, yetkili sunucu aboneliği bağlı kuyruk yöneticisine teslim edilmeyecektir.
4. Hiyerarşi ilişkisi iletişim kuralı iletileri başarısız oldu ve bağlantı durumu PUBSUB komutunda HATA olarak işaretlendi.

Teslim edilmeyen iletilerin WebSphere MQ ölü-harfli kuyruk işleyicisi ile işlenmesi

Ölü harf kuyruğunuz nedir, nasıl mesajlar konabiliyor, ve bunu nasıl yönetiyorsun?

Bazen *teslim edilemeyen ileti kuyruğu* olarak da bilinen bir *kuyruk-çıkış kuyruğu* (DLQ), hedef kuyruklarına teslim edilemeyen iletiler için bir tutma kuyruğudur. Bir ağdaki her kuyruk yöneticisinin ilişkili bir DLQ ' ya sahip olması gerekir.

İletiler, kuyruk yöneticilerine, ileti kanalı aracılara (MCA 'lar) ve uygulamalara göre DLQ' ya yerleştirilebilir. All messages on the DLQ must be prefixed with a *dead-letter üstbilgisi* structure, MQDLH.

Bir kuyruk yöneticisi ya da ileti kanalı aracısı tarafından DLQ 'ya ilişkin iletiler her zaman bir MQDLH (MQDLH) içerir; DLQ' ya ileti yerleştiren uygulamalar bir MQDLH sağlamalıdır. MQDLH yapısının *Reason* (Neden) alanı, iletinin DLQ üzerinde neden olduğunu belirten bir neden kodu içerir.

Tüm WebSphere MQ ortamlarının DLQ 'daki iletileri düzenli olarak işlemek için bir yordamı olması gerekir. WebSphere MQ , runmqdlq komutunu kullanarak çağırduğunuz *ölü-harflı kuyruk işleyicisi* (DLQ işleyicisi) adı verilen varsayılan bir yordam sağlar.

DLQ 'daki iletilerin işlenmesine ilişkin yönergeler, kullanıcı tarafından yazılan *kural çizelgesiaracılığıyla* DLQ işleyicisine sağlanır. Yani, DLQ işleyicisi, kurallar çizelgesindeki girişlere karşı DLQ 'daki iletilerle eşleşir; bir DLQ iletisi kural çizelgesindeki bir girişle eşleştiğinde, DLQ işleyicisi bu girişle ilişkili işlemi gerçekleştirir.

DLQ işleyicisini çağırma

Invoke the DLQ handler using the runmqdlq command. İşlemek istediğiniz DLQ 'yı ve kullanmak istediğiniz kuyruk yöneticisini iki şekilde adlayabilirsiniz.

İki yol aşağıdaki gibidir:

- Komut isteminden runmqdlq 'e parametre olarak. Örneğin:

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER <qrulename>
```

- Kurallar tablosunda. Örneğin:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

Örnekler, ABC1.DEAD.LETTER.QUEUE(kuyruk yöneticisi) ABC1.QUEUE.MANAGER

DLQ ya da kuyruk yöneticisini gösterildiği gibi belirtmezseniz, kuruluş için varsayılan kuyruk yöneticisi, o kuyruk yöneticisine ait DLQ 'yla birlikte kullanılır.

runmqdlq komutu, girişini stdin' dan alır; kural çizelgesinden stdin komutunu yeniden yönlendirerek, kural çizelgesini runmqdlq ile ilişkilendirin.

DLQ işleyicisini çalıştırmak için, DLQ 'ya hem kendisine, hem de DLQ üzerindeki iletilerin iletileceği ileti kuyruklarına erişmeniz için yetki verilmelidir. DLQ işleyicisi için, ileti bağlamındaki kullanıcı kimliğinin yetkisi olan kuyruklara ileti yerleştirmesi için, diğer kullanıcıların kimliğini de üstlenmek için yetkilendirilmiş olmanız gerekir.

runmqdlq komutuna ilişkin daha fazla bilgi için bkz. [runmqdlq](#).

Örnek DLQ işleyicisi, amqsdlq

runmqdlq komutu kullanılarak çağrılan DLQ işleyicisine ek olarak, WebSphere MQ , runmqdlq tarafından sağlanan işlevle benzer bir işleve sahip bir örnek DLQ işleyicisi, amqsdlq ile ilgili kaynağı sağlar.

Gereksinimlerinizi karşılayan bir DLQ işleyicisi sağlamak için amqsdlq uygulamasını özelleştirebilirsiniz. Örneğin, iletileri ölü harf üstbilgileri olmadan işleyebilecek bir DLQ işleyicisi istediğinize karar verebilirsiniz. (Hem varsayılan DLQ işleyicisi hem de örnek, amqsdlq, yalnızca bir ölü-harflı üstbilgisiyle başlayan DLQ 'daki iletileri işleme alan MQDLH. MQDLH ile başlamayan iletiler hata olarak tanımlanır ve DLQ 'da süresiz olarak kalır.)

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

WebSphere MQ 'da Windows için, amqsdlq kaynağı dizinde sağlanıyor:

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

ve derlenen sürüm şu dizinde sağlanıyor:

`MQ_INSTALLATION_PATH\tools\c\samples\bin`

In WebSphere MQ for UNIX and Linux systems, the source of `amqsdlq` is supplied in the directory:

`MQ_INSTALLATION_PATH/samp/dlq`

ve derlenen sürüm şu dizinde sağlanıyor:

`MQ_INSTALLATION_PATH/samp/bin`

DLQ işleyici kuralları çizelgesi

DLQ işleyici kuralları çizelgesi, DLQ işleyicinin, DLQ ' ya gelen iletileri nasıl işleyeceğini tanımlar.

Bir kural çizelgesinde iki tip giriş vardır:

- Çizelgedeki ilk giriş (isteğe bağlı) *denetim verileri* içerir.
- Çizelgedeki diğer tüm girişler, izlenecek DLQ işleyicisine ilişkin *kurallar* ' dır. Her kural, bir iletinin karşılaştırıldığı bir *örüntüden* (ileti özellikleri kümesi) ve DLQ üzerindeki bir ileti belirtilen örüntüyle eşleştğinde alınacak bir *işlem* ile oluşur. Bir kural çizelgesinde en az bir kural olmalıdır.

Kural çizelgesindeki her giriş bir ya da daha çok anahtar sözcükten oluşur.

Denetim verileri

Bu bölümde, bir DLQ işleyici kuralları çizelgesinde denetim verileri girdisine dahil edebildiğiniz anahtar sözcükler açıklanmaktadır.

Not:

- Düşey çizgi (!) alternatifleri ayırır, yalnızca biri belirlenebilir.
- Tüm anahtar sözcükler isteğe bağlıdır.

INPUTQ (QueueName| '___')

İşlemek istediğiniz DLQ ' nun adı:

1. `runmqdlq` komutuna parametre olarak sağladığınız herhangi bir INPUTQ değeri, kural çizelgesindeki INPUTQ değerini geçersiz kılar.
2. If you do not specify an INPUTQ value as a parameter to the `runmqdlq` command, but you **ya** specify a value in the rules table, the INPUTQ value in the rules table is used.
3. If no DLQ is specified or you specify INPUTQ(' ') in the rules table, the name of the DLQ belonging to the queue manager with the name that is supplied as a parameter to the `runmqdlq` command is used.
4. `runmqdlq` komutuna değiştirge olarak bir Inputq değeri belirtmezseniz ya da kural çizelgesinde bir değer olarak, kural çizelgesindeki INPUTQM anahtar sözcüğünde adı belirtilen kuyruk yöneticisine ait DLQ kullanılır.

INPUTQM (QueueManagerName| '___')

INPUTQ anahtar sözcüğündeki DLQ adlı DLQ ' nun sahibi olan kuyruk yöneticisinin adı:

1. `runmqdlq` komutuna parametre olarak sağladığınız INPUTQM değeri, kural çizelgesindeki INPUTQM değerini geçersiz kılar.
2. `runmqdlq` komutuna parametre olarak bir INPUTQM değeri belirtmezseniz, kural çizelgesindeki INPUTQM değeri kullanılır.
3. Kuyruk yöneticisi belirtilmediyse ya da kural çizelgesinde INPUTQM ('') belirtilirse, kuruluş için varsayılan kuyruk yöneticisi kullanılır.

RETRYINT (Aralık|_60)

DLQ işleyicinin, ilk denemede işlenememesi gereken ve yinelenen girişimler istendiği için, DLQ işleyicinin DLQ üzerindeki iletileri yeniden işlemesi gerektiği saniye cinsinden aralık. Varsayılan değer olarak, yeniden deneme aralığı 60 saniyedir.

WAIT (YES| NO |nnn)

DLQ işleyicinin, işleyebileceği başka ileti olmadığını saptadığında DLQ ' ya daha fazla ileti gelmesini beklemesi gerekip gerekmediğini belirler.

EVET

DLQ işleyicisi süresiz olarak bekler.

HAYIR

DLQ işleyicisi, DLQ ' un boş olduğunu algıladığında ya da işleyebileceği bir ileti içerip içermediği saptanırken sona erer.

nnn

DLQ işleyicisi, kuyruğun boş olduğunu ya da işleyebileceği hiçbir ileti içermediği algıladıktan sonra, sona erdirilmeden önce yeni çalışma için nnn saniye bekler.

Meşgul DLQ 'lar için WAIT (YES) ve düşük düzeyde etkinlik olan DLQ' lar için WAIT (NO) ya da WAIT (nnn) belirleyin. DLQ işleyicinin sona erdirilmesine izin verilirse, tetikleme kullanarak yeniden çağırın. Tetikleme hakkında daha fazla bilgi için bkz. [WebSphere MQ uygulamalarının tetikleyicileri kullanarak başlatılması](#).

An alternative to including control data in the rules table is to supply the names of the DLQ and its queue manager as input parameters to the runmqdlq command. Hem kural tablosunda hem de runmqdlq komutuna giriş olarak bir değer belirtirseniz, runmqdlq komutunda belirtilen değer öncelik kazanır.

Kural çizelgesine bir denetim verileri girişi eklerseniz, bu giriş, çizelgedeki **ilk** giriş olmalıdır.

Kurallar (örüntüler ve işlemler)

Kalıp eşleştirme anahtar sözcüklerinin (DLQ üzerindeki iletilerin eşleştiği) ve işlem anahtar sözcüklerinin (DLQ işleyicisinin eşleşen bir iletiyi nasıl işleyeceğini belirleyen) bir açıklaması. Örnek bir kural da sağlanmıştır.

Kalıp eşleştirme anahtar sözcükleri

DLQ üzerindeki iletilerin eşleştirildiği değerleri belirlemek için kullandığınız kalıp eşleştirmesi anahtar sözcükleri aşağıdaki gibidir. (Tüm örüntü-eşleşen anahtar sözcükler isteğe bağlıdır):

APPLIDAT (*ApplIdentityVerileri* | *)

DLQ üzerindeki iletinin ileti tanımlayıcısında, MQMD ' de belirtilen *ApplIdentityData* değeri.

APPLNAME (*PutApplAd* | *)

The name of the application that issued the MQPUT or MQPUT1 call, as specified in the *PutApplAd* field of the message descriptor, MQMD, of the message on the DLQ.

APPLTYPE (*PutApplTip* | *)

DLQ üzerindeki iletinin ileti tanımlayıcısında, MQMD ' de belirtilen *PutApplType* değeri.

DESTQ (*QueueName* | *)

İletin kaderinde bulunduğu ileti kuyruğunun adı.

DESTQM (*QueueManagerAdı* | *)

İletin kaderinde bulunduğu ileti kuyruğunun kuyruk yöneticisinin adı.

FEEDBACK (*Geribildirim* | *)

MsgType değeri MQFB_REPORT ise, *Geribildirim* , raporun niteini açıklar.

Simgesel adları kullanabilirsiniz. Örneğin, MQFB_COA simgesel adını kullanarak, hedef kuyruklarına varışlarının doğrulanabilmesi için DLQ ' da bu iletileri tanımlayabilirsiniz.

BIÇIM (*Biçim* | *)

İletin göndericisinin, ileti verilerinin biçimini tanımlamak için kullandığı ad.

MSGTYPE (*MsgType* | *)

DLQ ' da iletinin ileti tipi.

Simgesel adları kullanabilirsiniz. Örneğin, yanıt ihtiyacı olan DLQ ' da bu iletileri tanımlamak için MQMT_REQUEST simgesel adını kullanabilirsiniz.

PERSSIN (Kalıcılık|_*)

İletinin kalıcılık değeri. (Bir iletinin kalıcılığı, kuyruk yöneticisinin yeniden başlatılıp yeniden başlatılıp başlatılmayacağını belirler.)

Simgesel adları kullanabilirsiniz. Örneğin, kalıcı olan DLQ ile ilgili iletileri tanımlamak için MQPER_PERSISTENT simgesel adını kullanabilirsiniz.

NEDEN (ReasonCode|_*)

İletinin DLQ 'ya neden konduğunu açıklayan neden kodu.

Simgesel adları kullanabilirsiniz. Örneğin, hedef kuyrukları dolu olduğu için, DLQ 'ya yerleştirilen iletileri tanımlamak için MQRC_Q_FULL simgesel adını kullanabilirsiniz.

REPLYQ (QueueName|_*)

DLQ üzerindeki iletinin, ileti tanımlayıcısında, MQMD 'de belirtilen yanıtlama kuyruğunun adı.

REPLYQM (QueueManagerAd|_*)

DLQ üzerindeki iletinin, ileti tanımlayıcısında, MQMD 'de belirtildiği gibi, yanıtlanacak kuyruğa ilişkin kuyruk yöneticisinin adı.

USERID (UserIdentifier|_*)

DLQ üzerindeki iletiyi oluşturan kullanıcının kullanıcı kimliği (DLQ ileti tanımlayıcısı, MQMD 'de belirtildiği gibi), DLQ 'daki iletinin tanıtıcısı.

İşlem anahtar sözcükleri

Eşleşen bir iletinin nasıl işleneceğini tanımlamak için kullanılan işlem anahtar sözcükleri şunlardır:

İŞLEM (AT | YOKSAYIN | YENIDEN DENE | FWD)

Bu kuralda tanımlanan örüntüyle eşleşen DLQ 'da herhangi bir ileti için yapılacak işlem.

At

DLQ 'dan iletiyi silin.

IGNORE

İletiyi DLQ 'ya bırakın.

Yeniden deneme

İletiyi hedef kuyruğuna yerleştirmeye yönelik ilk girişim başarısız olursa, yeniden deneyin.

RETRE anahtar sözcüğü, bir işlemi gerçekleştirmek için yapılan deneme sayısını ayarlar. Denetim verilerinin RETRYINT anahtar sözcüğü, girişimler arasındaki aralığı denetler.

ZGD

İletiyi, FWDQ anahtar sözcüğündeki kuyruğa iletin.

ACTION anahtar sözcüğünü belirtmeniz gerekir.

FWDQ (QueueName| & DESTQ | & REPLYQ)

ACTION (FWD) istendiğinde iletinin iletileceği ileti kuyruğunun adı.

QueueName

İleti kuyruğunun adı. FWDQ (") geçerli değil.

& DESTQ

MQDLH yapısındaki *DestQName* alanında kuyruk adını alın.

& YANITLA

İleti açıklayıcısındaki *ReplyToQ* alanından kuyruk adını alın, MQMD.

Bir kural boş *ReplyToS* alanına sahip bir iletiyle eşleştiğinde hata iletilerini önlemek için, ileti örüntüsünde REPLYQ (? *) değerini belirtin.

FWDQM (QueueManagerAdı| & DESTQM | & REPLYQM | '___')

İletinin iletileceği kuyruğun kuyruk yöneticisi.

QueueManagerAdı

ACTION (FWD) istendiğinde iletinin iletileceği kuyruğun kuyruk yöneticisinin adı.

& DESTQM

Kuyruk yöneticisi adını, MQDLH yapısındaki *DestQMgrName* (DestQMgr Adı) alanından alın.

& REPLYQM

İleti tanımlayıcısındaki *ReplyToQMgr* alanından kuyruk yöneticisi adını alın, MQMD.

..

Varsayılan değer olan FWDQM ('), yerel kuyruk yöneticisini tanıtır.

ÜSTBILGI (YES| NO)

MQDLH ' nin, ACTION (FWD) istendiği bir iletide kalması gerekip gerekmediğini belirleyin. Varsayılan olarak, MQDLH iletiyle kalır. HEADER anahtar sözcüğü, FWD dışındaki işlemler için geçerli değil.

PUUTAT (DEF| CTX)

DLQ işleyicisine hangi iletilerin yerleştirilmesi gerektiği konusunda yetki:

DÖF

DLQ işleyicisinin yetkisini kullanarak iletileri girin.

CTX

İleti bağlamındaki, kullanıcı kimliğinin yetkisi olan iletileri yazın. PUTAUT (CTX) belirlerseniz, diğer kullanıcıların kimliğini üstlenmek için yetkili olmanız gerekir.

YENIDEN DENE (*RetryCount*| | 1)

Bir işlemi (denetim verilerinin RETRYINT anahtar sözcüğünde belirlenen aralıkta) denemek için 1-999.999.999 aralığında yer alan sayı. Belirli bir kuralı uygulamak için DLQ işleyicisinin yaptığı girişimler sayısı, DLQ işleyicisinin yürürlükteki örneğine özgüdür; sayım, yeniden başlatma işlemleri boyunca devam etmez. DLQ işleyicisi yeniden başlatılırsa, bir kuralı uygulamak için yapılan girişlerin sayısı sıfıra sıfırlanır.

Örnek kural

Bir DLQ işleyicisi kural çizelgesinden örnek bir kural vardır:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Bu kural, DLQ işleyicisine, MQPUT ve MQPUT1 tarafından engellendikleri için, hedef kuyruğa alma hedefine yönelik olarak DQ ' ya yönelik üç deneme girişiminde bulunmaya karar verir.

Bir kuralda kullanabileceğiniz tüm anahtar sözcükler bu bölümün geri kalan kısmında açıklanmıştır. Aşağıdakileri unutmayın:

- Bir anahtar sözcüğün varsayılan değeri (varsa) altı çizilir. Çoğu anahtar sözcük için, varsayılan değer * (yıldız işareti), herhangi bir değerle eşleşir.
- Düşey çizgi (!) alternatifleri ayırır, yalnızca biri belirlenebilir.
- ACTION dışındaki tüm anahtar sözcükler isteğe bağlıdır.

Kural çizelgesi kuralları

DLQ işleyici kuralları çizelgesinin sözdizimi, yapısı ve içeriği bu kurallarla uyumlu olmalıdır.

Kural çizelgesinin aşağıdaki kurallara uyması gerekir:

- Bir kural çizelgesinin en az bir kural içermesi gerekir.
- Anahtar sözcükler herhangi bir sırada olabilir.
- Bir anahtar sözcük herhangi bir kuralda yalnızca bir kez eklenebilir.
- Anahtar sözcükler büyük ve küçük harfe duyarlı değildir.
- Bir anahtar sözcük ve onun değiştirge değeri, en az bir boşluk ya da virgül ile diğer anahtar sözcüklerden ayrılmalıdır.
- Bir kuralın başında ya da sonunda ve anahtar sözcükler, noktalama işaretleri ve değerler arasında herhangi bir sayı boşluk olabilir.

- Her kural yeni bir satırda başlamalıdır.
- Windows sistemlerinde, çizelgenin son kuralı satır başı/satır besleme karakteriyle bitmelidir. Bu işlemi, kuralın sonundaki Enter tuşuna basmanızı sağlayarak, çizelgenin son satırının boş bir çizgi olmasını sağlayarak bunu başarabilirsiniz.
- Taşınabilirlik nedenleriyle, çizginin önemli uzunluğunun 72 karakterden uzun olmamalıdır.
- Kuralın, sonraki satırın ilk boş olmayan karakterinden devam ettiğini göstermek için, satırdaki son boş olmayan karakter olarak artı işareti (+) kullanın. Kuralın sonraki satırın başlangıcından devam ettiğini belirtmek için, satırdaki son boş olmayan karakter olarak eksi işareti (-) kullanın. Anahtar sözcükler ve değiştirgelerde devam karakterleri oluşabilir.

Örneğin:

```
APPLNAME(' ABC+
D')
```

'ABCD' ile sonuçlanır ve

```
APPLNAME(' ABC-
D')
```

' ABC D' nin sonuçları.

- Yıldız işareti (*) ile başlayan açıklama satırları, kural çizelgesinin herhangi bir yerinde oluşabilir.
- Boş satırlar yoksayılır.
- DLQ işleyici kuralları çizelgesindeki her giriş bir ya da daha çok anahtar sözcükten ve ilişkili değiştirgelerden oluşur. Değiştirgelerin sözdizimi kurallarına uygun olması gerekir:
 - Her parametre değerinin en az bir anlamlı karakter içermesi gerekir. Tırnak içine alınmış değerlerdeki sınırlayıcı tek tırnak işaretleri anlamlı olarak kabul edilmez. Örneğin, bu değiştirgeler geçerlidir:

FORMAT(' ABC ')	3 önemli karakter
FORMAT(ABC)	3 önemli karakter
FORMAT(' A ')	1 önemli karakter
FORMAT(A)	1 önemli karakter
FORMAT(' ')	1 önemli karakter

Bu değiştirgeler önemli karakterler içermediği için geçersizdir:

```
FORMAT(' ' )
FORMAT( )
FORMAT( )
FORMAT
```

- Genel arama karakterleri desteklenir. Sonda boşluk dışında herhangi bir tek karakter yerine soru işaretini (?) kullanabilirsiniz; sıfır ya da daha fazla bitişik karakter yerine yıldız işaretini (*) kullanabilirsiniz. Yıldız imi (*) ve soru işareti (?), **her zaman** parametre değerlerinde genel arama karakteri olarak yorumlanır.
- Şu anahtar sözcüklerin parametrelerinde genel arama karakterleri bulunamaz: ACTION, HEADER, RETRY, FWDQ, FWDQM ve PUTAUT.
- Parametre değerlerinde sondaki boşluklar ve DLQ üzerindeki iletilerde karşılık gelen alanlarda, joker karakter eşleşmeleri gerçekleştirilirken önemli değildir. Ancak, tek tırnak içine alınmış dizgiler içinde baştaki ve gömülü boşluklar, genel arama karakteri eşleşmeleri için önemlidir.

- Sayısal parametreler soru işareti (?) genel arama karakteri içeremez. Sayısal bir değiştirgenin bir parçası olarak değil, sayısal bir değiştirgenin tamamı yerine yıldız (*) işaretini kullanabilirsiniz. Örneğin, bunlar geçerli sayısal değiştirgelerdir:

MSGTYPE(2)	Yalnızca yanıt iletileri seçilebilir
MSGTYPE(*)	Herhangi bir ileti tipi seçilebilir
MSGTYPE(' * ')	Herhangi bir ileti tipi seçilebilir

Ancak, sayısal bir parametrenin parçası olarak bir yıldız işareti (*) içerdiğinden MSGTYPE(' 2* ') geçerli değildir.

- Sayısal parametreler 0-999 999 999 aralığında olmalı. Parametre değeri bu aralıktaki değerse, anahtar sözcüğün ilişkili olduğu alanda geçerli olmasa da kabul edilir. Sayısal parametreler için simgesel adlar kullanabilirsiniz.
- Bir dizilim değeri, anahtar sözcüğün ilgili olduğu MQDLH ya da MQMD ' deki alandan kısaysa, değer, alanın uzunluğuna kadar boşluklarla doldurulur. Yıldız imleri hariç olmak üzere değer, alandan daha uzunsa bir hataya neden olur. Örneğin, bunlar 8 karakterlik bir alana ilişkin geçerli dizgi değerleridir:

' ABCDEFGH '	8 karakter
' A*C*E*G*I '	Yıldız imleri dışında 5 karakter
' *A*C*E*G*I*K*M*O * '	Yıldız imleri dışında 8 karakter

- Boşluk, küçük harf karakterleri ya da nokta (.), eğik çizgi (?), alt çizgi (_) ve yüzde imi (%) içeren dizgileri tek tırnak işareti içine alın. Tek tırnak içine alınmayan küçük harfli karakterler büyük harfe çevrilir. Dizgide bir tırnak işareti varsa, hem başlangıç hem de sonun sonunu belirtmek için iki tek tırnak işareti kullanın. Dizilimin uzunluğu hesaplanırken, çift tırnak işaretlerinin her bir oluşumu tek bir karakter olarak sayılır.

Kural çizelgesinin nasıl işlendiği

DLQ işleyicisi, kuralın DLQ ' da bir iletiyle eşleştiği bir kural için kural çizelgesini arar.

Arama, tablodaki ilk kuralla başlar ve tablo aracılığıyla sırayla devam eder. DLQ işleyicisi, eşleştirme örüntüyle bir kural bulunduğunda, bu kuraldan işlem alır. DLQ işleyicisi, kural uygulandığında, bir kurala ilişkin yeniden deneme sayısını 1 artırır. İlk deneme başarısız olursa, DLQ işleyicisi RETRY anahtar sözcüğünde belirtilen sayıyla eşleşmeye çalışılincaya kadar yeniden dener. Tüm denemeler başarısız olursa, DLQ işleyicisi çizelgede sonraki eşleşen kuralı arar.

Bu işlem, bir işlem başarılı oluncaya kadar sonraki eşleştirme kuralları için yinelenir. Her eşleştirme kuralı RETRY anahtar sözcüğünde belirtilen sayıda kez denendiğinde ve tüm girişimler başarısız olduğunda, ACTION (IGNORE) varsayılır. Eşleşen bir kural bulunmazsa, ACTION (IGNORE) da varsayılır.

Not:

1. Eşleşen kural örüntüleri yalnızca, bir MQDLH ile başlayan DLQ ' daki iletiler için aranır. MQDLH ile başlamayan iletiler, düzenli olarak hata olarak bildirilir ve DLQ ' da süresiz olarak kalır.
2. Bir kuralın yalnızca bir eylemden oluşabilmesi için, varsayılan olarak tüm örüntü anahtar sözcüklerine izin verilebilir. Ancak, kuyrukta MQDLHs içeren ve çizelgeye ilişkin diğer kurallara uygun olarak işlenmemiş olan tüm iletilere yalnızca işlem kuralları uygulanmaktadır.
3. Kural çizelgesi, DLQ işleyicisi başlatıldığında doğrulanır ve o sırada hatalar işaretlenir. Kural çizelgesinde istediğiniz zaman değişiklik yapabilirsiniz, ancak DLQ işleyicisi yeniden başlatılincaya kadar bu değişiklikler yürürlüğe girmez.
4. DLQ işleyicisi, iletilerin içeriğini, MQDLH ' yi ya da ileti tanımlayıcısını değiştirmez. DLQ işleyicisi her zaman, iletileri MQPMO_PASS_ALL_CONTEXT ileti seçeneği ile diğer kuyruklara koyar.
5. Kural çizelgesi, geçerlilik denetimi sırasında yinelenen hataların oluşturulmasını ortadan kaldırmak için tasarlandığı için, kural çizelgesinde art arda sözdizimi hataları tanınmayabilir.

6. DLQ işleyicisi, DLQ ' yı MQOO_INPUT_AS_Q_DEF seçeneğiyle açar.
7. Aynı kural çizelgesini kullanarak, DLQ işleyicisinin birden çok eşgörünümü aynı kuyruğa koşturarak koşut zamanlı olarak çalışabilir. Ancak, bir DLQ ile DLQ işleyicisi arasında bire bir ilişki olması daha olağan bir durum.

Tüm DLQ iletilerinin işlenmesini sağlama

DLQ işleyicisi, DLQ üzerinde görülen, ancak kaldırılmamış olan tüm iletilerin kaydını tutar.

DLQ işleyicisini, DLQ ' dan gelen iletilerin küçük bir alt kümesini çıkarmak için süzgeç olarak kullanırsanız, DLQ işleyicisi, DLQ üzerinde işlem yapmadığı iletilerin kaydını tutmak zorunda kalabilir. DLQ işleyicisi, DLQ 'ya gelen yeni iletilerin, ilk giren ilk çıkış (FIFO) olarak tanımlansa bile, DLQ' ya yeni iletilerin görülmesini garanti edemez. Kuyruk boş değilse, DLQ, tüm iletileri denetlemek için düzenli aralıklarla yeniden taranır.

Bu nedenlerle, DLQ 'nun olabildiğince az sayıda ileti içerdiğinden emin olmaya çalışın; atılmayan ya da başka kuyruklara iletilen iletiler (herhangi bir nedenle) kuyrukta birikmesine izin verilirse, DLQ işleyicisinin iş yükü artar ve DLQ' nun kendisi doldurabilir.

DLQ işleyicisini etkinleştirmek için DLQ ' yı (DLQ) boşaltmak için belirli önlemleri alın. Örneğin, DLQ ' da ileti bırakan ACTION (IGNORE) kullanmayın. (Tablodaki diğer kurallar tarafından açık bir şekilde ele alınmayan iletiler için ACTION (IGNORE) varsayıldığını unutmayın.) Bunun yerine, başka bir şekilde yoksaymak istediğiniz iletiler için, iletileri başka bir kuyruğa taşıyan bir işlem kullanın. Örneğin:

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

Benzer şekilde, tabloda daha önceki kurallar tarafından ele alınmamış iletileri işlemek için bir catchall tablolarında son kuralı yapın. Örneğin, tabloda son kural şöyle bir şey olabilir:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Bu ileti, çizelgedeki son kurala kadar, el ile işlenebileceği REALLY . DEAD . QUEUEkuyruğuna iletilecek iletiler iletir. Böyle bir kuralınız yoksa, iletilerin DLQ ' da süresiz olarak kalabileceğini kabul edin.

Bir DLQ işleyicisi kural çizelgesi örneği

Tek bir denetim ögesi girişi ve birkaç kural içeren runmqdlq komutu için bir örnek kural çizelgesi.

```
*****
*           An example rules table for the runmqdlq command           *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* runmqdlq, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to runmqdlq,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.
* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation are always sending messages with incorrect
```

```
* addresses. When we find a request from the AAAA corporation,  
* we return it to the DLQ (DEADQ) of the reply-to queue manager  
* (&REPLYQM).  
* The AAAA DLQ handler attempts to redirect the message.
```

```
MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +  
  ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)
```

```
* The BBBB corporation never do things by half measures. If  
* the queue manager BBBB.1 is unavailable, try to  
* send the message to BBBB.2
```

```
DESTQM(bbbb.1) +  
  action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)
```

```
* The CCCC corporation considers itself very security  
* conscious, and believes that none of its messages  
* will ever end up on one of our DLQs.  
* Whenever we see a message from a CCCC queue manager on our  
* DLQ, we send it to a special destination in the CCCC organization  
* where the problem is investigated.
```

```
REPLYQM(CCCC.*) +  
  ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)
```

```
* Messages that are not persistent run the risk of being  
* lost when a queue manager terminates. If an application  
* is sending nonpersistent messages, it should be able  
* to cope with the message being lost, so we can afford to  
* discard the message. PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)  
* For performance and efficiency reasons, we like to keep  
* the number of messages on the DLQ small.  
* If we receive a message that has not been processed by  
* an earlier rule in the table, we assume that it  
* requires manual intervention to resolve the problem.  
* Some problems are best solved at the node where the  
* problem was detected, and others are best solved where  
* the message originated. We don't have the message origin,  
* but we can use the REPLYQM to identify a node that has  
* some interest in this message.  
* Attempt to put the message onto a manual intervention  
* queue at the appropriate node. If this fails,  
* put the message on the manual intervention queue at  
* this node.
```

```
REPLYQM('?*') +  
  ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)
```

```
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)
```

Birden çok kuruluş

UNIX, Linux, and Windows işletim sisteminde, bir sistemde birden çok IBM WebSphere MQ kopyası bulunmanız mümkündür.

Her bir IBM WebSphere MQ kopyasının kurulu olduğu yeri seçebilirsiniz, ancak her bir kopyanın ayrı bir kuruluş konumunda olması gerekir. A maximum of 128 installations of IBM WebSphere MQ can exist on a single machine at a time. One installation can be an installation of IBM WebSphere MQ Version 7.0.1 at fix pack 6, or later. Şimdi bir seçeneğiniz var:

- Bir makineden tek bir IBM WebSphere MQ kuruluşunu sürdürme ve yönetme basitliğini korur.
- Birden çok IBM WebSphere MQ kuruluşu etkinleştirilerek sunulan esneklikten yararlanın.

Birden çok IBM WebSphere MQ kopyasını kurmadan önce, birkaç karar vermelisiniz:

- Sistemde IBM WebSphere MQ Version 7.0.1 kopyası olacak mı?

When IBM WebSphere MQ Version 7.0.1 at fix pack 6, or later, is installed on the system, there are a number of restrictions to consider:

- UNIX and Linux sistemlerinde, IBM WebSphere MQ Version 7.0.1 , varsayılan konuma kurulmalıdır.

- IBM WebSphere MQ Version 7.0.1 must be the first installation on a system. You cannot install IBM WebSphere MQ Version 7.0.1 after installing version 7.1, or later. 7.0.1sürümünü kaldırırsanız, WebSphere MQ ürününün sonraki bir sürümü kurulu olduğunda bu sürüm yeniden kurulamaz.
- IBM WebSphere MQ Version 7.0.1 , otomatik olarak birincil kurulumdur. IBM WebSphere MQ Version 7.0.1 kurulu olduğunda birincil kurulum olarak başka bir kurulum seçemezsiniz.
- IBM WebSphere MQ' in her bir kopyasını nereye kuracaksınız?
You can choose the installation location for your installations at version 7.1, or later. Daha fazla bilgi için [Kurulum konumunu seçme](#) başlıklı konuya bakın.
- Birincil kuruluşa gereksiniminiz var mı?
Birincil kurulum, sistem genelindeki konumların başvurusu için kullanılan bir kurulumdur. Daha fazla bilgi için bkz [“Birincil kurulum seçilmesi” sayfa 116](#).
- Uygulamalarınız nasıl bağlanacak?
Uygulamalarınızın uygun IBM WebSphere MQ kitaplıklarını nasıl bulacağını göz önünde bulundurmanız gerekir. Daha fazla bilgi için bakınız: [Connecting applications in a multi installation environment, and Connecting .NET applications in a multiple installation environment](#).
- Var olan çıkışlarınızın değişmesi gerekiyor mu?
IBM WebSphere MQ varsayılan konuma kurulmamışsa, çıkışlarınızın güncellenmesi gerekir. Ek bilgi için [Yazma ve derleme çıkışları ve kurulabilir hizmetler](#) başlıklı konuya bakın.
- Hangi kuyruk yöneticisinin hangi kurulum ile ilişkilendirilecek?
Her kuyruk yöneticisi belirli bir kurulumla ilişkilendirilir. Kuyruk yöneticisinin ilişkilendirildiği kurulum, kuyruk yöneticisini yalnızca o kurulumdan gelen komutlarla denetleyebilecek şekilde sınırlarla ilişkilendirir. Daha fazla bilgi için bkz. [Bir kuyruk yöneticisinin kurulumla ilişkilendiriliyor](#).
- Ortamınızı her kurulumla nasıl çalışacak şekilde ayarlayacaksınız?
Bir sistemde birden çok kurulum ile, belirli kurulumlarla nasıl çalışacağını ve bu kurulumdan nasıl komut göndereceğini göz önünde bulundurmanız gerekir. Komutun tam yolunu belirleyebilir ya da ortam değişkenlerini ayarlamak için **setmqenv** ya da **crtmqenv** komutunu kullanabilirsiniz. Ortam değişkenlerinin ayarlanması, o kurulumla ilişkin komutların yolunu atmanıza olanak sağlar. Ek bilgi için bkz. [setmqenvve crtmqenv](#).

When you have answered these questions, you can install IBM WebSphere MQ using the steps provided in [IBM WebSphere MQkuruluyor](#).

Varolan IBM WebSphere MQ kurulumlarında ve birden çok kurulum yeteneğini kullanarak IBM WebSphere MQ ' un bir sürümünden başka bir sürüme geçiş yapmak istiyorsanız bkz. [UNIX, Linuxve Windowsüzerinde çok kurulumlu kuyruk yöneticisi var](#) .

.NET destek paketi ve birden çok kurulum için IBM ileti hizmeti istemcisi

Birden çok sürüm desteği için, *Java ve .NET Messaging and Web Services* özelliğinin IBM WebSphere MQ ürünüyle birlikte kurulması gerekir. Bu özellik, *IBM Message Service Client for .NET* destek paketi (IA9H) içinde yer alan tüm işlevleri içerir. Destek paketi sisteme kurulduysa, birden çok sürüm desteklenmez. IBM WebSphere MQürününü kurmadan önce destek paketini kaldırmanız gerekir. .NET özelliğinin kurulmasıyla ilgili daha fazla bilgi için bakınız: [Installing WebSphere MQ classic for .NET](#).

İlgili kavramlar

[UNIX, Linuxve Windows: 7.0.1 sürümünden 7.5sürümüne geçiş için yan yana geçiş](#)

[UNIX, Linuxve Windows: 7.0.1 sürümünden 7.5sürümüne çok aşamalı geçiş](#).

İlgili görevler

[Birden çok kurulumu yapılandırma](#)

[Bir sistemde WebSphere MQ ' nun kurulumları bulunuyor](#)

Birincil kuruluş seçilmesi

Birden çok IBM WebSphere MQ (UNIX, Linux, and Windows) kuruluşunu destekleyen sistemlerde, birincil kuruluş, IBM WebSphere MQ sistem genelinde yer alan locations konumlarına bağlı olan sistemdir. Birincil kuruluma sahip olmak isteğe bağlıdır, ancak uygun olur.

IBM WebSphere MQ Version 7.1' den önce, ürünün yalnızca bir eşgörünümü herhangi bir zamanda kurulabilir. Windows sistemlerinde, bazı genel ortam değişkenleri o kuruluşu işaret edecek şekilde ayarlandı. UNIX and Linux sistemlerinde, simgesel bağlantılar /usr/lib, /usr/bin ve /usr/include' ye eklenerek tek bir kuruluşa da işaret edilir.

Version 7.1'ten, UNIX, Linux, and Windows' ta birden çok IBM WebSphere MQ sürümünü kurabilirsiniz. Bu sistemlerden birinde, herhangi bir zamanda birden çok IBM WebSphere MQ kuruluşu olması ve isteğe bağlı olarak, bu kuruluşlardan birini birincil kuruluş olarak yapılandırmak mümkün olabilir. Birden çok sürüm olduğunda, tek bir kuruluşa işaret eden ortam değişkenleri ve simgesel bağlantılar daha az anlamlı olur. Ancak, bazı işlevler bu sistem genelindeki konumların çalışabilmelerini gerektirir. Örneğin, IBM WebSphere MQ' i ve üçüncü kişi ürünlerini denetlemek için özel kullanıcı komut dosyaları. Bu işlevler yalnızca birincil kuruluma çalışır.

UNIX and Linux sistemlerinde, bir kuruluşu birincil kuruluş olarak ayarlıyorsanız, bu kuruluşun dış kitaplıklarına ve denetim komutlarına ilişkin simgesel bağlantılar, /usr/lib ve /usr/bin olarak eklenir. Birincil kuruluma sahip değilseniz, simgesel bağlantılar yaratılmaz. Birincil kuruluşla ilgili olarak yapılan simgesel bağlantıların bir listesi için bkz. [Dış kitaplık ve denetim komutu bağlantıları UNIX and Linux üzerinde birincil kuruluşa bağlantılar.](#)

Windows sistemlerinde, genel ortam değişkenleri birincil kuruluşun kurulu olduğu dizinlere işaret eder. Bu ortam değişkenleri, IBM WebSphere MQ kitaplıklarını, denetim komutlarını ve üstbilgi dosyalarını bulmak için kullanılır. Ayrıca, Windows sistemlerinde, işletim sisteminin bazı özellikleri, daha sonra tek bir sürece yüklenen arabirim kitaplıklarının merkezi olarak kaydedilmesini gerektirir. With multiple versions of IBM WebSphere MQ, there would be conflicting sets of IBM WebSphere MQ libraries. Özellikler, bu çakışan kitaplık kümelerini tek bir sürece yüklemeye çalışacaktır. Bu nedenle, bu özellikler yalnızca birincil kuruluşla kullanılabilir. Birincil kuruluşla sınırlı olarak sınırlanmış bazı özelliklere ilişkin ayrıntılar için bkz. [Features that can be used only with the primary installation on Windows.](#)

Sistemde IBM WebSphere MQ Version 7.0.1 kurulumunuz varsa, bu kuruluş otomatik olarak birincil kurulumdur. Birincil kuruluş, Version 7.0.1 kurulu durumdayken değiştirilemez. Sistemdeki tüm kuruluşlar Version 7.1'ya da sonraki bir düzeye, birincil kuruluma sahip olup olmadığınızı seçebilirsiniz. Consider the options in [Çizelge 11 sayfa 116.](#)

Çizelge 11. Birincil kuruluş seçenekleri.

Bu çizelge, birincil kuruluşlara ilişkin geçerli kuruluş yapılandırmalarını gösterir. Tek bir Version 7.1 ile ya da daha sonraki bir sürümü ile birincil ya da birincil olmayan bir değer olabilir. Birden çok kuruluş (birden çok kuruluş), Version 7.0.1 ve bir ya da daha çok Version 7.1 konumunda ya da daha sonra, Version 7.0.1 birincil kullanıcı ve diğer kuruluşlar birincil kullanıcı olmamalıdır. Version 7.1'ya da sonraki düzeylerde birden çok kuruluş ile, bir kuruluş birincil ya da birincil dışındaki tüm kuruluşlar olabilir.

Seçenekler	Geçerli kuruluş yapılandırmaları		Daha fazla bilgi
	Birincil	Birincil olmayan	

Çizelge 11. Birincil kuruluş seçenekleri.

Bu çizelge, birincil kuruluşlara ilişkin geçerli kuruluş yapıları gösterir. Tek bir Version 7.1 ile ya da daha sonraki bir sürümü ile birincil ya da birincil olmayan bir değer olabilir. Birden çok kuruluş (birden çok kuruluş), Version 7.0.1 ve bir ya da daha çok Version 7.1 konumunda ya da daha sonra, Version 7.0.1 birincil kullanıcı ve diğer kuruluşlar birincil kullanıcı olmamalıdır. Version 7.1 ya da sonraki düzeylerde birden çok kuruluş ile, bir kuruluş birincil ya da birincil dışındaki tüm kuruluşlar olabilir.

(devamı var)

Seçenekler	Geçerli kuruluş yapılandırmaları	Daha fazla bilgi	
Version 7.1' in tek kuruluşu ya da daha sonraki bir sürümü.	Version 7.1 ya da sonraki bir sürümü.	Yok	Önceki sürümlerle aynı şekilde tek bir kuruluşla çalışmaya devam etmek istiyorsanız, birincil kuruluş olarak kuruluşunuzu yapılandırın. Bu seçenekle ilgili bilgi için bkz. Tek kurulum (IBM WebSphere MQ Version 7.1) ya da daha sonraki bir sürümü, birincil kuruluş olarak yapılandırıldı
	Yok	Version 7.1 ya da sonraki bir sürümü.	Tek bir kuruluşla çalışmaya devam etmek istiyorsanız, ancak sizin için oluşturulan simgesel bağlantılar ya da genel ortam değişkenleri istemezseniz, kuruluşunuzu birincil kullanıcı olarak yapılandırın. Bu seçeneğin etkileri hakkında bilgi için bkz. Tek kurulum (IBM WebSphere MQ Version 7.1) veya daha sonraki bir sürümü, birincil kullanıcı olarak yapılandırılmış
Birden çok kuruluş: Version 7.0.1 ve Version 7.1 ya da sonraki bir sürümü.	Version 7.0.1	Version 7.1 ya da sonraki bir sürümü.	If you want to have multiple installations of IBM WebSphere MQ, with one at version 7.0.1, the version 7.0.1 installation is automatically the primary installation. IBM WebSphere MQ sürümü 7.0.1 kurulu olduğunda, birincil kuruluş olan kuruluşu değiştiremezsiniz. Bu seçenek ve bunun etkileri hakkında bilgi için bkz. Multiple installats of IBM WebSphere MQ, one at Version 7.0.1
Birden çok kuruluş: Version 7.1 ya da sonraki bir sürümü.	Version 7.1 ya da sonraki bir sürümü.	Version 7.1 ya da sonraki bir sürümü.	If you want to have multiple installations of WebSphere MQ at version 7.1 or greater, you can choose whether to make one of the installations primary. Bu seçenekle ilgili bilgi için bkz. Birden çok IBM WebSphere MQ Version 7.1 kuruluşu, ya da daha sonraki bir sürümü
	Yok	Version 7.1 ya da sonraki bir sürümü.	

İlgili kavramlar

[Birincil kuruluş olarak yapılandırılan WebSphere MQ Version 7.1 ya da sonraki bir sürümünün tek kuruluşu](#)
[Tek bir WebSphere MQ Sürüm 7.1 kuruluşu ya da daha sonraki bir sürümü, birincil kullanıcı olarak yapılandırıldığı şekilde](#)

[Birden çok WebSphere MQ Sürüm 7.1 kuruluşu ya da üstü](#)

[Birden çok WebSphere MQ kuruluşu, biri sürüm 7.0.1' de](#)

İlgili görevler

[Birincil kuruluş değiştiriliyor](#)

[Kuruluş yerinin seçilmesi](#)
[Kuruluşunuzu planlama](#)
[Kuruluş adı seçilmesi](#)

Single installation of IBM WebSphere MQ Version 7.1, or later, configured as the primary installation

Bir IBM WebSphere MQ kuruluşunun birincil olarak imlenmesi, simgesel bağlantılar ya da genel ortam değişkenleri tarafından sisteme eklenir; böylece, uygulamalar tarafından kullanılan IBM WebSphere MQ komutları ve kitaplıkları, en az sistem kuruluşu için gerekli olan en az sistem ayarlarıyla otomatik olarak kullanılabilir.

IBM WebSphere MQ' u nereye kuracağınıza karar veriniz.

Where possible, configure applications and scripts to use the system search path to find the IBM WebSphere MQ control commands or IBM WebSphere MQ libraries. Uygulamaların ve komut kütüklerinin bu yapılandırması, sonraki IBM WebSphere MQ yayınına geçiş ya da ikinci bir kuruluş kurulması gibi gelecekteki görevlerin üstlenilmesine ilişkin en yüksek düzeyde esneklik sağlar. Uygulamalarınızı bağlamaya ilişkin seçenekler hakkında daha fazla bilgi için bkz. [Birden çok kuruluş ortamında uygulamaları bağlama](#).

Windows 'ta, ilk kuruluş otomatik olarak birincil kuruluş olarak yapılandırılır. UNIX and Linux platformlarında, bir sistemin ilk kuruluşu birincil kuruluş olacak şekilde el ile yapılandırılmalıdır. **setmqinst** komutunu kullanarak birincil kuruluşu ayarlayın. Daha fazla bilgi için [Birincil kuruluşun kaldırılması, büyütülmesi ve bakımının yapılması](#) başlıklı konuya bakın.

İlgili görevler

[Birincil kuruluş değiştiriliyor](#)
[Kuruluş yerinin seçilmesi](#)
[Kuruluşunuzu planlama](#)
[Kuruluş adı seçilmesi](#)

Single installation of IBM WebSphere MQ Version 7.1, or later, configured as non-primary

If you install IBM WebSphere MQ Version 7.1, or later, as non-primary you might have to configure a library path for applications to load IBM WebSphere MQ libraries. Windows 'ta, bazı ürün yetenekleri yalnızca IBM WebSphere MQ birincil olarak yapılandırıldığında kullanılabilir.

UNIX ve Linux sistemleri

The implications of running a non-primary installation on UNIX and Linux are:

- Gömülü kitaplık yolunu kullanarak IBM WebSphere MQ kitaplıklarını (örneğin, RPATH) konumlayan uygulamalar, aşağıdaki koşullar doğru olduğunda bu kitaplıkları bulamazlar:
 - IBM WebSphere MQ , RPATH' de belirtilen dizinden farklı bir dizine kurulur.
 - /usr içinde simgesel bağlantı yok
- Uygulamalar dış kitaplık yolunu kullanarak kitaplıklarını bulurlar; örneğin, LD_LIBRARY_PATH, dış kitaplık yolunu MQ_INSTALLATION_PATH/lib ya da MQ_INSTALLATION_PATH/lib64 dizinini içerecek şekilde yapılandırmanız gerekir. **setmqenv** ve **crtmqenv** komutları, dış kitaplık yolu da içinde olmak üzere, geçerli kabukta bir dizi ortam değişkeni yapılandırabilir.
- Çoğu IBM WebSphere MQ işlemi setuid/setgid olarak çalışır. Sonuç olarak, kullanıcı çıkışlarını yüklerken dış kitaplık yolunu yoksayabilirsiniz. User exits that reference IBM WebSphere MQ libraries can find those libraries only if they are found in the library path embedded within them. /usr' ta simgesel bir bağlantı varsa bunlar çözülür. IBM WebSphere MQ Version 7.1 üzerinde çalıştırılmak istenen kullanıcı çıkışları ya da daha sonra, artık IBM WebSphere MQ kitaplıklarına başvurmamaları için oluşturulabilirler. Bunun yerine, işlevin çalışabileceği IBM WebSphere MQ işlevlerine işlev göstergelerinden geçmek için

IBM WebSphere MQ ' e güvenirlir. Ek bilgi için [Yazma ve derleme çıkışları ve kurulabilir hizmetler](#) başlıklı konuya bakın.

Uygulamalarınızı bağlamaya ilişkin seçenekler hakkında daha fazla bilgi için bkz. [Birden çok kuruluş ortamında uygulamaları bağlama](#).

UNIX and Linux platformlarında, bir sistemin ilk kuruluşu otomatik olarak birincil kuruluş olarak yapılandırılmaz. Ancak, **dspmqr** komutunun yerini bulmak için /usr/bin içinde tek bir sembolik bağlantı bulunur. Simgesel bağlantı olmasını istemiyorsanız, aşağıdaki komutu kullanarak bu bağlantıyı kaldırmalısınız:

```
setmqinst -x -p MQ_INSTALLATION_PATH
```

Windows sistemleri

The implications of running a non-primary installation on Windows are:

- Applications normally find their libraries using the external library path, PATH. Gömülü kitaplık yolu ya da belirtik kitaplık konumu kavramı yoktur. Kuruluş birincil dışıysa, genel PATH ortam değişkeni IBM WebSphere MQ kuruluş dizinini içermez. For applications to find IBM WebSphere MQ libraries, update the PATH environment variable to reference the IBM WebSphere MQ installation directory. **setmqenv** ve **crtmqenv** komutları, dış kitaplık yolu da içinde olmak üzere, geçerli kabukta bir dizi ortam değişkeni yapılandırabilir.
- Bazı ürün yetenekleri yalnızca bir kuruluş birincil kuruluş olarak yapılandırıldığında kullanılabilir; bkz. [Yalnızca Windows üzerinde birincil kuruluşla birlikte kullanılacak özellikler](#).

Varsayılan olarak, Windows 'ta ilk kuruluş otomatik olarak birincil olarak yapılandırılır. Birincil kuruluş olarak el ile seçimi kaldırmanız gerekir.

İlgili görevler

[Birincil kuruluş değiştiriliyor](#)

[Kuruluş yerinin seçilmesi](#)

[Kuruluşunuzu planlama](#)

[Kuruluş adı seçilmesi](#)

İlgili başvurular

[setmqenv](#)

[crtmqenv](#)

Birden çok IBM WebSphere MQ Version 7.1 kuruluşu ya da daha sonraki bir sürümü

Birincil kuruluş olarak yapılandırılan IBM WebSphere MQ Version 7.1 ya da sonraki kuruluşlardan birine sahip olmak için bu seçeneği belirleyebilirsiniz. Seçiminiz, uygulamaların kitaplıklarının yerini saptamasına bağlıdır.

IBM WebSphere MQ Version 7.1 ile birlikte gönderilen mqm gibi IBM WebSphere MQ kitaplıkları, bağlantı kurdukları kuyruk yöneticisinin gerektirdiği düzeyin kitaplıklarını otomatik olarak kullanır. Bu, bir uygulamanın IBM WebSphere MQ kitaplıklarını bir IBM WebSphere MQ Version 7.1 kuruluşundan ayırdığı anlamına gelir; bu, o sistemdeki herhangi bir kuyruk yöneticisine bağlanabiliyor. Birincil olarak yapılandırılmış bir IBM WebSphere MQ Version 7.1 kuruluşunun, uygulamanın IBM WebSphere MQ arabirim kitaplığını bulmasını güvenceye almak durumunda, uygulamanın herhangi bir kuyruk yöneticisine bağlanabilmesini sağlar.

Uygulamaların birden çok kuruluş ortamında bağlanmaya ilişkin ek bilgi edinmek için [Birden çok kuruluş ortamında uygulamaların bağlanması](#) başlıklı konuya bakın.

Birincil kuruluşu kaldırdığınızda, birincil kuruluş otomatik olarak değiştirilmez. If you want another installation to be the primary installation, you must manually set the primary installation using the **setmqinst** command. Daha fazla bilgi için [Birincil kuruluşun kaldırılması, büyütülmesi ve bakımının yapılması](#) başlıklı konuya bakın.

İlgili kavramlar

[Birden çok kuruluş](#)

İlgili görevler

[Birincil kuruluş değiştiriliyor](#)

[Kuruluş yerinin seçilmesi](#)

[Kuruluşunuzu planlama](#)

[Kuruluş adı seçilmesi](#)

Birden çok IBM WebSphere MQ kuruluşu, biri Version 7.0.1

IBM WebSphere MQ Version 7.1 ya da daha sonraki bir sürümü, bazı sınırlamalarla birlikte IBM WebSphere MQ Version 7.0.1 ile birlikte bulunabilir.

- UNIX and Linux sistemlerinde, Version 7.0.1 yalnızca sabit, varsayılan bir konuma kurulabilir; bu nedenle, Version 7.1 ürününü ya da sonraki bir varsayılan yerinde kuramazsınız.
- IBM WebSphere MQ Version 7.0.1, otomatik olarak birincil kuruluş olarak yapılandırılır. UNIX and Linux sistemlerinde, simgesel bağlantılar otomatik olarak uygun IBM WebSphere MQ dizinlerine oluşturulur. Windows 'ta, sağlanan ürünün genel olarak kayıtlı olduğu her şey. IBM WebSphere MQ Version 7.0.1 bu şekilde çalışmak üzere kurulmuş olmalıdır. Burada IBM WebSphere MQ Version 7.0.1 kurulu, bir IBM WebSphere MQ Version 7.1 ya da sonraki bir kuruluş birincil olarak kurulamaz.

The libraries from IBM WebSphere MQ Version 7.1, or later, can work with any queue manager running under IBM WebSphere MQ Version 7.0.1, or later. Bir uygulamanın, daha sonraki sürümlerin yanı sıra Version 7.0.1 altında çalışan kuyruk yöneticilerine bağlanması gerekiyorsa, aşağıdaki koşullar doğru olursa olağan şekilde çalışmaya devam edebilir.

- Çalıştırma zamanında kitaplıklar IBM WebSphere MQ Version 7.1 ya da sonraki bir sürümü bulur.
- Yalnızca Version 7.0.1 içinde kullanılabilir olan işlevleri kullanır.

Uygulamaların birden çok kuruluş ortamında bağlanmaya ilişkin ek bilgi edinmek için [Birden çok kuruluş ortamında uygulamaların bağlanması](#) başlıklı konuya bakın.

IBM WebSphere MQ Version 7.0.1 ürününü kaldırdığınızda birincil kuruluş otomatik olarak değiştirilmez.

If you want another installation to be the primary installation, you must manually set the primary installation using the **setmqinst** command. Daha fazla bilgi için [Birincil kuruluşun kaldırılması, büyütülmesi ve bakımının yapılması](#) başlıklı konuya bakın.

İlgili kavramlar

[Birden çok kuruluş](#)

İlgili görevler

[Kuruluş yerinin seçilmesi](#)

[Kuruluşunuzu planlama](#)

[Kuruluş adı seçilmesi](#)

Depolama ve performans gereksinimlerinizin planlanması

IBM WebSphere MQ sisteminiz için gerçekçi ve ulaşılabilir depolama ve performans hedefleri ayarlamanız gerekir. Altyapınızdaki depolama ve performansı etkileyen etmenler hakkında bilgi almak için bağlantıları kullanın.

Gereksinimler, IBM WebSphere MQ ' u ve kullanmak istediğiniz bileşenleri kullanarak olduğunuz sistemlere bağlı olarak değişir.

Desteklenen donanım ve yazılım ortamlarıyla ilgili en son bilgiler için [IBM WebSphere MQ için Sistem Gereksinimleri](#) Web sitesine bakın:

www.ibm.com/software/integration/wmq/requirements/

IBM WebSphere MQ , kuyruk yöneticisi verilerini dosya sistemindeki depolar. IBM WebSphere MQ ile kullanmak üzere dizin yapılarını planlama ve yapılandırma hakkında bilgi edinmek için aşağıdaki bağlantıları kullanın:

- [“Planlama dosya sistemi desteği” sayfa 122](#)
- [“Paylaşılan dosya sistemlerine ilişkin gereksinimler” sayfa 123](#)
- [“Sharing IBM WebSphere MQ files” sayfa 132](#)
- [“UNIX and Linux sistemlerindeki dizin yapısı” sayfa 134](#)
- [“Windows sistemlerindeki dizin yapısı” sayfa 144](#)

Use the following links for information about system resources, shared memory, and process priority on UNIX and Linux:

- [“IBM WebSphere MQ ve UNIX System V IPC kaynakları” sayfa 147](#)
- [“ AIX üzerinde paylaşılan bellek” sayfa 148](#)
- [“ WebSphere MQ ve UNIX Process Priority” sayfa 148](#)

İlgili kavramlar

[“Planlama” sayfa 5](#)

IBM WebSphere MQ ortamınızı planlarken, yapılandırmak istediğiniz IBM WebSphere MQ mimarisini, kaynak gereksinmelerini, günlüğe kaydetme gereksinmesini ve yedekleme olanaklarını göz önünde bulundurmanız gerekir. Use the links in this topic to plan the environment where IBM WebSphere MQ runs.

[“IBM WebSphere MQ mimarisinin tasarlanması” sayfa 14](#)

Noktadan noktaya iletişim ve yayınlama/abone olma ileti stilleri için IBM WebSphere MQ tarafından desteklenen farklı mimariler hakkında bilgi edinin.

[UNIX ve Linux üzerinde donanım ve yazılım gereksinimleri](#)

[Windows üzerinde donanım ve yazılım gereksinimleri](#)

Disk alanı gereksinimleri

WebSphere MQ için depolama gereksinimleri, hangi bileşenlere kurduğunuz ve ne kadar çalışma alanı gereksiniminiz olduğunu bağlıdır.



Kurulmasını istediğiniz önkoşul bileşenleri de içinde olmak üzere, kurmak üzere seçtiğiniz isteğe bağlı bileşenler için disk saklama alanı gereklidir. Toplam depolama gereksinimi, kullandığınız kuyruk sayısına, kuyruklardaki iletilerin sayısını ve boyutunu ve iletilerin kalıcı olup olmadığını da bağlıdır. Ayrıca, kendi uygulama programlarınız için, disk, manyetik bant ya da diğer ortamlardaki arşivleme kapasitesinin yanı sıra, alan için de yer alan bir alan olmasını da gerektirir.

Aşağıdaki çizelgede, ürünün çeşitli birleşimlerini farklı platformlarda kurduğunuzda gereken yaklaşık disk alanı gösterilir. (Değerler en yakın 5 MB 'ye kadar yuvarlanır; burada bir MB 1.048.576 bayttır.)

Altyapı	İstemci kuruluşu ¹	Sunucu kuruluşu ²	WebSphere MQ MFT kuruluşu ³	Tam kuruluş ⁴
AIX	145 MB	190 MB	705 MB	915 MB
HP-UX	225 MB	310 MB	1075 MB	1340 MB
IBM i	215 MB	450 MB	80 MB	655 MB
System x içinLinux (32 bit)	85 MB	N/A	N/A	120 MB
System x içinLinux (64 bit)	125 MB	170 MB	575 MB	935 MB
Linux on POWER Systems-Big Endian	130 MB	170 MB	565 MB	715 MB

Altyapı	İstemci kuruluđu 1	Sunucu kuruluđu 2	WebSphere MQ MFT kuruluđu 3	Tam kuruluđu 4
Solaris x86-64, AMD64, EM64Tve uyumlu işlemciler	105 MB 5	150 MB 5	695 MB	860 MB
Solaris SPARC	105 MB 5	150 MB 5	680 MB	820 MB
Windows (32 bit kuruluđu) 6	390 MB	N/A	N/A	475 MB
Windows (64 bit kuruluđu) 6	445 MB	555 MB	710 MB	1005 MB

Kullanım notları

- İstemci kuruluđu aşağıdaki bileşenleri içerir:
 - Yürütme Ortamı
 - İstemci
- Sunucu kuruluđu aşağıdaki bileşenleri içerir:
 - Yürütme Ortamı
 - Sunucu
- IBM WebSphere MQ Managed File Transfer kuruluđu aşağıdaki bileşenleri içerir:
 - IBM WebSphere MQ Managed File Transfer Service, Logger, Agent, Tools ve Base bileşenleri
 - Yürütme Ortamı
 - Sunucu
 - Java
 - JRE
- Tam kuruluđu, kullanılabilir tüm bileşenleri içerir.
-  Solaris platformlarında, bu bileşen bileşimini almak için sessiz bir şekilde kurmanız gerekir.
-  Burada listelenen bileşenlerin tümü Windows sistemlerindeki kurulabilir özelliklerdir; işlevleri bazen diğer özelliklerde de yer alır. Bkz. [Windows sistemleri için WebSphere MQ özellikleri](#).

İlgili görevler

[Kurulacağı öğe seçilmesi](#)

Planlama dosya sistemi desteği

Kuyruk yöneticisi verileri dosya sisteminde depolanır. Kuyruk yöneticisi, çok eşgörünümli bir kuyruk yöneticisinin birden çok örneğinin aynı anda etkin olmasını önlemek için dosya sistemi kilitlemesini kullanır.

Paylaşılan kütük sistemleri

Paylaşılan kütük sistemleri, birden çok sistemin aynı anda aynı fiziksel depolama aygıtına erişmesine olanak sağlar. Birden çok sistem aynı fiziksel depolama aygıtına doğrudan kilitleme ve eşzamanlılık denetimi araçları olmaksızın doğrudan erişirse, bozulma meydana gelir. İşletim sistemleri, yerel süreçler için kilitleme ve eşzamanlılık denetimi ile yerel dosya sistemleri sağlar; ağ dosya sistemleri dağıtılmış sistemler için kilitleme ve eşzamanlılık denetimi sağlar.

Geçmişteki ağ dosya sistemleri, iletileri günlüğe kaydetme gereksinimlerini karşılamak için yeterince hızlı bir şekilde gerçekleştirilmemiş veya yeterli kilitleme ve eşzamanlılık denetimi sağlamış olmalıdır.

Günümüzde, ağ üzerinde çalışan dosya sistemleri iyi performans sağlayabilir ve *RFC 3530, Network File System (NFS) sürüm 4 iletişim kuralı* gibi güvenilir ağ dosya sistemi iletişim kuralları uygulamaları, iletileri güvenilir bir şekilde günlüğe kaydetmeye ilişkin gereksinimleri karşılayabilir.

Paylaşılan kütük sistemleri ve WebSphere MQ

Çok eşgörünümlü bir kuyruk yöneticisine ilişkin kuyruk yöneticisi verileri paylaşılan bir ağ dosyası sisteminde saklanır. Microsoft Windows, UNIX and Linux sistemlerinde, kuyruk yöneticisinin veri dosyaları ve günlük dosyaları paylaşılan ağ dosya sistemine yerleştirilmelidir.

Prior to release v7.0.1, WebSphere MQ does not support queue manager data stored on networked storage accessed as a shared file system. Kuyruk yöneticisi verileri paylaşılan ağ depolamaya yerleştirilmişse, kuyruk yöneticisi verilerinin aynı anda çalışan başka bir kuyruk yöneticisinin başka bir yönetim ortamı tarafından erişilmediğinden emin olmanız gerekir.

From v7.0.1 onwards, WebSphere MQ uses locking to prevent multiple instances of the same multi-instance queue manager being active at the same time. Aynı kilitleme, iki ayrı kuyruk yöneticisinin yanlışlıkla aynı kuyruk yöneticisi veri dosyaları kümesini kullanmasını da sağlar. Bir kuyruk yöneticisinin yalnızca bir eşgörünümünün kilidine bir kerede sahip olabilir. Sonuç olarak, WebSphere MQ , paylaşılan bir kütük sistemi olarak erişilen ağ üzerinde çalışan depolama sisteminde saklanan kuyruk yöneticisi verilerini destekler.

Ağ dosyası sistemlerinin tüm kilitleme protokolleri sağlam olmadığından ve bir dosya sistemi veri bütünlüğü yerine performans için yapılandırılabilirliğinden, bir ağ dosya sisteminin kuyruk yöneticisi verilerine erişimi denetleyip denetlemediğini ve doğru günlüklere erişip denetmeyeceğini test etmek için **amqmfsc** komutunu çalıştırmanız gerekir. Bu komut yalnızca UNIX ve IBM i sistemlerini uygular. Microsoft Windows üzerinde, desteklenen tek bir ağ dosya sistemi vardır ve **amqmfsc** komutu gerekli değildir.

İlgili görevler

“Paylaşılan dosya sistemi davranışı doğrulanıyor” sayfa 125

UNIX sistemlerinde paylaşılan bir kütük sisteminin, çok eşgörünümlü bir kuyruk yöneticisinin kuyruk yöneticisi verilerini saklamak için gereken gereksinimleri karşıladığını denetlemek için **amqmfsc** komutunu çalıştırın. Run the IBM WebSphere MQ MQI client sample program **amqsfhac** in parallel with **amqmfsc** to demonstrate that a queue manager maintains message integrity during a failure.

Paylaşılan dosya sistemlerine ilişkin gereksinimler

Paylaşılan dosya sistemleri, IBM WebSphere MQ ile güvenilir bir şekilde çalışmaması üzerine, veri yazma bütünlüğü, dosyalara özel erişim garantili ve kilitleri serbest bırakma olanağı sağlamalıdır.

Paylaşılan bir dosya sisteminin karşılaması gereken gereksinimler

Paylaşılan bir dosya sisteminin iletileri güvenilir bir şekilde günlüğe kaydetmek için yerine getirmeleri gereken üç temel gereksinim vardır:

1. Veri yazma bütünlüğü

Veri yazma bütünlüğüne bazen *Flush 'ta Diske Yaz* adı verilir. Kuyruk yöneticisi, fiziksel aygıtta başarıyla kesinleştirilmekte olan verilerle eşitlenebilmelidir. İşlemsel bir sistemde, diğer işleme devam etmeden önce bazı yazma işlemlerinin güvenli bir şekilde kesinleştirildiğine emin olmanız gerekir.

More specifically, IBM WebSphere MQ on UNIX platforms uses the *O_SYNC* open option and the *fsync()* system call to explicitly force writes to recoverable media, and is dependent upon these options operating correctly.



Uyarı: **Linux** Kütük sisteminin async seçeneğiyle sisteme bağlamalısınız; bu seçenek, zamanuyumlu yazma seçenekleri seçeneğini de destekler ve senkronize seçeneğinden daha iyi başarımlar sağlar.

Ancak, dosya sistemi Linux' den dışa aktarıldıysa, dosya sistemini eşitleme seçeneğini kullanarak dışa aktarmanız gerektiğini unutmayın.

2. Dosyalara özel erişim garantili

Birden çok kuyruk yöneticisini uyumlulaştırmak için, kuyruk yöneticisi için bir dosyada dışlayıcı kilit elde etmek için bir mekanizma olması gerekir.

3. Hata üzerindeki kilitleri serbest bırak

Bir kuyruk yöneticisi başarısız olursa ya da dosya sistemiyle bir iletişim hatası varsa, kuyruk yöneticisinin kilitlenmiş kütüklerinin kilidinin açılması ve kuyruk yöneticisinin kütük sistemine yeniden bağlanmasını beklemeden diğer işlemler tarafından kullanılabilir kılınması gerekir.

IBM WebSphere MQ ' in güvenilir bir şekilde çalışması için, paylaşılan bir dosya sisteminin bu gereksinimleri karşılaması gerekir. Böyle bir işlem yoksa, çok eşgörsümlü bir kuyruk yöneticisi yapılarındaki paylaşılan kütük sistemi kullanılırken kuyruk yöneticisi verileri ve günlükleri bozulabilir.

Microsoft Windows üzerinde çok eşgörsümlü kuyruk yöneticileri için ağa bağlı depolama alanına Microsoft Windows ağları tarafından kullanılan Common Internet File System (CIFS) iletişim kuralı tarafından erişilmelidir. The Common Internet File System (CIFS) client does not meet IBM WebSphere MQ's requirements for locking semantics on platforms other than Microsoft Windows, so multi-instance queue managers running on platforms other than Microsoft Windows must not use Common Internet File System (CIFS) as their shared file system.

Desteklenen diğer platformlardaki çok eşgörsümlü kuyruk yöneticileri için depolama alanına, Posix uyumlu bir ağ dosya sistemi iletişim kuralıyla erişilmesi ve kiralama tabanlı kitlemeyi desteklenmesi gerekir. Ağ Dosya Sistemi (NFS) Sürüm 4 gibi modern dosya sistemleri, hataları algılamak için kiralık kilitleri kullanın ve ardından bir hata sonrasında kilitleri serbest bırakın. Hata sonrası kilitleri serbest bırakmak için güvenilir bir mekanizmaya sahip olmayan Network File System Version 3 gibi daha eski dosya sistemleri, çok eşgörsümlü kuyruk yöneticileriyle birlikte kullanılmamalıdır.

Paylaşılan dosya sisteminin gereksinimleri karşılayıp karşılamadığını denetler.

Kullanmayı planladığınız paylaşılan dosya sisteminin bu gereksinimleri karşılayıp karşılamadığını denetlemeniz gerekir. Ayrıca, dosya sisteminin güvenilirlik için doğru yapılandırılıp yapılandırılmadığını da denetlemeniz gerekir. Paylaşılan dosya sistemleri bazen, güvenilirliğin giderileceği şekilde performansı artırmak için yapılandırma seçenekleri sağlar.

Olağan koşullarda, IBM WebSphere MQ öznitelik önbelleğe alma özelliği ile doğru biçimde çalışır ve önbelleğe almayı geçersiz kılmak gerekmez; örneğin, NFS montajı için NOAC ayarını geçersiz kılarak. Öznitelik önbelleğe alma işlemi, birden çok dosya sistemi istemcisi dosya sistemi sunucusunda aynı dosyaya yazma erişimi için çekişmeye devam ederken, her istemci tarafından kullanılan önbelleğe alınan öznitelikler sunucudaki özniteliklerle aynı olmayabilir. Bu şekilde erişilen dosyalara örnek olarak, çok eşgörsümlü bir kuyruk yöneticisi için kuyruk yöneticisi hata günlükleridir. Kuyruk yöneticisi hata günlükleri hem etkin hem de beklemedeki bir kuyruk yöneticisi yönetim ortamı tarafından yazılabilir ve önbelleğe alınan dosya öznitelikleri, dosyaların baştan başlama işlemi gerçekleşmeden önce hata günlüklerinin beklenenden daha büyük bir şekilde büyümesine neden olabilir.

Dosya sistemini denetlemenize yardımcı olması için “Paylaşılan dosya sistemi davranışı doğrulanıyor” sayfa 125 görevini çalıştırın. This task checks if your shared file system meets requirements 2 and 3. You need to verify requirement 1 in your shared file system documentation, or by experimenting with logging data to the disk.

Disk faults can cause errors when writing to disk, which IBM WebSphere MQ reports as First Failure Data Capture errors. Herhangi bir disk hatasına ilişkin paylaşılan kütük sistemini denetlemek için işletim sisteminize ilişkin dosya sistemi denetleyicisini çalıştırabilirsiniz. Örneğin, UNIX ve Linux platformlarında dosya sistemi denetimcisine fsck adı verilir. Windows altyapılarında, kütük sistemi denetimcisine CHKDSK ya da SCANDISK adı verilir.

NFS sunucusu güvenliği

Not: Yalnızca Network File System (NFS) sunucusuna kuyruk yöneticisi verileri koymanız gerekir. NFSişletim sisteminde, sistemi güvenli kılmak için takma komutuyla aşağıdaki üç seçeneği kullanın:

noexec

Bu seçeneği kullanarak, ikili dosyaların NFSüzerinde çalıştırılmasını durdurarak, uzak kullanıcının sistemde istenmeyen kod çalıştırmasını önler.

nosuid

Bu seçeneği kullanarak, uzak kullanıcının daha yüksek ayrıcalıklar kazanmasını önleyen, set-user-identifier ve set-group-identifier bitlerinin kullanılmasını engelliyorsunuz.

düğümv

Bu seçeneği kullanarak, karakter ve öbek özel aygıtlarının kullanılmasını ya da tanımlanmasını engelleyerek uzak kullanıcının bir chroot hapishanesinden çıkmasını önlemenizi sağlar.

Paylaşılan dosya sistemi davranışı doğrulanıyor

UNIX sistemlerinde paylaşılan bir kütük sisteminin, çok eşgörünümlü bir kuyruk yöneticisinin kuyruk yöneticisi verilerini saklamak için gereken gereksinimleri karşıladığını denetlemek için **amqmfscck** komutunu çalıştırın. Run the IBM WebSphere MQ MQI client sample program **amqsfhac** in parallel with **amqmfscck** to demonstrate that a queue manager maintains message integrity during a failure.

Başlamadan önce

Ağ üzerinde çalışan depolamaya sahip bir sunucuya ve WebSphere MQ ' un kurulu olduğu diğer iki sunucuya bağlı bir sunucuya gereksinim duyarsınız. Dosya sistemini yapılandırmak için yönetici (kök) yetkinizin olması ve **amqmfscck**' u çalıştırmak için bir WebSphere MQ Yöneticisi olmanız gerekir.

Bu görev hakkında

“Paylaşılan dosya sistemlerine ilişkin gereksinimler” sayfa 123 , çok eşgörünümlü kuyruk yöneticileriyle paylaşılan bir dosya sistemini kullanmaya ilişkin dosya sistemi gereksinimlerini açıklar. IBM WebSphere MQ teknik notu [WebSphere MQ çok eşgörünümlü kuyruk yöneticilerine ilişkin sınamaya ve destek bildirimini](#) , IBM ' in önceden sınanmış olduğu paylaşılan kütük sistemlerini listeler. Bu görevdeki yordam, listelenmemiş bir dosya sisteminin veri bütünlüğü sağlayıp sağlamadığını değerlendirmenize yardımcı olmak için bir dosya sisteminin nasıl sınayacağını açıklar.

Çok eşgörünümlü bir kuyruk yöneticisinin yedek sisteme geçişi, kuyruk yöneticisinin verilerine ya da günlük dosyalarına yazılmasını önleyen ağ sorunları da içinde olmak üzere, donanım ya da yazılım hatalarıyla tetiklenebilir. Genellikle, dosya sunucusunda hatalara neden olmak ilginizi çekmektedir. Ancak, kilitleri sınamak için IBM WebSphere MQ sunucularının başarısız olmasına, başarıyla serbest bırakılmasına da neden olmanız gerekir. Paylaşılan bir dosya sisteminde emin olmak için, aşağıdaki başarısızlıkları ve ortamınıza özgü diğer hataları test edin:

1. Disklerin eşitlenmesi de dahil olmak üzere dosya sunucusundaki işletim sistemi sona erdiriliyor.
2. Disklerin eşitlenmesi olmadan dosya sunucusunda işletim sistemi durduruluyor.
3. Sunucuların her birinde sıfırlama düğmesine basılması.
4. Ağ kablosunu sunucuların her birinden çekerek çıkarın.
5. Sunucuların her birinden güç kablosu çıkarılıyor.
6. Sunucuların her biri kapatılıyor.

Kuyruk yöneticisi verilerini ve günlüklerini paylaşmak için kullanacağınız ağ üzerinde çalışan depolamadaki dizini yaratın. Dizin sahibi bir WebSphere MQ Yöneticisi olmalı ya da diğer bir deyişle, UNIXüzerinde mqm grubunun bir üyesidir. Testleri çalıştıran kullanıcının WebSphere MQ Yöneticisi yetkisine sahip olması gerekir.

Dosya sistemini yapılandırmanıza yardımcı olması için [Linuxüzerinde çok eşgörünümlü bir kuyruk yöneticisi yarat](#) içindeki bir dosya sistemini dışa aktarma ve sisteme bağlama örneğini kullanın. Farklı dosya sistemleri, farklı yapılandırma adımları gerektirir. Dosya sistemi belgelerini okuyun.

Yordam

Her denetimde, dosya sistemi denetleyicisi çalışırken önceki listedeki tüm hatalara neden olur. If you intend to run **amqsfhac** at the same time as **amqmfscck**, do the task, [“İleti bütünlüğünü sınamak için amqsfhac çalıştırılıyor” sayfa 130](#) in parallel with this task.

1. Dışa aktarılan dizini iki IBM WebSphere MQ sunucusuna takın.

Dosya sistemi sunucusunda paylaşılan bir dizin (shared) ve çok eşgörümlü kuyruk yöneticilerine ilişkin verileri saklamak için bir alt dizin (qmdata) yaratın. Linuxüzerinde çok eşgörümlü kuyruk yöneticileri için paylaşılan bir dizin ayarlama örneği için, [Linuxüzerinde çok eşgörümlü kuyruk yöneticisi yaratıçindeki Örnek](#) başlıklı konuya bakın.

2. Temel dosya sistemi davranışını denetleyin.

Bir IBM WebSphere MQ sunucusunda, dosya sistemi denetleyicisini hiçbir parametre olmadan çalıştırın.

```
amqmfscck /shared/qmdata
```

Şekil 40. IBM WebSphere MQ Server 1 'de

3. Her iki IBM WebSphere MQ sunucusundan aynı dizine eşzamanlı olarak yazma işlemi olup olmadığını denetleyin.

Her iki IBM WebSphere MQ sunucusunda da, dosya sistemi denetleyicisini -c seçeneğiyle aynı anda çalıştırın.

```
amqmfscck -c /shared/qmdata
```

Şekil 41. IBM WebSphere MQ sunucusu 1 'de

```
amqmfscck -c /shared/qmdata
```

Şekil 42. IBM WebSphere MQ sunucusu 2 'de

4. Her iki IBM WebSphere MQ sunucusunda kilit bekleyen ve serbest bırakılan kilitlerin olup olmadığını denetleyin.

On both IBM WebSphere MQ servers run the file system checker at the same time with the -w option.

```
amqmfscck -w /shared/qmdata
```

Şekil 43. IBM WebSphere MQ sunucusu 1 'de

```
amqmfscck -w /shared/qmdata
```

Şekil 44. IBM WebSphere MQ sunucusu 2 'de

5. Veri bütünlüğü olup olmadığını denetleyin.

- a) Test dosyasını biçimlendirin.

Sınamakta olan dizinde büyük bir dosya yaratın. Dosya, sonraki aşamaların başarıyla tamamlanabilmesi için biçimlendirilir. Dosya, hata durumunda yedek sisteme geçmenin benzetimini

yapmak üzere ikinci aşamayı kesmek için yeterli süre olacak kadar büyük olmalıdır. 262144 sayfa (1 GB) varsayılan değerini deneyin. Program yavaş dosya sistemlerinde bu varsayılan değeri otomatik olarak azaltır, böylece biçimlendirme yaklaşık 60 saniye içinde tamamlanır.

```
amqmfscck -f /shared/qmdata
```

Sunucu aşağıdaki iletilerle yanıt verir:

```
Formatting test file for data integrity test.  
Test file formatted with 262144 pages of data.
```

Şekil 45. IBM WebSphere MQ Server 1 'de

b) Bir hataya neden olurken, dosya sistemi denetleyicisini kullanarak test dosyasına veri yazın.

Sınama programını aynı anda iki sunucu üzerinde çalıştırın. Hatayı deneyecek olan sunucuda test programını başlatın, ardından başarısızlığı atlatacağı sunucu üzerinde sınama programını başlatın. Çünkü soruşturmaya devam etmiyordun.

İlk sınama programı bir hata iletilisiyle durur. İkinci test programı, test dosyasındaki kilidi alır ve ilk test programının bıraktığı yerden başlayarak test dosyasına veri yazar. İkinci test programının tamamlanmasına izin verin.

Çizelge 12. Veri bütünlüğü denetiminin aynı anda iki sunucu üzerinde çalıştırılması

IBM WebSphere MQ Sunucu 1	IBM WebSphere MQ Sunucu 2
<pre>amqmfscck -a /shared/qmdata</pre>	
<pre>Please start this program on a second machine with the same parameters. File lock acquired. Start a second copy of this program with the same parameters on another server. Writing data into test file. To increase the effectiveness of the test, interrupt the writing by ending the process, temporarily breaking the network connection to the networked storage, rebooting the server or turning off the power. To increase the effectiveness of the test, interrupt the writing by ending the process, temporarily breaking the network connection to the networked storage, rebooting the server or turning off the power.</pre>	<pre>amqmfscck -a /shared/qmdata Waiting for lock... Waiting for lock... Waiting for lock... Waiting for lock... Waiting for lock... Waiting for lock...</pre>
Turn the power off here	

Çizelge 12. Veri bütünlüğü denetiminin aynı anda iki sunucu üzerinde çalıştırılması (devamı var)	
IBM WebSphere MQ Sunucu 1	IBM WebSphere MQ Sunucu 2
	File lock acquired. Reading test file Checking the integrity of the data read. Appending data into the test file after data already found. The test file is full of data. It is ready to be inspected for data integrity.

Testin zamanlaması, dosya sisteminin davranışına bağlıdır. Örneğin, bir dosya sisteminin, bir güç kesintisini izleyen ilk program tarafından elde edilen dosya kilitlerini serbest bırakması için genellikle 30-90 saniye sürer. İlk test programı dosyayı doldurmadan önce başarısızlığı tanıtmak için çok az zamanınız varsa, test dosyasını silmek için **amqmfscck** 'un -x seçeneğini kullanın. Daha büyük bir test dosyası ile baştan testi deneyin.

c) Test dosyasındaki verilerin bütünlüğünü doğrulayın.

```
amqmfscck -i /shared/qmdata
```

Sunucu aşağıdaki iletilerle yanıt verir:

```
File lock acquired
Reading test file checking the integrity of the data read.
The data read was consistent.
The tests on the directory completed successfully.
```

Şekil 46. IBM WebSphere MQ sunucusu 2 'de

6. Test dosyalarını silin.

```
amqmfscck -x /shared/qmdata
Test files deleted.
```

Şekil 47. IBM WebSphere MQ Server 2 'de

Sunucu, şu iletiyle yanıt verir:

```
Test files deleted.
```

Sonuçlar

Sınamalar başarıyla tamamlanırsa, sıfır dışında bir çıkış kodu döndürülür; tersi durumda sıfır dışında bir değer döndürülür.

Örnekler

İlk üç örnek kümesi, minimum çıkış üreten komutu gösterir.

Bir sunucuda temel dosya kilitlemeyi başarılı bir şekilde test edin

```
> amqmfscck /shared/qmdata
The tests on the directory completed successfully.
```

Bir sunucuda temel dosya kilitleme sınaması başarısız oldu

```
> amqmfscck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck' error '2'.
```


İki sunucu üzerinde kilitleme sınaması başarıyla tamamlandı

Çizelge 13. İki sunucu üzerinde başarılı kilitleme	
IBM WebSphere MQ Sunucu 1	IBM WebSphere MQ Sunucu 2
> amqmfscck -w /shared/qmdata Please start this program on a second machine with the same parameters. Lock acquired. Press Return or terminate the program to release the lock.	
	> amqmfscck -w /shared/qmdata Waiting for lock...
[Return pressed] Lock released.	
	Lock acquired. The tests on the directory completed successfully

İkinci üç örnek kümesi, ayrıntılı kipi kullanarak aynı komutları gösterir.

Bir sunucuda temel dosya kilitlemeyi başarılı bir şekilde test edin

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd1 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd1, F_SETLK, F_RDLCK)
System call: fd2 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd2, F_SETLK, F_RDLCK)
System call: close(fd2)
System call: write(fd1)
System call: close(fd1)
The tests on the directory completed successfully.
```

Bir sunucuda temel dosya kilitleme sınaması başarısız oldu

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_RDLCK)
System call: fdSameFile = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fdSameFile, F_SETLK, F_RDLCK)
System call: close(fdSameFile)
System call: write(fd)
AMQxxxx: Error calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck', errno 2
(Permission denied).
```

İki sunucu üzerinde kilitleme sınaması başarıyla tamamlandı

Çizelge 14. İki sunucu üzerinde başarılı kilitleme-verbose kipi

IBM WebSphere MQ Sunucu 1	IBM WebSphere MQ Sunucu 2
<pre>> amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL O_CREAT O_RDWR, 0666)' Calling 'fchmod(fd, 0666)' Calling 'fstat(fd)' Please start this program on a second machine with the same parameters. Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. Press Return or terminate the program to release the lock.</pre>	
	<pre>> amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL O_CREAT O_RDWR,0666)' Calling 'fd = open("/shared/qmdata/amqmfscck.lkw, O_RDWR, 0666)' Calling 'fcntl(fd, F_SETLK, F_WRLCK) 'Waiting for lock...</pre>
<pre>[Return pressed] Calling 'close(fd)' Lock released.</pre>	
	<pre>Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. The tests on the directory completed successfully</pre>

İlgili başvurular

amqmfscck (dosya sistemi denetimi)

İleti bütünlüğünü sınamak için amqsfhac çalıştırılıyor

amqsfhac , ağ üzerinde çalışan depolamayı kullanan bir kuyruk yöneticisinin, bir arızmanın ardından veri bütünlüğünü koruduğunu denetler.

Başlamadan önce

Bu sınamaya için dört sunucuya gereksinim duyarsınız. Çok eşgörünümlü kuyruk yöneticisi için iki sunucu, dosya sistemi için bir sunucu ve bir IBM WebSphere MQ MQI client uygulaması olarak **amqsfhac** ' u çalıştırmak için kullanılan iki sunucu.

Çok eşgörünümlü bir kuyruk yöneticisine ilişkin dosya sistemini ayarlamak için [Yordam](#) ' daki ["1" sayfa 126](#) adımı izleyin.

Bu görev hakkında

Yordam

1. Create a multi-instance queue manager on another server, QM1, using the file system you created in step ["1" sayfa 126](#) in [Yordam](#).

Bkz. [Çok eşgörünümlü kuyruk yöneticisi yaratılması](#).

2. Kuyruk yöneticisini, yüksek kullanılabilirlikli kılabilir her iki sunucuda da başlatın.

Sunucu 1 'de:

```
strmqm -x QM1
```

Sunucu 2 'de:

```
strmqm -x QM1
```

3. Set up the client connection to run **amqsfhac**.

- İstemci bağlantısı kurmak için İstemci kuruluşunun doğrulanması başlıklı bölümdeki yordamı ya da Yeniden yapılandırma istemcisi örnekleri içindeki örnek komut dosyalarını kullanın.
- İstemci kanalını, QM1çalıştıran iki sunucuya karşılık gelen iki IP adresine sahip olacak şekilde değiştirin.

Örnek komut dosyasında şunu değiştirin:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
```

Hedef:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
```

Burada `server1` ve `server2`, iki sunucunun anasistem adlarıdır ve 2345, kanal dinleyicisinin dinlediği kapıdır. Genellikle bu varsayılan değer 1414değerine ayarlanır. 1414 olanağını varsayılan dinleyici yapılandırmasıyla kullanabilirsiniz.

4. Sınamaya ilişkin QM1 üzerinde iki yerel kuyruk yaratın.

Aşağıdaki MQSC komut dosyasını çalıştırın:

```
DEFINE QLOCAL(TARGETQ) REPLACE  
DEFINE QLOCAL(SIDEQ) REPLACE
```

5. Test the configuration with **amqsfhac**

```
amqsfhac QM1 TARGETQ SIDEQ 2 2 2
```

6. Dosya sistemi bütünlüğünü test ederken ileti bütünlüğünü test edin.

Yordam' in "5" sayfa 126 adımı sırasında **amqsfhac** komutunu çalıştırın.

```
amqsfhac QM1 TARGETQ SIDEQ 10 20 0
```

Etkin kuyruk yöneticisi yönetim ortamını durdurursanız, **amqsfhac** etkinleştikten sonra diğer kuyruk yöneticisi eşgörünümüye yeniden bağlanır. Durdurulan kuyruk yöneticisi yönetim ortamını yeniden başlatın; böylece, sonraki testinizde başarısızlığı tersine çevirebilirsiniz. Test programının, hata durumunda yedek sisteme geçiş işlemi için yeterli süre çalışması için ortamınızla denemelere dayalı yinelemelerin sayısını artırmanız gerekebilir.

Sonuçlar

An example of running **amqsfhac** in step "6" sayfa 131 is shown in Şekil 48 sayfa 132. Test başarılı oldu.

Sinama bir sorun algıladıysa, çıkış hatayı bildirecektir. Bazı testlerde MQRC_CALL_INTERRUPTED, "Resolving to backed out" raporunu raporlayabilir. Sonuç için bir fark yok. Sonuç, diske yazma işleminin, hata gerçekleşmeden önce ya da sonra ağa bağlı dosya depolaması tarafından işlenip işlenmediğine bağlıdır.

```

Sample AMQSFHAC start
qmname = QM1
qname = TARGETQ
sidename = SIDEQ
transize = 10
iterations = 20
verbose = 0
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Resolving MQRC_CALL_INTERRUPTED
MQGET browse side tranid=14 pSideinfo->tranid=14
Resolving to committed
Iteration 7
Iteration 8
Iteration 9
Iteration 10
Iteration 11
Iteration 12
Iteration 13
Iteration 14
Iteration 15
Iteration 16
Iteration 17
Iteration 18
Iteration 19
Sample AMQSFHAC end

```

Şekil 48. Output from a successful run of **amqsfhac**

İlgili başvurular

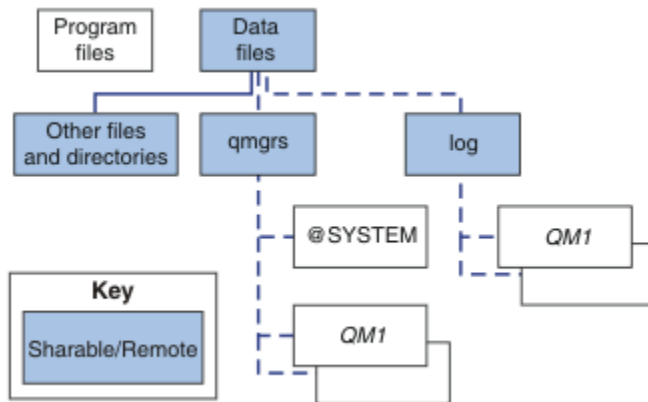
[Yüksek kullanılabilirlik örnek programları](#)

Sharing IBM WebSphere MQ files

Bazı IBM WebSphere MQ dosyalarına yalnızca etkin bir kuyruk yöneticisi tarafından erişilir, diğer dosyalar paylaşılır.

WebSphere MQ dosyaları, program dosyaları ve veri dosyaları olarak bölünüyorlar. Program dosyaları tipik olarak, WebSphere MQ çalıştıran her sunucuda yerel olarak kurulur. Kuyruk yöneticileri, varsayılan veri dizinindeki veri dosyalarına ve dizinlere erişimi paylaşır. Bunlar, Şekil 49 sayfa 132’inde gösterilen qmgrs ve log dizinlerinin her birinde bulunan kendi kuyruk yöneticisi dizini ağaçlarına dışlayıcı erişim gerektirir.

Şekil 49 sayfa 132 , WebSphere MQ dizin yapısının üst düzey bir görünütüdür. Kuyruk yöneticileri arasında paylaşılabilen ve uzak kılınabilen dizinleri gösterir. Ayrıntılar altyapıya göre değişir. Noktalı çizgiler, yapılandırılabilir yolları gösterir.



Şekil 49. WebSphere MQ dizin yapısının genel görünümü

Program dosyaları

Program dosyaları dizini genellikle varsayılan konumda bırakılır, yerel olur ve sunucudaki tüm kuyruk yöneticileriyle paylaşılır.

Veri dosyaları

The data files directory is typically local in the default location, /var/mqm on UNIX and Linux systems and configurable on installation on Pencereler. Kuyruk yöneticileri arasında paylaşılır. Varsayılan konum uzakını yapabilirsiniz, ancak bunu WebSphere MQ' un farklı kuruluşları arasında paylaşmayın. WebSphere MQ yapılanış noktasındaki DefaultPrefix özniteliği bu yolu gösterir.

qmgrs

v7.0.1' tan, kuyruk yöneticisi verilerinin konumunu belirtmenin iki farklı yolu vardır.

ÖnekKullanma

Prefix özniteliği, qmgrs dizininin yerini belirtir. WebSphere MQ , kuyruk yöneticisi dizini adını kuyruk yöneticisi adından oluşturur ve bunu, qmgrs dizininin bir alt dizini olarak yaratır.

Prefix özniteliği, QueueManager stanza içinde bulunur ve DefaultPrefix özniteliğinden edinilmiş olan değerden devralınır. Yönetim basitliği için varsayılan olarak, kuyruk yöneticileri genellikle aynı qmgrs dizinini paylaşır.

QueueManager kısmı mqm . ini dosyasında yer alan bir dosyadır.

Herhangi bir kuyruk yöneticisi için qmgrs dizininin konumunu değiştirirseniz, Prefix özniteliğinin değerini değiştirmeniz gerekir.

Bir UNIX and Linux platformu için [Şekil 49 sayfa 132](#) içindeki QM1 dizinine ilişkin Prefix özniteliği,

```
Prefix=/var/mqm
```

DataPatholanağının kullanılması

DataPath özniteliği, kuyruk yöneticisi veri dizininin yerini belirtir.

DataPath özniteliği, kuyruk yöneticisi veri dizininin adı da içinde olmak üzere, tam yolu belirtir. DataPath özniteliği, kuyruk yöneticisi verileri dizininin tamamlanmamış bir yolu belirten Prefix özniteliğinden farklıdır.

DataPath özniteliği belirtilirse, QueueManager stanza içinde yer alır. Belirtilmişse, Prefix (Önek) özniteindeki herhangi bir değere göre öncelik kazanır.

QueueManager kısmı mqm . ini dosyasında yer alan bir dosyadır.

Herhangi bir kuyruk yöneticisine ilişkin kuyruk yöneticisi veri dizininin yerini değiştirirseniz, DataPath özniteliğinin değerini değiştirmelisiniz.

Bir UNIX ya da Linux platformu için [Şekil 49 sayfa 132](#) içindeki QM1 dizinine ilişkin DataPath özniteliği,

```
DataPath=/var/mqm/qmgrs/QM1
```

log

Günlük dizini, kuyruk yöneticisi yapılanışındaki Log kısmı içindeki her kuyruk yöneticisi için ayrı olarak belirtilir. The queue manager configuration is in qm . ini.

DataPath/QmgrName/@IPCC alt dizinleri

DataPath/QmgrName/@IPCC alt dizinleri paylaşılan dizin yolunda yer alıyor. Bunlar, IPC dosya sistemi nesnelere için dizin yolu oluşturmak üzere kullanılır. Bir kuyruk yöneticisi sistemler arasında paylaşıldığında, kuyruk yöneticisinin ad alanını ayırt edebilmeleri gerekir. V7.0.1öncesinde, bir kuyruk yöneticisi yalnızca bir sistemde kullanılıyordu. IPC dosya sistemi nesnelere dizin yolunu tanımlamak için bir alt dizin kümesi yeterli oldu, bkz. [Şekil 50 sayfa 134](#).

```
DataPath/QmgrName/@IPCC/esem
```

Şekil 50. Örnek IPC altdizini, pre-V7.0.1

V7.0.1 ve sonraki yayın düzeylerinde, IPC dosya sistemi nesnelere sistem tarafından ayırt edilmesi gerekir. Bir alt dizin, kuyruk yöneticisinin çalıştığı her sistem için dizin yoluna eklenir, bkz. [Şekil 51 sayfa 134](#).

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

Şekil 51. Örnek IPC altdizini, V7.0.1 ve sonraki yayın düzeyleri

myHostName , işletim sistemi tarafından döndürülen anasistem adının ilk 20 karakterine uymaktadır. Bazı sistemlerde, kesme işleminden önce anasistem adı en çok 64 karakter uzunluğunda olabilir. The generated value of *myHostName* might cause a problem for two reasons:

1. İlk 20 karakter benzersiz değil.
2. Anasistem adı, her zaman bir sisteme aynı anasistem adını ayırmayan bir DHCP algoritması tarafından oluşturulur.

In these cases, set *myHostName* using the environment variable, MQC_IPC_HOST; see [Şekil 52 sayfa 134](#).

```
export MQC_IPC_HOST=myHostName
```

Şekil 52. Örnek: ayar MQC_IPC_HOST

Diğer dosyalar ve dizinler

İzleme dosyalarını içeren dizin ve ortak hata günlüğü gibi diğer dosyalar ve dizinler, olağan durumda yerel dosya sisteminde paylaşılır ve saklanır.

Up until v7.0.1, WebSphere MQ relied upon external management to guarantee queue managers exclusive access to the queue manager data and log files. From v7.0.1 onwards, with support of shared file systems, WebSphere MQ manages exclusive access to these files using file system locks. Bir dosya sistemi kilidi, belirli bir kuyruk yöneticisinin yalnızca bir eşgörünümünün bir kerede etkin olmasını sağlar.

Belirli bir kuyruk yöneticisinin ilk eşgörünümünü başlattığınızda, kuyruk yöneticisi dizininin sahipliğini alır. İkinci bir yönetim ortamı başladığınızda, yalnızca ilk yönetim ortamı durduysa, bu yönetim ortamı sahiplik alabilir. İlk kuyruk yöneticisi çalışmaya devam ederse, ikinci eşgörünüm başlatılamaz ve kuyruk yöneticisi başka bir yerde çalıştırıyor demektir. İlk kuyruk yöneticisi durduysa, ikinci kuyruk yöneticisi kuyruk yöneticisi kütüklerinin sahipliğini devralır ve çalışmakta olan kuyruk yöneticisi olur.

İkinci kuyruk yöneticisinin ilkinden devralan prosedürü otomatikleştirebilirsiniz. İlk kuyruk yöneticisini, başka bir kuyruk yöneticisinin devralmasına izin veren `strmqm -x` seçeneği ile başlatın. İkinci kuyruk yöneticisi, kuyruk yöneticisi kütüklerinin sahipliğini alma girişiminde bulunmadan önce kuyruk yöneticisi kütüklerinin kilidi açılıncaya kadar bekler ve başlar.

UNIX and Linux sistemlerindeki dizin yapısı

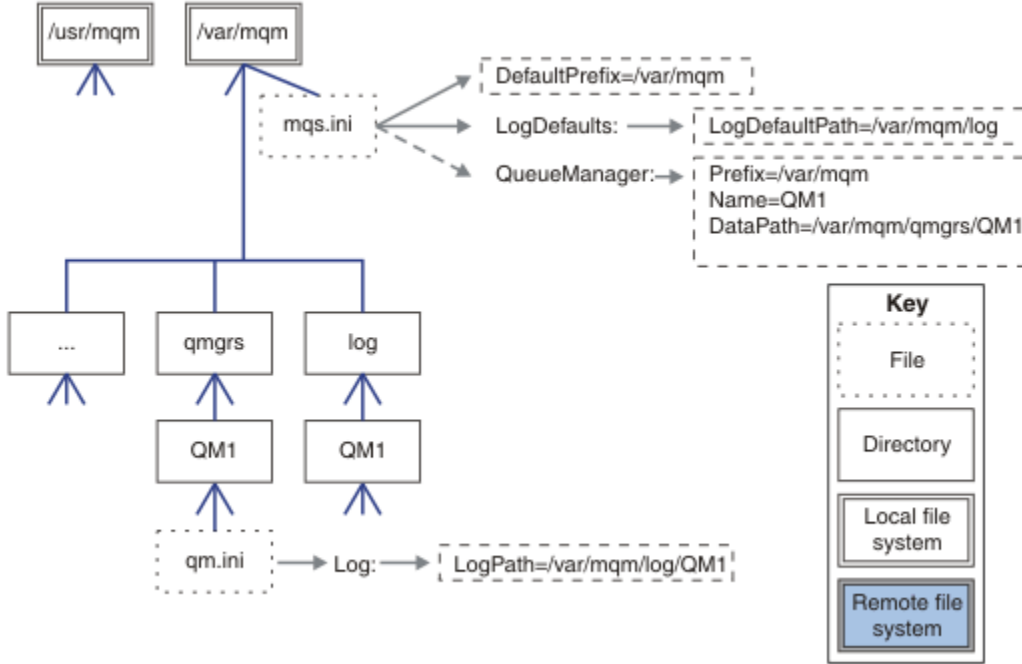
UNIX and Linux sistemlerindeki WebSphere MQ dizin yapısı, daha kolay yönetim, daha iyi performans ve daha iyi güvenilirlik için farklı dosya sistemlerine eşlenebilir.

Çok eşgörünümlü kuyruk yöneticileri çalıştırmak için paylaşılan dosya sistemlerinden yararlanmak için WebSphere MQ ' nun esnek dizin yapısını kullanın.

Use the command `crtmqm QM1` to create the directory structure shown in [Şekil 53 sayfa 135](#) where R is the release of the product. v7.0.1 sisteminden başlayarak WebSphere MQ sisteminde yaratılmış bir kuyruk yöneticisi için tipik bir dizin yapısıdır. Bazı dizinler, dosyalar ve .ini öznitelik ayarları netlik için atlanır ve başka bir kuyruk yöneticisi adı mangging tarafından değiştirilebilir. Dosya sistemlerinin adları farklı sistemlere göre değişir.

Tipik kurulumda, yarattığınız her kuyruk yöneticisi yerel dosya sisteminde ortak log ve qmgrs dizinlerine işaret eder. Çok eşgörmümlü bir yapılandırmada, log ve qmgrs dizinleri, başka bir WebSphere MQkuruluşuyla paylaşılan bir ağ dosya sisteminde yer alıyor.

Şekil 53 sayfa 135 , AIX üzerinde WebSphere MQ v7.R için varsayılan yapılandırmayı göstermektedir; burada R , ürünün yayınının yayınıdır. Diğer çok eşgörmümlü yapılandırmalara ilişkin örnekler için bkz. "UNIX and Linux sistemlerindeki izin yapılandırmaları örneği" sayfa 139.



Şekil 53. UNIX and Linux sistemleri için örnek varsayılan WebSphere MQ v7.R izin yapısı

The product is installed into /usr/mqm on AIX and /opt/mqm on the other systems, by default. Çalışma dizinleri /var/mqm dizinine kurulur.

Not: IBM WebSphere MQkurulmadan önce /var/mqm kütük sistemini yarattıysanız, mqm kullanıcısının tam izinlerine sahip olduğundan emin olun; örneğin, dosya kipi 755.

log ve qmgrs dizinleri varsayılan konumlarında, mqs . ini dosyasındaki LogDefaultPath ve DefaultPrefix özniteliklerinin varsayılan değerleri tarafından tanımlandığı şekilde gösterilir. Bir kuyruk yöneticisi yaratıldığında, varsayılan olarak kuyruk yöneticisi veri dizini *DefaultPrefix*/qmgrs dizininde yaratılır ve günlük kütüğü dizini *LogDefaultPath* /logdizinde yaratılır. LogDefaultYolu ve DefaultPrefix yalnızca kuyruk yöneticilerinin ve günlük dosyalarının varsayılan olarak oluşturulduğu etkiye neden olur. Bir kuyruk yöneticisi dizininin gerçek konumu mqs . ini dosyasına kaydedilir ve günlük dosyası dizininin konumu qm . ini dosyasına kaydedilir.

Bir kuyruk yöneticisine ilişkin günlük dosyası dizini, LogPath özniteindeki qm . ini dosyasında tanımlanır. Bir kuyruk yöneticisi için LogPath özniteliğini ayarlamak için **crtmqm** komutu üzerinde -ld seçeneğini kullanın; örneğin, **crtmqm -ld LogPath QM1**. If you omit the ld parameter the value of LogDefaultYolu is used instead.

Kuyruk yöneticisi verileri dizini, mqs . ini dosyasındaki QueueManager stanza içindeki DataPath özniteisinde tanımlanır. Bir kuyruk yöneticisi için DataPath ayarlamak için **crtmqm** komutundaki -md seçeneğini kullanın; örneğin, **crtmqm -md DataPath QM1**. md parametresini çıkarırsanız, bunun yerine DefaultPrefix ya da Prefix özniteliğinin değerini kullanın. Önek , DefaultPrefix ' ten öncelikli olarak uygulanır.

Tipik olarak, tek bir komutta hem günlük hem de veri dizinlerini belirten QM1 oluşturur.

```
crtmqm  
-md DataPath -ld  
LogPath QM1
```

Kuyruk yöneticisi durdurulduğunda, `qm.ini` dosyasındaki `DataPath` ve `LogPath` özniteliklerini düzenleyerek, var olan bir kuyruk yöneticisinin kuyruk yöneticisi günlüğünün ve veri dizinlerinin yerini değiştirebilirsiniz.

The path to the `errors` directory, like the paths to all the other directories in `/var/mqm`, is not modifiable. Ancak, dizinlerin farklı dosya sistemlerine ya da sembolik olarak farklı dizinlere bağlanabileceği.

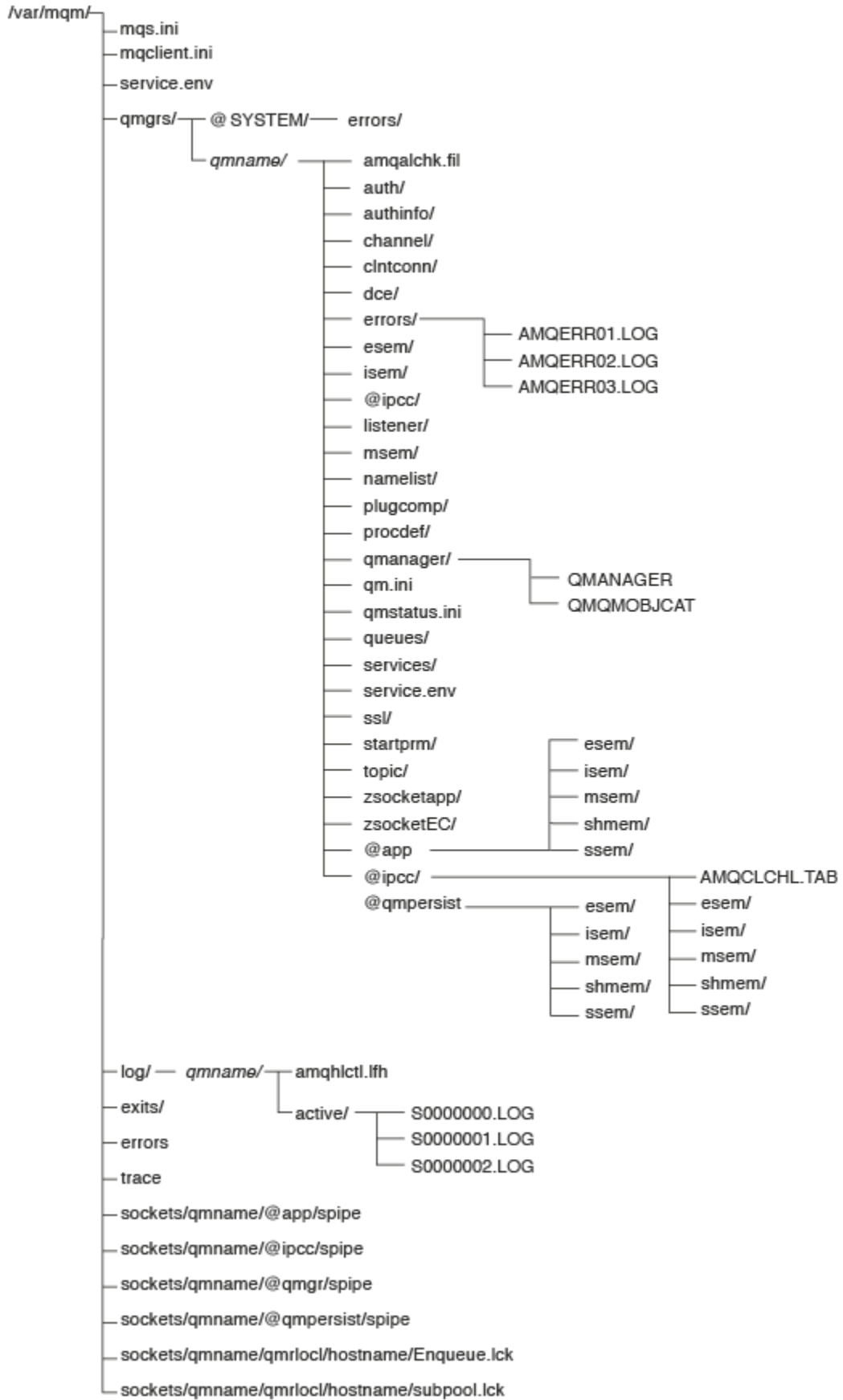
UNIX and Linux sistemlerindeki izin içeriği

Kuyruk yöneticisiyle ilişkili dizinlerin içeriği.

Ürün dosyalarının konumu hakkında bilgi için bkz. [Kuruluş konumunu seçme](#)

Diğer izin yapılandırmalarıyla ilgili bilgi için bkz. [“Planlama dosya sistemi desteği” sayfa 122.](#)

Şekil 54 sayfa 137’inde, bir kuyruk yöneticisi bir süre kullanımda olduktan sonra yerleşim düzeni WebSphere MQ ' nun temsilcidir. Sahip olduğunuz gerçek yapı, kuyruk yöneticiliklerinde hangi işlemlerin gerçekleştirildiğine bağlıdır.



Şekil 54. Kuyruk yöneticisi başlatıldıktan sonra varsayılan dizin yapısı (UNIX sistemleri)

/var/mqm/

/var/mqm dizini, tek bir kuyruk yöneticisine değil, bir bütün olarak WebSphere MQ kurulumu için geçerli olan yapılandırma dosyalarını ve çıkış dizinlerini içerir.

<i>Çizelge 15. UNIX sistemlerindeki /var/mqm dizininin belgelenmiş içeriği</i>	
<u>mqs.ini</u>	WebSphere MQ installation wide configuration file, read when a queue manager starts. Dosya yolu, AMQ_MQS_INI_LOCATION ortam değişkeni kullanılarak değiştirilebilir. Bunun ayarlandığından ve strmqm komutunun çalıştırıldığı kabukta dışa aktarıldığından emin olun.
<u>mqclient.ini</u>	Varsayılan istemci yapılandırma dosyası WebSphere MQ MQI istemci programları tarafından okunur. Dosya yolu, MQCLNTCF ortam değişkeni kullanılarak değiştirilebilir.
<u>service.env</u>	Bir hizmet süreci için makine kapsamı ortam değişkenlerini içerir. Dosya yolu düzeltti.
<u>hatalar/</u>	Makine kapsamı hata günlükleri ve FFST dosyaları. Dizin yolu düzeltti. Ayrıca bkz. FFST: IBM WebSphere MQ for UNIX and Linux systems.
<u>yuvalar/</u>	Her kuyruk yöneticisine ilişkin bilgileri yalnızca sistem kullanımı için içerir.
<u>izleme/</u>	İzleme dosyaları. Dizin yolu düzeltti.
<u>çıkışlar/</u>	Kullanıcı kanalı çıkış programlarını içeren varsayılan dizin. mqs.ini dosyasındaki ApiExit kısmında yer değiştirebilen yer.
<u>exits64/</u>	

/var/mqm/qmgrs/qmname/

/var/mqm/qmgrs/qmname/ , kuyruk yöneticisine ilişkin dizinleri ve dosyaları içerir. Dizin, etkin kuyruk yöneticisi yönetim ortamı tarafından dışlayıcı erişim için kilitlenir. Dizin yolu doğrudan **mqs.ini** dosyasında değiştirilebilir ya da **strmqm** komutunun **md** seçeneği kullanılarak değiştirilebilir.

<i>Çizelge 16. UNIX sistemlerinde /var/mqm/qmgrs/qmname dizininin belgelenmiş içeriği</i>	
<u>qm.ini</u>	Kuyruk yöneticisi yapılandırma kütüğü, kuyruk yöneticisi başlatıldığında okunur.
<u>hatalar/</u>	Kuyruk yöneticisi kapsam hata günlükleri. qmname = @system , bilinmeyen ya da kullanılmayan bir kuyruk yöneticisine ilişkin kanalla ilgili iletileri içerir.
<u>@ipcc/</u> <u>AMQCLCHL.TAB</u>	WebSphere MQ sunucusu tarafından yaratılan ve WebSphere MQ MQI istemci programları tarafından okunan varsayılan istemci kanalı denetim çizelgesi. MQCHLLIB ve MQCHLTAB ortam değişkenleri kullanılarak dosya yolu değiştirilebilir.
<u>qmanager</u>	Kuyruk yöneticisi nesne dosyası: QMANAGER Kuyruk yöneticisi nesne kataloğu: QMQMOBJCAT

Çizelge 16. UNIX sistemlerinde /var/mqm/qmgrs/qmname dizininin belgelenmiş içeriği (devamı var)	
authinfo/	Kuyruk yöneticisi içinde tanımlanan her nesne, bu dizinlerdeki bir dosyayla ilişkilendirilir. Dosya adı yaklaşık olarak tanımlama adıyla eşleşir; bkz. Understanding WebSphere MQ file names .
kanal/	
clntconn/	
dinleyici/	
ad listesi/	
procdef/	
kuyruklar/	
hizmetler/	
konular/	
...	WebSphere MQ tarafından kullanılan diğer dizinler (örneğin, @ipccgibi), yalnızca WebSphere MQ tarafından değiştirilmelidir.

/var/mqm/log/qmname/

/var/mqm/log/qmname/ , kuyruk yöneticisi günlük dosyalarını içerir. Dizin, etkin kuyruk yöneticisi yönetim ortamı tarafından dışlayıcı erişim için kilitlenir. Dizin yolu, qm.ini dosyasında değiştirilebilir ya da **crtmqm** komutunun **ld** seçeneği kullanılarak.

Çizelge 17. UNIX sistemlerinde /var/mqm/log/qmname dizininin belgelenmiş içeriği	
amqhlctl.lfh	Denetim dosyasını günlüğe kaydet.
etkin/	Bu dizin, S0000000.LOG(LOG), S0000001.LOG, S0000002.LOG(LOG), vb.

UNIX and Linux sistemlerindeki dizin yapılandırmaları örneği

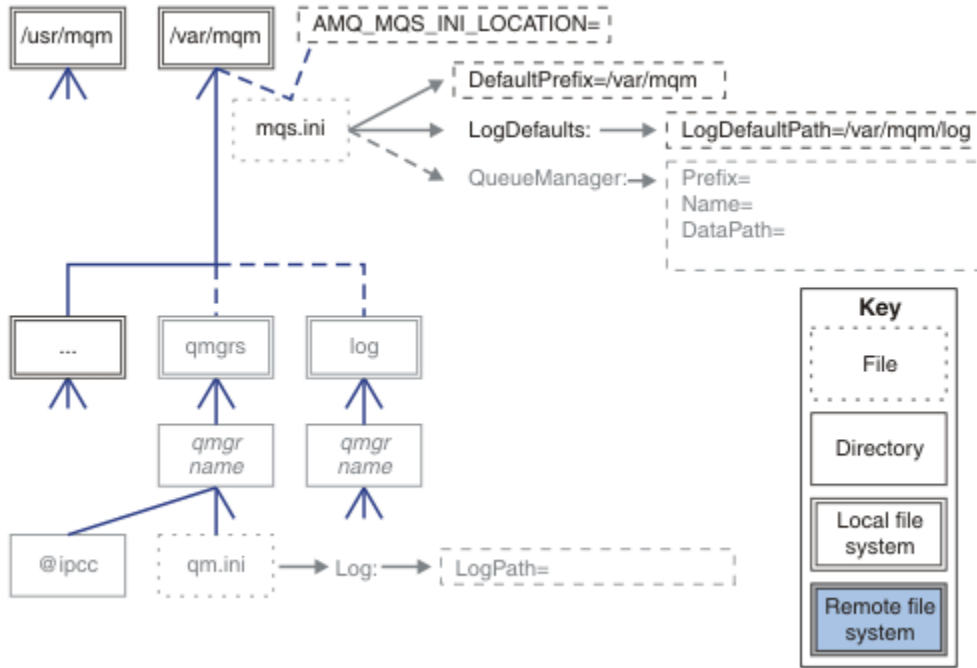
UNIX and Linux sistemlerinde diğer dosya sistemi yapılandırmalarına örnekler.

Çeşitli amaçlara ulaşmak için WebSphere MQ dizin yapısını çeşitli şekillerde özelleştirebilirsiniz.

- Çok eşgörünümlü bir kuyruk yöneticisi yapılandırmak için, uzak paylaşılan kütük sistemlerine qmgrs ve log dizinlerini yerleştirin.
- G/Ç çekişmesini azaltarak başarıyı artırmak için, veri ve günlük dizinleri için ayrı dosya sistemleri kullanın ve dizinleri farklı disklere ayırın.
- Daha yüksek performans üzerinde etkisi olan dizinler için daha hızlı depolama aygıtları kullanın. Fiziksel aygıt gecikme süresi, bir aygıtın yerel olarak mı, yoksa uzaktan mı monte edildiğinden, sürekli ileti sisteminin performansında sık sık daha önemli bir etkidir. Aşağıdaki liste, en çok hangi dizinlerin en düşük düzeyde duyarlı olduğunu gösterir.
 1. log
 2. qmgrs
 3. /usr/mqmiçinde olmak üzere diğer dizinler
- Örneğin, yedek disk dizisi gibi iyi dayanıklılığa sahip depolama için ayrılan dosya sistemlerinde qmgrs ve log dizinlerini oluşturun.
- Ortak hata günlüklerini bir ağ dosya sisteminde değil, yerel olarak var/mqm/errors ' ta saklamak daha iyidir. Böylece, ağ dosya sistemiyle ilgili hata günlüğe kaydedilebilir.

Şekil 55 sayfa 140 , alternative WebSphere MQ dizin yapılarının türetildiği bir şablondur. Şablonda noktalı çizgiler, yapılandırılabilir yolları temsil eder. Örneklerde, noktalı çizgiler, AMQ_MQS_INI_LOCATION ortam değişkeninde saklanan yapılandırma bilgilerine karşılık gelen katı çizgileriyle değiştirilir ve mqsc.ini ve qm.ini dosyalarında yer alır.

Not: Yol bilgileri, mqs . ini ya da qm . ini dosyalarında görüldüğü şekilde gösterilir. If you supply path parameters in the **crtmqm** command, omit the name of the queue manager directory: the queue manager name is added to the path by WebSphere MQ after it has been mangled.



Şekil 55. Dizin yapısı örüntü şablonu

Yapılandırılan dizin yapılarına ilişkin örnekler. The first example shows a typical default directory structure for WebSphere MQ v7.0.1 created by issuing the **crtmqm QM1** command. The second example shows how a typical directory structure appears for a queue manager created using a WebSphere MQ release earlier than v7.0.1. Dizin yapısı değişmez.

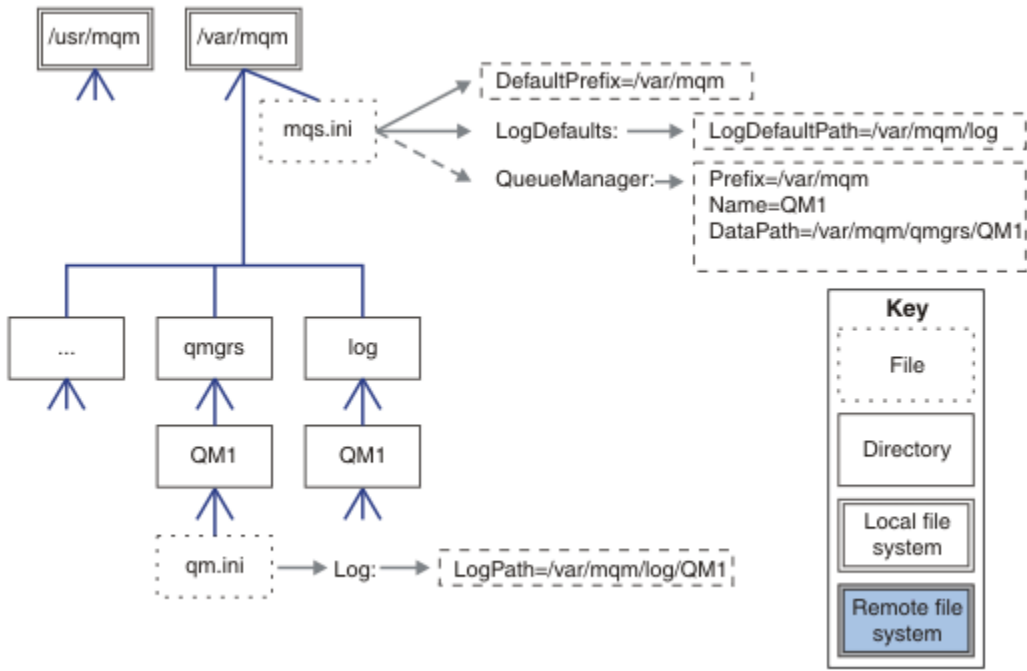
7.0.1 sürümünde yeni yaratılan kuyruk yöneticilerinin, önceki v7yayın düzeylerine farklı bir yapılandırma dosyası vardır. v7.0.0.2' a geri dönmek için v7.0.1 düzeltme paketini kaldırmamız gerekiyor, yapılandırma dosyalarını yeniden oluşturmanız gerekir. Yeni kuyruk yöneticisi veri dizininin yolunu tanımlamak için yalnızca Prefix (Önek) özniteliğini kullanmanız gerekebilir ya da kuyruk yöneticisi veri dizininin ve günlük dizinlerini farklı bir konuma taşımanız gerekebilir. Kuyruk yöneticisini yeniden yapılandırmanın en güvenli yolu, kuyruk yöneticisi verilerini ve günlük dizinlerini kaydetmek, kuyruk yöneticisini silmek ve yeniden yaratmak ve daha sonra, saklananlar ile veri ve günlük dizinlerini yeni konumlarında değiştirmendir.

Typical directory structure for release v7.0.1 onwards

Şekil 56 sayfa 141 is the default directory structure created in v7.0.1 by issuing the command **crtmqm QM1**.

The mqs . ini file has a stanza for the QM1 queue manager, created by referring to the value of DefaultPrefix . The günlük stanza in the qm . ini file has a value for LogPath, set by reference to LogDefaultYolu in mqs . ini .

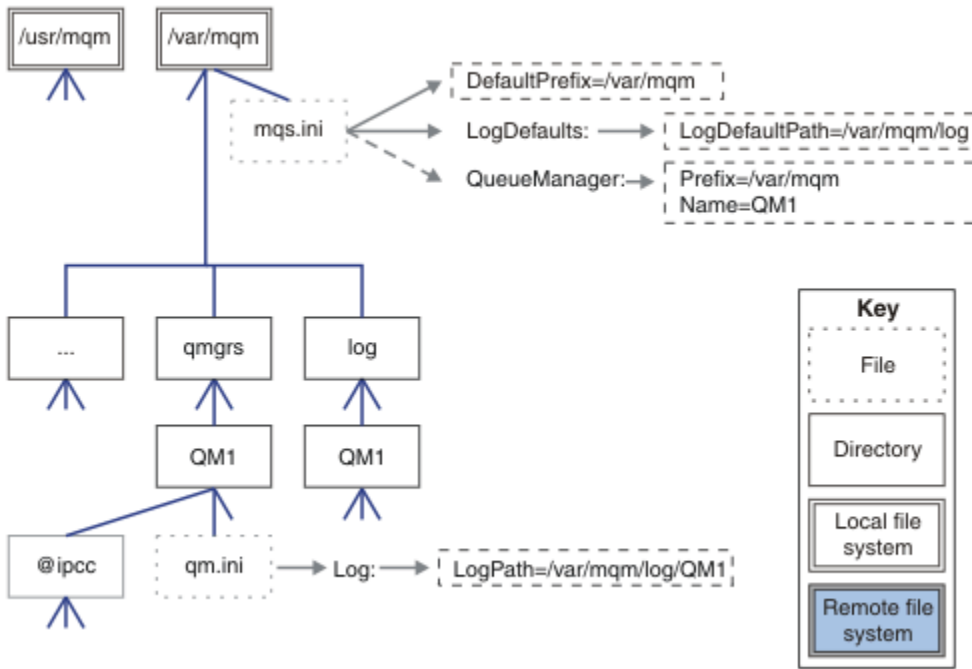
Use the optional **crtmqm** parameters to override the default values of DataPath and LogPath.



Şekil 56. UNIX and Linux sistemleri için örnek varsayılan WebSphere MQ v7.R dizin yapısı

v7.0.1öncesi yayın düzeyleri için tipik dizin yapısı

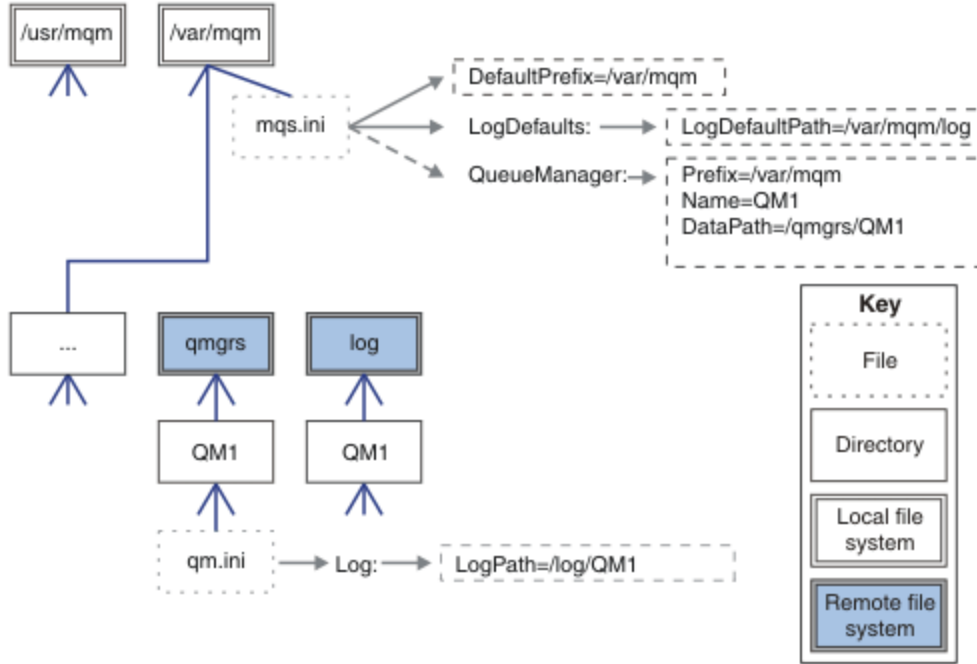
The DataPath attribute did not exist before WebSphere MQ v7.0.1; the attribute is not present in the mqs.ini file. qmgrs dizininin konumu, Prefix özneliği kullanılarak yapılandırılmıştı. Tek tek dizinlerin konumu, farklı dosya sistemi konumlarını göstermek için simgesel bağlantılar kullanılarak yapılandırılabilir.



Şekil 57. v7.0.1öncesi yayın düzeyleri için tipik dizin yapısı

Varsayılan qmgrs ve log Dizinlerini Paylaş (Yayın v7.0.1)

“Her şeyi paylaşın (Yayın v7.0.1)” sayfa 143 ' in alternatifi, qmgrs ve log dizinlerini ayrı olarak paylaşmasıdır (Şekil 58 sayfa 142). Bu yapılandırılmada, varsayılan mqs.ini yerel /var/mqm dosya sisteminde saklandığı için AMQ_MQS_INI_LOCATION ayarlamaya gerek yoktur. mqclient.ini ve mqserver.ini gibi dosyalar ve dizinler de paylaşılmaz.

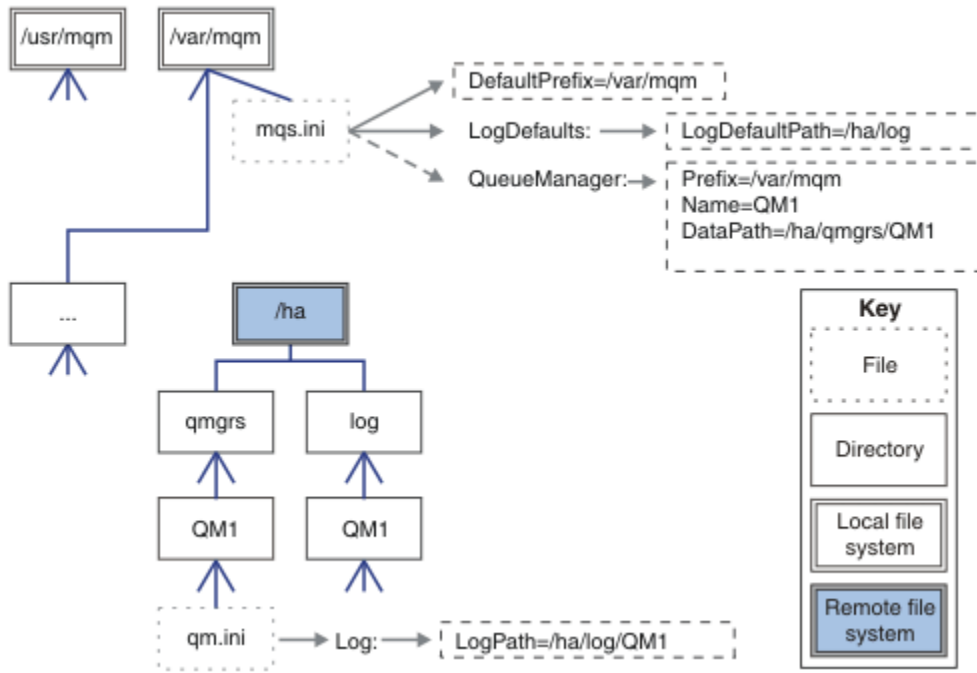


Şekil 58. Share qmgrs and log directories

Adlandırılmış qmgrs ve log dizinlerini (Yayın v7.0.1) paylaşır

The configuration in Şekil 59 sayfa 143 places the log and qmgrs in a common named remote shared file system called /ha . Aynı fiziksel yapılandırma iki farklı şekilde yaratılabilir.

1. LogDefaultPath=/ha seçeneğini ayarlayın ve daha sonra, komutu çalıştırın; **crtmqm -md /ha/qmgrs QM1**. Sonuç, tam olarak Şekil 59 sayfa 143 içinde gösterildiği gibi olur.
2. Leave the default paths unchanged and then run the command, **crtmqm -ld /ha/log - md /ha/qmgrs QM1**.



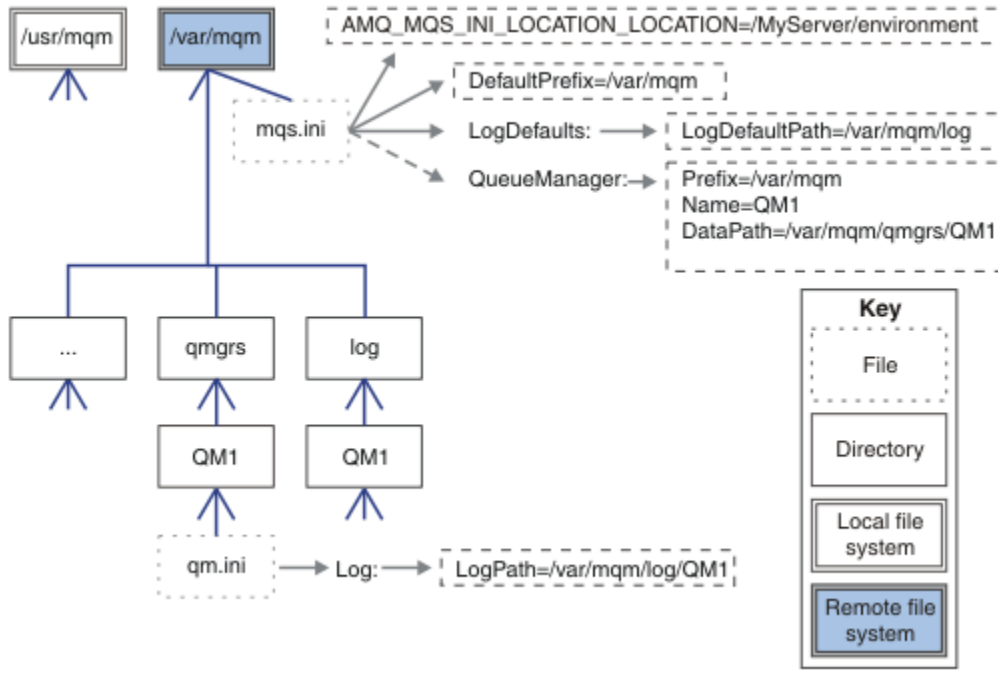
Şekil 59. Share named qmgrs and log directories

Her şeyi paylaşın (Yayın v7.0.1)

Şekil 60 sayfa 144 , hızlı ağ üzerinde dosya depolaması olan sistem için basit bir yapılandırmadır.

/var/mqm bağlantısını uzak paylaşılan bir dosya sistemi olarak sisteme bağlayın. Varsayılan olarak, QM1'u başlattığınızda /var/mqm 'u arar, paylaşılan dosya sisteminde bulur ve /var/mqmdosyasında mqs.ini dosyasını okur. Tüm sunucularınızda kuyruk yöneticileri için tek bir /var/mqm/mqs.ini dosyası kullanmak yerine, her sunucuda AMQ_QS_INI_LOCATION ortam değişkenini farklı mqs.ini dosyalarına işaret edecek şekilde ayarlayabilirsiniz.

Not: The contents of the generic error file in /var/mqm/errors/ are shared between queue managers on different servers.



Şekil 60. Her şeyi paylaşın

Çok eşgörünümlü kuyruk yöneticileri için bunu kullanamayacağınızı göz önünde bulundurun. Bunun nedeni, çok eşgörünümlü bir kuyruk yöneticisinde bulunan her anasistemin, semaforlar ve paylaşılan bellek gibi yerel verileri izlemek için /var/mqm ' un kendi yerel kopyasına sahip olması gerekir. Bu varlıklar anasistemler arasında paylaşılamaz.

Windows sistemlerindeki dizin yapısı

Windows üzerindeki kuyruk yöneticisi yapılandırma bilgilerini ve dizinlerini bulma.

IBM WebSphere MQ for Windows kurulumu için varsayılan dizin şudur:

32 bit

C:\Program Files\IBM\WebSphere MQ

64 bit

C:\Program Files (x86)\IBM\WebSphere MQ

Kurulum bilgileri Windows kayıt defterinde saklanır. IBM WebSphere MQ bilgilerinin depolandığı kayıt defteri anahtarı:

32 bit

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\

64 bit

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\IBM\WebSphere MQ\

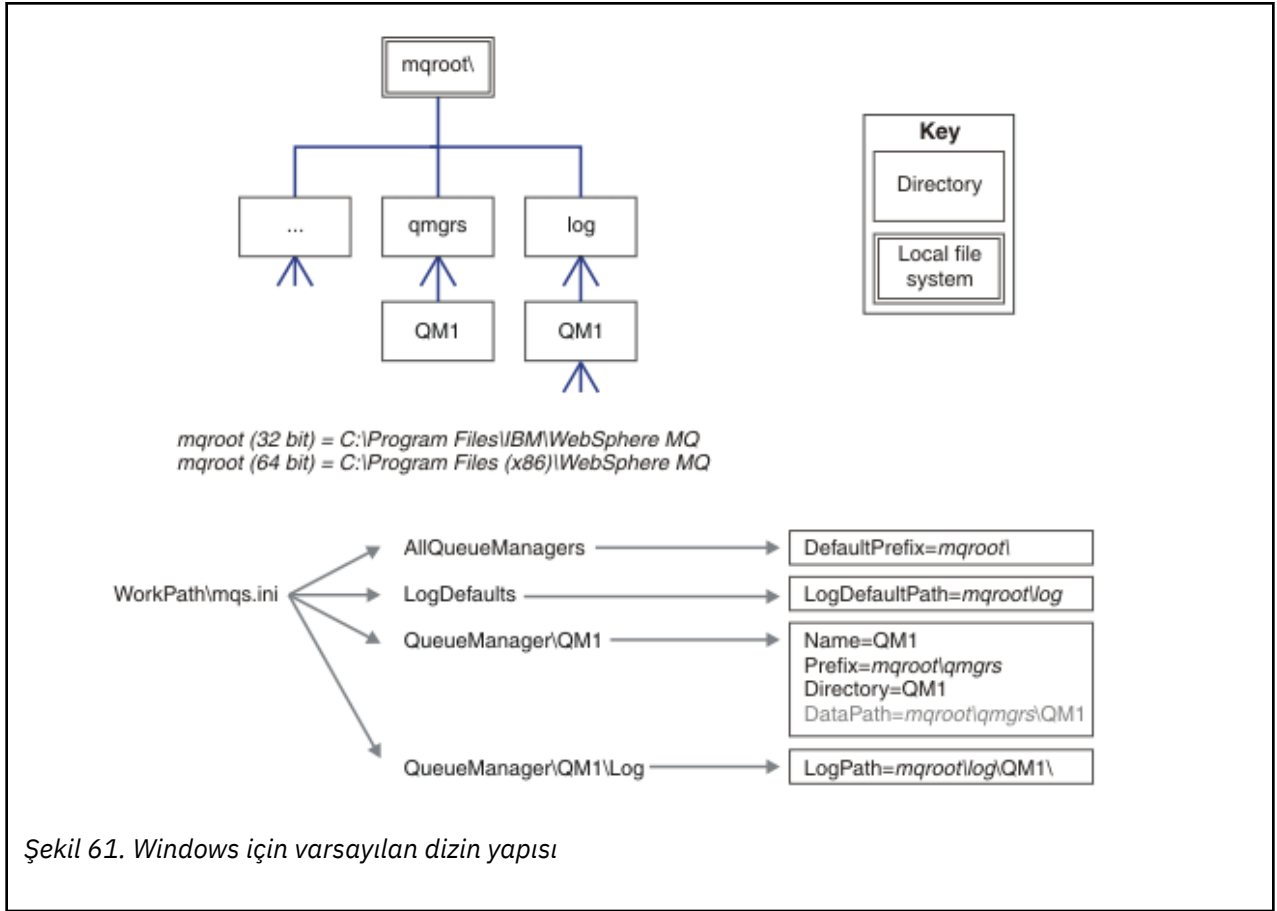
Her kurulum için belirli bir alt anahtar vardır:

```
Installation\<InstallationName>\
```

IBM WebSphere MQ veri dizinini işaret eden yol, *WorkPath* adlı bir dizgi değerinde saklanır ve günlükler için varsayılan dizin *LogDefaultPath* dizininde saklanır. Kuyruk yöneticisi veri dizinleri *WorkPath\qmgrs\Qmgrname* içinde oluşturulur. Kuyruk yöneticisi günlükleri *LogDefaultPath\QmgrName* içinde yaratılır. Bkz. [Şekil 61 sayfa 145](#).

If you define the queue manager data and log directories when installing IBM WebSphere MQ, then *WorkPath* and *LogDefaultYolu* are updated with the customized path information.

WorkPath ve *LogDefaultYolu* yalnızca kuyruk yöneticisi yaratmak için kullanılır.



Çok eşgörünümli kuyruk yöneticileri

Çok eşgörünümli bir kuyruk yöneticisini yapılandırmak için, günlük ve veri dizinlerinin ağ üzerinde çalışan depolama alanına yerleştirilmeleri gerekir. Tercihen, tercihen kuyruk yöneticisinin eşgörünümünü çalıştıran sunuculardan herhangi birine farklı bir sunucu üzerinde olmalıdır.

Kuyruk yöneticisi verilerinin ve günlük dizinlerinin konumunu daha kolay belirtmek için **crtmqm** komutu **-md** ve **-ld**' de iki parametre sağlanır. **-md** parametresini belirtmenin etkisi dört katlıdır:

1. `mqc.ini` stanza `QueueManager\QmgrName`, kuyruk yöneticisi verileri dizinini gösteren yeni bir değişken (`DataPath`) içerir. Unlike the `Önek` variable, the path includes the name of the queue manager directory.
2. `mqc.ini` dosyasında saklanan kuyruk yöneticisi yapılandırma bilgileri `Ad`, `Önek`, `Dizin` ve `DataPath` olarak azaltılır.

Dizin içeriği

WebSphere MQ dizinlerinin yerini ve içeriğini listeler.

Bir WebSphere MQ yapılandırmasında üç ana dosya kümesi ve dizin vardır.

1. Yalnızca bakım uygulandığında güncelleştirilen, `readme` (benioku) dosyası, WebSphere MQ Explorer eklentisi ve yardım dosyaları ve lisanslar dosyaları gibi diğer salt okunur dosyalar. Bu dosyalar [Çizelge 18 sayfa 146](#) içinde açıklanmıştır.
2. Belirli bir kuyruk yöneticisine özgü olmayan değiştirilebilir, değiştirilebilir dosyalar ve dizinler. Bu dosyalar ve dizinler [Çizelge 19 sayfa 146](#) içinde anlatılır.
3. Bir sunucudaki her kuyruk yöneticisine özgü dosyalar ve dizinler. Bu dosyalar ve dizinler [Çizelge 18 sayfa 146](#) içinde açıklanır.

Kaynak dizinleri ve dosyaları

Kaynak dizinleri ve dosyaları, bir kuyruk yöneticisini çalıştırmak için tüm yürütülebilir kodları ve kaynakları içerir. The variable, *FilePath*, in the installation specific IBM WebSphere MQ configuration registry key, contains the path to the resource directories.

Dosya yolu	İçindekiler
<i>FilePath</i> \bin	Komutlar ve DLL ' ler
<i>FilePath</i> \bin64	Komutlar ve DLL ' ler (64 bit)
<i>FilePath</i> \conv	Veri dönüştürme çizelgeleri
<i>FilePath</i> \doc	Sihirbaz yardım dosyaları
<i>FilePath</i> \MQExplorer	Explorer ve Explorer yardımı Eclipse eklentileri
<i>FilePath</i> \gskit8	Genel güvenlik seti
<i>FilePath</i> \java	JRE de içinde olmak üzere Java kaynakları
<i>FilePath</i> \licenses	Lisans bilgisi
<i>FilePath</i> \Non_IBM_License	Lisans bilgisi
<i>FilePath</i> \properties	Dahili olarak kullanılır
<i>FilePath</i> \Tivoli	
<i>FilePath</i> \tools	Geliştirme kaynakları ve örnekleri
<i>FilePath</i> \Uninst	Dahili olarak kullanılır
<i>FilePath</i> \README.TXT	Readme dosyası

Dizin, kuyruk yöneticisine özgü değil

Bazı dizinler, belirli bir kuyruk yöneticisine özgü olmayan izleme dosyaları ve hata günlükleri gibi dosyaları içerir. *DefaultPrefix* değişkeni, bu dizinlerin yolunu içerir. *DefaultPrefix* , *AllQueueManagers* stanza 'nın bir parçasıdır.

Dosya yolu	İçindekiler
<i>DefaultPrefix</i> \Config	Dahili olarak kullanılır
<i>DefaultPrefix</i> \conv	Veri dönüştürme içinde açıklananccsid.tbl veri dönüştürme denetimi dosyası
<i>DefaultPrefix</i> \errors	Kuyruk yöneticisi olmayan hata günlükleri, AMQERRnn . LOG
<i>DefaultPrefix</i> \exits	Kanal çıkış programları
<i>DefaultPrefix</i> \exits64	Kanal çıkış programları (64 bit)
<i>DefaultPrefix</i> \ipc	Kullanılmadı
<i>DefaultPrefix</i> \Qmgrs	Çizelge 20 sayfa 147 içinde anlatılan
<i>DefaultPrefix</i> \trace	iz dosyaları
<i>DefaultPrefix</i> \amqmjpse.txt	Dahili olarak kullanılır

Kuyruk yöneticisi dizinleri

Bir kuyruk yöneticisi yarattığınızda, kuyruk yöneticisine özgü yeni bir dizin kümesi yaratılır.

-md *filepath* parametresiyle bir kuyruk yöneticisi oluşturursanız, yol, mq.s.ini dosyasının kuyruk yöneticisi stanza içindeki *DataPath* değişkeninde saklanır. If you create a queue manager without setting the -md *filepath* parameter, the queue manager directories are created in the path stored in *DefaultPrefix*, and the path is copied into the *Önek* variable in the queue manager stanza of the mq.s.ini file.

Çizelge 20. <i>DataPath</i> ve <i>Prefix/Qmgrs/QmgrName</i> dizinlerindeki dizinler ve dosyalar	
Dosya yolu	İçindekiler
<i>DataPath</i> \@ipcc	AMQCLCHL . TABİçin varsayılan konum, istemci bağlantı tablosu.
<i>DataPath</i> \authinfo	Dahili olarak kullanılır.
<i>DataPath</i> \channel	
<i>DataPath</i> \clntconn	
<i>DataPath</i> \errors	Hata günlükleri, AMQERRnn . LOG
<i>DataPath</i> \listener	Dahili olarak kullanılır.
<i>DataPath</i> \namelist	
<i>DataPath</i> \plugcomp	
<i>DataPath</i> \procdef	
<i>DataPath</i> \qmanager	
<i>DataPath</i> \queues	
<i>DataPath</i> \services	
<i>DataPath</i> \ssl	
<i>DataPath</i> \startpim	
<i>DataPath</i> \topic	
<i>DataPath</i> \active	
<i>DataPath</i> \active.dat	
<i>DataPath</i> \amqalchk.fil	
<i>DataPath</i> \master	
<i>DataPath</i> \master.dat	
<i>DataPath</i> \qm.ini	Kuyruk yöneticisi yapılanışı
<i>DataPath</i> \qmstatus.ini	Kuyruk yöneticisi durumu
<i>Prefix</i> \Qmgrs\QmgrName	Dahili olarak kullanılır
<i>Prefix</i> \Qmgrs\@SYSTEM	Kullanılmadı
<i>Prefix</i> \Qmgrs\@SYSTEM\errors	

IBM WebSphere MQ ve UNIX System V IPC kaynakları

Kuyruk yöneticisi bazı IPC kaynaklarını kullanır. Kullanılan kaynakların ne olduğunu öğrenmek için **ipcs -a** 'i kullanın.

Bu bilgiler, yalnızca UNIX and Linux sistemlerinde çalışan IBM WebSphere MQ için geçerlidir.

IBM WebSphere MQ , sistem bileşenleri arasında veri depolamak ve aktarmak için System V interprocess communication (IPC) kaynaklarını (*semafores* ve *shared memory parsees*) kullanır. Bu kaynaklar kuyruk yöneticisi işlemleri ve kuyruk yöneticisine bağlanan uygulamalar tarafından kullanılır. IBM WebSphere MQ MQI istemcileri, IBM WebSphere MQ izleme denetimi dışında, IPC kaynaklarını kullanmaz. Makinede kullanılmakta olan IPC kaynaklarının sayısı ve büyüklüğü ile ilgili tam bilgi almak için UNIX komutunu **ipcs -a** kullanın.

AIX üzerinde paylaşılan bellek

Bazı uygulama tipleri bir AIX bellek sınırlaması nedeniyle bağlanmayı başaramazsa, çoğu durumda bu durum, EXTSHM=ON ortam değişkeni tanımlanarak çözülebilir.

AIX üzerinde bazı 32 bit işlemler, WebSphere MQ kuyruk yöneticilerine bağlanma yeteneğini etkileyen bir işletim sistemi sınırlamasıyla karşılaşabilirler. WebSphere MQ ' ya yapılan her standart bağlantı, paylaşılan belleği kullanır; ancak, diğer UNIX and Linux platformlarının aksine, AIX , 32 bit süreçlerin yalnızca 11 paylaşılan bellek kümesini bağlamasına olanak tanır.

32 bit 'in çoğu bu sınırla karşılaşmıyor, ancak yüksek bellek gereksinimlerine sahip uygulamalar, 2102 neden koduyla WebSphere MQ ' ya bağlanmayı başaramayabilir: MQRC_RESOURCE_PROBLEM. Aşağıdaki uygulama tipleri bu hatayı görebilir:

- 32 bit Java sanal makinelerinde çalışan programlar
- Büyük ya da çok büyük bellek modellerini kullanan programlar
- Birden çok kuyruk yöneticisine ya da veritabanına bağlanan programlar
- Kendi başına paylaşılan bellek kümelerine bağlanan programlar

AIX , daha fazla paylaşılan bellek eklemelerine olanak sağlayan 32 bitlik işlemler için genişletilmiş bir paylaşılan bellek özelliği sunar. Bu özelliğe sahip bir uygulamayı çalıştırmak için, kuyruk yöneticilerinize ve programınıza başlamadan önce EXTSHM=ON ortam değişkenini dışa aktarın. EXTSHM=ON özelliği, bu hatanın çoğu durumda olmasını önler, ancak shmctl işlevinin SHM_SIZE seçeneğini kullanan programlarla uyumsuz.

WebSphere MQ MQI istemci uygulamaları ve 64 bit tüm süreçler bu sınırlamadan etkilenmez. EXTSHM ' nin ayarlanıp ayarlanmamasından bağımsız olarak WebSphere MQ kuyruk yöneticilerine bağlanabilirler.

WebSphere MQ ve UNIX Process Priority

Süreç önceliği *güzel* değerlerini ayarlarken iyi uygulamalar.

Bu bilgiler, yalnızca UNIX and Linux sistemlerinde çalışan WebSphere MQ için geçerlidir.

Arka planda bir işlem çalıştırırsanız, bu işlem, çağırın kabuk tarafından daha yüksek bir *güzel* değeri (ve dolayısıyla daha düşük öncelik) verilebilir. Bu, genel WebSphere MQ performans etkisinerine sahip olabilir. Yüksek düzeyde stresli durumlarda, daha yüksek bir öncelikte ve daha düşük öncelikte çok sayıda hazır çalıştırma iş parçacığı varsa, işletim sistemi zamanlama özellikleri, daha düşük öncelikli iş parçacıklarını işlemci süresinde mahrum edebilir.

It is good practice that independently started processes associated with queue managers, such as `runmqtsr`, have the same *güzel* values as the queue manager they are associated with. Kabuğun bu arka plan süreçlerine daha yüksek bir *güzel* değeri atamadığından emin olun. Örneğin, ksh 'de, artalan işlemlerinin *güzel* değerini yükseltmesini durdurmak için "`set +o bgnice`" ayarını kullanın. Bir "`ps -ef`" listesinin *NI* sütununu inceleyerek, çalışan süreçlerin *güzel* değerlerini doğrulayabilirsiniz.

Ayrıca, kuyruk yöneticisi olarak aynı *güzel* değerine sahip WebSphere MQ uygulama işlemlerini de başlatın. Farklı *güzel* değerleriyle çalıştırılırsa, bir uygulama iş parçacığı, bir kuyruk yöneticisi iş parçacıklarını engelleyebilir ya da tam tersi şekilde performansın düşmesine neden olabilir.

HP Integrity NonStop Server üzerinde IBM WebSphere MQ istemci ortamının planlanması

IBM WebSphere MQ ortamınızı planlarken, HP Integrity NonStop Server ortamını ve HP NonStop TMF 'yi göz önünde bulundurmanız gerekir. Use the information to plan the environment where IBM WebSphere MQ client for HP Integrity NonStop Server runs.

Before you plan your IBM WebSphere MQ client for HP Integrity NonStop Server architecture, familiarize yourself with the basic IBM WebSphere MQ client for HP Integrity NonStop Server concepts, see the topics in [IBM WebSphere MQ client for HP Integrity NonStop Server technical overview](#).

HP Integrity NonStop Server ortamını hazırlama

Kuruluştan önce, kuruluş programının hemen doğrulanıp doğrulanmadığına bağlı olarak ortam hazırlanmalıdır.

Kuruluş için aşağıdaki öğelere gereksinim duyarsınız:

- Gereksinimleri karşılayan bir kullanıcı kimliği. Kullanıcı kimliği gereksinimleriyle ilgili ayrıntılar için bkz. [HP Integrity NonStop Server üzerinde kullanıcı ve grup kurma](#).
- Kuruluş dosyaları için olabilecek OSS ve Guardian dosya sistemlerindeki doğrulanmış konumlar.
- Operasyonel bir OSS kabuk ve OSS dosya sistemi. Dosya sistemini aşağıdaki görevleri gerçekleştirerek doğrulayabilirsiniz:
 - OSS ortamında oturum açın (kabuk). Kullanmayı planladığınız OSS kuruluş kökü dizinine yazma erişiminiz olduğunu doğrulayın.
 - MQM grubundaki kullanıcı kimliğini kullanarak ACL ortamındaki oturum açın. Kullanmayı planladığınız birimin, gereksinimleri karşıladığını ve sizin için erişilebilir olduğunu ve alt birimin var olmadığını doğrulayın.

Bir diğer adı kullanarak hem OSS 'de, hem de ACL' de oturum açabilirsiniz; örneğin, bir ad ya da tam müdürün.

Kuruluşun kullanılabilir olduğunu doğrulamak için hemen devam etmek istiyorsanız, aşağıdaki isteğe bağlı öğelere de gereksinim duyabilirsiniz:

- OSS ortamındaki işletimsel ve erişilebilir bir Yerel Yuvalar altsistemi.
- Bir işletim TCP/IP altsistemi.

TMAF eşgüdümümlü genel iş birimlerini kullanmayı amaçlıyorsanız, aşağıdaki öğelere gereksinim duyarsınız:

- Bir operasyonel TMF altsistemi.
- Operasyonel Yol (TS/MP) altsistemi.

Bu kritik altsistemlerin durumu konusunda herhangi bir şüpheye yer alıyorsanız, sistem denetimcinizle birlikte çalışın.

IBM WebSphere MQ ve HP NonStop TMF

IBM WebSphere MQ client for HP Integrity NonStop Server can participate in HP NonStop Transaction Management Facility (HP NonStop TMF) coordinated units of work. HP NonStop TMF ile işlemlerin eşgüdümleme işlemi yalnızca, kuyruk yöneticisinin IBM WebSphere MQ Version 7.1 ya da sonraki yayın düzeyinde olduğu durumlarda desteklenir.

IBM WebSphere MQ tarafından sağlanan TMF/Gateway, uzak kuyruk yöneticisiyle iletişim kurmak için TMF koordinasyonundan eXtended Architecture (XA) hareket koordinasyonuna yönelik hareketleri dönüştürür. The IBM WebSphere MQ provided TMF/Gateway is the bridge between TMF and queue manager transactions, using the services provided by HP NonStop TMF, and has been designed to run in a Pathway environment.

HP NonStop TMF yazılımı, zorlu ortamlarda hareket koruması ve veritabanı tutarlılığı sağlar. HP NonStop TMF 'ye ilişkin ek bilgi için bkz. [HP NonStop TMF Introduction](#).

IBM WebSphere MQ tarafından sağlanan TMF/Gateway 'in nasıl yapılandırılacağı hakkında bilgi için bkz. [Configuring HP Integrity NonStop Server](#).

HP NonStop TMF ' yi kullanma

HP NonStop Transaction Management Facility (TMF), HP Integrity NonStop Server üzerindeki yerel hareket yöneticidir ve dosya sistemiyle ve ilişkisel veri tabanı yöneticileriyle, SQL/MP ve SQL/MX ile tümleştirilir.

HP Integrity NonStop Server için IBM WebSphere MQ istemcisi, genel iş birimlerini koordine etmek için TMF ' yi kullanabilir.

Genel iş birimlerinin koordinasyonunu sağlamak için TMF, hareket yöneticisi olarak işlev görür ve uygulama, genel iş birimlerini başlatmak, kesinleştirmek ve yedeklemek için TMF tarafından sağlanan API ' yi kullanmalıdır. Bir uygulama, BEGINTRANSACTION çağrılarak genel bir iş birimi başlatır ve ardından, syncpoint denetimi içinde MQPUT, MQPUT1 ve MQGET çağrıları yayınlayarak genel iş birimi içindeki IBM WebSphere MQ kaynaklarını günceller. Daha sonra uygulama, ENDTRANSACTION çağrılarak genel iş birimini kesinleştirebilir ya da ABORTTRANSACTION çağrılarak bu iş birimini geri verebilir.

TMF işlemlerini kullanan bir uygulama herhangi bir zamanda tek bir hareket üzerinde etkin bir şekilde çalışabilir; ancak RESUMETRANSACTION, bir uygulamanın etkin bir hareketten diğerine geçişine ya da önceden etkin hareketi tamamlamadan ya da iptal etmeden, TMF hareketiyle ilişkilendirilmesine olanak sağlar. MQPUT, MQPUT1 ya da MQGET çağrıları şu anda etkin olan TMF işlemi altında (varsa) ya da varsa yerel iş birimini (varsa) içerir. Bu nedenle, bu çağrıların doğru iş birimi içinde yapılmasından emin olmak için uygulama içinde dikkatli bir şekilde bulunmanız gerekir.

Bir genel iş biriminin yanı sıra, IBM WebSphere MQ kaynaklarının güncellenmesi sırasında bir uygulama Enscribe dosyalarını, SQL/MP veritabanlarını ya da SQL/MX veritabanlarını güncelleyebilir.

Genel iş birimlerinin kullanılması

Genel iş birimi, TMF işlemi olarak uygulanır. Bir uygulama, BEGINTRANSACTION çağrılarak genel bir iş birimi başlatır ve ENDTRANSACTION çağrılarak iş birimini kesinleştirir ya da ABORTTRANSACTION çağrılarak iş birimini yedekler. Bir uygulama, diğer TMF API çağrıları da kullanabilir.

Bir uygulama, TMF işlemini başka bir uygulamadan devralabilir. Örneğin, bir uygulama (ilk uygulama) işlemi yanıtlamadan önce hareket içinde yürütebilir ve daha sonra işlenmek üzere ikinci bir uygulamaya geri aktarılabilir. Bu nedenle, hem birinci hem de ikinci uygulamalar, IBM WebSphere MQ kuyruklarının güncellenmesi ve dosyalara ve veritabanlarına ilişkin güncellemelere ilişkin güncellemeleri içeren aynı genel iş birimine katılabilirler. Bir TMF hareketini uygulamalar arasında geçirebilme yeteneği, birden çok IBM WebSphere MQ uygulamasının aynı küresel iş birimi içinde ileti alışverişi işlemlerini gerçekleştirebileceği anlamına gelir.

Bir uygulama, birden çok etkin TMF işlemi aynı anda yönetebilir ve denetleyebilir. İşlemler uygulamanın kendisi tarafından başlatılabilir ya da diğer uygulamalardan devralınabilir ya da her ikisi birden olabilir. Bu, bir uygulamanın birden çok genel iş birimine aynı anda katılabileceği anlamına gelir.

İşlem başına koşut zamanlı etkin TMF işlemi sayısı üst sınırı 1000 'dir (bu da bir mimari sınırdır). Bir uygulama birden çok TMF işlemi yönetiyorsa, herhangi bir zamanda yalnızca bir işlem geçerli olabilir. Diğer bir seçenek olarak, hareketlerin hiçbiri geçerli olamaz. Uygulama, geçerli olma durumunu başka bir işlemle diğerine taşımak ya da herhangi bir işlemin geçerli olmadığını belirtmek için RESUMETRANSACTION, ACTIVATERECIVETRANSID ve TMF_SET_TX_ID gibi TMF API çağrıları kullanabilir. Uygulama, bir ileti alışverişi işleminin yerel bir iş birimi, genel iş birimi içinde mi, yoksa syncpoint denetimi dışında mı gerçekleştirileceğini belirlemek için bu denetim düzeyini kullanır:

- Bir uygulama MQPUT, MQPUT1 ya da MQGET çağrılırsa, hiçbir TMF hareketi geçerli değilse, IBM WebSphere MQ çağrısı yerel bir iş birimi içinde işler.
- Uygulama, geçerli bir TMF işlemi olduğunda, bir uygulama MQPUT, MQPUT1 ya da MQGET ile syncpoint denetimi çağırıyorsa, IBM WebSphere MQ , yürürlükteki TMF işlemi tarafından uygulanan genel iş birimi içinde aramayı işler.

- If an application calls MQPUT, MQPUT1, or MQGET outside of syncpoint control, IBM WebSphere MQ processes the call outside of syncpoint control, irrespective of whether the application has a current TMF transaction at the time of the call.

IBM WebSphere MQ , işleme sırasında bir yazılım ya da donanım hatası ortaya çıktığında ve IBM WebSphere MQ ya da işletim sistemi işlemin veri bütünlüğünü korumak için yedeklenmesinin gerekli olduğunu belirlemesi dışında, bir MQI çağrısı sırasında bir uygulamanın TMF hareketinin durumunu hiçbir zaman değiştirmez. Her MQI çağrısı, uygulamaya denetim döndürülmeden hemen önce uygulamanın hareket durumunu geri yükler.

Uzun süren işlemlerden kaçma

TMF işlemlerinin birkaç saniyeden daha uzun süre etkin kaldığı uygulamaları tasarlamaktan kaçının. Uzun süren işlemler, TMF ' nin dairesel denetim izinin doldurmasına neden olabilir. TMF kritik bir sistem genelindeki kaynaksa, TMF, uzun süre etkin olan uygulama işlemlerini yedekleyerek kendisini korur.

Bir uygulamadaki işlemlerin bir kuyruktan ileti olarak yönlendirdiğini ve uygulamanın kuyruktan ileti aldığını ve bir iş birimi içinde iletiyi işlediğini varsayın. Tipik olarak, bir uygulama kuyruktan ileti almak için MQGET ' yi bekleme seçeneğiyle ve syncpoint denetimi içinde çağırır.

Uygulama genel bir iş birimi kullanıyorsa, uzun süren bir işlemi önlemek için, MQGET çağrısının belirtilen bekleme aralığı kısa olmalıdır. Bu, uygulamanın bir iletiyi almadan önce MQGET çağrısını bir kereden fazla yayınlamaya gereksinim duyacak anlamına gelir.

Özel notlar

Bu belge, ABD'de kullanıma sunulan ürünler ve hizmetler için hazırlanmıştır.

IBM, bu belgede sözü edilen ürün, hizmet ya da özellikleri diğer ülkelerde kullanıma sunmayabilir. Bulduğunuz yerde kullanıma sunulan ürün ve hizmetleri yerel IBM müşteri temsilcisinden ya da çözüm ortağınızdan öğrenebilirsiniz. Bir IBM ürün, program ya da hizmetine gönderme yapılması, açık ya da örtük olarak yalnızca o IBM ürünü, programı ya da hizmetinin kullanılabilirliğini göstermez. Aynı işlevi gören ve IBM'in fikri mülkiyet haklarına zarar vermeyen herhangi bir ürün, program ya da hizmet de kullanılabilir. Ancak, IBM dışı ürün, program ya da hizmetlerle gerçekleştirilen işlemlerin değerlendirilmesi ve doğrulanması kullanıcının sorumluluğundadır.

IBM'in, bu belgedeki konularla ilgili patentleri ya da patent başvuruları olabilir. Bu belgenin size verilmiş olması, patentlerin izinsiz kullanım hakkının da verildiği anlamına gelmez. Lisansla ilgili sorularınızı aşağıdaki adrese yazabilirsiniz:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Çift byte (DBCS) bilgilerle ilgili lisans soruları için, ülkenizdeki IBM'in Fikri Haklar (Intellectual Property) bölümüyle bağlantı kurun ya da sorularınızı aşağıda adrese yazın:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japonya

Aşağıdaki paragraf, İngiltere ya da bu tür hükümlerin yerel yasalarla uyuşmadığı diğer ülkelerde geçerli değildir: INTERNATIONAL BUSINESS MACHINES CORPORATION BU YAYINI, HAK İHLALİ YAPILMAYACAĞINA DAİR GARANTİLERLE TİCARİLİK VEYA BELİRLİ BİR AMACA UYGUNLUK İÇİN ZİMNİ GARANTİLER DE DAHİL OLMAK VE FAKS BUNLARLA SINIRLI OLMAMAK ÜZERE AÇIK YA DA ZİMNİ HİÇBİR GARANTİ VERMEKSİZİN "OLDUĞU GİBİ" ESASIYLA SAĞLAMAKTADIR. Bazı ülkeler bazı işlemlerde garantinin açık ya da örtük olarak reddedilmesine izin vermez; dolayısıyla, bu bildirim sizin için geçerli olmayabilir.

Bu yayın teknik yanlışlar ya da yazım hataları içerebilir. Buradaki bilgiler üzerinde düzenli olarak değişiklik yapılmaktadır; söz konusu değişiklikler sonraki basımlara yansıtılacaktır. IBM, önceden bildirimde bulunmaksızın, bu yayında açıklanan ürünler ve/ya da programlar üzerinde iyileştirmeler ve/ya da değişiklikler yapabilir.

Bu belgede IBM dışı Web sitelerine yapılan göndermeler kullanıcıya kolaylık sağlamak içindir ve bu Web sitelerinin onaylanması anlamına gelmez. Bu Web sitelerinin içerdiği malzeme, bu IBM ürününe ilişkin malzemenin bir parçası değildir ve bu tür Web sitelerinin kullanılmasının sorumluluğu size aittir.

IBM'e bilgi ilettiğinizde, IBM bu bilgileri size karşı hiçbir yükümlülük almaksızın uygun gördüğü yöntemlerle kullanabilir ya da dağıtabilir.

(i) Bağımsız olarak yaratılan programlarla, bu program da içinde olmak üzere diğer programlar arasında bilgi değiş tokuşuna ve (ii) değiş tokuş edilen bilginin karşılıklı kullanımına olanak sağlamak amacıyla bu program hakkında bilgi sahibi olmak isteyen lisans sahipleri şu adrese yazabilirler:

IBM Corporation
Yazılım Birlikte Çalışabilirlik Koordinatörü, Bölüm 49XA
3605 Highway 52 N

Rochester, MN 55901
U.S.A.

Bu tür bilgiler, ilgili kayıt ve koşullar altında ve bazı durumlarda bedelli olarak edinilebilir.

Bu belgede açıklanan lisanslı program ve bu programla birlikte kullanılacak tüm lisanslı malzeme, IBM tarafından, IBM Müşteri Sözleşmesi, IBM Uluslararası Program Lisansı Sözleşmesi ya da eşdeğer herhangi bir sözleşmenin kayıt ve koşulları altında sağlanır.

Burada belirtilen performans verileri denetimli bir ortamda elde edilmiştir. Bu nedenle, başka işletim ortamlarında çok farklı sonuçlar alınabilir. Bazı ölçümler geliştirilme düzeyindeki sistemlerde yapılmıştır ve bu ölçümlerin genel kullanıma sunulan sistemlerde de aynı olacağı garanti edilemez. Ayrıca, bazı sonuçlar öngörü yöntemiyle elde edilmiş olabilir. Dolayısıyla, gerçek sonuçlar farklı olabilir. Bu belgenin kullanıcıları, kendi ortamları için geçerli verileri kendileri doğrulamalıdır.

IBM dışı ürünlerle ilgili bilgiler, bu ürünleri sağlayan firmalardan, bu firmaların yayın ve belgelerinden ve genel kullanıma açık diğer kaynaklardan alınmıştır. IBM bu ürünleri sınamamıştır ve IBM dışı ürünlerle ilgili performans doğruluğu, uyumluluk gibi iddiaları doğrulayamaz. IBM dışı ürünlerin yeteneklerine ilişkin sorular, bu ürünleri sağlayan firmalara yöneltilmelidir.

IBM'in gelecekteki yönelim ve kararlarına ilişkin tüm bildirimler değişebilir ve herhangi bir duyuruda bulunulmadan bunlardan vazgeçilebilir; bu yönelim ve kararlar yalnızca amaç ve hedefleri gösterir.

Bu belge, günlük iş ortamında kullanılan veri ve raporlara ilişkin örnekler içerir. Örneklerin olabildiğince açıklayıcı olması amacıyla kişi, şirket, marka ve ürün adları belirtilmiş olabilir. Bu adların tümü gerçek dışıdır ve gerçek iş ortamında kullanılan ad ve adreslerle olabilecek herhangi bir benzerlik tümüyle rastlantıdır.

YAYIN HAKKI LİSANSI:

Bu belge, çeşitli işletim platformlarında programlama tekniklerini gösteren, kaynak dilde yazılmış örnek uygulama programları içerir. Bu örnek programları, IBM'e herhangi bir ödemede bulunmadan, örnek programların yazıldığı işletim altyapısına ilişkin uygulama programlama arabirimiyle uyumlu uygulama programlarının geliştirilmesi, kullanılması, pazarlanması ya da dağıtılması amacıyla herhangi bir biçimde kopyalayabilir, değiştirebilir ve dağıtabilirsiniz. Bu örnekler her koşul altında tüm ayrıntılarıyla sınanmamıştır. Dolayısıyla, IBM bu programların güvenilirliği, bakım yapılabilirliği ya da işlevleri konusunda açık ya da örtük güvence veremez.

Bu bilgileri elektronik kopya olarak görüntülediyseniz, fotoğraflar ve renkli resimler görünmeyebilir.

Programlama arabirimi bilgileri

Programlama arabirimi bilgileri (sağlandıysa), bu programla birlikte kullanılmak üzere uygulama yazılımları yaratmanıza yardımcı olmak üzere hazırlanmıştır.

Bu kitap, müşterinin IBM WebSphere MQ hizmetlerini edinmek üzere program yazmasına olanak tanıyan, amaçlanan programlama arabirimlerine ilişkin bilgiler içerir.

Ancak, bu bilgiler tanılama, değiştirme ve ayarlama bilgilerini de içerebilir. Tanılama, değiştirme ve ayarlama bilgileri, uygulama yazılımlarınızda hata ayıklamanıza yardımcı olur.

Önemli: Bu tanılama, değiştirme ve ayarlama bilgilerini bir programlama arabirimi olarak kullanmayın; bu, değişiklik söz konusu olduğunda kullanılır.

Ticari Markalar

IBM, IBM logosu, ibm.com, IBM Corporation 'ın dünya çapında birçok farklı hukuk düzeninde kayıtlı bulunan ticari markalarıdır. IBM ticari markalarının güncel bir listesini Web üzerinde "Telif hakkı ve ticari marka bilgileri" www.ibm.com/legal/copytrade.shtml adresinde bulabilirsiniz. Diğer ürün ve hizmet adları IBM'in veya diğer şirketlerin ticari markaları olabilir.

Microsoft ve Windows, Microsoft Corporation'ın ABD ve/veya diğer ülkelerdeki ticari markalarıdır.

UNIX, The Open Group şirketinin ABD ve diğer ülkelerdeki tescilli ticari markasıdır.

Linux, Linus Torvalds'ın ABD ve/ya da diđer ülkelerdeki tescilli ticari markasıdır.

Bu ürün, Eclipse Project (<http://www.eclipse.org/>) tarafından geliştirilen yazılımları içerir.

Java ve Java tabanlı tüm markalar ve logolar, Oracle firmasının ve/ya da iştiraklerinin markaları ya da tescilli markalarıdır.



Parça numarası:

(1P) P/N: