

9.0

*IBM Message Service Client for .NET*

**IBM**

**Not**

Bu bilgileri ve desteklediđi ürünü kullanmadan önce, "[Özel notlar](#)" sayfa 241 bölümündeki bilgileri okuyun.

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

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

© **Copyright International Business Machines Corporation 2007, 2023.**

# İçindekiler

<b>Message Service Client for .NET.....</b>	<b>5</b>
IBM Message Service Client for .NET' e giriş.....	5
İleti Sistemi Biçemleri.....	6
XMS nesne modeli.....	7
Nesnelerin öznitelikleri ve özellikleri.....	8
Denetlenen nesnelere.....	9
XMS ileti modeli.....	10
Uygulamaların daha yeni XMS sürümünü kullanmasını önleyen.....	12
İleti alışverişi sunucusu ortamının ayarlanması.....	12
IBM MQ kuyruk yöneticisine bağlanan bir uygulama için kuyruk yöneticisi ve aracısının yapılandırılması.....	13
Aracıya gerçek zamanlı bağlantı kullanan bir uygulama için aracı yapılandırılması.....	15
WebSphere Application Server ile bağlantı sağlayan bir uygulama için hizmet bütünleştirme veriyolunun yapılandırılması.....	16
XMS örnek uygulamalarının kullanılması.....	17
Örnek uygulamalar.....	17
Örnek uygulamaların çalıştırılması.....	18
.NET örnek uygulamalarının oluşturulması.....	19
XMS uygulamalarının geliştirilmesi.....	19
XMS uygulamaları yazılıyor.....	19
Writing XMS .NET applications.....	42
Yönetilen nesnelere çalışma.....	48
XMS uygulamaları için iletişimi güvenlik altına alma.....	63
XMS ileti.....	66
Sorun giderme.....	79
.NET uygulamaları için izleme yapılandırması.....	79
.NET uygulamaları için FFDC yapılandırması.....	83
Sorun gidermeye ilişkin ipuçları.....	83
Message Service Clients for .NET başvurusu.....	85
.NET Arabirimler.....	85
XMS nesnelere özellikleri.....	172
<b>Özel notlar.....</b>	<b>241</b>
Programlama arabirimi bilgileri.....	242
Ticari Markalar.....	242



## IBM Message Service Client for .NET' e giriş

---

IBM Message Service Client for .NET provides an application programming interface (API) called XMS that has the same set of interfaces as the Java Message Service (JMS) API. IBM Message Service Client for .NET contains a fully managed implementation of XMS, which can be used by any .NET compliant language.

XMS Destek:

- noktadan noktaya iletişim İleti sistemi
- yayınlama/abone olma İleti sistemi
- Zamanuyumlu ileti teslimi
- Zamanuyumsuz ileti teslimi

Bir XMS uygulaması, aşağıdaki uygulama tipleriyle ileti alışverişi yapabilir:

- Bir XMS uygulaması
- Bir IBM MQ classes for JMS uygulaması
- Yerel bir IBM MQ uygulaması
- IBM MQ varsayılan ileti alışverişi sağlayıcısını kullanan bir JMS uygulaması

Bir XMS uygulaması aşağıdaki ileti sistemi sunucularından herhangi birine bağlanabilir ve bu sunucularla bağlantı kurabilirler:

### **IBM MQ Kuyruk Yöneticisi**

Uygulama bağ tanımlarında ya da istemci kipinde bağlanabilir.

### **WebSphere Application Server Service Integration Bus**

Uygulama doğrudan TCP/IP bağlantısı kullanabilir ya da TCP/IP üzerinden HTTP kullanabilir.

### **IBM Integration Bus**

İletiler, uygulama ile aracı arasında WebSphere MQ Real-time Transportile iletilir. İletiler WebSphere MQ Multicast Transportkullanılarak uygulamaya teslim edilebilir.

By connecting to an IBM MQ queue manager, an XMS application can use WebSphere MQ Enterprise Transport to communicate with IBM Integration Bus. Diğer bir seçenek olarak, bir XMS uygulaması IBM MQ' e bağlanarak yayınlatabilir ve abone olabilir.

### **İlgili kavramlar**

[“İleti Sistemi Biçemleri” sayfa 6](#)

[“XMS nesne modeli” sayfa 7](#)

XMS API, nesne yönelimli bir arabirimdir. XMS nesne modeli, JMS 1.1 nesne modelini temel alır.

[“XMS ileti modeli” sayfa 10](#)

XMS ileti modeli, IBM MQ classes for JMS ileti modeliyle aynıdır.

## IBM Message Service Client for .NET' e giriş

---

IBM Message Service Client for .NET provides an application programming interface (API) called XMS that has the same set of interfaces as the Java Message Service (JMS) API. IBM Message Service Client for .NET contains a fully managed implementation of XMS, which can be used by any .NET compliant language.

XMS Destek:

- noktadan noktaya iletişim İleti sistemi
- yayınlama/abone olma İleti sistemi
- Zamanuyumlu ileti teslimi
- Zamanuyumsuz ileti teslimi

Bir XMS uygulaması, aşağıdaki uygulama tipleriyle ileti alışverişi yapabilir:

- Bir XMS uygulaması
- Bir IBM MQ classes for JMS uygulaması
- Yerel bir IBM MQ uygulaması
- IBM MQ varsayılan ileti alışverişi sağlayıcısını kullanan bir JMS uygulaması

Bir XMS uygulaması aşağıdaki ileti sistemi sunucularından herhangi birine bağlanabilir ve bu sunucularla bağlantı kurabilirler:

#### **IBM MQ Kuyruk Yöneticisi**

Uygulama bağ tanımlarında ya da istemci kipinde bağlanabilir.

#### **WebSphere Application Server Service Integration Bus**

Uygulama doğrudan TCP/IP bağlantısı kullanılabilir ya da TCP/IP üzerinden HTTP kullanılabilir.

#### **IBM Integration Bus**

İletiler, uygulama ile aracı arasında WebSphere MQ Real-time Transportile iletilir. İletiler WebSphere MQ Multicast Transportkullanılarak uygulamaya teslim edilebilir.

By connecting to an IBM MQ queue manager, an XMS application can use WebSphere MQ Enterprise Transport to communicate with IBM Integration Bus. Diğer bir seçenek olarak, bir XMS uygulaması IBM MQ' e bağlanarak yayınlayabilir ve abone olabilir.

#### **İlgili kavramlar**

[“İleti Sistemi Biçemleri” sayfa 6](#)

[“XMS nesne modeli” sayfa 7](#)

XMS API, nesne yönelimli bir arabirimdir. XMS nesne modeli, JMS 1.1 nesne modelini temel alır.

[“XMS ileti modeli” sayfa 10](#)

XMS ileti modeli, IBM MQ classes for JMS ileti modeiyle aynıdır.

## **İleti Sistemi Biçemleri**

---

XMS , ileti alışverişindeki noktadan noktaya iletişim ve yayınlama/abone olma stillerini destekler.

İleti alışverişi biçemleri de ileti alışverişi etki alanları olarak adlandırılır.

### **Noktadan noktaya iletişim İleti sistemi**

A common form of noktadan noktaya iletişim messaging uses queuing. En basit durumda, bir uygulama, örtük ya da belirtik olarak bir hedef kuyruğu belirleyerek başka bir uygulamaya ileti gönderir. Temeldeki ileti alışverişi ve kuyruğa alma sistemi, iletiyi gönderme uygulamasından alır ve iletiyi hedef kuyruğuna yönlendirir. Alma uygulaması iletiyi kuyruktan alabilir.

Temeldeki ileti sistemi ve kuyruğa alma sistemi IBM Integration Busiçeriyorsa, IBM Integration Bus iletinin bir iletiyi eşleyebilir ve iletinin kopyalarını farklı kuyruklara yönlendirebilir. Sonuç olarak, iletiyi birden çok uygulama alabiliyor. IBM Integration Bus ayrıca bir iletiyi dönüştürebilir ve ona veri ekleyebilirler.

noktadan noktaya iletişim ileti sisteminin önemli bir özelliği, bir uygulamanın ileti gönderdiğinde yerel bir kuyruğa ileti yerleştirmesi olabilir. Temeldeki ileti sistemi ve kuyruğa alma sistemi, iletinin hangi hedef kuyruğunu göndereceğini belirler. Alma uygulaması, iletiyi hedef kuyruktan alır.

### **Yayınla/abone ol İleti sistemi**

yayınlama/abone olma ileti sisteminde, iki tip uygulama vardır: yayınlayıcı ve abone.

Bir *yayınlayıcı* bilgi kaynağı, yayın iletileri biçiminde bilgi sağlar. Bir yayınlayıcı bir iletiyi yayınladığında, bu ileti, iletinin içindeki bilgilerin konusunu tanımlayan bir konu belirtir.

*abone* , yayınlanan bilgilerin tüketicidir. Abone, abonelikler yaratarak ilgilendiği konuları belirtir.

Yayınlama/abone olma sistemi, yayıncılardan ve aboneliklerden gelen abonelikleri alır. Yayınları abonelere yönlendiriyor. Bir abone, yalnızca abone olduğu konuların yayınlarını alır.

yayınla/abone olma ileti sisteminin önemli bir özelliği, yayıncının bir iletiyi yayınlarken bir konuyu tanıması. Bu, aboneleri tanımaz. Abonesi olmayan bir konuyla ilgili bir ileti yayınlanırsa, hiçbir uygulama iletiyi almaz.

Bir uygulama hem yayınlayıcı hem de bir abone olabilir.

## XMS nesne modeli

---

XMS API, nesne yönelimli bir arabirimdir. XMS nesne modeli, JMS 1.1 nesne modelini temel alır.

Aşağıdaki liste, ana XMS sınıflarını ya da nesne tiplerini özetlemektedir:

### **ConnectionFactory**

Bir `ConnectionFactory` nesnesi, bir bağlantı için bir parametre kümesini sarsalıyor. Uygulama, bağlantı yaratmak için `ConnectionFactory` kullanır. Bir uygulama çalışma zamanında parametreleri sağlayabilir ve bir `ConnectionFactory` nesnesi oluşturabilirler. Diğer bir seçenek olarak, bağlantı değiştirgeleri denetimli nesnelere havuzunda saklanabilir. Bir uygulama, havuzdan bir nesneyi alabilir ve ondan bir `ConnectionFactory` nesnesi yaratabilir.

### **Connection**

Bir `Connection` nesnesi, bir uygulamadan bir ileti alışverişi sunucusuna etkin bir bağlantıyı sokardır. Uygulama, oturum yaratmak için bir bağlantı kullanır.

### **Destination**

Bir uygulama, `Destination` nesnesini kullanarak ileti gönderir ya da ileti alır. yayınla/abone olma etki alanında, bir `Destination` nesnesi bir konuyu ve noktadan noktaya iletişim etki alanında bir `Destination` nesnesi, bir kuyruğu sarmalayan bir nesne içerir. Uygulama, yürütme sırasında `Destination` nesnesi yaratmak için gereken parametreleri sağlayabilir. Diğer bir seçenek olarak, yönetilen nesnelere havuzunda saklanan bir nesne tanımlamasından bir `Destination` nesnesi de yaratabilir.

### **Session**

`Session` nesnesi, ileti göndermek ve almak için tek bir iş parçacıklı bağlamdır. Uygulama, `Message`, `MessageProducer` ve `MessageConsumer` nesneleri oluşturmak için bir `Session` nesnesi kullanır.

### **Message**

Bir `Message` nesnesi, bir uygulamanın `MessageProducer` nesnesi kullanarak gönderdiği ya da bir `MessageConsumer` nesnesi kullanılarak aldığı `Message` nesnesini sarmadır.

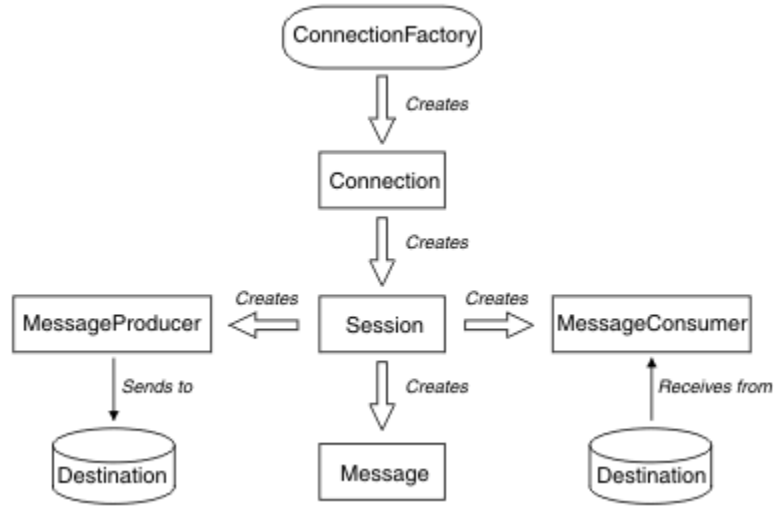
### **MessageProducer**

`MessageProducer` nesnesi, bir uygulamaya ileti göndermek için bir uygulama tarafından kullanılır.

### **MessageConsumer**

`MessageConsumer` nesnesi, bir hedef için gönderilen iletileri almak için bir uygulama tarafından kullanılır.

Şekil 1 sayfa 8 , bu nesnelere ve bunların ilişkilerini gösterir. Bu şekil, XMS nesnesinin ana tiplerini gösterir: `ConnectionFactory`, `Connection`, `Session`, `MessageProducer`, `MessageConsumer`, `Message`, `Destination`. Uygulama, bağlantı yaratmak için bir bağlantı üreticisi kullanır ve oturum yaratmak için bir bağlantı kullanır. Daha sonra, uygulama, ileti, ileti üreticileri ve ileti tüketicileri oluşturmak için bir oturumu kullanabilir. Uygulama, bir hedefe ileti göndermek için bir ileti üreticisi kullanır ve bir hedefe gönderilen iletileri almak için bir ileti tüketicisi kullanır.



Şekil 1. XMS nesneleri ve ilişkileri

.NET' ta XMS sınıfları, .NET arabirimleri kümesi olarak tanımlanır. When you are coding XMS .NET applications, you need only the declared interfaces.

XMS nesne modeli, Java Message Service Specification, Sürüm 1.1 içinde açıklanan etki alanı bağımsız arabirimlerine dayalıdır. Etki alanına özgü sınıflar ( Topic, TopicPublisher ve TopicSubscriber gibi) sağlanmaz.

## Nesnelerin öznitelikleri ve özellikleri

Bir XMS nesnesi, nesneye ilişkin özellikler olan öznitelikler ve özellikler olabilir; bu, farklı şekillerde uygulanan özelliklerdir.

### Öznitelikler

Öznitelik bir değer olmasa bile, her zaman var olan ve depolamayı kaplayan bir nesne karakteristiğidir. Bu bakımdan, öznitelik sabit uzunluklu veri yapısındaki bir alana benzer. Özniteliklerin ayırt edici özelliği, her özniteliğin değerini ayarlamak ve almak için kendi yöntemlerine sahip olması.

### Özellikler

Bir nesnenin özelliği var olur ve yalnızca değeri ayarlandıktan sonra depolanır. Bir özellik silinemez ya da değeri ayarlandıktan sonra depolama birimi kurtarılamaz. Değerini değiştirebilirsiniz. XMS , özellik değerlerini ayarlamak ve almak için bir dizi soysal yöntem sağlar.

### İlgili kavramlar

#### XMS temel tipleri

XMS , sekiz Java temel tipi (byte, short, int, long, float, double, double, char ve boolean) eşdeğerleri sağlar. This allows the interchange of messages between XMS and JMS without data becoming lost or corrupted.

Bir özellik değerinin bir veri tipinden diğerine örtük olarak dönüştürülmesi

Bir uygulama bir özelliğin değerini aldığı anda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Hangi dönüştürmelerin desteklendiğini ve XMS ' in dönüştürmeleri nasıl gerçekleştirdiğini birçok kural yönetir.

### İlgili başvurular

#### Uygulama verilerine ilişkin öğeler için veri tipleri

To ensure that an XMS application can exchange messages with an IBM MQ classes for JMS application, both the applications must be able to interpret the application data in the body of a message in the same way.



## Denetlenen nesnelere

Denetlenen nesnelere kullanarak, merkezi bir havuzdan yönetilecek istemci uygulamaları tarafından kullanılan bağlantı ayarlarını yönetebilirsiniz. Uygulama, merkezi havuzdan nesne tanımlarını alır ve bunları `ConnectionFactory` ve `Destination` nesnelere oluşturmak için kullanır. Yönetilen nesnelere kullanarak, çalıştırma sırasında kullandıkları kaynaklardan birkaç uygulama da yapabilirsiniz.

Örneğin, XMS uygulamaları, bir test ortamındaki bağlantılar ve hedefler kümesine başvuruda bulunan yönetilen nesnelere yazılıp test edilebilir. Uygulamalar konuşlandırıldığında, yönetilen nesnelere, uygulamaların üretim ortamındaki bağlantılara ve hedeflere gönderme yapma üzere yapılandırılacak şekilde değiştirilebilir.

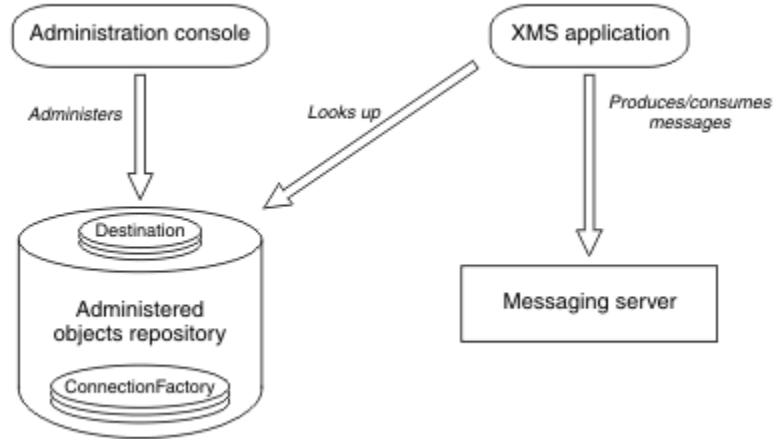
XMS, denetlenen iki nesneyi destekler:

- Sunucu ile ilk bağlantıyı yapmak için uygulamalar tarafından kullanılan bir `ConnectionFactory` nesnesi.
- Gönderilmekte olan iletilere ilişkin hedefi ve alınmakta olan iletilerin kaynağını belirlemek için uygulamalar tarafından kullanılan bir `Destination` nesnesi. Hedef, bir uygulamanın bağlı olduğu sunucudaki bir konu ya da kuyruğudur.

Yönetim aracı **JMSAdmin**, IBM MQ ile birlikte sağlanır. It is used to create and manage administered objects in a central repository of administered objects.

Havuzdaki denetlenen nesnelere, IBM MQ classes for JMS ve XMS uygulamaları tarafından kullanılabilir. XMS applications can use the `ConnectionFactory` and `Destination` objects to connect to an IBM MQ queue manager. Denetimci, havuzda tutulan nesne tanımlamalarını uygulama kodunu etkilemeden değiştirebilir.

Aşağıdaki çizgede, bir XMS uygulamasının tipik olarak yönetilen nesnelere nasıl kullandığı gösterilmektedir. Çizgenin sol tarafında, bir denetim konsolu kullanılarak denetlenen `ConnectionFactory` (`ConnectionFactory`) ve hedef nesne tanımlamaları içeren bir havuz gösterilir. Çizgenin sağ tarafı, havuzdaki nesne tanımlamalarını gösteren bir XMS uygulamasını gösterir ve ardından bir ileti alışverişini sunucusuna bağlanırken bu nesne tanımlamalarını kullanır.



Şekil 2. Typical use of administered objects by an XMS application

### İlgili kavramlar

Yönetilen nesnelere çalışma

Bu bölümdeki konular denetlenen nesnelere ilgili bilgi sağlar. XMS uygulamaları, merkezi yönetilen bir nesne havuzundan nesne tanımlarını alabilir ve bu nesnelere, bağlantı fabrikaları ve hedefler oluşturmak için kullanabilir.

Desteklenen yönetilen nesne havuzu tipleri

Dosya Sistemi ve LDAP denetimli nesnelere, IBM MQ ve WebSphere Application Server'e bağlanmak için kullanılabilir; ancak, COS Naming yalnızca WebSphere Application Server'e bağlanmak için kullanılabilir.

### İlgili görevler

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

## XMS ileti modeli

XMS ileti modeli, IBM MQ classes for JMS ileti modeliyle aynıdır.

Özellikle XMS , IBM MQ classes for JMS ' in uyguladığı ileti üstbilgi alanlarını ve ileti özelliklerini uygular:

- JMS üstbilgi alanları. Bu alanlar, JMS önekiyle başlayan adlara sahiptir.
- JMS tanımlı özellikler. Bu alanlarda, adları JMSX önekiyle başlayan özellikler vardır.
- IBM tanımlı özellikler. Bu alanlarda, adları JMS\_IBM\_ önekiyle başlayan özellikler vardır.

As a result, XMS applications can exchange messages with IBM MQ classes for JMS applications. Her iletilerde üstbilgi alanları ve özellikleri uygulama tarafından ayarlanır ve diğerleri XMS ya da IBM MQ classes for JMS tarafından ayarlanır. Some of the fields set by XMS or IBM MQ classes for JMS are set when the message is sent, and others when it is received. Üstbilgi alanları ve özellikleri, uygun olduğu durumlarda bir ileti alışverişi sunucusu aracılığıyla bir iletiliyle geçirilir. Bunlar, iletili alan herhangi bir uygulama tarafından kullanılabilir kılınabilirler.

## Kuruluş Sihirbazını kullanarak Message Service Client for .NET ' in kurulması

Kuruluş, bir InstallShield X/Windows MSI kuruluş programı kullanır. Eksiksiz ya da özel bir kuruluş seçebileceğiniz iki ayar seçeneği vardır.

### Bu görev hakkında

To install Message Service Client for .NET on Windows, follow this procedure.

### Yordam

1. SupportPac ' den kuruluş yapıyorsanız aşağıdaki adımları tamamlayın ya da doğrudan ["2" sayfa 10](#) adımına geçin.
  - a) Windows' ta yönetici olarak oturum açın.
  - b) dotNETClientsetup.exe kuruluş programını çalıştırın.
2. Kuruluş Sihirbazının aşağıdaki iletiliye açmasını ve görüntülemesini bekleyin:

```
Welcome to IBM Message Service Client for .NET installation wizard
```

**İleri'**yi tıklayın.

Sihirbaz, size lisans sözleşmesini okumanızı isteyebilir.

3. Lisans sözleşmesini okumanız ve lisans sözleşmesinin koşullarını kabul etmeniz istenirse, **Lisans sözleşmesinde koşulları kabul ediyorum** seçeneğini tıklayın ve ardından **İleri** düğmesini tıklayın.

Kuruluş Sihirbazı, gereksinimlerinize en uygun kuruluş tipini seçmenizi ister.

4. Gereksinim duyduğunuz kuruluş tipini seçin:

- Tüm program özelliklerini kurmak ve bunları varsayılan kuruluş dizinine kurmak için **Complete** (Tamamlandı) seçeneğini tıklayın.
- Hangi özellikleri kurmak istediğinizi seçmek ve nereye kurulacaklarını belirlemek için **Özel** ' i tıklayın.

## 5. İleri'yi tıklatın.

If you select the complete installation option, the installation wizard displays a message that it is ready to begin installation, as described in step “8” sayfa 11. Özel kuruluş seçeneğini belirlerseniz, kuruluş sihirbazı kurmak istediğiniz özellikleri seçmenizi ister ve “8” sayfa 11 adımına geçmeden önce “6” sayfa 11 adımını ve “7” sayfa 11 adımını tamamlamanız gerekir.

6. For a custom installation only, click an icon in the list of features to specify any changes to how you want the Message Service Client for .NET features to be installed. If you do not want to install Message Service Client for .NET in the directory suggested, choose another directory.

Message Service Client for .NET 'i şu anda var olmayan bir dizine kurmayı seçerseniz, kuruluş sihirbazı sizin için dizini oluşturur.

XMS uygulamalarını geliştirmek istiyorsanız, **Geliştirme Araçları ve Örnekler** özelliğinin seçili olduğundan emin olun. Bu özellik örnek uygulamaları, kitaplıkları ve .NET uygulamalarını derlemek için gereken diğer dosyaları sağlar. If you do not select this feature, only the files required to run XMS applications are installed.

7. Özel kuruluş seçeneğini kullanıyorsanız, “6” sayfa 11 adımında açıklandığı gibi gereksinim duyduğunuz seçenekleri seçtikten sonra **Next** (İleri) düğmesini tıklatın.

Kuruluş sihirbazı, kuruluşu başlatmaya hazır olduğunu bildiren bir ileti görüntüler.

8. Kuruluşu başlatmak için **Kur** düğmesini tıklatın.

Kuruluş sihirbazında, kuruluşun ilerleyişini gösteren bir çubuk görüntülenir. İlerleme çubuğunun tamamlanmak için bekleyin. Kuruluş başarıyla tamamlandığında, pencerede aşağıdaki ileti görüntülenir:

```
The installation wizard has successfully installed IBM Message Service Client for .NET. Click Finish to exit the wizard.
```

9. Kuruluş sihirbazını kapatmak için **Son** düğmesini tıklatın.

## Sonuçlar

Kullanıma hazır olan Message Service Client for .NET ürününü başarıyla kurdun.

## Sonraki adım

XMS ile birlikte verilen örnek uygulamalar da içinde olmak üzere, herhangi bir XMS uygulamasını çalıştırmadan önce, ileti alışverişi sunucusu ortamını ayarlamanız gerekir; ayrıntılar için bkz: “İleti alışverişi sunucusu ortamının ayarlanması” sayfa 12.

### İlgili kavramlar

#### JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

#### İleti alışverişi sunucusu ortamının ayarlanması

The topics in this section describe how to set up the messaging server environment to allow XMS applications to connect to a server.

#### XMS örnek uygulamalarının kullanılması

Kuruluş ve ileti alışverişi sunucusu kurulumunuzu doğrulamak ve kendi uygulamalarınızı oluşturmanıza yardımcı olmak için XMS ile birlikte verilen örnek uygulamaları kullanın. Örnekler, her bir API 'nin ortak özelliklerine genel bir bakış sağlar.

## WebSphere MQ' a bağlanan XMS uygulamaları için önkoşullar

Bazı önkoşullar, XMS uygulamanız WebSphere MQ olanağına bağlıysa geçerlidir.

Bir WebSphere MQ kuyruk yöneticisine bağlanan uygulamalar için, XMS uygulamasını çalıştırmak için kullandığınız makineye uygun WebSphere MQ istemci kitaplıklarını kurmanız gerekir. Bu kitaplıklar, yerel bir kuyruk yöneticisine sahip makinelerde önceden kuruludur.

.NET için XMS istemcisi için, IBM WebSphere MQ 7.0.1.0 ya da daha sonraki bir yayın düzeyiyle birlikte gönderilen istemci kitaplıklarını kullanın. Bunlar *.NET için WebSphere MQ sınıfları*. They enable client mode connections to IBM WebSphere MQ 7.0, 6.0, and 5.3 queue managers, and bindings mode connections to a local queue manager, if it is also IBM WebSphere MQ 7.0.1.0 or later.

Microsoft .NET Framework 2.0 Redistributable Package, XMS ' in kurulacağı bilgisayarda kurulu olmalıdır. Bu paket kullanılabilir değilse, XMS kurulumu başarısız olur. You then need to quit the installation procedure, install Microsoft .NET Framework 2.0 Redistributable Package on your computer, and rerun the installation procedure.

On the Microsoft download site, you need to look for dotnetfx.exe for Microsoft .NET Framework 2.0 Redistributable Package (x86) and NetFx64.exe for Microsoft .NET Framework 2.0 Redistributable Package (x64), whichever is applicable.

### İlgili kavramlar

İleti alışverişi sunucusu ortamının ayarlanması

The topics in this section describe how to set up the messaging server environment to allow XMS applications to connect to a server.

## Uygulamaların daha yeni XMS sürümünü kullanmasını önleyen

Varsayılan olarak, daha yeni bir XMS sürümü kuruluysa, önceki sürümü kullanan uygulamalar, yeniden derlemeye gerek kalmadan otomatik olarak daha yeni sürüme geçer.

### Bu görev hakkında

Birden çok sürüm birlikte bulunma özelliği, daha yeni bir XMS sürümünün kurulmasının önceki XMS sürümünün üzerine yazmamasını sağlar. Bunun yerine, benzer XMS .NET düzeneklerinin birden çok eşgörünümü, Global Assembly Cache (GAC) içinde birlikte bulunur, ancak farklı sürüm numaralarına sahip olur. Dahili olarak GAC, uygulama çağrılarını en son XMS sürümüne yönlendirmek için bir ilke dosyası kullanır. Uygulamaların yeniden derlenmesi gerekmeksizin çalışır ve daha yeni XMS .NET sürümünde kullanılabilir olan yeni özellikleri kullanabilirler.

Ancak, daha eski XMS sürümünü kullanmak için bir uygulama gerekiyorsa, uygulama yapılandırma dosyasında `publisherpolicy` özniteliğini `no` olarak ayarlayın.

**Not:** Uygulama yapılandırma dosyası, dosyanın ilişkilendirdiği yürütülebilir programın adından, `.config` son ekiyle birlikte bir ad içeren bir dosyadır. For example, the application configuration file for `text.exe` would have the name `text.exe.config`.

Ancak, herhangi bir zamanda, bir sistemin tüm uygulamaları aynı XMS .NET sürümünü kullanır.

## İleti alışverişi sunucusu ortamının ayarlanması

The topics in this section describe how to set up the messaging server environment to allow XMS applications to connect to a server.

Bir IBM MQ kuyruk yöneticisine bağlanan uygulamalar için, IBM MQ istemcisi (ya da bağ tanımları kipine ilişkin kuyruk yöneticisi) gereklidir.

Şu anda bir aracıya gerçek zamanlı bağlantı kullanan uygulamalar için herhangi bir önkoşul yok.

You must set up the messaging server environment before running any XMS applications, including the sample applications provided with XMS.

Bu bölüm aşağıdaki konuları içerir:

- [“IBM MQ kuyruk yöneticisine bağlanan bir uygulama için kuyruk yöneticisi ve aracısının yapılandırılması” sayfa 13](#)

- “Aracıya gerçek zamanlı bağlantı kullanan bir uygulama için aracı yapılandırılması” sayfa 15
- “WebSphere Application Server ile bağlantı sağlayan bir uygulama için hizmet bütünleştirme veriyolunun yapılandırılması” sayfa 16

### İlgili kavramlar

#### JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

#### XMS örnek uygulamalarının kullanılması

Kuruluş ve ileti alışverişi sunucusu kurulumunuzu doğrulamak ve kendi uygulamalarınızı oluşturmanıza yardımcı olmak için XMS ile birlikte verilen örnek uygulamaları kullanın. Örnekler, her bir API 'nin ortak özelliklerine genel bir bakış sağlar.

### İlgili görevler

#### Kuruluş sihirbazını kullanarak Message Service Client for .NET 'in kurulması

Kuruluş, bir InstallShield X/Windows MSI kuruluş programı kullanır. Eksiksiz ya da özel bir kuruluş seçebileceğiniz iki ayar seçeneği vardır.

### İlgili başvurular

#### WebSphere MQ' a bağlanan XMS uygulamaları için önkoşullar

Bazı önkoşullar, XMS uygulamanız WebSphere MQolanağına bağlıysa geçerlidir.

## IBM MQ kuyruk yöneticisine bağlanan bir uygulama için kuyruk yöneticisi ve aracısının yapılandırılması

Bu bölümde, IBM WebSphere MQ 7.0.1ya da daha sonraki bir sürümü kullandığınız varsayılır. Bir IBM MQ kuyruk yöneticisine bağlanan bir uygulamayı çalıştırabilmeniz için, kuyruk yöneticisini yapılandırmanız gerekir. Bir yayınlama/abone olma uygulaması için Queued yayınlama/abone olma arabirimini kullanıyorsanız, bazı ek yapılandırmalara gerek vardır.

### Başlamadan önce

XMS , IBM Integration Bus ya da WebSphere Message Broker 6.1 ile çalışır ya da daha sonra

Bu görevi başlatmadan önce aşağıdaki adımları gerçekleştirdiniz:

- Uygulamanızın çalışmakta olan bir kuyruk yöneticisine erişimi olduğundan emin olun.
- Uygulamanız bir yayınlama/abone olma uygulamasıysa ve Queued yayınlama/abone olma arabirimini kullanıyorsa, kuyruk yöneticisinde "PSMODE" özneliğinin "ENABLE" olarak ayarlandığından emin olun.
- Uygulamanızın, özellikleri kuyruk yöneticisine bağlanmak için uygun şekilde ayarlanmış bir bağlantı üreticisi kullandığından emin olun. Uygulamanız bir yayınlama/abone olma uygulamasıysa, aracıyı kullanmak için uygun bağlantı üreticisi özelliklerinin ayarlandığından emin olun. Bir bağlantı üreticisinin özellikleri hakkında daha fazla bilgi için “[ConnectionFactoryözellikleri](#)” sayfa 174.

### Bu görev hakkında

You configure the queue manager and broker to run XMS applications in the same way that you configure the queue manager and queued publish/subscribe interface to run IBM MQ JMS applications. Aşağıdaki adımlar, yapmanız gerekenleri özetlemektedir.

### Yordam

1. Kuyruk yöneticisinde, uygulamanızın gereksinim duyduğu kuyrukları yaratın.

Kuyruklar yaratmanıza genel bir bakış için [Kuyrukların tanımlanması](#) başlıklı konuya bakın.

Uygulamanız bir yayınlama/abone olma uygulamasıysa ve IBM MQ classes for JMS sistem kuyruklarına erişmesi gereken Queued yayınlama/abone olma arabirimini kullanıyorsa, kuyrukları yaratmadan önce Adım 4a 'ya kadar bekleyin.

2. Uygulamanızla ilişkili kullanıcı kimliğine, kuyruk yöneticisine bağlanma yetkisi ve kuyruklara erişmek için uygun yetkiyi verin.

Yetkilendirmeye ilgili genel bilgi edinmek için [Securing](#) konusuna bakın. If your application connects to the queue manager in client mode, see also [İstemciler ve sunucular](#).

3. Uygulamanızın istemci kipindeki kuyruk yöneticisine bağlanması durumunda, kuyruk yöneticisinde bir sunucu bağlantı kanalı tanımlandığından ve bir dinleyici başlatıldığından emin olun.

Kuyruk yöneticisine bağlanan her uygulama için bu adımı gerçekleştirmenize gerek yoktur. Bir sunucu bağlantısı kanal tanımlaması ve bir dinleyici, istemci kipinde bağlanan tüm uygulamaları destekleyebilir.

4. Uygulamanız bir yayınlama/abone olma uygulaması ise ve Queued yayınlama/abone olma arabirimini kullanıyorsa, aşağıdaki adımları gerçekleştirin.

- a) Kuyruk yöneticisiyle birlikte, IBM MQ ile birlikte verilen MQSC komutlarının komut dosyasını çalıştırarak IBM MQ classes for JMS sistem kuyruklarını yaratın. IBM Integration Bus ya da WebSphere Message Broker ile ilişkili kullanıcı kimliğinin, kuyruklara erişme yetkisine sahip olduğundan emin olun.

Komut dosyasının nerede bulacağı ve nasıl çalıştırılacağı hakkında bilgi için bkz. [IBM MQ classes for Javakomutunu kullanma](#).

Bu adımı kuyruk yöneticisi için yalnızca bir kez gerçekleştirin. IBM MQ classes for JMS sistem kuyrukları kümesi, kuyruk yöneticisine bağlanan tüm XMS ve IBM MQ classes for JMS uygulamalarını destekleyebilir.

- b) Uygulamanızla ilişkili kullanıcı kimliğine, IBM MQ classes for JMS sistem kuyruklarına erişme yetkisi verin.

Kullanıcı kimliği gereksinimlerinin ne olduğu hakkında bilgi için bkz. [IBM MQ classes for JMSkomutunu kullanma](#).

- c) For a broker of IBM Integration Bus or WebSphere Message Broker, create and deploy a message flow to service the queue where applications send messages that they publish.

Temel ileti akışı, iletileri yayınlamak için yayınlanan iletileri ve bir Yayın iletisi işleme düğümünü okumak için bir MQInput iletisi işleme düğümlerinden oluşur.

For information about how to create and deploy a message flow, see the IBM Integration Bus or WebSphere Message Broker product documentation available from the [IBM Integration Bus ürün belgeleri kitaplığı web sayfası](#).

Aracıda uygun bir ileti akışı önceden konuşlandırıldıysa, bu adımı gerçekleştirmenize gerek yoktur.

## Sonuçlar

Artık uygulamanıza başlayabilirsiniz.

### İlgili görevler

[Aracıya gerçek zamanlı bağlantı kullanan bir uygulama için aracı yapılandırılması](#)

Bir aracıya gerçek zamanlı bağlantı kullanan bir uygulamayı çalıştırabilmeniz için, o aracıyı yapılandırmanız gerekir.

[WebSphere Application Server ile bağlantı sağlayan bir uygulama için hizmet bütünleştirme veriyolunun yapılandırılması](#)

Bir WebSphere Application Server service integration technologies hizmet bütünleştirme veriyoluna bağlanan bir uygulamayı çalıştırabilmeniz için, hizmet bütünleştirmesini, varsayılan ileti alışverişi sağlayıcısını kullanan JMS uygulamalarını çalıştırmak için hizmet tümleştirme veriyolunu yapılandırdığınız şekilde yapılandırmalısınız.

## Aracıya gerçek zamanlı bağlantı kullanan bir uygulama için aracı yapılandırılması

Bir aracıya gerçek zamanlı bağlantı kullanan bir uygulamayı çalıştırabilmeniz için, o aracıyı yapılandırmanız gerekir.

### Başlamadan önce

Bu görevi başlatmadan önce aşağıdaki adımları gerçekleştirdiniz:

- Uygulamanızın çalışmakta olan bir aracıya erişiminin olduğundan emin olun.
- Uygulamanızın özellikleri, bir aracıyla gerçek zamanlı bağlantı için uygun şekilde ayarlanmış bir bağlantı üreticisi kullandığından emin olun. Bir bağlantı üreticisinin özellikleri hakkında daha fazla bilgi için bkz. [“ConnectionFactoryözellikleri” sayfa 174.](#)

### Bu görev hakkında

You configure a broker to run XMS applications in the same way that you configure a broker to run IBM MQ classes for JMS applications. Yapmanız gerekenleri özetlemek için aşağıdaki adımları gerçekleştirin:

### Yordam

1. Bir aracının dinlediği TCP/IP kapısındaki iletileri okumak ve iletileri yayınlamak için ileti akışı oluşturun ve konuşlandırın.

Bunu aşağıdaki yollardan biriyle de yapabilirsiniz:

- **Real-timeOptimizedFlow** ileti işleme düğümü içeren bir ileti akışı yaratın.
- Bir **Real-timeInput** ileti işleme düğümü ve bir Yayın iletisi işleme düğümü içeren bir ileti akışı yaratın.

Gerçek zamanlı bağlantılar için kullanılan bağlantı noktasını dinlemek için **Real-timeOptimizedFlow** ya da **Real-timeInput** düğümünü yapılandırmalısınız. XMS' ta, gerçek zamanlı bağlantılar için varsayılan kapı numarası 1506 'tır.

Aracıda uygun bir ileti akışı önceden konuşlandırıldıysa, bu adımı gerçekleştirmenize gerek yoktur.

2. If you require messages to be delivered to your application using IBM MQ classes for JMS, configure the broker to enable multicast. Güvenilir çok noktaya yayın gerektiren konular için güvenilir bir hizmet kalitesi belirterek, çok hedefli olması gereken konuları yapılandırın.
3. Uygulamanız aracıya bağlandığında bir kullanıcı kimliği ve parola sağladıysa ve aracıyı bu bilgileri kullanarak uygulamanızın kimliğini doğrulamamasını istiyorsanız, basit Telnet benzeri parola kimlik doğrulaması için kullanıcı adı sunucusunu ve aracıyı yapılandırın.

### Sonuçlar

Artık uygulamanıza başlayabilirsiniz.

#### İlgili görevler

[IBM MQ kuyruk yöneticisine bağlanan bir uygulama için kuyruk yöneticisi ve aracısının yapılandırılması](#)

Bu bölümde, IBM WebSphere MQ 7.0.1ya da daha sonraki bir sürümü kullandığınız varsayılır.

Bir IBM MQ kuyruk yöneticisine bağlanan bir uygulamayı çalıştırabilmeniz için, kuyruk yöneticisini yapılandırmanız gerekir. Bir yayınlama/abone olma uygulaması için Queued yayınlama/abone olma arabirimini kullanıyorsanız, bazı ek yapılandırmalara gerek vardır.

[WebSphere Application Server ile bağlantı sağlayan bir uygulama için hizmet bütünleştirme veriyolunun yapılandırılması](#)

Bir WebSphere Application Server service integration technologies hizmet bütünleştirme veriyoluna bağlanan bir uygulamayı çalıştırabilmeniz için, hizmet bütünleştirmesini, varsayılan ileti alışverişi sağlayıcısını kullanan JMS uygulamalarını çalıştırmak için hizmet tümleştirme veriyolunu yapılandırdığınız şekilde yapılandırmalısınız.

# WebSphere Application Server ile bağlantı sağlayan bir uygulama için hizmet bütünleştirme veriyolunun yapılandırılması

Bir WebSphere Application Server service integration technologies hizmet bütünleştirme veriyoluna bağlanan bir uygulamayı çalıştırabilmeniz için, hizmet bütünleştirmesini, varsayılan ileti alışverişi sağlayıcısını kullanan JMS uygulamalarını çalıştırmak için hizmet tümleştirme veriyolunu yapılandırdığınız şekilde yapılandırmalısınız.

## Başlamadan önce

Bu görevi başlatmadan önce aşağıdaki adımları izlemelisiniz:

- Bir ileti alışverişi veriyolunun yaratıldığından ve sunucunuzun veriyolu üyesi olarak veriyoluna eklendiğinden emin olun.
- Uygulamanızın, çalışmakta olan en az bir ileti alışverişi altyapısı içeren bir hizmet tümleştirme veriyoluna erişimi olduğundan emin olun.
- HTTP işlemi gerekliyse, bir HTTP ileti alışverişi altyapısı gelen iletim kanalı tanımlanmalıdır. Varsayılan olarak, SSL ve TCP kanalları, sunucu kuruluşu sırasında tanımlanır.
- Uygulamanızın, özellikleri bir önyükleme sunucusu kullanılarak hizmet tümleştirme veriyoluna bağlanmak için uygun şekilde ayarlanmış bir bağlantı üreticisi kullandığından emin olun. Gereken en düşük bilgi aşağıdaki gibi olmalıdır:
  - İleti alışverişi sunucusuyla (önyükleme sunucusu aracılığıyla) bir bağlantı kararlaştırılırken kullanılacak yeri ve protokolü tanımlayan sağlayıcı uç noktası. En basit haliyle, varsayılan ayarlarla kurulan bir sunucu için, sağlama uç noktası, sunucunun anasistem adına ayarlanabilir.
  - İletilerin gönderildiği veriyolunun adı.

Bir bağlantı üreticisinin özellikleri hakkında daha fazla bilgi için bkz. [“ConnectionFactoryözellikleri” sayfa 174.](#)

## Bu görev hakkında

Gerekli olan kuyruk ya da konu alanları tanımlanmalıdır. Varsayılan olarak, sunucu kuruluşu sırasında Default.Topic.Space adı verilen bir konu alanı tanımlanır, ancak ek konu alanları gerekiyorsa, bu konu alanlarını kendiniz yaratmalısınız. Bir konu alanında tek tek konuları önceden tanımlamanıza gerek yoktur; sunucu bu konuları gereken şekilde somutlaştırdığından, bu konuları devingen olarak somutlaştırabilirsiniz.

Aşağıdaki adımlar, yapmanız gerekenleri özetlemektedir.

## Yordam

1. Uygulamanızın noktadan noktaya ileti alışverişi için gereksinim duyduğu kuyukları yaratın.
2. Uygulamanızın yayınlama/abone olma ileti alışverişi için gereksinim duyduğu ek konu alanlarını yaratın.

## Sonuçlar

Artık uygulamanıza başlayabilirsiniz.

### İlgili görevler

[IBM MQ kuyruk yöneticisine bağlanan bir uygulama için kuyruk yöneticisi ve aracısının yapılandırılması](#)

Bu bölümde, IBM WebSphere MQ 7.0.1 ya da daha sonraki bir sürümü kullandığınız varsayılır.

Bir IBM MQ kuyruk yöneticisine bağlanan bir uygulamayı çalıştırabilmeniz için, kuyruk yöneticisini yapılandırmanız gerekir. Bir yayınlama/abone olma uygulaması için Queued yayınlama/abone olma arabirimini kullanıyorsanız, bazı ek yapılandırmalara gerek vardır.

[Aracıya gerçek zamanlı bağlantı kullanan bir uygulama için aracı yapılandırılması](#)

Bir aracıya gerçek zamanlı bağlantı kullanan bir uygulamayı çalıştırabilmeniz için, o aracıyı yapılandırmanız gerekir.



## XMS örnek uygulamalarının kullanılması

Kuruluş ve ileti alışverişi sunucusu kurulumunuzu doğrulamak ve kendi uygulamalarınızı oluşturmanıza yardımcı olmak için XMS ile birlikte verilen örnek uygulamaları kullanın. Örnekler, her bir API ' nin ortak özelliklerine genel bir bakış sağlar.

### İlgili kavramlar

#### JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

#### İleti alışverişi sunucusu ortamının ayarlanması

The topics in this section describe how to set up the messaging server environment to allow XMS applications to connect to a server.

### İlgili görevler

#### Kuruluş sihirbazını kullanarak Message Service Client for .NET ' in kurulması

Kuruluş, bir InstallShield X/Windows MSI kuruluş programı kullanır. Eksiksiz ya da özel bir kuruluş seçebileceğiniz iki ayar seçeneği vardır.

## Örnek uygulamalar

Örnek uygulamalar, her bir API ' nin ortak özelliklerine genel bir bakış sağlar. Bunları, kuruluş ve ileti alışverişi sunucusu kurulumunuzu doğrulamak ve kendi uygulamalarınızı oluşturmanıza yardımcı olmak için kullanabilirsiniz.

Kendi uygulamalarınızı yaratmak için yardıma gerek duyarsanız, örnek uygulamaları başlangıç noktası olarak kullanabilirsiniz. Her uygulama için hem kaynak hem de derlenmiş bir sürüm sağlanır. Örnek kaynak kodunu gözden geçirin ve uygulamanız için gereken her bir nesneyi (ConnectionFactory, Connection, Session, Destination, Destination, bir üretici ya da bir tüketici ya da her ikisi) yaratmak ve uygulamanızın nasıl çalışmasını istediğinizi belirtmek için gerekli olan belirli özellikleri yaratmak için gereken temel adımları tanımlayın. Daha fazla bilgi için bkz “XMS uygulamaları yazılıyor” sayfa 19. Örnekler, XMS' un gelecekteki yayın düzeylerinde değişikliğe tabidir.

Aşağıdaki tabloda, XMS ile sağlanan üç örnek uygulama kümesi gösterilmektedir (her bir API için bir tane).

Örneklerin adı	Tanım
SampleConsumerCS	Bir kuyruktan ileti alan ya da bir konuya abone olan ileti tüketici uygulaması.
SampleProducerCS	Bir kuyruğa ya da konuya ilişkin iletiler üreten bir ileti üreticisi uygulaması.
SampleConfigCS	Dosya tabanlı, yönetilen bir nesne havuzu oluşturmak için kullanabileceğiniz bir yapılandırma uygulaması. Uygulama, belirli bağlantı ayarlarınız için bir bağlantı üreticisi ve hedefi içerir. Bu denetlenen nesne havuzu, örnek tüketici ve üretici uygulamalarının her biriyle birlikte kullanılabilir.

Çeşitli API ' lerde aynı işlevleri destekleyen örnekler, sözdizimsel farklılıklara sahiptir.

- Örnek ileti tüketici ve üretici uygulamalarının her ikisi de aşağıdaki işlevleri destekler:
  - IBM MQ, IBM Integration Bus (bir aracıya gerçek zamanlı bağlantı kullanarak) ve bir WebSphere Application Server service integration bus bağlantısı
  - İlk bağlam arabirimini kullanarak nesne havuzu aramalarını denetleyerek
  - Connections to queues (IBM MQ and WebSphere Application Server service integration bus) and topics (IBM MQ, real-time connection to a broker, and WebSphere Application Server service integration bus)
  - Temel, bayt, harita, nesne, akış ve metin iletileri

- Örnek ileti tüketici uygulaması zamanuyumlu ve zamanuyumsuz alma kiplerini ve SQL Seçici deyimlerini destekler.
- Örnek ileti üreticisi uygulaması, kalıcı ve kalıcı olmayan teslim kiplerini destekler.

## İşletim kipleri

Örnekler iki kipten birinde işlev görebilirler:

### Basit kip

Örnekleri en az kullanıcı girişi ile çalıştırabilirsiniz.

### İleri düzey kip

Örneklerin çalışacağı şekilde daha ayrıntılı bir şekilde özelleştirebilirsiniz.

Tüm örnekler uyumlu ve bu nedenle diller arasında çalışabilir.

### İlgili kavramlar

[Kendi uygulamalarınızı oluşturma](#)

Örnek uygulamaları oluştururken kendi uygulamalarınızı oluşturun.

### İlgili görevler

[Örnek uygulamaların çalıştırılması](#)

.NET örnek uygulamalarını, basit ya da gelişmiş kipte etkileşimli olarak ya da otomatik olarak oluşturulan ya da özelleştirilmiş yanıt dosyaları kullanarak etkileşimli olmayan bir şekilde çalıştırabilirsiniz.

[.NET örnek uygulamalarının oluşturulması](#)

Örnek bir .NET uygulaması oluşturduğunuzda, seçilen örneğinizin yürütülebilir bir sürümü oluşturulur.

## Örnek uygulamaların çalıştırılması

.NET örnek uygulamalarını, basit ya da gelişmiş kipte etkileşimli olarak ya da otomatik olarak oluşturulan ya da özelleştirilmiş yanıt dosyaları kullanarak etkileşimli olmayan bir şekilde çalıştırabilirsiniz.

### Başlamadan önce

Sağlanan örnek uygulamaların herhangi birini çalıştırmadan önce, uygulamaların bir sunucuya bağlanabilmesi için ileti alışverişi sunucusu ortamını ilk kez ayarlamanız gerekir. Bkz. [“İleti alışverişi sunucusu ortamının ayarlanması” sayfa 12.](#)

### Yordam

.NET örnek uygulamasını çalıştırmak için aşağıdaki adımları tamamlayın:

**İpucu:** Örnek bir uygulama çalıştırırken tip mi? Bir dahaki sefere ne yapacağına dair yardım almak için.

1. Örnek uygulamayı çalıştırmak istediğiniz kipi seçin.

Advanced ya da Simple yazın.

2. Sorularına cevap ver.

Soru sonundaki araçlarda gösterilen varsayılan değeri seçmek için Enter tuşuna basın. Farklı bir değer seçmek için, uygun değeri yazın ve Enter tuşuna basın.

Burada bir örnek soru var:

```
Enter connection type [wpm]:
```

Bu durumda, varsayılan değer wpm ' dir (bir WebSphere Application Server service integration busbağlantısıdır).

## Sonuçlar

Örnek uygulamaları çalıştırdığınızda, yanıt dosyaları otomatik olarak geçerli çalışma dizininde oluşturulur. Yanıt dosyası adları `connection_type-sample_type.rsp` biçiminde olur; örneğin, `wpm-producer.rsp`. Gerekliyse, örnek uygulamayı aynı seçeneklerle yeniden çalıştırmak için, seçenekleri yeniden girmeniz gerekmemesi için, oluşturulan yanıt kütüğünü kullanabilirsiniz.

### İlgili kavramlar

#### Örnek uygulamalar

Örnek uygulamalar, her bir API 'nin ortak özelliklerine genel bir bakış sağlar. Bunları, kuruluş ve ileti alışverişi sunucusu kurulumunuzu doğrulamak ve kendi uygulamalarınızı oluşturmanıza yardımcı olmak için kullanabilirsiniz.

### İlgili görevler

#### .NET örnek uygulamalarının oluşturulması

Örnek bir .NET uygulaması oluşturduğunuzda, seçilen örneğinizin yürütülebilir bir sürümü oluşturulur.

## .NET örnek uygulamalarının oluşturulması

Örnek bir .NET uygulaması oluşturduğunuzda, seçilen örneğinizin yürütülebilir bir sürümü oluşturulur.

### Başlamadan önce

Uygun derleyiciyi kurun. Bu görev, sisteminizde Microsoft Visual Studio 2012 kurulu olduğunu ve bunu kullanmaya alışkın olduğunuzu varsayar.

### Yordam

Bir .NET örnek uygulaması oluşturmak için aşağıdaki adımları tamamlayın:

1. .NET örnekleriyle birlikte sağlanan `Samples.sln` çözüm dosyasını tıklayın.
2. Right-click the solution `Örnekler` in the Solution Explorer window and select **Oluşturma Çözümü**.

## Sonuçlar

Seçtiğiniz yapılandırmaya bağlı olarak, örneğin `bin/Debug` ya da `bin/Release` örneğinin uygun alt klasöründe yürütülebilir bir program yaratılır. This program has the same name as the folder, with a suffix of `CS`. Örneğin, ileti üreticisi örnek uygulamasının C# sürümünü oluşturuyorsanız, `SampleProducerCS.exe` klasörü `SampleProducer` klasöründe yaratılır.

### İlgili kavramlar

#### Örnek uygulamalar

Örnek uygulamalar, her bir API 'nin ortak özelliklerine genel bir bakış sağlar. Bunları, kuruluş ve ileti alışverişi sunucusu kurulumunuzu doğrulamak ve kendi uygulamalarınızı oluşturmanıza yardımcı olmak için kullanabilirsiniz.

#### "Kendi uygulamalarınızı oluşturma" sayfa 41

Örnek uygulamaları oluştururken kendi uygulamalarınızı oluşturun.

### İlgili görevler

#### Örnek uygulamaların çalıştırılması

.NET örnek uygulamalarını, basit ya da gelişmiş kipte etkileşimli olarak ya da otomatik olarak oluşturulan ya da özelleştirilmiş yanıt dosyaları kullanarak etkileşimli olmayan bir şekilde çalıştırabilirsiniz.

## XMS uygulamalarının geliştirilmesi

Bu bölümdeki konular, XMS uygulamalarını yazarken yararlı olabilecek bilgileri içerir.

XMS uygulamalarının yazılmasıyla ilgili bilgi edinmek için aşağıdaki konulara bakın:

### XMS uygulamaları yazılıyor

Bu bölümdeki konular, XMS uygulamalarını yazarken size yardımcı olacak bilgiler sağlar.

Bu bölüm, XMS uygulamalarını yazmak için genel kavramlar içerir. .NET uygulamaları yaratılmasına ilişkin bilgiler için ayrıca bkz. [“Writing XMS .NET applications” sayfa 42](#) .

Bu bölüm aşağıdaki konuları içerir:

- [“Threading modeli” sayfa 20](#)
- [“ConnectionFactories ve Connection nesnelere” sayfa 20](#)
- [“Oturumlar” sayfa 23](#)
- [“Hedefler” sayfa 27](#)
- [“İleti üreticileri” sayfa 32](#)
- [“İleti tüketicileri” sayfa 32](#)
- [“Kuyruk tarayıcıları” sayfa 36](#)
- [“İstekçi” sayfa 36](#)
- [“Nesne Silme” sayfa 36](#)
- [“XMS temel tipleri” sayfa 37](#)
- [“Bir özellik değerinin bir veri tipinden diğerine örtük olarak dönüştürülmesi” sayfa 38](#)
- [“Yineleyiciler” sayfa 41](#)
- [“Kodlanmış karakter takımı tanıtıcıları” sayfa 41](#)
- [“XMS hatası ve kural dışı durum kodları” sayfa 41](#)
- [“Kendi uygulamalarınızı oluşturma” sayfa 41](#)

### **İlgili kavramlar**

[Writing XMS .NET applications](#)

Bu bölümdeki konular, XMS .NET uygulamalarını yazarken size yardımcı olacak bilgiler sağlar.

### **İlgili başvurular**

[.NET Arabirimler](#)

Bu bölüm , .NET sınıfı arabirimlerini ve bunların özelliklerini ve yöntemlerini içerir.

## **Threading modeli**

Genel kurallar, çok iş parçacıklı bir uygulamanın XMS nesnelere nasıl kullanabileceğini yönetir.

- Farklı iş parçacıklarında yalnızca, aşağıdaki tiplerin nesnelere koşut zamanlı olarak kullanılabilir:
  - ConnectionFactory
  - Bağlantı
  - ConnectionMetaVerileri
  - Hedef
- Bir Oturum nesnesi tek bir kerede tek bir iş parçacığında kullanılabilir.

Bu kuralların kural dışı durumları, [“Message Service Clients for .NET başvurusu” sayfa 85](#) içindeki yöntemlerin arabirim tanımlamalarında "İş parçacığı bağlamı" etiketli girişlerle gösterilir.

### **İlgili kavramlar**

[Çalıştırma zamanında işlenebilecek hata koşulları](#)

API çağrılarında dönen dönüş kodları, çalıştırma zamanında işlenebilecek hata koşullarıdır. Bu tür bir hata ile başa çıkabilmenin yolu, C ya da C++ API ' yi kullanıp kullanmamanıza bağlıdır.

## **ConnectionFactories ve Connection nesnelere**

ConnectionFactory nesnesi, uygulamanın Connection nesnesi yaratmak için kullandığı bir şablon sağlar. Uygulama, bir Oturum nesnesi yaratmak için Bağlantı nesnesini kullanır.

For .NET, XMS application first uses an XMSFactoryFactory object to get a reference to a ConnectionFactory object that is appropriate to the required type of protocol. Bu ConnectionFactory nesnesi, yalnızca bu protokol tipi için bağlantı üretebilir.

Bir XMS uygulaması birden çok bağlantı yaratabilir ve birden çok iş parçacıklı bir uygulama, birden çok iş parçacığının koştuzamanlı olarak tek bir Connection nesnesini kullanabilir. Bağlantı nesnesi, bir uygulama ile ileti sistemi sunucusu arasındaki bir iletişim bağlantısını sarsalıyor.

Bir bağlantı birkaç amaca hizmet eder:

- Bir uygulama bağlantı yarattığında, uygulamanın doğrulanabileceği bir uygulama olabilir.
- Bir uygulama, benzersiz bir istemci tanıtıcısını bir bağlantıyla ilişkilendirebilir. Yayınlama/abone olma etki alanında dayanıklı abonelikleri desteklemek için istemci tanıtıcısı kullanılır. İstemci tanıtıcısı şu iki şekilde ayarlanabilir:

Bir bağlantı istemcisi tanıtıcısı atanmanın tercih edilen yolu, özellikleri kullanarak istemciye özgü bir ConnectionFactory nesnesinde yapılandırılacak ve bunu yarattığı bağlantıya saydam bir şekilde atayabilmekte.

İstemci tanıtıcısını atanmanın diğer bir yolu, Connection nesnesinde belirlenen sağlayıcıya özgü bir değer kullanmandır. Bu değer, yönetimsel olarak yapılandırılmış olan tanıtıcıyı geçersiz kılmaz. Yönetimsel olarak belirtilmiş bir tanıtıcının var olmadığı vaka için sağlanır. Yönetimsel olarak belirtilmiş bir tanıtıcı varsa, bunu sağlayıcıya özgü bir değer ile geçersiz kılma girişimi bir kural dışı durumun yayınlanmasına neden olur. Bir uygulama belirtik olarak bir tanıtıcıyı ayarlarsa, bağlantı yaratıldıktan hemen sonra ve bağlantıda başka herhangi bir işlem alınmadan önce bunu yapması gerekir; aksi takdirde, kural dışı durum yayınlanır.

XMS uygulaması tipik olarak bir bağlantı, bir ya da daha çok oturum ve ileti üreticileri ve ileti tüketicileri oluşturur.

Bağlantı yaratılması sistem kaynakları açısından göreceli olarak pahalı olduğundan, iletişim bağlantısı kurulmasını içerdiği için, uygulamanın kimlik doğrulanmasını da gerektirebilir.

### **İlgili görevler**

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

### **İlgili başvurular**

IConnectionFactory (.NET arabirimi için)

Bir uygulama, bağlantı oluşturmak için bir bağlantı üreticisi kullanır.

ConnectionFactoryözellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren ConnectionFactory nesnesinin özelliklerine genel bakış.

IDestination (.NET arabirimi için)

Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.

Hedef Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla, Hedef nesnenin özelliklerine genel bakış.

### ***Bağlantı başlatıldı ve durduruldu kipi***

Bir bağlantı, başlatılmış ya da durdurulmuş kipte çalışabilir.

Bir uygulama bağlantı yarattığında, bağlantı durdurulmuş kipte olur. Bağlantı durdurulduğunda, uygulama oturumları başlatabilir, iletileri gönderebilir, ancak zamanuyumlu ya da zamanuyumsuz olarak alamazlar.

Bir uygulama, Start Connection yöntemini çağırarak bağlantı başlatabilir. Bağlantı başlatıldığında, uygulama iletileri gönderebilir ve alabilir. Daha sonra, uygulama durdurulabilir ve Bağlantıyı Durdur ve Start Connection yöntemlerini çağırarak bağlantıyı yeniden başlatabilir.

## **İlgili kavramlar**

### Bağlantı kapatma

Bir uygulama, Bağlantıyı Kapat yöntemini çağırarak bağlantıyı kapatır.

### Kural Dışı Durum İşleme

Bir uygulama, yalnızca bir kural dışı durum dinleyicisi kullanılarak bağlantıyla ilgili olarak iletileri zamanuyumsuz olarak tüketebilmek için bir bağlantı kullanıyorsa.

### Hizmet tümleştirme veriyoluna bağlantı

Bir XMS uygulaması, doğrudan TCP/IP bağlantısı kullanarak ya da TCP/IP üzerinden HTTP kullanarak bir WebSphere Application Server hizmet tümleştirme veriyoluna bağlanabilir.

## ***Bağlantı kapatma***

Bir uygulama, Bağlantıyı Kapat yöntemini çağırarak bağlantıyı kapatır.

Bir uygulama bağlantıyı kapattığında, XMS aşağıdaki eylemleri gerçekleştirir:

- Bağlantıyla ilişkili tüm oturumları kapatır ve bu oturumlarla ilişkili belirli nesnelere siler. Hangi nesnelere silindiği hakkında daha fazla bilgi için bkz. “Nesne Silme” sayfa 36. Aynı zamanda, XMS oturumları içinde devam etmekte olan tüm işlemleri geri alır.
- İleti, ileti sistemi sunucusuyla iletişim bağlantısını sona erdirir.
- Bu, bellek ve bağlantı tarafından kullanılan diğer iç kaynakları serbest bırakır.

XMS , bağlantı kapatılmadan önce, bir oturum sırasında kabul edememiş olduğu iletilerin alındığını bildirmez. İletilerin alınmasını kabul etme hakkında daha fazla bilgi için bkz. “İleti alındı bildirimi” sayfa 24.

## **İlgili kavramlar**

### Bağlantı başlatıldı ve durduruldu kipi

Bir bağlantı, başlatılmış ya da durdurulmuş kipte çalışabilir.

### Kural Dışı Durum İşleme

Bir uygulama, yalnızca bir kural dışı durum dinleyicisi kullanılarak bağlantıyla ilgili olarak iletileri zamanuyumsuz olarak tüketebilmek için bir bağlantı kullanıyorsa.

### Hizmet tümleştirme veriyoluna bağlantı

Bir XMS uygulaması, doğrudan TCP/IP bağlantısı kullanarak ya da TCP/IP üzerinden HTTP kullanarak bir WebSphere Application Server hizmet tümleştirme veriyoluna bağlanabilir.

## ***Kural Dışı Durum İşleme***

Bir uygulama, yalnızca bir kural dışı durum dinleyicisi kullanılarak bağlantıyla ilgili olarak iletileri zamanuyumsuz olarak tüketebilmek için bir bağlantı kullanıyorsa.

XMS .NET özel durumlarının tümü System.Exception' dan türetilmiştir. Daha fazla bilgi için bkz “Error handling in .NET” sayfa 46.

## **İlgili kavramlar**

### Bağlantı başlatıldı ve durduruldu kipi

Bir bağlantı, başlatılmış ya da durdurulmuş kipte çalışabilir.

### Bağlantı kapatma

Bir uygulama, Bağlantıyı Kapat yöntemini çağırarak bağlantıyı kapatır.

### Hizmet tümleştirme veriyoluna bağlantı

Bir XMS uygulaması, doğrudan TCP/IP bağlantısı kullanarak ya da TCP/IP üzerinden HTTP kullanarak bir WebSphere Application Server hizmet tümleştirme veriyoluna bağlanabilir.

## ***Hizmet tümleştirme veriyoluna bağlantı***

Bir XMS uygulaması, doğrudan TCP/IP bağlantısı kullanarak ya da TCP/IP üzerinden HTTP kullanarak bir WebSphere Application Server hizmet tümleştirme veriyoluna bağlanabilir.

HTTP protokolü, doğrudan TCP/IP bağlantısının olmadığı durumlarda kullanılabilir. Ortak bir durum, iki kuruluşun ileti alışverişi yaptığı gibi bir güvenlik duvarı üzerinden iletişim kurmaktadır. Güvenlik duvarı

üzerinden iletişim kurmak için HTTP ' nin kullanılması genellikle *HTTP tünellemesi*. olarak adlandırılır. Ancak HTTP üstbilgileri aktarılan veri miktarına önemli ölçüde eklendiğinden ve HTTP iletişim kuralı TCP/IP ' den daha fazla iletişim akışı gerektirdiğinden, HTTP tünelleme doğrudan TCP/IP bağlantısı kullandığından daha yavaş olur.

Bir TCP/IP bağlantısı yaratmak için, bir uygulama XMSC\_WPM\_TARGET\_TRANSPORT\_CHAIN özelliği XMSC\_WPM\_TARGET\_TRANSPORT\_CHAIN\_BASIC olarak ayarlanmış bir bağlantı üreticisine sahip olabilir. Bu, özelliğin varsayılan değeridir. Bağlantı başarıyla yaratıldıysa, bağlantının XMSC\_WPM\_CONNECTION\_PROTOCOL özelliği XMSC\_WPM\_CP\_TCP olarak ayarlanır.

HTTP kullanan bir bağlantı yaratmak için, bir uygulamanın XMSC\_WPM\_TARGET\_TRANSPORT\_CHAIN özelliği, bir HTTP iletim kanalını kullanmak üzere yapılandırılmış bir gelen iletim zinciri adı olarak ayarlanmış bir bağlantı üreticisi kullanmalıdır. Bağlantı başarıyla yaratıldıysa, bağlantının XMSC\_WPM\_CONNECTION\_PROTOCOL özelliği XMSC\_WPM\_CP\_HTTP olarak ayarlanır. İletim zincirlerinin nasıl yapılandırılacağı hakkında bilgi için, WebSphere Application Server ürün belgelerinde Configuring transport chains başlıklı konuya bakın.

Bir uygulama, bir önyükleme sunucusuna bağlanırken benzer bir iletişim protokolleri seçimine sahiptir. Bir bağlantı üreticisinin XMSC\_WPM\_PROVIDER\_UÇ Noktaları özelliği, önyükleme sunucularının bir ya da daha çok uç noktası adreslerinden oluşan bir dizidir. Her uç adresin önyükleme iletim zinciri bileşeni, HTTP kullanan bir bağlantı için, bir önyükleme sunucusuyla ya da XMSC\_WPM\_BOOTSTRAP\_HTTP ile TCP/IP bağlantısı için XMSC\_WPM\_BOOTSTRAP\_TCP olabilir.

### **İlgili kavramlar**

Bağlantı başlatıldı ve durduruldu kipi

Bir bağlantı, başlatılmış ya da durdurulmuş kipte çalışabilir.

Bağlantı kapatma

Bir uygulama, Bağlantıyı Kapat yöntemini çağırarak bağlantıyı kapatır.

Kural Dışı Durum İşleme

Bir uygulama, yalnızca bir kural dışı durum dinleyicisi kullanılarak bağlantıyla ilgili olarak iletileri zamanuyumsuz olarak tüketebilmek için bir bağlantı kullanıyorsa.

### **İlgili görevler**

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

### **İlgili başvurular**

ConnectionFactory (.NET arabirimi için)

Bir uygulama, bağlantı oluşturmak için bir bağlantı üreticisi kullanır.

ConnectionFactory özellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren ConnectionFactory nesnesinin özelliklerine genel bakış.

IDestination (.NET arabirimi için)

Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.

Hedef Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla, Hedef nesnenin özelliklerine genel bakış.

### **Oturumlar**

Oturum, ileti göndermek ve almak için tek bir iş parçacıklı bağlamdır.

Bir uygulama, ileti, ileti üreticileri, ileti tüketicileri, kuyruk tarayıcıları ve geçici hedefler yaratmak için bir oturumu kullanabilir. Bir uygulama yerel hareketleri çalıştırmak için de oturum kullanabilir.

Bir uygulama, her oturumun diğer oturumlardan bağımsız olarak ileti ürettiği ve tükettiği birden çok oturum yaratabiliyor. Ayrı oturumlarda (ya da aynı oturumda bile) iki ileti tüketicinin aynı konuya abone olması durumunda, her biri bu konuda yayınlanan herhangi bir iletinin bir kopyasını alır.

Connection nesnesinden farklı olarak, bir Oturum nesnesi farklı iş parçacılarında koştuzamanlı olarak kullanılamaz. Bir oturum nesnesinin yalnızca Close Session yöntemi, oturum nesnesinin o sırada kullandığı bir iş parçacıktan başka bir iş parçacığıdan çağrılabilir. Oturum Kapat (Close Session) yöntemi bir oturumu sonlandırır ve oturuma ayrılan sistem kaynaklarını serbest bırakır.

Bir uygulamanın iletileri birden çok iş parçacığıda koştuzamanlı olarak işlemesi gerekiyorsa, uygulamanın her bir iş parçacığıda bir oturum yaratması ve söz konusu iş parçacığının içindeki gönderme ya da alma işlemi için bu oturumu kullanması gerekir.

### **Hareket edilen oturumlar**

XMS uygulamaları yerel işlemleri çalıştırabilir. *Yerel hareket*, yalnızca, uygulamanın bağlı olduğu kuyruk yöneticisinin ya da hizmet tümleştirme veriyolunun kaynaklarında yapılan değişiklikleri içeren bir işlemdir.

Bu konudaki bilgiler, yalnızca bir uygulama bir IBM MQ kuyruk yöneticisine ya da WebSphere Application Server hizmet bütünleştirme veriyoluna bağlıysa anlamlıdır. Bilgi, bir aracıyla gerçek zamanlı bağlantı için uygun değildir.

Bir uygulamanın, yerel hareketleri çalıştırmak için, önce bir Bağlantı nesnesinin Create Session (Oturum Yarat) yöntemini çağırarak, oturumun hareket ettiği bir değiştirge olarak belirtilerek, aktarım sırasında bir oturum yaratması gerekir. Daha sonra, oturum içinde gönderilen ve alınan tüm iletiler bir işlem sırasına göre gruplandırılır. Bir işlem, uygulama başlattığında ya da gönderdiği iletileri geri alırken, işlem başlatıldığından bu yana aldığı iletiler sona erer.

Bir işlem kesinleştirilmek için, bir uygulama Oturum nesnesinin Kesinleştir yöntemini çağırır. Bir işlem kesinleştirildiğinde, işlem içinde gönderilen tüm iletiler diğer uygulamalara teslim edilebilmesi için kullanılabilir duruma gelir ve işlem içinde alınan tüm iletiler, ileti alışverişi sunucusunun bunları yeniden uygulamaya teslim etme girişiminde bulunmayabilmesi için kabul edilir. Noktadan noktaya iletişim alanında, ileti alışverişi sunucusu alınan iletileri kuyruklarından da kaldırır.

Bir işlemi geri yüklemek için, bir uygulama Oturum nesnesinin Geri Alma yöntemini çağırır. Bir hareket geriye işlendiğinde, işlem içinde gönderilen tüm iletiler ileti alışverişi sunucusu tarafından atılır ve işlem içinde alınan tüm iletiler, teslim için yeniden kullanılabilir duruma gelir. Noktadan noktaya iletişim alanında, alınan iletiler kuyruklarına geri alınır ve diğer uygulamalar tarafından görülebilir duruma gelir.

Uygulama, hareket eden bir oturum yarattığında ya da Commit ya da Rollback yöntemini çağırıldığında otomatik olarak yeni bir işlem başlar. Bu nedenle, hareket eden bir oturumun her zaman etkin bir hareketi vardır.

Bir uygulama hareket edilen bir oturumu kapattığında, örtük bir geri alma gerçekleşir. Bir uygulama bağlantıyı kapattığında, tüm bağlantının hareket ettiği oturumlar için örtük bir geri alma gerçekleşir.

Hareket, hareket eden bir oturumda tamamen içerilidir. Bir hareket oturumlara yayılamaz. Diğer bir deyişle, bir uygulamanın iki ya da daha fazla sayıda dönüştürücü oturumuna ileti gönderip alması ve sonra tüm bu işlemleri tek bir işlem olarak kesinleştirmesi ya da geriye işlemeleri mümkün değildir.

### **İlgili kavramlar**

#### **İleti alındı bildirimi**

İletilmeyen her oturumda, uygulamanın aldığı iletilerin nasıl kabul edildiğine karar veren bir alındı bildirimi kipi vardır. Üç alındı bildirimi kipi vardır ve alındı bildirimi kipinin seçimi, uygulamanın tasarımını etkiler.

#### **Zamanuyumsuz ileti teslimi**

XMS, bir oturuma ilişkin tüm zamanuyumsuz ileti teslimlerini işlemek için bir iş parçacığı kullanır. Başka bir deyişle, bir kerede tek bir ileti dinleyicisi işlevi ya da bir onMessage() yöntemi çalıştırılabilir.

#### **Zamanuyumlu ileti teslimi**

Uygulama, MessageConsumer nesnelere ilişkin Alma yöntemlerini kullanıyorsa, iletiler bir uygulamaya zamanuyumlu olarak teslim edilir.

#### **İleti teslim kipi**

XMS, ileti tesliminin iki kipini destekler.

### **İleti alındı bildirimi**

İletilmeyen her oturumda, uygulamanın aldığı iletilerin nasıl kabul edildiğine karar veren bir alındı bildirimi kipi vardır. Üç alındı bildirimi kipi vardır ve alındı bildirimi kipinin seçimi, uygulamanın tasarımını etkiler.



Bu konudaki bilgiler, yalnızca bir uygulama bir IBM MQ kuyruk yöneticisine ya da WebSphere Application Server hizmet bütünleştirme veriyoluna bağlıysa anlamlıdır. Bilgi, bir aracıyla gerçek zamanlı bağlantı için uygun değildir.

XMS , JMS ' nin kullandığı iletilerin alınmasının onaylanmak için aynı mekanizmayı kullanır.

Bir oturum hareket edilmezse, uygulamanın aldığı iletilerin kabul edilme şekli, oturumun alındı bildirimi kipiyle belirlenir. Üç alındı bildirimi kipleri aşağıdaki paragraflarda açıklanmaktadır:

### **XMSCS\_AUTO\_RASK**

Oturum, uygulama tarafından alınan her iletiyi otomatik olarak kabul eder.

İletiler uygulamaya zamanuyumlu olarak teslim ediliyorsa, bir Receive çağrısı her başarıyla tamamlandığında, oturum bir iletinin alındığını kabul eder.

Uygulama başarıyla bir ileti alırsa, ancak hata onayının oluşmasını önlüyorsa, ileti teslim için kullanılabilir duruma gelir. Bu nedenle, uygulamanın yeniden teslim edilen bir iletiyi işleyebilmesi gerekir.

### **XMSC\_DUPS\_OK\_ALIND**

Bu oturum, uygulama tarafından alınan iletilerin seçim süresinde kabul ettiği kabul eder.

Bu alındı bildirimi kipinin kullanılması, oturumun yapması gereken iş miktarını azaltır, ancak ileti onayını önleyen bir hata, birden çok iletinin teslim edilmesi için kullanılabilir duruma gelmesinden kaynaklanabilir. Bu nedenle, uygulamanın yeniden teslim edilen iletileri işleyebilmesi gerekir.

### **XMSC\_CLIENT\_ALIND**

Uygulama, İleti sınıfının Acknowy yöntemini çağırarak aldığı iletileri kabul eder.

Uygulama, her iletinin alınmasını tek tek onaylayabilir ya da bir ileti kümesi alabilir ve Acknowy yöntemini yalnızca aldığı son ileti için çağırabilir. Yöntemin çağırıldığı son çağrıdan bu yana Acknowy yöntemi çağırıldığında alınan tüm iletiler onaylanır.

Bu onay kiplerinden herhangi biriyle birlikte, bir uygulama Oturum sınıfının Recon yöntemini çağırarak, bir oturumda iletilerin teslim edilmesini durdurabilir ve yeniden başlatabilir. Alındı bildirimi önceden onaylanmamış olan iletiler yeniden teslim edilir. Ancak, bunlar daha önce teslim edildikleri sırayla teslim edilmeyebilir. Bu arada, daha yüksek öncelikli iletiler gelmiş olabilir ve özgün iletilerin bazılarının süresi dolmuş olabilir. Noktadan noktaya iletişim alanında, özgün iletilerin bazıları başka bir uygulama tarafından tüketilmiş olabilir.

Bir uygulama, iletinin JMSReverulamış üstbilgi alanının içeriğini inceleyerek bir iletinin yeniden teslim edilip edilmediğini belirleyebilir. Bu işlem, Message sınıfının JMSReverlemiş yöntemini çağırarak bunu yapar.

### **İlgili kavramlar**

**Hareket edilen oturumlar**

XMS uygulamaları yerel işlemleri çalıştırabilir. *Yerel hareket* , yalnızca, uygulamanın bağlı olduğu kuyruk yöneticisinin ya da hizmet tümleştirme veriyolunun kaynaklarında yapılan değişiklikleri içeren bir işlemdir.

**Zamanuyumsuz ileti teslimi**

XMS , bir oturuma ilişkin tüm zamanuyumsuz ileti tesllerini işlemek için bir iş parçacığı kullanır. Başka bir deyişle, bir kerede tek bir ileti dinleyicisi işlevi ya da bir onMessage () yöntemi çalıştırılabilir.

**Zamanuyumlu ileti teslimi**

Uygulama, MessageConsumer nesnelere ilişkin Alma yöntemlerini kullanıyorsa, iletiler bir uygulamaya zamanuyumlu olarak teslim edilir.

**İleti teslim kipi**

XMS , ileti tesliminin iki kipini destekler.

### **Zamanuyumsuz ileti teslimi**

XMS , bir oturuma ilişkin tüm zamanuyumsuz ileti tesllerini işlemek için bir iş parçacığı kullanır. Başka bir deyişle, bir kerede tek bir ileti dinleyicisi işlevi ya da bir onMessage () yöntemi çalıştırılabilir.

Bir oturumda birden çok ileti tüketicisi iletileri zamanuyumsuz olarak alıyorsa ve bir ileti dinleyici işlevi ya da onMessage () yöntemi ileti tüketicisi için bir ileti gönderiyorsa, aynı iletiyi bekleyen diğer ileti

tüketicileri beklemeye devam etmelidir. Oturuma teslim edilmeyi bekleyen diğer iletiler de beklemeye devam etmelidir.

Bir uygulama iletileri eşzamanlı olarak teslim etmeyi gerektiriyorsa, birden çok oturum yaratın; böylece, XMS zamanuyumsuz ileti teslimi işlemek için birden çok iş parçacığı kullanır. Bu şekilde, birden çok ileti dinleyici işlevi ya da onMessage () yöntemi koşut zamanlı olarak çalışabilir.

Tüketiciye bir ileti dinleyicisi atayarak, oturum zamanuyumsuz olarak yapılmaz. Bir oturum, Connection.Start yöntemi çağrıldığında zamanuyumsuz hale gelir. Connection.Start yöntemi çağrılınca kadar tüm zamanuyumlu çağrılara izin verilir. Message delivery to consumers start when the Connection.Start is called.

Bir tüketici ya da üretici yaratma gibi zamanuyumlu çağrılar zamanuyumsuz bir oturumda yapılmalıdır; bu çağrılar Connection.Stop çağrılmalıdır. İletilerin teslimi başlatmak için Connection.Start yöntemi çağrılarak bir oturuma devam edilebilir. Bu durum için tek kural dışı durum, geri bildirme işlevine ileti gönderen iş parçacığıysa, Oturum iletileri teslim iş parçacığıdır. Bu iş parçacığı, iletileri geri çağırma işlevinde oturum (kapatma çağrısı dışında) herhangi bir çağrıyı gerçekleştirmez.

**Not:** Yönetilmeyen kipte, geri çağrılan işlev içindeki MQDISC çağrısı IBM MQ .NET istemcisi tarafından desteklenmez. Bu nedenle, istemci uygulaması Zamanuyumsuz alma kipinde MessageListener geri çağırması içinde oturum Yaratamaz ya da kapatamıyor. Create and dispose the session outside of the MessageListener method.

### **İlgili kavramlar**

Hareket edilen oturumlar

XMS uygulamaları yerel işlemleri çalıştırabilir. *Yerel hareket*, yalnızca, uygulamanın bağlı olduğu kuyruk yöneticisinin ya da hizmet tümleştirme veriyolunun kaynaklarında yapılan değişiklikleri içeren bir işlemdir.

İleti alındı bildirimi

İletilmeyen her oturumda, uygulamanın aldığı iletilerin nasıl kabul edildiğine karar veren bir alındı bildirimi kipi vardır. Üç alındı bildirimi kipi vardır ve alındı bildirimi kipinin seçimi, uygulamanın tasarımını etkiler.

Zamanuyumlu ileti teslimi

Uygulama, MessageConsumer nesnelere ilişkin Alma yöntemlerini kullanıyorsa, iletiler bir uygulamaya zamanuyumlu olarak teslim edilir.

İleti teslim kipi

XMS, ileti tesliminin iki kipini destekler.

### **Zamanuyumlu ileti teslimi**

Uygulama, MessageConsumer nesnelere ilişkin Alma yöntemlerini kullanıyorsa, iletiler bir uygulamaya zamanuyumlu olarak teslim edilir.

Alma yöntemlerinin kullanılması, bir ileti için belirli bir süre bekleyebilir ya da süresiz olarak bekleyebilir. Diğer bir seçenek olarak, bir uygulama bir iletiyi beklemek istemezse, Bekleme yöntemini kullanmadan Al 'ı kullanabilir.

### **İlgili kavramlar**

Hareket edilen oturumlar

XMS uygulamaları yerel işlemleri çalıştırabilir. *Yerel hareket*, yalnızca, uygulamanın bağlı olduğu kuyruk yöneticisinin ya da hizmet tümleştirme veriyolunun kaynaklarında yapılan değişiklikleri içeren bir işlemdir.

İleti alındı bildirimi

İletilmeyen her oturumda, uygulamanın aldığı iletilerin nasıl kabul edildiğine karar veren bir alındı bildirimi kipi vardır. Üç alındı bildirimi kipi vardır ve alındı bildirimi kipinin seçimi, uygulamanın tasarımını etkiler.

Zamanuyumsuz ileti teslimi

XMS, bir oturuma ilişkin tüm zamanuyumsuz ileti teslimlerini işlemek için bir iş parçacığı kullanır. Başka bir deyişle, bir kerede tek bir ileti dinleyicisi işlevi ya da bir onMessage () yöntemi çalıştırılabilir.

İleti teslim kipi

XMS, ileti tesliminin iki kipini destekler.

## **İleti teslim kipi**

XMS , ileti tesliminin iki kipini destekler.

- *Kalıcı* iletileri bir kez teslim edilir. Bir ileti alışverişi sunucusu, hata durumunda bile iletilerin geçiş sırasında kaybolmadığından emin olmak için iletileri günlüğe kaydetme gibi özel önlemleri alır.
- *Kalıcı olmayan* iletiler bir kereden fazla teslim edilir. Kalıcı olmayan iletiler, bir hata durumunda geçiş sırasında kaybedilebileceğinden, kalıcı iletilerden daha az güvenilirdir.

Teslim kipinin seçimi, güvenilirlik ve başarımlar arasında bir değiş tokardır. Kalıcı olmayan iletiler genellikle kalıcı iletilerden daha hızlı taşınır.

## **İlgili kavramlar**

Hareket edilen oturumlar

XMS uygulamaları yerel işlemleri çalıştırabilir. *Yerel hareket* , yalnızca, uygulamanın bağlı olduğu kuyruk yöneticisinin ya da hizmet tümleştirme veriyolunun kaynaklarında yapılan değişiklikleri içeren bir işlemdir.

İleti alındı bildirim

İletilmeyen her oturumda, uygulamanın aldığı iletilerin nasıl kabul edildiğine karar veren bir alındı bildirim kipi vardır. Üç alındı bildirim kipi vardır ve alındı bildirim kipinin seçimi, uygulamanın tasarımını etkiler.

Zamanuyumsuz ileti teslimi

XMS , bir oturuma ilişkin tüm zamanuyumsuz ileti teslimlerini işlemek için bir iş parçacığı kullanır. Başka bir deyişle, bir kerede tek bir ileti dinleyicisi işlevi ya da bir `onMessage()` yöntemi çalıştırılabilir.

Zamanuyumlu ileti teslimi

Uygulama, `MessageConsumer` nesnelere ilişkin `Alma` yöntemlerini kullanıyorsa, iletiler bir uygulamaya zamanuyumlu olarak teslim edilir.

## **Hedefler**

Bir XMS uygulaması, gönderilmekte olan iletilerin hedefini ve alınmakta olan iletilerin kaynağını belirtmek için bir Hedef nesne kullanır.

Bir XMS uygulaması yürütme sırasında bir hedef nesne yaratabilir ya da yönetilen nesnelere havuzundan önceden tanımlanmış bir hedef alabilir.

Bir `ConnectionFactory` ile olduğu gibi, XMS uygulamasının bir hedef belirlemesi için en esnek yol, bunu yönetilen bir nesne olarak tanımlamasıdır. Bu yaklaşımın kullanılması, C, C++ ve .NET dillerinde yazılan uygulamaların ve Java' in hedef tanımlarını paylaşabileceği şekilde ifade eder. Yönetilen Hedef nesnelere ilişkin özellikleri, herhangi bir kod değiştirilmeden değiştirilebilir.

.NET uygulamaları için, `CreateTopic` ya da `CreateQueue` yöntemini kullanarak bir hedef yaratıyorsunuz. Bu iki yöntem, .NET API ' da hem `ISSession` hem de `XMSFactoryFactory` nesnelere kullanılabilir. Daha fazla bilgi için bkz. ".NET ' teki hedefler" sayfa 44 ve "Varış Noktası" sayfa 103.

## **İlgili başvurular**

IDestination (.NET arabirimi için)

Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.

Hedef Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla, Hedef nesnenin özelliklerine genel bakış.

## **Konu URL tipi kaynak tanıtıcıları**

Konu tekstili kaynak tanıtıcısı (URI), konunun adını belirtir; bir ya da daha çok özellik de belirtebilir.

Bir konuya ilişkin URI, şu sıra ile başlar: //, onu izleyen konu adı ve (isteğe bağlı) kalan konu özelliklerini ayarlayan ad-değer çiftleri listesi. Bir konu adı boş olamaz.

Burada, .NET kodunun bir parçasında bir örnek vardır:

```
topic = session.CreateTopic("topic://Sport/Football/Results?multicast=7");
```

Bir URI içinde kullanabileceğiniz ad ve geçerli değerler de içinde olmak üzere, bir konunun özellikleri hakkında daha fazla bilgi için bkz. "[Hedef Özellikleri](#)" sayfa 180.

Abonelikte kullanmak üzere bir konu URI 'si belirtilirken joker karakterler kullanılabilir. Bu genel arama karakterlerinin sözdizimi, bağlantı tipi ve aracı sürümüne bağlıdır; aşağıdaki seçenekler kullanılabilir:

- Karakter düzeyi genel arama karakteri biçimiyle IBM WebSphere MQ 7.0 kuyruk yöneticisi
- IBM WebSphere MQ 7.0 queue manager with Topic level wildcard format
- WebSphere Application Server hizmet bütünleştirme veriyolu

## Karakter düzeyi genel arama karakteri biçimiyle IBM WebSphere MQ 7.0 kuyruk yöneticisi

Karakter düzeyi genel arama karakteriyle IBM WebSphere MQ 7.0 kuyruk yöneticisi, aşağıdaki genel arama karakterlerini kullanır:

- \* 0 ya da daha çok karakter için
- ? 1 karakter için
- % Bir çıkış karakteri için

[Çizelge 1 sayfa 28](#) , bu genel arama karakteri şemasının nasıl kullanılacağını gösteren bazı örnekler verir.

Bir örnek Kaynak Tanıtıcısı	Eşleşir	Örnekler
"topic://Sport *Results"	"Spor" ile başlayan ve "Sonuçlar" ile biten tüm konular	"topic://SportsResults" ve "topic://Sport/Hockey/National/Div3/Results"
" topic://Sport?Sonuçlar "	"Spor" ile başlayan ve ardından tek bir karakter olan tüm konular, ardından "Sonuçlar"	"topic://SportsResults" ve "topic://SportXResults"
"topic://Sport/ * ball*/Div? / Results*//???"	Konular	"topic://Sport/Football/Div1/Results/2002/Nov" ve "topic://Sport/Netball/National/Div3/Results/02/Jan"

## IBM WebSphere MQ 7.0 queue manager with Topic level wild card format

IBM WebSphere MQ 7.0 queue manager with Topic level wildcard format uses the following wildcard characters:

- # birden çok düzeye eşleştirmek için
- + tek bir düzeye eşleştirmek için

[Çizelge 2 sayfa 28](#) , bu genel arama karakteri şemasının nasıl kullanılacağını gösteren bazı örnekler verir.

Bir örnek Kaynak Tanıtıcısı	Eşleşir	Örnekler
"topic://Sport/ + / Results"	Spor ile Sonuçlar arasında tek bir sıradüzensel düzey adı olan tüm konular	"topic://Sport/Football/Results" ve "topic://Sport/Ju-Jitsu/Results"
"topic://Sport/#/Results"	"Sport/" ile başlayan ve "/Results" ile biten tüm konular	"topic://Sport/Football/Results" ve "topic://Sport/Hockey/National/Div3/Results"

Çizelge 2. IBM WebSphere MQ 7.0 kuyruk yöneticisi için konu düzeyi genel arama şemasını kullanan örnek URI 'ler (devamı var)

Birörnek Kaynak Tanıtıcısı	Eşleşir	Örnekler
"topic://Sport/Football/#"	"Sport/Football/" ile başlayan tüm konular	"topic://Sport/Football/Results" ve "topic://Sport/Football/TeamNews/Signings/Managery"

## WebSphere Application Server hizmet bütünleştirme veriyolu

WebSphere Application Server hizmet tümleştirme veriyolu şu genel arama karakterlerini kullanır:

- \* sıradüzendeki bir düzeydeki tüm karakterleri eşleştirmek için
- // 0 ya da daha fazla düzey eşleştirmek
- // . 0 ya da daha fazla düzeyle eşleştirmek için (Konu ifadesinin sonunda)

Çizelge 3 sayfa 29 , bu genel arama karakteri şemasının nasıl kullanılacağını gösteren bazı örnekler verir.

Çizelge 3. WebSphere Application Server hizmet tümleştirme veriyolu için genel arama şeması kullanan örnek URI 'ler

Birörnek Kaynak Tanıtıcısı	Eşleşir	Örnekler
"topic://Sport/ * ball/Results"	Spor ve Sonuçlar arasında "top" ile biten tek bir sıradüzensel düzey adı olan tüm konular	"topic://Sport/Football/Results" ve "topic://Sport/Netball/Results"
"topic://Sport//Results"	"Sport/" ile başlayan ve "/Results" ile biten tüm konular	"topic://Sport/Football/Results" ve "topic://Sport/Hockey/National/Div3/Results"
"topic://Sport/Football//."	"Sport/Football/" ile başlayan tüm konular	"topic://Sport/Football/Results" ve "topic://Sport/Football/TeamNews/Signings/Managery"
"topic://Sport/ * ball//Results//."	Konular	"topic://Sport/Football/Results" ve "topic://Sport/Netball/National/Div3/Results/2002/November"

### İlgili kavramlar

Kuyruk tek biçimli kaynak tanıtıcıları

Kuyruğun URI 'si, kuyruğun adını belirtir; ayrıca, kuyruğun bir ya da daha fazla özelliğini de belirtebilir.

Geçici hedefler

XMS uygulamaları geçici hedefler oluşturabilir ve kullanabilir.

Hedef joker karakterleri

XMS , hedef genel arama karakterleri için destek sağlar. Bu, joker karakterlerin, eşleştirme için gerekli olan yere aktarılabilmelerini sağlar. XMS ' un çalışabileceği her bir sunucu tipi için farklı bir joker karakter şeması vardır.

### İlgili başvurular

IDestination (.NET arabirimi için)

Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.

Hedef Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla, Hedef nesnenin özelliklerine genel bakış.

### Kuyruk tek biçimli kaynak tanıtıcıları

Kuyruğun URI 'si, kuyruğun adını belirtir; ayrıca, kuyruğun bir ya da daha fazla özelliğini de belirtebilir.

Kuyruğun URI 'si sıra kuyruğundan başlar: // ve ardından kuyruğun adı; kalan kuyruk özelliklerini ayarlayan ad-değer çiftlerinin bir listesini de içerebilir.

IBM MQ kuyrukları için ( WebSphere Application Server varsayılan ileti alışverişi sağlayıcısı kuyrukları için değil), kuyruğun bulunduğu kuyruk yöneticisi kuyruktan önce, kuyruk yöneticisi adını kuyruk adından ayırıyor ya da kuyruktan ayırıyor olabilir.

If a queue manager is specified, then it must be the one to which XMS is directly connected for the connection using this queue, or it must be accessible from this queue. Uzak kuyruk yöneticileri yalnızca kuyruktan ileti almak için, iletileri kuyruklara koymak için değil, kuyruklardan almak için desteklenir. Tüm ayrıntılar için, IBM MQ kuyruk yöneticisi belgelerine bakın.

Kuyruk yöneticisi belirtilmediyse, ek/ayırıcı isteğe bağlıdır ve varlığın varlığı ya da yokluğu, kuyruğun tanımı üzerinde hiçbir fark göstermez.

Aşağıdaki kuyruk tanımları, QM\_A adlı bir kuyruk yöneticisindeki QB adlı bir IBM MQ kuyruğu için, XMS ' un doğrudan bağlı olduğu bir kuyruk yöneticisine eşdeğerdir:

```
queue://QB
queue:///QB
queue://QM_A/QB
```

## İlgili kavramlar

Konu URL tipi kaynak tanıtıcıları

Konu tekstili kaynak tanıtıcısı (URI), konunun adını belirtir; bir ya da daha çok özellik de belirtebilir.

Geçici hedefler

XMS uygulamaları geçici hedefler oluşturabilir ve kullanabilir.

Hedef joker karakterleri

XMS , hedef genel arama karakterleri için destek sağlar. Bu, joker karakterlerin, eşleştirme için gerekli olan yere aktarılabilmelerini sağlar. XMS ' un çalışabileceği her bir sunucu tipi için farklı bir joker karakter şeması vardır.

## İlgili başvurular

IDestination (.NET arabirimi için)

Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.

Hedef Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla, Hedef nesnenin özelliklerine genel bakış.

## Geçici hedefler

XMS uygulamaları geçici hedefler oluşturabilir ve kullanabilir.

Uygulama, genellikle istek iletilerine yanıt almak için geçici bir hedef kullanır. Bir istek iletisinin yanıtının gönderileceği hedefi belirtmek için, bir uygulama istek iletisini temsil eden ileti nesnesinin Set JMSReplyTo yöntemini çağırır. Çağrıda belirtilen hedef geçici bir hedef olabilir.

Geçici bir hedef yaratmak için bir oturum kullanılsa da, geçici bir hedefin kapsamı gerçekte, oturumu yaratmak için kullanılan bağlantıdır. Bağlantı oturumlarından herhangi biri geçici hedef için ileti üreticileri ve ileti tüketicileri oluşturabilir. Geçici hedef, açık bir şekilde silininceye ya da bağlantı sona erinceye kadar kalır; hangisi önce olursa.

Bir uygulama geçici bir kuyruk yarattığında, uygulamanın bağlı olduğu ileti sistemi sunucusunda bir kuyruk yaratılır. Uygulama bir kuyruk yöneticisine bağlıysa, adı XMLC\_WMQ\_TEMPORARY\_MODEL özelliği tarafından belirtilen model kuyruğundan dinamik bir kuyruk yaratılır ve dinamik kuyruk adını oluşturmak için kullanılan önek XMLC\_WMQ\_TEMP\_Q\_PREFIX özelliği tarafından belirtilir. Uygulama bir hizmet bütünleştirme veriyoluna bağlıysa, veriyolunda geçici bir kuyruk yaratılır ve geçici kuyruğun adını oluşturmak için kullanılan önek XMSC\_WPM\_TEMP\_Q\_PREFIX özelliği tarafından belirtilir.

Bir hizmet tümleştirme veriyoluna bağlı bir uygulama geçici bir konu yarattığında, geçici konunun adını oluşturmak için kullanılan önek XMSC\_WPM\_TEMP\_TOPIC\_PREFIX özelliği tarafından belirtilir.

## İlgili kavramlar

Konu URL tipi kaynak tanıtıcıları

Konu tekstili kaynak tanıtıcısı (URI), konunun adını belirtir; bir ya da daha çok özellik de belirtebilir.

Kuyruk tek biçimli kaynak tanıtıcıları

Kuyruğun URI 'si, kuyruğun adını belirtir; ayrıca, kuyruğun bir ya da daha fazla özelliğini de belirtebilir.

Hedef joker karakterleri

XMS , hedef genel arama karakterleri için destek sağlar. Bu, joker karakterlerin, eşleştirme için gerekli olan yere aktarılabilmelerini sağlar. XMS ' un çalışabileceği her bir sunucu tipi için farklı bir joker karakter şeması vardır.

## İlgili başvurular

IDestination (.NET arabirimi için)

Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.

Hedef Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla, Hedef nesnenin özelliklerine genel bakış.

## Hedef joker karakterleri

XMS , hedef genel arama karakterleri için destek sağlar. Bu, joker karakterlerin, eşleştirme için gerekli olan yere aktarılabilmelerini sağlar. XMS ' un çalışabileceği her bir sunucu tipi için farklı bir joker karakter şeması vardır.

Şemalar şunlardır:

Bağlantı tipi	Joker karakter şeması	Tanım
WebSphere MQ Kuyruk Yöneticisi	* ? %	0 ya da daha fazla karakter 1 karakter Çıkış karakteri
Aracıya gerçek zamanlı bağlantı	# +	Birden çok düzeyi eşleştir Tek bir düzeyle eşleştir
WebSphere Service Integration Bus	* // //.	Sıradüzendeki bir düzeydeki herhangi bir karakteri eşleştir 0 ya da daha fazla düzey eşleştir 0 ya da daha fazla düzey eşleştir (Konu ifadesinin sonunda)

## İlgili kavramlar

Konu URL tipi kaynak tanıtıcıları

Konu tekstili kaynak tanıtıcısı (URI), konunun adını belirtir; bir ya da daha çok özellik de belirtebilir.

Kuyruk tek biçimli kaynak tanıtıcıları

Kuyruğun URI 'si, kuyruğun adını belirtir; ayrıca, kuyruğun bir ya da daha fazla özelliğini de belirtebilir.

Geçici hedefler

XMS uygulamaları geçici hedefler oluşturabilir ve kullanabilir.

## İlgili başvurular

IDestination (.NET arabirimi için)

Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.

## Hedef Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla, Hedef nesnenin özelliklerine genel bakış.

## İleti üreticileri

XMSiçinde, bir ileti üreticisi geçerli bir hedefle ya da ilişkili bir hedef olmadan yaratılabilir. Boş değerli bir hedef içeren ileti üreticisi yaratırken, ileti gönderirken geçerli bir hedef belirtilmelidir.

### ***İlişkilendirilmiş hedefi olmayan ileti üreticileri***

XMS .NET' ta, boş değerli bir hedefle bir ileti üreticisi yaratılabilir.

.NET API kullanılırken ilişkili hedefi olmayan bir ileti üreticisi yaratmak için, NULL (boş) ISUse nesnesinin `CreateProducer()` yöntemine (örneğin, `session.CreateProducer(null)`) parametre olarak NULL (boş) geçirilmelidir. Ancak, ileti gönderildiğinde geçerli bir hedef belirtilmelidir.

### ***İlişkilendirilmiş hedefi olan ileti üreticileri***

Bu senaryoda, ileti üreticisi geçerli bir hedef kullanılarak yaratılır. Gönderme işlemi sırasında hedef belirtilmemesi gerekir.

## İleti tüketicileri

İleti tüketicileri, dayanıklı ve dayanıklı olmayan aboneler, zamanuyumlu ve zamanuyumsuz ileti tüketicileri olarak sınıflandırılabilir.

### ***Sürekli aboneler***

Kalıcı abone, abone etkin değilken yayınlanan iletiler de içinde olmak üzere, bir konuda yayınlanan tüm iletileri alan bir ileti tüketicisi 'dir.

Bu konudaki bilgiler, yalnızca bir uygulama bir IBM MQ kuyruk yöneticisine ya da WebSphere Application Server hizmet bütünleştirme veriyoluna bağlıysa anlamlıdır. Bilgi, bir aracıyla gerçek zamanlı bağlantı için uygun değildir.

Bir konuya ilişkin dayanıklı bir abone yaratmak için, uygulama, bir Oturum nesnesinin Durable Subscriber yöntemini çağıran, kalıcı aboneliği ve konuyu temsil eden bir Hedef nesneyi tanımlayan bir ad olarak belirtilen bir Oturum nesnesinin Create Durable Subscriber yöntemini çağırır. Uygulama, bir ileti seçici ile ya da bir ileti seçicisiz dayanıklı bir abone yaratabilir ve kalıcı abonenin kendi bağlantısı tarafından yayınlanan iletileri alıp almayacağını belirtebilir.

Kalıcı bir abone yaratmak için kullanılan oturumun ilişkili bir istemci tanıtıcısı olmalıdır. İstemci tanıtıcısı, oturumu yaratmak için kullanılan bağlantıyla ilişkili ile aynıdır; [“ConnectionFactories ve Connection nesnelere” sayfa 20](#) içinde açıklandığı şekilde belirtilir.

Kalıcı aboneliği tanımlayan ad, istemci tanıtıcısı içinde benzersiz olmalıdır; dolayısıyla, istemci tanıtıcısı, sürekli aboneliğin tam, benzersiz tanıtıcısının bir parçası olmalıdır. İleti alışverişi sunucusu, kalıcı aboneliğin kaydını tutar ve bu konuda yayınlanan tüm iletilerin, kalıcı abone tarafından kabul edilinceye ya da süre bitimine kadar saklanmalarını sağlar.

İleti alışverişi sunucusu, dayanıklı abone kapandıktan sonra bile sürekli abonelik kaydını tutmaya devam eder. Daha önce yaratılmış bir kalıcı aboneliği yeniden kullanmak için, bir uygulama aynı abonelik adını belirten bir kalıcı abone oluşturmalı ve sürekli abonelik ile ilişkili olan aynı istemci tanıtıcısına sahip bir oturum oluşturmalıdır. Belirli bir kalıcı abonelik için zaman zaman yalnızca bir oturum dayanıklı bir aboneye sahip olabilir.

Kalıcı bir aboneliğin kapsamı, aboneliğin kaydını koruyan ileti alışverişi sunucusudur. Farklı ileti sistemi sunucularına bağlı iki uygulama, aynı abonelik adını ve istemci tanıtıcısını kullanarak her bir kalıcı abone yaratır; tamamen bağımsız iki kalıcı abonelik yaratılır.

Bir kalıcı aboneliği silmek için, uygulama, kalıcı aboneliği tanımlayan ad olarak belirterek, bir Oturum nesnesinin Unsubscription yöntemini çağırır. Oturumla ilişkili istemci tanıtıcısı, sürekli abonelik ile ilişkili olan istemci tanıtıcısı olmalıdır. İleti alışverişi sunucusu, sürdürmekte olduğu dayanıklı aboneliğin kaydını siler ve kalıcı aboneye başka ileti göndermez.



Var olan bir aboneliği değiştirmek için, bir uygulama aynı abonelik adını ve istemci tanıtıcısını kullanarak, ancak farklı bir konu ya da ileti seçici (ya da her ikisi) belirtilerek kalıcı bir abone yaratabilir. Sürekli aboneliği değiştirmek, aboneliğin silinmesine ve yeni bir aboneliğin oluşturulabilmesiyle eşdeğerdir.

For an application that connects to IBM WebSphere MQ 7.0 queue manager, XMS manages the subscriber queues. Bu nedenle, uygulamanın bir abone kuyruğu belirtmesi gerekli değildir. XMS , belirtilirse, abone kuyruğunu yoksayacak.

Ancak, IBM WebSphere MQ 6.0 kuyruk yöneticisine bağlanan bir uygulama için, her dayanıklı abonenin atanmış bir abone kuyruğu olmalıdır. Bir konuya ilişkin abone kuyruğunun adını belirtmek için, konuyu simgeleyen Hedef nesnesinin `XMLC_WMQ_DUR_SUCQ` özelliğini ayarlayın. Varsayılan abone kuyruğu `SYSTEM.JMS.D.SUBSCRIBER.QUEUE`.

IBM WebSphere MQ 6.0 kuyruk yöneticilerine bağlanan sürekli aboneler tek bir abone kuyruğunu paylaşabilir ya da her bir kalıcı abone, kendi özel abone kuyruğundan kendi iletilerini alabilir. Uygulamanız için hangi yaklaşımın benimseneceği hakkında bir tartışma için bkz. *XMSIBM WebSphere MQ Using Java*.

Kalıcı bir abonelik için abone kuyruğunu değiştiremediğinizi unutmayın. Abone kuyruğunu değiştirmenin tek yolu, aboneliği silmeniz ve yeni bir abonelik yaratmasıdır.

Bir hizmet bütünleştirme veriyoluna bağlanan bir uygulama için, her dayanıklı abonenin belirlenmiş bir kalıcı aboneliği ana dizini olmalıdır. Aynı bağlantıyı kullanan tüm kalıcı aboneler için dayanıklı aboneliği ana dizini belirtmek için, bağlantıyı oluşturmak için kullanılan `ConnectionFactory` nesnesinin `XMLC_WPM_DUR_SUB_HOME` özelliğini ayarlayın. Tek bir konuya ilişkin dayanıklı aboneliği ana dizini belirtmek için, konuyu temsil eden Hedef nesnenin `XMLC_WPM_DUR_XX_ENCODE_CASE_ONE` sub\_home özelliğini ayarlayın. Bir uygulamanın bağlantıyı kullanan dayanıklı bir abone oluşturabilmesi için, bağlantı için sürekli abonelik ana dizini belirtilmelidir. Bir hedef için belirtilen herhangi bir değer, bağlantı için belirtilen değeri geçersiz kılar.

### ***Dayanıklı olmayan aboneler***

Dayanıklı olmayan bir abone, yalnızca abone etkin durumdayken yayınlanan iletileri alan bir ileti tüketicisi olur. Abone etkinlik dışı durumdayken teslim edilen iletiler kaybedilir.

Bu konudaki bilgiler, yalnızca bir IBM WebSphere MQ 6.0 kuyruk yöneticisi üzerinden yayınlama/abone olma ileti alışverişi kullanıyorsanız, bu bilgiler anlamlıdır.

Bağlantı kapatılmadan önce ya da kapanırken tüketici nesnelere silinmezse, iletiler artık etkin olmayan aboneler için aracı kuyruklarında kalabilir.

Bu durumda, JMS için IBM WebSphere MQ classes for JMS Sınıfları ile birlikte sağlanan Temizleme yardımcı programını kullanarak kuyruklar bu iletilerden temizlenebilir. Bu yardımcı programın kullanılmasına ilişkin ayrıntılar *IBM WebSphere MQ Using Java*. içinde sağlanır. Ayrıca, kuyruğun üzerinde çok sayıda ileti kaldıysa, abone kuyruğunun kuyruk derinliğini de artırmanız gerekebilir.

### ***Zamanuyumlu ileti kullanıcıları***

Zamanuyumlu ileti tüketicisi, iletileri bir kuyruktan zamanuyumlu olarak alır.

Zamanuyumlu ileti tüketicisi bir kerede bir ileti alır. `Receive(wait interval)` yöntemi kullanıldığında, çağrı bir ileti için ya da ileti tüketicisi kapatılıncaya kadar milisaniye cinsinden belirtilen bir süreyi bekler.

`ReceiveNoWait ()` yöntemi kullanılırsa, zamanuyumlu ileti tüketicisi herhangi bir gecikme olmadan ileti alır; sonraki ileti kullanılabilirse, hemen alınır; tersi durumda, boş değerli bir ileti nesnesine ilişkin gösterge döndürülür.

### ***Zamanuyumsuz ileti tüketicileri***

Zamanuyumsuz ileti tüketicisi bir kuyruktan zamanuyumsuz bir iletiyi alır. Kuyruğun üzerinde yeni bir ileti bulunduğu anda, uygulama tarafından kaydedilen ileti dinleyicisi çağrılır.

### ***Poison messages in XMS***

Bir zehir iletisi, alan bir MDB uygulaması tarafından işlenemeyen bir iletidir. Bir zehir iletiyle karşılaşırsa, XMS `MessageConsumer` nesnesi, bunu iki kuyruk özelliğine, `BOQUEUE` ve `BOTHRESH` ' a göre yeniden istekte buluntabilirler.

Bazı durumlarda, bir MDB ' ye teslim edilen bir ileti, bir IBM MQ kuyruğuna geri döndürülebilirdi. Bu durum, örneğin, daha sonra geriye işlenen bir iş birimi içinde bir ileti teslim ediliyorsa bu oluşabilir. Geriye işlenen bir ileti genel olarak yeniden teslim edilir, ancak hatalı biçimlendirilmiş bir ileti, bir MDB ' nin başarısız olmasına ve dolayısıyla teslim edilememesine neden olabilir. böyle bir mesaja zehir mesajı denir. You can configure IBM MQ so that the poison message is automatically transferred to another queue for further investigation or is discarded. IBM MQ ' i bu şekilde nasıl yapılandırabilmeye ilişkin bilgi için bkz. ASF' de zehirli iletilerin işlenmesi.

Bazen, hatalı biçimlendirilmiş bir ileti kuyruğun üzerine gelir. Bu bağlamda kötü biçimlendirilmiş, alma uygulamasının iletiyi doğru işleyemediği anlamına gelir. Böyle bir ileti, alma uygulamasının başarısız olmasına ve bu hatalı biçimlendirilmiş iletiyi geri almasına neden olabilir. Daha sonra, ileti giriş kuyruğuna sürekli olarak teslim edilebilir ve uygulama tarafından sürekli olarak yedeklenebilir. Bu mesajlar zehirli mesajlar olarak bilinir. XMS MessageConsumer nesnesi, zehirli iletileri saptar ve diğer bir hedefe yönlendirir.

IBM MQ kuyruk yöneticisi, her iletinin kaç kez geriletildiğine ilişkin bir kayıt tutar. Bu sayı yapılandırılabilir bir eşik değerine ulaştığında, ileti tüketicisi iletiyi adlandırılmış bir geri alma kuyruğuna sunan bir kuyruğa iletir. Bu istekte bulunanların herhangi bir nedenle başarısız olması durumunda, ileti giriş kuyruğundan kaldırılır ve istek-mektup kuyruğuna ya da kuyruktan atılır ya da atılır.

XMS ConnectionConsumer nesnelere, aynı kuyruk özelliklerini kullanarak ve aynı kuyruk özelliklerini kullanarak zehirli iletileri ele sağlar. Birden çok bağlantı uygulaması aynı kuyruğu izliyorsa, zehir iletilerinin, istek gerçekleşmeden önce bir uygulamaya eşik değerinden daha fazla zaman sağlanabileceği olabilir. Bu davranış, tek tek bağlantı tüketicilerinin kuyrukları izlemesine ve zehirli iletilerin istekte bulunmaya yol göstermesine neden olur.

Eşik değeri ve arka çıkış kuyruğunun adı bir IBM MQ kuyruğuna ilişkin özniteliklerdir. Özniteliklerin adları şunlardır: BackoutThreshold ve BackoutRequeueQName. Uyguladıkları kuyruk şu şekildedir:

- Noktadan noktaya ileti sistemi için bu, temeldeki yerel kuyruğdur. Bu, ileti tüketicileri ve bağlantı okuyucuları kuyruk diğer adlarını kullandığında önemlidir.
- IBM MQ ileti alışverişi sağlayıcısı olağan kipinde yayınlama/abone olma ileti alışverişi için, bu, Konu tarafından yönetilen kuyruğun yaratıldığı model kuyruğdur.
- IBM MQ ileti alışverişi sağlayıcısı geçiş kipinde yayınlama/abone olma ileti alışverişi için, bu, TopicConnectionFactory nesnesinde tanımlanmış CCSID ' dir ya da Konu nesnesinde tanımlanan CCDSUB kuyruğu.

BackoutThreshold ve BackoutRequeueQName özniteliklerini ayarlamak için aşağıdaki MQSC komutunu verin:

```
ALTER QLOCAL(your.queue.name) BOTHRESH(threshold value)
BOQUEUE(your.backout.queue.name)
```

Yayınlama/abone olma ileti sistemi için, sisteminiz her abonelik için bir dinamik kuyruk oluşturuyorsa, bu öznitelik değerleri, JMS model kuyruğu için WebSphere MQ sınıflarından edinilir SYSTEM.JMS.MODEL.QUEUE. Bu ayarları değiştirmek için şunu kullanın:

```
ALTER QMODEL(SYSTEM.JMS.MODEL.QUEUE) BOTHRESH(threshold value)
BOQUEUE(your.backout.queue.name)
```

Geriletme eşiği değeri sıfırsa, ileti işleme işlevi devre dışı bırakılır ve veri girişi kuyruğunda zehirli iletiler kalır. Ters durumda, geriletme sayısı eşik değerine ulaştığında, ileti, adı belirtilen geriletme kuyruğuna gönderilir.

Geriletme sayısı eşik değerine ulaşırsa, ancak ileti geriletme kuyruğuna gidemiyorsa, ileti ölü-mektup kuyruğuna gönderilir ya da ileti kalıcı değilse, atılır.

Bu durum, geriletme kuyruğu tanımlanmadıysa ya da MessageConsumer nesnesi iletiyi çıkış kuyruğuna gönderemezse oluşur.

## Sistem konfigürasyonunu, zehirli ileti işleme gerçekleştirmek için yapılandırma

The queue that XMS .NET uses when inquiring the **BOTHRESH** and **BOQNAME** attributes depends on the style of messaging being performed:

- Noktadan noktaya ileti sistemi için bu, temeldeki yerel kuyruğdur. Bir XMS .NET uygulaması, diğer ad kuyruklarından ya da küme kuyruklarından ileti tükettiğinde bu önemlidir.
- Yayınlama/abone olma ileti alışverişi için, bir uygulamaya ilişkin iletileri tutmak üzere yönetilen bir kuyruk yaratılır. XMS .NET , **BOTHRESH** ve **BOQNAME** özniteliklerine ilişkin değerleri belirlemek için yönetilen kuyruğu sorgular.

The managed queue is created from a model queue associated with the Topic object that the application has subscribed to, and inherits the values of the **BOTHRESH** and **BOQNAME** attributes from the model queue. Kullanılan model kuyruğu, alma uygulamasının dayanıklı mı yoksa dayanıklı olmayan bir abonelik mi çıkardığına bağlıdır:

- Dayanıklı abonelikler için kullanılan model kuyruğu, Konu 'nın **MDURMDL** özniteliği tarafından belirtilir. Bu özniteliğin varsayılan değeri SYSTEM . DURABLE . MODEL . QUEUE.
- Kalıcı olmayan abonelikler için, kullanılan model kuyruğu, **MNDURMDL** özniteliği tarafından belirtilir. **MNDURMDL** özniteliğinin varsayılan değeri SYSTEM . NDURABLE . MODEL . QUEUE.

When inquiring the **BOTHRESH** and **BOQNAME** attributes, XMS .NET:

- Yerel kuyruğu ya da bir diğer ad kuyruğu için hedef kuyruğu açar.
- Inquires the **BOTHRESH** and **BOQNAME** attributes.
- Yerel kuyruğu ya da bir diğer ad kuyruğu için hedef kuyruğu kapatır.

Yerel kuyruk açılırken kullanılan açma seçenekleri ya da bir diğer ad kuyruğu için hedef kuyruk, kullanılmakta olan IBM MQ sürümüne bağlıdır:

- IBM MQ 9.0.0 Fix Pack 8 için ve daha önceki bir yerde, yerel kuyruk ya da diğer ad kuyruğuna ilişkin hedef kuyruk bir küme kuyruğysa, XMS .NET , kuyruğu MQ00\_INPUT\_AS\_Q\_DEF, MQ00\_SORGULAMA ve MQ00\_FAIL\_IF QUIESCING seçenekleriyle açar. Bu, alma uygulamasını çalıştıran kullanıcının sorgu yürütmesi ve küme kuyruğunun yerel örneğine erişim elde etmesi gerektiği anlamına gelir.

XMS .NET , açık seçeneklerle MQ00\_SORING ve MQ00\_FAIL\_IF QUIESCINGaçık seçeneklerle diğer tüm yerel kuyruk tiplerini açar. XMS .NET ' un özniteliklerin değerlerini sorgulaması için, alma uygulamasını çalıştıran kullanıcının yerel kuyrukta sorgu erişimi olmalıdır.

- **V9.0.0.9** When using XMS .NET from IBM MQ 9.0.0 Fix Pack 9, the user running the receiving application must have inquire access on the local queue, regardless of the type of the queue.

To move poison messages to either a backout requeue queue or the queue manager's dead letter queue, you must grant the user running the application tak and düzgeçiş authorities.

### ASF ' de zehirli iletilerin işlenmesi

Application Server Facilitis (ASF) olanağını kullandığınızda, MessageConsumeryerine ConnectionConsumer, zehirli iletileri işler. ConnectionConsumer , kuyruğun BackoutThreshold ve BackoutRequeueQName özelliklerine göre iletilmekte olan iletileri sağlar.

Bir uygulama ConnectionConsumers' ı kullandığında, bir iletinin yedekleneceği koşullar, uygulama sunucusunun sağladığı oturma bağlıdır:

- Oturum, AUTO\_RELSE ya da DUP\_OK\_RELSE ile birlikte işlem dışı olduğunda, bir ileti yalnızca sistem hatasından sonra ya da uygulama beklenmedik bir şekilde sona erdirildikten sonra yedeklenir.
- Oturum CLIENT\_RELSE ile işlem dışı olduğunda, onaylanmamış iletiler uygulama sunucusu tarafından Session . recover () çağrısıyla yedeklenebilir.

Typically, the client implementation of MessageListener or the application server calls Message . acknowledge () . Message . acknowledge () , oturum üzerinde şu ana kadar sunulan tüm iletileri onaylar.

- When the session is transacted, unacknowledged messages can be backed out by the application server calling `Session.rollback()`.

## Kuyruk tarayıcıları

Uygulama, kuyruğu kaldırmadan kuyruklardaki iletilere göz atmak için bir kuyruk tarayıcısı kullanır.

Bir kuyruk tarayıcısı yaratmak için, uygulama, bir `ISUse` nesnesinin `Kuyruk Tarayıcısı Yarat` yöntemini çağırır ve bu nesnenin göz atılmasına ilişkin kuyruğu tanımlayan bir `Hedef nesne` olarak belirtilir. Uygulama, bir ileti seçici ile ya da ileti seçicisiz bir kuyruk tarayıcısı yaratabilir.

Kuyruk tarayıcısı yarattıktan sonra, uygulama, kuyruklardaki iletilerin listesini almak için `IQueueBrowser` nesnesinin `GetEnumerator` yöntemini çağırabilir. Yöntem, İleti nesnelerinin listesini sarmalayan bir sıralı değer listeleme döndürür. Listedeki ileti nesnelerinin sırası, iletilerin kuyruktan alınacağı sıralama düzeniyle aynıdır. Daha sonra, uygulama sırayla her iletiye göz atmak için sıralı değer numaracısı kullanabilir.

İletiler kuyruğa yerleştirildiği ve kuyruktan kaldırıldığı için, sıralı değer listeleme dinamik olarak güncellenir. Uygulamanın kuyruktan sonraki iletiye göz atmak için `IEnumerator.MoveNext()` çağırısının her çağırısında, ileti kuyruğun yürürlükteki içeriğini yansıtır.

Bir uygulama, belirli bir kuyruk tarayıcısı için bir kereden fazla `GetEnumerator` yöntemini çağırabilir. Her arama, yeni bir sıralı değer listeleme döndürür. Bu nedenle, uygulama bir kuyruktaki iletilere göz atmak ve kuyruğun içinde birden çok konum korumak için birden çok sıralı değer listeleme kullanabilir.

Bir uygulama, kuyruktan kaldırmak üzere uygun bir iletiyi aramak için bir kuyruk tarayıcısını kullanabilir ve daha sonra iletiyi kaldırmak için ileti seçicisiyle bir ileti tüketicisi kullanabilir. İleti seçici, iletiyi `JMSMessageID` üstbilgi alanının değerine göre seçebilir. Bu ve diğer JMS iletisi üstbilgi alanları hakkında bilgi için bkz. [“XMS iletisinde üstbilgi alanları” sayfa 67](#).

## İstekçi

Uygulama, bir istek iletisini göndermek için istekte bulunan bir istekte bulunan ve sonra yanıtı beklemek ve yanıtlamak için kullanılır.

Birçok ileti sistemi uygulaması, bir istek iletisi gönderen ve ardından yanıt bekleyen algoritmalara dayalıdır. XMS , bu uygulama stilinin geliştirilmesine yardımcı olmak için `İstekte Bulunana (Requestor)` adı verilen bir sınıf sağlar.

Bir istek sahibi yaratmak için, bir uygulama `Requestor` sınıfının `Requestor` oluşturucusunu çağırır, bir Oturum nesnesi ve istek iletilerinin nereye gönderileceğini tanımlayan bir `Destination` nesnesi olarak belirtilir. Oturum, `XMSC_CLIENT_RAND` ' in onaylanmaması ya da bir alındı bildirimini kipinin alınmamasıdır. Oluşturucu otomatik olarak, yanıt iletilerinin gönderileceği geçici bir kuyruk ya da konu yaratır.

İstek, istekte bulunduktan sonra, istek iletisini göndermek için `Requestor` nesnesinin `Request` yöntemini çağırabilir ve istek iletisini alan uygulamadan bir yanıt almak ve almak için istekte bulunmayı sağlar. Arama, yanıt alınmaya kadar ya da oturum sona erinceye kadar bekler (hangisi önce olursa). Her istek iletisi için istekte bulunanın tek bir yanıtı olması gerekir.

Uygulama, istekte bulunan ' yi kapattığında, geçici kuyruk ya da konu silinir. Ancak ilişkili oturum kapanmıyor.

## Nesne Silme

Bir uygulama yarattığı bir XMS nesnesini sildiğinde, XMS nesneye ayrılmış iç kaynakları serbest bırakır.

Bir uygulama XMS nesnesi yarattığında, XMS bellek ve diğer iç kaynakları nesneye ayırır. XMS retains these internal resources until the application explicitly deletes the object by calling the object's close or delete method, at which point XMS releases the internal resources. Bir uygulama önceden silinmiş bir nesneyi silmeye çalışırsa, arama yoksayılar.

Bir uygulama bir Bağlantı ya da Oturum nesnesini sildiğinde, XMS belirli ilişkili nesnelere otomatik olarak siler ve iç kaynaklarını serbest bırakır. Bunlar, `Connection` ya da `Session` nesnesi tarafından yaratılan ve nesneden bağımsız bir işlevi olmayan nesnelere. Bu nesnelere [Çizelge 4 sayfa 37](#) içinde gösterilir.

**Not:** bir uygulama bağımlı oturumlarla bir bağlantıyı kapattıysa, bu oturumlara bağımlı tüm nesnelere de silinir. Yalnızca bir Bağlantı ya da Oturum nesnesinin bağımlı nesnelere olabilir.

Çizelge 4. Otomatik olarak silinen nesnelere		
Silinen nesne	Yöntem	Otomatik olarak silinen bağımlı nesnelere
Bağlantı	Bağlantıyı Kapat	ConnectionMetaVeri ve Oturum nesnelere
Oturum	Oturumu Kapat	MessageConsumer, MessageProducer, QueueBrowse ve Requestor nesnelere

## Managed IBM MQ XA transactions through XMS

Yönetilen IBM MQ XA hareketleri XMS aracılığıyla kullanılabilir.

XA hareketlerini XMS aracılığıyla kullanmak için, hareket eden bir oturumun yaratılması gerekir. XA hareketi kullanımda olduğunda, hareket denetimi Distributed Transaction Coordinator (DTC) genel hareketlerinden geçer ve XMS oturumları değildir. XA hareketleri kullanılırken, `Session.commit` ya da `Session.rollback`, XMS oturumunda yayınlanamaz. Bunun yerine, işlemler `Transscope.Commit` ya da `Transscope.Rollback` DTC yöntemlerini kullanarak kesinleştirin ya da geri alma yöntemini kullanın. XA hareketi için bir oturum kullanılırsa, oturum kullanılarak yaratılan üretici ya da tüketici XA hareketinin bir parçası olmalıdır. Bunlar XA hareket kapsamı dışındaki herhangi bir işlem için kullanılamaz. Bunlar XA hareketinin dışında `Producer.send` ya da `Consumer.receive` gibi işlemler için kullanılamaz.

Aşağıdaki durumlarda bir `IllegalStateException` kural dışı durum nesnesi verilir:

- XA transacted seansı, `Session.commit` ya da `Session.rollback` için kullanılır.
- XA transacted oturumunda kullanılan üretici ya da tüketici nesnelere XA hareketi kapsamının dışında kullanılır.

XA hareketleri zamanuysuz tüketicilerde desteklenmez.

### Not:

1. XA hareketinin kesinleştirilmesinden önce `Producer`, `Consumer`, `Session` ya da `Connection` nesnesinde bir kapatma işlemi yayınlanabilir. Bu durumda, işlemdeki iletiler geriye işlenir. Benzer şekilde, XA hareketinin kesinleştirilmesinden önce bağlantı kesilirse, işlemdeki tüm iletiler geriye işlenir. `Producer` nesnesi için geri alma, iletilerin kuyruğa konmadığından emin olur. `Consumer` nesnesi için geri döndürme, iletilerin kuyruğun üzerinde kaldığı anlamına gelir.
2. If a `Producer` object puts a message with `TimeToLive` in the `TransactionScope` and a `commit` is issued after the time is elapsed, the message can expire before the `commit` is issued. Bu durumda, ileti `Consumer` nesnelere için kullanılabilir kılınmaz.
3. `Session` nesnelere iş parçacıkları arasında desteklenmez. İş parçacıkları arasında paylaşılan `Session` nesnelere yapılan işlemlerin kullanılması desteklenmez.

## XMS temel tipleri

XMS, sekiz Java temel tipi (byte, short, int, long, float, double, char ve boolean) eşdeğerleri sağlar. This allows the interchange of messages between XMS and JMS without data becoming lost or corrupted.

Çizelge 5 sayfa 37 lists the Java equivalent data type, size, and minimum and maximum value of each XMS primitive type.

Çizelge 5. XMS veri tipleri ve bunların Java eşdeğerleri				
XMS veri tipi	Uyumlu Java veri tipi	Büyükölük	Minimum değer	Maksimum değer
System.Boolean	bool	32 bit	yanlış	doğru
System.SBYTE	Byte	8 bit	-2 <sup>7</sup> (-128)	2 <sup>7</sup> -1 (127)

Çizelge 5. XMS veri tipleri ve bunların Java eşdeğerleri (devamı var)				
XMS veri tipi	UyumluJava veri tipi	Büyüklik	Minimum değer	Maksimum değer
System.BYTE	Byte	8 bit	$-2^7$ (-128)	$2^7-1$ (127)
System.CHAR	Byte	8 bit	$-2^7$ (-128)	$2^7-1$ (127)
System.Int16	kısa	16 bit	$-2^{15}$ (-32768)	$2^{15}-1$ (32767)
System.Int32	int	32 bit	$-2^{31}$ (-2147483648)	$2^{31}-1$ (2147483647)
System.Int64	uzun	64 bit	$-2^{63}$ (-9223372036854775808)	$2^{63}-1$ (9223372036854775807)
System.Single	kayan nokta	32 bit	-3.402823E-38 (7 basamaklı basamak sayısı)	3.402823E+38 (7 basamaklı basamak sayısı)
System.Double	çift	64 bit	-1.79769313486231E-308 (15 basamaklık basamak sayısı)	1.79769313486231E+308 (15 basamaklı basamak sayısı)

### İlgili kavramlar

Nesnelerin öznitelikleri ve özellikleri

Bir XMS nesnesi, nesneye ilişkin özellikler olan öznitelikler ve özellikler olabilir; bu, farklı şekillerde uygulanan özelliklerdir.

Bir özellik değerinin bir veri tipinden diğerine örtük olarak dönüştürülmesi

Bir uygulama bir özelliğin değerini aldığı anda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Hangi dönüştürmelerin desteklendiğini ve XMS ' in dönüştürmeleri nasıl gerçekleştirdiğini birçok kural yönetir.

### İlgili başvurular

Uygulama verilerine ilişkin öğeler için veri tipleri

To ensure that an XMS application can exchange messages with an IBM MQ classes for JMS application, both the applications must be able to interpret the application data in the body of a message in the same way.

### Bir özellik değerinin bir veri tipinden diğerine örtük olarak dönüştürülmesi

Bir uygulama bir özelliğin değerini aldığı anda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Hangi dönüştürmelerin desteklendiğini ve XMS ' in dönüştürmeleri nasıl gerçekleştirdiğini birçok kural yönetir.

Bir nesnenin özelliğinin bir adı ve değeri vardır; değerin ilişkili bir veri tipi vardır; bu değer, bir özelliğin değerinin de *özellik tipi*. olarak da anılır.

Bir uygulama, nesnelerin özelliklerini almak ve ayarlamak için PropertyContext sınıfının yöntemlerini kullanır. Bir özelliğin değerini almak için, bir uygulama, özellik tipine uygun yöntemi çağırır. Örneğin, bir tamsayı özelliğinin değerini almak için, bir uygulama tipik olarak GetIntözelliği yöntemini çağırır.

Ancak, bir uygulama bir özelliğin değerini aldığı anda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Örneğin, bir tamsayı özelliğinin değerini almak için, bir uygulama, özelliğin değerini dizgi olarak döndüren GetStringÖzelliği yöntemini çağırabilir. XMS tarafından desteklenen dönüştürmeler Çizelge 6 sayfa 38içinde gösterilir.

Çizelge 6. Bir özellik tipinden diğer veri tiplerine desteklenen dönüştürmeler	
Özellik tipi	Desteklenen hedef veri tipleri
System.String	System.Boolean, System.Double, System.Float, System.Int32, System.Int64, System.SByte, System.Int16

Çizelge 6. Bir özellik tipinden diğer veri tiplerine desteklenen dönüştürmeler (devamı var)

Özellik tipi	Desteklenen hedef veri tipleri
System.Boolean	System.String, System.SByte, System.Int32, System.Int64, System.Int16
System.Char	System.String
System.Double	System.String
System.Float	System.String, System.Double
System.Int32	System.String, System.Int64
System.Int64	System.String
System.SByte	System.String, System.Int32, System.Int64, System.Int16
System.SByte dizisi	System.String
System.Int16	System.String, System.Int32, System.Int64

Desteklenen dönüştürmeleri aşağıdaki genel kurallara göre düzenler:

- Sayısal özellik değerleri bir veri tipinden diğerine dönüştürülebilmesi için, dönüştürme sırasında hiçbir veri kaybolmaz. Örneğin, System.Int32 veri tipine sahip bir özelliğin değeri, System.Int64 veri tipi ile bir değere dönüştürülebilir, ancak System.Int16 veri tipi ile bir değere dönüştürülemez.
- Herhangi bir veri tipindeki bir özellik değeri bir dizgiye dönüştürülebilir.
- Bir dizgi özelliği değeri, dizginin dönüştürme için doğru şekilde biçimlendirilebilmesi koşuluyla, diğer herhangi bir veri tipine dönüştürülebilir. If an application attempts to convert a string property value that is not formatted correctly, XMS may return errors.
- Bir uygulama, desteklenmeyen bir dönüştürme girişiminde bulunursa, XMS bir hata döndürebilir.

Bir özellik değeri bir veri tipinden diğerine dönüştürüldüğünde aşağıdaki kurallar geçerlidir:

- Bir Boole özellik değerini dizgiye dönüştürürken, true değeri "true" dizgisine dönüştürülür ve false değeri "false" dizgisine dönüştürülür.
- Bir Boole özelliği değerini System.SByte'da içinde olmak üzere sayısal bir veri tipine dönüştürürken, true değeri 1'e dönüştürülür ve false değeri 0 olarak dönüştürülür.
- Bir dizgi özelliği değerini Boole değerine dönüştürürken, "doğru" dizgisi (büyük-küçük harfe duyarlı değil) ya da "1" true değerine dönüştürülür ve "yanlış" dizgisi (büyük-küçük harfe duyarlı değildir) ya da "0" false değerine dönüştürülür. Diğer tüm dizgiler dönüştürülemez.
- Bir dizgi özelliği değerini System.Int32, System.Int64, System.SByte'ya da System.Int16 veri tipli bir değere dönüştürürken, dizginin şu biçimde olması gerekir:

[boşluk] [sign]haneler

Dizgi bileşenleri aşağıdaki gibi tanımlanır:

#### **boşluklar**

İsteğe bağlı olarak baştaki boş karakterler.

#### **İşaret**

İsteğe bağlı artı işareti (+) ya da eksi işareti (-) karakteri.

#### **Rakamlar**

Ardışık rakam karakterleri dizisi (0-9). En az bir sayı karakteri var olmalıdır.

Sayı karakterleri sırasından sonra dizgi, rakam karakteri olmayan diğer karakterleri içerebilir, ancak bu karakterlerin ilk kez ulaşıldığı anda dönüştürme durakları durur. Dizilimin ondalık bir tamsayıyı temsil ettiği varsayılır.

XMS may return an error if the string is not formatted correctly.

- Bir dizgi özelliği değerini System.Double ya da System.Float veri tipli bir değere dönüştürürken, dizginin şu biçimde olması gerekir:

[boşluk] [işaret] [basamaklar] [nokta[d\_basamaklar]] [e\_char[e\_işareti]e\_basamakları]

Dizgi bileşenleri aşağıdaki gibi tanımlanır:

#### **boşluklar**

(İsteğe bağlı) Baştaki boş karakterler.

#### **İşaret**

(İsteğe bağlı) Artı işareti (+) ya da eksi işareti (-) karakteri.

#### **Rakamlar**

Ardışık rakam karakterleri dizisi (0-9). *basamaklar* ya da *d\_basamaklar* içinde en az bir sayı karakteri mevcut olmalıdır.

#### **nokta**

(İsteğe bağlı) Ondalık ayırıcı (.).

#### **d\_haneler**

Ardışık rakam karakterleri dizisi (0-9). *basamaklar* ya da *d\_basamaklar* içinde en az bir sayı karakteri mevcut olmalıdır.

#### **e\_char**

E ya da Eolan bir üstel karakter.

#### **e\_sign**

(İsteğe bağlı) Üst öge için artı işareti (+) ya da eksi işareti (-) karakteri.

#### **e\_basamakları**

Üs için bitişik sayı karakterleri dizisi (0-9). Dizgi bir üstel karakter içeriyorsa en az bir sayı karakteri var olmalıdır.

Sayı karakterleri sırasından sonra ya da bir üssü temsil eden isteğe bağlı karakterler sonra, dizgi sayı karakteri olmayan diğer karakterleri içerebilir, ancak bu karakterlere ilk ulaşıldığında dönüştürme durakları durur. Dizginin, 10 'un gücü olan bir üstel bir ondalık kayan noktalı sayıyı temsil ettiği varsayılır.

XMS may return an error if the string is not formatted correctly.

- Bir sayısal özellik değerini System.SByte veri tipindeki bir özellik değeri de içinde olmak üzere bir dizgiye dönüştürürken, değer, o değere ilişkin ASCII karakterini içeren dizgi değil, değer dizgi gösterimine ondalık sayı olarak dönüştürülür. For example, the integer 65 is converted to the string "65", not the string "A".
- Bir bayt dizisi özellik değerini bir dizgiye dönüştürürken, her bayt bayt 'ı temsil eden 2 onaltılı karaktere dönüştürülür. Örneğin, bayt dizisi {0xF1, 0x12, 0x00, 0xFF} , "F11200FF" dizgisine dönüştürülür.

Bir özellik tipinden diğer veri tiplerine dönüştürmeler hem Özellik hem de PropertyContext sınıflarının yöntemleri tarafından desteklenmektedir.

### **İlgili kavramlar**

#### **Nesnelerin öznitelikleri ve özellikleri**

Bir XMS nesnesi, nesneye ilişkin özellikler olan öznitelikler ve özellikler olabilir; bu, farklı şekillerde uygulanan özelliklerdir.

#### **XMS temel tipleri**

XMS , sekiz Java temel tipi (byte, short, int, long, float, double, double, char ve boolean) eşdeğerleri sağlar. This allows the interchange of messages between XMS and JMS without data becoming lost or corrupted.

### **İlgili başvurular**

#### **Eşleme iletileri**

Bir eşlem iletilerinin gövdesi, her değer ilişkili bir veri tipine sahip olduğu ad-değer çiftleri kümesini içerir.

#### **Akım iletileri**

Bir akış iletilerinin gövdesi, her değer ilişkili bir veri tipine sahip olduğu bir değerler akışı içerir.

#### **Uygulama verilerine ilişkin öğeler için veri tipleri**



To ensure that an XMS application can exchange messages with an IBM MQ classes for JMS application, both the applications must be able to interpret the application data in the body of a message in the same way.

## Yineleyiciler

Bir yineleyici, bir nesne listesini ve listedeki geçerli konumu koruyan bir imleci sarmalayan bir imleçle kaplıdır. The concept of an Iterator, as available in IBM Message Service Client for C/C++ is implemented by using IEnumerator interface in IBM Message Service Client for .NET.

Bir yineleyici yaratıldığında, imlecin konumu ilk nesneden önce olur. Uygulama, sırayla her bir nesneyi almak için bir yineleyici kullanır.

The Iterator class of IBM Message Service Client for C/C++ is equivalent to the Enumerator class in Java. XMS .NET , Java ile benzerdir ve bir IEnumerator arabirimi kullanır.

Bir uygulama, aşağıdaki görevleri gerçekleştirmek için bir IEnumerator kullanabilir:

- Bir iletinin özelliklerini almak için
- Bir harita iletisinin gövdesindeki ad değeri çiftlerini almak için
- Kuyruklardaki iletilere göz atmak için
- Bir bağlantı tarafından desteklenen JMS tanımlı ileti özelliklerinin adlarını almak için

## Kodlanmış karakter takımı tanıtıcıları

XMS .NET' ta, tüm dizgiler yerli .NET dizgisi kullanılarak geçirilir. Bu, sabit bir kodlamaya sahip olduğu için, bunu yorumlamak için daha fazla bilgi gerekmez. Bu nedenle, XMS .NET uygulamaları için XMLC\_CLIENT\_CCSSID özelliği gerekli değildir.

## XMS hatası ve kural dışı durum kodları

XMS , hataları göstermek için bir dizi hata kodu kullanır. Bu hata kodları, yayından yayına farklılık gösterebileceği için, bu belgelerde açıkça listelenmez. Yalnızca XMS kural dışı durum kodları (XMS\_X\_ biçiminde ...), XMS' un yayın düzeylerinde aynı kaldıkları için belgelenir.

## Kendi uygulamalarınızı oluşturma

Örnek uygulamaları oluştururken kendi uygulamalarınızı oluşturun.

Kendi .NET uygulamalarınızı oluşturmak için gereksinim duyacak önkoşulları da listeleyen [“.NET örnek uygulamalarının oluşturulması” sayfa 19](#) içinde açıklandığı şekilde .NET uygulamanızı oluşturun. Kendi uygulamalarınızı nasıl oluşturabilmeye ilişkin ek bilgi için, her bir örnek uygulama için sağlanan makefiles kullanın.

**İpucu:** Bir hata durumunda sorun tanısına yardımcı olmak için, içerilen simgelerle uygulamaları derlemenin yararlı olduğunu bulabilirsiniz.

### İlgili kavramlar

#### Örnek uygulamalar

Örnek uygulamalar, her bir API ' nin ortak özelliklerine genel bir bakış sağlar. Bunları, kuruluş ve ileti alışverişi sunucusu kurulumunuzu doğrulamak ve kendi uygulamalarınızı oluşturmanıza yardımcı olmak için kullanabilirsiniz.

### İlgili başvurular

#### .NET Arabirimler

Bu bölüm , .NET sınıfı arabirimlerini ve bunların özelliklerini ve yöntemlerini içerir.

#### XMS nesnelere özellikleri

Bu bölüm , XMStarafından tanımlanan nesne özelliklerini belgelerdir.

## **Automatic IBM MQ Client reconnection through XMS**

Configure your XMS client to reconnect automatically following a network, queue manager, or server failure while using IBM WebSphere MQ 7.1 client and later, as the message provider.

Otomatik olarak yeniden bağlanmak üzere istemci bağlantısı yapılandırmak için, `MQConnectionFactory` sınıfının `WMQ_CONNECTION_NAME_LIST` ve `WMQ_CLIENT_RECONNECT_OPTIONS` özelliklerini kullanın. Otomatik istemci yeniden bağlantısı, bağlantı hatasından sonra ya da kuyruk yöneticisini durdurduktan sonra bir seçenek olarak bir istemciyi yeniden kurar. Bazı istemci uygulamalarının tasarımı, bunları otomatik yeniden bağlantı için uygun olmayan bir duruma getirir.

Bağlantı kurulduktan sonra otomatik olarak yeniden bağlanabilir istemci bağlantıları yeniden bağlanabilir duruma gelir.

**Not:** The properties `İstemci Yeniden Bağlan Seçenekleri`, `İstemci Yeniden Bağlan Zamanaşımı`, and `Bağlantı Ad Listesi` can also be set via Client Channel Definitions Table (CCDT) or by enabling the client reconnection via the `mqlclient.ini` file.

**Not:** Yeniden bağlantı özellikleri `ConnectionFactory` nesnesi ve CCDT ' de olduğu gibi ayarlandıysa, öncelik kuralı aşağıdaki gibidir. `ConnectionFactory` nesnesinde bağlantı adı listesi özelliğinin varsayılan değeri ayarlandıysa, CCDT öncelik kazanır. Bağlantı adı listesi varsayılan değerine ayarlanmazsa, `ConnectionFactory` nesnesinde belirlenen özellik değerleri öncelikli olarak uygulanır. Bağlantı adı listesinin varsayılan değeri `localhost(1414)` değeridir.

## **Writing XMS .NET applications**

Bu bölümdeki konular, XMS .NET uygulamalarını yazarken size yardımcı olacak bilgiler sağlar.

Bu bölümde, XMS .NET uygulamalarının yazılması için özel bilgiler sağlanmaktadır. XMS uygulamalarını yazma hakkında genel bilgi için bkz. [“XMS uygulamaları yazılıyor” sayfa 19](#).

Bu bölüm aşağıdaki konuları içerir:

- [“.NET için veri tipleri” sayfa 42](#)
- [“.NET içindeki yönetilen ve yönetilmeyen işlemler” sayfa 43](#)
- [“.NET ' teki hedefler” sayfa 44](#)
- [“.NET içindeki özellikler” sayfa 45](#)
- [“.NET içinde var olmayan özellikler işleme” sayfa 45](#)
- [“Error handling in .NET” sayfa 46](#)
- [“Message and exception listeners in .NET” sayfa 46](#)

### **İlgili kavramlar**

[XMS uygulamaları yazılıyor](#)

Bu bölümdeki konular, XMS uygulamalarını yazarken size yardımcı olacak bilgiler sağlar.

### **İlgili başvurular**

[.NET Arabirimler](#)

Bu bölüm , .NET sınıfı arabirimlerini ve bunların özelliklerini ve yöntemlerini içerir.

## **.NET için veri tipleri**

XMS .NET , `System.Boolean`, `System.Byte`, `System.SByte`, `System.Char`, `System.String`, `System.Single`, `System.Double`, `System.Decimal`, `System.Int16`, `System.Int32`, `System.Int64`, `System.UInt16`, `System.UInt32`, `System.UInt64`, ve `System.Object` türlerini destekler. XMS .NET veri tipleri, XMS C/C++ için veri tiplerinden farklıdır. Bu konuyu, karşılık gelen veri tiplerini tanımlamak için kullanabilirsiniz.

Aşağıdaki çizelge, ilgili XMS .NET ve XMS C/C++ veri tiplerini göstermektedir ve bunları kısaca açıklar.

Çizelge 7. XMS .NET ve XMS C/C++ için veri tipleri

XMS .NET türü	XMS C/C++ tipi	Tanım
System.SByte	xmsSBYTE xmsINT8	İmzalı 8 bitlik değer
System.Byte	xmsBYTE xmsUINT8	İmsiz 8 bitlik değer
System.Int16	xmsINT16 xmsSHORT	İmzalı 16 bit değer
System.UInt16	xmsUINT16 xmsUSHORT	İmzalanmamış 16 bit değer
System.Int32	xmsINT32 xmsINT	İmzalı 32 bitlik değer
System.UInt32	xmsUINT32 xmsUINT	İmzasız 32 bitlik değer
System.Int64	xmsLONG xmsINT64	İmzalanmış 64 bitlik değer
System.UInt64	xmsULONG xmsUINT64	İmzasız 64 bit değer
System.Char	xmsCHAR16	İmsiz 16 bit karakter (.NET için Unicode)
System.Single	xmsFLOAT	IEEE 32 bit kayar noktalı sayı
System.Double	xmsDOUBLE	IEEE 64 bit kayar noktalı sayı
System.Boolean	xmsBOOL	True/False değeri
Burada geçerli değil	xmsCHAR	İmzalı ya da İmzalanmamış 8 bitlik değer (imzalanmış ya da imzalanmamış, platforma bağlıdır)
System.Decimal	Burada geçerli değil	96-bit signed integer times 10 <sup>0</sup> through 10 <sup>28</sup>
System.Object	Burada geçerli değil	Tüm tiplerin tabanı
System.String	Burada geçerli değil	Dizgi Tipi

## .NET içindeki yönetilen ve yönetilmeyen işlemler

Yönetilen kod, özel olarak .NET ortak dil çalıştırma zamanı ortamı içinde yürütülür ve bu çalıştırma zamanı tarafından sağlanan hizmetlere tamamen bağımlıdır. Uygulamanın herhangi bir bölümü .NET ortak dil çalıştırma zamanı ortamı dışında hizmet çalıştırdığında ya da çağırırsa, uygulama, yönetilmeyen bir uygulama olarak sınıflandırılır.

Şu anda yönetilen .NET ortamında belirli gelişmiş işlevsellik desteklenemez.

Uygulamanız, tam olarak yönetilen ortamda desteklenmeyen bazı işlevler gerektiriyorsa, uygulamanızı uygulamanıza önemli bir değişiklik gerektirmeden, yönetilmeyen ortamı kullanacak şekilde

değiştirebilirsiniz. Ancak, bu seçim yapıldığında XMS yığınının yönetilmeyen koddan kullanılmasını sağlamanız gerekir.

## IBM MQ kuyruk yöneticisine yönelik bağlantılar

WMQ\_CM\_Client ile yönetilen bağlantılar, TCP dışı iletişimi ve kanal sıkıştırma özelliğini desteklemeyecek. Ancak, bu bağlantılar yönetilmeyen bir bağlantı kullanılarak desteklenebilir (WMQ\_CM\_CLIENT\_UNManaged). Ek bilgi için [Developing .NET applications](#) başlıklı konuya bakın.

Yönetilmeyen bir ortamdaki yönetilen bir nesnenin bir bağlantı üreticisi yaratıyorsanız, bağlantı kipinin değerini XMLC\_WMQ\_CM\_CLIENT\_UNCEOTED olarak el ile değiştirmeniz gerekir.

## WebSphere Application Server hizmet bütünleştirme veriyolu ileti alışverişi altyapısına yönelik bağlantılar

SSL protokolünün (HTTPS dahil) kullanılmasını gerektiren bir WebSphere Application Server hizmet bütünleştirme veriyolu ileti alışverişi altyapısına yönelik bağlantılar şu anda yönetilen kod olarak desteklenmiyor.

### İlgili başvurular

[XMLC\\_WMQ\\_CONNECTION\\_MODE](#)

## .NET ' teki hedefler

.NET' ta hedefler, protokol tipine göre yaratılır ve yalnızca oluşturulduğu iletişim kuralı tipinde kullanılabilir.

Hedefler, biri konular ve biri kuyruklar için olmak üzere iki işlev sağlar:

- `IDestination CreateTopic(String topic);`
- `IDestination CreateQueue(String queue);`

Bu işlevler, API ' de aşağıdaki iki nesne üzerinde bulunur:

- `ISession`
- `XMSFactoryFactory`

Her iki durumda da bu yöntemler, aşağıdaki biçimde, parametreleri içerebilen bir URI stili dizisini kabul edebilir:

```
"topic://some/topic/name?priority=5"
```

Diğer bir seçenek olarak, bu yöntemler yalnızca bir hedef adı, yani bir adı olmayan bir adı (`//` ya da kuyruk: `//` öneki ve değiştirgelemleri olmadan kabul edebilir) kabul edebilir.

Bu, aşağıdaki URI stili dizisinin anlamına gelir:

```
CreateTopic("topic://some/topic/name");
```

aşağıdaki hedef adı ile aynı sonucu üretir:

```
CreateTopic("some/topic/name");
```

WebSphere Application Server hizmet tümleştirme veriyolu JMS için, konular, hem *topicname* hem de *topicspace* ' i içeren, ancak parametreleri içermeyecek bir kısa forma da belirtilebilir:

```
CreateTopic("topicspace:topicname");
```

## .NETiçindeki özellikler

Bir .NET uygulaması, nesnelerin özelliklerini almak ve ayarlamak için PropertyContext arabirimindeki yöntemleri kullanır.

PropertyContext arabirimi, özellikleri alan ve ayarlayan yöntemleri kapsüller. Bu yöntemler, aşağıdaki sınıflar tarafından doğrudan ya da dolaylı olarak edinilir:

- [BytesMessage](#)
- [Bağlantı](#)
- [ConnectionFactory](#)
- [ConnectionMetaVerileri](#)
- [Hedef](#)
- [MapMessage](#)
- [İleti](#)
- [MessageConsumer](#)
- [MessageProducer](#)
- [ObjectMessage](#)
- [QueueBrowser](#)
- [Oturum](#)
- [StreamMessage](#)
- [TextMessage](#)

Bir uygulama bir özelliğin değerini ayarlarsa, yeni değer özelliğin sahip olduğu önceki değerini yerini alır.

XMS özellikleriyle ilgili daha fazla bilgi için bkz. "[XMS nesnelerinin özellikleri](#)" sayfa 172.

For ease of use, XMS property names and values in XMS are predefined as public constants in a struct called XMSC. Bu sabitlerin adları XMSC biçiminde olur. *sabit*; örneğin, XMSC.USERID (bir özellik adı değişmezi) ve XMSC.DELIVERY\_AS\_APP (bir değer değişmezi).

Additionally, you can access IBM MQ constants by using the IBM.XMS.MQC struct. IBM.XMS ad alanı zaten içe aktarıldığında, bu özelliklerin değerlerine MQC biçiminde (*sabit*) erişebilirsiniz. Örneğin, MQC.MQRO\_COA\_WITH\_FULL\_DATA.

Ayrıca, .NET için hem XMS .NET hem de IBM MQ sınıflarını kullanan ve hem de IBM.XMS ve IBM.WMQ ad alanları, her yinelemenin benzersiz olduğundan emin olmak için MQC struct ad alanını tam olarak nitelemeniz gerekir.

Gelişmiş işlevsellik, yönetilen .NET ortamında şu anda desteklenmektedir. Ayrıntılar için "[.NETiçindeki yönetilen ve yönetilmeyen işlemler](#)" sayfa 43 dosyasına bakın.

## .NETiçinde var olmayan özellikler işleme

The handling of non-existent properties in the XMS .NET is broadly consistent with the JMS specification, and also with the C and C++ implementations of XMS.

JMS ' de var olmayan bir özelliğe erişimde, bir yöntem var olmayan (null) değeri gereken tipe dönüştürmeyi denediğinde Java sistemi kural dışı durumu ile sonuçlanabilir. Bir özellik yoksa, aşağıdaki kural dışı durumlar ortaya çıkar:

- `getStringÖzelliği` ve `getObjectÖzelliği` boş değer döndürür
- `getBooleanProperty` returns false because `Boolean.valueOf(null)` returns false
- `getIntProperty` etc.throw `java.lang.NumberFormatException` because `Integer.valueOf(null)` throws the exception

XMS .NET ' ta bir özellik yoksa aşağıdaki istisnalar ortaya çıkar:

- `GetStringÖzelliği` ve `GetObjectÖzelliği` (ve `GetBytesÖzelliği`) boş değer döndürür ( Javaile aynıdır)

- GetBoolean özelliği, System.NullReferenceException yayını atar.
- GetInt Özelliği vb. atar: System.NullReferenceException

Bu uygulama Java' den farklıdır; ancak, JMS belirtimiyle ve XMS C ve C++ arabirimleriyle genel olarak tutarlı olur. Java uygulaması gibi XMS .NET , çağırılan çağrıya System.Convert çağrısından herhangi bir kural dışı durum yayıyor. Ancak Java aksine, XMS sistem dönüştürme yordamlarıyla boş değer geçirerek, .NET çerçevesinin yerel davranışını kullanmak yerine, NullReferenceKural Dışı Durumlarını belirttik olarak atar. Uygulamanız "abc" gibi bir dizgiye bir özellik ayarlarsa ve GetInt özelliği çağırırsa, Convert.ToInt32("abc") tarafından verilen System.FormatException çağırısına, Java ile tutarlı bir şekilde geçirilir. MessageFormatKural dışı durumu, yalnızca SetProperty ve getProperty için kullanılan tipler uyumsuzsa oluşur. Bu davranış aynı zamanda Java ile de uyumludur.

## Error handling in .NET

XMS .NET özel durumlarının tümü System.Exception' dan türetilmiştir. XMS yöntemi çağrıları, MessageFormatKural Dışı Durumu, genel XMSExceptions ya da NullReferenceKural Dışı Durumu gibi sistem özel durumları gibi belirli XMS özel durumları yayınlatabilir.

Uygulama gereksinimlerine uygun olarak, belirli bir yakalama öbeklerinde ya da genel System.Exception yakalama öbeklerinde bu hatalardan birini yakalamak için uygulamalar yazın.

## Message and exception listeners in .NET

Bir .NET uygulaması, iletileri zamanuyumsuz olarak almak için bir ileti dinleyicisi kullanır ve bir bağlantı ile ilgili sorunun zamanuyumsuz olarak bildirilmesi için bir kural dışı durum dinleyicisini kullanır.

Hem iletinin hem de kural dışı durum dinleyicilerinin işlevi, .NET ile C + + için aynıdır. Ancak, bazı küçük uygulama farklılıkları vardır.

### .NET içindeki ileti dinleyicileri

İletileri zamanuyumsuz olarak almak için aşağıdaki adımları tamamlamanız gerekir:

1. İleti dinleyici temsilcisinin imzasıyla eşleşen bir yöntem tanımlayın. Tanımladığınız yöntem bir statik ya da somut örnek yöntemi olabilir ve erişilebilir herhangi bir sınıfta tanımlanabilir. Temsilci imzası şu şekildedir:

```
public delegate void MessageListener(IMessage msg);
```

ve yöntemi aşağıdaki gibi tanımlayabilirsiniz:

```
void SomeMethodName(IMessage msg);
```

2. Bu yöntemi, aşağıdaki benzer bir şeyi kullanarak temsilci olarak somutlaştırın:

```
MessageListener OnMsgMethod = new MessageListener(SomeMethodName)
```

3. Temsilciyi, tüketicinin MessageListener özelliğine göre ayarlayarak bir ya da daha fazla tüketiciyle kaydedin:

```
consumer.MessageListener = OnMsgMethod;
```

MessageListener ayarını boş değere ayarlayarak, yetkilendirmeyi kaldırabilirsiniz:

```
consumer.MessageListener = null;
```

## .NET ' te kural dışı durum

Kural dışı durum dinleyicisi, ileti dinleyicisiyle aynı şekilde çalışır, ancak farklı bir temsilci tanımlaması vardır ve bağlantı daha sonra ileti tüketicisi olarak atanır. Bu, C + + ile aynıdır.

1. Yöntemi tanımlayın. Temsilci imzası şu şekildedir:

```
public delegate void ExceptionListener(Exception ex);
```

ve bu nedenle tanımlanan yöntem şöyle olabilir:

```
void SomeMethodName(Exception ex);
```

2. Bu yöntemi, aşağıdakine benzer bir şeyi kullanarak temsilci olarak somutlaştırır:

```
ExceptionListener OnExMethod = new ExceptionListener(SomeMethodName)
```

3. Register the delegate with the connection by setting its ExceptionListener property:

```
connection.ExceptionListener = OnExMethod ;
```

You can remove the delegate by resetting the ExceptionListener to:

```
null: connection.ExceptionListener = null;
```

Hiçbir başvuru kalmadığında, sistem çöp toplayıcısı tarafından otomatik olarak kural dışı durumlar ya da iletiler silinir.

Aşağıda bir örnek kod yer alıyor:

```
using System;
using System.Threading;
using IBM.XMS;

public class Sample
{
    public static void Main()
    {
        XMSFactoryFactory factoryFactory = XMSFactoryFactory.GetInstance(XMSC.CT_RTT);

        IConnectionFactory connectionFactory = factoryFactory.CreateConnectionFactory();
        connectionFactory.SetStringProperty(XMSC.RTT_HOST_NAME, "localhost");
        connectionFactory.SetStringProperty(XMSC.RTT_PORT, "1506");

        //
        // Create the connection and register an exception listener
        //

        IConnection connection = connectionFactory.CreateConnection();
        connection.ExceptionListener = new ExceptionListener(Sample.OnException);

        ISession session = connection.CreateSession(false, AcknowledgeMode.AutoAcknowledge);
        IDestination topic = session.CreateTopic("topic://xms/sample");

        //
        // Create the consumer and register an async message listener
        //

        IMessageConsumer consumer = session.CreateConsumer(topic);
        consumer.MessageListener = new MessageListener(Sample.OnMessage);

        connection.Start();

        while (true)
        {
            Console.WriteLine("Waiting for messages...");
            Thread.Sleep(1000);
        }
    }
}
```

```
static void OnMessage(IMessage msg)
{
    Console.WriteLine(msg);
}

static void OnException(Exception ex)
{
    Console.WriteLine(ex);
}
}
```

## Yönetilen nesnelere çalışma

Bu bölümdeki konular denetlenen nesnelere ilgili bilgi sağlar. XMS uygulamaları, merkezi yönetilen bir nesne havuzundan nesne tanımlarını alabilir ve bu nesnelere, bağlantı fabrikaları ve hedefler oluşturmak için kullanılabilir.

This section provides information to help with creating and managing administered objects, describing the types of administered object repository that XMS supports. Bu bölümde ayrıca, bir XMS uygulamasının, denetlenen nesnelere havuzuna nasıl bağlantı yaptığı ve gerekli denetlenen nesnelere almak için nasıl bir bağlantı yaptığı da açıklanmaktadır.

Bu bölüm aşağıdaki konuları içerir:

- [“Desteklenen yönetilen nesne havuzu tipleri” sayfa 48](#)
- [“Denetlenen nesnelere için özellik eşlemesi” sayfa 49](#)
- [“Denetlenen ConnectionFactory nesnelere için gerekli özellikler” sayfa 52](#)
- [“Yönetilen Hedef nesnelere için gerekli özellikler” sayfa 54](#)
- [“Denetlenen nesnelere yaratılıyor” sayfa 55](#)
- [“InitialContext nesnelere” sayfa 57](#)
- [“InitialContext özellikleri” sayfa 58](#)
- [“XMS başlangıç bağlamları için URI biçimi” sayfa 59](#)
- [“JNDI Arama web hizmeti” sayfa 60](#)
- [“Yönetilen nesnelere alınması” sayfa 62](#)

### İlgili kavramlar

#### Denetlenen nesnelere

Denetlenen nesnelere kullanarak, merkezi bir havuzdan yönetilecek istemci uygulamaları tarafından kullanılan bağlantı ayarlarını yönetebilirsiniz. Uygulama, merkezi havuzdan nesne tanımlarını alır ve bunları ConnectionFactory ve Destination nesnelere oluşturmak için kullanır. Yönetilen nesnelere kullanarak, çalıştırma sırasında kullandıkları kaynaklardan birkaç uygulama da yapabilirsiniz.

### Desteklenen yönetilen nesne havuzu tipleri

Dosya Sistemi ve LDAP denetimli nesnelere, IBM MQ ve WebSphere Application Server'ye bağlanmak için kullanılabilir; ancak, COS Naming yalnızca WebSphere Application Server' e bağlanmak için kullanılabilir.

Dosya Sistemi nesne dizinleri, diziselleştirilmiş Java Naming Directory Interface (JNDI) nesnelere sahip olur. LDAP nesne dizinleri, JNDI nesnelere içeren dizinlerdir. File System and LDAP object directories can be administered by using either the JMSAdmin tool, which is provided with IBM WebSphere MQ 6.0, or the IBM MQ Explorer, which is provided with IBM WebSphere MQ 7.0 and later. Both the File system and the LDAP object directories can be used to administer client connections by centralizing IBM WebSphere MQ connection factories and destinations. Ağ yöneticisi, aynı merkezi havuza gönderme yapan birden çok uygulamayı konuşlandırabilir ve bu, merkezi havuzda yapılan bağlantı ayarlarındaki değişiklikleri yansıtmak için otomatik olarak güncellenir.

Bir COS adlandırma dizini, WebSphere Application Server service integration bus bağlantı fabrikalarını ve hedeflerini içerir ve WebSphere Application Server yönetim konsolu kullanılarak denetlenebilirler. For an XMS application to retrieve objects from the COS naming directory, a JNDI lookup web service must be



deployed. Bu web hizmeti tüm WebSphere Application Server service integration technologiesüzerinde kullanılamaz. Ayrıntılar için ürün belgelerine bakın.

**Not:** Nesne dizinindeki değişikliklerin yürürlüğe girmesi için uygulama bağlantılarını yeniden başlatın.

### **İlgili kavramlar**

Denetlenen nesnelere için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

InitialContext özellikleri

InitialContext oluşturucusunun değiştiricileri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesnelere havuzunun yerini içerir. Bir uygulamanın havuzla bağlantı kurmasını sağlamak için, URI içinde yer alan bilgilerden daha fazla bilgi sağlamak gerekebilir.

XMS başlangıç bağlamları için URI biçimi

Denetlenen nesnelere havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak sağlanır. URI biçimi bağlam tipine bağlıdır.

JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

Yönetilen nesnelere alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

Denetlenen nesnelere

Denetlenen nesnelere kullanarak, merkezi bir havuzdan yönetilecek istemci uygulamaları tarafından kullanılan bağlantı ayarlarını yönetebilirsiniz. Uygulama, merkezi havuzdan nesne tanımlarını alır ve bunları ConnectionFactory ve Destination nesnelere oluşturmak için kullanır. Yönetilen nesnelere kullanarak, çalıştırma sırasında kullandıkları kaynaklardan birkaç uygulama da yapabilirsiniz.

### **İlgili görevler**

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

InitialContext nesnelere

Bir uygulama, denetlenen nesnelere havuzuna, gerekli denetlenen nesnelere almak için bir bağlantı kurmak üzere kullanılacak bir başlangıç bağlamı oluşturmalıdır.

### **İlgili başvurular**

Denetlenen ConnectionFactory nesnelere için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

Yönetilen Hedef nesnelere için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

## **Denetlenen nesnelere için özellik eşlemesi**

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

Örneğin, bir IBM MQ JMS bağlantı üreticisinden alınan özelliklere sahip bir XMS bağlantı üreticisi yaratmak için, özelliklerin ikisi arasında eşlenmesi gerekir.

Tüm özellik eşlemeleri otomatik olarak gerçekleştirilir.

Aşağıdaki çizelge, bağlantı üreticilerinin ve hedeflerin en sık kullanılan özelliklerinden bazıları arasındaki eşlemeleri göstermektedir. Bu çizelgede gösterilen özellikler yalnızca küçük bir örnek kümedir ve gösterilen tüm özelliklerin tüm bağlantı tipleri ve sunucularla ilişkili olmamasını sağlar.

<i>Çizelge 8. Bağlantı üreticisi ve hedef özellikleri için ad eşleme örnekleri</i>		
<b>IBM MQ JMS özellik adı</b>	<b>XMS özellik adı</b>	<b>WebSphere Application Server service integration bus özellik adı</b>
KALICI (	<u>XMLC_DELIVERY_MODE</u>	
SÜRE BITİMİ (EXP)	<u>XMLC_TIME_TO_LIVE</u>	
PRIORITY (PRI)	<u>XMSC_PRIORITY</u>	
	<u>XMLC_WPM_HOST_NAME</u>	serverName
	<u>XMLC_WPM_BUS_ADı</u>	busName
	<u>XMSC_WPM_TOPIC_SPACE</u>	topicName

<i>Çizelge 9. XMS .NET özellikler</i>					
<b>Özellik</b>	<b>Nesne tipi</b>				
	<b>CF</b>	<b>QCF</b>	<b>TCF</b>	<b>Kuyruk</b>	<b>Konu</b>
<u>UYGULAMA ADı</u>	Y	Y	Y	Uyglnmz	Uyglnmz
<u>ASYNCEXCEPTI ON</u>	Y	Y	Y	Uyglnmz	Uyglnmz
<u>CCDTURL</u>	Y	Y	Y	Uyglnmz	Uyglnmz
<u>Kanal</u>	Y	Y	Y	Uyglnmz	Uyglnmz
<u>CONNECTIONN AMELIST</u>	Y	Y	Y	Uyglnmz	Uyglnmz
<u>CLIENTRECON NECTOTIONS</u>	Y	Y	Y	Uyglnmz	Uyglnmz
<u>CLIENTRECON NECTTIMEOUT</u>	Y	Y	Y	Uyglnmz	Uyglnmz
<u>CLIENTID</u>	Uyglnmz	Y	Uyglnmz	Uyglnmz	Uyglnmz
<u>COMMPHDR "1" sayfa 51</u>	Y	Uyglnmz	Y	Uyglnmz	Uyglnmz
<u>COMPUTE "1" sayfa 51</u>	Y	Y	Y	Uyglnmz	Uyglnmz
<u>CONNOPT "1" sayfa 51</u>	Y	Y	Y	Uyglnmz	Uyglnmz
<u>CONNTAG "1" sayfa 51</u>	Y	Y	Y	Uyglnmz	Uyglnmz
<u>Açıklama "1" sayfa 51</u>	Uyglnmz	Y	Uyglnmz	Y	Y
<u>EXPIRY "1" sayfa 51</u>	Uyglnmz	Uyglnmz	Uyglnmz	Y	Y
<u>FAILIFQUIESCE</u>	Y	Y	Y	Y	Y

Çizelge 9. XMS .NET özellikler (devamı var)

Özellik	Nesne tipi				
	CF	QCF	TCF	Kuyruk	Konu
Anasistem adı	Uyglınmz	Y	Uyglınmz	Uyglınmz	Uyglınmz
YEREL ADRES	Uyglınmz	Y	Uyglınmz	Uyglınmz	Uyglınmz
Kalıcılık	Uyglınmz	Uyglınmz	Uyglınmz	Y	Y
PORT	Uyglınmz	Y	Uyglınmz	Uyglınmz	Uyglınmz
ÖNCELİK "1" sayfa 51	Uyglınmz	Uyglınmz	Uyglınmz	Y	Y
PROVIDERVER SION "1" sayfa 51	Uyglınmz	Y	Uyglınmz	Uyglınmz	Uyglınmz
QASYER	Y	Y	Y	Y	Uyglınmz
KUYRUK "1" sayfa 51	Uyglınmz	Uyglınmz	Uyglınmz	Y	Uyglınmz
MUTABAKAT ZAMANLANDI	Y	Y	Y	Uyglınmz	Uyglınmz
Konu "1" sayfa 51	Uyglınmz	Uyglınmz	Uyglınmz	Uyglınmz	Y
TAŞIMA "1" sayfa 51	Uyglınmz	Y	Uyglınmz	Uyglınmz	Uyglınmz

**Not:**

1. Bu özelliklerde uygulama düzeyi özellikleri yoktur, ancak isteğe bağlı olarak yönetilen özellikler kullanılarak ayarlanabilir.

**İlgili kavramlar**

Desteklenen yönetilen nesne havuzu tipleri

Dosya Sistemi ve LDAP denetimli nesnelere, IBM MQ ve WebSphere Application Server'e bağlanmak için kullanılabilir; ancak, COS Naming yalnızca WebSphere Application Server'e bağlanmak için kullanılabilir.

InitialContext özellikleri

InitialContext oluşturucusunun değiştiricileri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesnelere havuzunun yerini içerir. Bir uygulamanın havuzla bağlantı kurmasını sağlamak için, URI içinde yer alan bilgilerden daha fazla bilgi sağlamak gerekebilir.

XMS başlangıç bağlamları için URI biçimi

Denetlenen nesnelere havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak sağlanır. URI biçimi bağlam tipine bağlıdır.

JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

Yönetilen nesnelere alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

**İlgili görevler**

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

### InitialContext nesneleri

Bir uygulama, denetlenen nesnelere havuzuna, gerekli denetlenen nesnelere almak için bir bağlantı kurmak üzere kullanılacak bir başlangıç bağlamı oluşturmalıdır.

### **İlgili başvurular**

Denetlenen ConnectionFactory nesneleri için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

Yönetilen Hedef nesneleri için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

IDestination (.NET arabirimi için)

Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.

Hedef Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla, Hedef nesnenin özelliklerine genel bakış.

IConnectionFactory (.NET arabirimi için)

Bir uygulama, bağlantı oluşturmak için bir bağlantı üreticisi kullanır.

ConnectionFactoryözellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren ConnectionFactory nesnesinin özelliklerine genel bakış.

## **Denetlenen ConnectionFactory nesneleri için gerekli özellikler**

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

Aşağıdaki çizelgede listelenen özellikler, bir uygulamanın ileti sistemi sunucusuna bağlantı yaratmak için ayarlanmak üzere gereken en küçük özelliklerdir. Bir bağlantının oluşturulma şeklini özelleştirmek istiyorsanız, uygulamanız gereken şekilde ConnectionFactory nesnesinin ek özelliklerini ayarlayabilir. Daha fazla bilgi için, bkz. "ConnectionFactoryözellikleri" sayfa 174. Kullanılabilir özelliklerin eksiksiz bir listesi yer alır.

## **IBM MQ kuyruk yöneticisiyle bağlantı**

*Çizelge 10. Bir IBM MQ kuyruk yöneticisine bağlanmak için yönetilen ConnectionFactory nesnelere ilişkin özellik ayarları*

<b>zorunluXMS</b>	<b>Eşdeğer IBM MQ JMS özelliği gerekli</b>
<u>XMLC_CONNECTION_TYPE</u>	XMS , bu özelliği bağlantı üreticisi sınıf adı ve TRANSPORT (TRAN) özelliğinden çıkar.
<u>XMLC_WMQ_HOST_NAME</u>	ANASISTEM ADI (ANASISTEM)
<u>XMLC_WMQ_PORT</u>	PORT
<u>XMLC_WMQ_QUEUE_YÖNETİCISI</u>	Kuyruk yöneticisinin adı

## **Aracıya gerçek zamanlı bağlantı**

*Çizelge 11. Bir aracıya gerçek zamanlı bağlantılar için denetlenen ConnectionFactory nesnelere ilişkin özellik ayarları*

<b>zorunluXMS</b>	<b>Eşdeğer IBM MQ JMS özelliği gerekli</b>
<u>XMLC_CONNECTION_TYPE</u>	XMS , bu özelliği bağlantı üreticisi sınıf adı ve TRANSPORT (TRAN) özelliğinden çıkar.
<u>XMLC_RTT_HOST_NAME</u>	ANASISTEM ADI (ANASISTEM)

Çizelge 11. Bir aracıya gerçek zamanlı bağlantılar için denetlenen ConnectionFactory nesnelere ilişkin özellik ayarları (devamı var)

<b>zorunluXMS</b>	<b>Eşdeğer IBM MQ JMS özelliği gerekli</b>
<u>XMLC_RTT_PORT</u>	PORT

## WebSphere Application Server service integration busbağlantısı

Çizelge 12. Bir WebSphere Application Server service integration busbağlantısı için yönetilen ConnectionFactory nesnelere ilişkin özellik ayarları

<b>XMS özellik</b>	<b>Tanım</b>
<u>XMLC_CONNECTION_TYPE</u>	Bir uygulamanın bağlandığı ileti sistemi sunucusu tipi.. Bu, bağlantı üretici sınıfı adından saptanır.
<u>XMLC_WPM_BUS_ADı</u>	Bir bağlantı üreticisi için, uygulamanın bağlandığı hizmet tümleştirme veriyolunun adı ya da hedefin varolduğu hizmet tümleştirme veriyolunun adı gibi bir hedef için bağlantı kurar.

### İlgili kavramlar

Desteklenen yönetilen nesne havuzu tipleri

Dosya Sistemi ve LDAP denetimli nesnelere, IBM MQ ve WebSphere Application Server'e bağlanmak için kullanılabilir; ancak, COS Naming yalnızca WebSphere Application Server'e bağlanmak için kullanılabilir.

Denetlenen nesnelere ilişkin özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

InitialContext özellikleri

InitialContext oluşturucusunun değiştiricileri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesnelere havuzunun yerini içerir. Bir uygulamanın havuzla bağlantı kurmasını sağlamak için, URI içinde yer alan bilgilerden daha fazla bilgi sağlamak gerekebilir.

XMS başlangıç bağlamları için URI biçimi

Denetlenen nesnelere havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak sağlanır. URI biçimi bağlam tipine bağlıdır.

JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

Yönetilen nesnelere alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

IBM MQ kuyruk yöneticisine güvenli bağlantı

Bir XMS .NET uygulamasını bir IBM MQ kuyruk yöneticisinde güvenli bağlantı sağlamak üzere etkinleştirmek için, ilgili özelliklerin ConnectionFactory nesnesinde tanımlı olması gerekir.

WebSphere Application Server service integration bus ileti alışverişi altyapısına güvenli bağlantılar

To enable an XMS .NET application to make secure connections to a WebSphere Application Server service integration bus messaging engine, the relevant properties must be defined in the ConnectionFactory object.

### İlgili görevler

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

InitialContext nesnelere

Bir uygulama, denetlenen nesnelere havuzuna, gerekli denetlenen nesnelere almak için bir bağlantı kurmak üzere kullanılacak bir başlangıç bağlamı oluşturmalıdır.

### İlgili başvurular

Yönetilen Hedef nesnelere için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

IConnectionFactory (.NET arabirimi için)

Bir uygulama, bağlantı oluşturmak için bir bağlantı üreticisi kullanır.

ConnectionFactory özellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren ConnectionFactory nesnesinin özelliklerine genel bakış.

## Yönetilen Hedef nesnelere için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

Çizelge 13. Yönetilen Hedef nesnelere ilişkin özellik ayarları		
Bağlantı tipi	Özellik	Tanım
IBM MQ Kuyruk Yöneticisi	KUYRUK (QU)	Bağlanmak istediğiniz kuyruk
	KONU (TOP)	Uygulamanın hedef olarak kullandığı konu
Aracıya gerçek zamanlı bağlantı	KONU (TOP)	Uygulamanın hedef olarak kullandığı konu
WebSphere Application Server service integration bus	topicName	Uygulamanız bir konuya bağlanıyorsa
	queueName	Uygulamanızın bir kuyruğa bağlanıyorsa

### İlgili kavramlar

Desteklenen yönetilen nesne havuzu tipleri

Dosya Sistemi ve LDAP denetimli nesnelere, IBM MQ ve WebSphere Application Server'ye bağlanmak için kullanılabilir; ancak, COS Naming yalnızca WebSphere Application Server' e bağlanmak için kullanılabilir.

Denetlenen nesnelere için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

InitialContext özellikleri

InitialContext oluşturucusunun değiştiricileri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesnelere havuzunun yerini içerir. Bir uygulamanın havuzla bağlantı kurmasını sağlamak için, URI içinde yer alan bilgilerden daha fazla bilgi sağlamak gerekebilir.

XMS başlangıç bağlamı için URI biçimi

Denetlenen nesnelere havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak sağlanır. URI biçimi bağlam tipine bağlıdır.

JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

Yönetilen nesnelere alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

### İlgili görevler

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

### InitialContext nesnelere

Bir uygulama, denetlenen nesnelere havuzuna, gerekli denetlenen nesnelere almak için bir bağlantı kurmak üzere kullanılacak bir başlangıç bağlamı oluşturmalıdır.

### **İlgili başvurular**

Denetlenen ConnectionFactory nesnelere için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

### IDestination (.NET arabirimi için)

Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.

### Hedef Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla, Hedef nesnenin özelliklerine genel bakış.

## **Denetlenen nesnelere yaratılıyor**

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

### **Başlamadan önce**

XMS ' in desteklediği yönetilen nesne havuzunun farklı tipleriyle ilgili daha fazla ayrıntı için bkz. "[Desteklenen yönetilen nesne havuzu tipleri](#)" sayfa 48.

### **Bu görev hakkında**

To create the administered objects for IBM MQ use the IBM MQ Explorer or IBM MQ JMS administration (JMSAdmin) tool.

To create the administered objects for IBM MQ or IBM Integration Bus, use the IBM MQ JMS administration (JMSAdmin) tool.

WebSphere Application Server service integration bus için yönetilen nesnelere oluşturmak üzere WebSphere Application Server yönetim konsolunu kullanın.

Aşağıdaki adımlar, denetlenen nesnelere oluşturmak için neler yapacağınızı özetlemektedir.

### **Yordam**

1. Bir bağlantı üreticisi yaratın ve seçtiğiniz sunucuya uygulamanızı kullanarak bir bağlantı yaratmak için gereken özellikleri tanımlayın.

XMS ' un bir bağlantı yapmak için gerekli olması gereken en düşük özellikler "[Denetlenen ConnectionFactory nesnelere için gerekli özellikler](#)" sayfa 52 içinde tanımlanır.

2. Uygulamanızın bağlanacağı ileti sunucusu sunucusunda gerekli hedefi yaratın:

- IBM MQ kuyruk yöneticisine yönelik bağlantı için bir kuyruk ya da konu oluşturun.
- Bir aracıya gerçek zamanlı bağlantı için bir konu oluşturun.
- WebSphere Application Server service integration bus ile bağlantı kurmak için bir kuyruk ya da konu yaratın.

XMS ' un bir bağlantı yapmak için gerekli olması gereken en düşük özellikler "[Yönetilen Hedef nesnelere için gerekli özellikler](#)" sayfa 54 içinde tanımlanır.

### **İlgili kavramlar**

Desteklenen yönetilen nesne havuzu tipleri

Dosya Sistemi ve LDAP denetimli nesnelere, IBM MQ ve WebSphere Application Server'ye bağlanmak için kullanılabilir; ancak, COS Naming yalnızca WebSphere Application Server' e bağlanmak için kullanılabilir.

Denetlenen nesnelere için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

#### InitialContext özellikleri

InitialContext oluşturucusunun değiştirgeleleri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesnelere havuzunun yerini içerir. Bir uygulamanın havuzla bağlantı kurmasını sağlamak için, URI içinde yer alan bilgilerden daha fazla bilgi sağlamak gerekebilir.

#### XMS başlangıç bağlamları için URI biçimi

Denetlenen nesnelere havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak sağlanır. URI biçimi bağlam tipine bağlıdır.

#### JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

#### Yönetilen nesnelere alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

#### Denetlenen nesnelere

Denetlenen nesnelere kullanarak, merkezi bir havuzdan yönetilecek istemci uygulamaları tarafından kullanılan bağlantı ayarlarını yönetebilirsiniz. Uygulama, merkezi havuzdan nesne tanımlarını alır ve bunları ConnectionFactory ve Destination nesnelere oluşturmak için kullanır. Yönetilen nesnelere kullanarak, çalıştırma sırasında kullandıkları kaynaklardan birkaç uygulama da yapabilirsiniz.

#### ConnectionFactory ve Connection nesnelere

ConnectionFactory nesnesi, uygulamanın Connection nesnesi yaratmak için kullandığı bir şablon sağlar. Uygulama, bir Oturma nesnesi yaratmak için Bağlantı nesnesini kullanır.

#### Hizmet tümleştirme veriyoluna bağlantı

Bir XMS uygulaması, doğrudan TCP/IP bağlantısı kullanarak ya da TCP/IP üzerinden HTTP kullanarak bir WebSphere Application Server hizmet tümleştirme veriyoluna bağlanabilir.

### **İlgili görevler**

#### InitialContext nesnelere

Bir uygulama, denetlenen nesnelere havuzuna, gerekli denetlenen nesnelere almak için bir bağlantı kurmak üzere kullanılacak bir başlangıç bağlamı oluşturmalıdır.

### **İlgili başvurular**

#### Denetlenen ConnectionFactory nesnelere için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

#### Yönetilen Hedef nesnelere için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

#### ConnectionFactory (.NET arabirimi için)

Bir uygulama, bağlantı oluşturmak için bir bağlantı üreticisi kullanır.

#### ConnectionFactory özellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren ConnectionFactory nesnesinin özelliklerine genel bakış.

#### IDestination (.NET arabirimi için)

Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.

#### Hedef Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla, Hedef nesnenin özelliklerine genel bakış.



## InitialContext nesneleri

Bir uygulama, denetlenen nesnelere havuzuna, gerekli denetlenen nesnelere almak için bir bağlantı kurmak üzere kullanılacak bir başlangıç bağlamı oluşturmalıdır.

### Bu görev hakkında

Bir InitialContext nesnesi, havuzla bir bağlantıyı sarsalıyor. XMS API, aşağıdaki görevleri gerçekleştirmek için yöntemler sağlar:

- Bir InitialContext nesnesi oluşturma
- Denetlenen nesne havuzunda denetlenen bir nesneye bakın.

Bir InitialContext nesnesi yaratılmasıyla ilgili daha fazla ayrıntı için bkz. [“InitialContext” sayfa 106](#) for .NET ve [“InitialContextözellikleri” sayfa 182](#).

### İlgili kavramlar

Desteklenen yönetilen nesne havuzu tipleri

Dosya Sistemi ve LDAP denetimli nesnelere, IBM MQ ve WebSphere Application Server'ye bağlanmak için kullanılabilir; ancak, COS Naming yalnızca WebSphere Application Server' e bağlanmak için kullanılabilir.

Denetlenen nesnelere için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

InitialContext özellikleri

InitialContext oluşturucusunun değiştiricileri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesnelere havuzunun yerini içerir. Bir uygulamanın havuzla bağlantı kurmasını sağlamak için, URI içinde yer alan bilgilerden daha fazla bilgi sağlamak gerekebilir.

XMS başlangıç bağlamları için URI biçimi

Denetlenen nesnelere havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak sağlanır. URI biçimi bağlam tipine bağlıdır.

JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

Yönetilen nesnelere alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

### İlgili görevler

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

### İlgili başvurular

Denetlenen ConnectionFactory nesnelere için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

Yönetilen Hedef nesnelere için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

InitialContext (.NET arabirimi için)

Bir uygulama, denetlenen nesnelere havuzundan alınan nesne tanımlamalarından nesne yaratmak için InitialContext nesnesini kullanır.

InitialContextözellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren, InitialContext nesnesinin özelliklerine genel bakış.

## InitialContext özellikleri

InitialContext oluşturucusunun deęiřtirgeleri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesnelere havuzunun yerini içerir. Bir uygulamanın havuzla baęlantı kurmasını saęlamak için, URI içinde yer alan bilgilerden daha fazla bilgi saęlamak gerekebilir.

JNDI 'da ve XMS' un .NET uygulamasında, ek bilgi bir ortam Hashtable içinde oluşturucuya saęlanır.

Denetlenen nesne havuzunun konumu, XMLC\_IC\_URL özelliğinde tanımlanır. Bu özellik genellikle Create çağrısında geçerilir, ancak arama işleminden önce farklı bir adlandırma dizinine baęlanmak için deęiřtirilebilir. FileSystem ya da LDAP baęlantıları için bu özellik, dizinin adresini tanımlar. COS adlandırması için bu, JNDI dizinine baęlanmak için bu özellikleri kullanan web hizmetinin adresidir.

Ařaęıdaki özellikler, JNDI dizinine baęlanmak için kullanacakları web hizmetine deęiřtirilmemiř olarak geçerilir.

- XMLC\_IC\_PROVIDER\_URL
- XMSC\_IC\_SECURITY\_CREDENTIAL
- XMSC\_IC\_SECURITY\_AUTHENTICATION
- XMSC\_IC\_SECURITY\_PRINCIPAL
- XMSC\_IC\_SECURITY\_PROTOCOL

### İlgili kavramlar

Desteklenen yönetilen nesne havuzu tipleri

Dosya Sistemi ve LDAP denetimli nesnelere, IBM MQ ve WebSphere Application Server'ye baęlanmak için kullanılabilir; ancak, COS Naming yalnızca WebSphere Application Server' e baęlanmak için kullanılabilir.

Denetlenen nesnelere için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server baęlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini saęlamak için, bu tanımlamalardan alınan özelliklerin, XMS baęlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

XMS bařlangıç baęlantıları için URI biçimi

Denetlenen nesnelere havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak saęlanır. URI biçimi baęlam tipine baęlıdır.

JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceęi bir forma çevirir.

Yönetilen nesnelere alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

### İlgili görevler

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alıřveriři sunucusuyla baęlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

InitialContext nesnelere

Bir uygulama, denetlenen nesnelere havuzuna, gerekli denetlenen nesnelere almak için bir baęlantı kurmak üzere kullanılacak bir bařlangıç baęlamı oluřturmalıdır.

### İlgili bařvurular

Denetlenen ConnectionFactory nesnelere için gerekli özellikler

Bir uygulama bir baęlantı üreticisi yarattığında, bir ileti alıřveriři sunucusuna yönelik baęlantı yaratmak için bir dizi özellik tanımlanmalıdır.

Yönetilen Hedef nesnelere için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

InitialContext (.NET arabirimi için)

Bir uygulama, denetlenen nesnelere havuzundan alınan nesne tanımlamalarından nesne yaratmak için InitialContext nesnesini kullanır.

#### InitialContext özellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren, InitialContext nesnesinin özelliklerine genel bakış.

## XMS başlangıç bağlamları için URI biçimi

Denetlenen nesnelere havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak sağlanır. URI biçimi bağlam tipine bağlıdır.

### FileSystem bağlamı

FileSystem bağlamına ilişkin URL, dosya sistemi tabanlı dizinin yerini gösterir. URL 'nin yapısı RFC 1738 tarafından tanımlandığı şekilde, *Uniform Resource Locators (URL)* (Uniform Resource Locators (URL)): URL adresi file:// örneğine sahiptir ve bu örneğin ardından sözdizimi, XMS ' in çalıştığı sistemde açılabilen geçerli bir dosya tanımdır.

Bu sözdizimi altyapıya özgü olabilir ve '/' ayırıcılar ya da '\' ayırıcıları kullanabilir. '\' kullanırsanız, her ayırıcının ek bir '\' kullanılarak çıkış karakteri olması gerekir. Bu, .NET çerçevesinin ayırıcıyı aşağıdaki gibi bir kaçış karakteri olarak yorumlamaya çalışmamasını önler.

Bu örnekler, bu sözdizimini göstermektedir:

```
file://myBindings
file:///admin/.bindings
file://\admin\.bindings
file://c:/admin/.bindings
file://c:\admin\.bindings
file://\\madison\\shared\admin\.bindings
file:///usr/admin/.bindings
```

### LDAP bağlamı

For the LDAP context, the basic structure of the URL is as defined by RFC 2255, *LDAP URL Biçimi*, with the case-insensitive prefix ldap://

Aşağıdaki örnekte, kesin sözdizim gösterilmektedir:

```
LDAP://[Hostname][:Port] ["/" [DistinguishedName]]
```

Bu sözdizimi, RFC içinde tanımlandığı şekilde, ancak öznitelikler, kapsam, süzgeçler ya da uzantılar için destek verilmeksizin tanımlanır.

Bu sözdizimine ilişkin örnekler:

```
ldap://madison:389/cn=JMSSData,dc=IBM,dc=UK
ldap://madison/cn=JMSSData,dc=IBM,dc=UK
LDAP:///cn=JMSSData,dc=IBM,dc=UK
```

### WSS bağlamı

For the WSS context, the URL is in the form of a web services endpoint, with the prefix http://.

Alternatively, you can use the prefix `cosnaming://` or `wsvc://`,

Bu iki örnek, http üzerinden erişilen URL ile WSS bağlamını kullandığınızın anlamına gelir. Bu, ilk bağlam tipinin doğrudan doğruya URL adresinden türetilebilmesini sağlar.

Bu sözdizimine örnek olarak şunlar verilebilir:

```
http://madison.ibm.com:9080/xmsjndi/services/JndiLookup  
cosnaming://madison/jndilookup
```

### **İlgili kavramlar**

Desteklenen yönetilen nesne havuzu tipleri

Dosya Sistemi ve LDAP denetimli nesnelere, IBM MQ ve WebSphere Application Server'ye bağlanmak için kullanılabilir; ancak, COS Naming yalnızca WebSphere Application Server'e bağlanmak için kullanılabilir.

Denetlenen nesnelere için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

InitialContext özellikleri

InitialContext oluşturucusunun değiştiricileri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesnelere havuzunun yerini içerir. Bir uygulamanın havuzla bağlantı kurmasını sağlamak için, URI içinde yer alan bilgilerden daha fazla bilgi sağlamak gerekebilir.

JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

Yönetilen nesnelere alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

### **İlgili görevler**

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

InitialContext nesnelere

Bir uygulama, denetlenen nesnelere havuzuna, gerekli denetlenen nesnelere almak için bir bağlantı kurmak üzere kullanılacak bir başlangıç bağlamı oluşturmalıdır.

### **İlgili başvurular**

Denetlenen ConnectionFactory nesnelere için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

Yönetilen Hedef nesnelere için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

InitialContext (.NET arabirimi için)

Bir uygulama, denetlenen nesnelere havuzundan alınan nesne tanımlamalarından nesne yaratmak için InitialContext nesnesini kullanır.

InitialContext özellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren, InitialContext nesnesinin özelliklerine genel bakış.

## **JNDI Arama web hizmeti**

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

Web hizmeti, kuruluş dizini içinde bulunan SIBXJndiLookupEAR.ear kurumsal arşiv dosyası içinde sağlanır. For the current release of IBM Message Service Client for .NET, SIBXJndiLookupEAR.ear can be found in the *install\_dir*\java\lib directory. Bu, yönetim konsolu ya da wsadmin komut dosyası oluşturma aracı kullanılarak bir WebSphere Application Server service integration bus sunucusu içinde kurulabilir. Web hizmeti uygulamalarının konuşlandırılmasına ilişkin ek bilgi edinmek için ürün belgelerine bakın.

Web hizmetini XMS uygulamaları içinde tanımlamak için, InitialContext nesnesinin XMLC\_IC\_URL özelliğini web hizmeti uç noktası URL 'sine ayarlamanız yeterlidir. Örneğin, web hizmeti MyServeradlı bir sunucu anasisteminde konuşlandırıldıysa, bir web hizmeti uç noktası URL 'si örneği:

```
wsvc://MyHost:9080/SIBXJndiLookup/services/JndiLookup
```

XMLC\_IC\_URL özelliğinin ayarlanması, tanımlı uç noktadaki web hizmetini çağırmak için InitialContext arama çağrılarına izin verir; böylece, gereken denetimli nesne COS adlandırma hizmetinden aranır.

.NET uygulamaları web hizmetini kullanabilir. Sunucu tarafı devreye alımı, XMS C, /C++ ve, XMS .NET için aynıdır. XMS .NET, web hizmetini doğrudan Microsoft .NET çerçevesi aracılığıyla çağırır.

### **İlgili kavramlar**

#### Desteklenen yönetilen nesne havuzu tipleri

Dosya Sistemi ve LDAP denetimli nesnelere, IBM MQ ve WebSphere Application Server'ye bağlanmak için kullanılabilir; ancak, COS Naming yalnızca WebSphere Application Server' e bağlanmak için kullanılabilir.

#### Denetlenen nesnelere için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

#### InitialContext özellikleri

InitialContext oluşturucusunun değiştiricileri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesnelere havuzunun yerini içerir. Bir uygulamanın havuzla bağlantı kurmasını sağlamak için, URI içinde yer alan bilgilerden daha fazla bilgi sağlamak gerekebilir.

#### XMS başlangıç bağlamları için URI biçimi

Denetlenen nesnelere havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak sağlanır. URI biçimi bağlam tipine bağlıdır.

#### Yönetilen nesnelere alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

#### İleti alışverişi sunucusu ortamının ayarlanması

The topics in this section describe how to set up the messaging server environment to allow XMS applications to connect to a server.

#### XMS örnek uygulamalarının kullanılması

Kuruluş ve ileti alışverişi sunucusu kurulumunuzu doğrulamak ve kendi uygulamalarınızı oluşturmanıza yardımcı olmak için XMS ile birlikte verilen örnek uygulamaları kullanın. Örnekler, her bir API 'nin ortak özelliklerine genel bir bakış sağlar.

### **İlgili görevler**

#### Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

#### InitialContext nesnelere

Bir uygulama, denetlenen nesnelere havuzuna, gerekli denetlenen nesnelere almak için bir bağlantı kurmak üzere kullanılacak bir başlangıç bağlamı oluşturmalıdır.

#### Kuruluş sihirbazını kullanarak Message Service Client for .NET 'in kurulması

Kuruluş, bir InstallShield X/Windows MSI kuruluş programı kullanır. Eksiksiz ya da özel bir kuruluş seçebileceğiniz iki ayar seçeneği vardır.

### **İlgili başvurular**

#### Denetlenen ConnectionFactory nesnelere için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

#### Yönetilen Hedef nesnelere için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

## Yönetilen nesnelerin alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

Alınacak nesneler aşağıdaki tipte adlara sahip olabilir:

- Hedef nesneyi tanımlayan basit bir ad; örneğin, SalesOrdersadlı bir kuyruk hedefi.
- SubContexts '/' ile ayrılmış, '/' ile ayrılmış ve nesne adıyla sona ermesi gereken bileşik ad. Bileşik ad örneği: "Warehouse/PickLists/DispatchQueue2"; burada Warehouse ve Picklists, adlandırma dizininde SubContexts ve DispatchQueue2 bir Hedef nesnesinin adıdır.

### İlgili kavramlar

Desteklenen yönetilen nesne havuzu tipleri

Dosya Sistemi ve LDAP denetimli nesneler, IBM MQ ve WebSphere Application Server'ye bağlanmak için kullanılabilir; ancak, COS Naming yalnızca WebSphere Application Server' e bağlanmak için kullanılabilir.

Denetlenen nesneler için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

InitialContext özellikleri

InitialContext oluşturucusunun değiştiricileri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesneler havuzunun yerini içerir. Bir uygulamanın havuzla bağlantı kurmasını sağlamak için, URI içinde yer alan bilgilerden daha fazla bilgi sağlamak gerekebilir.

XMS başlangıç bağlamları için URI biçimi

Denetlenen nesnelerin havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak sağlanır. URI biçimi bağlam tipine bağlıdır.

JNDI Arama web hizmeti

To access a COS naming directory from XMS, a JNDI Lookup web service must be deployed on a WebSphere Application Server service integration bus server. Bu web hizmeti, COS adlandırma hizmetindeki Java bilgilerini XMS uygulamalarının okuyabileceği bir forma çevirir.

### İlgili görevler

Denetlenen nesneler yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

InitialContext nesneleri

Bir uygulama, denetlenen nesneler havuzuna, gerekli denetlenen nesnelere almak için bir bağlantı kurmak üzere kullanılacak bir başlangıç bağlamı oluşturmalıdır.

### İlgili başvurular

Denetlenen ConnectionFactory nesneleri için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

Yönetilen Hedef nesneleri için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

InitialContext (.NET arabirimi için)

Bir uygulama, denetlenen nesneler havuzundan alınan nesne tanımlamalarından nesne yaratmak için InitialContext nesnesini kullanır.

InitialContext özellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren, InitialContext nesnesinin özelliklerine genel bakış.

## XMS uygulamaları için iletişimi güvenlik altına alma

Bu bölümde, XMS uygulamalarının bir WebSphere Application Server service integration bus ileti alışverişi altyapısına ya da IBM MQ kuyruk yöneticisine SSL aracılığıyla bağlanmasını sağlamak için güvenli iletişim kurulmasına ilişkin bilgiler sağlanır.

Bölüm aşağıdaki konuları içerir:

- [“IBM MQ kuyruk yöneticisine güvenli bağlantı” sayfa 63](#)
- [“Bir IBM MQ kuyruk yöneticisine yönelik bağlantılar içinCipherSuite ve CipherSpec ad eşlemeleri” sayfa 64](#)
- [“WebSphere Application Server service integration bus ileti alışverişi altyapısına güvenli bağlantılar” sayfa 64](#)
- [“WebSphere Application Server service integration busile bağlantı içinCipherSuite ve CipherSpec ad eşlemeleri” sayfa 65](#)

### IBM MQ kuyruk yöneticisine güvenli bağlantı

Bir XMS .NET uygulamasını bir IBM MQ kuyruk yöneticisinde güvenli bağlantı sağlamak üzere etkinleştirmek için, ilgili özelliklerin ConnectionFactory nesnesinde tanımlı olması gerekir.

The protocol used in the encryption negotiation can be either Secure Sockets Layer (SSL) or Transport Layer Security (TLS), depending on which CipherSuite you specify in the ConnectionFactory object.

IBM WebSphere MQ 7.0.0 Fix Pack 1 ve daha sonraki istemci kitaplıklarını kullanıyorsanız ve bir IBM WebSphere MQ 7.0 kuyruk yöneticisine bağlansanız, XMS uygulamasında aynı kuyruk yöneticisine birden çok bağlantı yaratabilirsiniz. Ancak, farklı kuyruk yöneticisine bağlantı yapılmasına izin verilmez. Girişimde bulunursanız, MQRC\_SSL\_ALREADY\_INITIALIZED hatasını elde edin.

IBM WebSphere MQ 6.0 ve sonraki istemci kitaplıklarını kullanıyorsanız, önce herhangi bir önceki SSL bağlantısını kapadığınızda bir SSL bağlantısı yaratabilirsiniz. Aynı süreçten aynı ya da farklı kuyruk yöneticilerine birden çok eşzamanlı SSL bağlantısının kullanılmasına izin verilmez. Birden fazla istek girişiminde bulunursanız, SSL bağlantısı için istenen bazı parametrelerin yoksayıldığı anlamına gelen MQRC\_SSL\_ALREADY\_INITIALIZEDuyarısını elde edebilirsiniz.

ConnectionFactory properties for connections via SSL to an IBM MQ queue manager, with a brief description, are shown in the following table:

Özellik adı	Tanım
<a href="#">XMLC_WMQ_SSL_CERT_STORES</a>	Bir kuyruk yöneticisiyle SSL bağlantısında kullanılacak sertifika iptal listelerini (CRL ' ler) tutan sunucuların konumları.
<a href="#">XMLC_WMQ_SSL_CIPHER_SPEC</a>	Kuyruk yöneticisiyle güvenli bir bağlantıda kullanılacak CipherSpec adı.
<a href="#">XMSC_WMQ_SSL_CIPHER_SUITE</a>	Bir kuyruk yöneticisine yönelik TLS bağlantısında kullanılacak CipherSuite ' in adı. Güvenli bağlantının pazarlığı sırasında kullanılan iletişim kuralı, belirtilen CipherSuite' e (CipherSuite) bağlıdır.
<a href="#">XMLC_WMQ_SSL_CRYPTO_HW</a>	İstemci sistemine bağlı şifreleme donanımlarıyla ilgili yapılandırma ayrıntıları.
<a href="#">XMLC_WMQ_SSL_FIPS_REQUIRED</a>	Bu özelliğin değeri, bir uygulamanın FIPS uyumlu olmayan şifreleme takımlarını kullanıp kullanamayabileceğini belirler. Bu özellik true değerine ayarlanırsa, istemci-sunucu bağlantısı için yalnızca FIPS algoritmaları kullanılır.

Çizelge 14. SSL aracılığıyla bir IBM MQ kuyruk yöneticisine bağlanmak için kullanılan ConnectionFactory özellikleri (devamı var)

Özellik adı	Tanım
<u>XMLC_WMQ_SSL_KEY_REPOSITORY</u>	Anahtarların ve sertifikaların depolanmış olduğu anahtar veritabanı dosyasının konumu.
<u>XMLC_WMQ_SSL_KEY_RESETCOUNT</u>	KeyResetCount, gizli anahtar yeniden anlaşılmadan önce, bir SSL etkileşimi içinde gönderilen ve alınan şifrelenmemiş baytların toplam sayısını temsil eder.
<u>XMLC_WMQ_SSL_PEER_NAME</u>	Bir kuyruk yöneticisiyle SSL bağlantısında kullanılacak eş adı.

### İlgili başvurular

IConnectionFactory (.NET arabirimi için)

Bir uygulama, bağlantı oluşturmak için bir bağlantı üreticisi kullanır.

ConnectionFactoryözellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren ConnectionFactory nesnesinin özelliklerine genel bakış.

Denetlenen ConnectionFactory nesneleri için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

### **Bir IBM MQ kuyruk yöneticisine yönelik bağlantılar içinCipherSuite ve CipherSpec ad eşlemeleri**

InitialContext , SSLCIPHERSUITE JMSAdmin Connection Factory özelliği SSLCIPHERSUITE ile XMS near-eşdeğeri XMLC\_WMQ\_SSL\_CIPHER\_SPEC çevreleri arasında çevrim yapar.

XMLC\_WMQ\_SSL\_CIPHER\_SUITE için bir değer belirtirseniz, ancak XMLC\_WMQ\_SSL\_CIPHER\_SPEC için değeri atlayın. Benzer bir çeviri gereklidir.

Çizelge 15 sayfa 64 , kullanılabilir CipherSpecs ve JSSE CipherSuite eşdeğerlerini listeler.

Çizelge 15. Kullanılabilir CipherSpecs ve JSSE CipherSuite eşdeğerleri

CipherSpec	Eşdeğer JSSE CipherSuite
TLS_RSA_WITH_3DES_EDE_CBC_SHA	SSL_RSA_WITH_3DES_EDE_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA	SSL_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA	SSL_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_DES_CBC_SHA	SSL_RSA_WITH_DES_CBC_SHA

### Not:

- TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA kullanımdan kaldırıldı. However, it can still be used to transfer up to 32 GB of data before the connection is terminated with error AMQ9288. Bu hatayı önlemek için, üçlü DES kullanmaktan kaçınmanız ya da bu CipherSpeckomutunu kullanırken gizli anahtar sıfırlamayı etkinleştirmeniz gerekir.

### **WebSphere Application Server service integration bus ileti alışverişi altyapısına güvenli bağlantılar**

To enable an XMS .NET application to make secure connections to a WebSphere Application Server service integration bus messaging engine, the relevant properties must be defined in the ConnectionFactory object.

XMS , WebSphere Service Integration Busile bağlantı için SSL ve HTTPS desteği sağlar. SSL ve HTTPS, kimlik doğrulama ve gizlilik için güvenli bağlantılar sağlar.



WebSphere güvenliği gibi, XMS güvenliği, güvenli bir bağlantı kararlaştırılırken kullanılan algoritmaları belirtmek üzere CipherSuites kullanımını içeren JSSE güvenlik standartları ve adlandırma kurallarıyla ilgili olarak yapılandırılır. The protocol used in the encryption negotiation can be either SSL or TLS, depending on which CipherSuite you specify in the ConnectionFactory object.

Çizelge 16 sayfa 65 , ConnectionFactory nesnesinde tanımlanması gereken özellikleri listeler.

Özellik adı	Tanım
<u>XMLC_WPM_SSL_CIPHER_SUITE</u>	Bir WebSphere Service Integration Bus ileti alışverişi altyapısına TLS bağlantısında kullanılacak CipherSuite ' in adı. Güvenli bağlantının pazarlığı sırasında kullanılan iletişim kuralı, belirtilen CipherSuite' e (CipherSuite) bağlıdır.
<u>XMLC_WPM_SSL_KEYRING_LABEL</u>	Sunucu ile kimlik doğrulaması yapılırken kullanılacak sertifika.

Aşağıda, WebSphere Application Server service integration bus ileti alışverişi altyapısına yönelik güvenli bağlantılar için bir ConnectionFactory özellikleri örneği yer alıyor:

```
cf.setStringProperty(XMSC_WPM_PROVIDER_ENDPOINTS, host_name:port_number:chain_name);
cf.setStringProperty(XMSC_WPM_SSL_KEY_REPOSITORY, key_repository_pathname);
cf.setStringProperty(XMSC_WPM_TARGET_TRANSPORT_CHAIN, transport_chain);
cf.setStringProperty(XMSC_WPM_SSL_CIPHER_SUITE, cipher_suite);
cf.setStringProperty(XMSC_WPM_SSL_KEYRING_STASH_FILE, stash_file_pathname);
```

Where chain\_name, BootstrapTunneledSecureMessaging ya da BootstrapSecureMessaging değerine ayarlanmalıdır ve kapı\_numarası, önyükleme sunucusunun gelen istekleri dinlediği kapının numarasıdır.

Aşağıda, örnek değerler eklenmiş bir WebSphere Application Server service integration bus ileti alışverişi altyapısına güvenli bağlantılar için ConnectionFactory özellikleri örneği gösterilmektedir:

```
/* CF properties needed for an SSL connection */
cf.setStringProperty(XMSC_WPM_PROVIDER_ENDPOINTS, "localhost:7286:BootstrapSecureMessaging");
cf.setStringProperty(XMSC_WPM_TARGET_TRANSPORT_CHAIN, "InboundSecureMessaging");
cf.setStringProperty(XMSC_WPM_SSL_KEY_REPOSITORY, "C:\\Program Files\\IBM\\gsk7\\bin\\
\\XMSkey.kdb");
cf.setStringProperty(XMSC_WPM_SSL_KEYRING_STASH_FILE, "C:\\Program Files\\IBM\\gsk7\\bin\\
\\XMSkey.sth");
cf.setStringProperty(XMSC_WPM_SSL_CIPHER_SUITE, "SSL_RSA_EXPORT_WITH_RC4_40_MD5");
```

### İlgili başvurular

ConnectionFactory (.NET arabirimi için)

Bir uygulama, bağlantı oluşturmak için bir bağlantı üreticisi kullanır.

ConnectionFactoryözellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren ConnectionFactory nesnesinin özelliklerine genel bakış.

Denetlenen ConnectionFactory nesneleri için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

### **WebSphere Application Server service integration busile bağlantı içinCipherSuite ve CipherSpec ad eşlemeleri**

GSKit, CipherSuitesyerine CipherSpecs kullandığından, XMLC\_WPM\_SSL\_CIPHER\_SUITE özelliğinde belirtilen JSSE stili CipherSuite adları GSKit stili CipherSpec adlarıyla eşlenmelidir.

Çizelge 17 sayfa 66 , tanınan her CipherSuiteiçin eşdeğer CipherSpec ' i listeler.

Çizelge 17. Kullanılabilir CipherSuites ve eşdeğeri CipherSpecs

CipherSuite	CipherSpec eşdeğeri
TLS_RSA_WITH_DES_CBC_SHA	TLS_RSA_WITH_DES_CBC_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA	TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA	TLS_RSA_WITH_AES_256_CBC_SHA

**Not:**

- TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA kullanımdan kaldırıldı. However, it can still be used to transfer up to 32 GB of data before the connection is terminated with error AMQ9288. Bu hatayı önlemek için, üçlü DES kullanmaktan kaçınmanız ya da bu CipherSpeckomutunu kullanırken gizli anahtar sıfırlamayı etkinleştirmeniz gerekir.

## XMS ileti

Bu bölümde, XMS iletilerinin yapısı ve içeriği ele alınmıştır ve uygulamaların XMS iletilerinin nasıl işleyeceğini açıklar.

Bu bölüm aşağıdaki konuları içerir:

- [“XMS iletilerinin parçaları” sayfa 66](#)
- [“XMS iletilerinde üstbilgi alanları” sayfa 67](#)
- [“XMS iletilerinin özellikleri” sayfa 68](#)
- [“XMS iletilerinin gövdesi” sayfa 71](#)
- [“İleti seçicileri” sayfa 77](#)
- [“Mapping XMS messages onto IBM MQ messages” sayfa 78](#)

### İlgili başvurular

[IMessage \(.NET arabirimi için\)](#)

Bir ileti nesnesi, uygulamanın gönderdiği ya da aldığı bir iletiyi gösterir. IMessage is a superclass for the message classes such as IMapMessage.

## XMS iletilerinin parçaları

XMS iletileri, bir üstbilgidenden, bir özellik kümesinden ve bir gövden oluşur.

### Üstbilgi

Bir iletilerin üstbilgisi alanlar içerir ve tüm iletiler aynı üstbilgi alanları kümesini içerir. XMS ve uygulamalar, iletileri tanımlamak ve yönlendirmek için üstbilgi alanlarının değerlerini kullanır. Üstbilgi alanlarıyla ilgili daha fazla bilgi için bkz. [“XMS iletilerinde üstbilgi alanları” sayfa 67](#).

### Özellik kümesi

İletinin özellikleri, iletiye ilişkin ek bilgileri belirtir. Tüm iletiler üstbilgi alanları kümesine sahip olsa da, her ileti farklı bir özellik kümesine sahip olabilir. Daha fazla bilgi için, bkz. [“XMS iletilerinin özellikleri” sayfa 68](#).

### Gövde

Bir iletilerin gövdesinde uygulama verileri bulunur. Daha fazla bilgi için, bkz. [“XMS iletilerinin gövdesi” sayfa 71](#).

Bir uygulama, almak istediği iletileri seçebilir. Seçim ölçütlerini belirten ileti seçicileri kullanarak. Ölçütler, belirli üstbilgi alanlarının değerlerini ve bir iletilerin özelliklerinin herhangi birinin değerlerini temel alarak olabilir. İleti seçiciyle ilgili daha fazla bilgi için bkz. [“İleti seçicileri” sayfa 77](#).

### İlgili başvurular

[XMS iletilerinde üstbilgi alanları](#)

To allow an XMS application to exchange messages with a WebSphere JMS application, the header of an XMS message contains the JMS message header fields.

#### XMS iletisinin özellikleri

XMS , üç tür ileti özelliğini destekler: JMS tanımlı özellikler, IBM tanımlı özellikler ve uygulama tarafından tanımlanan özellikler.

#### XMS iletisinin gövdesi

Bir iletinin gövdesinde uygulama verileri bulunur. Ancak, bir iletinin gövdesi olamaz ve yalnızca üstbilgi alanlarını ve özellikleri sıkıştırır.

#### İleti seçicileri

Bir XMS uygulaması, almak istediği iletileri seçmek için ileti seçicileri kullanır.

#### Mapping XMS messages onto IBM MQ messages

Bir XMS iletisinin JMS üstbilgi alanları ve özellikleri, IBM MQ iletisinin üstbilgi yapılarındaki alanlarla eşlenir.

## **XMS iletisinde üstbilgi alanları**

To allow an XMS application to exchange messages with a WebSphere JMS application, the header of an XMS message contains the JMS message header fields.

Bu üstbilgi alanlarının adları, JMS önekiyle başlar. JMS ileti üstbilgisi alanlarına ilişkin açıklamalar için *Java Message Service Belirtimi'* e bakın.

XMS implements the JMS message header fields as attributes of a Message object. Her üstbilgi alanının, değerini belirlemek ve almak için kendi yöntemleri vardır. Bu yöntemlere ilişkin açıklamalar için bkz. "İleti" sayfa 118. Bir üstbilgi alanı her zaman okunabilir ve yazılabilir bir alandır.

Çizelge 18 sayfa 67 , JMS ileti üstbilgisi alanlarını listeler ve iletilen bir ileti için her bir alanın değerinin nasıl ayarlanacağını gösterir. Uygulama bir ileti gönderdiğinde ya da JMSReelled durumunda, bir uygulama ileti aldığı anda, bazı alanlar otomatik olarak XMS tarafından ayarlanır.

<b>JMS ileti üstbilgisi alanının adı</b>	<b>İletilen bir ileti için değerin nasıl ayarlanıyor ( yöntem biçiminde [sınıf])</b>
JMSCorrelationID	Set JMSCorrelationID [ İleti]
JMSDeliveryMode	[MessageProducer] Gönder
JMSHedef	[MessageProducer] Gönder
JMSSüresi	[MessageProducer] Gönder
JMSMessageID	[MessageProducer] Gönder
JMSönceliği	[MessageProducer] Gönder
JMSRedird	[MessageConsumer] Al
JMSReplyTo	Set JMSReplyTo [ İleti]
JMSTimestamp	[MessageProducer] Gönder
JMSType	JMSType 'ı ayarla [ İleti]

### **İlgili başvurular**

#### XMS iletisinin parçaları

XMS iletisi, bir üstbilgiden, bir özellik kümesinden ve bir gövden oluşur.

#### XMS iletisinin özellikleri

XMS , üç tür ileti özelliğini destekler: JMS tanımlı özellikler, IBM tanımlı özellikler ve uygulama tarafından tanımlanan özellikler.

### XMS iletisinin gövdesi

Bir iletinin gövdesinde uygulama verileri bulunur. Ancak, bir iletinin gövdesi olamaz ve yalnızca üstbilgi alanlarını ve özellikleri sıkıştırır.

### İleti seçicileri

Bir XMS uygulaması, almak istediği iletileri seçmek için ileti seçicileri kullanır.

### Mapping XMS messages onto IBM MQ messages

Bir XMS iletisinin JMS üstbilgi alanları ve özellikleri, IBM MQ iletisinin üstbilgi yapılarındaki alanlarla eşlenir.

## **XMS iletisinin özellikleri**

XMS, üç tür ileti özelliğini destekler: JMS tanımlı özellikler, IBM tanımlı özellikler ve uygulama tarafından tanımlanan özellikler.

An XMS application can exchange messages with a WebSphere JMS application because XMS supports the following predefined properties of a Message object:

- WebSphere JMS tarafından desteklenen aynı JMStanımlı özellikler. Bu özelliklerin adları JMSX önekiyle başlar.
- WebSphere JMS tarafından desteklenen aynı IBMtanımlı özellikler. Bu özelliklerin adları JMS\_IBM\_ önekiyle başlar.

Önceden tanımlanmış her özelliğin iki adı vardır:

- A JMS name, for a JMS-defined property, or a WebSphere JMS name, for an IBM-defined property.

Bu ad, özelliğin JMS ya da WebSphere JMS içinde bulunduğu addır ve aynı zamanda bu özelliğe sahip bir iletiyle iletilen addır. Bir XMS uygulaması, bir ileti seçici ifadesindeki özelliği tanımlamak için bu adı kullanır.

- Bir ileti seçici ifadesi dışında tüm durumlarda özelliği tanımlamak için bir XMS adı. Her XMS adı, IBM.XMS.XMSC sınıfında adlandırılmış bir sabit olarak tanımlanır. Adlandırılmış değişmez değeri, ilgili JMS ya da WebSphere JMS adıdır.

Önceden tanımlanmış özelliklere ek olarak, bir XMS uygulaması kendi ileti özellikleri kümesini yaratabilir ve kullanabilir. Bu özellikler *Uygulama tanımlı özellikler* olarak adlandırılır.

Bir uygulama ileti yarattıktan sonra, iletinin özellikleri okunabilir ve yazılabilir olur. Uygulama iletiyi gönderdikten sonra özellikler okunabilir ve yazılabilir olarak kalır. Bir uygulama bir ileti aldığı anda, iletinin özellikleri salt okunur olur. Bir ileti, bir iletinin özellikleri salt okunur olduğunda İleti sınıfının Clear Properties yöntemini çağırıyorsa, özellikler okunabilir ve yazılabilir olur. Yöntem özellikleri de temizler.

Alınan ileti, ileti özelliklerini temizledikten sonra iletildiğinde, ileti özellikleri temizlenen standart bir WMQ XMS for .NET BytesMessage iletilmesinin davranışına uygun bir şekilde işlev görecektir.

Ancak, aşağıdaki özellikler kaybedildiğinden, bu önerilmez:

- JMS\_IBM\_Encoding özellik değeri, ileti verilerinin anlamlı bir şekilde kodu çözülemesini ima ediyor.
- JMS\_IBM\_Format özellik değeri; (MQMD ya da yeni MQRFH2) ileti üstbilgisi ile var olan üstbilgiler arasında üstbilgi zincirinin kırılacağını ima eder.

Bir iletinin tüm özelliklerinin değerlerini saptamak için, bir uygulama İleti sınıfının "get Properties" yöntemini çağırabilir. Yöntem, her özellik nesnesinin bir iletinin bir özelliğini gösterdiği özellik nesnelerinin listesini sarmalayan bir yineleyici yaratır. Böylece uygulama, her bir Özellik nesnesini sırayla almak için Yineator sınıfının yöntemlerini kullanabilir ve her özelliğin adını, veri tipini ve değerini almak için Özellik sınıfı yöntemlerini kullanabilir.

### **İlgili başvurular**

#### XMS iletisinin parçaları

XMS iletisi, bir üstbilgiden, bir özellik kümesinden ve bir gövden oluşur.

#### XMS iletisinde üstbilgi alanları

To allow an XMS application to exchange messages with a WebSphere JMS application, the header of an XMS message contains the JMS message header fields.

#### XMS iletisinin gövdesi

Bir iletinin gövdesinde uygulama verileri bulunur. Ancak, bir iletinin gövdesi olamaz ve yalnızca üstbilgi alanlarını ve özellikleri sıkıştırır.

#### İleti seçicileri

Bir XMS uygulaması, almak istediği iletileri seçmek için ileti seçicileri kullanır.

#### Mapping XMS messages onto IBM MQ messages

Bir XMS iletisinin JMS üstbilgi alanları ve özellikleri, IBM MQ iletisinin üstbilgi yapılarındaki alanlarla eşlenir.

### **JMS-iletinin tanımlı özellikleri**

Bir iletinin JMS tanımlı özellikleri hem XMS hem de WebSphere JMStarafından desteklenir.

Çizelge 19 sayfa 69 , hem XMS hem de WebSphere JMStarafından desteklenen bir iletinin JMS tanımlı özelliklerini listeler. JMStanımlı özelliklerin bir açıklaması için bkz. *Java Message Service Belirtimi*. JMStanımlı özellikler, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

Çizelge, her özelliğin veri tipini belirtir ve iletilen bir ileti için özelliğin değerinin nasıl ayarının kullanılacağını belirtir. Uygulama ileti gönderdiğinde ya da JMSXDeliveryCount(JMSXDeliveryCount) durumunda, bazı özellikler XMS tarafından otomatik olarak ayarlanır; bir uygulama ileti alır.

Çizelge 19. JMS-iletinin tanımlı özellikleri

JMS tanımlı özelliğinXMS adı	JMS ad	Veri türü	İletilen bir ileti için değerini nasıl ayarlanıyor ( yöntem biçiminde [sınıf])
JMS_APPID	JMSXAppID	System.String	[MessageProducer] Gönder
JMSX_DELIVERY_COUNT	JMSXDeliveryCount	System.Int32	[MessageConsumer] Al
JMSX_GROUPID	JMSXGroupID	System.String	Dizgi Özelliğini Ayarla [PropertyContext]
JMSX_GROUPSEQ	JMSXGroupSeq	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMSX_USERID	JMSXUserID	System.String	[MessageProducer] Gönder

### **IBM-iletinin tanımlı özellikleri**

Bir iletinin birden çok IBMtanımlı özelliği, XMS ve WebSphere JMStarafından desteklenir.

Çizelge 20 sayfa 70 lists the IBM defined properties of a message that are supported by both XMS and WebSphere JMS. IBMtanımlı özelliklerle ilgili daha fazla bilgi için IBM MQ ya da WebSphere Application Server ürün belgelerine bakın.

Çizelge, her özelliğin veri tipini belirtir ve iletilen bir ileti için özelliğin değerinin nasıl ayarının kullanılacağını belirtir. Uygulama bir ileti gönderdiğinde bazı özellikler otomatik olarak XMS tarafından ayarlanır.

Çizelge 20. IBM-iletinin tanımlı özellikleri

IBM tanımlı özelliğinXMS adı	WebSphere JMS adı	Veri türü	İletilen bir ileti için değerin nasıl ayarlanıyor (yöntem biçiminde [sınıf])
JMS_IBM_XX_ENCODE_C ASE_ONE charter_set	JMS_IBM_Character_Set	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_ENCODING	JMS_IBM_kodlama	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_EXCEPTIONMESSAGE	JMS_IBM_ExceptionMessage	System.String	[MessageConsumer] Al
JMS_IBM_EXCEPTIONREASON	JMS_IBM_ExceptionReason	System.Int32	[MessageConsumer] Al
JMS_IBM_EXCEPTIMESTAMP	JMS_IBM_ExceptionTimestamp	System.Int64	[MessageConsumer] Al
JMS_IBM_EXCEPTIONPROBLEM Hedef	JMS_IBM_ExceptionProblemHedef	System.String	[MessageConsumer] Al
JMS_IBM_FEEDBACK	JMS_IBM_Geri	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_	JMS_IBM_Biçimi	System.String	Dizgi Özelliğini Ayarla [PropertyContext]
JMS_IBM_LAST_MSG_IN_GROUP	JMS_IBM_Last_Msg_In_Group	System.Boolean	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_MSGTYPE	JMS_IBM_MsgType	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_PUTAPPLTYPE	JMS_IBM_PutApplTipi	System.Int32	[MessageProducer] Gönder
JMS_IBM_PUTDATE	JMS_IBM_PutDate	System.String	[MessageProducer] Gönder
JMS_IBM_PUTTIME	JMS_IBM_PutTime	System.String	[MessageProducer] Gönder
JMS_IBM_REPORT_COA	JMS_IBM_Report_COA	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_REPORT_COD	JMS_IBM_Report_COD	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_REPORT_DISCARD_MSG	JMS_IBM_Report_Disard_Msg	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_REPORT_EXCEPTION	JMS_IBM_Report_Exception	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_REPORT_EXPIRATION	JMS_IBM_Report_Son_Kullanma	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_REPORT_NAN	JMS_IBM_Report_NAN	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]

Çizelge 20. IBM-iletinin tanımlı özellikleri (devamı var)

IBM tanımlı özelliğin XMS adı	WebSphere JMS adı	Veri türü	İletilen bir ileti için değerin nasıl ayarlanıyor (yöntem biçiminde [sınıf])
JMS_IBM_REPORT_PAN	JMS_IBM_Report_PAN	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_REPORT_PASS_CORREL_Tanıtıcı	JMS_IBM_Report_Pass_Correl_ID	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_REPORT_PASS_MSG_ID	JMS_IBM_Report_Pass_Msg_ID	System.Int32	Tamsayı Özelliğini Ayarla [PropertyContext]
JMS_IBM_SYSTEM_MESS AGEID	JMS_IBM_System_Messa geID	System.String	[MessageProducer] Gönder

### İletiyeye ilişkin uygulama tanımlı özellikler

Bir XMS uygulaması kendi ileti özellikleri kümesini yaratabilir ve kullanabilir. Bir uygulama bir ileti gönderdiğinde, bu özellikler de iletiyle iletilir. Alma uygulaması, ileti seçicilerinin kullanılması, daha sonra bu özelliklerin değerlerine dayalı olarak hangi iletilerin almak istediğini seçebilir.

Bir WebSphere JMS uygulamasının bir XMS uygulaması tarafından gönderilen iletileri seçmesine ve işlemesine izin vermek için, uygulama tanımlı bir özelliğin adı, IBM MQ ürün belgelerinde belgelendiği gibi, ileti seçici ifadelerinde tanıtıcıların oluşturulmasına ilişkin kurallara uygun olmalıdır. Uygulama tanımlı bir özelliğin değeri şu veri tiplerinden birine sahip olmalıdır: System.Boolean, System.SByte, System.Int16, System.Int32, System.Int64, System.Float, System.Double ya da System.String.

### XMS iletisinin gövdesi

Bir iletinin gövdesinde uygulama verileri bulunur. Ancak, bir iletinin gövdesi olamaz ve yalnızca üstbilgi alanlarını ve özellikleri sıkıştırır.

XMS , ileti gövdesinin beş tipini destekler:

#### Bayt

Gövde bir bayt akışı içerir. Bu gövde tipine sahip bir ileti *byte iletisi*. olarak adlandırılır. IBytesMessage arabirimi, bir bayt iletisinin gövdedeki işleyişi işleme yöntemlerini içerir. Daha fazla bilgi için bkz. "[Byte iletileri](#)" sayfa 73.

#### Eşlem

Gövde, her değerin ilişkili bir veri tipine sahip olduğu ad-değer çiftleri kümesini içerir. Bu gövde tipine sahip bir ileti *Eşlem iletisi*. olarak adlandırılır. IMapMessage arabirimi, bir eşlem iletisinin gövdesinin işlenmesine ilişkin yöntemleri içerir. Daha fazla bilgi için bkz. "[Eşleme iletileri](#)" sayfa 74.

#### Nesne

Gövde, diziselleştirilmiş bir Java ya da .NET nesnesi içerir. Bu gövde tipine sahip bir ileti *Nesne iletisi*. olarak adlandırılır. IObjectMessage arabirimi, bir nesne iletisinin gövdesinin işlenmesine ilişkin yöntemleri içerir. Daha fazla bilgi için bkz. "[Nesne iletileri](#)" sayfa 74.

#### Akış

Gövde, her değerin ilişkili bir veri tipine sahip olduğu bir değerler akışı içerir. Bu gövde tipine sahip bir ileti *Akım iletisi*. olarak adlandırılır. IStreamMessage arabirimi, bir akış iletisinin gövdesinin işlenmesine ilişkin yöntemleri içerir. Daha fazla bilgi için bkz. "[Akım iletileri](#)" sayfa 75.

#### Metin

Gövde bir dizgi içeriyor. Bu gövde tipine sahip bir ileti *Metin iletisi*. olarak adlandırılır. ITextMessage arabirimi, bir metin iletisinin gövdesinin işlenmesine ilişkin yöntemleri içerir. Daha fazla bilgi için bkz. "[Metin iletileri](#)" sayfa 76.

IMessage arabirimi tüm ileti nesnelerinin üst ögesidir ve ileti alışverişi işlevlerinde XMS ileti tiplerinden herhangi birini temsil etmek için kullanılabilir.

Bu veri tiplerinin her birinin boyutu ve alt sınırı ve en düşük değerleri hakkında bilgi için bkz. [Çizelge 5 sayfa 37](#).

### **İlgili başvurular**

XMS iletisinin parçaları

XMS iletisi, bir üstbilgiden, bir özellik kümesinden ve bir gövden oluşur.

XMS iletisinde üstbilgi alanları

To allow an XMS application to exchange messages with a WebSphere JMS application, the header of an XMS message contains the JMS message header fields.

XMS iletisinin özellikleri

XMS , üç tür ileti özelliğini destekler: JMS tanımlı özellikler, IBM tanımlı özellikler ve uygulama tarafından tanımlanan özellikler.

İleti seçicileri

Bir XMS uygulaması, almak istediği iletileri seçmek için ileti seçicileri kullanır.

Mapping XMS messages onto IBM MQ messages

Bir XMS iletisinin JMS üstbilgi alanları ve özellikleri, IBM MQ iletisinin üstbilgi yapılarındaki alanlarla eşlenir.

### **Uygulama verilerine ilişkin öğeler için veri tipleri**

To ensure that an XMS application can exchange messages with an IBM MQ classes for JMS application, both the applications must be able to interpret the application data in the body of a message in the same way.

Bu nedenle, XMS uygulaması tarafından bir iletinin gövdesinde yazılan her uygulama verisi öğesinin, [Çizelge 21 sayfa 72](#) içinde listelenen veri tiplerinden birine sahip olması gerekir. Her veri tipi için, tabloda uyumlu Java veri türü gösterilir. XMS , uygulama verilerinin öğelerini yalnızca bu veri tipleriyle yazabilmek için yöntemleri sağlar.

<i>Çizelge 21. Java veri tipleriyle uyumlu XMS veri tipleri</i>		
<b>XMS Veri türü</b>	<b>Gösterir</b>	<b>Uyumlu Java veri tipi</b>
System.Boolean	Boole değeri true ya da false	boole
System.Char16	Çift baytlık karakter	DAMGA
System.SByte	İmzalı 8 bitlik tamsayı	Byte
System.Int16	İmzalı 16 bitlik tamsayı	kısa
System.Int32	İmzalı 32 bitlik tamsayı	int
System.Int64	İşaretsiz 64-bitlik tamsayı	uzun
System.Float	İmzalı kayan noktalı sayı	kayan nokta
System.Double	İmzalı çift duyarlıklı kayan noktalı sayı	çift
System.String	Karakter dizgisi	Dizgi

Bu veri tiplerinin her birinin boyutu, üst sınırı ve alt sınır değerleri hakkında bilgi için bkz. [“XMS temel tipleri” sayfa 37](#).

### **İlgili kavramlar**

Nesnelerin öznitelikleri ve özellikleri

Bir XMS nesnesi, nesneye ilişkin özellikler olan öznitelikler ve özellikler olabilir; bu, farklı şekillerde uygulanan özelliklerdir.



## XMS temel tipleri

XMS , sekiz Java temel tipi (byte, short, int, long, float, double, double, char ve boolean) eşdeğerleri sağlar. This allows the interchange of messages between XMS and JMS without data becoming lost or corrupted.

## Bir özellik değerinin bir veri tipinden diğerine örtük olarak dönüştürülmesi

Bir uygulama bir özelliğin değerini aldığı anda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Hangi dönüştürmelerin desteklendiğini ve XMS ' in dönüştürmeleri nasıl gerçekleştirdiğini birçok kural yönetir.

## **İlgili başvurular**

### Byte iletileri

Bir baytlar iletisinin gövdesi, bayt akışını içerir. Gövde yalnızca gerçek verileri içerir ve bu verileri yorumlamak için gönderme ve alma uygulamalarının sorumluluğunda olur.

### Eşleme iletileri

Bir eşlem iletisinin gövdesi, her değer ilişkili bir veri tipine sahip olduğu ad-değer çiftleri kümesini içerir.

### Nesne iletileri

Bir nesne iletisinin gövdesi serileştirilmiş birJava ya da .NET nesnesi içeriyor.

### Akım iletileri

Bir akış iletisinin gövdesi, her değer ilişkili bir veri tipine sahip olduğu bir değerler akışı içerir.

### Metin iletileri

Metin iletisi gövdesi bir dizgi içeriyor.

## ***Byte iletileri***

Bir baytlar iletisinin gövdesi, bayt akışını içerir. Gövde yalnızca gerçek verileri içerir ve bu verileri yorumlamak için gönderme ve alma uygulamalarının sorumluluğunda olur.

Bytes messages are useful if an XMS application needs to exchange messages with applications that are not using the XMS or JMS application programming interface.

Bir uygulama byte iletisi yarattıktan sonra, iletinin gövdesi salt yazılır olur. Uygulama, .NET için IBytesMessage arabiriminin uygun yazma yöntemlerini çağırarak uygulama verilerini gövde içine toplar. Uygulamanın, byte ileti akışına bir değer yazdığı anda, değer, uygulama tarafından yazılan önceki değerden hemen sonra birleştirilir. XMS , birleştirilen son baytın konumunu anımsamak için bir iç imleç tutar.

Uygulama iletiyi gönderdiğinde, iletinin gövdesi salt okunur hale gelir. Bu modda, uygulama iletiyi sürekli olarak gönderebilir.

Bir uygulama byte iletisi aldığı anda, iletinin gövdesi salt okunur olur. Uygulama, byte ileti akışının içeriğini okumak için IBytesMessage arabiriminin uygun okuma yöntemlerini kullanabilir. Uygulama, sıra içindeki byte 'ları okur ve XMS , okunan son baytın konumunu hatırlamak için bir iç imleç tutar.

Bir uygulama, bir byte iletisinin gövdesi yazılabilir olduğunda IBytesMessage arabiriminin ilk durumuna getirme yöntemini çağırırsa, gövde salt okunur olur. Yöntem ayrıca, imleci byte ileti akışının başlangıcındaki konumlarını da yeniden konumlandırır.

Bir uygulama, bir byte iletisinin gövdesi salt okunur olduğunda .NET için IMessage arabiriminin Clear Body yöntemini çağırduğunda, gövde yazılabilir hale gelir. Yöntem aynı zamanda vücudu da temizler.

## **İlgili başvurular**

### Uygulama verilerine ilişkin öğeler için veri tipleri

To ensure that an XMS application can exchange messages with an IBM MQ classes for JMS application, both the applications must be able to interpret the application data in the body of a message in the same way.

### Eşleme iletileri

Bir eşlem iletisinin gövdesi, her değer ilişkili bir veri tipine sahip olduğu ad-değer çiftleri kümesini içerir.

### Nesne iletileri

Bir nesne iletisinin gövdesi serileştirilmiş birJava ya da .NET nesnesi içeriyor.

### Akım iletileri

Bir akış iletisinin gövdesi, her değerin ilişkili bir veri tipine sahip olduğu bir değerler akışı içerir.

#### Metin iletileri

Metin iletisi gövdesi bir dizgi içeriyor.

#### IBytesMessage (.NET arabirimi için)

Bir bayt iletisi, gövdesi byte akışından oluşan bir iletidir.

### **Eşleme iletileri**

Bir eşlem iletisinin gövdesi, her değerin ilişkili bir veri tipine sahip olduğu ad-değer çiftleri kümesini içerir.

Her ad-değer çiftinde ad, değeri tanımlayan bir dizgidir ve değer, Çizelge 21 sayfa 72 içinde listelenen XMS veri tiplerinden birine sahip uygulama verileri ögesidir. Ad-değer çiftlerinin sırası tanımlı değil. MapMessage sınıfı, ad-değer çiftlerini ayarlamaya ve almaya ilişkin yöntemleri içerir.

Bir uygulama, adını belirterek bir ad-değer çiftine rasgele erişebiliyor.

Bir .NET uygulaması, eşlem iletisinin gövdesindeki adlara ilişkin sıralı değer listesi almak için MapNames özelliğini kullanabilir.

Bir uygulama bir ad-değer çiftinin değerini aldığı anda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Örneğin, bir eşleme iletisinin gövdesinden bir tamsayı almak için, bir uygulama MapMessage sınıfının GetString yöntemini çağırabilir; bu yöntem, tamsayıyı dizgi olarak döndürür. Desteklenen dönüştürmeler, XMS bir özellik değerini bir veri tipinden diğerine dönüştürdüğünde desteklenenler gibi aynıdır. Desteklenen dönüştürmeler hakkında daha fazla bilgi için bkz. "Bir özellik değerinin bir veri tipinden diğerine örtük olarak dönüştürülmesi" sayfa 38.

Bir uygulama, bir eşlem iletisi yarattıktan sonra, iletinin gövdesi okunabilir ve yazılabilir olur. Uygulama iletiyi gönderdikten sonra gövde okunabilir ve yazılabilir olarak kalır. Bir uygulama bir eşlem iletisi aldığı anda, iletinin gövdesi salt okunurdur. Bir uygulama, bir eşlem iletisinin gövdesi salt okunur olduğunda, ileti sınıfının Clear Body (Gövde Temizle) yöntemini çağırdığında, gövde okunabilir ve yazılabilir olur. Yöntem aynı zamanda vücudu da temizler.

### **İlgili kavramlar**

Bir özellik değerinin bir veri tipinden diğerine örtük olarak dönüştürülmesi

Bir uygulama bir özelliğin değerini aldığı anda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Hangi dönüştürmelerin desteklendiğini ve XMS ' in dönüştürmeleri nasıl gerçekleştirdiğini birçok kural yönetir.

### **İlgili başvurular**

Uygulama verilerine ilişkin öğeler için veri tipleri

To ensure that an XMS application can exchange messages with an IBM MQ classes for JMS application, both the applications must be able to interpret the application data in the body of a message in the same way.

#### Byte iletileri

Bir baytlar iletisinin gövdesi, bayt akışını içerir. Gövde yalnızca gerçek verileri içerir ve bu verileri yorumlamak için gönderme ve alma uygulamalarının sorumluluğunda olur.

#### Nesne iletileri

Bir nesne iletisinin gövdesi serileştirilmiş birJava ya da .NET nesnesi içeriyor.

#### Akım iletileri

Bir akış iletisinin gövdesi, her değerin ilişkili bir veri tipine sahip olduğu bir değerler akışı içerir.

#### Metin iletileri

Metin iletisi gövdesi bir dizgi içeriyor.

#### IMapMessage (.NET arabirimi için)

Eşlem iletisi, her değerin ilişkili bir veri tipine sahip olduğu bir ad-değer çiftleri kümesinden oluşan bir iletidir.

### **Nesne iletileri**

Bir nesne iletisinin gövdesi serileştirilmiş birJava ya da .NET nesnesi içeriyor.

Bir XMS uygulaması bir nesne iletisi alabilir, üstbilgi alanlarını ve özelliklerini değiştirebilir ve daha sonra başka bir hedefe gönderilebilir. Ayrıca, bir uygulama, bir nesne iletisinin gövini kopyalayabilir ve başka bir nesne iletisi oluşturmak için bu iletiyi kullanabilir. XMS , bir nesne iletisinin gövdeyi bayt dizisi olarak değerlendirir.

Bir uygulama bir nesne iletisi yarattıktan sonra, iletinin gövdesi okunabilir ve yazılabilir olur. Uygulama iletiyi gönderdikten sonra gövde okunabilir ve yazılabilir olarak kalır. Bir uygulama bir nesne iletisi aldığı anda, iletinin gövdesi salt okunurdur. Bir uygulama, bir nesne iletisinin gövdesi salt okunur olduğunda .NET için IMessage arabiriminin Clear Body yöntemini çağırdığında, gövde okunabilir ve yazılabilir hale gelir. Yöntem aynı zamanda vücudu da temizler.

### **İlgili başvurular**

Uygulama verilerine ilişkin öğeler için veri tipleri

To ensure that an XMS application can exchange messages with an IBM MQ classes for JMS application, both the applications must be able to interpret the application data in the body of a message in the same way.

Byte iletileri

Bir baytlar iletisinin gövdesi, bayt akışını içerir. Gövde yalnızca gerçek verileri içerir ve bu verileri yorumlamak için gönderme ve alma uygulamalarının sorumluluğunda olur.

Eşleme iletileri

Bir eşlem iletisinin gövdesi, her değerin ilişkili bir veri tipine sahip olduğu ad-değer çiftleri kümesini içerir.

Akım iletileri

Bir akış iletisinin gövdesi, her değerin ilişkili bir veri tipine sahip olduğu bir değerler akışı içerir.

Metin iletileri

Metin iletisi gövdesi bir dizgi içeriyor.

IObjectMessage (.NET arabirimi için)

Bir nesne iletisi, gövdesi serileştirilmiş bir Java ya da .NET nesnesinden farklı olan bir iletidir.

### **Akım iletileri**

Bir akış iletisinin gövdesi, her değerin ilişkili bir veri tipine sahip olduğu bir değerler akışı içerir.

Bir değerin veri tipi, [Çizelge 21 sayfa 72'](#) de listelenen XMS veri tiplerinden biridir.

Uygulama bir akış iletisi yarattıktan sonra, iletinin gövdesine yazılır. Uygulama, .NET için IMessage arabiriminin uygun yazma yöntemlerini çağırarak uygulama verilerini gövde içine toplar. Uygulamanın ileti akışına her değer yazdığı anda, değer ve veri tipi, uygulama tarafından yazılan önceki değerden hemen sonra birleştirilir. XMS , birleştirilen son değerin konumunu hatırlamak için bir iç imleç tutar.

Uygulama iletiyi gönderdiğinde, iletinin gövdesi salt okunur hale gelir. Bu modda, uygulama iletiyi birkaç kez gönderebilir.

Bir uygulama bir akış iletisi aldığı anda, iletinin gövdesi salt okunurdur. Uygulama, ileti akışının içeriğini okumak için, .NET için IMessage arabiriminin uygun okuma yöntemlerini kullanabilir. The application reads the values in sequence, and XMS maintains an internal cursor to remember the position of the last value that was read.

Bir uygulama ileti akışından bir değer okuduğunda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. For example, to read an integer from the message stream, an application can call the ReadString method, which returns the integer as a string. Desteklenen dönüştürmeler, XMS bir özellik değerini bir veri tipinden diğerine dönüştürdüğünde desteklenenler gibi aynıdır. Desteklenen dönüştürmeler hakkında daha fazla bilgi için bkz. [“Bir özellik değerinin bir veri tipinden diğerine örtük olarak dönüştürülmesi” sayfa 38.](#)

Bir uygulama ileti akışından bir değer okumayı denerken hata oluşursa, imleç gelişmiş değildir. Uygulama, değeri başka bir veri tipi olarak okuma girişiminde bulunarak hatadan kurtulabilir.

Bir uygulama, bir akış iletisinin gövdesi salt yazılır olduğunda XMS için IMessage arabiriminin ilk durumuna getirme yöntemini çağırıyorsa, gövde salt okunur olur. Yöntem ayrıca, imleci ileti akışının başlangıcındaki konumlara doğru konular.

Bir uygulama, bir akış iletisinin gövdesi salt okunur olduğunda XMS için IMessage arabiriminin Clear Body yöntemini çağırıyorsa, gövde salt yazılır olur. Yöntem aynı zamanda vücudu da temizler.

### **İlgili kavramlar**

Bir özellik değerinin bir veri tipinden diğerine örtük olarak dönüştürülmesi

Bir uygulama bir özelliğin değerini aldığı anda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Hangi dönüştürmelerin desteklendiğini ve XMS ' in dönüştürmeleri nasıl gerçekleştirdiğini birçok kural yönetir.

### **İlgili başvurular**

Uygulama verilerine ilişkin öğeler için veri tipleri

To ensure that an XMS application can exchange messages with an IBM MQ classes for JMS application, both the applications must be able to interpret the application data in the body of a message in the same way.

Byte iletileri

Bir baytlar iletisinin gövdesi, bayt akışını içerir. Gövde yalnızca gerçek verileri içerir ve bu verileri yorumlamak için gönderme ve alma uygulamalarının sorumluluğunda olur.

Eşleme iletileri

Bir eşlem iletisinin gövdesi, her değer ilişkili bir veri tipine sahip olduğu ad-değer çiftleri kümesini içerir.

Nesne iletileri

Bir nesne iletisinin gövdesi serileştirilmiş birJava ya da .NET nesnesi içeriyor.

Metin iletileri

Metin iletisi gövdesi bir dizgi içeriyor.

IStreamMessage (.NET arabirimi için)

Akış iletisi, her değer ilişkili bir veri tipine sahip olduğu bir değerler akışından oluşan bir iletidir. Gövdenin içeriği sırayla yazılır ve okunurdu.

### **Metin iletileri**

Metin iletisi gövdesi bir dizgi içeriyor.

Bir uygulama bir metin iletisi yarattıktan sonra, iletinin gövdesi okunabilir ve yazılabilir olur. Uygulama iletiyi gönderdikten sonra gövde okunabilir ve yazılabilir olarak kalır. Bir uygulama bir metin iletisi aldığı anda, iletinin gövdesi salt okunurdu. Bir uygulama, bir metin iletisinin gövdesi salt okunur olduğunda .NET için IMessage arabiriminin Clear Body yöntemini çağırduğunda, gövde okunabilir ve yazılabilir hale gelir. Yöntem aynı zamanda vücudu da temizler.

### **İlgili başvurular**

Uygulama verilerine ilişkin öğeler için veri tipleri

To ensure that an XMS application can exchange messages with an IBM MQ classes for JMS application, both the applications must be able to interpret the application data in the body of a message in the same way.

Byte iletileri

Bir baytlar iletisinin gövdesi, bayt akışını içerir. Gövde yalnızca gerçek verileri içerir ve bu verileri yorumlamak için gönderme ve alma uygulamalarının sorumluluğunda olur.

Eşleme iletileri

Bir eşlem iletisinin gövdesi, her değer ilişkili bir veri tipine sahip olduğu ad-değer çiftleri kümesini içerir.

Nesne iletileri

Bir nesne iletisinin gövdesi serileştirilmiş birJava ya da .NET nesnesi içeriyor.

Akım iletileri

Bir akış iletisinin gövdesi, her değer ilişkili bir veri tipine sahip olduğu bir değerler akışı içerir.

ITextMessage (.NET arabirimi için)

Metin iletisi, gövdesi bir dizeyi oluşturan bir iletidir.

## İleti seçicileri

Bir XMS uygulaması, almak istediği iletileri seçmek için ileti seçicileri kullanır.

Bir uygulama bir ileti tüketicisi yarattığında, bir ileti seçici ifadesini tüketicisiyle ilişkilendirebilir. İleti seçici ifadesi seçim ölçütlerini belirtir.

Bir uygulama IBM WebSphere MQ 7.0 kuyruk yöneticisine bağlanırken, ileti seçimi kuyruk yöneticisi tarafında yapılır. XMS herhangi bir seçim yapmaz ve yalnızca kuyruk yöneticisinden aldığı iletiyi sağlayarak daha iyi performans sağlar.

Bir uygulama, her biri kendi ileti seçici ifadesiyle birden çok ileti tüketicisi yaratabilir. Gelen bir ileti, birden fazla ileti tüketicisi seçim ölçütlerini karşılıyorsa, XMS bu tüketicinin her birine ileti gönderir.

Bir ileti seçici ifadesi, bir iletinin aşağıdaki özelliklerine gönderme yapabilir:

- JMS tanımlı özellikler
- IBM tanımlı özellikler
- Uygulama tanımlı özellikler

Ayrıca, aşağıdaki ileti üstbilgisi alanlarına da gönderme yapabilir:

- JMSCorrelationID
- JMSDeliveryMode
- JMSMessageID
- JMSönceliği
- JMSTimestamp
- JMSType

Ancak, bir ileti seçici ifadesi, bir iletinin gövdesindeki verilere gönderme yapamazlar.

Aşağıda, bir ileti seçici ifadesi örneği yer alıyor:

```
JMSPriority > 3 AND manufacturer = 'Jaguar' AND model in ('xj6','xj12')
```

XMS , bu ileti seçici ifadesiyle bir ileti tüketicisine yalnızca, ileti 3değerinden daha büyük bir önceliğe sahipse; uygulama tanımlı bir özellik, üretici, Jaguar ; değerine sahip ve başka bir uygulama tanımlı özellik, model ( xj6 ya da xj12 . değeriyle) içeren bir ileti sağlar.

XMS içinde bir ileti seçici ifadesi oluşturmak için kullanılan sözdizimi kuralları, IBM MQ classes for JMS' de olanlar ile aynıdır. Bir ileti seçici ifadesinin nasıl oluşturulacağı hakkında bilgi için, IBM MQ ürün belgelerine bakın; bir ileti seçici ifadesinde, JMStanımlı özelliklerin adları JMS adları olmalıdır ve IBM-tanımlı özelliklerin adları IBM MQ classes for JMS adlarıyla olmalıdır. Bir ileti seçici ifadesindeki XMS adlarını kullanamazsınız.

### İlgili başvurular

[XMS iletisinin parçaları](#)

XMS iletisi, bir üstbilgiden, bir özellik kümesinden ve bir gövden oluşur.

[XMS iletisinde üstbilgi alanları](#)

To allow an XMS application to exchange messages with a WebSphere JMS application, the header of an XMS message contains the JMS message header fields.

[XMS iletisinin özellikleri](#)

XMS , üç tür ileti özelliğini destekler: JMS tanımlı özellikler, IBM tanımlı özellikler ve uygulama tarafından tanımlanan özellikler.

[XMS iletisinin gövdesi](#)

Bir iletinin gövdesinde uygulama verileri bulunur. Ancak, bir iletinin gövdesi olamaz ve yalnızca üstbilgi alanlarını ve özellikleri sıkıştırır.

[Mapping XMS messages onto IBM MQ messages](#)

Bir XMS iletisinin JMS üstbilgi alanları ve özellikleri, IBM MQ iletisinin üstbilgi yapılarındaki alanlarla eşlenir.

## Mapping XMS messages onto IBM MQ messages

Bir XMS iletisinin JMS üstbilgi alanları ve özellikleri, IBM MQ iletisinin üstbilgi yapılarındaki alanlarla eşlenir.

When an XMS application is connected to an IBM MQ queue manager, messages sent to the queue manager are mapped onto IBM MQ messages in the same way that IBM MQ classes for JMS messages are mapped onto IBM MQ messages in similar circumstances.

Bir Hedef nesnenin `XMLC_WMQ_TARGET_CLIENT` özelliği `XMSC_WMQ_TARGET_DEST_JMS` olarak ayarlanırsa, hedefe gönderilen bir iletinin JMS üstbilgi alanları ve özellikleri, IBM MQ iletisinin `MQMD` ve `MQRFH2` üstbilgi yapılarındaki alanlarla eşlenir. `XMLC_WMQ_TARGET_CLIENT` özelliğinin bu şekilde ayarlanması, iletiyi alan uygulamanın bir `MQRFH2` üstbilgisini işleyebileceğini varsayar. Alma uygulaması başka bir XMS uygulaması, bir IBM MQ classes for JMS uygulaması ya da bir `MQRFH2` üstbilgisini işlemek için tasarlanmış yerel bir IBM MQ uygulaması olabilir.

Hedef nesnenin `XMLC_WMQ_TARGET_CLIENT` özelliği `XMLC_WMQ_TARGET_DEST_MQ` olarak ayarlandıysa, hedefe gönderilen bir iletinin JMS üstbilgi alanları ve özellikleri, IBM MQ iletisinin `MQMD` üstbilgi yapısındaki alanlarla eşlenir. İleti bir `MQRFH2` üstbilgisi içermiyor ve `MQMD` üstbilgi yapısındaki alanlarla eşlenemeyen herhangi bir JMS üstbilgi alanı ve özellikleri yoksa. Bu nedenle, iletiyi alan uygulama, `MQRFH2` üstbilgisini işlemek için tasarlanmamış bir yerel IBM MQ olabilir.

IBM MQ messages received from a queue manager are mapped onto XMS messages in the same way that IBM MQ messages are mapped onto IBM MQ classes for JMS messages in similar circumstances.

Gelen bir IBM MQ iletisinde `MQRFH2` üstbilgisi varsa, sonuçtaki XMS iletisinin tipi, `MQRFH2` üstbilgisinin `mcd` klasöründe bulunan **Msd** özelliğinin değeri tarafından belirlenir. **Msd** özelliği `MQRFH2` üstbilgisinde yoksa ya da IBM MQ iletisinin `MQRFH2` üstbilgisi yoksa, sonuçtaki XMS iletisinin, tipi `MQMD` üstbilgisindeki *Format* alanının değeri tarafından belirlenen bir gövde vardır. *Format* alanı `MQFMT_STRING` olarak ayarlandıysa, XMS iletisi bir metin iletidir. Ters durumda, XMS iletisi bir bayt iletidir. IBM MQ iletisinde `MQRFH2` üstbilgisi yoksa, yalnızca `MQMD` üstbilgisindeki alanlardan türetilen JMS üstbilgi alanları ve özellikleri ayarlanır.

IBM MQ classes for JMS iletilerini IBM MQ iletilerine eşleme hakkında daha fazla bilgi için, IBM MQ ürün belgelerine bakın.

### İlgili başvurular

#### XMS iletisinin parçaları

XMS iletisi, bir üstbilgiden, bir özellik kümesinden ve bir gövden oluşur.

#### XMS iletisinde üstbilgi alanları

To allow an XMS application to exchange messages with a WebSphere JMS application, the header of an XMS message contains the JMS message header fields.

#### XMS iletisinin özellikleri

XMS , üç tür ileti özelliğini destekler: JMS tanımlı özellikler, IBM tanımlı özellikler ve uygulama tarafından tanımlanan özellikler.

#### XMS iletisinin gövdesi

Bir iletinin gövdesinde uygulama verileri bulunur. Ancak, bir iletinin gövdesi olamaz ve yalnızca üstbilgi alanlarını ve özellikleri sıkıştırır.

#### İleti seçicileri

Bir XMS uygulaması, almak istediği iletileri seçmek için ileti seçicileri kullanır.

## ***Reading and writing the message descriptor from a IBM Message Service Client for .NET application***

Bir IBM MQ iletisinin tüm ileti tanımlayıcısı (`MQMD`) alanlarına, `StrucId` ve `Version`; `BackoutCount` dışındaki tüm ileti tanımlayıcısı (`MQMD`) alanlarına erişebilirsiniz, ancak bu alanlara yazılamaz. Bu özellik yalnızca

IBM WebSphere MQ 6.0 ya da sonraki bir kuyruk yöneticisine bağlanılırken kullanılabilir ve daha sonra açıklanan hedef özellikler tarafından denetlendiğinde kullanılabilir.

The message attributes provided by the IBM Message Service Client for .NET facilitates XMS applications to set MQMD fields and also to drive IBM WebSphere MQ applications.

Yayınlama/abone olma ileti alışverişi kullanılırken bazı kısıtlamalar geçerli olur. For example, MQMD fields like MsgID and CorrelId, if set, are ignored.

Bu konuda açıklanan işlev, bir IBM WebSphere MQ 6.0 kuyruk yöneticisine bağlanıyorsanız yayınlama/abone olma ileti alışverişi için kullanılamaz. Ayrıca, **PROVIDERVERSION** özelliği 6olarak ayarlandığında da kullanılamaz.

### **Accessing IBM MQ Message data from a IBM Message Service Client for .NET application**

You can access the complete IBM MQ message data including the MQRFH2 header (if present) and any other IBM MQ headers (if present) within a IBM Message Service Client for .NET application as the body of a JMSBytesMessage.

Bu konuda açıklanan işlev yalnızca bir IBM WebSphere MQ 7.0 ya da daha sonraki bir kuyruk yöneticisine bağlanılırken kullanılabilir ve WebSphere MQ ileti alışverişi sağlayıcısı normal kiptedir.

Hedef nesne özellikleri, XMS uygulamasının tüm IBM MQ iletisine (varsa, MQRFH2 üstbilgisi dahil) JMSBytesMessagegövdesine nasıl erişeceğini belirler.

## **Sorun giderme**

Bu bölümde, IBM Message Service Client for .NETkullanırken sorunları saptamanıza ve bu sorunları anlamanıza yardımcı olacak bilgiler sağlanmaktadır.

Bu bölüm aşağıdaki konuları içerir:

- [“.NET uygulamaları için izleme yapılandırması” sayfa 79](#)
- [“.NET uygulamaları için FFDC yapılandırması” sayfa 83](#)
- [“Sorun gidermeye ilişkin ipuçları” sayfa 83](#)

## **.NET uygulamaları için izleme yapılandırması**

XMS .NET uygulamaları için, izleme olanağını bir uygulama yapılandırma dosyasından ve XMS ortam değişkenlerinden yapılandırabilirsiniz. İzlemek istediğiniz bileşenleri seçebilirsiniz. İzleme genellikle IBM Destek 'in rehberliğinde kullanılır.

XMS .NET için izleme, standart .NET izleme altyapısına dayalıdır.

Hata izlemesi dışında tüm izleme varsayılan olarak geçersiz kılınmaktadır. İzleme ayarlarını aşağıdaki yöntemlerden biriyle izlemeyi ve izleme ayarlarını yapabilirsiniz:

- By using an application configuration file with a name that consists of the name of the executable program to which the file relates, with the suffix `.config`. For example, the application configuration file for `text.exe` would have the name `text.exe.config`. Using an application configuration file is the preferred way of enabling trace for XMS .NET applications. Daha fazla ayrıntı için bkz. [“Bir uygulama yapılandırma dosyasını kullanarak yapılandırmayı izleme” sayfa 80](#).
- XMS ortam değişkenlerini XMS C ya da C++ uygulamaları için kullanarak kullanabilirsiniz. Daha fazla ayrıntı için bkz. [“Trace configuration using XMS environment variables” sayfa 82](#).

Etkin izleme kütüğünün adı `xms_tracePID.log` 'dur. Burada `PID` , uygulamanın işlem tanıtıcısını gösterir. Etkin izleme kütüğünün büyüklüğü varsayılan olarak 20 MB ' den sınırlı olur. Bu sınıra ulaşıldığında, dosya yeniden adlandırılır ve arşivlenir. Arşivlenen dosyalar `xms_tracePID_YY.MM.DD_HH.MM.SS.log`

Varsayılan olarak, saklanan izleme dosyalarının sayısı dört, bir etkin dosya ve üç arşivlenen dosya olur. Bu dört dosya, uygulama duruncaya kadar, en eski dosya en yeni dosya tarafından kaldırılıp değiştirilinceye kadar hareketli arabellek olarak kullanılır. Uygulama yapılandırma dosyasında farklı bir sayı belirterek

izleme dosyalarının sayısını değiştirebilirsiniz. Ancak, en az iki dosya (bir etkin dosya ve bir arşivlenmiş dosya) olmalıdır.

İki izleme kütüğü biçimi kullanılabilir:

- Temel biçim izleme dosyaları, WebSphere Application Server biçiminde, insan tarafından okunabilen bir dosyadır. Bu biçim, varsayılan izleme kütüğü biçimidir. Temel biçim, izleme çözümleyici araçları ile uyumlu değildir.
- Gelişmiş biçim izleme dosyaları, izleme çözümleyici araçları ile uyumludur. İzleme dosyalarını uygulama yapılandırma dosyasında ileri düzey biçimde üretmek istediğinizi belirtmelisiniz.

İzleme girişleri aşağıdaki bilgileri içerir:

- İzlemenin günlüğe kaydedildiği tarih ve saat
- Sınıf adı
- İzleme tipi
- İzleme iletisi

Aşağıdaki örnekte, bazı izlerden bir veri alımı gösterilmiştir:

```
[09/11/2005 14:33:46:914276] 00000004 IBM.XMS.Comms.IoRequest > Allocate Entry
[09/11/2005 14:33:46:914276] 00000004 IBM.XMS.Comms.IoRequest > Initialize Entry
[09/11/2005 14:33:46:914276] 00000004 IBM.XMS.Comms.IoRequest < Initialize Exit
[09/11/2005 14:33:46:914276] 00000004 IBM.XMS.Comms.IoRequest < Allocate Exit
```

Önceki örnekte biçim şöyledir:

```
[Date Time:Microsecs] Thread-id Classname Trace-type Methodname Entry
or Exit
```

Burada Trace - type şudur:

- > Giriş için
- < Çıkış İçin
- Hata ayıklama bilgileri için d

## Bir uygulama yapılandırma dosyasını kullanarak yapılandırmayı izleme

XMS .NET uygulamaları için izleme yapılandırmanın tercih edilen yolu bir uygulama yapılandırma dosyasıdır. Bu dosyanın izleme bölümünde, izlenecek olanları, izleme kütüğü yerini ve izin verilen büyüklük üst sınırını, kullanılan izleme dosyalarının sayısını ve izleme kütüğü biçimini tanımlayan parametreleri içerir.

Uygulama yapılandırma dosyasını kullanarak izlemeyi açmak için, dosyayı uygulamanıza ilişkin yürütülür dosya ile aynı dizine yerleştirmeniz yeterlidir.

İzleme, bileşen ve izleme tipine göre etkinleştirilebilir. Bir izleme grubunun tamamı için izlemeyi de açmak da olanaklıdır. Bir sıradüzendeki bileşenler için izlemeyi tek tek ya da toplu olarak çevirebilirsiniz. Kullanılabilir izleme tipleri şunlardır:

- Hata ayıklama izleme
- Kural Dışı Durum İzlemesi
- Uyarılar, bilgi iletileri ve hata iletileri
- Yöntem girişi ve çıkış izleme

Aşağıdaki örnekte, bir uygulama yapılandırma dosyasının İzleme bölümünde tanımlanan izleme ayarları gösterilmektedir:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
```



```

<sectionGroup name="IBM.XMS">
  <section name="Trace"
    type="System.Configuration.SingleTagSectionHandler"/>
</sectionGroup>
</configSections>

<IBM.XMS>
  <Trace traceSpecification="*=all=enabled" traceFilePath=""
    traceFileSize="20000000" traceFileNumber="3"
    traceFormat="advanced"/>
</IBM.XMS>
</configuration>

```

Çizelge 22 sayfa 81 , parametre ayarlarını daha ayrıntılı olarak açıklar.

Çizelge 22. Uygulama yapılandırma dosyası izleme parametresi ayarları	
Değiştirge	Tanım
traceSpecification= <i>ComponentName</i> = <i>type</i> = <i>state</i>	<p><i>ComponentName</i> , izlenmesini istediğiniz sınıfın adıdır. Bu adda bir * genel arama karakteri kullanabilirsiniz. Örneğin, *=all=enabled tüm sınıfları izlemek istediğinizi belirtmiş ve IBM.XMS.impl.*=all=enabled yalnızca API izlemesi gerektirdiğini belirtir.</p> <p><i>type</i> aşağıdaki izleme tiplerinden herhangi biri olabilir:</p> <ul style="list-style-type: none"> <li>• tümü</li> <li>• hata ayıklama</li> <li>• olay</li> <li>• EntryExit</li> </ul> <p><i>state</i> etkinleştirilebilir ya da devre dışı bırakılabilir.</p> <p>Birden çok izleme ögesini bir ':' (iki nokta) sınırlayıcısını kullanarak birlikte dize getirebilirsiniz.</p>
traceFilePath=" <i>filename</i> "	<p>Bir traceFile yolu belirtmezseniz ya da traceFileYolu varsa, ancak boş bir dizgi içeriyorsa, izleme dosyası yürürlükteki dizine yerleştirilir. İzleme dosyasını adlandırılmış bir dizinde saklamak için, traceFileYolunda dizin adını belirtin; örneğin:</p> <pre>traceFilePath="c:\somepath"</pre>
traceFileSize=" <i>size</i> "	<p>İzleme dosyası için izin verilen büyüklük üst sınırı. Bir dosya bu boyuta eriştiğinde arşivlenir ve yeniden adlandırılır. Varsayılan değer üst sınırı 20 KB 'dir ( traceFileSize="20000000" olarak belirtilir).</p>
traceFileNumber=" <i>number</i> "	<p>Alıkonacağı izleme dosyalarının sayısı. Varsayılan değer 4 'tür (bir etkin dosya ve 3 arşiv dosyası). İzin verilen alt sınır 2 'dir.</p>
traceFormat=" <i>format</i> "	<p>Varsayılan izleme biçimi temel biçimdir. Trace files are produced in this format if you specify traceFormat="basic", or if you do not specify a traceFormat, or if the traceFormat is present but contains an empty string.</p> <p>İzlemenin çözümleyici araçlarıyla uyumlu bir izleme gerektiriyorsa, traceFormat="advanced" belirtmeniz gerekir.</p>

Uygulama yapılandırma dosyasındaki izleme ayarları dinamiktir ve dosya her kaydedildiğinde ya da değiştirildiğinde yeniden yönlendirilir. Dosya düzenlendikten sonra, hatalar dosyada bulunursa, izleme dosyası ayarları varsayılan değerlerine geri döndürülür.

### İlgili kavramlar

#### Trace configuration using XMS environment variables

As an alternative to using an application configuration file, you can turn on trace using XMS environment variables. Bu ortam değişkenleri yalnızca, uygulama yapılandırma dosyasında izleme belirtimi yoksa kullanılır.

### Trace configuration using XMS environment variables

As an alternative to using an application configuration file, you can turn on trace using XMS environment variables. Bu ortam değişkenleri yalnızca, uygulama yapılandırma dosyasında izleme belirtimi yoksa kullanılır.

Bir XMS .NET uygulamasına ilişkin izlemeyi yapılandırmak için, uygulamayı çalıştırmadan önce aşağıdaki ortam değişkenlerini ayarlayın:

Ortam değişkenleri	Varsayılan	Ayarlar	Anlamı
XMS_TRACE_ON	Burada geçerli değil	Geçerli değil: Bu değişkenin değeri yok sayılır	XMS_TRACE_ON değeri ayarlandıysa, varsayılan olarak tüm izleme etkinleştirilir.
XMS_TRACE_FILE_PATH	Geçerli çalışma dizini	/dirpath/	İzleme ve FFDC kayıtlarının yazıldığı dizin yolu. XMS , başka bir yer belirtmediğiniz sürece, yürürlükteki çalışma dizininde FFDC ve izleme dosyaları yaratır. XMS_TRACE_FILE_PATH ortam değişkenini, XMS ' un FFDC ve izleme dosyalarını yaratmasını istediğiniz dizinin tam olarak nitelenmiş yol adı olarak ayarlayarak, başka bir yer belirtebilirsiniz. İzlemek istediğiniz uygulamayı başlatmadan önce bu ortam değişkenini ayarlamanız gerekir. Uygulamanın çalıştığı kullanıcı kimliğinin,directoryadlı dizine yazma yetkisine sahip olduğundan emin olmalısınız. Burada XMS , FFDC ve izleme dosyalarını yaratır.

Çizelge 23. .NET izleme için ortam değişkeni ayarları (devamı var)			
Ortam değişkenleri	Varsayılan	Ayarlar	Anlamı
XMS_TRACE_BIÇIMI	Temel Düzey	TEMEL, ILERI	BASIC ya da ADVANCED olabilen gerekli izleme biçimini belirtir. Varsayılan biçim BASIC 'dir. ADVANCED biçimi, izleme çözümleyici araçlarıyla uyumludur.
XMS_IZLEME_BELIRTIMI	Burada geçerli değil	Bkz. “Bir uygulama yapılandırma dosyasını kullanarak yapılandırmayı izleme” sayfa 80	“Bir uygulama yapılandırma dosyasını kullanarak yapılandırmayı izleme” sayfa 80’inde belirtilen biçimi izleyen izleme belirtimini geçersiz kılar.

### İlgili kavramlar

Bir uygulama yapılandırma dosyasını kullanarak yapılandırmayı izleme

XMS .NET uygulamaları için izleme yapılandırmanın tercih edilen yolu bir uygulama yapılandırma dosyasıdır. Bu dosyanın izleme bölümünde, izlenecek olanları, izleme kütüğü yerini ve izin verilen büyüklük üst sınırını, kullanılan izleme dosyalarının sayısını ve izleme kütüğü biçimini tanımlayan parametreleri içerir.

## .NET uygulamaları için FFDC yapılandırması

XMS' in .NET uygulaması için, her FFDC için bir FFDC dosyası üretilir.

First Failure Data Capture (FFDC) dosyaları, insan tarafından okunabilir metin dosyalarında depolanır. These files have names of the form `xmsffdcprocessID_DateTimestamp.txt`. Bir dosya adı örneği: `xmsffdc264_2006.01.06T13.18.52.990955.txt`. Zaman damgası mikrosaniye çözünürlüğünü içerir.

Dosyalar, kural dışı durumun ortaya çıktığı tarih ve saat ile başlar ve ardından kural dışı durum tipi izler. Dosyalar, bu FFDC ' nin oluştuğu yeri bulmak için kullanılabilen benzersiz bir kısa probeId içerir.

FFDC ' yi açmak için herhangi bir yapılandırma gerçekleştirmenize gerek yoktur. Varsayılan değer olarak, tüm FFDC dosyaları yürürlükteki dizine yazılır. Ancak, gerekirse, uygulama yapılandırma dosyasının İzleme bölümünde `ffdcDirectory` değerini değiştirerek farklı bir dizin belirtebilirsiniz. Aşağıdaki örnekte, tüm izleme dosyaları `c:\client\ffdc` dizininde günlüğe kaydedilir:

```
<IBM.XMS>
  <Trace ffdc=true ffdcDirectory="c:\client\ffdc"/>
</IBM.XMS>
```

Uygulama yapılandırma dosyasının İzleme kısmında FFDC ' yi false değerine ayarlayarak izlemeyi geçersiz kılabilirsiniz.

Uygulama yapılandırma kütüğü kullanmıyorsanız, FFDC açık ve izleme kapalıdır.

## Sorun gidermeye ilişkin ipuçları

XMSkullanımıyla ilgili sorunları gidermenize yardımcı olması için bu ipuçlarını kullanın.

## Bir XMS uygulaması kuyruk yöneticisine bağlanamıyor (MQRC\_NOT\_AUTONIZED)

XMS .NET istemcisinde, IBM MQ JMS istemcisinin davranışından farklı davranışlar olabilir. Therefore, you may find that your XMS application cannot connect to your queue manager, although your JMS application can.

- Bu soruna basit bir çözüm, 12 karakterden uzun olmayan ve kuyruk yöneticisinin yetki listesinde yer alan en fazla 12 karakterden oluşan bir kullanıcı kimliği kullanmayı denemeyi denemesidir. eğer bu çözüm ideal değilse, farklı ama daha karmaşık bir yaklaşım güvenlik çıkışlarını kullanmak olacaktır. Bu sorunla ilgili daha fazla yardıma gereksinim duyarsanız, yardım almak için IBM Destek bölümünü arayın.
- Bağlantı üreticisinin XMLC\_USERID özelliğini ayarladıysanız, bu, oturum açan kullanıcının kullanıcı kimliği ve parolasıyla eşleşmelidir. Bu özelliği ayarlamadıysanız, kuyruk yöneticisi varsayılan olarak oturum açmış olan kullanıcının kullanıcı kimliğini kullanır.
- IBM MQ için kullanıcı kimlik doğrulaması, şu anda oturum açmış olan kullanıcının ayrıntıları ve XMSC.USERID ve XMSC.PASSWORD alanları. Bu, IBM MQ ile tutarlılık sağlamak için tasarlanmıştır. Kimlik doğrulamaya ilişkin ek bilgi için çevrimiçi IBM MQ ürün belgelerindeki *Kimlik Doğrulama Bilgileri* başlıklı konuya bakın.

## Bağlantı, ileti alışverişi altyapısına yeniden yönlendirdi

Bir WebSphere Application Server 6.0.2 hizmet bütünleştirme veriyoluna bağlandığında, tüm bağlantılar özgün sağlayıcı uç noktasından, veriyolunun o istemci bağlantısı için seçtiği ileti alışverişi altyapısına yönlendirilebilir. Bunu yaparken, bağlantıyı her zaman bir IP adresi yerine anasistem adı tarafından belirlenen bir anasistem sunucusuna yönlendirecektir. Bu nedenle, anasistem adı çözülemeyecekse, bağlantı sorunları yaşayabilirsiniz.

WebSphere Application Server 6.0.2 hizmet tümleştirme veriyoluna başarıyla bağlanmak için, istemci anasistem makinenizdeki anasistem adları ve IP adresleri arasında bir eşleme sağlamanız gerekebilir. Örneğin, eşlemeyi istemci ana makinenizdeki yerel bir anasistem çizelgesinde belirleyebilirsiniz.

## Telnet benzeri parola kimlik doğrulaması desteği

XMS .NET Real Time Transport iletişim kuralı yalnızca basit telnet benzeri parola doğrulamasını destekler. XMS .NET Real Time Transport iletişim kuralı, Koruma Kalitesi 'ni desteklemiyor.

## Özellik tipi çift için değerler ayarlanıyor

Bir Windows 64 bit altyapısında, değerler Double.Epsilondeğerinden küçükse, özellik tipi çift için değer belirlenirken ya da alınırken SetDoubleProperty () ya da GetDoubleözelligi () yöntemleri doğru çalışmayabilir.

Örneğin, çift tipi çift olan bir özellik için 4.9E-324 değerini ayarlamaya çalışırsanız, Windows 64 bit altyapıları bunu 0.0olarak değerlendirir. So, in a distributed messaging environment, if a JMS or another application sets the value for a double property as 4.9E-324 on any UNIX or Windows 32-bit machine, and XMS .NET runs on a 64-bit machine, the value returned by GetDoubleProperty() is 0.0. This is a known issue with Microsoft .NET Framework 2.0 Framework.

## Çalıştırma zamanında işlenebilecek hata koşulları

API çağrılarında dönen dönüş kodları, çalıştırma zamanında işlenebilecek hata koşullarıdır. Bu tür bir hata ile başa çıkabilmenin yolu, C ya da C++ API ' yi kullanıp kullanmamanıza bağlıdır.

## Çalıştırma zamanında hataların saptanması

Bir uygulama bir C API işlevini çağırıyorsa ve arama başarısız olursa, XMS\_OK dışında bir dönüş kodu içeren bir yanıt, başarısızlığın nedenine ilişkin daha fazla bilgi içeren bir XMS hata bloku ile geri döndürülür.

C++ API, bir yöntem kullanıldığında kural dışı durum yayınlar.

Bir uygulama, bağlantıyla ilgili bir sorunun zamanuyumsuz olarak bildirilmesi için bir kural dışı durum dinleyicisi kullanır. Kural dışı durum dinleyicisi, XMS C ya da C++ API kullanılarak kullanıma hazırlanır ve kullanıma hazırlanır.

## Çalıştırma zamanında hataların nasıl ele işleneceği

Bazı hata koşulları, bazı kaynakların kullanılmadığını gösterir ve bir uygulamanın alabileceği işlem, uygulamanın çağırdığı XMS işlevine bağlıdır. Örneğin, sunucuyla bağlantı kurulamazsa, uygulama, bağlantı yapıncaya kadar düzenli olarak yeniden deneme yapmak isteyebilirsiniz. Bir XMS hata öbeği ya da kural dışı durumu, hangi işlemin yapılması gerektiğini saptamak için yeterli bilgi içermeyebilir ve bu durumlarda, genellikle daha özel tanımlama bilgileri içeren bağlantılı bir hata öbeği ya da kural dışı durumu vardır.

C API 'de, XMS\_OK dışında bir dönüş kodu içeren bir yanıt için her zaman test edin ve API çağrısına her zaman bir hata öbeği geçirin. İşlem genellikle hangi API işlevinin uygulama tarafından uygulansa bağlı olur.

C++ API 'de, bir try öbeğindeki yöntemlere her zaman çağrılar ekleyin ve XMS kural dışı durumunun tüm tiplerini yakalamak için, catch yapısındaki Kural Dışı Durum sınıfını belirtin.

Kural dışı durum dinleyicisi, herhangi bir zamanda başlatılabilen zamanuyumsuz bir hata durumu yoludur. Kural dışı durum dinleyicisi işlevi başlatıldığında, kendi iş parçasığında genellikle normal bir XMS API hata koşulundan daha ciddi bir hata ortaya konması durumunda olur. Any appropriate action may be taken, but you must be careful to follow the rules for the XMS threading model as described in [“Threading modeli”](#) sayfa 20.

### İlgili kavramlar

[Threading modeli](#)

Genel kurallar, çok iş parçacıklı bir uygulamanın XMS nesnelere nasıl kullanılabileceğini yönetir.

## Message Service Clients for .NET başvurusu

Bu başvuru kısmı, Message Service Client for .NET olanağını kullanmanıza yardımcı olacak bilgileri içerir. Bu bilgiler, XMS ile programlama içinde yer alan görevleri gerçekleştirmenize yardımcı olur.

## .NET Arabirimler

Bu bölüm , .NET sınıfı arabirimlerini ve bunların özelliklerini ve yöntemlerini içerir.

Aşağıdaki tablo, IBM .XMS ad alanı içinde tanımlanan tüm arabirimleri özetlemektedir.

Arabirim	Tanım
<a href="#">“IBytesMessage” sayfa 88</a>	Bir bayt iletisi, gövdesi byte akışından oluşan bir iletidir.
<a href="#">“IConnection” sayfa 98</a>	Bağlantı nesnesi, uygulamanın ileti sunucusu için etkin bağlantısını gösterir.
<a href="#">“IConnectionFactory” sayfa 101</a>	Bir uygulama, bağlantı oluşturmak için bir bağlantı üreticisi kullanır.
<a href="#">“IConnectionMetaVerileri” sayfa 103</a>	Bir ConnectionMetaVeri nesnesi, bir bağlantıyla ilgili bilgi sağlar.
<a href="#">“Varış Noktası” sayfa 103</a>	Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.
<a href="#">“ExceptionHandler” sayfa 105</a>	Bir uygulama, bağlantıyla ilgili bir sorunun zamanuyumsuz olarak bildirilmesi için bir kural dışı durum dinleyicisi kullanır.

Çizelge 24. .NET sınıf arabirimlerinin özeti (devamı var)

Arabirim	Tanım
<a href="#">“IllegalStateKural Dışı Durumu” sayfa 105</a>	XMS throws this exception if an application calls a method at an incorrect or inappropriate time, or if XMS is not in an appropriate state for the requested operation.
<a href="#">“InitialContext” sayfa 106</a>	Bir uygulama, denetlenen nesnelere havuzundan alınan nesne tanımlamalarından nesne yaratmak için InitialContext nesnesini kullanır.
<a href="#">“InvalidClientIDException” sayfa 108</a>	XMS , bir uygulama bir bağlantı için istemci tanıtıcısı belirlemeye çalışırsa, ancak istemci tanıtıcısı geçerli değilse ya da önceden kullanımda olduğu için bu kural dışı durumu atar.
<a href="#">“InvalidDestinationKural Dışı Durumu” sayfa 108</a>	Bir uygulama geçerli olmayan bir hedef belirtiyorsa,XMS bu kural dışı durumu atar.
<a href="#">“InvalidSelectorKural Dışı Durumu” sayfa 109</a>	Bir uygulama, sözdizimi geçerli olmayan bir ileti seçici ifadesi sağlıyorsa,XMS bu kural dışı durumu verir.
<a href="#">“IMapMessage” sayfa 109</a>	Eşlem iletisi, her değer ilişkili bir veri tipine sahip olduğu bir ad-değer çiftleri kümesinden oluşan bir iletidir.
<a href="#">“İleti” sayfa 118</a>	Bir ileti nesnesi, uygulamanın gönderdiği ya da aldığı bir iletiyi gösterir. IMessage is a superclass for the message classes such as IMapMessage.
<a href="#">“IMessageConsumer” sayfa 124</a>	Bir uygulama, bir hedefe gönderilen iletileri almak için bir ileti tüketicisini kullanır.
<a href="#">“MessageEOFException” sayfa 127</a>	XMS , bir uygulama byte iletilerinin gövdesini okurken bir byte ileti akışının sonuna rastlanırsa,XMS bu kural dışı durumu döndürür.
<a href="#">“MessageFormatKural Dışı Durumu” sayfa 127</a>	XMS throws this exception if XMS encounters a message with a format that is not valid.
<a href="#">“IMessageListener (temsilci)” sayfa 128</a>	Bir uygulama, iletileri zamanuyumsuz olarak almak için bir ileti dinleyicisini kullanır.
<a href="#">“MessageNotReadableException” sayfa 128</a>	XMS , bir uygulama yalnızca yazma işlemi olan bir iletinin gövdesini okumayı denediğinde bu kural dışı durumu atar.
<a href="#">“MessageNotWritableException” sayfa 128</a>	XMS , bir uygulama salt okunur bir iletinin gövdesine yazma girişiminde bulunursa bu kural dışı durumu atar.
<a href="#">“IMessageProducer” sayfa 129</a>	Bir uygulama, bir hedefe ileti göndermek için ileti üreticisini kullanır.
<a href="#">“IObjectMessage” sayfa 134</a>	Bir nesne iletisi, gövdesi serileştirilmiş bir Java ya da .NET nesnesinden farklı olan bir iletidir.
<a href="#">“IPropertyContext” sayfa 135</a>	IPropertyContext , özellikleri alan ve ayarlayan yöntemler içeren, soyut bir üst sınıftır. Bu yöntemler diğer sınıflar tarafından devralınır.

Çizelge 24. .NET sınıf arabirimlerinin özeti (devamı var)

Arabirim	Tanım
<a href="#">“IQueueBrowser” sayfa 144</a>	Bir uygulama, kuyruğu kaldırmadan kuyruklardaki iletilere göz atmak için bir kuyruk tarayıcısı kullanır.
<a href="#">“İstekte Bulunan” sayfa 146</a>	Bir uygulama, istek iletisini göndermek için istekte bulunan bir programı kullanır ve ardından yanıtı bekler ve alır.
<a href="#">“ResourceAllocationÖzel Durumu” sayfa 148</a>	XMS throws this exception if XMS cannot allocate the resources required by a method.
<a href="#">“SecurityException” sayfa 148</a>	XMS throws this exception if the user identifier and password provided to authenticate an application are rejected. XMS , bir yetki denetimi başarısızlıkla sonuçlanırsa ve bir yöntemin tamamlanmasını önlediğinde bu kural dışı durumu da atar.
<a href="#">“ISession” sayfa 148</a>	Oturum, ileti göndermek ve almak için tek bir iş parçacıklı bağlamdır.
<a href="#">“IStreamMessage” sayfa 159</a>	Akış iletisi, her değerin ilişkili bir veri tipine sahip olduğu bir değerler akışından oluşan bir iletidir.
<a href="#">“ITextMessage” sayfa 168</a>	Metin iletisi, gövdesi bir dizeyi oluşturan bir iletidir.
<a href="#">“TransactionInProgressException” sayfa 169</a>	XMS , bir işlem devam etmekte olduğu için geçerli olmayan bir işlem isteğinde bulunduysa, bu kural dışı durumu döndürür.
<a href="#">“TransactionRolledBackException” sayfa 170</a>	XMS , bir uygulama yürürlükteki hareketi kesinleştirmek için Session.commit() uygulaması çağırıyorsa, ancak işlem geriye işlenirse bu kural dışı durumu atar.
XMSC	.NET için, XMS özellik adları ve değerleri, bu sınıfta genel sabitler olarak tanımlanır. Daha fazla ayrıntı için bkz. “XMS nesnelerinin özellikleri” sayfa 172.
<a href="#">“XMSException” sayfa 170</a>	XMS bir .NET yöntemi çağırısını işlerken bir hata saptarsa, XMS kural dışı durum yayınlar. Kural dışı durum, hatayla ilgili bilgileri sarmalayan bir nesnedir.  Farklı tiplerde XMS kural dışı durumu vardır ve bir XMSException nesnesi yalnızca bir kural dışı durum tipidir. Ancak, XMSException sınıfı, diğer XMS kural dışı durum sınıflarının üst sınıfıdır. XMS , diğer kural dışı durum tiplerinden hiçbirinin uygun olmadığı durumlarda bir XMSException nesnesi atar.
<a href="#">“XMSFactoryFactory” sayfa 171</a>	Bir uygulama yönetilen nesnelere kullanmıyorsa, bağlantı üreticileri, kuyruklar ve konular yaratmak için bu sınıfı kullanın.

Her yöntemin tanımı, yöntem çağırısı işlerken bir hata algılandığında XMS ' un döndürebileceği kural dışı durum kodlarını listeler. Her bir kural dışı durum kodu, karşılık gelen bir kural dışı durum içeren adlandırılmış değişimle temsil edilir.

### İlgili kavramlar

[Kendi uygulamalarınızı oluşturma](#)

Örnek uygulamaları oluştururken kendi uygulamalarınızı oluşturun.

#### XMS uygulamaları yazılıyor

Bu bölümdeki konular, XMS uygulamalarını yazarken size yardımcı olacak bilgiler sağlar.

#### Writing XMS .NET applications

Bu bölümdeki konular, XMS .NET uygulamalarını yazarken size yardımcı olacak bilgiler sağlar.

### **İlgili başvurular**

#### XMS nesnelerinin özellikleri

Bu bölüm , XMStarafından tanımlanan nesne özelliklerini belgelerdir.

## **IBytesMessage**

Bir bayt iletisi, gövdesi byte akışından oluşan bir iletidir.

### **Devralma sıradüzeni:**

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IBytesMessage
```

### **İlgili başvurular**

#### Byte iletileri

Bir baytlar iletisinin gövdesi, bayt akışını içerir. Gövde yalnızca gerçek verileri içerir ve bu verileri yorumlamak için gönderme ve alma uygulamalarının sorumluluğunda olur.

### **.NET özellikler**

*BodyLength -Body Length (Gövde Uzunluğu)*

### **Arabirim:**

```
Int64 BodyLength
{
    get;
}
```

İletinin gövdesi salt okunurken, iletinin gövdesinin uzunluğunu bayt cinsinden alın.

Döndürülen değer, iletiyi okumak için imlecin bulunduğu yere bakmaksızın, tüm gövdenin uzunluğuna sahip olur.

### **Özel durumlar:**

- XMSException
- MessageNotReadableException

## **Yöntemler**

*ReadBoolean -Boolean değerini okuyun*

### **Arabirim:**

```
Boolean ReadBoolean();
```

Bayt ileti akışından bir Boole değeri okuyun.



**Parametreler:**

Yok

**Döndürülen:**

Okunan boole deęeri.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadSignedByte-Okuma Byte 'ı*

**Arabirim:**

```
Int16 ReadSignedByte();
```

Bayt ileti akışından, imzalı 8 bitlik bir tamsayı olarak bir sonraki baytı okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan bayt.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadBytes -Okuma Baytları*

**Arabirim:**

```
Int32 ReadBytes(Byte[] array);  
Int32 ReadBytes(Byte[] array, Int32 length);
```

İmlecin yürürlükteki konumundan başlayarak, byte ileti akımından oluşan bayt dizisini okuyun.

**Parametreler:****dizi (çıkış)**

Okunan bayt dizisine sahip arabellek. Çaęrı, arabelleęin uzunluęundan büyük ya da bu uzunluęa eşit olmadan önce akıştan okunacaksa kalan bayt sayısı doldurulur. Ters durumda, arabellek kalan tüm baytlarla kısmen doldurulur.

Girişte boş deęerli bir gösterge belirtirseniz, yöntem bunları okumadan baytların üzerine atlar. Çaęrı, arabelleęin uzunluęundan büyük ya da bu uzunluęa eşit olmadan önce akıştan okunacaksa kalan bayt sayısı, atlanan bayt sayısı arabelleęin uzunluęuna eşit olur. Ters durumda, kalan tüm byte 'lar atlanır. İmleç, bayt ileti akımında okunmak üzere bir sonraki konuma kalır.

**uzunluk (giriş)**

Arabelleęin bayt cinsinden uzunluęu

**Döndürülen:**

Arabelleęe okunan bayt sayısı. Arabellek kısmen doldurulduysa, okunacağı başka bayt kalmadığını belirten deęer, arabellek uzunluęundan az olur. Aramadan önce akıştan okunmak üzere kalan herhangi bir byte yoksa, deęer XMSC\_END\_OF\_STREAM olur.

Giriş sırasında boş değerli bir gösterge belirtirseniz, yöntem değer döndürmez.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException

*ReadChar -Okuma karakteri*

**Arabirim:**

```
Char ReadChar();
```

Bayt ileti akışından sonraki 2 baytı bir karakter olarak okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan karakter.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadDouble -Çift Duyarlıklı Yüzer Nokta Numarasını Oku*

**Arabirim:**

```
Double ReadDouble();
```

Çift duyarlıklı kayan noktalı sayı olarak bayt ileti akışından sonraki 8 baytı okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan çift duyarlıklı kayan noktalı sayı.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadFloat -Kayan Nokta Numarasını okuyun*

**Arabirim:**

```
Single ReadFloat();
```

Bir kayar noktalı sayı olarak, bayt ileti akışından sonraki 4 bayta ilişkin bilgileri okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan kayan noktalı sayı.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadInt -Tamsayı Oku***Arabirim:**

```
Int32 ReadInt();
```

Bir sonraki 4 baytı, byte ileti akışından imzalı 32 bitlik bir tamsayı olarak okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan tamsayı.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadLong -Uzun Tamsayı Oku***Arabirim:**

```
Int64 ReadLong();
```

Bir sonraki 8 baytı, byte ileti akışından imzalı 64 bitlik bir tamsayı olarak okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan uzun tamsayı.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadShort -Kısa Tamsayı Oku***Arabirim:**

```
Int16 ReadShort();
```

Bayt ileti akışından sonraki 2 baytı, imzalı 16 bitlik bir tamsayı olarak okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan kısa tamsayı.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadByte -İmzalanmamış Byte 'ı Oku*

**Arabirim:**

```
Byte ReadByte();
```

Bayt ileti akışından imzalanmamış 8 bitlik bir tamsayı olarak bir sonraki baytı okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan bayt.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadUnsignedKısa-İmzalanmamış Kısa Tamsayı*

**Arabirim:**

```
Int32 ReadUnsignedShort();
```

İmzalanmamış 16 bitlik bir tamsayı olarak bayt ileti akışından sonraki 2 baytı okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunmamış işaretsiz kısa tamsayı.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadUTF -UTF Dizesini Oku*

**Arabirim:**

```
String ReadUTF();
```

Bayt ileti akışından UTF-8 içinde kodlanan bir dizgi okuyun.

**Not:** ReadUTF() çağrılmadan önce, arabelleğin imlecinin byte ileti akışının başlangıcına işaret ettiğini doğrulayın.

**Parametreler:**

Yok

**Döndürülen:**

Okunan dizeyi sarmalayan bir String nesnesi.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*Sıfırla-İlk Duruma Getir*

**Arabirim:**

```
void Reset();
```

İletinin gövdeini salt okunur kipine getirin ve imleci, byte ileti akışının başlangıcındaki yerini yeniden konumlayın.

**Parametreler:**

Yok

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotReadableException

*WriteBoolean -Yazma Boole Değeri*

**Arabirim:**

```
void WriteBoolean(Boolean value);
```

Bayt ileti akışına bir boole değeri yazın.

**Parametreler:**

**değer (giriş)**

Yazılacak boole değeri.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

## *WriteByte -Yazma Byte 'ı*

### **Arabirim:**

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

Bayt ileti akışına bir bayt yazın.

### **Parametreler:**

#### **değer (giriş)**

Yazılacak bayt.

### **Döndürülen:**

Hükümsüz

### **Özel durumlar:**

- XMSEException
- MessageNotWritableException

## *WriteBytes -Yazma Baytları*

### **Arabirim:**

```
void WriteBytes(Byte[] value);
```

Bayt ileti akışlarına bir bayt dizisi yazın.

### **Parametreler:**

#### **değer (giriş)**

Yazılacak bayt dizisi.

### **Döndürülen:**

Hükümsüz

### **Özel durumlar:**

- XMSEException
- MessageNotWritableException

## *WriteBytes -Yazma Kısmi Byte Dizisi*

### **Arabirim:**

```
void WriteBytes(Byte[] value, int offset, int length);
```

Belirtilen uzunluğun tanımladığı şekilde, byte ileti akışına kısmi byte dizisi yazın.

### **Parametreler:**

#### **değer (giriş)**

Yazılacak bayt dizisi.

#### **görelî konum (giriş)**

Yazılacak bayt dizisine ilişkin başlangıç noktası.

#### **uzunluk (giriş)**

Yazılacak bayt sayısı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteChar -Yazma Karakteri*

**Arabirim:**

```
void WriteChar(Char value);
```

Önce, byte ileti akışına 2 byte, önce yüksek sıralı bayt olarak bir karakter yazın.

**Parametreler:****değer (giriş)**

Yazılacak karakter.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteDouble -Çift Duyarlıklı Kayan Nokta Numarası Yaz*

**Arabirim:**

```
void WriteDouble(Double value);
```

Çift duyarlıklı kayan noktalı sayıyı uzun bir tamsayıya dönüştür ve uzun tamsayıyı, önce yüksek sıralı bayt olan 8 bayt olarak byte ileti akımuna yazın.

**Parametreler:****değer (giriş)**

Yazılacak çift duyarlıklı kayan noktalı sayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteFloat -Kayan Noktalı Sayı Yazma*

**Arabirim:**

```
void WriteFloat(Single value);
```

Bir kayan noktalı sayıyı bir tamsayıya dönüştürüp tamsayıyı, önce yüksek sıralı bayt olan 4 bayt olarak byte ileti akışına yazın.

**Parametreler:****değer (giriş)**

Yazılacak kayan noktalı sayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteInt -Yazma Tamsayı*

**Arabirim:**

```
void WriteInt(Int32 value);
```

Bayt ileti akışına 4 byte, önce yüksek sıralı bayt olarak bir tamsayı yazın.

**Parametreler:****değer (giriş)**

Yazılacak tamsayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteLong -Yazma Uzun Tamsayı*

**Arabirim:**

```
void WriteLong(Int64 value);
```

Bayt ileti akışına önce 8 bayt, yüksek sıralı bayt olarak uzun bir tamsayı yazın.

**Parametreler:****değer (giriş)**

Yazılacak uzun tamsayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteObject -Yazma Nesnesi*

**Arabirim:**

```
void WriteObject(Object value);
```



Belirtilen nesneyi byte ileti akışa yazın.

**Parametreler:**

**değer (giriş)**

Yazılacak nesne; bir temel tipe gönderme olması gerekir.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteShort -Yazma Kısa Tamsayı*

**Arabirim:**

```
void WriteShort(Int16 value);
```

Bayt ileti akışına kısa bir tamsayı yazın; 2 byte olarak, önce yüksek sıralı byte 'ı yazın.

**Parametreler:**

**değer (giriş)**

Yazılacak kısa tamsayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteUTF -UTF dizgisi yazın*

**Arabirim:**

```
void WriteUTF(String value);
```

Write a string, encoded in UTF-8, to the bytes message stream.

**Parametreler:**

**değer (giriş)**

Yazılacak dizeyi sarsalayan bir String nesnesi.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

***Edinilmiş özellikler ve yöntemler***

Aşağıdaki özellikler [IMessage](#) arabiriminden devralınır:

[JMSCorrelationID](#), [JMSDeliveryMode](#), [JMSDestination](#), [JMSExpiration](#), [JMSMessageID](#), [JMSPriority](#), [JMSRevered](#), [JMSReplyTo](#), [JMSTimestapp](#), [JMSType](#), [Properties](#)

Aşağıdaki yöntemler IMessage arabiriminden devralınır:

clearBody, clearProperties, PropertyExists

Aşağıdaki yöntemler IPropertyContext arabiriminden devralınır:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortÖzellik, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongÖzelliği, SetObjectProperty, SetShortProperty, SetStringProperty

## IConnection

Bağlantı nesnesi, uygulamanın ileti sunucusu için etkin bağlantısını gösterir.

### Devralma sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IConnection
```

Connection nesnesine ilişkin XMS tanımlı özelliklerinin bir listesi için bkz. "Bağlantı Özellikleri" sayfa 173.

## .NET özellikler

*ClientID -İstemci tanıtıcısını al ve ayarla*

### Arabirim:

```
String ClientID
{
    get;
    set;
}
```

Bağlantıya ilişkin istemci tanıtıcısını alın ve ayarlayın.

İstemci tanıtıcısı, ConnectionFactorydosyası içinde yönetici tarafından önceden yapılandırılabilir ya da ClientIDayarına göre atanmış olabilir.

İstemci tanıtıcısı yalnızca, yayınlama/abone olma etki alanındaki sürekli abonelikleri desteklemek için kullanılır ve noktadan noktaya iletişim alanında yoksayılr.

Bir uygulama, bir bağlantı için istemci tanıtıcısı ayarlarsa, bağlantı yaratıldıktan hemen sonra ve bağlantıda başka bir işlem gerçekleştirmeden önce uygulamanın yapması gerekir. Uygulama bu noktadan sonra bir istemci tanıtıcısı ayarlamaya çalışırsa, çağrı kural dışı durum IllegalStatekural dışı durumunu atar.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

### Özel durumlar:

- XMSException
- IllegalStateKural Dışı Durumu
- InvalidClientIDException

*ExceptionListener -Kural Dışı Durum Dinleyiciyi Al ve Ayarla*

### Arabirim:

```
ExceptionListener ExceptionListener
{
    get;
```

```
    set;  
}
```

Bağlantıda kayıtlı olan kural dışı durum dinleyicisini alın ve bağlantıyla bir kural dışı durum dinleyicisi kaydettirin.

Bağlantıda bir kural dışı durum dinleyicisi kaydedilmediyse, yöntem boş değer döndürür. Bağlantıda bir kural dışı durum dinleyicisi zaten kayıtlı ise, kural dışı durum dinleyicisi yerine boş değer belirterek kayıt işlemini iptal edebilirsiniz.

Kural dışı durum dinleyicilerinin kullanılmasına ilişkin ek bilgi edinmek için [“Message and exception listeners in .NET” sayfa 46’ e bakın.](#)

#### **Özel durumlar:**

- XMSEException

#### *Meta Veri-Meta Verileri Al*

#### **Arabirim:**

```
IConnectionMetaData MetaData  
{  
    get;  
}
```

Bağlantıya ilişkin meta verileri alın.

#### **Özel durumlar:**

- XMSEException

## **Yöntemler**

#### *Bağlantıyı Kapat-Bağlantıyı Kapat*

#### **Arabirim:**

```
void Close();
```

Bağlantıyı kapatın.

Bir uygulama önceden kapatılmış bir bağlantıyı kapatmaya çalışırsa, arama yoksayılr.

#### **Parametreler:**

Yok

#### **Döndürülen:**

Hükümsüz

#### **Özel durumlar:**

- XMSEException

#### *CreateSession -Oturum Yarat*

#### **Arabirim:**

```
ISession CreateSession(Boolean transacted,  
                        AcknowledgeMode acknowledgeMode);
```

Bir oturum oluşturun.

**Parametreler:**

**translated (giriş)**

True değeri, oturumun aktarıldığı anlamına gelir. False değeri, oturumun aktarılamadığı anlamına gelir.

Bir aracıya gerçek zamanlı bağlantı için değer False olmalıdır.

**acknowledgeMode (giriş)**

Bir uygulama tarafından alınan iletilerin nasıl kabul edildiğine ilişkin bilgi içerir. Değer, AcknowledgeMode sıralı değer listeleycisinden biri olmalıdır:

AcknowledgeMode.AutoAcknowledge  
AcknowledgeMode.ClientAcknowledge  
AcknowledgeMode.DupsOkAcknowledge

Bir aracıya gerçek zamanlı bağlantı için değer AcknowledgeMode.AutoAcknowledge ya da AcknowledgeMode.DupsOkAcknowledge olmalıdır.

Oturum hareket ettiyse bu değıştirge yoksayılr. Alındı bildirim i kipleri hakkında daha fazla bilgi için bkz. [“İleti alındı bildirim i” sayfa 24.](#)

**Döndürülen:**

Oturum nesnesi

**Özel durumlar:**

- XMSException

*Başlatma-Bağlantı Başlat*

**Arabirim:**

```
void Start();
```

Bağlantıya ilişkin gelen iletilerin sağlananını başlatın ya da yeniden başlatın. Bağlantı önceden başlatıldıysa, arama yoksayılr.

**Parametreler:**

Yok

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSException

*Durdur-Bağlantıyı Durdur*

**Arabirim:**

```
void Stop();
```

Bağlantı için gelen iletilerin sağlananını durdurun. Bağlantı önceden durdurulduysa, arama yoksayılr.

**Parametreler:**

Yok

**Döndürülen:**

Hükümsüz

## Özel durumlar:

- XMSException

## Edinilmiş özellikler ve yöntemler

Aşağıdaki yöntemler IPropertyContext arabiriminden devralınır:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortÖzellik, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongÖzelliği, SetObjectProperty, SetShortProperty, SetStringProperty

## IConnectionFactory

Bir uygulama, bağlantı oluşturmak için bir bağlantı üreticisi kullanır.

### Devralma sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IConnectionFactory
```

Bir ConnectionFactory nesnesine ilişkin XMS tanımlı özelliklerinin bir listesi için bkz. [“ConnectionFactoryözellikleri” sayfa 174.](#)

### İlgili kavramlar

ConnectionFactories ve Connection nesneleri

ConnectionFactory nesnesi, uygulamanın Connection nesnesi yaratmak için kullandığı bir şablon sağlar. Uygulama, bir Oturum nesnesi yaratmak için Bağlantı nesnesini kullanır.

Hizmet tümleştirme veriyoluna bağlantı

Bir XMS uygulaması, doğrudan TCP/IP bağlantısı kullanarak ya da TCP/IP üzerinden HTTP kullanarak bir WebSphere Application Server hizmet tümleştirme veriyoluna bağlanabilir.

IBM MQ kuyruk yöneticisine güvenli bağlantı

Bir XMS .NET uygulamasını bir IBM MQ kuyruk yöneticisinde güvenli bağlantı sağlamak üzere etkinleştirmek için, ilgili özelliklerin ConnectionFactory nesnesinde tanımlı olması gerekir.

WebSphere Application Server service integration bus ileti alışverişi altyapısına güvenli bağlantılar

To enable an XMS .NET application to make secure connections to a WebSphere Application Server service integration bus messaging engine, the relevant properties must be defined in the ConnectionFactory object.

Denetlenen nesnelere için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

### İlgili görevler

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

### İlgili başvurular

Denetlenen ConnectionFactory nesnelere için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

## Yöntemler

*CreateConnection -Create Connection Factory (Bağlantı Üreticisi Yarat) (varsayılan kullanıcı kimliğini kullanarak)*

#### **Arabirim:**

```
IConnection CreateConnection();
```

Varsayılan özelliklerle bir bağlantı üreticisi yaratın.

WebSphere MQ ve XMLC\_USERID ' ye bağlanıyorsanız, kuyruk yöneticisi varsayılan olarak oturum açmış kullanıcının userID değerini kullanır. Tek tek kullanıcıların daha fazla bağlantı düzeyinde kimlik doğrulamasına gereksinim duyarsanız, WebSphere MQ içinde yapılandırılmış bir istemci kimlik doğrulaması çıkışı yazabilirsiniz.

#### **Parametreler:**

Yok

#### **Özel durumlar:**

- XMSEException

*CreateConnection -Bağlantı Yarat (belirli bir kullanıcı kimliğini kullanarak)*

#### **Arabirim:**

```
IConnection CreateConnection(String userId, String password);
```

Belirtilen kullanıcı kimliğini kullanarak bir bağlantı yaratın.

WebSphere MQ ve XMLC\_USERID ' ye bağlanıyorsanız, kuyruk yöneticisi varsayılan olarak oturum açmış kullanıcının userID değerini kullanır. Tek tek kullanıcıların daha fazla bağlantı düzeyinde kimlik doğrulamasına gereksinim duyarsanız, WebSphere MQ içinde yapılandırılmış bir istemci kimlik doğrulaması çıkışı yazabilirsiniz.

Bağlantı durdurulmuş kipte yaratılır. Uygulama **Connection.start()** çağrılınca kadar hiçbir ileti teslim edilmez.

#### **Parametreler:**

##### **userID (giriş)**

Bir dizgi nesnesi, uygulamanın kimliğini doğrulamak için kullanılacak kullanıcı kimliğini sarsalıyor. Boş değer sağlıyorsa, kimlik doğrulamadan bağlantıyı yaratma girişiminde bulunmanız gerekir.

##### **parola (giriş)**

Uygulamanın kimliğini doğrulamak için kullanılacak parolayı sarmalayan bir String nesnesi. Boş değer sağlıyorsa, kimlik doğrulamadan bağlantıyı yaratma girişiminde bulunmanız gerekir.

#### **Döndürülen:**

Bağlantı nesnesi.

#### **Özel durumlar:**

- XMSEException
- XMS\_X\_SECURITY\_EXCEPTION

### ***Edinilmiş özellikler ve yöntemler***

Aşağıdaki yöntemler IPropertyContext arabiriminden devralınır:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortÖzellik, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongÖzelliği, SetObjectProperty, SetShortProperty, SetStringProperty

## IConnectionMetaVerileri

Bir ConnectionMetaVeri nesnesi, bir bağlantıyla ilgili bilgi sağlar.

### Devralma sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IConnectionMetaData
```

Bir ConnectionMetaVeri nesnesine ilişkin XMS tanımlı özelliklerinin bir listesi için bkz. [“ConnectionMetaVerileri Özellikleri” sayfa 179.](#)

### .NET özellikler

*JMSXPropertyNames -JMS Tanımlı İleti Özelliklerini Al*

#### Arabirim:

```
System.Collections.IEnumerator JMSXPropertyNames
{
    get;
}
```

Bağlantı tarafından desteklenen, JMS tanımlı ileti özelliklerinin adlarına ilişkin sıralı değer listesi döndürür. JMS tanımlı ileti özellikleri, aracıya gerçek zamanlı bir bağlantı tarafından desteklenmez.

#### Özel durumlar:

- XMSException

### Edinilmiş özellikler ve yöntemler

Aşağıdaki yöntemler [IPropertyContext](#) arabiriminden devralınır:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortÖzellik](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongÖzelligi](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

## Variş Noktası

Bir hedef, bir uygulamanın iletileri göndereceği ya da bir uygulamanın iletileri aldığı bir kaynaktır ya da her ikisi de olabilir.

### Devralma sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IDestination
```

Hedef nesneye ilişkin XMS tanımlı özelliklerinin bir listesi için bkz. [“Hedef Özellikleri” sayfa 180.](#)

### İlgili kavramlar

[ConnectionFactory](#) ve [Connection](#) nesneleri

[ConnectionFactory](#) nesnesi, uygulamanın [Connection](#) nesnesi yaratmak için kullandığı bir şablon sağlar. Uygulama, bir Oturum nesnesi yaratmak için [Bağlantı](#) nesnesini kullanır.

[Hizmet tümleştirme veriyoluna bağlantı](#)

Bir XMS uygulaması, doğrudan TCP/IP bağlantısı kullanarak ya da TCP/IP üzerinden HTTP kullanarak bir WebSphere Application Server hizmet tümleştirme veriyoluna bağlanabilir.

## Hedefler

Bir XMS uygulaması, gönderilmekte olan iletilerin hedefini ve alınmakta olan iletilerin kaynağını belirtmek için bir Hedef nesne kullanır.

## Hedef joker karakterleri

XMS , hedef genel arama karakterleri için destek sağlar. Bu, joker karakterlerin, eşleştirme için gerekli olan yere aktarılabilmesini sağlar. XMS ' un çalışabileceği her bir sunucu tipi için farklı bir joker karakter şeması vardır.

## Konu URL tipi kaynak tanıtıcıları

Konu tekstili kaynak tanıtıcısı (URI), konunun adını belirtir; bir ya da daha çok özellik de belirtebilir.

## Kuyruk tek biçimli kaynak tanıtıcıları

Kuyruğun URI 'si, kuyruğun adını belirtir; ayrıca, kuyruğun bir ya da daha fazla özelliğini de belirtebilir.

## Geçici hedefler

XMS uygulamaları geçici hedefler oluşturabilir ve kullanabilir.

## Denetlenen nesnelere için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

## **İlgili görevler**

### Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

## **İlgili başvurular**

### Yönetilen Hedef nesnelere için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

## **.NET özellikler**

### *Ad-Hedef Adı Al*

#### **Arabirim:**

```
String Name
{
    get;
}
```

Hedefin adını alın. Ad, bir kuyruğun adını ya da bir konunun adını sarmalayan bir dizilimdir.

#### **Özel durumlar:**

- XMSException

### *TypeId -Hedef Tipi Al*

#### **Arabirim:**

```
DestinationType TypeId
{
    get;
}
```

Hedefin tipini alın. Hedefin tipi aşağıdaki değerlerden biridir:

```
DestinationType.Queue
DestinationType.Topic
```



## Özel durumlar:

- XMSEException

## Edinilmiş özellikler ve yöntemler

Aşağıdaki yöntemler [IPropertyContext](#) arabiriminden devralınır:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortÖzellik](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongÖzelliği](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

## ExceptionHandler

### Devralma sıradüzeni:

Yok

Bir uygulama, bağlantıyla ilgili bir sorunun zamanuyumsuz olarak bildirilmesi için bir kural dışı durum dinleyicisi kullanır.

Bir uygulama yalnızca, iletileri zamanuyumsuz olarak tüketebilmek için ve başka hiçbir amaçla bir bağlantı kullanıyorsa, uygulamanın bir sorunla ilgili bilgi edinebilmenin tek yolu bir kural dışı durum dinleyicisi kullanılmasıdır. Başka durumlarda, bir kural dışı durum dinleyicisi, XMS' a bir sonraki zamanuyumlu çağrıya kadar beklemekten daha kısa bir şekilde bir sorunla ilgili bilgi edinmek için daha acil bir yol sağlayabilir.

## Ata

*ExceptionHandler -Kural Dışı Durum Dinleyicisi*

### Arabirim:

```
public delegate void ExceptionListener(Exception ex)
```

Bir sorunla ilgili uygulamayı bir bağlantıyla bilgilendirir.

Bu yetkilendirmeyi uygulayan yöntemler, bağlantıyla birlikte kaydedilebilir.

Kural dışı durum dinleyicilerinin kullanılmasına ilişkin ek bilgi edinmek için [“Message and exception listeners in .NET”](#) sayfa 46' e bakın.

### Parametreler:

#### kural dışı durum (giriş)

XMStarafından oluşturulan bir kural dışı duruma işaret eden gösterge.

### Döndürülen:

Hükümsüz

## IllegalStateKural Dışı Durumu

### Devralma sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.Exception
|
+----IBM.XMS.IllegalStateException
```

XMS throws this exception if an application calls a method at an incorrect or inappropriate time, or if XMS is not in an appropriate state for the requested operation.

## ***Edinilmiş özellikler ve yöntemler***

Aşağıdaki yöntemler, [XMSException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

## **InitialContext**

Bir uygulama, denetlenen nesnelere havuzundan alınan nesne tanımlamalarından nesne yaratmak için InitialContext nesnesini kullanır.

### **Devralma sıradüzeni:**

Yok

### **İlgili kavramlar**

#### InitialContext özellikleri

InitialContext oluşturucusunun değiştiricileri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesnelere havuzunun yerini içerir. Bir uygulamanın havuzla bağlantı kurmasını sağlamak için, URI içinde yer alan bilgilerden daha fazla bilgi sağlamak gerekebilir.

#### XMS başlangıç bağlamı için URI biçimi

Denetlenen nesnelere havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak sağlanır. URI biçimi bağlam tipine bağlıdır.

#### Yönetilen nesnelere alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

### **İlgili görevler**

#### InitialContext nesnelere

Bir uygulama, denetlenen nesnelere havuzuna, gerekli denetlenen nesnelere almak için bir bağlantı kurmak üzere kullanılacak bir başlangıç bağlamı oluşturmalıdır.

## ***.NET özellikler***

*Ortam-Ortamı alın*

### **Arabirim:**

```
Hashtable Environment
{
    get;
}
```

Ortamı alın.

### **Özel durumlar:**

- Kural dışı durumlar, kullanılmakta olan dizin hizmetine özgüdür.

## ***Oluşturucular***

*InitialContext -Başlangıç Bağlamı Yarat*

### **Arabirim:**

```
InitialContext(Hashtable env);
```

Bir InitialContext nesnesi yaratın.

**Parametreler:**

Denetlenen nesnelere havuzlarıyla bağlantı kurmak için gereken bilgiler, oluşturucuda bir ortam hashtable 'ında sağlanır.

**Özel durumlar:**

- XMSEException

**Yöntemler**

*AddToOrtamı-Ortama Yeni Bir Özellik Ekle*

**Arabirim:**

```
Object AddToEnvironment(String propName, Object propVal);
```

Ortama yeni bir özellik ekleyin.

**Parametreler:****propName (giriş)**

Eklenecek özelliğin adını sarmalayan bir String nesnesi.

**propVal (giriş)**

Eklenecek özelliğin değeri.

**Döndürülen:**

Özelliğin eski değeri.

**Özel durumlar:**

- Kural dışı durumlar, kullanılmakta olan dizin hizmetine özgüdür.

*Kapat-Bu bağlamı kapat*

**Arabirim:**

```
void Close();
```

Bu bağlamı kapatın.

**Parametreler:**

Yok

**Döndürülen:**

Yok

**Özel durumlar:**

- Kural dışı durumlar, kullanılmakta olan dizin hizmetine özgüdür.

*Arama-Başlangıç Bağlamında Nesne Arama*

**Arabirim:**

```
Object Lookup(String name);
```

Denetlenen nesnelere havuzundan alınan bir nesne tanımlamasından nesne yaratın.

**Parametreler:****ad (giriş)**

Alınacak denetimli nesnenin adını sarsalayan bir String nesnesi. Ad yalın bir ad ya da karmaşık bir ad olabilir. Daha fazla ayrıntı için bkz. [“Yönetilen nesnelere alınması” sayfa 62.](#)

**Döndürülen:**

Alınmakta olan nesnenin tipine bağlı olarak bir IConnectionFactory ya da bir IDestination değeri. İşlev dizine erişebiliyorsa, ancak gerekli nesneyi bulamazsa, boş değer döndürülür.

**Özel durumlar:**

- Kural dışı durumlar, kullanılmakta olan dizin hizmetine özgüdür.

*RemoveFromEnvironment-Ortamdan Bir Özelliği Kaldırma*

**Arabirim:**

```
Object RemoveFromEnvironment(String propName);
```

Bir özelliği ortamdaki bir özelliği kaldırın.

**Parametreler:****propName (giriş)**

Kaldırılacak özelliğin adını sarmalayan bir String nesnesi.

**Döndürülen:**

Kaldırılan nesne.

**Özel durumlar:**

- Kural dışı durumlar, kullanılmakta olan dizin hizmetine özgüdür.

## InvalidClientIDException

**Devralma sıradüzeni:**

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.InvalidClientIDException
```

XMS , bir uygulama bir bağlantı için istemci tanıtıcısı belirlemeye çalışırsa, ancak istemci tanıtıcısı geçerli değilse ya da önceden kullanımda olduğu için bu kural dışı durumu atar.

***Edinilmiş özellikler ve yöntemler***

Aşağıdaki yöntemler, [XMSEException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

## InvalidDestinationKural Dışı Durumu

**Devralma sıradüzeni:**

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.InvalidDestinationException
```

Bir uygulama geçerli olmayan bir hedef belirtiyorsa,XMS bu kural dışı durumu atar.

***Edinilmiş özellikler ve yöntemler***

Aşağıdaki yöntemler, [XMSEException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

## InvalidSelectorKural Dışı Durumu

### Devralma sıradüzeni:

```
IBM.XMS.XMSException
|
+----IBM.XMS.XMSException
      |
      +----IBM.XMS.InvalidSelectorException
```

Bir uygulama, sözdizimi geçerli olmayan bir ileti seçici ifadesi sağlıyorsa,XMS bu kural dışı durumu verir.

### Edinilmiş özellikler ve yöntemler

Aşağıdaki yöntemler, [XMSException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

## IMapMessage

Eşlem iletisi, her değer ilişkili bir veri tipine sahip olduğu bir ad-değer çiftleri kümesinden oluşan bir iletidir.

### Devralma sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
      |
      +----IBM.XMS.IMapMessage
```

Bir uygulama ad-değer çiftinin değerini aldığı anda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Bu örtük dönüştürme biçimiyle ilgili daha fazla bilgi için bkz. [“Eşleme iletileri”](#) sayfa 74.

### İlgili başvurular

[Eşleme iletileri](#)

Bir eşlem iletisinin gövdesi, her değer ilişkili bir veri tipine sahip olduğu ad-değer çiftleri kümesini içerir.

### .NET özellikler

*MapNames -Eşlem Adlarını Al*

#### Arabirim:

```
System.Collections.IEnumerator MapNames
{
    get;
}
```

Eşlem iletisinin gövdesindeki adlara ilişkin bir sıralı değer listesi alın.

#### Özel durumlar:

- XMSException

### Yöntemler

*GetBoolean -Boole Deęeri Al*

**Arabirim:**

```
Boolean GetBoolean(String name);
```

Eşlem iletisinin gövdesinden ada göre tanımlanan Boole deęerini alın.

**Parametreler:**

**ad (giriş)**

Bir String nesnesi, boole deęerini tanımlayan adı sarmalayan bir nesne.

**Döndürülen:**

Eşlem iletisinin gövdesinden alınan Boole deęeri.

**Özel durumlar:**

- XMSEException

*GetByte -Byte Al*

**Arabirim:**

```
Byte GetByte(String name);  
Int16 GetSignedByte(String name);
```

Eşlemin gövdesinden ada göre tanımlanan baytı alın.

**Parametreler:**

**ad (giriş)**

Byte 'ı tanımlayan adı sarmalayan bir String nesnesi.

**Döndürülen:**

Harita iletisinin gövdesinden alınan bayt. Byte üzerinde veri dönüştürme işlemi gerçekleştirilmez.

**Özel durumlar:**

- XMSEException

*GetBytes -Alma Baytları*

**Arabirim:**

```
Byte[] GetBytes(String name);
```

Eşlem iletisinin gövdesinden ada göre tanımlanan bayt dizisini alın.

**Parametreler:**

**ad (giriş)**

Bir dize nesnesi, bayt dizisini tanımlayan adı sarmalayan bir nesnedir.

**Döndürülen:**

Dizideki bayt sayısı.

**Özel durumlar:**

- XMSEException

## *GetChar -Karakter Al*

### **Arabirim:**

```
Char GetChar(String name);
```

Eşlemin gövdesinden ada göre tanımlanan karakteri alın.

### **Parametreler:**

#### **ad (giriş)**

Bir String nesnesi, karakteri tanımlayan adı sarmalayan bir nesnedir.

### **Döndürülen:**

Eşlem iletisinin gövdesinden alınan karakter.

### **Özel durumlar:**

- XMSEException

## *GetDouble -Çift Duyarlıklı Kayan Nokta Numarası Al*

### **Arabirim:**

```
Double GetDouble(String name);
```

Eşlem iletisinin gövdesinden ada göre tanımlanan çift duyarlıklı kayan nokta numarasını alın.

### **Parametreler:**

#### **ad (giriş)**

Çift duyarlıklı kayan nokta numarasını tanımlayan adı sarmalayan bir String nesnesi.

### **Döndürülen:**

İleti eşleme iletisinin gövdesinden alınan çift duyarlıklı kayan noktalı sayı.

### **Özel durumlar:**

- XMSEException

## *GetFloat -Kayar Noktalı Sayı Al*

### **Arabirim:**

```
Single GetFloat(String name);
```

Eşlem iletisinin gövdesinden ada göre tanımlanan kayan noktalı sayıyı alın.

### **Parametreler:**

#### **ad (giriş)**

Bir dize nesnesi, kayan nokta numarasını tanımlayan adı sarmalayan bir nesnedir.

### **Döndürülen:**

Eşleme iletisinin gövdesinden alınan kayan noktalı sayı.

### **Özel durumlar:**

- XMSEException

### *GetInt -Tamsayı Al*

#### **Arabirim:**

```
Int32 GetInt(String name);
```

Eşlem iletilisinin gövdesinden ada göre tanımlanan tamsayıyı alın.

#### **Parametreler:**

##### **ad (giriş)**

Bir dizgi nesnesi, tamsayıyı tanımlayan adı sarmalayan bir nesne.

#### **Döndürülen:**

Eşlem iletilisinin gövdesinden alınan tamsayı.

#### **Özel durumlar:**

- XMSEException

### *GetLong -Uzun Tamsayı Al*

#### **Arabirim:**

```
Int64 GetLong(String name);
```

Eşlem iletilisinin gövdesinden ada göre tanımlanan uzun tamsayıyı alın.

#### **Parametreler:**

##### **ad (giriş)**

Uzun tamsayıyı tanımlayan adı sarmalayan bir String nesnesi.

#### **Döndürülen:**

Eşlem iletilisinin gövdesinden alınan uzun tamsayı.

#### **Özel durumlar:**

- XMSEException

### *GetObject -Nesne Al*

#### **Arabirim:**

```
Object GetObject(String name);
```

Eşlem iletilisinin gövdesinden bir ad-değer çiftinin değerine bir başvuru alın. Ad-değer çifti adıyla tanımlanır.

#### **Parametreler:**

##### **ad (giriş)**

Ad-değer çiftinin adını sarsalayan bir String nesnesi.

#### **Döndürülen:**

Aşağıdaki nesne tiplerinden biri olan değer:

Boolean  
Byte  
Byte[]  
Char  
Double  
Single



Int32  
Int64  
Int16  
String

**Özel durumlar:**

XMSEException

*GetShort -Kısa Tamsayı Al*

**Arabirim:**

```
Int16 GetShort(String name);
```

Eşlem iletilisinin gövdesinden ada göre tanımlanan kısa tamsayıyı alın.

**Parametreler:**

**ad (giriş)**

Bir dizgi nesnesi, kısa tamsayıyı tanımlayan adı sarkıyor.

**Döndürülen:**

Eşlem iletilisinin gövdesinden alınan kısa tamsayı.

**Özel durumlar:**

- XMSEException

*GetString -Get String*

**Arabirim:**

```
String GetString(String name);
```

Eşlemin gövdesinden ada göre tanımlanan dizgiyi alın.

**Parametreler:**

**ad (giriş)**

Bir String nesnesi, eşlem iletilisinin gövdesindeki dizeyi tanımlayan adı sarkıyor.

**Döndürülen:**

Eşleme iletilisinin gövdesinden alınan dizeyi sarmalayan bir String nesnesi. Veri dönüştürme gerekiyorsa, bu değer dönüştürmenin ardından dizgidir.

**Özel durumlar:**

- XMSEException

*ItemExists -Denetim Adı-Değer çifti Var*

**Arabirim:**

```
Boolean ItemExists(String name);
```

Eşlem iletilisinin gövdesinin, belirtilen ada sahip bir ad-değer çifti içerip içermediğini denetleyin.

**Parametreler:****ad (giriş)**

Ad-değer çiftinin adını sarsalayan bir String nesnesi.

**Döndürülen:**

- True, if the body of the map message contains a name-value pair with the specified name.
- False, if the body of the map message does not contain a name-value pair with the specified name.

**Özel durumlar:**

- XMSEException

*SetBoolean -Boole Değerini Ayarla***Arabirim:**

```
void SetBoolean(String name, Boolean value);
```

Eşlem iletisinin gövdesinde bir boole değeri ayarlayın.

**Parametreler:****ad (giriş)**

Bir dizgi nesnesi, eşlem iletisinin gövdesindeki boole değerini saptamak için adı sarkıyor.

**değer (giriş)**

Ayarlanacak boole değeri.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException

*SetByte -Set Byte***Arabirim:**

```
void SetByte(String name, Byte value);  
void SetSignedByte(String name, Int16 value);
```

Eşlem iletisinin gövdesinde bir bayt belirleyin.

**Parametreler:****ad (giriş)**

Bir dizgi nesnesi, eşlem iletisinin gövdesindeki byte 'ı tanıtmak için adı sarkıyor.

**değer (giriş)**

Ayarlanacak bayt.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException

## *SetBytes -Set Bytes*

### **Arabirim:**

```
void SetBytes(String name, Byte[] value);
```

Eşlem iletisinin gövdesinde bir bayt dizisi ayarlayın.

### **Parametreler:**

#### **ad (giriş)**

Bir String nesnesi, eşlem iletisinin gövdesindeki byte dizilerinin tanınabilmek için adını sarkıyor.

#### **değer (giriş)**

Ayarlanacak bayt dizisi.

### **Döndürülen:**

Hükümsüz

### **Özel durumlar:**

- XMSEException

## *SetChar -Karakter kümesi*

### **Arabirim:**

```
void SetChar(String name, Char value);
```

Eşlem iletisinin gövdesinde 2 byte 'lık bir karakter belirleyin.

### **Parametreler:**

#### **ad (giriş)**

Bir String nesnesi, eşlem iletisinin gövdesindeki karakteri tanımlamak için adı sarkıyor.

#### **değer (giriş)**

Ayarlanacak karakter.

### **Döndürülen:**

Hükümsüz

### **Özel durumlar:**

- XMSEException

## *SetDouble -Çift Duyarlıklı Kayan Nokta Numarası Ayarla*

### **Arabirim:**

```
void SetDouble(String name, Double value);
```

Eşlem iletisinin gövdesinde bir çift duyarlıklı kayan noktalı sayı belirleyin.

### **Parametreler:**

#### **ad (giriş)**

Eşlem iletisinin gövdesindeki çift duyarlıklı kayan noktalı sayıyı tanımlamak için adı sarmalayan bir String nesnesi.

#### **değer (giriş)**

Ayarlanacak çift duyarlıklı kayan noktalı sayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException

*SetFloat -Değişken Nokta Numarasını Ayarla***Arabirim:**

```
void SetFloat(String name, Single value);
```

Eşlem iletilisinin gövdesinde bir kayar noktalı sayı belirleyin.

**Parametreler:****ad (giriş)**

Bir dizgi nesnesi, eşlem iletilisinin gövdesindeki kayan noktalı sayıyı tanımlamak için adı sarkıyor.

**değer (giriş)**

Ayarlanacak kayan noktalı sayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException

*SetInt -Tamsayı Ayarla***Arabirim:**

```
void SetInt(String name, Int32 value);
```

Eşlem iletilisinin gövdesinde bir tamsayı belirtin.

**Parametreler:****ad (giriş)**

Bir dizgi nesnesi, eşlem iletilisinin gövdesindeki tamsayıyı tanımlamak için adı sarkıyor.

**değer (giriş)**

Ayarlanacak tamsayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException

*SetLong -Uzun Tamsayı Belirle***Arabirim:**

```
void SetLong(String name, Int64 value);
```

Eşlem iletilisinin gövdesinde uzun bir tamsayı belirtin.

**Parametreler:****ad (giriş)**

Bir String nesnesi, eşlem iletisinin gövdesindeki uzun tamsayıyı tanımlamak için adı sarkıyor.

**değer (giriş)**

Ayarlanacak uzun tamsayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException

*SetObject -Nesneyi Ayarla*

**Arabirim:**

```
void SetObject(String name, Object value);
```

Eşlem iletisinin gövdesinde bir XMS temel tipi olması gereken bir değer ayarlayın.

**Parametreler:****ad (giriş)**

Bir dizgi nesnesi, eşlem iletisinin gövdesindeki değeri tanımlamak için adı sarkıyor.

**değer (giriş)**

Ayarlanacak değeri içeren bayt dizisi.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException

*SetShort -Kısa Tamsayı Ayarla*

**Arabirim:**

```
void SetShort(String name, Int16 value);
```

Eşlem iletisinin gövdesinde kısa bir tamsayı belirtin.

**Parametreler:****ad (giriş)**

Bir dizgi nesnesi, eşlem iletisinin gövdesindeki kısa tamsayıyı tanımlamak için adı sarkıyor.

**değer (giriş)**

Ayarlanacak kısa tamsayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException

## SetString -Dizgi Ayarla

### Arabirim:

```
void SetString(String name, String value);
```

Eşlem iletisinin gövdesinde bir dizgi belirleyin.

### Parametreler:

#### ad (giriş)

Bir String nesnesi, eşlem iletisinin gövdesindeki diziyi tanımlamak için adı sarsalıyor.

#### değer (giriş)

Ayarlanacak diziyi sarmalayan bir String nesnesi.

### Döndürülen:

Hükümsüz

### Özel durumlar:

- XMSException

## Edinilmiş özellikler ve yöntemler

Aşağıdaki özellikler [IMessage](#) arabiriminden devralınır:

[JMSCorrelationID](#), [JMSDeliveryMode](#), [JMSDestination](#), [JMSExpiration](#), [JMSMessageID](#), [JMSPriority](#), [JMSRevered](#), [JMSReplyTo](#), [JMSTimestapp](#), [JMSType](#), [Properties](#)

Aşağıdaki yöntemler [IMessage](#) arabiriminden devralınır:

[clearBody](#), [clearProperties](#), [PropertyExists](#)

Aşağıdaki yöntemler [IPropertyContext](#) arabiriminden devralınır:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortÖzellik](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongÖzelliği](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

## İleti

Bir ileti nesnesi, uygulamanın gönderdiği ya da aldığı bir iletiyi gösterir. [IMessage](#) is a superclass for the message classes such as [IMapMessage](#).

### Devralma sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
```

İleti nesnesindeki JMS iletisi üstbilgisi alanlarının bir listesi için bkz. "[XMS iletisinde üstbilgi alanları](#)" sayfa 67. Bir ileti nesnesine ilişkin JMS tanımlı özelliklerinin bir listesi için bkz. "[JMS-iletinin tanımlı özellikleri](#)" sayfa 69. Bir ileti nesnesine ilişkin IBM tanımlı özelliklerinin bir listesi için bkz. "[IBM-iletinin tanımlı özellikleri](#)" sayfa 69. İleti nesnesine ilişkin JMS\_IBM\_MQMD\* özelliklerinin bir listesi için bkz. "[JMS\\_IBM\\_MQMD\\* özellikleri](#)" sayfa 184

İletiler, çöp toplayıcı tarafından silinir. Bir ileti silindiğinde, bu ileti kullanmakta olduğu kaynakları boşaltır.

### İlgili başvurular

#### [XMS ileti](#)

Bu bölümde, XMS iletilerinin yapısı ve içeriği ele alınmıştır ve uygulamaların XMS iletilerinin nasıl işleyeceğini açıklar.

## **.NET özellikler**

*GetJMSCorrelationKimlik-Alma ve Ayarlama JMSCorrelationID*

### **Arabirim:**

```
String JMSCorrelationID
{
    get;
    set;
}
```

İletinin ilinti tanıtıcısını String nesnesi olarak ayarlayın ve ayarlayın.

### **Özel durumlar:**

- XMSEException

*JMSDeliveryMode -Get and Set JMSDeliveryMode*

### **Arabirim:**

```
DeliveryMode JMSDeliveryMode
{
    get;
    set;
}
```

İletiyi ilişkin teslim kipini alın ve ayarlayın.

İletinin teslim kipi aşağıdaki değerlerden biridir:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

Gönderilmeyen yeni oluşturulan bir ileti için teslim kipi `DeliveryMode.Persistent` dir. Kalıcı, teslim kipinin `DeliveryMode.NonPersistent` olduğu bir aracıya gerçek zamanlı bağlantı dışında. Alınan bir ileti için yöntem, `receiving.send ()` çağrısı tarafından ayarlanan uygulama teslim kipini döndürür; ancak, alma kipi `JMSDeliveryMode` ayarlanarak teslim kipini değiştirmedeği sürece, ileti, `IMessageProducer.send ()` çağrısıyla ayarlanan teslim kipini döndürür.

### **Özel durumlar:**

- XMSEException

*JMSDestination-JMSDestination Al*

### **Arabirim:**

```
IDestination JMSDestination
{
    get;
    set;
}
```

İletinin hedefini alın ve ayarlayın.

Hedef, ileti gönderildiğinde `IMessageProducer.send ()` çağrısıyla ayarlanır. `JMSDestination` değeri yok sayıldı. Ancak, alınan bir iletinin hedefini değiştirmek için `JMSDestination` 'ı kullanabilirsiniz.

Gönderilmeyen yeni oluşturulan bir ileti için, gönderme uygulaması `JMSDestination` ayarlanarak bir hedef belirlemezse, yöntem boş değerli bir hedef nesne döndürür. For a message that was received, the

method returns a Destination object for the destination that was set by the IMessageProducer.send() call when the message was sent unless the receiving application changes the destination by setting JMSTDestination.

**Özel durumlar:**

- XMSEException

*JMSEExpiration-JMSEExpiration Al*

**Arabirim:**

```
Int64 JMSEExpiration
{
    get;
    set;
}
```

İletinin son kullanma tarihini alın ve ayarlayın.

Süre bitim süresi, ileti gönderildiğinde IMessageProducer.send () çağrısına göre ayarlanır. Değeri, gönderen uygulama tarafından belirtildiği gibi, iletinin gönderildiği zamana kadar, iletinin gönderildiği saat eklenerek hesaplanır. Süre bitimi, 1 Ocak 1970 'te 00:00:00 GMT ' den bu yana milisaniye olarak ifade edilir.

Gönderilmeyen yeni oluşturulan bir ileti için, gönderme uygulaması JMSEExpiration ayarını ayarlayarak farklı bir süre bitim süresi ayarlamadıkça, süre bitim saati 0 olur. For a message that was received, the method returns the expiration time that was set by the IMessageProducer.send() call when the message was sent unless the receiving application changes the expiration time by setting JMSEExpiration.

If the time to live is 0, the IMessageProducer.send() call sets the expiration time to 0 to indicate that the message does not expire.

XMS , süresi dolmuş iletileri atar ve bunları uygulamalara teslim etmez.

**Özel durumlar:**

- XMSEException

*JMSMessageID - Get and Set JMSMessageID*

**Arabirim:**

```
String JMSMessageID
{
    get;
    set;
}
```

İletinin ileti tanıtıcısını alın ve ileti tanıtıcısını sarmalayan bir dizgi nesnesi olarak ayarlayın.

İleti tanıtıcısı, ileti gönderildiğinde IMessageProducer.send () çağrısıyla belirlenir. For a message that was received, the method returns the message identifier that was set by the IMessageProducer.send() call when the message was sent unless the receiving application changes the message identifier by setting JMSMessageID.

İletinin ileti tanıtıcısı yoksa, yöntem boş değer döndürür.

**Özel durumlar:**

- XMSEException



**Arabirim:**

```
Int32 JMSPriority
{
    get;
    set;
}
```

İletinin önceliğini alın ve ayarlayın.

Öncelik, ileti gönderildiğinde IMessageProducer.send () çağrısına göre ayarlanır. The value is an integer in the range 0, the lowest priority, to 9, the highest priority.

Gönderilmeyen yeni oluşturulan bir ileti için, gönderme uygulaması JMSPriority ayarını tanımlayarak farklı bir öncelik ayarlamadıkça, öncelik 4 olur. Alınan bir ileti için, yöntem, JMSPriority değerini ayarlayarak, alma uygulamasının önceliği değiştirmedeği sürece, ileti gönderildiğinde, yöntem IMessageProducer.send () çağrısıyla ayarlanan önceliği döndürür.

**Özel durumlar:**

- XMSEException

**Arabirim:**

```
Boolean JMSRedelivered
{
    get;
    set;
}
```

İletinin yeniden teslim edilip edilmediğine ilişkin bir gösterge elde edin ve iletinin yeniden teslim edilip edilmediğini belirtin. Bu gösterge, ileti alındığında IMessageConsumer.alncall () çağrısına göre ayarlanır.

Bu özellik aşağıdaki değerlere sahiptir:

- True, if the message is being redelivered.
- False, if the message is not being redelivered.

Bir aracıya gerçek zamanlı bağlantı için değer her zaman False' dir.

İleti gönderilmeden önce, JMSResaled tarafından ayarlanan yeniden teslim edilmenin bir göstergesi, ileti gönderildiğinde IMessageProducer.send () çağrısı dikkate alınmaz ve ileti alındığında IMessageConsumer.receive () çağrısıyla yoksayılır ve değiştirilir. Ancak, alınan bir iletinin göstergesini değiştirmek için JMSReverled olanağını kullanabilirsiniz.

**Özel durumlar:**

- XMSEException

**Arabirim:**

```
IDestination JMSReplyTo
{
    get;
    set;
}
```

İletinin yanıtının gönderileceği hedefi alın ve ayarlayın.

Bu özelliğin değeri, iletiye verilen yanıtın gönderileceği hedef nesne nesnesidir. Boş değerli bir hedef nesne, yanıt beklenmediği anlamına gelir.

#### Özel durumlar:

- XMSEException

*JMSTimestamp-JMSTimestamp 'ı Al ve Ayarla*

#### Arabirim:

```
Int64 JMSTimestamp
{
    get;
    set;
}
```

İletinin gönderildiği saati alın ve ayarlayın.

Zaman damgası, ileti gönderildiğinde IMessageProducer.send () çağrısına göre ayarlanır ve 1 Ocak 1970 'te 00:00:00 GMT ' den bu yana milisaniye olarak ifade edilir.

Gönderilmeyen yeni oluşturulan bir ileti için, gönderme uygulaması JMSTimestamp ayarlanarak farklı bir zaman damgası ayarlamadıkça, zaman damgası 0 olur. For a message that was received, the method returns the time stamp that was set by the IMessageProducer.send() call when the message was sent unless the receiving application changes the time stamp by setting JMSTimestamp.

#### Özel durumlar:

- XMSEException

#### Notlar:

1. Zaman damgası tanımsızsa, yöntem 0 değerini döndürür, ancak kural dışı durum yayınlanmaz.

*JMSType-JMSType 'ı Al ve Ayarla*

#### Arabirim:

```
String JMSType
{
    get;
    set;
}
```

İletinin tipini alın ve ayarlayın.

JMSType değeri, iletinin tipini sarmalayan bir dizilimdir. Veri dönüştürme gerekliyse, bu değer dönüştürmenin ardından tiptir.

#### Özel durumlar:

- XMSEException

*PropertyNames -Özellikleri Al*

#### Arabirim:

```
System.Collections.IEnumerator PropertyNames
{
```

```
    get;  
}
```

İletinin ad özelliklerine ilişkin sıralı değer listesi alın.

#### Özel durumlar:

- XMSEException

## Yöntemler

*Alındı Bilgisi-Akbilgi*

#### Arabirim:

```
void Acknowledge();
```

Bu iletiyi ve oturum tarafından alınan önceden kabul edilmemiş tüm iletileri kabul edin.

Bir uygulama, oturumun alındı bildirimini kipi AcknowledgeMode.ClientAcknowledgeise bu yöntemi çağırabilir. Oturum başka bir alındı bildirimini kipine sahipse ya da işlem hareketliyse, yöntemeye yapılan çağrılar yoksayılr.

Alınan ancak kabul edilmeyen iletiler yeniden teslim edilebilir.

İletilerin kabul edilmesi hakkında daha fazla bilgi için bkz. [“İleti alındı bildirimini” sayfa 24.](#)

#### Parametreler:

Yok

#### Döndürülen:

Hükümsüz

#### Özel durumlar:

- XMSEException
- IllegalStateKural Dışı Durumu

*ClearBody -Gövde Temizle*

#### Arabirim:

```
void ClearBody();
```

İletinin gövdeini temizleyin. Üstbilgi alanları ve ileti özellikleri temizlenmez.

Bir uygulama bir ileti gövdeyi temizlerse, gövde yeni oluşturulan bir iletide boş bir gövle aynı durumda kalır. Yeni oluşturulan bir iletide boş bir gövdenin durumu, ileti gövdesinin tipine bağlıdır. Daha fazla bilgi için, bkz. [“XMS iletisinin gövdesi” sayfa 71.](#)

Bir uygulama, vücudun hangi durumda olursa olsun, bir ileti gövdinden herhangi bir zamanda temizlenemez. Bir ileti gövdesi salt okunursa, uygulamanın gövdeye yazabileceği tek yol, uygulamanın önce göveyi temizleyebilmesini sağlar.

#### Parametreler:

Yok

#### Döndürülen:

Hükümsüz

#### Özel durumlar:

- XMSEException

## *ClearProperties -Özellikleri Temizle*

### **Arabirim:**

```
void ClearProperties();
```

İletinin özelliklerini temizleyin. Üstbilgi alanları ve ileti gövdesi temizlenmez.

Bir uygulama bir iletinin özelliklerini temizlerse, özellikler okunabilir ve yazılabilir hale gelir.

Bir uygulama, özelliklerin hangi durumda olursa olsun, her an bir iletinin özelliklerini temizleyebilir. Bir iletinin özellikleri salt okunur ise, uygulamanın yazılabilir hale gelebileceği tek yol, uygulamanın önce özellikleri temizleyebilmesini sağlar.

### **Parametreler:**

Yok

### **Döndürülen:**

Hükümsüz

### **Özel durumlar:**

- XMSEException

## *PropertyExists -Denetleme Özelliği Var*

### **Arabirim:**

```
Boolean PropertyExists(String propertyName);
```

İletinin belirtilen ada sahip bir özelliği olup olmadığını denetleyin.

### **Parametreler:**

#### **propertyName (giriş)**

Özelliğin adını sarmalayan bir String nesnesi.

### **Döndürülen:**

- True, if the message has a property with the specified name.
- False, if the message does not have a property with the specified name.

### **Özel durumlar:**

- XMSEException

## **Edinilmiş özellikler ve yöntemler**

Aşağıdaki yöntemler IPropertyContext arabiriminden devralınır:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortÖzellik, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongÖzelliği, SetObjectProperty, SetShortProperty, SetStringProperty

## **IMessageConsumer**

Bir uygulama, bir hedefe gönderilen iletileri almak için bir ileti tüketicisi kullanır.

## Devralma sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessageConsumer
```

Bir MessageConsumer nesnesine ilişkin XMS tanımlı özelliklerinin bir listesi için bkz. [“MessageConsumerözellikleri” sayfa 187.](#)

## .NET özellikler

*MessageListener -İleti Dinleyicisini Al ve Ayarla*

### Arabirim:

```
MessageListener MessageListener
{
    get;
    set;
}
```

İleti tüketicisi ile kaydedilen ileti dinleyicisini edinin ve ileti dinleyicisine ileti dinleyicisini kaydettirin.

İleti tüketicisi ile kayıtlı bir ileti dinleyicisi yoksa, MessageListener boş değerli olur. İleti dinleyicisi zaten ileti tüketicisi ile kayıtlıysa, bunun yerine boş değer belirterek kayıt işlemini iptal edebilirsiniz.

İleti dinleyicileri kullanma hakkında daha fazla bilgi için bkz. [“Message and exception listeners in .NET” sayfa 46.](#)

### Özel durumlar:

- XMSEException

*MessageSelector -İleti Seçiciyi Al*

### Arabirim:

```
String MessageSelector
{
    get;
}
```

İleti tüketicisi için ileti seçiciyi alın. Dönüş değeri, ileti seçici ifadesini sarmalayan bir String nesnesidir. Veri dönüştürme gerekiyorsa, bu değer dönüştürmeden sonraki ileti seçici ifadesidir. İleti tüketicisi bir ileti seçiciye sahip değilse, MessageSelector değeri bir boş Dizgi nesnesidir.

### Özel durumlar:

- XMSEException

## Yöntemler

*Kapat-İleti Tüketicisi Kapat*

### Arabirim:

```
void Close();
```

İleti tüketicisini kapatın.

Bir uygulama önceden kapatılmış bir ileti tüketicisi kapatmaya çalışırsa, arama yoksayılr.

**Parametreler:**

Yok

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException

*Alma-Alma*

**Arabirim:**

```
IMessage Receive();
```

İleti tüketicisi için bir sonraki iletiyi alın. Çağrı, bir ileti için süresiz olarak ya da ileti tüketicisi kapatılıncaya kadar bekler.

**Parametreler:**

Yok

**Döndürülen:**

İleti nesnesine bir gösterge. Çağrı bir iletiyi beklerken ileti tüketicisi kapatılırsa, yöntem boş değerli bir ileti nesnesine ilişkin bir gösterge döndürür.

**Özel durumlar:**

- XMSEException

*Alma-Alma (bekleme aralığıyla birlikte)*

**Arabirim:**

```
IMessage Receive(Int64 delay);
```

İleti tüketicisi için bir sonraki iletiyi alın. Çağrı, ileti için yalnızca belirli bir dönemi bekler ya da ileti tüketicisi kapatılıncaya kadar bekler.

**Parametreler:**

**gecikme (giriş)**

Çağrının bir ileti için bekleyeceği süre (milisaniye). Bir bekleme aralığı 0 belirlerseniz, arama ileti için süresiz olarak bekler.

**Döndürülen:**

İleti nesnesine bir gösterge. Bekleme süresi boyunca hiçbir ileti gelmezse ya da çağrı bir iletiyi beklerken ileti tüketicisi kapatılırsa, yöntem boş değerli bir Message nesnesine bir gösterge döndürür, ancak kural dışı durum yayınlanmaz.

**Özel durumlar:**

- XMSEException

*ReceiveNoBekleme-Bekleme ile Alma*

**Arabirim:**

```
IMessage ReceiveNowait();
```

İleti tüketicisi için bir sonraki iletiyi alırsanız, hemen bir sonraki iletiyi alın.

**Parametreler:**

Yok

**Döndürülen:**

Bir ileti nesnesine ilişkin gösterge. Hemen kullanılabilir bir ileti yoksa, yöntem boş değerli bir ileti nesnesine işaretçi döndürür.

**Özel durumlar:**

- [XMSEException](#)

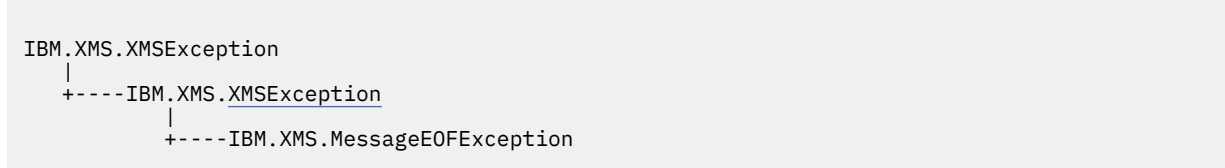
### ***Edinilmiş özellikler ve yöntemler***

Aşağıdaki yöntemler [IPropertyContext](#) arabiriminden devralınır:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortÖzellik](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongÖzelligi](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

### **MessageEOFException**

**Devralma sıradüzeni:**



XMS , bir uygulama byte iletilerinin gövdeini okurken bir byte ileti akışının sonuna rastlanırsa,XMS bu kural dışı durumu döndürür.

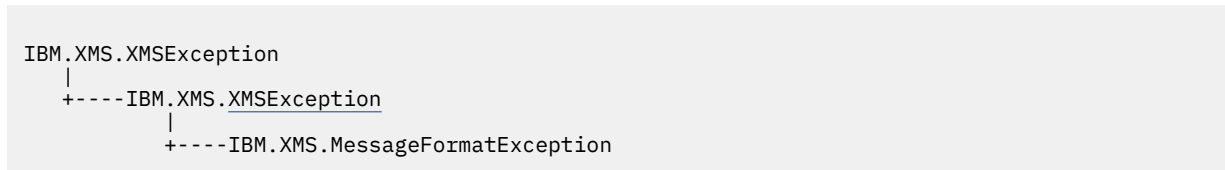
### ***Edinilmiş özellikler ve yöntemler***

Aşağıdaki yöntemler, [XMSEException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

### **MessageFormatKural Dışı Durumu**

**Devralma sıradüzeni:**



XMS throws this exception if XMS encounters a message with a format that is not valid.

### ***Edinilmiş özellikler ve yöntemler***

Aşağıdaki yöntemler, [XMSEException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

## IMessageListener (temsilci)

### Devralma sıradüzeni:

Yok

Bir uygulama, iletileri zamanuyumsuz olarak almak için bir ileti dinleyicisini kullanır.

### Ata

*MessageListener -İleti Dinleyicisi*

### Arabirim:

```
public delegate void MessageListener(IMessage msg);
```

İleti tüketicisine zamanuyumsuz bir ileti sunun.

Bu yetkilendirmeyi uygulayan yöntemler, bağlantıyla birlikte kaydedilebilir.

İleti dinleyicileri kullanma hakkında daha fazla bilgi için bkz. "[Message and exception listeners in .NET](#)" sayfa 46.

### Parametreler:

#### **mesg (giriş)**

İleti nesnesi.

### Döndürülen:

Hükümsüz

## MessageNotReadableException

### Devralma sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageNotReadableException
```

XMS , bir uygulama yalnızca yazma işlemi olan bir iletinin gövdenini okumayı denediğinde bu kural dışı durumu atar.

### **Edinilmiş özellikler ve yöntemler**

Aşağıdaki yöntemler, [XMSEException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

## MessageNotWritableException

### Devralma sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageNotWritableException
```

XMS , bir uygulama salt okunur bir iletinin gövdesine yazma girişiminde bulunursa bu kural dışı durumu atar.



## **Edinilmiş özellikler ve yöntemler**

Aşağıdaki yöntemler, [XMSException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

## **IMessageProducer**

Bir uygulama, bir hedefe ileti göndermek için ileti üreticiyi kullanır.

### **Devralma sıradüzeni:**

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessageProducer
```

Bir MessageProducer nesnesine ilişkin XMS tanımlı özelliklerinin bir listesi için bkz. [“MessageProducerözellikleri” sayfa 188.](#)

## **.NET özellikler**

*DeliveryMode -Varsayılan Teslim Kipini Al ve Ayarla*

### **Arabirim:**

```
DeliveryMode DeliveryMode
{
    get;
    set;
}
```

İleti üreticisi tarafından gönderilen iletiler için varsayılan teslim kipini alın ve ayarlayın.

Varsayılan teslim kipi aşağıdaki değerlerden birine sahiptir:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

Bir aracıya gerçek zamanlı bağlantı için değer `DeliveryMode.NonPersistent` olmalıdır.

Varsayılan değer, varsayılan değer `DeliveryMode.NonPersistent` olduğu bir aracıya gerçek zamanlı bağlantı dışında `DeliveryMode.Persistent` değeridir.

### **Özel durumlar:**

- `XMSException`

*Hedef-Hedef Al*

### **Arabirim:**

```
IDestination Destination
{
    get;
}
```

İleti üreticisine ilişkin hedefi alın.

### **Parametreler:**

Yok

### **Döndürülen:**

Hedef nesne. İleti üreticinin bir hedefi yoksa, yöntem boş değerli bir hedef nesne döndürür.

### Özel durumlar:

- XMSException

*DisableMsgKimliği-İleti Tanıtıcısı İşaretini Al ve Ayarla İşaretini Ayarla*

### Arabirim:

```
Boolean DisableMessageID
{
    get;
    set;
}
```

Alma uygulamasının ileti üreticisi tarafından gönderilen iletilere eklenmesi için ileti tanıtıcıları gerektirip gerektirmediğine ilişkin bir bilgi edinin ve alıcı uygulamanın ileti üreticisi tarafından gönderilen iletilere eklenmesi için ileti tanıtıcıları gerektirip gerektirmediğini belirtin.

Kuyruk yöneticisine ya da bir aracıya gerçek zamanlı bir bağlantı üzerinde yapılan bir bağlantıda bu işaret yoksayılr. Bir hizmet bütünleştirme veriyoluna yapılan bağlantıda, işaret onurlandırılır.

DisabledMsgTanıtıcısı şu değerlere sahiptir:

- `True`, bir alma uygulaması ileti üreticinin gönderdiği iletilerde ileti tanıtıcıları gerektirmiyorsa.
- `False`, alma uygulaması, ileti üreticisi tarafından gönderilen iletilere eklenecek ileti tanıtıcıları gerektiriyorsa.

### Özel durumlar:

- XMSException

*DisableMsgTS-Zaman Damgası İşaretini Al ve Ayarla İşaretini Ayarla*

### Arabirim:

```
Boolean DisableMessageTimestamp
{
    get;
    set;
}
```

Alma uygulamasının, ileti üreticisi tarafından gönderilen iletilere eklenecek zaman damgalarını gerektirip gerektirmediğine ilişkin bir bilgi edinin ve bir alma uygulamasının, ileti üreticisi tarafından gönderilen iletilere dahil edilecek zaman damgalarını gerektirip gerektirmediğini belirtin.

Bir aracıya gerçek zamanlı bir bağlantı üzerinde bu işaret yoksayılr. Kuyruk yöneticisine ya da hizmet bütünleştirme veriyoluna yönelik bir bağlantıda, işaret onurlandırılır.

TSDisableMsgDisableMsgTS aşağıdaki değerlere sahiptir:

- `True`, bir alma uygulamasının ileti üreticisi tarafından gönderilen iletilere dahil edilmesi için zaman damgalarını gerektirmemesi durumunda.
- `False`, alma uygulaması, ileti üreticisi tarafından gönderilen iletilere dahil edilecek zaman damgalarını gerektiriyorsa.

### Döndürülen:

### Özel durumlar:

- XMSException

## Öncelik-Varsayılan Önceliği Al ve Ayarla

### Arabirim:

```
Int32 Priority
{
    get;
    set;
}
```

İleti üreticisi tarafından gönderilen iletiler için varsayılan öncelik değerini alın ve ayarlayın.

Varsayılan ileti önceliğinin değeri, en düşük öncelik olan 0 aralığındaki bir tamsayıdır; en yüksek öncelik 9'a, en yüksek önceliğe sahip olur.

Bir aracıya gerçek zamanlı bir bağlantıda, bir iletinin önceliği yok sayılır.

### Özel durumlar:

- XMSEException

## TimeToLive-Varsayılan Zamanı Canlı Olarak Al ve Etkin Olarak Ayarla

### Arabirim:

```
Int64 TimeToLive
{
    get;
    set;
}
```

Bir iletinin süresi dolmadan önce var olan varsayılan süreyi elde edin ve ayarlayın.

Zaman, ileti üreticinin iletiyi gönderdiğinde ve milisaniye olarak yaşamanın varsayılan zamanı olduğu zamandan itibaren ölçülür. 0 değeri, bir iletinin hiçbir zaman süre bitimine geçmediği anlamına gelir.

Bir aracıya gerçek zamanlı bağlantı için bu değer her zaman 0 'tır.

### Özel durumlar:

- XMSEException

## Yöntemler

### İleti Üreticisini Kapat-Kapat

### Arabirim:

```
void Close();
```

İleti üreticisini kapatın.

Bir uygulama önceden kapatılmış bir ileti üreticisini kapatmaya çalışırsa, arama yoksayılr.

### Parametreler:

Yok

### Döndürülen:

Hükümsüz

### Özel durumlar:

- XMSEException

## Gönder-Gönder

### Arabirim:

```
void Send(IMessage msg) ;
```

İleti üreticisi yaratıldığında belirlenen hedefe bir ileti gönderir. İleti üreticisi varsayılan teslim kipini, önceliğini ve yaşama süresini kullanarak iletiyi gönderin.

### Parametreler:

#### **msg (giriş)**

İleti nesnesi.

### Döndürülen:

Hükümsüz

### Özel durumlar:

- XMSException
- MessageFormatKural Dışı Durumu
- InvalidDestinationKural Dışı Durumu

## Gönder-Gönder (teslim kipi belirtme, öncelik ve yaşama süresi)

### Arabirim:

```
void Send(IMessage msg,  
          DeliveryMode deliveryMode,  
          Int32 priority,  
          Int64 timeToLive);
```

İleti üreticisi yaratıldığında belirlenen hedefe bir ileti gönderir. Belirtilen teslim kipini, önceliğini ve yaşama süresini kullanarak iletiyi gönderin.

### Parametreler:

#### **msg (giriş)**

İleti nesnesi.

#### **deliveryMode (giriş)**

İletiyeye ilişkin teslim kipi; aşağıdaki değerlerden biri olmalıdır:

DeliveryMode.Persistent  
DeliveryMode.NonPersistent

Bir aracıya gerçek zamanlı bağlantı için değer DeliveryMode.NonPersistent olmalıdır.

#### **öncelik (giriş)**

İletinin önceliği. Değer, en düşük öncelik için 0 aralığında, en düşük öncelik için 9 değerine sahip bir tamsayı olabilir. Bir aracıya gerçek zamanlı bir bağlantı üzerinde değer yok sayılır.

#### **timeToLive (giriş)**

İletinin milisaniye cinsinden yaşaması için kullanılan süre. 0 değeri, iletinin hiçbir zaman süre bitimine geçmediği anlamına gelir. Bir aracıya gerçek zamanlı bağlantı için değer 0 olmalıdır.

### Döndürülen:

Hükümsüz

### Özel durumlar:

- XMSException
- MessageFormatKural Dışı Durumu

- InvalidDestinationKural Dışı Durumu
- IllegalStateKural Dışı Durumu

*Gönder-Gönder (belirtilen hedefe)*

**Arabirim:**

```
void Send(IDestination dest, IMessage msg) ;
```

İleti üreticisi yaratıldığında hedef belirtilmediyse, belirtilen hedefe bir ileti gönderir. İleti üreticisi varsayılan teslim kipini, önceliğini ve yaşama süresini kullanarak iletiyi gönderin.

Genellikle, bir ileti üreticisi yarattığınızda bir hedef belirtiyorsunuz, ancak yapmazsanız, ileti gönderdiğinizde hedef belirtmeniz gerekir.

**Parametreler:**

**en dest (giriş)**

Hedef nesne.

**msg (giriş)**

İleti nesnesi.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSException
- MessageFormatKural Dışı Durumu
- InvalidDestinationKural Dışı Durumu

*Gönder-Gönder (belirli bir hedefe bir teslim kipi, öncelik ve yaşama süresi belirtme)*

**Arabirim:**

```
void Send(IDestination dest,  
IMessage msg,  
DeliveryMode deliveryMode,  
Int32 priority,  
Int64 timeToLive) ;
```

İleti üreticisi yaratıldığında hedef belirtilmediyse, belirtilen hedefe bir ileti gönderir. Belirtilen teslim kipini, önceliğini ve yaşama süresini kullanarak iletiyi gönderin.

Genellikle, bir ileti üreticisi yarattığınızda bir hedef belirtiyorsunuz, ancak yapmazsanız, ileti gönderdiğinizde hedef belirtmeniz gerekir.

**Parametreler:**

**en dest (giriş)**

Hedef nesne.

**msg (giriş)**

İleti nesnesi.

**deliveryMode (giriş)**

İletiyeye ilişkin teslim kipi; aşağıdaki değerlerden biri olmalıdır:

DeliveryMode.Persistent

DeliveryMode.NonPersistent

Bir aracıya gerçek zamanlı bağlantı için değer DeliveryMode.NonPersistent olmalıdır.

### **öncelik (giriş)**

İletinin önceliği. Değer, en düşük öncelik için 0 aralığında, en düşük öncelik için 9 değerine sahip bir tamsayı olabilir. Bir aracıya gerçek zamanlı bir bağlantı üzerinde değer yok sayılır.

### **timeToLive (giriş)**

İletinin milisaniye cinsinden yaşaması için kullanılan süre. 0 değeri, iletinin hiçbir zaman süre bitimine geçmediği anlamına gelir. Bir aracıya gerçek zamanlı bağlantı için değer 0 olmalıdır.

### **Döndürülen:**

Hükümsüz

### **Özel durumlar:**

- XMSException
- MessageFormatKural Dışı Durumu
- InvalidDestinationKural Dışı Durumu
- IllegalStateKural Dışı Durumu

## **Edinilmiş özellikler ve yöntemler**

Aşağıdaki yöntemler [IPropertyContext](#) arabiriminden devralınır:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortÖzellik](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongÖzellik](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

## **IObjectMessage**

Bir nesne iletisi, gövdesi serileştirilmiş bir Java ya da .NET nesnesinden farklı olan bir iletidir.

### **Devralma sıradüzeni:**

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IObjectMessage
```

### **İlgili başvurular**

[Nesne iletileri](#)

Bir nesne iletisinin gövdesi serileştirilmiş bir Java ya da .NET nesnesi içeriyor.

## **.NET özellikler**

*Nesne-Nesneyi Bayt Olarak Al ve Ayarla*

### **Arabirim:**

```
System.Object Object
{
    get;
    set;
}

Byte[] GetObject();
```

Nesne iletisinin gövdelerini oluşturan nesneyi alın ve ayarlayın.

## Özel durumlar:

- XMSException
- MessageNotReadableException
- MessageEOFException
- MessageNotWritableException

## Edinilmiş özellikler ve yöntemler

Aşağıdaki özellikler IMessage arabiriminden devralınır:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRevered, JMSReplyTo, JMSTimestapp, JMSType, Properties

Aşağıdaki yöntemler IMessage arabiriminden devralınır:

clearBody, clearProperties, PropertyExists

Aşağıdaki yöntemler IPropertyContext arabiriminden devralınır:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortÖzellik, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongÖzelliği, SetObjectProperty, SetShortProperty, SetStringProperty

## IPropertyContext

IPropertyContext , özellikleri alan ve ayarlayan yöntemler içeren, soyut bir üst sınıftır. Bu yöntemler diğer sınıflar tarafından devralınır.

### Devralma sıradüzeni:

Yok

## Yöntemler

*GetBooleanözelliği-Boolean Özelliği Al*

### Arabirim:

```
Boolean GetBooleanProperty(String property_name);
```

Belirtilen ada sahip Boole özelliğinin değerini alın.

### Parametreler:

#### **özellik\_adi (giriş)**

Özelliğın adını sarmalayan bir String nesnesi.

### Döndürülen:

Özelliğın değeri.

### İş parçacığı bağlamı:

Alt sınıf tarafından belirlenir

## Özel durumlar:

- XMSException

## *GetByteözelliği-Byte Özelliği Al*

### **Arabirim:**

```
Byte    GetByteProperty(String property_name) ;  
Int16   GetSignedByteProperty(String property_name) ;
```

Ad ile tanımlanan bayt özelliğinin değerini alın.

### **Parametreler:**

#### **özellik\_adi (giriş)**

Özelliğın adını sarmalayan bir String nesnesi.

### **Döndürülen:**

Özelliğın deęeri.

### **İş parçacığı baęlamı:**

Alt sınıf tarafından belirlenir

### **Özel durumlar:**

- XMSException

## *GetBytesProperty-Get Byte Array özelliği*

### **Arabirim:**

```
Byte[]  GetBytesProperty(String property_name) ;
```

Ad ile tanımlanan bayt dizisi özelliğinin değerini alın.

### **Parametreler:**

#### **özellik\_adi (giriş)**

Özelliğın adını sarmalayan bir String nesnesi.

### **Döndürülen:**

Dizideki bayt sayısı.

### **İş parçacığı baęlamı:**

Alt sınıf tarafından belirlenir

### **Özel durumlar:**

- XMSException

## *GetCharözelliği-Karakter Özelliği Al*

### **Arabirim:**

```
Char    GetCharProperty(String property_name) ;
```

Ad ile tanımlanan 2 baytlık karakter özelliğinin değerini alın.

### **Parametreler:**

#### **özellik\_adi (giriş)**

Özelliğın adını sarmalayan bir String nesnesi.

### **Döndürülen:**

Özelliğın deęeri.

### **İş parçacığı baęlamı:**

Alt sınıf tarafından belirlenir



### Özel durumlar:

- XMSEException

*GetDoubleÖzelliđi-Double Precision Floating Point Özelliđi Al*

### Arabirim:

```
Double GetDoubleProperty(String property_name) ;
```

Ad ile tanımlanan çift duyarlıklı kayan nokta özelliđinin deđerini alın.

### Parametreler:

#### özelliđ\_adi (giriş)

Özelliđin adını sarmalayan bir String nesnesi.

### Döndürülen:

Özelliđin deđeri.

### İş parçacığı bađlamı:

Alt sınıf tarafından belirlenir

### Özel durumlar:

- XMSEException

*GetFloatÖzelliđi-Kayar Noktalı özelliđi Al*

### Arabirim:

```
Single GetFloatProperty(String property_name) ;
```

Ad ile tanımlanan kayan nokta özelliđinin deđerini alın.

### Parametreler:

#### özelliđ\_adi (giriş)

Özelliđin adını sarmalayan bir String nesnesi.

### Döndürülen:

Özelliđin deđeri.

### İş parçacığı bađlamı:

Alt sınıf tarafından belirlenir

### Özel durumlar:

- XMSEException

*GetIntÖzelliđi- GetIntÖzelliđi*

### Arabirim:

```
Int32 GetIntProperty(String property_name) ;
```

Ad ile tanımlanan tamsayı özelliđinin deđerini alın.

### Parametreler:

#### özelliđ\_adi (giriş)

Özelliđin adını sarmalayan bir String nesnesi.

**Döndürülen:**

Özelliğın deęeri.

**İş parçacığı baęlamı:**

Alt sınıf tarafından belirlenir

**Özel durumlar:**

- XMSException

*GetLongözellięi-Uzun Tamsayı Özellięi Al*

**Arabirim:**

```
Int64 GetLongProperty(String property_name) ;
```

Ad ile tanımlanan uzun tamsayı özelliğinin deęerini alın.

**Parametreler:****özellię\_adi (giriş)**

Özelliğın adını sarmalayan bir String nesnesi.

**Döndürülen:**

Özelliğın deęeri.

**İş parçacığı baęlamı:**

Alt sınıf tarafından belirlenir

**Özel durumlar:**

- XMSException

*GetObjectözellięi-Nesne özellięini al*

**Arabirim:**

```
Object GetObjectProperty( String property_name) ;
```

Ad ile tanımlanan özelliğın deęerini ve veri tipini alın.

**Parametreler:****özellię\_adi (giriş)**

Özelliğın adını sarmalayan bir String nesnesi.

**Döndürülen:**

Aşğıdaki nesne tiplerinden biri olan özelliğın deęeri:

Boolean  
Byte  
Byte[]  
Char  
Double  
Single  
Int32  
Int64  
Int16  
String

**İş parçacığı baęlamı:**

Alt sınıf tarafından belirlenir

### Özel durumlar:

- XMSEException

### *GetShortÖzelliği-Kısa Tamsayı Özelliği Al*

#### Arabirim:

```
Int16 GetShortProperty(String property_name) ;
```

Ad ile tanımlanan kısa tamsayı özelliğinin değerini alın.

#### Parametreler:

##### **özellik\_adi (giriş)**

Özelliğın adını sarmalayan bir String nesnesi.

#### Döndürülen:

Özelliğın deęeri.

#### İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

### Özel durumlar:

- XMSEException

### *GetStringÖzelliği- GetStringÖzelliği*

#### Arabirim:

```
String GetStringProperty(String property_name) ;
```

Ad ile tanımlanan dizgi özelliğinin deęerini alın.

#### Parametreler:

##### **özellik\_adi (giriş)**

Özelliğın adını sarmalayan bir String nesnesi.

#### Döndürülen:

Özelliğın deęeri olan dizeyi sarmalayan bir String nesnesi. Veri dönüştürme gerekiyorsa, bu deęer dönüştürmenin ardından dizgidir.

#### İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

### Özel durumlar:

- XMSEException

### *SetBooleanözelliği-Boole Özelliği Ayarla*

#### Arabirim:

```
void SetBooleanProperty( String property_name, Boolean value) ;
```

Ad ile tanımlanan Boole özelliğinin deęerini ayarlayın.

**Parametreler:****özellik\_adi (giriş)**

Özellik adını sarmalayan bir String nesnesi.

**değer (giriş)**

Özellik değeri.

**Döndürülen:**

Hükümsüz

**İş parçacığı bağlamı:**

Alt sınıf tarafından belirlenir

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*SetByteözelliği-Byte Özelliği Ayarla***Arabirim:**

```
void SetByteProperty( String property_name, Byte value) ;  
void SetSignedByteProperty( String property_name, Int16 value) ;
```

Ad tarafından tanımlanan byte özelliğinin değerini ayarlayın.

**Parametreler:****özellik\_adi (giriş)**

Özellik adını sarmalayan bir String nesnesi.

**değer (giriş)**

Özellik değeri.

**Döndürülen:**

Hükümsüz

**İş parçacığı bağlamı:**

Alt sınıf tarafından belirlenir

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*SetBytesProperty-Set Byte Array özelliği***Arabirim:**

```
void SetBytesProperty( String property_name, Byte[] value ) ;
```

Ad tarafından tanımlanan bayt dizisi özelliğinin değerini ayarlayın.

**Parametreler:****özellik\_adi (giriş)**

Özellik adını sarmalayan bir String nesnesi.

**değer (giriş)**

Bir bayt dizisi olan özelliğinin değeri.

**Döndürülen:**

Hükümsüz

**İş parçacığı bağlamı:**

Alt sınıf tarafından belirlenir

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*SetCharÖzelliği-Karakter Takımı Özelliği***Arabirim:**

```
void SetCharProperty( String property_name, Char value) ;
```

Ad tarafından tanımlanan 2 baytlık karakter özelliğinin değerini ayarlayın.

**Parametreler:****özellik\_adi (giriş)**

Özelliğın adını sarmalayan bir String nesnesi.

**değer (giriş)**

Özelliğın değeri.

**Döndürülen:**

Hükümsüz

**İş parçacığı bağlamı:**

Alt sınıf tarafından belirlenir

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*SetDoubleözelliği-Double Precision Floating Point özelliği ayarla***Arabirim:**

```
void SetDoubleProperty( String property_name, Double value) ;
```

Ad tarafından tanımlanan çift duyarlıklı kayan nokta özelliğinin değerini ayarlayın.

**Parametreler:****özellik\_adi (giriş)**

Özelliğın adını sarmalayan bir String nesnesi.

**değer (giriş)**

Özelliğın değeri.

**Döndürülen:**

Hükümsüz

**İş parçacığı bağlamı:**

Alt sınıf tarafından belirlenir

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

## *SetFloatÖzelliđi-Deđiřken Nokta özelliđini ayarla*

### **Arabirim:**

```
void SetFloatProperty( String property_name, Single value) ;
```

Ad tarafından tanımlanan kayar noktalı özellik deđerini ayarlayın.

### **Parametreler:**

#### **özelliđ\_adi (giriř)**

Özelliđin adını sarmalayan bir String nesnesi.

#### **deđer (giriř)**

Özelliđin deđeri.

### **Döndürülen:**

Hükümsüz

### **İř parçacığı bađlamı:**

Alt sınıf tarafından belirlenir

### **Özel durumlar:**

- XMSException
- MessageNotWritableException

## *SetIntÖzelliđi-Tamsayı Özelliđi Ayarla*

### **Arabirim:**

```
void SetIntProperty( String property_name, Int32 value) ;
```

Ad ile tanımlanan tamsayı özelliđinin deđerini ayarlayın.

### **Parametreler:**

#### **özelliđ\_adi (giriř)**

Özelliđin adını sarmalayan bir String nesnesi.

#### **deđer (giriř)**

Özelliđin deđeri.

### **Döndürülen:**

Hükümsüz

### **İř parçacığı bađlamı:**

Alt sınıf tarafından belirlenir

### **Özel durumlar:**

- XMSException
- MessageNotWritableException

## *SetLongÖzellik-Uzun Tamsayı Özelliđi Ayarla*

### **Arabirim:**

```
void SetLongProperty( String property_name, Int64 value) ;
```

Ad tarafından tanımlanan uzun tamsayı özelliđinin deđerini ayarlayın.

**Parametreler:****özellik\_adi (giriş)**

Özelliğin adını sarmalayan bir String nesnesi.

**değer (giriş)**

Özelliğin değeri.

**Döndürülen:**

Hükümsüz

**İş parçacığı bağlamı:**

Alt sınıf tarafından belirlenir

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*SetObjectözelliği-Nesne özelliğini ayarla*

**Arabirim:**

```
void SetObjectProperty( String property_name, Object value) ;
```

Ad tarafından tanımlanan bir özelliğin değerini ve veri tipini ayarlayın.

**Parametreler:****özellik\_adi (giriş)**

Özelliğin adını sarmalayan bir String nesnesi.

**objectType (giriş)**

Aşağıdaki nesne tiplerinden biri olması gereken özelliğin değeri:

Boolean  
Byte  
Byte[]  
Char  
Double  
Single  
Int32  
Int64  
Int16  
String

**değer (giriş)**

Özelliğin bayt dizisi olarak değeri.

**uzunluk (giriş)**

Dizideki bayt sayısı.

**Döndürülen:**

Hükümsüz

**İş parçacığı bağlamı:**

Alt sınıf tarafından belirlenir

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

## SetShortÖzellik-Kısa Tamsayı Özelliği Ayarla

### Arabirim:

```
void SetShortProperty( String property_name, Int16 value) ;
```

Ad tarafından tanımlanan kısa tamsayı özelliğinin değerini ayarlayın.

### Parametreler:

#### özellik\_adi (giriş)

Özelliğın adını sarmalayan bir String nesnesi.

#### değer (giriş)

Özelliğın deęeri.

### Döndürülen:

Hükümsüz

### İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

### Özel durumlar:

- XMSException
- MessageNotWritableException

## SetStringÖzelliği-Dizgi Özelliğini Ayarla

### Arabirim:

```
void SetStringProperty( String property_name, String value);
```

Ad ile tanımlanan dizgi özelliğinin deęerini ayarlayın.

### Parametreler:

#### özellik\_adi (giriş)

Özelliğın adını sarmalayan bir String nesnesi.

#### değer (giriş)

Özelliğın deęeri olan dizgeyi sarmalayan bir String nesnesi.

### Döndürülen:

Hükümsüz

### İş parçacığı baęlamı:

Alt sınıf tarafından belirlenir

### Özel durumlar:

- XMSException
- MessageNotWritableException

## IQueueBrowser

Bir uygulama, kuyruęu kaldırmadan kuyruklardaki iletilere göz atmak için bir kuyruk tarayıcısı kullanır.

### Devralma sıradüzeni:

```
IBM.XMS.IPropertyContext
System.Collections.IEnumerable
|
+---- IBM.XMS.IQueueBrowser
```



## **.NET Özellikler**

### *MessageSelector -İleti Seçiciyi Al*

#### **Arabirim:**

```
String MessageSelector
{
    get;
}
```

Kuyruk tarayıcısı için ileti seçiciyi alın.

İleti seçici, ileti seçici ifadesini sarmalayan bir String nesnesidir. Veri dönüştürme gerekiyorsa, bu değer dönüştürmeden sonraki ileti seçici ifadesidir. Kuyruk tarayıcısının bir ileti seçicisi yoksa, yöntem boş bir String nesnesi döndürür.

#### **Özel durumlar:**

- XMSEException

### *Kuyruk-Kuyruk Alma*

#### **Arabirim:**

```
IDestination Queue
{
    get;
}
```

Kuyruk tarayıcısıyla ilişkili kuyruğu, kuyruğu gösteren hedef nesne olarak alın.

#### **Özel durumlar:**

- XMSEException

## **Yöntemler**

### *Kapat-Kuyruk Tarayıcısını Kapat*

#### **Arabirim:**

```
void Close();
```

Kuyruk tarayıcısını kapatın.

Bir uygulama önceden kapatılmış bir kuyruk tarayıcısını kapatmaya çalışırsa, arama yoksayılır.

#### **Parametreler:**

Yok

#### **Döndürülen:**

Hükümsüz

#### **Özel durumlar:**

- XMSEException

**Arabirim:**

```
IEnumerator GetEnumerator();
```

Kuyruklardaki iletilerin bir listesini alın.

Yöntem, İleti nesnelere listesini sarmalayan bir sıralı değer listeleme döndürür. İleti nesnelere sırası, iletilerin kuyruktan alınacağı sıralama düzeniyle aynıdır. Daha sonra, uygulama sırayla her iletiye göz atmak için sıralı değer numaracıyı kullanabilir.

İletiler kuyruğa yerleştirildiği ve kuyruktan kaldırıldığı için, sıralı değer listeleme dinamik olarak güncellenir. Uygulamanın kuyruktan sonraki iletiye göz atmak için `IEnumerator.MoveNext()` çağrısının her çağrısında, ileti kuyruğun yürürlükteki içeriğini yansıtır.

Bir uygulama bu yöntemi bir kuyruk tarayıcısı için bir kereden fazla çağırırsa, her çağrı yeni bir sıralı değer listeleme döndürür. Bu nedenle, uygulama bir kuyruktaki iletilere göz atmak ve kuyruğun içinde birden çok konum korumak için birden çok sıralı değer listeleme kullanabilir.

**Parametreler:**

Yok

**Döndürülen:**

Yineleyici nesnesi.

**Özel durumlar:**

- `XMSEException`

***Edinilmiş özellikler ve yöntemler***

Aşağıdaki yöntemler `IPropertyContext` arabiriminden devralınır:

`GetBooleanProperty`, `GetByteProperty`, `GetBytesProperty`, `GetCharProperty`, `GetDoubleProperty`, `GetFloatProperty`, `GetIntProperty`, `GetLongProperty`, `GetObjectProperty`, `GetShortÖzellik`, `GetStringProperty`, `SetBooleanProperty`, `SetByteProperty`, `SetBytesProperty`, `SetCharProperty`, `SetDoubleProperty`, `SetFloatProperty`, `SetIntProperty`, `SetLongÖzelliği`, `SetObjectProperty`, `SetShortProperty`, `SetStringProperty`

**İstekte Bulunan**

Bir uygulama, istek iletisini göndermek için istekte bulunan bir programı kullanır ve ardından yanıtı bekler ve alır.

**Devralma sıradüzeni:**

Yok

***Oluşturucular***

*İstekte Bulunanın-İstekte Bulunanın Yaratılması*

**Arabirim:**

```
Requestor(ISession sess, IDestination dest);
```

İstekte bulunanın oluşturulması.

**Parametreler:**

**takıntı (giriş)**

Bir Oturum nesnesi. Oturum hareket edilmemelidir ve aşağıdaki alındı bildirim kiplerinden birine sahip olmalıdır:

AcknowledgeMode.AutoAcknowledge  
AcknowledgeMode.DupsOkAcknowledge

### **en dest (giriş)**

Uygulamanın istek iletileri gönderebileceği hedefi gösteren hedef nesne.

### **İş parçacığı bağlamı:**

İstekçiyle ilişkili oturum

### **Özel durumlar:**

- XMSEException

## **Yöntemler**

### *Yakın-Kapanış İstek Sahibi*

#### **Arabirim:**

```
void Close();
```

İstekte bulunanın kapatılması.

Bir uygulama önceden kapatılmış bir istekte bulunmayı kapatmaya çalışırsa, arama yoksayıdır.

**Not:** Bir uygulama bir istek isteğini kapattığında, ilişkili oturum da kapanmaz. Bu bakımdan XMS , JMS ' ye kıyasla farklı davranır.

#### **Parametreler:**

Yok

#### **Döndürülen:**

Hükümsüz

### **İş parçacığı bağlamı:**

Herhangi

### **Özel durumlar:**

- XMSEException

### *İstek-İstek Yanıtı*

#### **Arabirim:**

```
IMessage Request(IMessage requestMessage);
```

Bir istek iletisi gönderin ve istek iletisini alan uygulamadan bir yanıt almak ve almak için bekleyin.

Bu yöntem öbeklerine bir yanıt alınıncaya kadar ya da oturum sona erinceye kadar bir çağrı (hangisi daha erken olursa olsun).

#### **Parametreler:**

### **requestMessage (giriş)**

İleti nesnesi, istek iletisini sarsalıyor.

#### **Döndürülen:**

İleti nesnesine ilişkin bir işaretçi, yanıt iletisini sarsalıyor.

### **İş parçacığı bağlamı:**

İstekçiyle ilişkili oturum

### **Özel durumlar:**

- XMSEException

## ResourceAllocation Özel Durumu

### Devralma sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.ResourceAllocationException
```

XMS throws this exception if XMS cannot allocate the resources required by a method.

### **Edinilmiş özellikler ve yöntemler**

Aşağıdaki yöntemler, [XMSEException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

## SecurityException

### Devralma sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.SecurityException
```

XMS throws this exception if the user identifier and password provided to authenticate an application are rejected. XMS, bir yetki denetimi başarısızlıkla sonuçlanırsa ve bir yöntemin tamamlanmasını önlediğinde bu kural dışı durumu da atar.

### **Edinilmiş özellikler ve yöntemler**

Aşağıdaki yöntemler, [XMSEException](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

## ISession

Oturum, ileti göndermek ve almak için tek bir iş parçacıklı bağlamdır.

### Devralma sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.ISession
```

Bir Oturum nesnesinin XMS tanımlı özelliklerinin bir listesi için bkz. "[Oturum Özellikleri](#)" sayfa 188.

## **.NET özellikleri**

*AcknowledgeMode -Onay Kipini Al*

### Arabirim:

```
AcknowledgeMode AcknowledgeMode
{
    get;
}
```

Oturum için alındı bildirim modunu alın.

Onay kipi, oturum oluşturulduğunda belirtilir.

Oturum hareket edilmediği sürece, alındı bildirimi kipi aşağıdaki değerlerden biridir:

```
AcknowledgeMode.AutoAcknowledge  
AcknowledgeMode.ClientAcknowledge  
AcknowledgeMode.DupsOkAcknowledge
```

Alındı bildirimi kipleri hakkında daha fazla bilgi için bkz. [“İleti alındı bildirimi” sayfa 24.](#)

Hareket eden bir oturumda alındı bildirimi kipi yoktur. Oturum çevrilebiliyorsa, yöntem `AcknowledgeMode.SessionTransacted` değerini döndürür.

#### Özel durumlar:

- `XMSEException`

*Hareket Eden-Hareket Edip Etmeyeceğini Belirleyin*

#### Arabirim:

```
Boolean Transacted  
{  
    get;  
}
```

Oturumun hareket edip etmeyeceğini belirleyin.

Transacted (derleme sonucu):

- `True`, oturum çevrilebiliyorsa.
- `False`, oturum hareket edilmezse.

Bir aracıya gerçek zamanlı bağlantı için, yöntem her zaman `False` değerini döndürür.

#### Özel durumlar:

- `XMSEException`

## Yöntemler

*Oturumu Kapat-Oturumu Kapat*

#### Arabirim:

```
void Close();
```

Oturumu kapatın. Oturum hareketliyse, devam etmekte olan herhangi bir hareket geriye işlenir.

Bir uygulama önceden kapatılmış bir oturumu kapatmaya çalışırsa, arama yoksayıdır.

#### Parametreler:

Yok

#### Döndürülen:

Hükümsüz

#### İş parçacığı bağlamı:

Herhangi

#### Özel durumlar:

- `XMSEException`

## *Kesinleştir-Kesinleştir*

### **Arabirim:**

```
void Commit();
```

Geçerli harekette işlenen tüm iletileri kesinleştir.

Oturum, hareket eden bir oturum olmalıdır.

### **Parametreler:**

Yok

### **Döndürülen:**

Hükümsüz

### **Özel durumlar:**

- XMSEException
- IllegalStateKural Dışı Durumu
- TransactionRolledBackException

## *CreateBrowser -Kuyruk Tarayıcısı Yarat*

### **Arabirim:**

```
IQueueBrowser CreateBrowser(IDestination queue) ;
```

Belirlenen kuyruk için bir kuyruk tarayıcısı yaratın.

### **Parametreler:**

#### **kuyruk (giriş)**

Kuyruğu gösteren bir hedef nesne.

### **Döndürülen:**

QueueBrowser nesnesi.

### **Özel durumlar:**

- XMSEException
- InvalidDestinationKural Dışı Durumu

## *CreateBrowser -Kuyruk Tarayıcısı Yarat (ileti seçici ile)*

### **Arabirim:**

```
IQueueBrowser CreateBrowser(IDestination queue, String selector) ;
```

İleti seçiciyi kullanarak, belirlenen kuyruk için bir kuyruk tarayıcısı yaratın.

### **Parametreler:**

#### **kuyruk (giriş)**

Kuyruğu gösteren bir hedef nesne.

#### **seçici (giriş)**

Bir ileti seçici ifadesini sarmalayan bir String nesnesi. Kuyruk tarayıcıya, yalnızca, ileti seçici ifadesiyle eşleşen özelliklere sahip iletiler teslim edilir.

Boş Dizi nesnesi, kuyruk tarayıcısı için herhangi bir ileti seçiciye sahip olmadığı anlamına gelir.

**Döndürülen:**

QueueBrowser nesnesi.

**Özel durumlar:**

- XMSException
- InvalidDestinationKural Dışı Durumu
- InvalidSelectorKural Dışı Durumu

*CreateBytesMessage-Bytes İletisi Oluştur***Arabirim:**

```
IBytesMessage CreateBytesMessage();
```

Bir bayt iletisi oluşturun.

**Parametreler:**

Yok

**Döndürülen:**

BytesMessage nesnesi.

**Özel durumlar:**

- XMSException
- IllegalStateKural Dışı Durumu (Oturum kapatılmış)

*CreateConsumer -Tüketici yarat***Arabirim:**

```
IMessageConsumer CreateConsumer(IDestination dest) ;
```

Belirtilen hedef için bir ileti tüketicisi oluşturun.

**Parametreler:****en dest (giriş)**

Hedef nesne.

**Döndürülen:**

MessageConsumer nesnesi.

**Özel durumlar:**

- XMSException
- InvalidDestinationKural Dışı Durumu

*CreateConsumer -Tüketici yarat (ileti seçici ile)***Arabirim:**

```
IMessageConsumer CreateConsumer(IDestination dest,  
String selector) ;
```

İleti seçiciyi kullanarak, belirtilen hedef için ileti tüketicisi yaratın.

**Parametreler:****en dest (giriş)**

Hedef nesne.

**seçici (giriş)**

Bir ileti seçici ifadesini sarmalayan bir String nesnesi. İleti tüketicisi, yalnızca, ileti seçici ifadesiyle eşleşen özelliklere sahip iletiler iletime teslim edilir.

Boş Dizi nesnesi, ileti tüketicisi için ileti seçicisinin olmadığı anlamına gelir.

**Döndürülen:**

MessageConsumer nesnesi.

**Özel durumlar:**

- XMSException
- InvalidDestinationKural Dışı Durumu
- InvalidSelectorKural Dışı Durumu

*CreateConsumer -Tüketici yarat (ileti seçici ve yerel ileti işareti ile)*

**Arabirim:**

```
IMessageConsumer CreateConsumer(IDestination dest,  
                                String selector,  
                                Boolean noLocal) ;
```

İleti seçiciyi kullanarak belirtilen hedef için bir ileti tüketicisi yaratın ve hedef, ileti tüketicisinin kendi bağlantısı tarafından yayınlanan iletileri alıp almadığını belirten bir konudur.

**Parametreler:****en dest (giriş)**

Hedef nesne.

**seçici (giriş)**

Bir ileti seçici ifadesini sarmalayan bir String nesnesi. İleti tüketicisi, yalnızca, ileti seçici ifadesiyle eşleşen özelliklere sahip iletiler iletime teslim edilir.

Boş Dizi nesnesi, ileti tüketicisi için ileti seçicisinin olmadığı anlamına gelir.

**noLocal (giriş)**

True değeri, ileti tüketicisinin kendi bağlantısı tarafından yayınlanan iletileri almadığı anlamına gelir. False değeri, ileti tüketicisi tarafından yayınlanan iletileri kendi bağlantısı tarafından alacak anlamına gelir. Varsayılan değer, False'tur.

**Döndürülen:**

MessageConsumer nesnesi.

**Özel durumlar:**

- XMSException
- InvalidDestinationKural Dışı Durumu
- InvalidSelectorKural Dışı Durumu

*CreateDurableAbonesi-Durable Abonesi Oluştur*

**Arabirim:**

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                         String subscription) ;
```



Belirtilen konu için dayanıklı bir abone yaratın.

Bu yöntem, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

Kalıcı aboneler hakkında daha fazla bilgi için bkz. [“Sürekli aboneler” sayfa 32.](#)

#### **Parametreler:**

##### **en dest (giriş)**

Konuyu simgeleyen bir hedef nesne. Konu geçici bir konu olmamalıdır.

##### **abonelik (giriş)**

Bir dizgi nesnesi, kalıcı aboneliği tanımlayan bir adı sarkıyor. Adın, bağlantı için istemci tanıtıcısı içinde benzersiz olması gerekir.

#### **Döndürülen:**

Dayanıklı aboneyi gösteren MessageConsumer nesnesi.

#### **Özel durumlar:**

- XMSEException
- InvalidDestinationKural Dışı Durumu

*CreateDurableAbonesi-Durable Abonesi Oluştur (ileti seçici ve yerel ileti işareti ile)*

#### **Arabirim:**

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                           String subscription,  
                                           String selector,  
                                           Boolean noLocal) ;
```

İleti seçiciyi kullanarak ve kalıcı abonenin kendi bağlantısı tarafından yayınlanan iletileri alıp almadığını belirterek, belirtilen konu için dayanıklı bir abone yaratın.

Bu yöntem, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

Kalıcı aboneler hakkında daha fazla bilgi için bkz. [“Sürekli aboneler” sayfa 32.](#)

#### **Parametreler:**

##### **en dest (giriş)**

Konuyu simgeleyen bir hedef nesne. Konu geçici bir konu olmamalıdır.

##### **abonelik (giriş)**

Bir dizgi nesnesi, kalıcı aboneliği tanımlayan bir adı sarkıyor. Adın, bağlantı için istemci tanıtıcısı içinde benzersiz olması gerekir.

##### **seçici (giriş)**

Bir ileti seçici ifadesini sarmalayan bir String nesnesi. Yalnızca, ileti seçici ifadesiyle eşleşen özelliklere sahip iletiler dayanıklı aboneye teslim edilir.

Boş Dizgi nesnesi, dayanıklı abone için ileti seçiciye sahip olmadığı anlamına gelir.

##### **noLocal (giriş)**

True değeri, kalıcı abonenin kendi bağlantısı tarafından yayınlanan iletileri almadığı anlamına gelir. False değeri, kalıcı abonenin kendi bağlantısı tarafından yayınlanan iletileri alacağı anlamına gelir. Varsayılan değer, False'tur.

#### **Döndürülen:**

Dayanıklı aboneyi gösteren MessageConsumer nesnesi.

#### **Özel durumlar:**

- XMSEException
- InvalidDestinationKural Dışı Durumu
- InvalidSelectorKural Dışı Durumu

### *CreateMapIletisi-Eşlem İletisi Yarat*

#### **Arabirim:**

```
IMapMessage CreateMapMessage();
```

Bir eşlem iletisi yaratın.

#### **Parametreler:**

Yok

#### **Döndürülen:**

MapMessage nesnesi.

#### **Özel durumlar:**

- XMSException
- IllegalStateKural Dışı Durumu (Oturum kapatılmış)

### *CreateMessage -İleti yarat*

#### **Arabirim:**

```
IMessage CreateMessage();
```

Vücuduna sahip olmayan bir ileti oluşturun.

#### **Parametreler:**

Yok

#### **Döndürülen:**

İleti nesnesi.

#### **Özel durumlar:**

- XMSException
- IllegalStateKural Dışı Durumu (Oturum kapatılmış)

### *CreateObjectİletisi-Nesne İletisi Oluştur*

#### **Arabirim:**

```
IObjectMessage CreateObjectMessage();
```

Bir nesne iletisi oluşturun.

#### **Parametreler:**

Yok

#### **Döndürülen:**

ObjectMessage nesnesi.

#### **Özel durumlar:**

- XMSException
- IllegalStateKural Dışı Durumu (Oturum kapatılmış)

## *CreateProducer -Üretici yarat*

### **Arabirim:**

```
IMessageProducer CreateProducer(IDestination dest) ;
```

İletileri belirtilen hedefe göndermek için bir ileti üreticisi yaratın.

### **Parametreler:**

#### **en dest (giriş)**

Hedef nesne.

Boş değerli bir hedef nesne belirtirseniz, ileti üreticisi hedef nesne olmadan yaratılır. Bu durumda, bir ileti göndermek için, uygulamanın ileti üreticisini her kullanımında bir hedef belirtmesi gerekir.

### **Döndürülen:**

MessageProducer nesnesi.

### **Özel durumlar:**

- XMSException
- InvalidDestinationKural Dışı Durumu

## *CreateQueue -Kuyruk Yarat*

### **Arabirim:**

```
IDestination CreateQueue(String queue) ;
```

İleti alışverişi sunucusunda bir kuyruğu temsil etmek için bir hedef nesne yaratın.

Bu yöntem, ileti sistemi sunucusunda kuyruğu yaratmaz. Bir uygulamanın bu yöntemi çağırabilmesi için önce kuyruğu oluşturmanız gerekir.

### **Parametreler:**

#### **kuyruk (giriş)**

Bir String nesnesi, kuyruğun adını sarsalıyor ya da kuyruğu tanımlayan bir URI (uniform resource identifier; bir tip kaynak tanıtıcısını (URI) sarsalıyor).

### **Döndürülen:**

Kuyruğu gösteren hedef nesne.

### **Özel durumlar:**

- XMSException

## *CreateStreamIletisi-Akış İletisi Oluştur*

### **Arabirim:**

```
IStreamMessage CreateStreamMessage();
```

Bir akış iletisi oluşturun.

### **Parametreler:**

Yok

### **Döndürülen:**

StreamMessage nesnesi.

**Özel durumlar:**

- XMSException
- XMS\_ILLEGAL\_STATE\_EXCEPTION

*CreateTemporaryKuyruğu-Geçici Kuyruk Yarat***Arabirim:**

```
IDestination CreateTemporaryQueue() ;
```

Geçici bir kuyruk yaratın.

Geçici kuyruğun kapsamı bağlantıdır. Yalnızca, bağlantı tarafından yaratılan oturumlar geçici kuyruğu kullanabilir.

Geçici kuyruk açık bir şekilde silininceye ya da bağlantı sona erinceye kadar kalır; hangisi daha erken olursa olsun.

Geçici kuyruklara ilişkin daha fazla bilgi için bkz. [“Geçici hedefler” sayfa 30](#).

**Parametreler:**

Yok

**Döndürülen:**

Geçici kuyruğu gösteren Hedef nesnesi.

**Özel durumlar:**

- XMSException

*CreateTemporaryKonusu-Geçici Konu Yarat***Arabirim:**

```
IDestination CreateTemporaryTopic() ;
```

Geçici bir konu oluşturun.

Geçici konunun kapsamı, bağlantıdır. Yalnızca, bağlantı tarafından yaratılan oturumlar geçici konuyu kullanabilir.

Geçici konu, açık bir şekilde silininceye ya da bağlantı sona erinceye kadar devam eder.

Geçici konularla ilgili daha fazla bilgi için bkz. [“Geçici hedefler” sayfa 30](#).

**Parametreler:**

Yok

**Döndürülen:**

Geçici konuyu simgeleyen Hedef nesnesi.

**Özel durumlar:**

- XMSException

*CreateTextİletisi-Metin İletisi Oluştur***Arabirim:**

```
ITextMessage CreateTextMessage();
```

Boş gövdesi içeren bir metin iletisi oluşturun.

**Parametreler:**

Yok

**Döndürülen:**

TextMessage nesnesi.

**Özel durumlar:**

- XMSEException

*CreateTextİletisi-Metin İletisi Oluştur (kullanıma hazırlandı)*

**Arabirim:**

```
ITextMessage CreateTextMessage(String initialValue);
```

Cesedi belirtilen metinle ilk kullanıma hazırlanmış bir metin iletisi oluşturun.

**Parametreler:**

**initialValue (giriş)**

Metin iletisinin gövesini kullanıma hazırlamak için metni sarmalayan bir String nesnesi.

Yok

**Döndürülen:**

TextMessage nesnesi.

**Özel durumlar:**

- XMSEException

*CreateTopic -Konu Oluştur*

**Arabirim:**

```
IDestination CreateTopic(String topic) ;
```

Bir konuyu temsil etmek için bir hedef nesne yaratın.

**Parametreler:**

**konu (giriş)**

Bir dizgi nesnesi, konunun adını sarsalıyor ya da konuyu tanımlayan bir URI (uniform resource identifier; bir tip kaynak tanıtıcısını (URI) sarsalıyor.

**Döndürülen:**

Konuyu simgeleyen Hedef nesnesi.

**Özel durumlar:**

- XMSEException

*Kurtar-Yeniden Kurtar*

**Arabirim:**

```
void Recover();
```

Oturumu kurtarın. İleti teslimi durdurulur ve tanınmayan en eski iletiyle yeniden başlatılır.

Oturum, hareket eden bir oturum olmamalıdır.

Bir oturumu kurtarma hakkında daha fazla bilgi için bkz. [“İleti alındı bildirimi” sayfa 24.](#)

**Parametreler:**

Yok

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- IllegalStateKural Dışı Durumu

*Geridönüş-Geri Alma*

**Arabirim:**

```
void Rollback();
```

Geçerli harekette işlenen tüm iletileri geri alma.

Oturum, hareket eden bir oturum olmalıdır.

**Parametreler:**

Yok

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- IllegalStateKural Dışı Durumu

*Aboneliği Kaldır-Aboneliği Kaldır*

**Arabirim:**

```
void Unsubscribe(String subscription);
```

Dayanıklı bir aboneliği silin. İleti alışverişi sunucusu, sürdürmekte olduğu dayanıklı aboneliğin kaydını siler ve kalıcı aboneye başka ileti göndermez.

Bir uygulama, aşağıdaki koşullardan herhangi birinde kalıcı bir aboneliği silemez:

- Kalıcı abonelik için etkin bir ileti tüketicisi varken
- Tüketilen bir ileti, bekleyen bir hareketin bir parçası olmakla birlikte
- Tüketilen bir ileti onaylanmazken

Bu yöntem, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

**Parametreler:**

**abonelik (giriş)**

Kalıcı aboneliği tanımlayan adı sarmalayan bir String nesnesi.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException

- InvalidDestinationKural Dışı Durumu
- IllegalStateKural Dışı Durumu

## Edinilmiş özellikler ve yöntemler

Aşağıdaki yöntemler [IPropertyContext](#) arabiriminden devralınır:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortÖzellik](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongÖzelliği](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

## IStreamMessage

Akış iletisi, her değer ilişkili bir veri tipine sahip olduğu bir değerler akışından oluşan bir iletidir. Gövdenin içeriği sırayla yazılır ve okunur.

### Devralma sıradüzeni:

```

IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IStreamMessage

```

Bir uygulama iletisi akışından bir değer okuduğunda, değer XMS tarafından başka bir veri tipine dönüştürülebilir. Bu örtük dönüştürme biçimiyle ilgili daha fazla bilgi için bkz. [“Akım iletileri” sayfa 75.](#)

### İlgili başvurular

[Akım iletileri](#)

Bir akış iletisinin gövdesi, her değer ilişkili bir veri tipine sahip olduğu bir değerler akışı içerir.

### Yöntemler

*ReadBoolean - Boolean değerini okuyun*

#### Arabirim:

```
Boolean ReadBoolean();
```

İleti akışından bir Boole değeri okuyun.

#### Parametreler:

Yok

#### Döndürülen:

Okunan boole değeri.

#### Özel durumlar:

- XMSException
- MessageNotReadableException
- MessageEOFException

## *ReadByte -Byte Okuma Deęeri*

### **Arabirim:**

```
Int16  ReadSignedByte();  
Byte   ReadByte();
```

İleti akışından imzalı 8 bitlik bir tamsayı okuyun.

### **Parametreler:**

Yok

### **Döndürülen:**

Okunan bayt.

### **Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

## *ReadBytes -Okuma Baytları*

### **Arabirim:**

```
Int32  ReadBytes(Byte[] array);
```

İleti akışından bir bayt dizisi okuyun.

### **Parametreler:**

#### **dizi (giriş)**

Okunan bayt dizisine ve byte cinsinden arabelleğin uzunluğuna sahip arabellek.

Dizideki bayt sayısı arabelleğin uzunluğundan az ya da bu uzunluğa eşitse, tüm dizi arabelleğe okunuyor. Dizideki bayt sayısı arabelleğin uzunluğundan büyükse, arabellek dizinin bir bölümüyle doldurulur ve bir iç imleç sonraki baytın okunabilmesinin konumunu gösterir. A subsequent call to readBytes() reads bytes from the array starting from the current position of the cursor.

Girişte boş değerli bir gösterge belirtirseniz, arama, okumadan byte dizilerinin üzerine atlar.

### **Döndürülen:**

Arabelleğe okunan bayt sayısı. Arabellek kısmen doldurulsa, değer, okunmak üzere kalan dizide başka byte kalmadığını gösteren arabellek uzunluğundan az olur. Aramadan önce diziden okunmak üzere kalan herhangi bir byte yoksa, değer XMSC\_END\_OF\_BYTEARRAY olur.

Giriş sırasında boş değerli bir gösterge belirtirseniz, yöntem değer döndürmez.

### **Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

## *ReadChar -Okuma karakteri*

### **Arabirim:**

```
Char   ReadChar();
```



İleti akışından 2 byte 'lık bir karakter okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan karakter.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadDouble -Çift Duyarlıklı Yüzer Nokta Numarasını Oku*

**Arabirim:**

```
Double ReadDouble();
```

İleti akışından, 8 baytlık çift duyarlıklı kayan noktalı sayı okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan çift duyarlıklı kayan noktalı sayı.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadFloat -Kayan Nokta Numarasını okuyun*

**Arabirim:**

```
Single ReadFloat();
```

İleti akışından 4 baytlık bir kayan noktalı sayı okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan kayan noktalı sayı.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

### *ReadInt -Tamsayı Oku*

#### **Arabirim:**

```
Int32 ReadInt();
```

İleti akışından, imzalı 32 bitlik bir tamsayı okuyun.

#### **Parametreler:**

Yok

#### **Döndürülen:**

Okunan tamsayı.

#### **Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

### *ReadLong -Uzun Tamsayı Oku*

#### **Arabirim:**

```
Int64 ReadLong();
```

İleti akışından imzalı 64 bitlik bir tamsayı okuyun.

#### **Parametreler:**

Yok

#### **Döndürülen:**

Okunan uzun tamsayı.

#### **Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

### *ReadObject -Nesneyi Oku*

#### **Arabirim:**

```
Object ReadObject();
```

İleti akışından bir değer okuyun ve veri tipini döndürün.

#### **Parametreler:**

Yok

#### **Döndürülen:**

Aşağıdaki nesne tiplerinden biri olan değer:

Boolean  
Byte  
Byte[]  
Char  
Double

Single  
Int32  
Int64  
Int16  
String

**Özel durumlar:**

XMSEException

*ReadShort -Kısa Tamsayı Oku*

**Arabirim:**

```
Int16 ReadShort();
```

İleti akışından imzalı 16 bitlik bir tamsayı okuyun.

**Parametreler:**

Yok

**Döndürülen:**

Okunan kısa tamsayı.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*ReadString -Dize Oku*

**Arabirim:**

```
String ReadString();
```

İleti akışından bir dizgi okuyun. Gerekliyse, XMS dizgideki karakterleri yerel kod sayfasına dönüştürür.

**Parametreler:**

Yok

**Döndürülen:**

Okunan dizeyi sarmalayan bir String nesnesi. Veri dönüştürme gerekiyorsa, bu dönüştürme işleminden sonra dizgidir.

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*Sıfırla-İlk Duruma Getir*

**Arabirim:**

```
void Reset();
```

İletinin gövdeini salt okunur kipi getirin ve imleci ileti akışının başlangıcındaki yerini yeniden konumlayın.

**Parametreler:**

Yok

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotReadableException
- MessageEOFException

*WriteBoolean -Yazma Boole Değeri*

**Arabirim:**

```
void WriteBoolean(Boolean value);
```

İleti akışa bir boole değeri yazın.

**Parametreler:**

**değer (giriş)**

Yazılacak boole değeri.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteByte -Yazma Byte 'ı*

**Arabirim:**

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

İleti akışa bir bayt yazın.

**Parametreler:**

**değer (giriş)**

Yazılacak bayt.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

## *WriteBytes -Yazma Baytları*

### **Arabirim:**

```
void WriteBytes(Byte[] value);
```

İleti akışına bir bayt dizisi yazın.

### **Parametreler:**

#### **değer (giriş)**

Yazılacak bayt dizisi.

#### **uzunluk (giriş)**

Dizideki bayt sayısı.

### **Döndürülen:**

Hükümsüz

### **Özel durumlar:**

- XMSEException
- MessageNotWritableException

## *WriteChar -Yazma Karakteri*

### **Arabirim:**

```
void WriteChar(Char value);
```

İleti akışına 2 byte, önce yüksek sıralı bayt olarak bir karakter yazın.

### **Parametreler:**

#### **değer (giriş)**

Yazılacak karakter.

### **Döndürülen:**

Hükümsüz

### **Özel durumlar:**

- XMSEException
- MessageNotWritableException

## *WriteDouble -Çift Duyarlıklı Kayan Nokta Numarası Yaz*

### **Arabirim:**

```
void WriteDouble(Double value);
```

Çift duyarlıklı kayan noktalı sayıyı uzun bir tamsayıya dönüştür ve uzun tamsayıyı ileti akışına 8 bayt, yüksek sıralı bayt olarak yazın.

### **Parametreler:**

#### **değer (giriş)**

Yazılacak çift duyarlıklı kayan noktalı sayı.

### **Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteFloat -Kayar Noktalı Sayı Yazma*

**Arabirim:**

```
void WriteFloat(Single value);
```

Bir kayar noktalı sayıyı bir tamsayıya dönüştürüp tamsayıyı 4 bayt, yüksek sıralı bayt olarak ileti akışısına yazın.

**Parametreler:****değer (giriş)**

Yazılacak kayan noktalı sayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteInt -Yazma Tamsayı*

**Arabirim:**

```
void WriteInt(Int32 value);
```

İleti akışına ilk olarak 4 bayt, yüksek sıralı bayt olarak bir tamsayı yazın.

**Parametreler:****değer (giriş)**

Yazılacak tamsayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteLong -Yazma Uzun Tamsayı*

**Arabirim:**

```
void WriteLong(Int64 value);
```

İleti akışına önce 8 bayt, yüksek sıralı bayt olarak uzun bir tamsayı yazın.

**Parametreler:****değer (giriş)**

Yazılacak uzun tamsayı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException
- MessageNotWritableException

*WriteObject -Yazma Nesnesi*

**Arabirim:**

```
void WriteObject(Object value);
```

Belirtilen veri tipi ile ileti akışına bir değer yazın.

**Parametreler:****objectType (giriş)**

Aşağıdaki nesne tiplerinden biri olması gereken değer:

Boolean  
Byte  
Byte[]  
Char  
Double  
Single  
Int32  
Int64  
Int16  
String

**değer (giriş)**

Yazılacak değeri içeren bir bayt dizisi.

**uzunluk (giriş)**

Dizideki bayt sayısı.

**Döndürülen:**

Hükümsüz

**Özel durumlar:**

- XMSEException

*WriteShort -Yazma Kısa Tamsayı*

**Arabirim:**

```
void WriteShort(Int16 value);
```

İleti akışına 2 byte, önce yüksek sıralı bayt olarak kısa bir tamsayı yazın.

**Parametreler:****değer (giriş)**

Yazılacak kısa tamsayı.

**Döndürülen:**

Hükümsüz

## Özel durumlar:

- XMSEException
- MessageNotWritableException

*WriteString -Yazma Dizgisi*

## Arabirim:

```
void WriteString(String value);
```

İleti akışa bir dizgi yazın.

## Parametreler:

### değer (giriş)

Yazılacak dizeyi sarsalayan bir String nesnesi.

## Döndürülen:

Hükümsüz

## Özel durumlar:

- XMSEException
- MessageNotWritableException

## Edinilmiş özellikler ve yöntemler

Aşağıdaki özellikler IMessage arabiriminden devralınır:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRevered, JMSReplyTo, JMSTimestapp, JMSType, Properties

Aşağıdaki yöntemler IMessage arabiriminden devralınır:

clearBody, clearProperties, PropertyExists

Aşağıdaki yöntemler IPropertyContext arabiriminden devralınır:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortÖzellik, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongÖzelliği, SetObjectProperty, SetShortProperty, SetStringProperty

## ITextMessage

Metin iletisi, gövdesi bir dizeyi oluşturan bir iletidir.

## Devralma sıradüzeni:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.ITextMessage
```

## İlgili başvurular

Metin iletileri

Metin iletisi gövdesi bir dizgi içeriyor.

## .NET özellikler



### Arabirim:

```
String Text
{
    get;
    set;
}
```

Metin iletisinin gövdeini oluşturan dizgiyi alın ve ayarlayın.

Gerekliyse, XMS dizgideki karakterleri yerel kod sayfasına dönüştürür.

### Özel durumlar:

- XMSEException
- MessageNotReadableException
- MessageNotWritableException
- MessageEOFException

### Edinilmiş özellikler ve yöntemler

Aşağıdaki özellikler IMessage arabiriminden devralınır:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRevered, JMSReplyTo, JMSTimestapp, JMSType, Properties

Aşağıdaki yöntemler IMessage arabiriminden devralınır:

clearBody, clearProperties, PropertyExists

Aşağıdaki yöntemler IPropertyContext arabiriminden devralınır:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortÖzellik, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongÖzelligi, SetObjectProperty, SetShortProperty, SetStringProperty

### TransactionInProgressException

#### Devralma sıradüzeni:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.TransactionInProgressException
```

XMS , bir işlem devam etmekte olduğu için geçerli olmayan bir işlem isteğinde bulunduysa, bu kural dışı durumu döndürür.

### Edinilmiş özellikler ve yöntemler

Aşağıdaki yöntemler, XMSEException arabiriminden edinilir:

GetErrorCode, GetLinkedException

## TransactionRolledBackException

### Devralma sıradüzeni:

```
IBM.XMS.XMSEnception
|
+----IBM.XMS.XMSEnception
      |
      +----IBM.XMS.TransactionRolledBackException
```

XMS , bir uygulama yürürlükteki hareketi kesinleştirmek için Session.commit() uygulaması çağırırsa, ancak işlem geriye işlenirse bu kural dışı durumu atar.

### Edinilmiş özellikler ve yöntemler

Aşağıdaki yöntemler, [XMSEnception](#) arabiriminden edinilir:

[GetErrorCode](#), [GetLinkedException](#)

## XMSEnception

XMS bir .NET yöntemi çağırısını işlerken bir hata saptarsa, XMS kural dışı durum yayınlar. Kural dışı durum, hatayla ilgili bilgileri sarmalayan bir nesnedir.

### Devralma sıradüzeni:

```
System.Exception
|
+----IBM.XMS.XMSEnception
```

Farklı tiplerde XMS kural dışı durumu vardır ve bir XMSEnception nesnesi yalnızca bir kural dışı durum tipidir. Ancak, XMSEnception sınıfı, diğer XMS kural dışı durum sınıflarının üst sınıfıdır. XMS , diğer kural dışı durum tiplerinden hiçbirinin uygun olmadığı durumlarda bir XMSEnception nesnesi atar.

### .NET özellikler

*ErrorCode -Hata Kodu Al*

#### Arabirim:

```
public String ErrorCode
{
    get {return errorCode_;}
}
```

Hata kodunu alın.

#### Özel durumlar:

- XMSEnception

*LinkedException -Bağlı Kural Dışı Durumu Al*

#### Arabirim:

```
public Exception LinkedException
{
    get { return linkedException_;}
    set { linkedException_ = value;}
}
```

Kural dışı durumlar zincirinde bir sonraki kural dışı durumu alın.

Zincirde başka kural dışı durumlar yoksa, yöntem boş değer döndürür.

**Özel durumlar:**

- XMSEException

## **XMSFactoryFactory**

Bir uygulama yönetilen nesnelere kullanmıyorsa, bağlantı üreticileri, kuyruklar ve konular yaratmak için bu sınıfı kullanın.

**Devralma sıradüzeni:**

Yok

### **.NET özellikler**

*Meta Veri-Meta verileri al*

**Arabirim:**

```
IConnectionMetaData MetaData
```

XMSFactoryFactory nesnesinin bağlantı tipine uygun meta verileri alın.

**Özel durumlar:**

Yok

### **Yöntemler**

*CreateConnectionÜreticisi-Bağlantı Üreticisi Yarat*

**Arabirim:**

```
IConnectionFactory CreateConnectionFactory();
```

Bildirilmiş tip için bir ConnectionFactory nesnesi yaratın.

**Parametreler:**

Yok

**Döndürülen:**

ConnectionFactory nesnesi.

**Özel durumlar:**

- XMSEException

*CreateQueue -Kuyruk Yarat*

**Arabirim:**

```
IDestination CreateQueue(String name);
```

İleti alışverişi sunucusunda bir kuyruğu temsil etmek için bir hedef nesne yaratın.

Bu yöntem, ileti sistemi sunucusunda kuyruğu yaratmaz. Bir uygulamanın bu yöntemi çağırabilmesi için önce kuyruğu oluşturmanız gerekir.

**Parametreler:****ad (giriş)**

Bir String nesnesi, kuyruğun adını sarsalıyor ya da kuyruğu tanımlayan bir URI (uniform resource identifier; bir tip kaynak tanıtıcısını (URI) sarsalıyor).

**Döndürülen:**

Kuyruğu gösteren hedef nesne.

**Özel durumlar:**

- XMSEException

*CreateTopic -Konu Oluştur*

**Arabirim:**

```
IDestination CreateTopic(String name);
```

Bir konuyu temsil etmek için bir hedef nesne yaratın.

**Parametreler:****ad (giriş)**

Bir dizgi nesnesi, konunun adını sarsalıyor ya da konuyu tanımlayan bir URI (uniform resource identifier; bir tip kaynak tanıtıcısını (URI) sarsalıyor).

**Döndürülen:**

Konuyu simgeleyen Hedef nesnesi.

**Özel durumlar:**

- XMSEException

*GetInstance (GetInstance)- XMSFactoryFactoryyönetim ortamını alın*

**Arabirim:**

```
static XMSFactoryFactory GetInstance(int connectionType);
```

Bir XMSFactoryFactoryyönetim ortamı yaratın. XMS uygulaması, gerekli protokol tipine uygun bir ConnectionFactory nesnesine başvuru almak için bir XMSFactoryFactory nesnesi kullanır. Bu ConnectionFactory nesnesi, yalnızca bu protokol tipi için bağlantı üretebilir.

**Parametreler:****connectionType (giriş)**

ConnectionFactory nesnesinin bağlantı ürettiği bağlantı tipi:

- XMSC.CT\_WPM
- XMSC.CT\_RTT
- XMSC.CT\_WMQ

**Döndürülen:**

Bildirilmiş bağlantı tipine adanmış XMSFactoryFactory nesnesi.

**Özel durumlar:**

- NotSupported()

## XMS nesnelerinin özellikleri

Bu bölüm , XMStarafından tanımlanan nesne özelliklerini belgelerdir.

bölüm , aşağıdaki kısımlar' i içerir:

- “Bağlantı Özellikleri” sayfa 173
- “ConnectionFactoryözellikleri” sayfa 174
- “ConnectionMetaVerileri Özellikleri” sayfa 179
- “Hedef Özellikleri” sayfa 180
- “InitialContextözellikleri” sayfa 182
- “İleti Özellikleri” sayfa 183
- “MessageConsumerözellikleri” sayfa 187
- “MessageProducerözellikleri” sayfa 188
- “Oturum Özellikleri” sayfa 188

Her bölüm , belirtilen tipteki bir nesnenin özelliklerini listeler ve her özelliğin kısa bir açıklamasını sağlar.

Bu bölüm , her özelliğin bir tanımını sağlayan “Özellik Tanımları” sayfa 188 bölüm' u da içerir.

Bir uygulama bu bölüme içinde açıklanan nesnelerin kendi özelliklerini tanımlıyorsa, bu, bir hataya neden olmaz, ancak beklenmedik sonuçlara neden olabilir.

**Not:** Bu bölümdeki özellik adları ve değerleri, C ve C + + için kullanılan form XMSC .NAME biçiminde gösterilir. Ancak, .NET' ta, özellik adının biçimi, nasıl kullanmanıza bağlı olarak XMSC .NAME ya da XMSC\_NAME olabilir:

- Bir özellik belirtiyorsanız, özellik adı aşağıdaki örnekte gösterildiği gibi XMSC .NAME biçiminde olmalıdır:

```
cf.SetStringProperty(XMSC.WMQ_CHANNEL, "DOTNET.SVRCONN");
```

- Bir dizgi belirtiyorsanız, özellik adı aşağıdaki örnekte gösterildiği gibi XMSC\_NAME biçiminde olmalıdır:

```
cf.SetStringProperty("XMSC_WMQ_CHANNEL", "DOTNET.SVRCONN");
```

.NET' ta, özellik adları ve değerleri, XMSC sınıfında sabit değerler olarak sağlanır. Bu sabitler dizgileri tanımlar ve herhangi bir XMS .NET uygulaması tarafından kullanılır. Önceden tanımlanmış bu değişmezleri kullanıyorsanız, özellik adları ve değerleri XMSC.NAME biçimlerinde bulunur; örneğin, XMSC.USERID, XMSC\_USERID yerine.

Veri tipleri aynı zamanda C/C + + için de kullanılır. .NET içinde ilgili değerleri “[.NET için veri tipleri](#)” sayfa 42 içinde bulabilirsiniz.

## İlgili kavramlar

[Kendi uygulamalarınızı oluşturma](#)

Örnek uygulamaları oluştururken kendi uygulamalarınızı oluşturun.

## İlgili başvurular

[.NET Arabirimler](#)

Bu bölüm , .NET sınıfı arabirimlerini ve bunların özelliklerini ve yöntemlerini içerir.

## Bağlantı Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla Connection nesnesinin özelliklerine genel bakış.

Çizelge 25. Bağlantı Özellikleri	
Özelliğin adı	Tanım
<a href="#">“XMSC_WMQ_RESOLVED_QUEUE_YÖNETİCELERİ” sayfa 221</a>	Bu özellik, bağlı olduğu kuyruk yöneticisinin adını elde etmek için kullanılır.
<a href="#">“XMSC_WMQ_RESOLVED_QUEUE_MANAGER_ID” sayfa 221</a>	Bu özellik, bağlantı tamamlandıktan sonra kuyruk yöneticisinin tanıtıcısıyla doldurulur.

<i>Çizelge 25. Bağlantı Özellikleri (devamı var)</i>	
<b>Özelliğın adı</b>	<b>Tanım</b>
<u>XMSSC_WPM_CONNECTION_PROTOCOL</u>	İleti alışverişı altyapısına yönelik bağlantı için kullanılan iletişim protokolü. Bu özellik salt okunurdur.
<u>XMLC_WPM_HOST_NAME</u>	Uygulamanın bağı olduğı ileti alışverişı altyapısını içeren sistemin anasistem adı ya da IP adresi. Bu özellik salt okunurdur.
<u>XMLC_WPM_ME_NAME</u>	Uygulamanın bağı olduğı ileti alışverişı altyapısının adı. Bu özellik salt okunurdur.
<u>XMLC_WPM_PORT</u>	Uygulamanın bağı olduğı ileti alışverişı altyapısının dinlediğı kapının numarası. Bu özellik salt okunurdur.

Connection nesnesi, bağlantıyı yaratmak için kullanılan bağlantı üreticisinin özelliklerinden türetilen salt okunur özelliklere de sahiptir. Bu özellikler yalnızca, bağlantı yaratıldığı sırada ayarlanan bağlantı üreticisi özelliklerinden değil, ayarlanmamış özelliklerin varsayılan değerlerinden türetilir. Özellikler arasında, yalnızca uygulamanın bağı olduğı ileti alışverişı sunucusu tipi için ilgili olanlar da bulunur. Özelliklerin adları, bağlantı üreticisi özelliklerinin adlarıyla aynıdır.

## **ConnectionFactoryözellikleri**

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren ConnectionFactory nesnesinin özelliklerine genel bakış.

<i>Çizelge 26. ConnectionFactoryözellikleri</i>	
<b>Özelliğın adı</b>	<b>Tanım</b>
<u>"XMSC_ASYNC_EXCAIMS" sayfa 197</u>	This property determines whether XMS informs an ExceptionListener only when a connection is broken, or when any exception occurs asynchronously to an XMS API call. Bu özellik, ExceptionListener 'in kayıtlı olduğı bu ConnectionFactory ' den oluşturulan tüm Connections için geçerlidir.
<u>XMLC_CLIENT_ID</u>	Bağılantı için istemci tanıtıcısı.
<u>XMLC_CONNECTION_TYPE</u>	Bir uygulamanın bağılandığı ileti sistemi sunucusu tipi.
<u>XMLC_PAROLASI</u>	Bir ileti sistemi sunucusuna bağlanma girişiminde bulunduğında uygulamanın kimliğini doğrulamak için kullanılacak parola.
<u>"XMLC_RTT_BROKER_PING_INTERVAL" sayfa 203</u>	XMS .NET ' in herhangi bir etkinliğı algılamak için Real Time Messaging Server bağlantısını denetlemesinden sonra geçen zaman aralığı (milisaniye).
<u>XMSC_RTT_CONNECTION_PROTOCOL</u>	Bir aracıya gerçek zamanlı bağlantı için kullanılan iletişim protokolü.
<u>XMLC_RTT_HOST_NAME</u>	Bir aracının çalıştığı sistemin anasistem adı ya da IP adresi.
<u>XMLC_RTT_LOCAL_ADRESİ</u>	Bir aracıya gerçek zamanlı bağlantı için kullanılacak yerel ağ arabiriminin anasistem adı ya da IP adresi.
<u>XMLC_RTT_MULTICAST</u>	Bir bağlantı üreticisi ya da hedefi için çoklu yayın ayarı.
<u>XMLC_RTT_PORT</u>	Bir aracının gelen istekleri dinlediğı kapı numarası.

Çizelge 26. ConnectionFactoryözellikleri (devamı var)	
Özellğin adı	Tanım
<u>XMLC_USERID</u>	Bir ileti alışverişi sunucusuna bağlanmayı denediğinde uygulamanın kimliğini doğrulamak için kullanılacak bir kullanıcı kimliği.
<u>XMLC_WMQ_BROKER_CONTROLO</u>	Bir aracı tarafından kullanılan denetim kuyruğunun adı. <b>Not:</b> This property can be used with Version 2.0 of IBM Message Service Client for .NET but has no effect for an application connected to an IBM WebSphere MQ 7.0 queue manager unless the XMSC_WMQ_PROVIDER_VERSION property of the connection factory is set to a version number less than 7.
<u>XMLC_WMQ_BROKER_PUBQ</u>	Uygulamaların yayınlayacağı iletileri gönderdikleri bir aracı tarafından izlenen kuyruğun adı. <b>Not:</b> This property can be used with Version 2.0 of IBM Message Service Client for .NET but has no effect for an application connected to an IBM WebSphere MQ 7.0 queue manager unless the XMSC_WMQ_PROVIDER_VERSION property of the connection factory is set to a version number less than 7.
<u>XMLC_WMQ_BROKER_QMGR</u>	Bir aracının bağlandığı kuyruk yöneticisinin adı. <b>Not:</b> This property can be used with Version 2.0 of IBM Message Service Client for .NET but has no effect for an application connected to an IBM WebSphere MQ 7.0 queue manager unless the XMSC_WMQ_PROVIDER_VERSION property of the connection factory is set to a version number less than 7.
<u>XMLC_WMQ_BROKER_SUBQ</u>	Kalıcı olmayan ileti tüketicisi için abone kuyruğunun adı. <b>Not:</b> This property can be used with Version 2.0 of IBM Message Service Client for .NET but has no effect for an application connected to an IBM WebSphere MQ 7.0 queue manager unless the XMSC_WMQ_PROVIDER_VERSION property of the connection factory is set to a version number less than 7.
<u>XMLC_WMQ_BROKER_VERSION</u>	Bir bağlantı için ya da hedef için uygulama tarafından kullanılan aracı tipi. <b>Not:</b> This property can be used with Version 2.0 of IBM Message Service Client for .NET but has no effect for an application connected to an IBM WebSphere MQ 7.0 queue manager unless the XMSC_WMQ_PROVIDER_VERSION property of the connection factory is set to a version number less than 7.
<u>“XMLC_WMQ_CCDTURL” sayfa 207</u>	İstemci kanal tanımlama çizelgesini içeren dosyanın adını ve yerini tanımlayan ve aynı zamanda dosyanın nasıl erişilebileceğini belirten bir URL adresi.
<u>XMLC_WMQ_CHANNEL</u>	Bağlantı için kullanılacak kanalın adı.

Çizelge 26. ConnectionFactory özellikleri (devamı var)	
Özellik adı	Tanım
<u>“XMLC_WMQ_CLIENT_RECONNECT_OPTIONS”</u> sayfa 208	Bu özellik, bu üretici tarafından yaratılan yeni bağlantılara ilişkin istemci yeniden bağlanma seçeneklerini belirler.
<u>“XMLC_WMQ_CLIENT_RECONNECT_TIMEOUT”</u> sayfa 208	Bu özellik, bir istemci bağlantısının yeniden bağlanmayı denediği süreyi (saniye olarak) belirtir.
<u>XMLC_WMQ_CONNECTION_MODE</u>	Bir uygulamanın kuyruk yöneticisine bağlanmasını sağlayan kip.
<u>“XMLC_WMQ_CONNECTION_NAME_LIST”</u> sayfa 209	Bu özellik, istemcinin bağlantısı kesildikten sonra istemcinin yeniden bağlanmayı denediği anasistemleri belirler.
<u>XMSSC_WMQ_FAIL_IF_QUIESCE</u>	Uygulamanın bağlı olduğu kuyruk yöneticisi susturma durumdaysa, belirli yöntemlere yapılan çağrılarının başarısız olup olmadığını belirler.
<u>XMLC_WMQ_HOST_NAME</u>	Bir kuyruk yöneticisinin çalıştığı sistemin anasistem adı ya da IP adresi.
<u>XMLC_WMQ_LOCAL_ADRESİ</u>	Bir kuyruk yöneticisine yönelik bir bağlantı için, bu özellik kullanılacak yerel ağ arabirimini ya da kullanılacak yerel ağ kapısını ya da yerel kapı aralığını ya da her ikisini de belirtir.
<u>XMSSC_WMQ_MESSAGE_SELECTION</u>	İleti seçmesinin XMS istemcisi ya da aracı tarafından yapılıp yapılmayacağını belirler.  <b>Not:</b> This property can be used with Version 2.0 of IBM Message Service Client for .NET but has no effect for an application connected to an IBM WebSphere MQ 7.0 queue manager unless the XMSC_WMQ_PROVIDER_VERSION property of the connection factory is set to a version number less than 7.
<u>XMLC_WMQ_MSG_BATCH_SIZE</u>	Zamanuyumsuz ileti teslimi kullanılırken, bir kuyruktan bir toplu iş kuyruğundan alınacak ileti sayısı üst sınırı.  <b>Not:</b> This property can be used with Version 2.0 of IBM Message Service Client for .NET but has no effect for an application connected to an IBM WebSphere MQ 7.0 queue manager unless the XMSC_WMQ_PROVIDER_VERSION property of the connection factory is set to a version number less than 7.
<u>XMLC_WMQ_POLLING_INTERVAL</u>	Bir oturumdaki her ileti dinleyicisinin kuyruğunda uygun bir ileti yoksa, bu değer, her ileti dinleyicisinin kuyruğundan bir ileti almak için yeniden denemesinden önce geçen, milisaniye cinsinden en yüksek aralığa sahip olur.  <b>Not:</b> This property can be used with Version 2.0 of IBM Message Service Client for .NET but has no effect for an application connected to an IBM WebSphere MQ 7.0 queue manager unless the XMSC_WMQ_PROVIDER_VERSION property of the connection factory is set to a version number less than 7.
<u>“XMLC_WMQ_PROVIDER_VERSION”</u> sayfa 218	Uygulamanın bağlanmayı amaçladığı kuyruk yöneticisinin sürüm, yayın, değişiklik düzeyi ve düzeltme paketi.



Çizelge 26. ConnectionFactoryözellikleri (devamı var)	
Özelliğin adı	Tanım
<u>XMLC_WMQ_PORT</u>	Bir kuyruk yöneticisinin gelen istekleri dinlediği kapı numarası.
<u>XMLC_WMQ_PUB_ACK_INTERVAL</u>	Bir yayıncı tarafından XMS istemcisinden önce yayınlanan ileti sayısı, aracıdan bir alındı bildirimini istemeden önce. <b>Not:</b> This property can be used with Version 2.0 of IBM Message Service Client for .NET but has no effect for an application connected to an IBM WebSphere MQ 7.0 queue manager unless the XMSC_WMQ_PROVIDER_VERSION property of the connection factory is set to a version number less than 7.
<u>“XMSC_WMQ_PUT_ASYNC_ALLOWES” sayfa 213</u>	Bu özellik, ileti üreticilerinin bu hedefe ileti göndermek için zamanuyumsuz yerleştirmeleri kullanmalarına izin verilip verilmeyeceğini belirler.
<u>XMLC_WMQ_QMGR_CCSD</u>	İleti Kuyruğu Arabirimi 'nde (MQI) tanımlanan karakter verileri alanlarının, XMS istemcisi ile WebSphere MQ istemcisi arasında değiş tokuş edilen kodlanmış karakter takımı ya da kod sayfasının tanıtıcısı (CCSID).
<u>XMLC_WMQ_QUEUE_YÖNETİCİSİ</u>	Bağlanmak için kuyruk yöneticisinin adı.
<u>XMLC_WMQ_RECEIVE_EXIT</u>	Bir kanal alma çıkışısının çalıştırılacağı tanıtılıyor.
<u>XMLC_WMQ_RECEIVE_EXIT_INIT</u>	Bir kanala geçirilen kullanıcı verileri, çağrıldığında çıkışa geçer.
<u>XMLC_WMQ_SECURITY_EXIT</u>	Kanal güvenlik çıkışını tanımlar.
<u>XMLC_WMQ_SECURITY_EXIT_INIT</u>	Bir kanal güvenliği çıkışa çağrıldığında geçirilen kullanıcı verileri.
<u>“XMLC_WMQ_SEND_CHECK_COUNT” sayfa 222</u>	Tek bir hareket dışı XMS oturumu içinde zamanuyumsuz koyma hataları olup olmadığını denetlemek için izin verilecek gönderme çağrılarının sayısı.
<u>XMLC_WMQ_SEND_EXIT</u>	Bir kanal gönderme çıkışını tanımlar.
<u>XMLC_WMQ_SEND_EXIT_INIT</u>	Kanal gönderme çıkışlarına geçirilen kullanıcı verileri çağrıldığında bu verileri gönderir.
<u>“XMLC_WMQ_SHARE_CONV_ALLOWED” sayfa 222</u>	Bir istemci bağlantısının yuvasını, kanal tanımlarının eşleşmesi durumunda aynı kuyruk yöneticisiyle aynı işlem düzeyinde diğer üst düzey XMS bağlantılarıyla paylaşım paylaşamayacağını belirler. Bu özellik, uygulama geliştirme, bakım ya da işletim nedenleri için gerekirse, Connections ürününün ayrı yuvalarda tam olarak yalıtılmasına izin vermek için sağlanır.
<u>XMLC_WMQ_SSL_CERT_STORES</u>	Bir kuyruk yöneticisiyle SSL bağlantısında kullanılacak sertifika iptal listelerini (CRL 'ler) tutan sunucuların konumları.
<u>XMLC_WMQ_SSL_CIPHER_SPEC</u>	Kuyruk yöneticisiyle güvenli bir bağlantıda kullanılacak CipherSpec adı.

<i>Çizelge 26. ConnectionFactoryözellikleri (devamı var)</i>	
<b>Özelliğın adı</b>	<b>Tanıım</b>
<u>XMSC_WMQ_SSL_CIPHER_SUIT</u>	Bir kuyruk yöneticisine yönelik TLS bağlantısında kullanılacak CipherSuite ' in adı. Güvenli bağlantının pazarlığı sırasında kullanılan iletişim kuralı, belirtilen CipherSuite' e (CipherSuite) bağlıdır.
<u>XMLC_WMQ_SSL_CRYPTO_HW</u>	İstemci sistemine bağlı şifreleme donanımlarıyla ilgili yapılandırma ayrıntıları.
<u>XMLC_WMQ_SSL_FIPS_REQUIRED</u>	Bu özelliğın değeri, bir uygulamanın FIPS uyumlu olmayan şifreleme takımlarını kullanıp kullanamayabileceğini belirler. Bu özellik true değerine ayarlanırsa, istemci-sunucu bağlantısı için yalnızca FIPS algoritmaları kullanılır.
<u>XMLC_WMQ_SSL_KEY_REPOSITORY</u>	Anahtarların ve sertifikaların depolanmış olduğu anahtar veritabanı dosyasının konumu.
<u>XMLC_WMQ_SSL_KEY_RESETCOUNT</u>	KeyResetCount, gizli anahtar yeniden anlaşılmadan önce, bir SSL etkileşimi içinde gönderilen ve alınan şifrelenmemiş baytların toplam sayısını temsil eder.
<u>XMLC_WMQ_SSL_PEER_NAME</u>	Bir kuyruk yöneticisiyle SSL bağlantısında kullanılacak eş adı.
<u>XMSSC_WMQ_SYNCPOINT_ALL_GETS</u>	Tüm iletilerin eşitleme noktası denetimi içindeki kuyruklardan alınıp alınmayacağı.
<u>"XMLC_WMQ_TARGET_İSTEMCENI" sayfa 229</u>	
<u>XMLC_WMQ_TEMP_Q_PREFIX</u>	Uygulama, XMS geçici kuyruk yarattığında yaratılan WebSphere MQ dinamik kuyruğunun adını oluşturmak için kullanılan önek.
<u>XMSC_WMQ_TEMP_TOPIC_PREFIX</u>	Geçici konular oluştururken, XMS , "TEMP/TEMPTOPICPREFIX/unique_id" formunun bir konu dizgisi oluşturur ya da bu özellik varsayılan değeri içeriyorsa, "TEMP/unique_id" dizgisi oluşturulur. Boş olmayan bir değer belirlenmesi, bu bağlantı altında oluşturulan geçici konulara aboneler için yönetilen kuyruklar oluşturmak için belirli model kuyruklarının tanımlanmasına olanak sağlar.
<u>XMLC_WMQ_TEMPORARY_MODEL</u>	Uygulama XMS geçici kuyruk yarattığında, dinamik bir kuyruğun yaratıldığı WebSphere MQ model kuyruğunun adı.
<u>XMLC_WPM_BUS_ADI</u>	Bir bağlantı üreticisi için, uygulamanın bağlandığı hizmet tümleştirme veriyolunun adı ya da hedefin varolduğu hizmet tümleştirme veriyolunun adı gibi bir hedef için bağlantı kuralı.
<u>XMSSC_WPM_CONNECTION_PROXIMITY</u>	Bağlantıya ilişkin bağlantı yakınlığı ayarı.
<u>XMLC_WPM_DUR_SUB_HOME</u>	Bir bağlantıya ya da hedefe ilişkin tüm kalıcı aboneliklerin yönetiliyor olduğu ileti alışverişi altyapısının adı.
<u>XMLC_WPM_LOCAL_ADRESİ</u>	Bir hizmet bütünleştirme veriyoluna bağlantı için, bu özellik kullanılacak yerel ağ arabirimini ya da kullanılacak yerel kapıyı ya da yerel kapı aralığını ya da her ikisini de belirtir.
<u>XMLC_WPM_NON_PERSISTENT_MAP</u>	Bağlantı kullanılarak gönderilen kalıcı olmayan iletilerin güvenilirliği düzeyi.

Çizelge 26. ConnectionFactory özellikleri (devamı var)	
Özellik adı	Tanım
<u>XMLC_WPM_PERSISTENT_MAP</u>	Bağlantı kullanılarak gönderilen kalıcı iletilerin güvenilirliği düzeyi.
<u>XMSC_WPM_PROVIDER_ENDERS</u>	Önyükleme sunucularının bir ya da daha çok uç noktası adresi dizisi.
<u>XMLC_WPM_TARGET_GROUP</u>	İleti alışverişi altyapılarının hedef grubunun adı.
<u>XMSSC_WPM_TARGET_MAITLE</u>	İleti alışverişi altyapılarının hedef grubunun önemini.
<u>XMSC_WPM_TARGET_TRANSPORT_CHAIN</u>	Bir ileti alışverişi işlemcisine bağlanmak için uygulamanın kullanması gereken gelen iletim zincirinin adı.
<u>XMLC_WPM_TARGET_TYPE</u>	İleti alışverişi altyapılarının hedef grubunun tipi.
<u>XMSC_WPM_TEMP_Q_PREFIX</u>	Uygulama XMS geçici kuyruk yarattığında, hizmet tümleştirme veriyolunda yaratılan geçici kuyruğun adını oluşturmak için kullanılan önek.
<u>XMSC_WPM_TEMP_TOPIC_PREFIX</u>	Uygulama tarafından oluşturulan geçici bir konunun adını oluşturmak için kullanılan önek.

### İlgili kavramlar

ConnectionFactory nesnelere ve Connection nesnelere

ConnectionFactory nesnesi, uygulamanın Connection nesnesi yaratmak için kullandığı bir şablon sağlar. Uygulama, bir Oturum nesnesi yaratmak için Bağlantı nesnesini kullanır.

Hizmet tümleştirme veriyoluna bağlantı

Bir XMS uygulaması, doğrudan TCP/IP bağlantısı kullanarak ya da TCP/IP üzerinden HTTP kullanarak bir WebSphere Application Server hizmet tümleştirme veriyoluna bağlanabilir.

IBM MQ kuyruk yöneticisine güvenli bağlantı

Bir XMS .NET uygulamasını bir IBM MQ kuyruk yöneticisinde güvenli bağlantı sağlamak üzere etkinleştirmek için, ilgili özelliklerin ConnectionFactory nesnesinde tanımlı olması gerekir.

WebSphere Application Server service integration bus ileti alışverişi altyapısına güvenli bağlantılar

To enable an XMS .NET application to make secure connections to a WebSphere Application Server service integration bus messaging engine, the relevant properties must be defined in the ConnectionFactory object.

Denetlenen nesnelere için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

### İlgili görevler

Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

### İlgili başvurular

Denetlenen ConnectionFactory nesnelere için gerekli özellikler

Bir uygulama bir bağlantı üreticisi yarattığında, bir ileti alışverişi sunucusuna yönelik bağlantı yaratmak için bir dizi özellik tanımlanmalıdır.

## ConnectionMetaVerileri Özellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren ConnectionMetaVeri nesnesinin özelliklerine genel bakış.

<i>Çizelge 27. ConnectionMetaVerileri Özellikleri</i>	
<b>Özellğin adı</b>	<b>Tanım</b>
<u>XMLC_JMS_MAJOR_VERSION</u>	XMS ' in temel aldığı JMS belirtiminin ana sürüm numarası. Bu özellik salt okunurdur.
<u>XMLC_JMS_MINOR_VERSION</u>	XMS ' in dayalı olduğu JMS belirtiminin ikincil sürüm numarası. Bu özellik salt okunurdur.
<u>XMLC_JMS_VERSION</u>	XMS ' in dayandığı JMS belirtiminin sürüm tanıtıcısı. Bu özellik salt okunurdur.
<u>XMLC_MAJOR_VERSION</u>	XMS istemcisinin sürüm numarası. Bu özellik salt okunurdur.
<u>XMLC_MINOR_VERSION</u>	XMS istemcisinin yayın numarası. Bu özellik salt okunurdur.
<u>XMLC_PROVIDER_NAME</u>	XMS istemcisi sağlayıcısı. Bu özellik salt okunurdur.
<u>XMSC_SÜRÜM</u>	XMS istemcisinin sürüm tanıtıcısı. Bu özellik salt okunurdur.

## Hedef Özellikleri

Daha ayrıntılı başvuru bilgilerine ilişkin bağlantılarla, Hedef nesnenin özelliklerine genel bakış.

<i>Çizelge 28. Hedef Özellikleri</i>	
<b>Özellğin adı</b>	<b>Tanım</b>
<u>XMLC_DELIVERY_MODE</u>	Hedefe gönderilen iletilerin teslim kipi.
<u>XMSC_PRIORITY</u>	Hedefe gönderilen iletilerin önceliği.
<u>XMLC_RTT_MULTICAST</u>	Bir bağlantı üreticisi ya da hedefi için çoklu yayın ayarı.
<u>XMLC_TIME_TO_LIVE</u>	Hedefe gönderilen iletiler için canlanacak süre.
<u>XMLC_WMQ_BROKER_VERSION</u>	Bir bağlantı için ya da hedef için uygulama tarafından kullanılan aracı tipi.
<u>XMLC_WMQ_CCSD</u>	Kodlanmış karakter takımının ya da kod sayfasının tanıtıcısı (CCSID), bir iletinin gövdesindeki karakter verilerinin dizgilerinin XMS istemcisi iletiyi hedefe ilettiğinde yer alıyor.
<u>XMLC_WMQ_DUR_SUBQ</u>	Hedeften ileti alan dayanıklı bir aboneye ilişkin abone kuyruğunun adı. <b>Not:</b> This property can be used with Version 2.0 of IBM Message Service Client for .NET but has no effect for an application connected to an IBM WebSphere MQ 7.0 queue manager unless the XMSC_WMQ_PROVIDER_VERSION property of the connection factory is set to a version number less than 7.
<u>XMLC_WMQ_ENCODING</u>	Bir iletinin gövdesindeki sayısal veriler, XMS istemcisi iletiyi hedefe ilettiğinde gösterilir.
<u>XMSSC_WMQ_FAIL_IF_QUIESCE</u>	Uygulamanın bağlı olduğu kuyruk yöneticisi susturma durumdaysa, belirli yöntemlere yapılan çağrıların başarısız olup olmadığını belirler.

Çizelge 28. Hedef Özellikleri (devamı var)	
Özellik adı	Tanım
<a href="#">“XMLC_WMQ_MESSAGE_BÖLGESİ” sayfa 211</a>	Bu özellik, bir XMS uygulamasının bir IBM WebSphere MQ iletisinin MQRFH2 ' yi ileti bilgi yükünün (ileti gövdesinin bir parçası olarak) bir parçası olarak işleip işlemediğini belirler.
<a href="#">“XMSC_WMQ_MQMD_MESSAGE_CONTEXT” sayfa 212</a>	Determines what level of message context is to be set by the XMS application. Uygulamanın, bu özelliğin geçerli olması için uygun bağlam yetkisi ile çalışıyor olması gerekir.
<a href="#">“XMLC_WMQ_MQMD_READENABLED” sayfa 213</a>	Bu özellik, bir XMS uygulamasının MQMD alanlarının değerlerini ayıklayıp ayıklayamayacağını belirler.
<a href="#">“XMSSC_WMQ_MQMD_WRITE_ENABLED” sayfa 213</a>	Bu özellik, bir XMS uygulamasının MQMD alanlarının değerlerini alıp veremeyeceğini belirler.
<a href="#">“XMSC_WMQ_READ_AHEAD_ALLOWD” sayfa 214</a>	Bu özellik, ileti tüketicilerinin ve kuyruk tarayıcılarının, bu hedeften kalıcı olmayan, hareket dışı iletileri almadan önce bu hedeften iç arabelleğe alınmak için önden okuma kullanmasına izin verilip verilmeyeceğini belirler.
<a href="#">“XMSC_WMQ_READ_AHEAD_CLOSE_POLICY” sayfa 214</a>	Bu özellik, ileti tüketicisi kapatıldığında, zamanuyumsuz ileti dinleyicisine gönderilmekte olan iletilerin, iç okuma yazma arabelleğindeki iletilere ne olacağını belirler.
<a href="#">“XMLC_WMQ_RECEIVE_CCSD” sayfa 220</a>	Kuyruk yöneticisi ileti dönüştürmesi için hedef CCSID ' yi belirleyen hedef özellik. XMLC_WMQ_RECEIVE_CONVERSION WMQ_RECEIVE_CONVERSION_QMGR değerine ayarlanmıyorsa, değer yoksayıdır.
<a href="#">“XMLC_WMQ_RECEIVE_CONVERSION” sayfa 220</a>	Veri dönüştürme işleminin kuyruk yöneticisi tarafından gerçekleştirilip gerçekleştirilmeyeceğini belirleyen hedef özellik.
<a href="#">XMLC_WMQ_TARGET_İSTEMCI</a>	Hedef hedefine gönderilen iletilerin bir MQRFH2 üstbilgisi içerip içermediğini belirtir.
<a href="#">XMSC_WMQ_TEMP_TOPIC_PREFIX</a>	Geçici konular oluştururken, XMS , "TEMP/TEMPTOPICPREFIX/unique_id" formunun bir konu dizgisi oluşturur ya da bu özellik varsayılan değeri içeriyorsa, "TEMP/unique_id" dizgisi oluşturulur. Boş olmayan bir değer belirlenmesi, bu bağlantı altında oluşturulan geçici konulara aboneler için yönetilen kuyruklar oluşturmak için belirli model kuyruklarının tanımlanmasına olanak sağlar.
<a href="#">XMLC_WPM_BUS_ADı</a>	Bir bağlantı üreticisi için, uygulamanın bağlandığı hizmet tümleştirme veriyolunun adı ya da hedefin varolduğu hizmet tümleştirme veriyolunun adı gibi bir hedef için bağlantı kurar.
<a href="#">XMSC_WPM_TOPIC_SPACE</a>	Konu yer alan konu alanının adı.

### İlgili kavramlar

[ConnectionFactories ve Connection nesneleri](#)

ConnectionFactory nesnesi, uygulamanın Connection nesnesi yaratmak için kullandığı bir şablon sağlar. Uygulama, bir Oturum nesnesi yaratmak için Bağlantı nesnesini kullanır.

[Hizmet tümleştirme veriyoluna bağlantı](#)

Bir XMS uygulaması, doğrudan TCP/IP bağlantısı kullanarak ya da TCP/IP üzerinden HTTP kullanarak bir WebSphere Application Server hizmet tümleştirme veriyoluna bağlanabilir.

#### Hedefler

Bir XMS uygulaması, gönderilmekte olan iletilerin hedefini ve alınmakta olan iletilerin kaynağını belirtmek için bir Hedef nesne kullanır.

#### Hedef joker karakterleri

XMS , hedef genel arama karakterleri için destek sağlar. Bu, joker karakterlerin, eşleştirme için gerekli olan yere aktarılabilmelerini sağlar. XMS ' un çalışabileceği her bir sunucu tipi için farklı bir joker karakter şeması vardır.

#### Konu URL tipi kaynak tanıtıcıları

Konu tekstili kaynak tanıtıcısı (URI), konunun adını belirtir; bir ya da daha çok özellik de belirtebilir.

#### Kuyruk tek biçimli kaynak tanıtıcıları

Kuyruğun URI 'si, kuyruğun adını belirtir; ayrıca, kuyruğun bir ya da daha fazla özelliğini de belirtebilir.

#### Geçici hedefler

XMS uygulamaları geçici hedefler oluşturabilir ve kullanabilir.

#### Denetlenen nesnelere için özellik eşlemesi

Uygulamaların IBM MQ JMS ve WebSphere Application Server bağlantı üreticisi ve hedef nesne tanımlarını kullanabilmesini sağlamak için, bu tanımlamalardan alınan özelliklerin, XMS bağlantı üreticileri ve hedeflerinde ayarlanabilen ilgili XMS özellikleriyle eşlenmeleri gerekir.

#### **İlgili görevler**

##### Denetlenen nesnelere yaratılıyor

XMS uygulamalarının bir ileti alışverişi sunucusuyla bağlantı yapmak için gerekli olan ConnectionFactory ve Destination nesne tanımlamaları, uygun denetim araçları kullanılarak yaratılmalıdır.

#### **İlgili başvurular**

##### Yönetilen Hedef nesnelere için gerekli özellikler

Hedef yaratan bir uygulama, yönetilen bir hedef nesnede uygulamanın bazı özellikleri ayarlamalıdır.

## **InitialContextözellikleri**

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren, InitialContext nesnesinin özelliklerine genel bakış.

<i>Çizelge 29. InitialContextözellikleri</i>	
<b>Özellik adı</b>	<b>Tanım</b>
<u>XMLC_IC_PROVIDER_URL</u>	COS adlandırma hizmetinin web hizmetiyle aynı sunucuda olması gerekmeyecek şekilde, JNDI adlandırma dizinini bulmak için kullanılır.
<u>XMSC_IC_SECURITY_AUTHENTICATION</u>	Java Bağlam arabirimi SECURITY_AUTHENTICATION ögesine dayalıdır. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.
<u>XMSC_IC_SECURITY_CREDENTIAL</u>	Java Bağlam arabirimi SECURITY_CREDENTIAL ögesini temel alır. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.
<u>XMSC_IC_SECURITY_PRINCIPAL</u>	Java Context interface SECURITY_PRINCIPAL ögesine dayalıdır. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.
<u>XMSC_IC_SECURITY_PROTOCOL</u>	Java Context interface SECURITY_PROTOCOL Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.
<u>XMLC_IC_URL</u>	LDAP ve FileSystem bağlamları için, denetlenen nesnelere içeren havuzun adresi. COS adlandırma bağlamları için, dizindeki nesnelere yukarı bakan web hizmetinin adresi.

## İlgili kavramlar

### InitialContext özellikleri

InitialContext oluşturucusunun deęiřtirgeleri, tek tip kaynak göstergesi (URI) olarak verilen, denetlenen nesnelere havuzunun yerini içerir. Bir uygulamanın havuzla bağlantı kurmasını sağlamak için, URI içinde yer alan bilgilerden daha fazla bilgi sağlamak gerekebilir.

### XMS başlangıç bağlamları için URI biçimi

Denetlenen nesnelere havuzunun konumu, tek tip bir kaynak göstergesi (URI) olarak sağlanır. URI biçimi bağlam tipine bağlıdır.

### Yönetilen nesnelere alınması

XMS retrieves an administered object from the repository using the address provided when the InitialContext object is created, or in the InitialContext properties.

## İlgili görevler

### InitialContext nesnelere

Bir uygulama, denetlenen nesnelere havuzuna, gerekli denetlenen nesnelere almak için bir bağlantı kurmak üzere kullanılacak bir başlangıç bağlamı oluşturmalıdır.

## İleti Özellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren, İleti nesnesinin özelliklerine genel bakış.

Çizelge 30. İleti Özellikleri	
Özellik adı	Tanım
<a href="#">JMS_IBM_CHARACTER_SET</a>	The identifier (CCSID) of the coded character set, or code page, that the strings of character data in the body of the message is in when the XMS client forwards the message to its intended destination. XMS ' de bu özellik sayısal bir değere sahiptir ve CCSID ile eşlenir. Ancak bu özellik bir JMS özelliğine dayalıdır, bu nedenle bir dizgi tipi değeri vardır ve bu sayısal CCSID ' yi temsil eden Java karakter kümesine eşlenir.
<a href="#">JMS_IBM_ENCODING</a>	İletinin gövdesindeki sayısal veriler, XMS istemcisi iletiyi amaçlanan hedefine ilettiğinde gösterilir.
<a href="#">JMS_IBM_EXCEPTIONMESSAGE</a>	İletinin kural dışı durum hedefine neden gönderildiğini açıklayan metin. Bu özellik salt okunurdur.
<a href="#">JMS_IBM_EXCEPTIONPROBLEMHEDDEF</a>	İletinin, ileti kural dışı durum hedefine gönderilmeden önce olduğu hedef adı.
<a href="#">JMS_IBM_EXCEPTIONREASON</a>	İletinin kural dışı durum hedefine neden gönderildiğini gösteren bir neden kodu.
<a href="#">JMS_IBM_EXCETTIONTIMESTAMP</a>	İletinin kural dışı durum hedefine gönderildiği saat.
<a href="#">JMS_IBM_FEEDBACK</a>	Bir rapor iletisinin niteliyi gösteren kod.
<a href="#">JMS_IBM_FORMA</a>	İleti, uygulama verilerinin iletiyle aynı.
<a href="#">JMS_IBM_LAST_MSG_IN_GROUP</a>	İletinin, bir ileti grubundaki son ileti olup olmadığını belirtin.
<a href="#">JMS_IBM_MSGTYPE</a>	İletinin tipi.
<a href="#">JMS_IBM_PUTAPPLTYPE</a>	İletiyi gönderen uygulama tipi.
<a href="#">JMS_IBM_PUTDATE</a>	İletinin gönderildiği tarih.
<a href="#">JMS_IBM_PUTTIME</a>	İletinin gönderildiği saat.

Çizelge 30. İleti Özellikleri (devamı var)	
Özellik adı	Tanım
<u>JMS_IBM_REPORT_COA</u>	Bir rapor iletilerinde özgün iletiden ne kadar uygulama verisi eklenmesi gerektiğini belirten 'varışta' doğrulama isteği ' rapor iletileri.
<u>JMS_IBM_REPORT_COD</u>	Bir rapor iletilerine özgün iletiden ne kadar uygulama verisi eklenmesi gerektiğini belirterek, 'teslimde teslim edilme' isteğini isteyin.
<u>JMS_IBM_REPORT_DISCARD_MSG</u>	İletinin, amaçlanan hedefine teslim edilemezse atıldığını isteyin.
<u>JMS_IBM_REPORT_EXCETION</u>	Bir rapor iletilerine özgün iletiden ne kadar uygulama verisi eklenmesi gerektiğini belirten istek kural dışı durum raporu iletileri.
<u>JMS_IBM_REPORT_EXPIRATION</u>	Bir rapor iletilerine özgün iletiden ne kadar uygulama verisi eklenmesi gerektiğini belirten süre bitimi raporu iletileri.
<u>JMS_IBM_REPORT_NAN</u>	Negatif eylem bildirim raporu iletileri iste.
<u>JMS_IBM_REPORT_PAN</u>	Pozitif işlem bildirim raporu iletileri isteyin.
<u>JMS_IBM_REPORT_PASS_CORREL_ID</u>	Herhangi bir rapor ya da yanıt iletilerinin ilinti tanıtıcısının, özgün iletilerin ilinti tanıtıcısıyla aynı olduğunu isteyin.
<u>JMS_IBM_REPORT_PASS_MSG_ID</u>	Herhangi bir rapor ya da yanıt iletilerinin ileti tanıtıcısının, özgün iletilerin ileti tanıtıcısıyla aynı olduğunu isteyin.
<u>JMS_IBM_RETAIN</u>	Bu özelliğin ayarlanması, kuyruk yöneticisinin bir iletiyi Alıkonan Yayını olarak ele almak için alıkonmasını belirtir.
<u>JMS_IBM_SYSTEM_MESSAGEID</u>	Hizmet bütünleştirme veriyolu içinde benzersiz olan iletiyi tanımlayan bir tanıtıcı. Bu özellik salt okunurdur.
<u>JMSX_APPID</u>	İletiyi gönderen uygulamanın adı.
<u>JMSX_DELIVERY_COUNT</u>	İletiyi teslim etme girişimi sayısı.
<u>JMSX_GROUPID</u>	İletinin ait olduğu ileti grubunun tanıtıcısı.
<u>JMSX_GROUPSEQ</u>	İleti grubu içindeki iletilerin sıra numarası.
<u>JMSX_USERID</u>	İletiyi gönderen uygulama ile ilişkili kullanıcı kimliği.

### JMS\_IBM\_MQMD\* özellikleri

IBM Message Service Client for .NET , istemci uygulamalarının API ' leri kullanarak MQMD alanlarını okumalarını/yazmalarını sağlar. Ayrıca, MQ ileti verilerine erişim de sağlar. MQMD ' ye varsayılan olarak erişim geçersiz kılınır ve XMSC\_WMQ\_MQMD\_WRITE\_ENABLED ve XMLC\_WMQ\_MQMD\_READ\_ENAB\_READENABLED hedef özellikleri kullanılarak uygulama tarafından belirtik olarak etkinleştirilmelidir. Bu iki özellik birbirlerinden bağımsızdır.

StrucId ve Version dışındaki tüm MQMD alanları ek İleti nesnesi özellikleri olarak ortaya çıkar ve JMS\_IBM\_MQMD önekli olur.

JMS\_IBM\_MQMD\* özellikleri, önceki çizelgede açıklanan JMS\_IBM\* gibi diğer özelliklerden daha öncelikli olarak uygulanır.



## İletilerin gönderilmesi

StrucId ve Version dışındaki tüm MQMD alanları temsil edilir. Bu özellikler yalnızca MQMD alanlarına gönderme yapar; hem MQMD ' de hem de MQRFH2 üstbilgisinde bir özellik oluşur, MQRFH2 içindeki sürüm belirlenmez ya da çıkarılmaz. JMS\_IBM\_MQMD\_BackoutCount dışında, bu özelliklerden herhangi biri ayarlanabilir. JMS\_IBM\_MQMD\_BackoutCount için ayarlanan herhangi bir değer yok sayılır.

Bir özelliğin uzunluk üst sınırı varsa ve siz çok uzun bir değer sağlıyorsa, değer kesilir.

Bazı özellikler için, Hedef nesnede XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT özelliğini de ayarlamanız gerekir. Uygulamanın, bu özelliğin geçerli olması için uygun bağlam yetkisi ile çalışıyor olması gerekir. XMSSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT değerini uygun bir değere ayarlamadıysanız, özellik değeri yok sayılır. XMSSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT değerini uygun bir değere ayarladığınızda, ancak kuyruk yöneticisi için yeterli bağlam yetkiniz yoksa, kural dışı durum yayınlanır. XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT için belirli değerleri gerektiren özellikler aşağıdaki gibidir.

Aşağıdaki özellikler XMSSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT için XMLC\_WMQ\_MDCTX\_SET\_IDENTITY\_CONTEXT ya da XMLC\_WMQ\_MDCTX\_SET\_ALL\_CONTEXT olarak ayarlanmasını gerektirir:

- JMS\_IBM\_MQMD\_UserIdentifier
- JMS\_IBM\_MQMD\_AccountingToken
- JMS\_IBM\_MQMD\_ApplIdentityVerileri

Aşağıdaki özellikler XMSSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT için XMSC\_WMQ\_MDCTX\_SET\_ALL\_CONTEXT olarak ayarlanmasını gerektirir:

- JMS\_IBM\_MQMD\_PutApplTipi
- JMS\_IBM\_MQMD\_PutApplAdı
- JMS\_IBM\_MQMD\_PutDate
- JMS\_IBM\_MQMD\_PutTime
- JMS\_IBM\_MQMD\_ApplOriginVerileri

## İletileri alma

XMLC\_WMQ\_MQMD\_READ\_READ\_ENABLED özelliği true olarak ayarlandıysa, üreten uygulama kümesinin ürettiği gerçek özelliklerden bağımsız olarak tüm bu özellikler alınmış bir iletiyle kullanılabilir. Bir uygulama, ilk olarak JMS belirtimine göre tüm özellikler temizlenmedikçe, alınan iletinin özelliklerini değiştiremez. Alınan ileti, özellikler değiştirilmeden iletilebilir.

**Not:** Uygulamanız, XMLC\_WMQ\_MQMD\_READ\_READ\_READ\_ENABLED özelliği true olarak ayarlanmış bir hedeften bir ileti alırsa ve bunu XMSC\_WMQ\_MQMD\_WRITE\_ENABLED değerine ayarlanmış bir hedefe iletirse, alınan iletinin tüm MQMD alanı değerleri, iletilen iletiye kopyalanıyor. Özellikler tablosu

Çizelge 31. MQMD alanlarını temsil eden ileti nesnesinin özellikleri		
Özellik	Tanım	Tip
JMS_IBM_MQMD_REPORT	Rapor iletileri için seçenekler	System.Int32
JMS_IBM_MQMD_MSGTYPE	İleti tipi	System.Int32
JMS_IBM_MQMD_EXPIRY	ileti kullanım süresi	System.Int32
JMS_IBM_MQMD_FEEDBACK	Geribildirim ya da neden kodu	System.Int32
JMS_IBM_MQMD_ENCODING	İleti verilerinin sayısal kodlaması	System.Int32
JMS_IBM_MQMD_CODEDCHARSETID	İleti verilerinin karakter kümesi tanıtıcısı	System.Int32
JMS_IBM_MQMD_FORMA	İleti verilerinin adını biçimle	System.String

Çizelge 31. MQMD alanlarını temsil eden ileti nesnesinin özellikleri (devamı var)		
Özellik	Tanım	Tip
JMS_IBM_MQMD_PRIORITY <b>Not:</b> JMS_IBM_MQMD_PRIORITY için 0-9 aralığında olmayan bir değer atarsanız, bu değer JMS belirtimini ihlal eder.	İleti önceliği	System.Int32
JMS_IBM_MQMD_PERSISTENCE	İleti kalıcılığı	System.Int32
JMS_IBM_MQMD_MSGID <b>Not:</b> JMS belirtimi, ileti tanıtıcısının JMS sağlayıcısı tarafından ayarlanması gerektiğini ve bunun benzersiz ya da boş değer olması gerektiğini belirtiyor. JMS_IBM_MQMD_MSGID değerine bir değer atarsanız, bu değer JMSMessageID'ye kopyalanır. Bu nedenle JMS sağlayıcısı tarafından ayarlanmamış ve benzersiz olmayabilir: bu değer JMS belirtimini ihlal eder.	İleti Tanıtıcısı	Bayt Dizisi <b>Not:</b> Bir iletideki bayt dizisi özelliklerinin kullanımı, JMS belirtimini ihlal eder.
JMS_IBM_MQMD_CORRELID <b>Not:</b> JMS_IBM_MQMD_CORRELID değerine, 'ID:' dizisiyle başlayan bir değer atarsanız, bu değer JMS belirtimini ihlal eder.	İlinti tanıtıcısı	Bayt Dizisi <b>Not:</b> Bir iletideki bayt dizisi özelliklerinin kullanımı, JMS belirtimini ihlal eder.
JMS_IBM_MQMD_BACKUTCOUNT	Geriletme sayacı	System.Int32
JMS_IBM_MQMD_REPLYTOQ	Yanıt kuyruğunun adı	System.String
JMS_IBM_MQMD_REPLYTOQMGR	Yanıt kuyruğu yöneticisinin adı	System.String
JMS_IBM_MQMD_USERIDENTIFIER	Kullanıcı kimliği	System.String
JMS_IBM_MQMD_ACCOUNTINGTOKEN	Hesap simgesi	Bayt Dizisi <b>Not:</b> Bir iletideki bayt dizisi özelliklerinin kullanımı, JMS belirtimini ihlal eder.
JMS_IBM_MQMD_APPLIDENTITDATA	Kimlikle ilgili uygulama verileri	System.String
JMS_IBM_MQMD_PUTAPPLTYPE	İletiyi koyan uygulamanın tipi	System.Int32
JMS_IBM_MQMD_PUTAPPLNAME	İletiyi koyan uygulamanın adı	System.String
JMS_IBM_MQMD_PUTDATE	İletinin konulduğu tarih	System.String
JMS_IBM_MQMD_PUTTIME	İletinin konulduğu saat	System.String
JMS_IBM_MQMD_APPLORIGINDATA	Köken ile ilgili uygulama verileri	System.String
JMS_IBM_MQMD_GROUPID	Grup tanıtıcısı	Bayt Dizisi <b>Not:</b> Bir iletideki bayt dizisi özelliklerinin kullanımı, JMS belirtimini ihlal eder.
JMS_IBM_MQMD_MSGSEQSAYI	Grup içindeki yerel iletinin sıra numarası	System.Int32

Çizelge 31. MQMD alanlarını temsil eden ileti nesnesinin özellikleri (devamı var)		
Özellik	Tanım	Tip
JMS_IBM_MQMD_OFFSET	Mantıksal iletinin başlangıcındaki fiziksel iletilerde verilerin görelî konumu	System.Int32
JMS_IBM_MQMD_MSGFLAGS	İleti İşaretleri	System.Int32
JMS_IBM_MQMD_ORIGINALLENGTH	Özgün iletinin uzunluđu	System.Int32

Ek ayrıntılar için [MQMD](#) başlıklı konuya bakın.

## Örnekler

Bu örnek, MQMD.UserIdentifier , "JoeBloggs" olarak ayarlanır.

```
// Create a ConnectionFactory, connection, session, producer, message
// ...

// Create a destination
// ...

// Enable MQMD write
dest.setBooleanProperty(XMSC_WMQ_MQMD_WRITE_ENABLED,
    XMSC_WMQ_MQMD_WRITE_ENABLED_YES);

// Optionally, set a message context if applicable for this MD field
dest.setIntProperty(XMSC_WMQ_MQMD_MESSAGE_CONTEXT,
    XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT);

// On the message, set property to provide custom UserId
msg.setStringProperty(JMS_IBM_MQMD_USERIDENTIFIER, "JoeBloggs");

// Send the message
// ...
```

JMS\_IBM\_MQMD\_USERIDENTIFIER ayarlanmadan önce XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT ayarını ayarlamamız gerekir. XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT kullanımı hakkında daha fazla bilgi için, İleti nesnesi özelliklerine bakın.

Benzer şekilde, bir ileti almadan önce ve daha sonra getStringözellığı gibi iletinin alma yöntemlerini kullanarak MQMD alanlarının içeriğini XMC\_WMQ\_MQMD\_READ\_ENABLED değerine true değerine ayarlayarak çıkartabilirsiniz. Alınan özellikler salt okunurdur.

Bu örnek, MQMD.ApplIdentityData , bir iletinin kuyruğundan ya da bir konudan alındı.

```
// Create a ConnectionFactory, connection, session, consumer
// ...

// Create a destination
// ...

// Enable MQMD read
dest.setBooleanProperty(XMSC_WMQ_MQMD_READ_ENABLED, XMSC_WMQ_MQMD_READ_ENABLED_YES);

// Receive a message
// ...

// Get required MQMD field value using a property
System.String value = rcvMsg.getStringProperty(JMS_IBM_MQMD_APPLIDENTITYDATA);
```

## MessageConsumerözellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren MessageConsumer nesnesinin özelliklerine genel bakış.

Çizelge 32. MessageConsumer özellikleri	
Özellik adı	Tanım
<u>XMSC_IS_SUBSCRIPTION_MULTICAST</u>	Indicates whether messages are being delivered to the message consumer using WebSphere MQ Multicast Transport. Bu özellik salt okunurdur.
<u>XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST</u>	İletilerin, güvenilir bir hizmet kalitesi ile WebSphere MQ Multicast Transport kullanılarak ileti tüketicisine teslim edilmekte olup olmadığını belirtir. Bu özellik salt okunurdur.

Ayrıntılar için [IMessageConsumer](#)' in .NET özellikleri dosyasına bakın.

## MessageProducer özellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren, MessageProducer nesnesinin özelliklerine genel bakış. Bkz. [.Daha fazla ayrıntı için IMessageProducer](#) ' in NET özellikleri.

## Oturum Özellikleri

Daha ayrıntılı başvuru bilgilerine bağlantılar içeren, Oturum nesnesinin özelliklerine genel bakış. Bkz. [.Daha fazla ayrıntı için NET UZUUse](#) .

## Özellik Tanımları

Bu bölüm , her nesne özelliği için bir tanımlama sağlar.

Her özellik tanımlaması aşağıdaki bilgileri içerir:

- Özelliğin veri tipi
- Özelliğin sahip olduğu nesne tipleri
- Hedef özelliği için, tek tip bir kaynak tanıtıcısında kullanılacak ad (URI)
- Özelliğe ilişkin daha ayrıntılı bir açıklama
- Özelliğin geçerli değerleri
- Özelliğin varsayılan değeri

Adları aşağıdaki örneklerden biriyle başlayan özellikler yalnızca belirtilen bağlantı tipi için anlamlıdır:

### XMLC\_RTT

Özellikler yalnızca bir aracıya gerçek zamanlı bağlantı için anlamlıdır. Özelliklerin adları, `xmsc_rtt` . hüstbilgi dosyasında adlandırılmış sabitler olarak tanımlanır.

### XMLC\_WMQ

Özellikler yalnızca, bir uygulama bir WebSphere MQ kuyruk yöneticisine bağlandığında anlamlıdır. Özelliklerin adları, `xmsc_wmq` . hüstbilgi dosyasında adlandırılmış sabitler olarak tanımlanır.

### XMLC\_WPM

Özellikler yalnızca, bir uygulama bir WebSphere hizmet tümleştirme veriyoluna bağlandığında anlamlıdır. Özelliklerin adları, `xmsc_wpm` . hüstbilgi dosyasında adlandırılmış sabitler olarak tanımlanır.

Tanımlamalarında aksi belirtilmedikçe, geri kalan özellikler tüm bağlantı tipleri için anlamlıdır. Özelliklerin adları, `xmsc` . hüstbilgi dosyasında adlandırılmış sabitler olarak tanımlanır. Adları JMSX önekiyle başlayan özellikler, bir iletinin JMS tanımlı özellikleridir ve adları JMS\_IBM önekiyle başlayan özellikler, bir iletinin IBM tanımlı özellikleridir. İletilerin özellikleri hakkında daha fazla bilgi için bkz. [“XMS iletisinin özellikleri” sayfa 68](#).

Aksi belirtilmediği sürece, her özellik hem noktadan noktaya iletişim , hem de yayınlama/abone olma etki alanlarında ilişkilidir.

Bir uygulama, özellik salt okunur olarak atanmadıkça, herhangi bir özelliğin değerini alabilir ve ayarlayabilirler.

### **JMS\_IBM\_XX\_ENCODE\_CASE\_ONE charter\_set**

**Veri tipi:**

System.Int32

**Özellik:**

İleti

The identifier (CCSID) of the coded character set, or code page, that the strings of character data in the body of the message is in when the XMS client forwards the message to its intended destination. XMS ' de bu özellik sayısal bir değere sahiptir ve CCSID ile eşlenir. Ancak bu özellik bir JMS özelliğine dayalıdır, bu nedenle bir dizgi tipi değeri vardır ve bu sayısal CCSID ' yi temsil eden Java karakter kümesine eşlenir. This property overrides any CCSID specified for the destination by the XMLC\_WMQ\_CCSID property.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

### **JMS\_IBM\_ENCODING**

**Veri tipi:**

System.Int32

**Özellik:**

İleti

İletinin gövdesindeki sayısal veriler, XMS istemcisi iletiyi amaçlanan hedefine ilettiğinde gösterilir. This property overrides any encoding specified for the destination by the XMLC\_WMQ\_ENCODING property. Bu özellik, ikili tamsayıların, paketlenmiş onlu tamsayıların ve kayan noktalı sayıların gösterilmesini belirtir.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **Encoding** alanında belirtilebilecek değerlerle aynıdır.

Bir uygulama, özelliği ayarlamak için aşağıdaki adı belirtilen değişmezleri kullanabilir:

<b>Adlandırılmış sabit</b>	<b>Anlamı</b>
MQENC_INTEGER_NORMAL	Normal tamsayı kodlaması
MQENC_INTEGER_TERS	Tersine çevrilmiş tamsayı kodlaması
MQENC_DECIMAL_NORMAL	Olağan paketlenmiş onlu kodlama
MQENC_DECIMAL_TERDI	Tersine çevrilmiş paketlenmiş ondalık kodlama
MQENC_FLOAT_IEEE_NORMAL	Olağan IEEE kayar noktalı kodlama
MQENC_FLOAT_IEEEE_TERS	Ters IEEE kayar noktalı kodlama
MQENC_FLOAT_S390	z/OS mimarisi kayan nokta kodlaması
MQENC_NATIVE	Yerli makine kodlaması

Özelliğe ilişkin bir değer oluşturmak için, uygulama bu değişmezlerden üçünü aşağıdaki gibi ekleyebilir:

- İkili tamsayıların gösterimini belirtmek için, adı MQENC\_XX\_ENCODE\_CASE\_ONE integer ile başlayan bir değişmez.
- Paketlenmiş onlu tamsayıların gösterimini belirtmek için, adı MQENC\_DECIMAL ile başlayan bir değişmez.
- Yüzer nokta sayılarının gösterimini belirtmek için, adı MQENC\_FLOAT olan bir değişmez.

Diğer bir seçenek olarak, uygulama özelliği, değeri ortama bağlı olan MQENC\_NATIVE değerine ayarlayabilir.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

### ***JMS\_IBM\_EXCEPTIONMESSAGE***

**Veri tipi:**

Dizgi

**Özellik:**

İleti

İletinin kural dışı durum hedefine neden gönderildiğini açıklayan metin. Bu özellik salt okunurdur.

Bu özellik, yalnızca bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında ve bir kural dışı durum hedefinden bir ileti alındığında anlamlıdır.

### ***JMS\_IBM\_EXCEPTIONPROBLEMINVARIŞ***

**Veri tipi:**

Dizgi

**Özellik:**

İleti

İletinin, ileti kural dışı durum hedefine gönderilmeden önce olduğu hedef adı.

Bu özellik, yalnızca bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında ve bir kural dışı durum hedefinden bir ileti alındığında anlamlıdır.

### ***JMS\_IBM\_EXCEPTIONREASON***

**Veri tipi:**

System.Int32

**Özellik:**

İleti

İletinin kural dışı durum hedefine neden gönderildiğini gösteren bir neden kodu.

Bu özellik, yalnızca bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında ve bir kural dışı durum hedefinden bir ileti alındığında anlamlıdır.

### ***JMS\_IBM\_EXCEPTIMESTAMP***

**Veri tipi:**

System.Int64

**Özellik:**

İleti

İletinin kural dışı durum hedefine gönderildiği saat.

Saat, 1 Ocak 1970 'te 00:00:00 GMT ' den bu yana milisaniye olarak ifade edilir.

Bu özellik, yalnızca bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında ve bir kural dışı durum hedefinden bir ileti alındığında anlamlıdır.

### ***JMS\_IBM\_FEEDBACK***

**Veri tipi:**

System.Int32

**Özellik:**

İleti

Bir rapor iletisinin niteliyi gösteren kod.

Özellikğin geçerli değerleri, bir ileti tanımlayıcısının **Feedback** alanında belirtilebilecek geribildirim kodlarıdır ve neden kodlarıdır.

Varsayılan olarak özellik ayarlanmaz.

### ***JMS\_\_IBM\_***

**Veri tipi:**

Dizgi

**Özellik:**

İleti

İleti, uygulama verilerinin iletiyle aynı.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **Format** alanında belirtilebilecek değerlerle aynıdır.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

### ***JMS\_IBM\_LAST\_MSG\_IN\_GROUP***

**Veri tipi:**

System.Boolean

**Özellik:**

İleti

İletinin, bir ileti grubundaki son ileti olup olmadığını belirtin.

İleti, bir ileti grubundaki son iletiyse, özelliği true olarak ayarlayın. Ters durumda, özelliği false olarak ayarlayın ya da özelliği ayarlamayın. Varsayılan olarak özellik ayarlanmaz.

True değeri, ileti tanımlayıcısının **MsgFlags** alanında belirtilebilen MQMF\_LAST\_MSG\_IN\_GROUP durum işaretine karşılık gelir.

Bu özellik, yayınlama/abone olma etki alanında yoksayılr ve bir uygulama bir hizmet tümleştirme veriyoluna bağlandığında, bu özellik için uygun değildir.

### ***JMS\_IBM\_MSGTYPE***

**Veri tipi:**

System.Int32

**Özellik:**

İleti

İletinin tipi.

Özelliğin geçerli değerleri şunlardır:

<b>Geçerli değer</b>	<b>Anlamı</b>
MQMT_DATAGRAM	İleti, yanıt gerektirmeyen bir iletidir.
MQMT_REQUEST	İleti, yanıt gerektiren bir iletidir.
MQMT_REPLY	İleti bir yanıt iletidir.
MQMT_REPORT	İleti, bir rapor iletidir.

Bu değerler, bir ileti tanımlayıcısının **MsgType** alanında belirtilebilecek ileti tipleriyle karşılık gelir.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

### ***JMS\_IBM\_PUTAPPLTYPE***

**Veri tipi:**

System.Int32

**Özellik:**

İleti

İletiyi gönderen uygulama tipi.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **PutApp1Type** alanında belirtilebilecek uygulama tipleridir.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

**JMS\_IBM\_PUTDATE****Veri tipi:**

Dizgi

**Özellik:**

İleti

İletinin gönderildiği tarih.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **PutDate** alanında belirtilebilecek değerlerle aynıdır.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

**JMS\_IBM\_PUTTIME****Veri tipi:**

Dizgi

**Özellik:**

İleti

İletinin gönderildiği saat.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **PutTime** alanında belirtilebilecek değerlerle aynıdır.

Varsayılan olarak özellik ayarlanmaz.

Bir uygulama bir hizmet bütünleştirme veriyoluna bağlandığında bu özellik ilgili değildir.

**JMS\_IBM\_REPORT\_COA****Veri tipi:**

System.Int32

**Özellik:**

İleti

Bir rapor iletisinde özgün iletiden ne kadar uygulama verisi eklenmesi gerektiğini belirten 'varışta' doğrulama isteği ' rapor iletileri.

Özelliğin geçerli değerleri şunlardır:

**Geçerli değer**

MQRO\_COA

MQRO\_CO\_WITH\_DATA

MQRO\_COA\_WITHL\_FULL\_DATA

**Anlamı**

Bir rapor iletisinde yer alan özgün iletiden uygulama verileri olmadan, 'varışta onayla' raporu iletilerine istek iletin.

Bir rapor iletisinde yer alan özgün iletiden ilk 100 bayt olan uygulama verilerinin ilk 100 baytı ile birlikte 'varışta doğrulama' isteği iletilerini isteyin.

Bir rapor iletisinde yer alan özgün iletiden tüm uygulama verileriyle birlikte 'varışta onayla' iletisi iletilerini isteyin.



Bu deęerler, bir ileti tanımlayıcısının **Report** alanında belirtilebilecek rapor seęeneklerine karşılık gelir. Bu seęenekler hakkında daha fazla bilgi için bkz. [Rapor \(MQUZE\)](#).

Varsayılan olarak özellik ayarlanmaz.

### **JMS\_IBM\_REPORT\_COD**

**Veri tipi:**

System.Int32

**Özellik:**

İleti

Bir rapor iletisine özgün iletiden ne kadar uygulama verisi eklenmesi gerektiğini belirterek, 'teslimde teslim edilme' isteğini isteyin.

Özelliğın geçerli deęerleri şunlardır:

**Geçerli deęer**

**Anlamı**

MQRO\_COD

Bir rapor iletisinde yer alan özgün iletiden herhangi bir uygulama verisi olmadan, 'teslimde teslim edin' isteęi iletilsin.

MQRO\_COD\_WITH\_DATA

Bir rapor iletisinde yer alan özgün iletiden ilk 100 bayt olan uygulama verilerinin ilk 100 baytı ile, 'teslimde teslim edilme' isteğini isteyin.

MQRO\_COD\_WITH\_FULL\_DATA

Bir rapor iletisinde yer alan özgün iletiden tüm uygulama verileriyle birlikte 'teslimde teslim edin' isteęi iletilsin.

Bu deęerler, bir ileti tanımlayıcısının **Report** alanında belirtilebilecek rapor seęeneklerine karşılık gelir.

Varsayılan olarak özellik ayarlanmaz.

### **JMS\_IBM\_REPORT\_DISCARD\_MSG**

**Veri tipi:**

System.Int32

**Özellik:**

İleti

İletinin, amaçlanan hedefine teslim edilemezse atıldığını isteyin.

İletiyi MQRO\_DISCARD\_MSG olarak ayarlamak için, iletinin amaçlanan hedefine teslim edilemezse atıldığını isteyin. İletinin, bunun yerine bir ölü mektup kuyruęuna konmasını ya da bir kural dıřı durum hedefine gönderilmesini istiyorsanız, bu özellięi ayarlamayın. Varsayılan olarak özellik ayarlanmaz.

MQRO\_DISCARD\_MSG deęeri, bir ileti tanımlayıcısının **Report** alanında belirtilebilecek bir rapor seęeneęine karşılık gelir.

### **JMS\_IBM\_REPORT\_EXCETION**

**Veri tipi:**

System.Int32

**Özellik:**

İleti

Bir rapor iletisine özgün iletiden ne kadar uygulama verisi eklenmesi gerektiğini belirten istek kural dıřı durum raporu iletileri.

Özelliğın geçerli deęerleri şunlardır:

**Geçerli değer**

MORO\_EXCEPTION

MORO\_EXCEPTION\_WITH\_DATA

MORO\_EXCEPTION\_WITH\_FULL\_DATA

**Anlamı**

Bir rapor iletisinde yer alan özgün iletiden uygulama verileri olmadan, istek kural dışı durumu rapor iletileri.

Bir rapor iletisinde yer alan özgün iletiden ilk 100 bayt uygulama verisi içeren, istek kural dışı durumu rapor iletileri.

Bir rapor iletisinde yer alan özgün iletiden tüm uygulama verilerine sahip istek kural dışı durum raporu iletileri.

Bu değerler, bir ileti tanımlayıcısının **Report** alanında belirtilebilecek rapor seçeneklerine karşılık gelir. Varsayılan olarak özellik ayarlanmaz.

**JMS\_IBM\_REPORT\_EXPIRATION****Veri tipi:**

System.Int32

**Özellik:**

İleti

Bir rapor iletisine özgün iletiden ne kadar uygulama verisi eklenmesi gerektiğini belirten süre bitimi raporu iletileri.

Özelliğin geçerli değerleri şunlardır:

**Geçerli değer**

MORO\_EXPIRATION

MORO\_EXPIRATION\_WITH\_DATA

MORO\_EXPIRATION\_WITH\_FULL\_DATA

**Anlamı**

Bir rapor iletisinde yer alan özgün iletiden herhangi bir uygulama verisi olmadan, süre bitimi rapor iletilerini talep edin.

Bir rapor iletisinde yer alan özgün iletiden ilk 100 bayt olan uygulama verilerinin ilk 100 baytı ile son kullanma tarihi raporu isteğinde bulunmaya devam edin.

Bir rapor iletisinde yer alan özgün iletiden tüm uygulama verileriyle birlikte, süre bitimi rapor iletilerini talep edin.

Bu değerler, bir ileti tanımlayıcısının **Report** alanında belirtilebilecek rapor seçeneklerine karşılık gelir. Varsayılan olarak özellik ayarlanmaz.

**JMS\_IBM\_REPORT\_NAN****Veri tipi:**

System.Int32

**Özellik:**

İleti

Negatif eylem bildirim raporu iletileri iste.

Negatif işlem bildirim raporu iletileri istemek için özelliği MORO\_NAN olarak ayarlayın. Olumsuz işlem bildirim raporu iletileri gerekmiyorsa, özelliği ayarlamayın. Varsayılan olarak özellik ayarlanmaz.

MORO\_NAN değeri, bir ileti tanımlayıcısının **Report** alanında belirtilebilecek bir rapor seçeneğine karşılık gelir.

## **JMS\_IBM\_REPORT\_PAN**

### **Veri tipi:**

System.Int32

### **Özellik:**

İleti

Pozitif işlem bildirim rapor iletileri isteyin.

Pozitif işlem bildirim rapor iletileri istemek için özelliği MQRO\_PAN olarak ayarlayın. Pozitif işlem bildirim raporu iletileri gerektirmiyorsa, özelliği ayarlamayın. Varsayılan olarak özellik ayarlanmaz.

MQRO\_PAN değeri, bir ileti tanımlayıcısının **Report** alanında belirtilebilecek bir rapor seçeneğine karşılık gelir.

## **JMS\_IBM\_REPORT\_PASS\_CORREL\_ID**

### **Veri tipi:**

System.Int32

### **Özellik:**

İleti

Herhangi bir rapor ya da yanıt iletilerinin ilinti tanıtıcısının, özgün iletilerinin ilinti tanıtıcısıyla aynı olduğunu isteyin.

Özelliğin geçerli değerleri şunlardır:

#### **Geçerli değer**

MQRO\_PASS\_COREL\_ID

#### **Anlamı**

Herhangi bir rapor ya da yanıt iletilerinin ilinti tanımlayıcısının, özgün iletilerinin ilinti tanıtıcısıyla aynı olduğunu isteyin.

MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID

Herhangi bir rapor ya da yanıt iletilerinin ilinti tanımlayıcısının, özgün iletilerinin ileti tanıtıcısıyla aynı olduğunu isteyin.

Bu değerler, bir ileti tanımlayıcısının **Report** alanında belirtilebilecek rapor seçeneklerine karşılık gelir.

Özelliğin varsayılan değeri MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID 'dir.

## **JMS\_IBM\_REPORT\_PASS\_MSG\_ID**

### **Veri tipi:**

System.Int32

### **Özellik:**

İleti

Herhangi bir rapor ya da yanıt iletilerinin ileti tanıtıcısının, özgün iletilerinin ileti tanıtıcısıyla aynı olduğunu isteyin.

Özelliğin geçerli değerleri şunlardır:

#### **Geçerli değer**

MQRO\_PASS\_MSG\_ID

#### **Anlamı**

Herhangi bir rapor ya da yanıt iletilerinin ileti tanıtıcısının, özgün iletilerinin ileti tanıtıcısıyla aynı olduğunu kabul edin.

MQRO\_NEW\_MSG\_ID

Her bir rapor ya da yanıt iletilerisi için yeni bir ileti tanıtıcısı oluşturulduğunu isteyin.

Bu değerler, bir ileti tanımlayıcısının **Rapor** alanında belirtilebilecek rapor seçeneklerine karşılık gelir.

Özelliğin varsayılan değeri MQRO\_NEW\_MSG\_ID 'dir.

## **JMS\_IBM\_RETAIN**

**Veri tipi:**

System.Int32

**Özellik:**

İleti

Bu özelliğin ayarlanması, kuyruk yöneticisinin bir iletiyi Alıkonan Yayını olarak ele almak için alıkonmasını belirtir. Bir abone konulardan ileti aldığı anda, önceki yayınlarda alınan iletilerin ötesinde, abone olduktan hemen sonra ek iletiler alabilirler. Bu iletiler, abone olunan konular için isteğe bağlı tutulan yayınlardır. Aboneliğe eşleşen her konu için, alıkonan bir yayın varsa, yayını abone olan ileti tüketicisine teslim etmek için kullanılır.

RETAIN\_YAYIN, bu özellik için tek geçerli değer. Varsayılan olarak bu özellik belirlenmez.

**Not:** Bu özellik yalnızca yayınlama/abone olma etki alanında anlamlıdır

## **JMS\_IBM\_SYSTEM\_MESSAGEID**

**Veri tipi:**

Dizgi

**Özellik:**

İleti

Hizmet bütünleştirme veriyolu içinde benzersiz olan iletiyi tanımlayan bir tanıtıcı. Bu özellik salt okunurdur.

Bu özellik, yalnızca bir uygulama bir hizmet tümleştirme veriyoluna bağlandığında anlamlıdır.

## **JMS\_APPID**

**Veri tipi:**

Dizgi

**Özellik:**

İleti

İletiyi gönderen uygulamanın adı.

Bu özellik, JMS adı JMSXAppIDolan JMS tanımlı özellidir. Özelliğe ilişkin ek bilgi için *Java Message Service Specification, Sürüm 1.1* başlıklı konuya bakın.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

## **JMSX\_DELIVERY\_COUNT**

**Veri tipi:**

System.Int32

**Özellik:**

İleti

İletiyi teslim etme girişimi sayısı.

Bu özellik, JMS adı JMSXDeliveryCountolan JMS özelliğine sahip özeldir. Özelliğe ilişkin ek bilgi için *Java Message Service Specification, Sürüm 1.1* başlıklı konuya bakın.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

## **JMSX\_GROUPID**

**Veri tipi:**

Dizgi

**Özellik:**

İleti

İletinin ait olduğu ileti grubunun tanıtıcısı.

Bu özellik, JMS adı JMSXGroupIDolan JMS tanımlı özeldir. Özelliğe ilişkin ek bilgi için *Java Message Service Specification, Sürüm 1.1* başlıklı konuya bakın.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

**JMSX\_GROUPSEQ****Veri tipi:**

System.Int32

**Özellik:**

İleti

İleti grubu içindeki iletinin sıra numarası.

Bu özellik, JMS adı JMSXGroupSeqolan JMS tanımlı özeldir. Özelliğe ilişkin ek bilgi için *Java Message Service Specification, Sürüm 1.1* başlıklı konuya bakın.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

**JMSX\_USERID****Veri tipi:**

Dizgi

**Özellik:**

İleti

İletiyi gönderen uygulama ile ilişkili kullanıcı kimliği.

Bu özellik, JMS adı JMSXUserIDolan JMS tanımlı özeldir. Özelliğe ilişkin ek bilgi için *Java Message Service Specification, Sürüm 1.1* başlıklı konuya bakın.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, bir aracıya gerçek zamanlı bağlantı için geçerli değildir.

**XMSC\_ASYNC\_EXCAIMS****Veri tipi:**

System.Int32

**Özellik:**

ConnectionFactory

**Uygulanabilir nesnelere:**

JMS denetim aracı uzun adı: ASYNCEXCEPTION

JMS yönetim aracı kısa adı: AEX

This property determines whether XMS informs an ExceptionListener only when a connection is broken, or when any exception occurs asynchronously to an XMS API call. Bu özellik, ExceptionListener'in kayıtlı olduğu bu ConnectionFactory ' den oluşturulan tüm Connections için geçerlidir.

Bu özelliğe ilişkin geçerli değerler şunlardır:

**XMLC\_ASYNC\_EXCEPTIONS\_ALL**

Zamanuyumlu bir API çağrısının kapsamı dışında, zamanuyumsuz olarak algılanan herhangi bir kural dışı durum oluştu ve tüm bağlantı bozuk kural dışı durumları ExceptionListener'e gönderildi.

## **XMSC\_ASYNC\_EXCEPTIONS\_CONNECTIONBROKEN**

Yalnızca, bozuk bir bağlantının ExceptionListener' e gönderildiğini gösteren kural dışı durumlar vardır. Zamanuyumsuz işleme sırasında oluşan diğer kural dışı durumlar ExceptionListener(ExceptionListener) olarak bildirilmez ve bu nedenle, uygulama bu kural dışı durumlarla ilgili bilgilendirilmez.

Varsayılan olarak bu özellik XMC\_ASYNC\_EXCEPTIONS\_ALL olarak ayarlıdır.

## **XMLC\_CLIENT\_ID**

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

### **Uygulanabilir nesnelere:**

JMS denetim aracı uzun adı: CLIENTID

JMS yönetim aracı kısa adı: CID

Bağlantı için istemci tanıtıcısı.

Bir istemci tanıtıcısı yalnızca, yayınlama/abone olma etki alanında dayanıklı abonelikleri desteklemek için kullanılır ve noktadan noktaya iletişim etki alanında yoksayılar. İstemci tanıtıcılarını ayarlama hakkında daha fazla bilgi için bkz. [“ConnectionFactory ve Connection nesnelere” sayfa 20.](#)

Bu özellik, bir aracıyla gerçek zamanlı bağlantı için uygun değildir.

## **XMLC\_CONNECTION\_TYPE**

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory

Bir uygulamanın bağlandığı ileti sistemi sunucusu tipi.

Özelliğin geçerli değerleri şunlardır:

### **Geçerli değer**

XMLC\_CT\_RTT

XMLC\_CT\_WMQ

XMLC\_CT\_WPM

### **Anlamı**

Bir aracıya gerçek zamanlı bağlantı.

WebSphere MQ kuyruk yöneticisiyle bağlantı.

Bir WebSphere hizmet tümleştirme veriyoluna bağlantı.

Varsayılan olarak özellik ayarlanmaz.

## **XMLC\_DELIVERY\_MODE**

### **Veri tipi:**

System.Int32

### **Özellik:**

Hedef

### **Bir URI ' de kullanılan ad:**

persistence (bir WebSphere MQ hedefi için)

deliveryMode (bir WebSphere varsayılan ileti sistemi sağlayıcısı hedefi için)

### **Uygulanabilir nesnelere:**

JMS yönetim aracı uzun adı: PERSISTENCE

JMS denetim aracı kısa adı: PER

Hedefe gönderilen iletilerin teslim kipi.

Özelliğın geçerli değeri şunlardır:

### **Geçerli değeri**

XMSC\_DELIVERY\_NOT\_PERSISTENT

### **Anlamı**

Hedefe gönderilen bir ileti kalıcı değil. İleti üreticisinin varsayılan teslim kipi ya da gönderme çağrısında belirtilen herhangi bir teslim kipi yoksayılr. If the destination is a WebSphere MQ queue, the value of the queue attribute *DefPersistence* is also ignored.

XMSC\_DELIVERY\_PERSISTENT

Hedefe gönderilen bir ileti kalıcıdır. İleti üreticisinin varsayılan teslim kipi ya da gönderme çağrısında belirtilen herhangi bir teslim kipi yoksayılr. If the destination is a WebSphere MQ queue, the value of the queue attribute *DefPersistence* is also ignored.

XMLC\_DELIVERY\_AS\_APP

Hedefe gönderilen bir ileti, Gönderme çağrısında belirlenen teslim kipine sahip olur. Gönderme çağrısı herhangi bir teslim kipi belirtmezse, bunun yerine ileti üreticisinin varsayılan teslim kipi kullanılır. Hedef bir WebSphere MQ kuyruğuyrsa, *DefPersistence* kuyruk özniteliğinin değeri yoksayılr.

XMLC\_DELIVERY\_AS\_DEST

Hedef bir WebSphere MQ kuyruksa, kuyruğa konan bir ileti, *DefPersistence* kuyruk özniteliğinin değeri tarafından belirtilen teslim kipine sahip olur. İleti üreticisinin varsayılan teslim kipi ya da gönderme çağrısında belirtilen herhangi bir teslim kipi yoksayılr.

Hedef bir WebSphere MQ kuyruğu değilse, anlam XMC\_DELIVERY\_AS\_APP ile aynı olur.

Varsayılan değeri XMSC\_DELIVERY\_AS\_APP ' dir.

## ***XMLC\_IC\_PROVIDER\_URL***

### **Veri tipi:**

Dizgi

### **Özellik:**

InitialContext

COS adlandırma hizmetinin web hizmetiyle aynı sunucuda olması gerekmeyecek şekilde, JNDI adlandırma dizinini bulmak için kullanılır.

## ***XMSC\_IC\_SECURITY\_AUTHENTICATION***

### **Veri tipi:**

Dizgi

### **Özellik:**

InitialContext

Java Bağlam arabirimi SECURITY\_AUTHENTICATION ögesine dayalıdır. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.

## ***XMSC\_IC\_SECURITY\_CREDENTIAL***

### **Veri tipi:**

Dizgi

**Özellik:**

InitialContext

Java Bağlam arabirimi SECURITY\_CREDENTIAL ögesini temel alır. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.

***XMSC\_IC\_SECURITY\_PRINCIPAL*****Veri tipi:**

Dizgi

**Özellik:**

InitialContext

Java Context interface SECURITY\_PRINCIPAL ögesine dayalıdır. Bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.

***XMSC\_IC\_SECURITY\_PROTOCOL*****Veri tipi:**

Dizgi

**Özellik:**

InitialContext

Java Context interface SECURITY\_PROTOCOL ögesine dayalı olarak, bu özellik yalnızca COS adlandırma bağlamı için geçerlidir.

***XMLC\_IC\_URL*****Veri tipi:**

Dizgi

**Özellik:**

InitialContext

LDAP ve FileSystem bağlamları için, denetlenen nesnelere içeren havuzun adresi.

COS adlandırma bağlamları için, dizindeki nesnelere yukarı bakan web hizmetinin adresi.

***XMSC\_IS\_SUBSCRIPTION\_MULTICAST*****Veri tipi:**

System.Boolean

**Özellik:**

MessageConsumer

Indicates whether messages are being delivered to the message consumer using WebSphere MQ Multicast Transport. Bu özellik salt okunurdur.

İletiler WebSphere MQ Multicast Transport kullanılarak ileti tüketicisine teslim ediliyorsa özelliğin değeri true (doğru) değeridir. Ters durumda, değeri false (yanlış) olur.

Bu özellik, yalnızca bir aracıya gerçek zamanlı bağlantı için geçerlidir.

***XMSC\_IS\_SUBSCRIPTION\_RELIABLE\_MULTICAST*****Veri tipi:**

System.Boolean

**Özellik:**

MessageConsumer

İletilerin, güvenilir bir hizmet kalitesi ile WebSphere MQ Multicast Transport kullanılarak ileti tüketicisine teslim edilmekte olup olmadığını belirtir. Bu özellik salt okunurdur.



İletiler güvenilir bir hizmet kalitesi ile WebSphere MQ Multicast Transport kullanılarak ileti tüketicisine teslim ediliyorsa, özelliğin değeri true olur. Ters durumda, değer false (yanlış) olur.

Bu özellik, yalnızca bir aracıya gerçek zamanlı bağlantı için geçerlidir.

### ***XMLC\_JMS\_MAJOR\_VERSION***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionMetaVerileri

XMS ' in temel aldığı JMS belirtiminin ana sürüm numarası. Bu özellik salt okunurdur.

### ***XMLC\_JMS\_MINOR\_VERSION***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionMetaVerileri

XMS ' in dayalı olduğu JMS belirtiminin ikincil sürüm numarası. Bu özellik salt okunurdur.

### ***XMLC\_JMS\_VERSION***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionMetaVerileri

XMS ' in dayandığı JMS belirtiminin sürüm tanıtıcısı. Bu özellik salt okunurdur.

### ***XMLC\_MAJOR\_SÜRÜMÜ***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionMetaVerileri

XMS istemcisinin sürüm numarası. Bu özellik salt okunurdur.

### ***XMLC\_MINOR\_SÜRÜMÜ***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionMetaVerileri

XMS istemcisinin yayın numarası. Bu özellik salt okunurdur.

### ***XML\_PAROLA***

**Veri tipi:**

Bayt dizisi

**Özellik:**

ConnectionFactory

Bir ileti sistemi sunucusuna bağlanma girişiminde bulunduğu uygulamaların kimliğini doğrulamak için kullanılabilir parola. Parola, [XMLC\\_USERID](#) özelliği ile kullanılır.

Varsayılan olarak özellik ayarlanmaz.

Çoklu platformlar'ta WebSphere MQ ' a bağlanıyorsanız ve bağlantı üreticisinin XMLC\_USERID özelliğini ayarlıyorsanız, bu, oturum açmış kullanıcının **userid** ile eşleşmesi gerekir. Bu özellikleri ayarlamadıysanız, kuyruk yöneticisi varsayılan olarak oturum açmış olan kullanıcının **userid** değerini kullanır. Tek tek kullanıcıların daha fazla bağlantı düzeyinde kimlik doğrulamasına gereksinim duyarsanız, WebSphere MQ içinde yapılandırılmış bir istemci kimlik doğrulaması çıkışı yazabilirsiniz. WebSphere MQ İstemcileri elkitabında Kimlik Doğrulaması konusunda bir istemci kimlik doğrulaması çıkışı oluşturma hakkında daha fazla bilgi edinebilirsiniz.

z/OS 'ta WebSphere MQ ' a bağlanırken kullanıcının kimliğini doğrulamak için bir güvenlik çıkışı kullanmanız gerekir.

## ***XMLC\_PRIORITY***

### **Veri tipi:**

System.Int32

### **Özellik:**

Hedef

### **Bir URI ' de kullanılan ad:**

öncelik

Hedefe gönderilen iletilerin önceliği.

Özelliğin geçerli değerleri şunlardır:

#### **Geçerli değer**

0 aralığında bir tamsayı, en düşük öncelik 9 'a, en yüksek önceliğe

#### **Anlamı**

Hedefe gönderilen bir ileti, belirtilen önceliğe sahip. İleti üreticisinin varsayılan önceliği ya da Gönderme çağrısında belirlenen öncelik sayısı dikkate alınmaz. If the destination is a WebSphere MQ queue, the value of the queue attribute **DefPriority** is also ignored.

XMLC\_PRIORITY\_AS\_APP

Hedefe gönderilen bir ileti, Gönderme çağrısında belirtilen önceliğe sahiptir. Gönderme çağrısı herhangi bir öncelik belirtmezse, bunun yerine ileti üreticisinin varsayılan önceliği kullanılır. Hedef bir WebSphere MQ kuyruğuysa, **DefPriority** kuyruk özniteliğinin değeri yoksayılır.

XMLC\_PRIORITY\_AS\_DEST

Hedef bir WebSphere MQ kuyruksa, kuyruğa konan bir ileti, **DefPriority** kuyruk özniteliğinin değeri tarafından belirtilen önceliğe sahip olur. İleti üreticisinin varsayılan önceliği ya da Gönderme çağrısında belirlenen öncelik sayısı dikkate alınmaz.

Hedef bir WebSphere MQ kuyruğu değilse, anlam, XMLC\_PRIORITY\_AS\_APP ile aynı olur.

Varsayılan değer XMLC\_PRIORITY\_AS\_APP ' dir.

WebSphere MQ Gerçek Zamanlı İletim ve WebSphere MQ Multicast Transport , bir iletinin önceliğine dayalı olarak hiçbir işlem kabul etmemez.

## ***XMLC\_PROVIDER\_NAME***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionMetaVerileri

XMS istemcisi sağlayıcısı. Bu özellik salt okunurdur.

## ***XMLC\_RTT\_BROKER\_PING\_INTERVAL***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionFactory

XMS .NET ' in herhangi bir etkinliği algılamak için Real Time Messaging Server bağlantısını denetlemesinden sonra geçen zaman aralığı (milisaniye). Herhangi bir etkinlik algılanmıyorsa, istemci ping işlemi başlatır; ping işlemi için herhangi bir yanıt algılanmıyorsa bağlantı kapatılır.

Özelliğin varsayılan değeri 30000 'dir.

## ***XMSC\_RTT\_CONNECTION\_PROTOCOL***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionFactory

Bir aracıya gerçek zamanlı bağlantı için kullanılan iletişim protokolü.

Özelliğin değeri XMLC\_RTT\_CP\_TCP olmalıdır; bu, TCP/IP üzerinden bir aracıya gerçek zamanlı bağlantı anlamına gelir. Varsayılan değer XMSC\_RTT\_CP\_TCP ' dir.

## ***XMLC\_RTT\_HOST\_NAME***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Bir aracının çalıştığı sistemin anasistem adı ya da IP adresi.

Bu özellik, aracıyı tanımlamak için [XMLC\\_RTT\\_PORT](#) özelliği ile kullanılır.

Varsayılan olarak özellik ayarlanmaz.

## ***XMLC\_RTT\_LOCAL\_ADRESI***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Bir aracıya gerçek zamanlı bağlantı için kullanılacak yerel ağ arabiriminin anasistem adı ya da IP adresi.

Bu özellik yalnızca, uygulamanın çalıştığı sistemde iki ya da daha çok ağ arabirimi varsa ve gerçek zamanlı bağlantı için hangi arabirimin kullanılması gerektiğini belirtmenize gerek duyarsa yararlıdır. Sistemde yalnızca bir ağ arabirimi varsa, yalnızca o arabirim kullanılabilir. Sistemde iki ya da daha çok ağ arabirimi varsa ve özellik ayarlanmadıysa, arabirim rasgele seçilir.

Varsayılan olarak özellik ayarlanmaz.

## ***XMLC\_RTT\_MULTICAST***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionFactory ve Hedef

**Bir URI ' de kullanılan ad:**

multicast

Bir bağlantı üreticisi ya da hedefi için çoklu yayın ayarı. Yalnızca bir konu olan bir hedef bu özelliğe sahip olabilir.

Bir uygulama bu özelliği, bir aracıyla gerçek zamanlı bir bağlantıyla ilişki içinde çoklu yayınlamayı etkinleştirmek için kullanır ve çoklu yayın etkinse, aracıdan ileti tüketicisi için ileti göndermek için çok hedefli bir şekilde kullanılan yolu belirtmek için kullanılır. Özelliğin, bir ileti üreticinin aracıya ileti göndermesine ilişkin bir etkisi yoktur.

Özelliğin geçerli değerleri şunlardır:

#### **Geçerli değer**

XMSC\_RTT\_MULTICAST\_DISABLEAL

#### **Anlamı**

İletiler, WebSphere MQ Multicast Transportkullanılarak ileti tüketicisine teslim edilmiyor. Bu değer, bir ConnectionFactory nesnesi için varsayılan değerdir.

XMLC\_RTT\_MULTICAST\_ASCF

İletiler, ileti tüketicisi ile ilişkili bağlantı üreticisine ilişkin çoklu yayın ayarına göre ileti tüketicisi olarak teslim edilir. Bağlantı oluşturucuna ilişkin çoklu yayın ayarı, bağlantının oluşturulduğu sırada belirtilir. Bu değer yalnızca bir Hedef nesne için geçerlidir ve Hedef nesne için varsayılan değerdir.

XMSC\_RTT\_MULTICAST\_ENABLED

Konu, aracıda çoklu yayın için yapılandırıldıysa, iletiler WebSphere MQ Multicast Transportkullanılarak bir ileti tüketicisine teslim edilir. Konu güvenilir çoklu yayın için yapılandırıldıysa, güvenilir bir hizmet kalitesi kullanılır.

XMSC\_RTT\_MULTICAST\_GÜVENİLİR

Konu, aracıda güvenilir çoklu yayın için yapılandırıldıysa, iletiler güvenilir bir hizmet kalitesi ile WebSphere MQ Multicast Transport kullanılarak bir ileti tüketicisine teslim edilir. Konu güvenilir çoklu yayın için yapılandırılmamışsa, konu için bir ileti tüketicisi yaratamazsınız.

XMSC\_RTT\_MULTICAST\_NOT\_GÜVENİLİR

Konu, aracıda çoklu yayın için yapılandırıldıysa, iletiler WebSphere MQ Multicast Transportkullanılarak bir ileti tüketicisine teslim edilir. Bu konu güvenilir çoklu yayın için yapılandırılmış olsa da, güvenilir bir hizmet kalitesi kullanılmamaktadır.

### ***XMLC\_RTT\_KAPISI***

#### **Veri tipi:**

System.Int32

#### **Özellik:**

ConnectionFactory

Bir aracının gelen istekleri dinlediği kapı numarası. Aracıda, bu kapıyı dinlemek için Real-timeInput ya da Real-timeOptimizedAkış iletisi işleme düğümünü yapılandırmalısınız.

Bu özellik, aracıyı tanıtmak için XMLC\_RTT\_HOST\_NAME özelliği ile kullanılır.

Özelliğin varsayılan değeri XMLC\_RTT\_DEFAULT\_PORT ya da 1506 'tur.

## **XMLC\_TIME\_TO\_LIVE**

### **Veri tipi:**

System.Int32

### **Özellik:**

Hedef

### **Bir URI ' de kullanılan ad:**

süre bitimi ( WebSphere MQ hedefi için)

timeToLive (bir WebSphere varsayılan ileti sistemi sağlayıcısı hedefi için)

Hedefe gönderilen iletiler için canlanacak süre.

Özelliğin geçerli değerleri şunlardır:

#### **Geçerli değer**

0

Pozitif bir tamsayı

#### **Anlamı**

Hedefe gönderilen bir iletinin süresi hiçbir zaman sona ermez.

Hedefe gönderilen bir iletinin milisaniye olarak (milisaniye) yaşaması için belirtilen süre var. İleti üreticisinin varsayılan süresi ya da gönderme çağrısında belirlenen herhangi bir süre için varsayılan süre yoksayılr.

XMLC\_TIME\_TO\_LIVE\_AS\_APP

Hedefe gönderilen bir iletinin, Gönderme çağrısında belirlenen süre içinde yaşama süresi vardır. Gönderme çağrısı, yaşamak için zaman belirtmezse, bunun yerine ileti üreticisinin yaşaması için varsayılan süre kullanılır.

Varsayılan değer XMLC\_TIME\_TO\_LIVE\_AS\_APP ' dir.

## **XMLC\_USERID**

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Bir ileti alışverişi sunucusuna bağlanmayı denediğinde uygulamanın kimliğini doğrulamak için kullanılacak bir kullanıcı kimliği. Kullanıcı kimliği XMLC\_PARPPARE özelliğiyle kullanılır.

Varsayılan olarak özellik ayarlanmaz.

**Multi** IBM MQ for Multiplatforms' a bağlanıyorsanız ve bağlantı üreticisinin XMLC\_USERID özelliğini ayarlıyorsanız, bu, oturum açmış kullanıcının **userid** ile eşleşmesi gerekir. Bu özellikleri ayarlamadıysanız, kuyruk yöneticisi varsayılan olarak oturum açmış olan kullanıcının **userid** değerini kullanır. Tek tek kullanıcıların daha fazla bağlantı düzeyinde kimlik doğrulamasına gereksinim duyarsanız, IBM MQ içinde yapılandırılmış bir istemci kimlik doğrulama çıkışı yazabilirsiniz.

**z/OS** IBM MQ for z/OS ' a bağlanırken kullanıcının kimliğini doğrulamak için bir güvenlik çıkışı kullanmanız gerekir.

## **XMLSC\_FORT**

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionMetaVerileri

XMS istemcisinin sürüm tanıtıcısı. Bu özellik salt okunurdur.

## ***XMLC\_WMQ\_BROKER\_CONTROLQ***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Bir aracı tarafından kullanılan denetim kuyruğunun adı.

Özelliğın varsayılan değeri SYSTEM.BROKER.CONTROL.QUEUE.

Bu özellik, yalnızca yayınlama/abone olma etki alanında anlamlıdır.

## ***XMLC\_WMQ\_BROKER\_PUBQ***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Uygulamaların yayınlayacağı iletileri gönderdikleri bir aracı tarafından izlenen kuyruğun adı.

Özelliğın varsayılan değeri SYSTEM.BROKER.DEFAULT.STREAM.

Bu özellik, yalnızca yayınlama/abone olma etki alanında anlamlıdır.

## ***XMLC\_WMQ\_BROKER\_QMGR***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Bir aracının bağlandığı kuyruk yöneticisinin adı.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, yalnızca yayınlama/abone olma etki alanında anlamlıdır.

## ***XMLC\_WMQ\_BROKER\_SUBQ***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Kalıcı olmayan ileti tüketicisi için abone kuyruğunun adı.

Abone kuyruğunun adı aşağıdaki karakterlerle başlamalıdır:

SYSTEM.JMS.ND.

Tüm kalıcı olmayan ileti okuyucularının bir abone kuyruğunu paylaşmasını istiyorsanız, paylaşılan kuyruğun tam adını belirtin. Uygulamanın kalıcı olmayan bir ileti tüketicisi oluşturabilmesi için, belirtilen adı taşıyan bir kuyruk var olmalıdır.

Her kalıcı olmayan ileti tüketicisinin iletileri kendi özel abone kuyruğundan almasını istiyorsanız, bir yıldız işareti (\*) ile biten bir kuyruk adı belirtin. Daha sonra, bir uygulama dayanıklı olmayan bir ileti tüketicisi yarattığında, XMS istemcisi ileti tüketicisi tarafından özel kullanım için dinamik bir kuyruk yaratır. XMS istemcisi, dinamik kuyruğu yaratmak için kullanılan nesne tanımlayıcısında yer alan **DynamicQName** alanının içeriğini ayarlamak için özelliğın değerini kullanır.

Özelliğın varsayılan değeri SYSTEM.JMS.ND.SUBSCRIBER.QUEUE; bu, XMS ' un varsayılan olarak paylaşılan kuyruk yaklaşımını kullandığı anlamına gelir.

Bu özellik, yalnızca yayınlama/abone olma etki alanında anlamlıdır.

## ***XMLC\_WMQ\_BROKER\_VERSION***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory ve Hedef

### **Bir URI ' de kullanılan ad:**

brokerVersion

Bir bağlantı için ya da hedef için uygulama tarafından kullanılan aracı tipi. Yalnızca bir konu olan bir hedef bu özelliğe sahip olabilir.

Özelliğin geçerli değerleri şunlardır:

### **Geçerli değer**

XMSC\_WMQ\_BROKER\_V1

### **Anlamı**

Uygulama bir WebSphere MQ Yayınla/Abone Ol aracısını kullanıyor.

Uygulama, WebSphere MQ Yayınla/Abone Ol 'tan WebSphere Message Broker ' e geçiş yapsanız da uygulamayı değiştirmediyse, bu değeri de kullanabilir.

XMSC\_WMQ\_BROKER\_V2

Uygulama, IBM Integration Bus aracısını kullanıyor.

XMSC\_WMQ\_BROKER\_BELIRTIOMEMIŞ

Aracı yeni düzeye geçirildikten sonra, RFH2 üstbilgilerinin artık kullanılmaması için bu özelliği ayarlayın. Geçişten sonra, bu özellik artık ilgili değildir.

Bir connectionfactory için varsayılan değer XMSC\_WMQ\_BROKER\_BELIRTIOMEMIŞ, ancak varsayılan olarak, özellik bir hedef için ayarlanmamış demektir. Bir hedef için özelliğin ayarlanması, bağlantı üreticisi özelliği tarafından belirtilen herhangi bir değeri geçersiz kılar.

## ***XMLC\_WMQ\_CCDURL***

### **Veri tipi:**

System.String

### **Özellik:**

ConnectionFactory

### **Uygulanabilir nesnelere:**

JMS yönetim aracı uzun adı: CCDURL

JMS yönetim aracı kısa adı: CCDT

İstemci kanal tanımlama çizelgesini içeren dosyanın adını ve yerini tanımlayan ve aynı zamanda dosyanın nasıl erişilebileceğini belirten bir URL adresi.

Varsayılan olarak bu özellik belirlenmez.

## ***XMLC\_WMQ\_CCSID***

### **Veri tipi:**

System.Int32

### **Özellik:**

Hedef

### **Bir URI ' de kullanılan ad:**

CCSID

Kodlanmış karakter takımının ya da kod sayfasının tanıtcısı (CCSID), bir iletinin gövdesindeki karakter verilerinin dizgilerinin XMS istemcisi iletiyi hedefe ilettiğinde yer alıyor. Tek bir ileti için ayarlanırsa, JMS\_IBM\_CHARACTER\_SET özelliği, bu özellik tarafından hedef için belirtilen CCSID ' yi geçersiz kılar.

Özelliğın varsayılan değeri 1208 'dir.

Bu özellik, hedeften alınan iletilere değil, yalnızca hedefe gönderilen iletilerdir.

### ***XMLC\_WMQ\_CHANNEL***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

**Uygulanabilir nesnelere:**

JMS yönetim aracı uzun adı: KANAL

JMS yönetim aracı kısa adı: CHAN

Bağlantı için kullanılacak kanalın adı.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, yalnızca bir uygulama istemci kipinde bir kuyruk yöneticisine bağlandığında anlamlıdır.

### ***XMLC\_WMQ\_CLIENT\_RECONNECT\_OPTIONS***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

**Uygulanabilir nesnelere:**

JMS yönetim aracı uzun adı: CLIENTRECONNECTORTIONS

JMS denetim aracı kısa adı: CROPT

Bu özellik, bu üretici tarafından yaratılan yeni bağlantılara ilişkin istemci yeniden bağlanma seçeneklerini belirler. XMSC içinde bulunur ve aşağıdakilerden biridir:

- WMQ\_CLIENT\_RECONNECT\_AS\_DEF (varsayılan). `mqcClient.ini` dosyasında belirtilen değeri kullanın. Channels stanza içindeki **DefRecon** özelliğini kullanarak değeri ayarlayın. Aşağıdakilerden birine ayarlanabilir:
  1. EVET. WMQ\_CLIENT\_RECONNECT seçeneği olarak işlev görür
  2. No. Varsayılan. Herhangi bir yeniden bağlanma seçeneği belirtmiyor
  3. QMGR. WMQ\_CLIENT\_RECONNECT\_Q\_MGR seçeneği olarak işlev görür
  4. Geçersiz kılındı. WMQ\_CLIENT\_RECONNECT\_DISABLED seçeneği olarak işlev görür
- WMQ\_CLIENT\_RECONNECT. Bağlantı adı listesinde belirtilen kuyruk yöneticilerinden herhangi birine yeniden bağlanın.
- WMQ\_CLIENT\_RECONNECT\_Q\_MGR. Başlangıçta bağlı olduğu kuyruk yöneticisine yeniden bağlanır. Bağlanma girişiminde bulunursa (bağlantı adı listesinde belirtilen) kuyruk yöneticisinin başlangıçta bağlı kuyruk yöneticisine farklı bir QMID değeri varsa, bu değer MQRC\_RECONNECT\_QMID\_MISMATCH değerini döndürür.
- WMQ\_CLIENT\_RECONNECT\_DEVRE dışı. Yeniden bağlantı devre dışı bırakıldı.

### ***XMLC\_WMQ\_CLIENT\_RECONNECT\_TIMEOUT***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

**Uygulanabilir nesnelere:**

JMS yönetim aracı uzun adı: CLIENTRECONNECTTIMEOUT



JMS yönetim aracı kısa adı: CRT

XMLC\_WMQ\_CLIENT\_RECONNECT\_TIMEOUT özelliği yalnızca Yönetilen XMS .NET istemcisi için geçerlidir.

Bu özellik, bir istemci bağlantısının yeniden bağlanmayı denediği süreyi (saniye olarak) belirtir.

Bu süre için yeniden bağlanma girişiminde bulunduktan sonra, istemci MQRC\_RECONNECT\_FAILED ile başarısız olur. Bu özelliğe ilişkin varsayılan ayar XMSC.WMQ\_CLIENT\_RECONNECT\_TIMEOUT\_DEFAULT.

Bu özelliğin varsayılan değeri 1800 'dür.

## ***XMLC\_WMQ\_CONNECTION\_MODE***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory

Bir uygulamanın kuyruk yöneticisine bağlanmasını sağlayan kip.

Özelliğin geçerli değerleri şunlardır:

<b>Geçerli değer</b>	<b>Anlamı</b>
XMLC_WMQ_CM_BINDINGS	En iyi performans için, bağ tanımları kipinde bir kuyruk yöneticisine yönelik bağlantı. Bu değer, C/C + + için varsayılan değerdir.
XMLC_WMQ_CM_İSTEMCENI	Tam olarak yönetilen bir yığını sağlamak için istemci kipinde bir kuyruk yöneticisine yönelik bağlantı. Bu değer, .NET için varsayılan değerdir.
XMLC_WMQ_CM_CLIENT_UNCEOED (yalnızca .NET için)	Yönetilmeyen bir istemci yığınına zorlayan bir kuyruk yöneticisine yönelik bağlantı.

### **İlgili kavramlar**

.NET içindeki yönetilen ve yönetilmeyen işlemler

Yönetilen kod, özel olarak .NET ortak dil çalıştırma zamanı ortamı içinde yürütülür ve bu çalıştırma zamanı tarafından sağlanan hizmetlere tamamen bağımlıdır. Uygulamanın herhangi bir bölümü .NET ortak dil çalıştırma zamanı ortamı dışında hizmet çalıştırdığında ya da çağırıyorsa, uygulama, yönetilmeyen bir uygulama olarak sınıflandırılır.

## ***XMLC\_WMQ\_CONNECTION\_NAME\_LIST***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

### **Uygulanabilir nesnelere:**

JMS denetim aracı uzun adı: CONNECTIONNAMELIST

JMS yönetim aracı kısa adı: CNLIST

Bu özellik, istemcinin bağlantısı kesildikten sonra istemcinin yeniden bağlanmayı denediği anasistemleri belirler.

Bağlantı adı listesi, anasistem/ip kapı çiftlerinin virgülle ayrılmış listesidir. Bu özelliğe ilişkin varsayılan ayar WMQ\_CONNECTION\_NAME\_LIST\_VARSAYILIDIR.

Örneğin,127.0.0.1 (1414) ,host2.example.com(1400)

Bu özelliğin varsayılan ayarı localhost (1414)' dir.

## ***XMLC\_WMQ\_DUR\_SUBQ***

### **Veri tipi:**

Dizgi

### **Özellik:**

Hedef

Hedeften ileti alan dayanıklı bir aboneye ilişkin abone kuyruğunun adı. Yalnızca bir konu olan bir hedef bu özelliğe sahip olabilir.

Abone kuyruğunun adı aşağıdaki karakterlerle başlamalıdır:

SYSTEM.JMS.D.

Tüm dayanıklı abonelerin bir abone kuyruğunu paylaşmasını istiyorsanız, paylaşılan kuyruğun tam adını belirtin. Uygulamanın kalıcı bir abone oluşturabilmesi için, belirtilen adı taşıyan bir kuyruk var olmalıdır.

Her kalıcı abonenin, iletileri kendi özel abone kuyruğundan almasını istiyorsanız, bir yıldız (\*) ile biten bir kuyruk adı belirtin. Daha sonra, bir uygulama dayanıklı bir abone oluşturduğunda, XMS istemcisi, kalıcı abone tarafından özel kullanım için dinamik bir kuyruk yaratır. XMS istemcisi, dinamik kuyruğu yaratmak için kullanılan nesne tanımlayıcısında yer alan **DynamicQName** alanının içeriğini ayarlamak için özelliğin değerini kullanır.

Özelliğin varsayılan değeri SYSTEM.JMS.D.SUBSCRIBER.QUEUE; bu, XMS ' un varsayılan olarak paylaşılan kuyruk yaklaşımını kullandığı anlamına gelir.

Bu özellik, yalnızca yayınlama/abone olma etki alanında anlamlıdır.

## ***XMLC\_WMQ\_ENCODING***

### **Veri tipi:**

System.Int32

### **Özellik:**

Hedef

Bir iletinin gövdesindeki sayısal veriler, XMS istemcisi iletiyi hedefe ilettiğinde gösterilir. Tek bir ileti için ayarlanmışsa, JMS\_IBM\_ENCODING özelliği, bu özelliğin hedefi için belirtilen kodlamayı geçersiz kılar. Bu özellik, ikili tamsayıların, paketlenmiş onlu tamsayıların ve kayan noktalı sayıların gösterilmesini belirtir.

Özelliğin geçerli değerleri, bir ileti tanımlayıcısının **Encoding** alanında belirtilebilecek değerlerle aynıdır.

Bir uygulama, özelliği ayarlamak için aşağıdaki adı belirtilen değişmezleri kullanabilir:

### **Adlandırılmış sabit**

### **Anlamı**

MQENC\_INTEGER\_NORMAL

Normal tamsayı kodlaması

MQENC\_INTEGER\_TERS

Tersine çevrilmiş tamsayı kodlaması

MQENC\_DECIMAL\_NORMAL

Olağan paketlenmiş onlu kodlama

MQENC\_DECIMAL\_TERDI

Tersine çevrilmiş paketlenmiş ondalık kodlama

MQENC\_FLOAT\_IEEE\_NORMAL

Olağan IEEE kayar noktalı kodlama

MQENC\_FLOAT\_IEEEE\_TERS

Ters IEEE kayar noktalı kodlama

MQENC\_FLOAT\_S390

z/OS mimarisi kayan nokta kodlaması

MQENC\_NATIVE

Yerli makine kodlaması

Özelliğe ilişkin bir değer oluşturmak için, uygulama bu değişmezlerden üçünü aşağıdaki gibi ekleyebilir:

- İkili tamsayıların gösterimini belirtmek için, adı MQENC\_XX\_ENCODE\_CASE\_ONE integer ile başlayan bir değişmez.
- Paketlenmiş onlu tamsayıların gösterimini belirtmek için, adı MQENC\_DECIMAL ile başlayan bir değişmez.

- Yüzer nokta sayılarının gösterimini belirtmek için, adı MQENC\_FLOAT olan bir değişmez.

Diğer bir seçenek olarak, uygulama özelliği, değeri ortama bağlı olan MQENC\_NATIVE değerine ayarlayabilir.

Özelliğin varsayılan değeri MQENC\_NATIVE değeridir.

Bu özellik, hedeften alınan iletilere değil, yalnızca hedefe gönderilen iletilerdir.

## ***XMSC\_WMQ\_FAIL\_IF\_QUIESCE***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory ve Hedef

### **Bir URI ' de kullanılan ad:**

failIfQuiesce

### **Uygulanabilir nesnelere:**

JMS yönetim aracı uzun adı: FAILIFQUIESCE

JMS yönetim aracı kısa adı: FIQ

Uygulamanın bağlı olduğu kuyruk yöneticisi susturma durumdaysa, belirli yöntemlere yapılan çağrıların başarısız olup olmadığını belirler.

Özelliğin geçerli değerleri şunlardır:

### **Geçerli değer**

### **Anlamı**

XMLC\_WMQ\_FIQ\_YES

Kuyruk yöneticisi susturulmuş durumdaysa, bazı yöntemlere yönelik çağrılar başarısız olur. Uygulama kuyruk yöneticisinin susturulmuş olduğunu saptarsa, uygulama hemen görevini tamamlayabilir ve kuyruk yöneticisinin durmasını sağlayarak bağlantıyı kapatabilir.

XMLC\_WMQ\_FIQ\_NO

Kuyruk yöneticisi susturulmuş durumda olduğu için yöntem çağrıları başarısız oldu. Bu değeri belirtirseniz, uygulama kuyruk yöneticisinin susturulmuş olduğunu algılayamaz. Uygulama kuyruk yöneticisine yönelik işlemleri gerçekleştirmeye devam edebilir ve bu nedenle kuyruk yöneticisinin durmasını engelleyebilir.

Bir bağlantı üreticisine ilişkin varsayılan değer XMLC\_WMQ\_FIQ\_YES ise, ancak varsayılan olarak, özellik bir hedef için ayarlanmamış demektir. Bir hedef için özelliğin ayarlanması, bağlantı üreticisi özelliği tarafından belirtilen herhangi bir değeri geçersiz kılar.

## ***XMLC\_WMQ\_MESSAGE\_BÖLGESİ***

### **Veri tipi:**

System.Int32

### **Özellik:**

Hedef

Bu özellik, bir XMS uygulamasının bir IBM WebSphere MQ iletisinin MQRFH2 ' yi ileti bilgi yükünün (ileti gövdesinin bir parçası olarak) bir parçası olarak işleip işlemediğini belirler.

**Not:** Bir hedefe ileti gönderirken, XMLC\_WMQ\_MESSAGE\_BOLD özelliği var olan XMS Hedef özelliği XMLC\_WMQ\_TARGET\_CLIENT ögesini geçersiz kılar.

Bu özelliğe ilişkin geçerli değerler şunlardır:

### **XMLC\_WMQ\_MESSAGE\_BODY\_JMS**

**Alma:** Gelen XMS ileti tipi ve gövdesi, alınan IBM WebSphere MQ iletisinde MQRFH2 (varsa) ya da MQMD ( MQRFH2 yoksa) içeriğiyle belirlenir.

**Gönder:** Giden XMS ileti gövdesi, XMS İleti özellikleri ve üstbilgi alanlarına dayalı olarak önlenmiş ve otomatik olarak oluşturulmuş bir MQRFH2 üstbilgisi içerir.

### **XMLC\_WMQ\_MESSAGE\_BODY\_MQ**

**Receive:** Gelen XMS ileti tipi her zaman ByteMessage'dır, alınan IBM WebSphere MQ iletisinin içeriğinden bağımsız olarak ya da alınan MQMD' nin biçim alanı olur. XMS ileti gövdesi, temeldeki ileti sistemi sağlayıcısı API çağrısı tarafından döndürülen değiştirilmemiş ileti verileridir. İleti gövdesindeki verilerin karakter takımı ve kodlaması, MQMD ' nin CodedCharSetId ve Encoding alanları tarafından belirlenir. İleti gövdesindeki verilerin biçimi, MQMD ' nin Biçim alanı tarafından belirlenir.

**Gönder:** Giden XMS ileti gövdesi, uygulama bilgi yükünü is; olarak içerir; ve gövdeye otomatik olarak oluşturulan IBM WebSphere MQ üstbilgisi eklenmez.

### **XMSSC\_WMQ\_MESSAGE\_BODY\_BELIRTILMEMİŞ**

**Alma:** XMS istemcisi, bu özellik için uygun bir değer belirler. Alma yolunda, bu değer WMQ\_MESSAGE\_BODY\_JMS özellik değeridir.

**Gönder:** XMS istemcisi bu özellik için uygun bir değer belirler. Gönderme yolunda bu değer XMLC\_WMQ\_TARGET\_CLIENT özellik değeri olur.

Varsayılan olarak bu özellik XMSSC\_WMQ\_MESSAGE\_BODY\_BELIRTILMEMİŞ olarak ayarlanır.

### **XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT**

#### **Veri tipi:**

System.Int32

#### **Özellik:**

Hedef

Determines what level of message context is to be set by the XMS application. Uygulamanın, bu özelliğin geçerli olması için uygun bağlam yetkisi ile çalışıyor olması gerekir.

Bu özelliğe ilişkin geçerli değerler şunlardır:

#### **XMLC\_WMQ\_MDCTX\_VARSAYILANı**

Giden iletiler için, MQOPER API çağrısı ve MQPMO yapısı belirttik ileti bağlamı seçeneklerini belirtmez.

#### **XMLC\_WMQ\_MDCTX\_SET\_IDENTITY\_CONTEXT**

MQOPER API çağrısı, MQOO\_SET\_IDENTITY\_CONTEXT ileti bağlamı seçeneğini belirtir ve MQPMO yapısı MQPMO\_SET\_IDENTITY\_CONTEXT belirtisini belirtir.

#### **XMLC\_WMQ\_MDCTX\_SET\_ALL\_CONTEXT**

MQOPEP API çağrısı, MQOO\_SET\_ALL\_CONTEXT ileti bağlamı seçeneğini belirtir ve MQPMO yapısı MQPMO\_SET\_ALL\_CONTEXT ' yi belirtir.

Varsayılan olarak bu özellik XMSC\_WMQ\_MDCTX\_default değerine ayarlıdır.

**Not:** Bir uygulama Sistem Integration Bus' a bağlandığında bu özellik ilgili değildir.

Aşağıdaki özellikler, aranan etkinin olması için ileti gönderirken XMLC\_WMQ\_MDCTX\_SET\_IDENTITY\_CONTEXT ya da XMLC\_WMQ\_MDCTX\_SET\_ALL\_CONTEXT özellik değerine ayarlanacak XMSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT özelliğini gerektirir:

- JMS\_IBM\_MQMD\_USERIDENTIFIER
- JMS\_IBM\_MQMD\_ACCOUNTINGTOKEN
- JMS\_IBM\_MQMD\_APPLIDENTITDATA

Aşağıdaki özellikler, aranan etkinin olması için ileti gönderirken XMLC\_WMQ\_MDCTX\_SET\_ALL\_CONTEXT özellik değerine ayarlanacak XMSSC\_WMQ\_MQMD\_MESSAGE\_CONTEXT özelliğinin değerini gerektirir:

- JMS\_IBM\_MQMD\_PUTAPPLTYPE
- JMS\_IBM\_MQMD\_PUTAPPLNAME
- JMS\_IBM\_MQMD\_PUTDATE

- JMS\_IBM\_MQMD\_PUTTIME
- JMS\_IBM\_MQMD\_APPLORIGINDATA

### ***XMLC\_WMQ\_MQMD\_READENABLED***

**Veri tipi:**

System.Int32

**Özellik:**

Hedef

Bu özellik, bir XMS uygulamasının MQMD alanlarının değerlerini ayıklayıp ayıklayamayacağını belirler.

Bu özelliğe ilişkin geçerli değerler şunlardır:

#### **XMLC\_WMQ\_READENABLED\_NO**

İletileri gönderirken, gönderilen bir iletteki JMS\_IBM\_MQMD\* özellikleri, MQMD ' deki güncellenen alan değerlerini yansıtacak şekilde güncellenmez.

İleti alınırken, bazı ya da tümü gönderen tarafından ayarlanmış olsa da, alınan bir iletide JMS\_IBM\_MQMD\* özelliklerinin hiçbiri kullanılabilir değil.

#### **XMLC\_WMQ\_READ\_ENABLED\_YES**

İletileri gönderirken, gönderilen bir iletteki tüm JMS\_IBM\_MQMD\* özellikleri, gönderenin belirttik olarak ayarlamadığı özellikler de içinde olmak üzere, MQMD ' deki güncellenmiş alan değerlerini yansıtacak şekilde güncellenir.

İleti alınırken, tüm JMS\_IBM\_MQMD\* özellikleri, gönderenin belirttik olarak ayarlamadığı özellikler de içinde olmak üzere, alınan bir iletlerde kullanılabilir.

Varsayılan olarak bu özellik XMSC\_WMQ\_READ\_ENABLED\_NO olarak ayarlıdır.

### ***XMSSC\_WMQ\_MQMD\_WRITE\_ENABLED***

**Veri tipi:**

System.Int32

**Özellik:**

Hedef

Bu özellik, bir XMS uygulamasının MQMD alanlarının değerlerini alıp veremeyeceğini belirler.

Bu özelliğe ilişkin geçerli değerler şunlardır:

#### **XMLC\_WMQ\_WRITE\_ENABLED\_NO**

Tüm JMS\_IBM\_MQMD\* özellikleri yoksayılr ve değerleri, temeldeki MQMD yapısıyla kopyalanmaz.

#### **XMLC\_WMQ\_WRITE\_ENABLED\_YES**

JMS\_IBM\_MQMD\* özellikleri işlendi. Değerleri, temeldeki MQMD yapısına kopyalanır.

Varsayılan olarak bu özellik XMLC\_WMQ\_WRITE\_ENABLED\_NO olarak ayarlıdır.

### ***XMSC\_WMQ\_PUT\_ASYNC\_ALLOWES***

**Veri tipi:**

System.Int32

**Özellik:**

Hedef

Bu özellik, ileti üreticilerinin bu hedefe ileti göndermek için zamanuyumsuz yerleştirmeleri kullanmalarına izin verilip verilmeyeceğini belirler.

Bu özelliğe ilişkin geçerli değerler şunlardır:

**XMLC\_WMQ\_PUT\_ASYNC\_ALLOWED\_AS\_DEST**

Kuyruğa ya da konu tanımlamasına gönderme yaparak zamanuyumsuz yerleştirmeye izin verilip verilmediğini saptayın.

**XMLC\_WMQ\_PUT\_ASYNC\_ALLOWED\_AS\_Q\_DEF**

Kuyruk tanımlamasına gönderme yaparak zamanuyumsuz yerleştirmeye izin verilip verilmediğini saptayın.

**XMLC\_WMQ\_PUT\_ASYNC\_ALLOWED\_AS\_TOPIC\_DEF**

Konu tanımlamasına gönderme yaparak zamanuyumsuz yerleştirmeye izin verilip verilmediğini saptayın.

**XMLC\_WMQ\_PUT\_ASYNC\_ALLOWED\_DISABET**

Zamanuyumsuz yerleştirmeye izin verilmez.

**XMLC\_WMQ\_PUT\_ASYNC\_ALLOWED\_ENABLED**

Zamanuyumsuz yerleştirmeye izin verilir.

Varsayılan olarak bu özellik XMSC\_WMQ\_PUT\_ASYNC\_ALLOWED\_AS\_DEST olarak ayarlıdır.

**Not:** Bir uygulama Sistem Integration Bus' a bağlanırken bu özellik ilgili değildir.

**XMSC\_WMQ\_READ\_AHEAD\_ALLOWED****Veri tipi:**

System.Int32

**Özellik:**

Hedef

Bu özellik, ileti tüketicilerinin ve kuyruk tarayıcılarının, bu hedeften kalıcı olmayan, hareket dışı iletileri almadan önce bu hedeften iç arabelleğe alınmak için önden okuma kullanmasına izin verilip verilmeyeceğini belirler.

Bu özelliğe ilişkin geçerli değerler şunlardır:

**XMSSC\_WMQ\_READ\_AHEAD\_ALLOWED\_AS\_Q\_DEF**

İleriye okuma işlemine, kuyruk tanımlamasına başvuruda bulunarak izin verilip verilmediğini belirleyin.

**XMSC\_WMQ\_READ\_AHEAD\_ALLOWED\_AS\_KONU\_DEF**

İleriye okumanın, konu tanımlamasına gönderme yaparak izin verilip verilmediğini belirleyin.

**XMSSC\_WMQ\_READ\_AHEAD\_ALLOWED\_AS\_DEST**

İleriye okuma işlemine, kuyruk ya da konu tanımından gönderme yaparak izin verilip verilmediğini belirleyin.

**XMSSC\_WMQ\_READ\_AHEAD\_ALLOWED\_DISABLEAL**

İletileri tüketirken ya da iletirken okuma işlemine izin verilmez.

**XMSC\_WMQ\_READ\_AHEAD\_ALLOWED\_ENABLED**

İlerlemenize izin verilir.

Varsayılan olarak bu özellik XMSC\_WMQ\_READ\_AHEAD\_ALLOWED\_AS\_DEST olarak ayarlıdır.

**XMSC\_WMQ\_READ\_AHEAD\_CLOSE\_POLICY****Veri tipi:**

System.Int32

**Özellik:**

Hedef

Bu özellik, ileti tüketicisi kapatıldığında, zamanuyumsuz ileti dinleyicisine gönderilmekte olan iletilerin, iç okuma yazma arabelleğindeki iletilere ne olacağını belirler.

Bu özellik, bir hedeften iletiler tüketirken ve bir hedefe ileti gönderirken uygulanmadığında, kapanış kuyruğu seçeneklerinin belirlenmesi için geçerlidir.

Bu özellik, iletiler kuyrukta bulunmaya devam etmesinden bu yana, Kuyruk Tarayıcıları için yoksayıdır.

Bu özelliğe ilişkin geçerli değerler şunlardır:

#### **XMSC\_WMQ\_READ\_AHEAD\_CLOSE\_POLICY\_DELIVER\_CURRENT**

Geri dönmeden önce yalnızca yürürlükteki ileti dinleyicisi çağrısı tamamlanır, iç okuma arabelleğindeki iletiler bırakılır ve bu iletiler atılır.

#### **XMSC\_WMQ\_READ\_AHEAD\_CLOSE\_POLICY\_DELIVER\_ALL**

İç okuma yazma arabelleğindeki tüm iletiler, geri dönmeden önce uygulama iletilisi dinleyicisine teslim edilir.

Varsayılan olarak bu özellik XMSC\_WMQ\_READ\_AHEAD\_CLOSE\_POLICY\_DELIVER\_CURRENT olarak ayarlıdır.

#### **Not:**

##### **• Olağandışı uygulama sonlandırması**

Bir XMS uygulaması aniden sona erdiğinde, ileriye okuma arabelleğindeki tüm iletiler kaybedilir.

##### **• İşlemlerde Etkileri**

Uygulamalar hareketi kullandığında, okuma tamamlama özelliği devre dışı bırakılır. Yani, uygulama, hareket eden seansları kullandıklarında hiçbir şekilde bir fark görmeyecek.

##### **• Oturum Onay kiplerinin etkileri**

Alındı bildirim kipleri XMSC\_AUTO\_RELEASE ya da XMSSC\_DUPS\_OK\_RELEASE olduğunda, okuma öncesinde okuma yazma işlemi etkin olmayan bir oturumda etkindir. Oturum onaylaması kipi XMSSC\_CLIENT\_RENDE, hareket eden ya da çevrilemeyen oturumlardan bağımsız olarak XMSSC\_CLIENT\_RENDE, okuma öncesinde okuma işlemi geçersiz kılınır.

##### **• Kuyruk Tarayıcıları ve Kuyruk Tarayıcısı Seçicileri üzerinde Etkileri**

The Queue Browsers and Queue Browser Selectors, used in XMS applications, get the performance advantage from read ahead. Kuyruk Tarayıcısının kapatılması performansı düşürmez, çünkü ileti daha sonraki işlemler için kuyrukta kullanılabilir olmaya devam eder. Kuyruk tarayıcılarında ve kuyruk tarayıcısı seçicilerinin, okuma önlerindeki performans avantajlarından ayrı olarak başka herhangi bir etkisi yoktur.

## ***XMLC\_WMQ\_HOST\_NAME***

#### **Veri tipi:**

Dizgi

#### **Özellik:**

ConnectionFactory

#### **Uygulanabilir nesnelere:**

JMS yönetim aracı uzun adı: HOSTNAME

JMS yönetim aracı kısa adı: HOST

Bir kuyruk yöneticisinin çalıştığı sistemin anasistem adı ya da IP adresi.

Bu özellik, yalnızca bir uygulama istemci kipinde bir kuyruk yöneticisine bağlandığında kullanılır. Bu özellik, kuyruk yöneticisini tanımlamak için [XMLC\\_WMQ\\_PORT](#) özelliğiyle birlikte kullanılır.

Özelliğin varsayılan değeri localhost' dir.

## ***XMLC\_WMQ\_LOCAL\_ADRESİ***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

### **Uygulanabilir nesnelere:**

JMS yönetim aracı uzun adı: LOCALADDRESS

JMS yönetim aracı kısa adı: LA

Bir kuyruk yöneticisine yönelik bir bağlantı için, bu özellik kullanılacak yerel ağ arabirimini ya da kullanılacak yerel ağ kapısını ya da yerel kapı aralığını ya da her ikisini de belirtir.

Özelliğin değeri, aşağıdaki biçimi taşıyan bir dizgidir:

[*anasistem\_adi*] [ (*düşük\_kapısı*) [,*yükseklik\_kapısı*]]

Değişkenlerin anlamları aşağıdaki gibidir:

### ***anasistem\_adi***

Bağlantı için kullanılacak yerel ağ arabiriminin anasistem adı ya da IP adresi.

Bu bilgileri yalnızca, uygulamanın çalıştığı sistemin iki ya da daha çok ağ arabirimi varsa ve bağlantı için hangi arabirimin kullanılması gerektiğini belirtmeniz gerekiyorsa gereklidir. Sistemde yalnızca bir ağ arabirimi varsa, yalnızca o arabirim kullanılabilir. Sistemde iki ya da daha çok ağ arabirimi varsa ve hangi arabirimin kullanılması gerektiğini belirtmezseniz, arabirim rasgele seçilir.

### ***düşük\_kapı***

Bağlantı için kullanılacak yerel kapı numarası.

*high\_port* da belirtilirse, *low\_port* , bir kapı numaralarındaki en düşük kapı numarasını yorumlanır.

### ***yükseklik\_kapısı***

Bir kapı numaralarındaki en yüksek kapı numarası. Bağlantı için belirtilen aralıktaki kapılardan biri kullanılmalıdır.

Dizilimin uzunluk üst sınırı 48 karakterdir.

Aşağıda, özelliğin geçerli değerlerine ilişkin bazı örnekler bulunmaktadır:

JUPITER  
9.20.4.98  
JÜPITER (1000)  
9.20.4.98(1000,2000)  
(1000)  
(1000,2000)

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, yalnızca bir uygulama istemci kipinde bir kuyruk yöneticisine bağlandığında anlamlıdır.

## ***XMSC\_WMQ\_MESSAGE\_SELECTION***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory

İleti seçmesinin XMS istemcisi ya da aracı tarafından yapıp yapılmayacağını belirler.

Özelliğin geçerli değerleri şunlardır:

<b>Geçerli değer</b>	<b>Anlamı</b>
XMLC_WMQ_MSEL_İSTEMCESİ	İleti seçimi XMS istemcisi tarafından yapılır.



<b>Geçerli değer</b>	<b>Anlamı</b>
XMLC_WMQ_MSEL_BROKER	İleti seçimi aracı tarafından yapılır.

Varsayılan değer XMSC\_WMQ\_MSEL\_Client değeridir.

Bu özellik, yalnızca yayınlama/abone olma etki alanında anlamlıdır. Message selection by the broker is not supported if the XMLC\_WMQ\_BROKER\_VERSION property is set to XMSC\_WMQ\_BROKER\_V1.

### ***XMLC\_WMQ\_MSG\_BATCH\_SIZE***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionFactory

Zamanuyumsuz ileti teslimi kullanılırken, bir kuyruktan bir toplu iş kuyruğundan alınacak ileti sayısı üst sınırı.

Bir uygulama, belirli koşullar altında zamanuyumsuz ileti teslimi kullanırken, XMS istemcisi her iletiyi uygulamaya tek tek iletmeye önce bir kuyruktan ileti toplu iş alır. Bu özellik, toplu işte olabilecek ileti sayısı üst sınırını belirtir.

Özelliğin değeri pozitif bir tamsayıdır ve varsayılan değer 10 'tır. Yalnızca ele almanız gereken belirli bir başarımlı sorunuz varsa, özelliği farklı bir değere ayarlamayı düşünün.

Bir uygulama, bir ağ üzerinden bir kuyruk yöneticisine bağlıysa, bu özelliğin değerini yükseltmek ağ başlarını ve yanıt sürelerini azaltabilir, ancak iletilerin istemci sisteminde saklanabilmesi için gereken bellek miktarını artırır. Bunun tersine, bu özelliğin değerinin düşürülmesi ağı, ağ kafalarını ve yanıt sürelerini artırabilir, ancak iletilerin depolması için gereken bellek miktarını azaltabilir.

### ***XMLC\_WMQ\_POLLING\_INTERVAL***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionFactory

Bir oturumdaki her ileti dinleyicisinin kuyruğunda uygun bir ileti yoksa, bu değer, her ileti dinleyicisinin kuyruğundan bir ileti almak için yeniden denemesinden önce geçen, milisaniye cinsinden en yüksek aralığa sahip olur.

Bir oturumda ileti dinleyicilerinin herhangi biri için uygun bir ileti bulunmuyorsa, bu özelliğin değerini artırmayı düşünün.

Özelliğin değeri pozitif bir tamsayıdır. Varsayılan değer 5000 'dir.

### ***XMLC\_WMQ\_PORT***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionFactory

**Uygulanabilir nesnelere:**

JMS yönetim aracı uzun adı: PORT

JMS yönetim aracı kısa adı: PORT

Bir kuyruk yöneticisinin gelen istekleri dinlediği kapı numarası.

Bu özellik, yalnızca bir uygulama istemci kipinde bir kuyruk yöneticisine bağlandığında kullanılır. Bu özellik, kuyruk yöneticisini tanıtmak için XMLC\_WMQ\_HOST\_NAME özelliği ile kullanılır.

Özelliğin varsayılan değeri XMLC\_WMQ\_DEFAULT\_CLIENT\_PORT ya da 1414 'tür.

## ***XMLC\_WMQ\_PROVIDER\_VERSION***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Uygulamanın bağlanmayı amaçladığı kuyruk yöneticisinin sürüm, yayın, değişiklik düzeyi ve düzeltme paketi. Bu özelliğe ilişkin geçerli değerler şunlardır:

- Belirtilmedi

Ya da aşağıdaki biçimlerden birindeki bir dizgi

- V.R.M.F
- V.R.M
- V.R
- V

Burada V, R, M ve F, sıfıra eşit ya da sıfırdan büyük tamsayı değerleridir.

7 ya da daha büyük bir değer, bu sürümün bir IBM WebSphere MQ 7.0 kuyruk yöneticisine bağlantı için bir IBM WebSphere MQ 7.0 ConnectionFactory olarak amaçlandığı anlamına gelir. A value earlier than 7 (for example "6.0.2.0"), indicates that it is intended for use with queue managers earlier than Version 7.0. Varsayılan değer, belirlenmemiş, kuyruk yöneticisinin herhangi bir düzeylerine bağlantı sağlar ve kuyruk yöneticisinin yeteneklerine dayalı olarak uygulanabilir özellikleri ve işlevleri belirler.

Varsayılan olarak bu özellik "belirtilmemiş" olarak ayarlıdır.

### **Not:**

- XMLC\_WMQ\_PROVIDER\_VERSION 6 olarak ayarlandıysa, yuva paylaşımı olmaz. 2.
- XMLC\_WMQ\_PROVIDER\_VERSION 7 değerine ayarlıysa ve kanal için SHARECNV sunucusunda 0 değeri belirlendiğinde bağlantı başarısız olur.
- XMLC\_WMQ\_PROVIDER\_VERSION BELIRTILMEYEN ve SHARECNV olarak ayarlandıysa, IBM WebSphere MQ 7.0 ' a özgü özellikler 0 olarak ayarlanır.

IBM WebSphere MQ Client 'ın sürümü, bir XMS istemci uygulamasının IBM WebSphere MQ 7.0 ' in özel özelliklerini kullanıp kullanamayacağı konusunda önemli rol oynar. Aşağıdaki tablo, davranışı açıklamalı.

**Not:** XMLC\_WMQ\_OVERRIDEPROVIDERVERSION sistem özelliği XMLC\_WMQ\_PROVIDER\_VERSION özelliğini geçersiz kılar. Bağlantı üreticisi ayarını değiştiremiyorsanız bu özellik kullanılabilir.

Çizelge 33. XMS istemcisi- IBM WebSphere MQ 7.0 ' ye özgü özellikleri kullanma yeteneği.

#	XMLC_WMQ_PROVIDER_VERSION	IBM WebSphere MQ İstemci Sürümü	IBM WebSphere MQ 7.0 özellikler
1	belirlenmedi	7	AÇIK
2	belirlenmedi	6	KAPALI
3	7	7	AÇIK
4	7	6	Özel durum
5	6	6	KAPALI
6	6	7	KAPALI

## ***XMLC\_WMQ\_PUB\_ACK\_INTERVAL***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory

Bir yayıncı tarafından XMS istemcisinden önce yayınlanan ileti sayısı, aracından bir alındı bildirimi istemeden önce.

Bu özelliğin değerini azalrsanız, istemci daha sık kabul eder ve dolayısıyla yayıncının performansı azalır. Değeri yükseltiyorsanız, aracı başarısız olursa, istemci daha uzun süre kural dışı durum yayınlamaya devam eder.

Özelliğin değeri pozitif bir tamsayıdır. Varsayılan değer 25 'tür.

## ***XMLC\_WMQ\_QMGR\_CCSID***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory

İleti Kuyruğu Arabirimi 'nde (MQI) tanımlanan karakter verileri alanlarının, XMS istemcisi ile WebSphere MQ istemcisi arasında değiş tokuş edilen kodlanmış karakter takımı ya da kod sayfasının tanıtıcısı (CCSID). Bu özellik, ileti gövdelerindeki karakter verileri dizgileri için geçerli değildir.

When XMS application connects to a queue manager in client mode, the XMS client links to the WebSphere MQ client. İki istemci arasında değiş tokuş edilen bilgiler, MQI ' da tanımlı olan karakter verileri alanlarını içerir. Olağan koşullarda, WebSphere MQ istemcisi, bu alanların, istemcilerin çalıştığı sistemin kod sayfasında olduğunu varsayar. XMS istemcisi bu alanları farklı bir kod sayfasında sağlarsa ve almayı bekliyorsa, bu özelliği WebSphere MQ istemcisine bildirecek şekilde ayarlamanız gerekir.

WebSphere MQ istemcisi karakter verilerinin bu alanlarını kuyruk yöneticisine ilettiğinde, kuyruk yöneticisi tarafından kullanılan kod sayfasında gerekliyse, bunların içindeki veriler dönüştürülmelidir. Similarly, when the WebSphere MQ client receives these fields from the queue manager, the data in them must be converted if necessary into the code page in which the XMS client expects to receive the data. WebSphere MQ istemcisi, bu veri dönüştürmelerini gerçekleştirmek için bu özelliği kullanır.

Varsayılan olarak özellik ayarlanmaz.

Bu özelliğin ayarlanması, yerli WebSphere MQ istemci uygulamalarını destekleyen bir WebSphere MQ istemcisi için MQCCSID ortam değişkeninin ayarına eşdeğerdir. Bu ortam değişkenine ilişkin ek bilgi için *WebSphere MQ İstemcileri* konusuna bakın.

## ***XMLC\_WMQ\_QUEUE\_YÖNETİCİSİ***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

### **Uygulanabilir nesnelere:**

JMS yönetim aracı uzun adı: QYORDER

JMS yönetim aracı kısa adı: QMGR

Bağlanmak için kuyruk yöneticisinin adı.

Varsayılan olarak özellik ayarlanmaz.

## ***XMLC\_WMQ\_RECEIVE\_CC SID***

Kuyruk yöneticisi ileti dönüştürmesi için hedef CCSID ' yi belirleyen hedef özellik.  
XMLC\_WMQ\_RECEIVE\_CONVERSION WMQ\_RECEIVE\_CONVERSION\_QMGR değerine ayarlanmıyorsa, değer yoksayılır.

### **Veri tipi:**

Tamsayı

### **Değer:**

Herhangi bir pozitif tamsayı.

Varsayılan değer 1208 'dir.

İletide GMO\_CONVERT değerinin belirtilmesi isteğe bağlıdır. Bir GMO\_CONVERT değeri belirtilirse, dönüştürme, belirtilen değere göre gerçekleşir.

## ***XMLC\_WMQ\_RECEIVE\_CONVERSION***

Veri dönüştürme işleminin kuyruk yöneticisi tarafından gerçekleştirilip gerçekleştirilmeyeceğini belirleyen hedef özellik.

### **Veri tipi:**

Tamsayı

### **Değerler:**

XMLC\_WMQ\_RECEIVE\_CONVERSION\_CLIENT\_MSG (DEFAULT): Yalnızca XMS istemcisine veri dönüştürme işlemi gerçekleştirin. Dönüştürme her zaman 1208 kod sayfası kullanılarak yapılır.

XMLC\_WMQ\_RECEIVE\_CONVERSION\_QMGR: XMS istemcisine ileti göndermeden önce kuyruk yöneticisine veri dönüştürme işlemi gerçekleştirin.

## ***XMLC\_WMQ\_RECEIVE\_EXIT***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Bir kanal alma çıkışının çalıştırılacağı tanıtılıyor.

Özelliğin değeri, bir kanal alma çıkışını tanımlayan bir dizgidir ve aşağıdaki biçime sahiptir:

**libraryName**(entryPointAd)

Burada,

- **libraryName** , yönetilen çıkışa ilişkin '.dll' in tam yoludur.
- **entryPointAd** , ad alanı tarafından nitelenen sınıf adıdır

Örneğin, C:\MyReceiveExit.dll(MyReceiveExitNameSpace.MyReceiveExitClassName)

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama yönetilen istemci kipindeki bir kuyruk yöneticisine bağlandığında anlamlıdır. Ayrıca yalnızca yönetilen çıkışlar desteklenir.

## ***XMLC\_WMQ\_RECEIVE\_EXIT\_INIT***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Bir kanala geçirilen kullanıcı verileri, çağrıldığında çıkışa geçer.

Özelliğin değeri bir dizgidir. Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama yönetilen istemci kipindeki bir kuyruk yöneticisine bağlandığında ve [“XMLC\\_WMQ\\_RECEIVE\\_EXIT” sayfa 220 özelliği](#) ayarlandığında anlamlıdır.

### ***XMLC\_WMQ\_RESOLVED\_QUEUE\_YÖNETİCİLERİ***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Bu özellik, bağlı olduğu kuyruk yöneticisinin adını elde etmek için kullanılır.

Bir CCDT (İstemci Kanalı Tanımlama Çizelgesi) ile kullanıldığında, bu ad, Bağlantı Üreticisi 'nde belirtilen kuyruk yöneticisi adından farklı olabilir.

### ***XMLC\_WMQ\_RESOLVED\_QUEUE\_MANAGER\_ID***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Bu özellik, bağlantı tamamlandıktan sonra kuyruk yöneticisinin tanıtıcısıyla doldurulur.

### ***XMLC\_WMQ\_SECURITY\_EXIT***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Kanal güvenlik çıkışı tanımlar.

Özelliğin değeri, kanal güvenliği çıkışı tanımlayan ve aşağıdaki biçime sahip bir dizgidir:

**libraryName**(entryPointAd)

Burada,

- **libraryName** , yönetilen çıkışa ilişkin .dll dosyasının tam yoludur.
- **entryPointAd** , ad alanı tarafından nitelenen sınıf adıdır

Örneğin, C:\MySecurityExit.dll(MySecurityExitNamespace.MySecurityExitClassName)

Dizilimin uzunluk üst sınırı 128 karakterdir.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama yönetilen istemci kipindeki bir kuyruk yöneticisine bağlandığında anlamlıdır. Ayrıca yalnızca yönetilen çıkışlar desteklenir.

### ***XMLC\_WMQ\_SECURITY\_EXIT\_INIT***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Bir kanal güvenliği çıkışa çağrıldığında geçirilen kullanıcı verileri.

Kullanıcı verisi dizgisinin uzunluk üst sınırı 32 karakterdir.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama yönetilen istemci kipindeki bir kuyruk yöneticisine bağlandığında ve [“XMLC\\_WMQ\\_SECURITY\\_EXIT” sayfa 221 özelliği](#) ayarlandığında anlamlıdır.

## ***XMLC\_WMQ\_SEND\_EXIT***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Bir kanal gönderme çıkışı tanımlar.

Özelliğin değeri bir dizgidir. Kanal gönderme çıkışta aşağıdaki biçim bulunur:

**libraryName**(entryPointAd)

Burada,

- **libraryName** , yönetilen çıkışa ilişkin .dll dosyasının tam yoludur.
- **entryPointAd** , ad alanı tarafından nitelenen sınıf adıdır

Örneğin, C:\MySendExit.dll(MySendExitNameSpace.MySendExitClassName)

Varsayılan olarak özellik ayarlanmaz.

Bu özellik yalnızca, bir uygulama yönetilen istemci kipindeki bir kuyruk yöneticisine bağlandığında anlamlıdır. Ayrıca yalnızca yönetilen çıkışlar desteklenir.

## ***XMLC\_WMQ\_SEND\_EXIT\_INIT***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Kanal gönderme çıkışlarına geçirilen kullanıcı verileri çağrıldığında bu verileri gönderir.

Özelliğin değeri, kullanıcı verilerinin virgülle ayrılmış bir ya da daha fazla öğesinin dizgisidir. Varsayılan olarak özellik ayarlanmaz.

Kanal gönderme çıkışlarına geçirilen kullanıcı verilerinin belirtilmesine ilişkin kurallar, kanal alma çıkışlarına geçirilen kullanıcı verilerinin belirtilmesine ilişkin kurallarla aynıdır. Bu nedenle kurallar için bkz. "[XMLC\\_WMQ\\_RECEIVE\\_EXIT\\_INIT](#)" sayfa 220.

Bu özellik yalnızca, bir uygulama yönetilen istemci kipindeki bir kuyruk yöneticisine bağlandığında ve "[XMLC\\_WMQ\\_SEND\\_EXIT](#)" sayfa 222 özelliği ayarlandığında anlamlıdır.

## ***XMLC\_WMQ\_SEND\_CHECK\_COUNT***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory

Tek bir hareket dışı XMS oturumu içinde zamanuyumsuz koyma hataları olup olmadığını denetlemek için izin verilecek gönderme çağrılarının sayısı.

Varsayılan olarak bu özellik 0 değerine ayarlıdır.

## ***XMLC\_WMQ\_SHARE\_CONV\_ALLOWED***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory

### **Uygulanabilir nesnelere:**

JMS denetim aracı uzun adı: SHARECVALLED

JMS yönetim aracı kısa adı: SCALD

Bir istemci bağlantısının yuvasını, kanal tanımlarının eşleşmesi durumunda aynı kuyruk yöneticisiyle aynı işlem düzeyinde diğer üst düzey XMS bağlantılarıyla paylaşım paylaşamayacağını belirler. Bu özellik, uygulama geliştirme, bakım ya da işletim nedenleri için gerekirse, Connections ürününün ayrı yuvalarda tam olarak yalıtılmasına izin vermek için sağlanır. Bu özelliğin ayarlanması, temeldeki yuvanın paylaşılması için yalnızca XMS ' a işaret eder. Bu, tek bir yuvayla kaç bağlantının paylaşılacağını göstermez. The number of connections sharing a socket is determined by SHARECNV value which is negotiated between WebSphere MQ Client and WebSphere MQ Server.

Bir uygulama, özelliği ayarlamak için aşağıdaki adlandırılmış değişmezleri ayarlayabilir:

- XMLC\_WMQ\_SHARE\_CONV\_ALLOWED\_YANR-Bağlantılar bir yuvayı paylaşmaz.
- XMLC\_WMQ\_SHARE\_CONV\_ALLOWED\_TRUE-Bağlantılar bir yuvayı paylaşır.

Varsayılan olarak özellik, XMSSC\_WMQ\_SHARE\_CONV\_ALLOWED\_ENABLED olarak ayarlanır.

Bu özellik, yalnızca bir uygulama istemci kipinde bir kuyruk yöneticisine bağlandığında anlamlıdır.

## ***XMLC\_WMQ\_SSL\_CERT\_STORES***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Bir kuyruk yöneticisiyle SSL bağlantısında kullanılacak sertifika iptal listelerini (CRL ' ler) tutan sunucuların konumları.

Özelliğin değeri, virgüllerle ayrılan bir ya da daha fazla URL ' nin bir listesidir. Her URL ' de aşağıdaki biçim vardır:

```
[user[/password]@]ldap://[serveraddress][:portnum][, ...]
```

Bu biçim, temel MQJMS biçimiyle uyumlu, ancak genişletilmiş biçimiyle uyumludur.

Boş bir serveraddress olması geçerlidir. Bu durumda XMS , değerin "localhost" dizgisinin olduğunu varsayar.

Örnek bir liste şöyledir:

```
myuser/mypassword@ldap://server1.mycom.com:389  
ldap://server1.mycom.com  
ldap://  
ldap://:389
```

Yalnızca .NET için: IBM MQ 8.0' tan, IBM MQ (WMQ\_CM\_Client) ile yönetilen bağlantılardan ve IBM MQ (WMQ\_CM\_CLIENT\_UNManaged) ile yönetilen bağlantılardan (WMQ\_CM\_CLIENT\_UNMAED) her ikisi de TLS/SSL bağlantılarını destekler.

Varsayılan olarak özellik ayarlanmaz.

### **İlgili bilgiler**

[Yönetilmeyen .NET istemcisi için SSL ve TLS desteği](#)

[Yönetilen .NET istemcisi için SSL ve TLS desteği](#)

## ***XMLC\_WMQ\_SSL\_CIPHER\_SPEC***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Kuyruk yöneticisiyle güvenli bir bağlantıda kullanılacak CipherSpec adı.

IBM WebSphere MQ TLS desteği ile kullanabileceğiniz şifre belirtileri aşağıdaki çizelgede listelenir.

Kişisel bir sertifika istediğinizde, genel ve özel anahtar çifti için bir anahtar boyutu belirtiyorsunuz.

SSL el sıkışması sırasında kullanılan anahtar büyüklüğü, çizelgede belirtildiği gibi CipherSpectarafından belirlenmediği sürece sertifikada saklanan boyuttur. Varsayılan olarak bu özellik belirlenmez.

CipherSpec adı	Kullanılan iletişim kuralı	Hash algoritması	Şifreleme Algoritması	Şifreleme bitleri	FIPS <sup>1</sup>	Suite B 128 bit	Suite B 192 bit
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	SHA-1	AES	128	Evet	Hayır	Hayır
TLS_RSA_WITH_AES_256_CBC_SHA <sup>2</sup>	TLS 1.0	SHA-1	AES	256	Evet	Hayır	Hayır
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	DES	56	Hayır	Hayır	Hayır
TLS_RSA_WITH_3DES_EDE_CBC_SHA <sup>4</sup>	TLS 1.0	SHA-1	3DES	168	Evet	Hayır	Hayır
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Evet	Hayır	Hayır
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Evet	Hayır	Hayır
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Evet	Hayır	Hayır
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	SHA-256	AES	256	Evet	Hayır	Hayır
ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Hayır	Hayır	Hayır
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Evet	Hayır	Hayır
ECDHE_RSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Hayır	Hayır	Hayır
ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Evet	Hayır	Hayır
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Evet	Hayır	Hayır
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	Evet	Hayır	Hayır
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Evet	Hayır	Hayır
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	Evet	Hayır	Hayır
ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Evet	Evet	Hayır
ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Evet	Hayır	Evet
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Evet	Hayır	Hayır



CipherSpec adı	Kullanılan iletişim kuralı	Hash algoritması	Şifreleme Algoritması	Şifreleme bitleri	FIPS <sup>1</sup>	Suite B 128 bit	Suite B 192 bit
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Evet	Hayır	Hayır
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	SHA-256	Yok	0	Hayır	Hayır	Hayır
ECDHE_RSA_NULL_SHA256	TLS 1.2	SHA-256	Yok	0	Hayır	Hayır	Hayır
ECDHE_ECDSA_NULL_SHA256	TLS 1.2	SHA-256	Yok	0	Hayır	Hayır	Hayır
TLS_RSA_WITH_NULL_NULL	TLS 1.2	Yok	Yok	0	Hayır	Hayır	Hayır
TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Hayır	Hayır	Hayır

#### Notlar:

1. CipherSpec ' in Federal Information Processing Standards (FIPS) 140-2 ile uyumlu olup olmadığını belirtir. FIPS 'nin bir açıklaması ve FIPS 140-2 uyumlu işlem için WebSphere MQ ' nun nasıl yapılandırılacağı hakkında bilgi için çevrimiçi IBM IBM WebSphere MQ ürün belgelerinde *Federal Information Processing Standards (FIPS)* başlıklı konuya bakın.
2. Bu CipherSpec , uygun olmayan ilke dosyaları Explorer tarafından kullanılan JRE ' ye uygulanmıyorsa, WebSphere MQ Explorer 'dan bir kuyruk yöneticisine bağlantı sağlamak için kullanılamaz.
3. Bu CipherSpec , 19 Mayıs 2007 tarihinden önce FIPS 140-2 sertifikasına sahiptir.
4. When WebSphere MQ is configured for FIPS 140-2 compliant operation, this CipherSpec can be used to transfer up to 32 GB of data before the connection is terminated with error AMQ9288. Bu hatayı önlemek için, üçlü DES kullanmaktan kaçının (kullanımdan kaldırılacaktır) ya da FIPS 140-2 yapılandırmasındaki bu CipherSpec komutunu kullanırken gizli anahtar sıfırlamayı etkinleştirin.

#### İlgili bilgiler

[güvenlik](#)

[İletilerin veri bütünlüğü](#)

[CipherSpecsbelirtme](#)

#### **XMLC\_WMQ\_SSL\_CIPHER\_SUIT**

#### Veri tipi:

Dizgi

#### Özellik:

ConnectionFactory

Bir kuyruk yöneticisine yönelik TLS bağlantısında kullanılacak CipherSuite ' in adı. Güvenli bağlantının pazarlığı sırasında kullanılan iletişim kuralı, belirtilen CipherSuite ' e (CipherSuite) bağlıdır.

Bu özellik aşağıdaki kurallı değerlere sahiptir:

- SSL\_RSA\_WITH\_DES\_CBC\_SHA
- SSL\_RSA\_EXPORT1024\_WITH\_DES\_CBC\_SHA
- SSL\_RSA\_EXPORT1024\_WITH\_RC4\_56\_SHA
- SSL\_RSA\_EXPORT\_WITH\_RC4\_40\_MD5
- SSL\_RSA\_WITH\_RC4\_128\_MD5

- SSL\_RSA\_WITH\_RC4\_128\_SHA
- SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- SSL\_RSA\_WITH\_AES\_128\_CBC\_SHA
- SSL\_RSA\_WITH\_AES\_256\_CBC\_SHA
- SSL\_RSA\_WITH\_DES\_CBC\_SHA
- SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA

Bu değer, [XMLC\\_WMQ\\_SSL\\_CIPHER\\_SPEC](#) için bir seçenek olarak sağlanabilir.

[XMLC\\_WMQ\\_SSL\\_CIPHER\\_SPEC](#) için boş olmayan bir değer belirtilirse, bu değer [XMLC\\_WMQ\\_SSL\\_CIPHER\\_SUITE](#) ayarını geçersiz kılar. [XMLC\\_WMQ\\_SSL\\_CIPHER\\_SPEC](#) bir değeri yoksa, GSKit 'e verilecek şifreleme takımı olarak [XMLC\\_WMQ\\_SSL\\_CIPHER\\_SUITE](#) değeri kullanılır. Bu durumda değer, “Bir IBM MQ kuyruk yöneticisine yönelik bağlantılar için CipherSuite ve CipherSpec ad eşlemeleri” [sayfa 64](#) içinde açıklandığı şekilde, eşdeğer CipherSpec değeriyle eşlenir.

Hem [XMLC\\_WMQ\\_SSL\\_CIPHER\\_SPEC](#) hem de [XMLC\\_WMQ\\_SSL\\_CIPHER\\_SUITE](#) boşsa, pChDef->SSLCipherSpec alanı boşluklarla doldurulur.

Yalnızca .NET için: IBM MQ 8.0' tan, IBM MQ (WMQ\_CM\_Client) ile yönetilen bağlantılardan ve IBM MQ (WMQ\_CM\_CLIENT\_UNManaged) ile yönetilen bağlantılardan (WMQ\_CM\_CLIENT\_UNMAED) her ikisi de TLS/SSL bağlantılarını destekler.

Varsayılan olarak özellik ayarlanmaz.

### İlgili bilgiler

[Yönetilmeyen .NET istemcisi için SSL ve TLS desteği](#)

[Yönetilen .NET istemcisi için SSL ve TLS desteği](#)

## ***XMLC\_WMQ\_SSL\_CRYPTO\_HW***

### Veri tipi:

Dizgi

### Özellik:

ConnectionFactory

İstemci sistemine bağlı şifreleme donanımlarıyla ilgili yapılandırma ayrıntıları.

Bu özellik aşağıdaki kurallı değerlere sahiptir:

- GSK\_ACCELERATOR\_RAINBOW\_CS\_OFF
- GSK\_ACCELERATOR\_RAINBOW\_CS\_ON
- GSK\_ACCELERATOR\_NCIPHER\_NF\_OFF
- GSK\_ACCELERATOR\_NCIPHER\_NF\_ON

PKCS11 şifreleme donanımları için özel bir biçim vardır (burada DriverPath, TokenLabel ve TokenPassword , kullanıcı tarafından belirlenen dizgilerdir):

```
GSK_PKCS11=PKCS#11 DriverPath; PKCS#11 TokenLabel;PKCS#11 TokenPassword
```

XMS , dizginin içeriğini yorumlamıyor ya da değiştirmiyor. Sağlanan değeri, en çok 256 tek baytlık karakter üst sınırı olarak MQSCO.CryptoHardware (Şifreleme) alanı

Yalnızca .NET için: IBM MQ 8.0' tan, IBM MQ (WMQ\_CM\_Client) ile yönetilen bağlantılardan ve IBM MQ (WMQ\_CM\_CLIENT\_UNManaged) ile yönetilen bağlantılardan (WMQ\_CM\_CLIENT\_UNMAED) her ikisi de TLS/SSL bağlantılarını destekler.

Varsayılan olarak özellik ayarlanmaz.

### İlgili bilgiler

[Yönetilmeyen .NET istemcisi için SSL ve TLS desteği](#)

Yönetilen .NET istemcisi için SSL ve TLS desteği

## ***XMLC\_WMQ\_SSL\_FIPS\_REQUIRED***

### **Veri tipi:**

Boole

### **Özellik:**

ConnectionFactory

Bu özelliğin değeri, bir uygulamanın FIPS uyumlu olmayan şifreleme takımlarını kullanıp kullanamayabileceğini belirler. Bu özellik true değerine ayarlanırsa, istemci-sunucu bağlantısı için yalnızca FIPS algoritmaları kullanılır.

Bu özellik, MQSCO.FipsRequired:

<i>Çizelge 34. MQSCO.FlipsRequired özelliği</i>		
<b>Değer</b>	<b>Tanım</b>	<b>Karşılık gelen MQSCO.FipsRequired</b>
yanlış	Herhangi bir CipherSpec kullanılabilir.	MQSSL_FIPS_NO (varsayılan değer)
doğru	Bu istemci bağlantısına uygulanan CipherSpec ' de yalnızca FIPS onaylı şifreleme algoritmaları kullanılabilir.	MQSSL_FIPS_YES

XMS , ilgili değeri MQSCO.FipsRequired , MQCONNX çağrılmadan önce.

MQSCO.FipsRequired , yalnızca WebSphere MQ sürüm 6 'dan edinilebilir. WebSphere MQ sürüm 5.3ise, bu özellik ayarlandıysa, XMS kuyruk yöneticisiyle bağlantı kurma girişiminde bulunmaz ve bunun yerine uygun bir kural dışı durum yayınlar.

Yalnızca .NET için: IBM MQ 8.0' tan, IBM MQ (WMQ\_CM\_Client) ile yönetilen bağlantılardan ve IBM MQ (WMQ\_CM\_CLIENT\_UNManaged) ile yönetilen bağlantılardan (WMQ\_CM\_CLIENT\_UNMAED) her ikisi de TLS/SSL bağlantılarını destekler.

### **İlgili bilgiler**

Yönetilmeyen .NET istemcisi için SSL ve TLS desteği

Yönetilen .NET istemcisi için SSL ve TLS desteği

## ***XMLC\_WMQ\_SSL\_KEY\_REPOSITORY***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Anahtarların ve sertifikaların depolanmış olduğu anahtar veritabanı dosyasının konumu.

XMS , dizeyi 256 tek baytlık karakter sınırına kadar, MQSCO.KeyRepository alanı. WebSphere MQ , bu dizeyi tam yol da dahil olmak üzere bir dosya adı olarak yorumlar.

Yalnızca .NET için: IBM MQ 8.0' tan, IBM MQ (WMQ\_CM\_Client) ile yönetilen bağlantılardan ve IBM MQ (WMQ\_CM\_CLIENT\_UNManaged) ile yönetilen bağlantılardan (WMQ\_CM\_CLIENT\_UNMAED) her ikisi de TLS/SSL bağlantılarını destekler.

Varsayılan olarak özellik ayarlanmaz.

### **İlgili bilgiler**

Yönetilmeyen .NET istemcisi için SSL ve TLS desteği

Yönetilen .NET istemcisi için SSL ve TLS desteği

## ***XMLC\_WMQ\_SSL\_KEY\_RESETCOUNT***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory

KeyResetCount, gizli anahtar yeniden anlaşılmeden önce, bir SSL etkileşimi içinde gönderilen ve alınan şifrelenmemiş baytların toplam sayısını temsil eder. Bayt sayısı, MCA tarafından gönderilen denetim bilgilerini içerir.

XMS , bu özellik için sağladığınız değeri MQSCO.KeyResetCount , MQCONNX çağrılmadan önce.

MQSCO.KeyRestCount , yalnızca WebSphere MQ sürüm 6 'dan kullanılabilir. WebSphere MQ sürüm 5.3ise, bu özellik ayarlandıysa, XMS kuyruk yöneticisiyle bağlantı kurma girişiminde bulunmaz ve bunun yerine uygun bir kural dışı durum yayınlar.

Yalnızca .NET için: IBM MQ 8.0' tan, IBM MQ (WMQ\_CM\_Client) ile yönetilen bağlantılardan ve IBM MQ (WMQ\_CM\_CLIENT\_UNManaged) ile yönetilen bağlantılardan (WMQ\_CM\_CLIENT\_UNMAED) her ikisi de TLS/SSL bağlantılarını destekler.

Bu özelliğin varsayılan değeri sıfır, bu da gizli anahtarların hiçbir zaman yeniden görüşmemekte olduğunu gösterir.

### **İlgili bilgiler**

[Yönetilmeyen .NET istemcisi için SSL ve TLS desteği](#)

[Yönetilen .NET istemcisi için SSL ve TLS desteği](#)

## ***XMLC\_WMQ\_SSL\_PEER\_NAME***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Bir kuyruk yöneticisiyle SSL bağlantısında kullanılacak eş adı.

Bu özellik için kurallı değerler listesi yok. Bunun yerine, bu dizeyi SSLPEERkurallarına göre oluşturmanız gerekir.

Eşdüzey bir ad örneği:

```
"CN=John Smith, O=IBM ,OU=Test , C=GB"
```

XMS , dizgiyi doğru tek byte kod sayfasına kopyalar ve MQCONNX çağrılmadan önce doğru değerleri MQCD.SSLPeerNamePtr ve MQCD.SSLPeerNameLength içine yerleştirir.

Bu özellik, uygulamanın istemci kipinde bir kuyruk yöneticisine bağlandığı durumlarda anlamlıdır.

Yalnızca .NET için: IBM MQ 8.0' tan, IBM MQ (WMQ\_CM\_Client) ile yönetilen bağlantılardan ve IBM MQ (WMQ\_CM\_CLIENT\_UNManaged) ile yönetilen bağlantılardan (WMQ\_CM\_CLIENT\_UNMAED) her ikisi de TLS/SSL bağlantılarını destekler.

Varsayılan olarak özellik ayarlanmaz.

### **İlgili bilgiler**

[Yönetilmeyen .NET istemcisi için SSL ve TLS desteği](#)

[Yönetilen .NET istemcisi için SSL ve TLS desteği](#)

[SLAYICI ADI](#)

## ***XMSC\_WMQ\_SYNCPOINT\_ALL\_GETS***

### **Veri tipi:**

System.Boolean

### **Özellik:**

ConnectionFactory

Tüm iletilerin eşitleme noktası denetimi içindeki kuyruklardan alınıp alınmayacağı.

Özelliğin geçerli değerleri şunlardır:

#### **Geçerli değer**

yanlış

doğru

#### **Anlamı**

Koşullar uygun olduğunda, XMS istemcisi, iletileri eşitleme noktası denetimi dışında kuyruklardan alabilir.

XMS istemcisi, eşitleme noktası denetimi içindeki kuyruklardan tüm iletileri almalıdır.

Varsayılan değer: false.

## ***XMLC\_WMQ\_TARGET\_İSTEMCENİ***

### **Veri tipi:**

System.Int32

### **Özellik:**

Hedef

### **Bir URI ' de kullanılan ad:**

targetClient

Hedef hedefine gönderilen iletilerin bir MQRFH2 üstbilgisi içerip içermediğini belirtir.

Bir uygulama MQRFH2 üstbilgisi içeren bir ileti gönderirse, alıcı uygulamanın üstbilgiyi işleyebilmelidir.

Özelliğin geçerli değerleri şunlardır:

#### **Geçerli değer**

XMLC\_WMQ\_TARGET\_DEST\_JMS

XMLC\_WMQ\_TARGET\_DEST\_MQ

#### **Anlamı**

Hedefte gönderilen iletiler bir MQRFH2 üstbilgisi içerir. Uygulama iletileri başka bir XMS uygulamasına, WebSphere JMS uygulamasına ya da bir MQRFH2 üstbilgisini işlemek için tasarlanmış bir yerel WebSphere MQ uygulamasına gönderiyorsa bu değeri belirtin.

Hedefte gönderilen iletiler bir MQRFH2 üstbilgisi içermiyor. Uygulama, iletileri bir MQRFH2 üstbilgisini işlemek üzere tasarlanmamış bir yerel WebSphere MQ uygulamasına gönderiyorsa, bu değeri belirtin.

Varsayılan değer XMSC\_WMQ\_TARGET\_DEST\_JMS ' dir.

## ***XMLC\_WMQ\_TEMP\_Q\_PREFIX***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Uygulama, XMS geçici kuyruk yarattığında yaratılan WebSphere MQ dinamik kuyruğunun adını oluşturmak için kullanılan önek.

The rules for forming the prefix are the same as the rules for forming the contents of the **DynamicQName** field in an object descriptor, but the last non-blank character must be an asterisk(\*). If the property is

not set, the value used is CSQ.\* on z/OS and AMQ.\* on the other platforms. Varsayılan olarak özellik ayarlanmaz.

Bu özellik, yalnızca noktadan noktaya iletişim etki alanında anlamlıdır.

### ***XMSC\_WMQ\_TEMP\_TOPIC\_PREFIX***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory, Hedef

Geçici konular oluştururken, XMS, "TEMP/TEMPTOPICPREFIX/unique\_id" formunun bir konu dizgisi oluşturur ya da bu özellik varsayılan değeri içeriyorsa, "TEMP/unique\_id" dizgisi oluşturulur. Boş olmayan bir değer belirlenmesi, bu bağlantı altında oluşturulan geçici konulara aboneler için yönetilen kuyruklar oluşturmak için belirli model kuyruklarının tanımlanmasına olanak sağlar.

Bir IBM WebSphere MQ konu dizgisi için yalnızca geçerli karakterlerden oluşan, boş değerli olmayan dizgi, bu özellik için geçerli bir değerdir.

Varsayılan olarak bu özellik "" (boş dizgi) olarak ayarlıdır.

**Not:** Bu özellik, yalnızca yayınlama/abone olma etki alanında anlamlıdır.

### ***XMLC\_WMQ\_TEMPORARY\_MODEL***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Uygulama XMS geçici kuyruk yarattığında, dinamik bir kuyruğun yaratıldığı WebSphere MQ model kuyruğunun adı.

Özelliğin varsayılan değeri SYSTEM.DEFAULT.MODEL.QUEUE.

Bu özellik, yalnızca noktadan noktaya iletişim etki alanında anlamlıdır.

### ***XMLC\_WMQ\_WILDCARD\_BIÇIMI***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionFactory, Hedef

Bu özellik, genel arama karakteri sözdiziminin hangi sürümünü kullanılacağını belirler.

Yayınlama/abone olma özelliğini IBM WebSphere MQ '\*' ve '?' ile kullanırken genel arama karakteri olarak kabul edilir. Yayınlama kullanılırken '#' ve '+' karakterleri, IBM Integration Bus ile abone olma özelliğini kullanırken joker karakterler olarak ele alınır. Bu özellik, XMLC\_WMQ\_ROKER\_VERSION özelliğinin yerini alır.

Bu özelliğe ilişkin geçerli değerler şunlardır:

#### ***XMSC\_WMQ\_WILDCARD\_TOPIC\_ONLY***

Konu düzeyi genel arama karakterlerini yalnızca başka bir deyişle tanıır. '#' ve '+' karakteri joker karakter olarak kabul edilir. Bu değer XMSC\_WMQ\_BROKER\_V2 ile aynıdır.

#### ***XMLC\_WMQ\_WILDCARD\_CHAR\_ONLY***

Karakter genel arama karakterlerini tanıır, başka bir deyişle '\*' ve '?' genel arama karakteri olarak kabul edilir. Bu değer XMSC\_WMQ\_BROKER\_V1 ile aynıdır.

Varsayılan olarak bu özellik XMSC\_WMQ\_WILDCARD\_TOPIC\_TOPIC\_TOPIC\_ONLY olarak ayarlanır.

## ***XMLC\_WPM\_BUSU\_ADı***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory ve Hedef

### **Bir URI ' de kullanılan ad:**

busName

Bir bağlantı üreticisi için, uygulamanın bağlandığı hizmet tümleştirme veriyolunun adı ya da hedefin varolduğu hizmet tümleştirme veriyolunun adı gibi bir hedef için bağlantı kurar.

Konu olan bir hedef için bu özellik, ilişkili konu alanının bulunduğu hizmet tümleştirme veriyolunun adıdır. Bu konu alanı, XMSSC\_WPM\_TOPIC\_SPACE özelliği tarafından belirtilir.

Özellik bir hedef için ayarlanmazsa, kuyruğun ya da ilişkili konu alanının, uygulamanın bağlandığı hizmet tümleştirme veriyolunda var olduğu varsayılır.

Varsayılan olarak özellik ayarlanmaz.

## ***XMSC\_WPM\_CONNECTION\_PROTOCOL***

### **Veri tipi:**

System.Int32

### **Özellik:**

Bağlantı

İleti alışverişi altyapısına yönelik bağlantı için kullanılan iletişim protokolü. Bu özellik salt okunurdur.

Özelliğin olası değerleri aşağıdaki gibidir:

<b>Değer</b>	<b>Anlamı</b>
XMLC_WPM_CP_HTTP	Bağlantı TCP/IP üzerinden HTTP kullanır.
XMLC_WPM_CP_TCP	Bağlantı TCP/IP ' yi kullanır.

## ***XMSC\_WPM\_CONNECTION\_PROXIMITY***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory

Bağlantıya ilişkin bağlantı yakınlığı ayarı. Bu özellik, uygulamanın bağlandığı ileti alışverişi altyapısının önyükleme sunucusuna ne kadar yakın olması gerektiğini belirler.

Özelliğin geçerli değerleri şunlardır:

<b>Geçerli değer</b>	<b>Bağlantı yakınlığı ayarı</b>
XMSC_WPM_CONNECTION_PROXIMITY_BUS	Veriyolu
XMSC_WPM_CONNECTION_PROXIMITY_CLUSTER	Küme
XMSC_WPM_CONNECTION_PROXPROTY_HOST	Anasistem
XMSC_WPM_CONNECTION_PROXIMITY_SERVER	Sunucu

Varsayılan değer XMSSC\_WPM\_CONNECTION\_PROXIMITY\_BUS değeridir.

## ***XMLC\_WPM\_DUR\_SUB\_HOME***

### **Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

**Bir URI ' de kullanılan ad:**

durableSubscriptionAna Sayfası

Bir bağlantıya ya da hedefe ilişkin tüm kalıcı aboneliklerin yönetiliyor olduğu ileti alışverişi altyapısının adı. Dayanıklı abonelere teslim edilecek iletiler, aynı ileti alışverişi altyapısının yayın noktasında saklanır.

Bir uygulamanın bağlantıyı kullanan dayanıklı bir abone oluşturabilmesi için, bağlantı için sürekli abonelik ana dizini belirtilmelidir. Bir hedef için belirtilen herhangi bir değer, bağlantı için belirtilen değeri geçersiz kılar.

Varsayılan olarak özellik ayarlanmaz.

Bu özellik, yalnızca yayınlama/abone olma etki alanında anlamlıdır.

***XMLC\_WPM\_HOST\_NAME*****Veri tipi:**

Dizgi

**Özellik:**

Bağlantı

Uygulamanın bağlı olduğu ileti alışverişi altyapısını içeren sistemin anasistem adı ya da IP adresi. Bu özellik salt okunurdur.

***XMLC\_WPM\_LOCAL\_ADRESİ*****Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Bir hizmet bütünleştirme veriyoluna bağlantı için, bu özellik kullanılacak yerel ağ arabirimini ya da kullanılacak yerel kapıyı ya da yerel kapı aralığını ya da her ikisini de belirtir.

Özelliğin değeri, aşağıdaki biçimi taşıyan bir dizgidir:

[*anasistem\_adi*] [ (*düşük\_kapısı*) [,*yükseklik\_kapısı*]]

Değişkenlerin anlamları aşağıdaki gibidir:

***anasistem\_adi***

Bağlantı için kullanılacak yerel ağ arabiriminin anasistem adı ya da IP adresi.

Bu bilgileri yalnızca, uygulamanın çalıştığı sistemin iki ya da daha çok ağ arabirimi varsa ve bağlantı için hangi arabirimin kullanılması gerektiğini belirtmeniz gerekiyorsa gereklidir. Sistemde yalnızca bir ağ arabirimi varsa, yalnızca o arabirim kullanılabilir. Sistemde iki ya da daha çok ağ arabirimi varsa ve hangi arabirimin kullanılması gerektiğini belirtmezseniz, arabirim rasgele seçilir.

***düşük\_kapı***

Bağlantı için kullanılacak yerel kapı numarası.

*high\_port* da belirtilirse, *low\_port* , bir kapı numaralarındaki en düşük kapı numarasını yorumlanır.

***yükseklik\_kapısı***

Bir kapı numaralarındaki en yüksek kapı numarası. Bağlantı için belirtilen aralıktaki kapılardan biri kullanılmalıdır.

Aşağıda, özelliğin geçerli değerlerine ilişkin bazı örnekler bulunmaktadır:

JUPITER

9.20.4.98

JÜPITER (1000)

9.20.4.98(1000,2000)



(1000)  
(1000,2000)

Varsayılan olarak özellik ayarlanmaz.

### ***XMLC\_WPM\_ME\_ADı***

**Veri tipi:**

Dizgi

**Özellik:**

Baęlantı

Uygulamanın baęlı olduęu ileti alışveriři altyapısının adı. Bu özellik salt okunurdur.

### ***XMLC\_WPM\_NON\_PERSISTENT\_MAP***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionFactory

Baęlantı kullanılarak gönderilen kalıcı olmayan iletilerin güvenilirlięi düzeyi.

Özellikğin geçerli deęerleri řunlardır:

**Geçerli deęer**

XMSC\_WPM\_MAPPING\_AS\_DESTIND

**Güvenilirlik düzeyi**

Hizmet tümleřtirme veriyolundaki kuyruk ya da konu alanı için belirtilen varsayılan güvenilirlik düzeyine göre belirlenir.

XMSSC\_WPM\_MAPPING\_BEST\_ZAHMET\_NON\_  
Kalıcı

Kalıcı olmayan en iyi çalıřma

XMSSC\_WPM\_MAPPING\_EXPRESS\_NON\_  
Kalıcı

Express kalıcı olmayan

XMSSC\_WPM\_MAPPING\_RELIABLE\_NON\_  
Kalıcı

Güvenilir olmayan güvenilir

XMSC\_WPM\_MAPPING\_RELIABLE\_PERSISTENT

Güvenilir kalıcı

XMSC\_WPM\_MAPPING\_ASSURED\_PERSISTENT

Güvenli kalıcı

Varsayılan deęer XMSSC\_WPM\_MAPPING\_EXPRESS\_NON\_PERSISTENT deęeridir.

### ***XMLC\_WPM\_PERSISTENT\_MAP***

**Veri tipi:**

System.Int32

**Özellik:**

ConnectionFactory

Baęlantı kullanılarak gönderilen kalıcı iletilerin güvenilirlięi düzeyi.

Özellikğin geçerli deęerleri řunlardır:

**Geçerli değer**

XMSC\_WPM\_MAPPING\_AS\_DESTIND

XMSSC\_WPM\_MAPPING\_BEST\_ZAHMET\_NON\_  
KalıcıXMSSC\_WPM\_MAPPING\_EXPRESS\_NON\_  
KalıcıXMSSC\_WPM\_MAPPING\_RELIABLE\_NON\_  
Kalıcı

XMSC\_WPM\_MAPPING\_RELIABLE\_PERSISTENT

XMSC\_WPM\_MAPPING\_ASSURED\_PERSISTENT

**Güvenilirlik düzeyi**Hizmet tümleştirme veriyolundaki  
kuyruk ya da konu alanı için  
belirtilen varsayılan güvenilirlik  
düzeyine göre belirlenir.

Kalıcı olmayan en iyi çalışma

Express kalıcı olmayan

Güvenilir olmayan güvenilir

Güvenilir kalıcı

Güvenli kalıcı

Varsayılan değer XMSSC\_WPM\_MAPPING\_RELIABLE\_PERSISTENT değeridir.

***XMLC\_WPM\_KAPISI*****Veri tipi:**

System.Int32

**Özellik:**

Bağlantı

Uygulamanın bağlı olduğu ileti alışverişi altyapısının dinlediği kapının numarası. Bu özellik salt okunurdur.

***XMSC\_WPM\_PROVIDER\_UÇ Noktaları*****Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Önyükleme sunucularının bir ya da daha çok uç noktası adresi dizisi. Uç nokta adresleri virgüllerle ayrılır.

Önyükleme sunucusu, uygulamanın bağlandığı ileti alışverişi altyapısının seçilmesinden sorumlu olan bir uygulama sunucudur. Bir önyükleme sunucusunun uç noktası adresi şu biçimde olmalıdır:

*anasistem\_adi:kapi\_numarası:chain\_name*

Bir uç nokta adresinin bileşenlerinin anlamları aşağıdaki gibidir:

***anasistem\_adi***

Önyükleme sunucusunun yer aldığı sistemin anasistem adı ya da IP adresi. Anasistem adı ya da IP adresi belirlenmezse, varsayılan değer localhost olur.

***kapi\_numarası***

Önyükleme sunucusunun gelen istekleri dinlediği kapının numarası. Kapi numarası belirtilmezse, varsayılan değer 7276 'tır.

***chain\_name***

Önyükleme sunucusu tarafından kullanılan bir önyükleme iletim zincirinin adı. Geçerli değerler şunlardır:

**Geçerli değer**

XMSC\_WPM\_BOOTSTRAP\_HTTP

**Önyükleme iletimi zincirinin adı**

BootstrapTunneledİleti Sistemi

**Geçerli değer**

XMSSC\_WPM\_BOOTSTRAP\_HTTPS  
XMSSC\_WPM\_BOOTSTRAP\_SSL  
XMLC\_WPM\_BOOTSTRAP\_TCP

**Önyükleme iletimi zincirinin adı**

BootstrapTunneledSecureMessaging  
BootstrapSecureİleti Sistemi  
BootstrapBasicİleti Sistemi

Ad belirtilmezse, varsayılan değer XMSSC\_WPM\_BOOTSTRAP\_TCP ' dir.

Herhangi bir uç nokta adresi belirtilmezse, varsayılan değer olarak localhost:7276:BootstrapBasicMessagingvarsayılan değeri olur.

***XMLC\_WPM\_SSL\_CIPHER\_SUIT*****Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Bir WebSphere Service Integration Bus ileti alışverişi altyapısına TLS bağlantısında kullanılacak CipherSuite ' in adı. Güvenli bağlantının pazarlığı sırasında kullanılan iletişim kuralı, belirtilen CipherSuite' e (CipherSuite) bağlıdır.

<i>Çizelge 35. WebSphere Service Integration Bus ileti alışverişi altyapısıyla bağlantı içinCipherSuite seçenekleri</i>	
<b>şifre paketi</b>	<b>Kullanılan iletişim kuralı</b>
TLS_RSA_WITH_DES_CBC_SHA	TLSv1
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_256_CBC_SHA	TLSv1

**Notlar:**

1. TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA ve TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA CipherSuites , yalnızca Windows ya da Solaris üzerinde desteklenir. (Bu, GSKit tarafından dikte edilir.)
2. TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA kullanımdan kaldırıldı. However, it can still be used to transfer up to 32 GB of data before the connection is terminated with error AMQ9288. Bu hatayı önlemek için, üçlü DES kullanmaktan kaçınmanız ya da bu CipherSpeckomutunu kullanırken gizli anahtar sıfırlamayı etkinleştirmeniz gerekir.

Bu özellik için varsayılan bir değer yoktur. SSL ya da TLS kullanmak istiyorsanız, bu özellik için bir değer belirlemeniz gerekir; tersi durumda, uygulamanız sunucuya başarıyla bağlanamıyor.

***XMLC\_WPM\_SSL\_FIPS\_REQUIRED*****Veri tipi:**

Boole

**Özellik:**

ConnectionFactory

Bu özelliğin değeri, bir uygulamanın FIPS uyumlu olmayan şifreleme takımlarını kullanıp kullanamayabileceğini belirler. Bu özellik true değerine ayarlanırsa, istemci-sunucu bağlantısı için yalnızca FIPS algoritmaları kullanılır.Bu özelliğin değerinin TRUE olarak ayarlanması, uygulamanın FIPS uyumlu olmayan şifreleme takımlarını kullanmasını önler.

Varsayılan değer olarak, özellik FALSE olarak ayarlanır (yani FIPS kipi kapalı).

## ***XMSSC\_WPM\_SSL\_KEY\_REPOSITORY***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Güvenli bağlantıda kullanılacak genel ya da özel anahtarları içeren anahtarlık dosyası olan dosyanın yolu.

Anahtarlık dosyası özelliğinin, XMSSC\_WPM\_SSL\_MS\_CERTIFICATE\_STORE özel değerine ayarlanması, Microsoft Windows anahtar veri tabanının kullanılmasını belirtir. **Control Panel > Internet Options > Content > Certificates**(Denetim Masası-Internet Seçenekleri-> İçerik-> Sertifikalar)underaltında bulunan Microsoft Windows anahtar veritabanını kullanarak, ayrı bir anahtar dosyası veritabanı gereksinmesini kaldırır. Bu değışmezin Windows x64 üzerinde ve diğler platformlarda kullanılmasına izin verilmez.

Varsayılan olarak özellik ayarlanmaz.

## ***XMLC\_WPM\_SSL\_KEYRING\_LABEL***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Sunucu ile kimlik doğrulaması yapılırken kullanılacak sertifika. Hiçbir değer belirtilmemişse, varsayılan sertifika kullanılır.

Varsayılan olarak özellik ayarlanmaz.

## ***XMLC\_WPM\_SSL\_KEYRING\_PW***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Anahtarlık dosyasına ilişkin parola.

Bu özellik, anahtarlık dosyasına ilişkin parolayı yapılandırmak için XMLC\_WPM\_SSL\_KEYRING\_STASH\_FILE dosyasını kullanmak için bir alternatif olarak kullanılabilir.

Varsayılan olarak özellik ayarlanmaz.

## ***XMLC\_WPM\_SSL\_KEYRING\_STASH\_FILE***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Anahtar havuzu dosyasının parolasını içeren ikili dosyanın adı.

Bu özellik, anahtarlık dosyasına ilişkin parolayı yapılandırmak için XMLC\_WPM\_SSL\_KEYRING\_PW dosyasının kullanılması için bir alternatif olarak kullanılabilir.

Varsayılan olarak özellik ayarlanmaz.

## ***XMLC\_WPM\_TARGET\_GROUP***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

İleti alışverişi altyapılarının hedef grubunun adı. Hedef grubun niteliği XMLC\_WPM\_TARGET\_TYPE özelliği tarafından belirlenir.

Bir ileti alışverişi altyapısıyla ilgili aramayı, hizmet tümleştirme veriyolundaki ileti alışverişi altyapılarının bir alt grubuna sınırlamak için bu özelliği ayarlayın. Uygulamanızın hizmet tümleştirme veriyolundaki herhangi bir ileti alışverişi işlemcisine bağlanabilmesini istiyorsanız, bu özelliği ayarlamayın.

Varsayılan olarak özellik ayarlanmaz.

## ***XMSSC\_WPM\_HEDEF\_ANLAM***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory

İleti alışverişi altyapılarının hedef grubunun öneminin.

Özelliğin geçerli değerleri şunlardır:

### **Geçerli değer**

XMSSC\_WPM\_TARGET\_SIGNANCE\_  
Tercih Edilen

XMSSC\_WPM\_TARGET\_SIGNANCE\_  
ZORUNLU

### **Anlamı**

Hedef gruptaki bir ileti alışverişi altyapısı varsa, hedef gruptaki bir ileti alışverişi altyapısı seçilir. Ters durumda, aynı hizmet tümleştirme veriyolunda olması koşuluyla, hedef grubun dışındaki bir ileti alışverişi altyapısı seçilidir.

Seçilen ileti alışverişi altyapısının hedef grupta olması gerekir. Hedef gruptaki bir ileti alışverişi altyapısı kullanılamıyorsa, bağlantı işlemi başarısız olur.

Özelliğin varsayılan değeri XMSSC\_WPM\_TARGET\_SENCEANCE\_PREFER\_PRECIENCE'DIR.

## ***XMSSC\_WPM\_TARGET\_TRANSPORT\_CHAIN***

### **Veri tipi:**

Dizgi

### **Özellik:**

ConnectionFactory

Bir ileti alışverişi işlemcisine bağlanmak için uygulamanın kullanması gereken gelen iletim zincirinin adı.

Özelliğin değeri, ileti alışverişi altyapısını barındıran uygulama sunucusunda var olan herhangi bir gelen iletim zincirinin adı olabilir. Önceden tanımlanmış gelen iletim zincirlerinden biri için aşağıdaki adlandırılmış sabit değer sağlanır:

### **Adlandırılmış sabit**

XMSSC\_WPM\_TARGET\_TRANSPORT\_CHAIN\_BASIC

### **İletim zincirinin adı**

InboundBasicİleti Sistemi

Özelliğin varsayılan değeri XMSSC\_WPM\_TARGET\_TRANSPORT\_CHAIN\_BASIC değeridir.

## ***XMLC\_WPM\_TARGET\_TIP***

### **Veri tipi:**

System.Int32

### **Özellik:**

ConnectionFactory

İleti alışverişi altyapılarının hedef grubunun tipi. Bu özellik, XMLC\_WPM\_TARGET\_GROUP özelliği tarafından tanımlanan hedef grubun niteliyi belirler.

Özelliğın geçerli değeri şunlardır:

**Geçerli değeri**

XMLC\_WPM\_TARGET\_TYPE\_KULLANICILARI

XMSC\_WPM\_TARGET\_TYPE\_CUSTOM

XML\_WPM\_TARGET\_TYPE\_ME

**Anlamı**

Hedef grubun adı, bir veriyolu üyesinin adıdır. Hedef grup, veriyolu üyesindeki tüm ileti alışverişi altyapıları.

Hedef grubun adı, kullanıcı tanımlı ileti alışverişi altyapıları grubunun adıdır. Hedef grup, kullanıcı tanımlı gruba kayıtlı tüm ileti alışverişi altyapılarıdır.

Hedef grubun adı, bir ileti alışverişi altyapısının adıdır. Hedef grup, belirtilen ileti alışverişi altyapısıdır.

Varsayılan olarak özellik ayarlanmaz.

***XMSC\_WPM\_TEMP\_Q\_PREFIX***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Uygulama XMS geçici kuyruk yarattığında, hizmet tümleştirme veriyolunda yaratılan geçici kuyruğun adını oluşturmak için kullanılan önek. Önek en çok 12 karakter içerebilir.

Geçici bir kuyruğun adı, "\_Q" karakteriyle ve ardından önek ile başlar. Adın geri kalan kısmı sistem tarafından oluşturulan karakterlerden oluşur.

Varsayılan olarak, özellik belirlenmez; bu, geçici bir kuyruğun adının bir öneki olmadığı anlamına gelir.

Bu özellik, yalnızca noktadan noktaya iletişim etki alanında anlamlıdır.

***XMSC\_WPM\_TEMP\_TOPIC\_PREFIX***

**Veri tipi:**

Dizgi

**Özellik:**

ConnectionFactory

Uygulama tarafından oluşturulan geçici bir konunun adını oluşturmak için kullanılan önek. Önek en çok 12 karakter içerebilir.

Geçici bir konunun adı, "\_T" karakterleriyle ve ardından önek ile başlar. Adın geri kalan kısmı sistem tarafından oluşturulan karakterlerden oluşur.

Varsayılan olarak, özellik belirlenmez; bu, geçici bir konunun adının bir öneki olmadığı anlamına gelir.

Bu özellik, yalnızca yayınlama/abone olma etki alanında anlamlıdır.

***XMSC\_WPM\_TOPIC\_SPACE***

**Veri tipi:**

Dizgi

**Özellik:**

Hedef

**Bir URI ' de kullanılan ad:**

topicSpace

Konu yer alan konu alanının adı. Yalnızca bir konu olan bir hedef bu özelliğe sahip olabilir.

Varsayılan deęer olarak, zellik ayarlanmaz, varsayılan konu alanı varsayıldığı anlamına gelir.  
Bu zellik, yalnızca yayınlama/abone olma etki alanında anlamlıdır.





## Özel notlar

Bu belge, ABD'de kullanıma sunulan ürünler ve hizmetler için hazırlanmıştır.

IBM, bu belgede sözü edilen ürün, hizmet ya da özellikleri diğer ülkelerde kullanıma sunmayabilir. Bulduğunuz yerde kullanıma sunulan ürün ve hizmetleri yerel IBM müşteri temsilcisinden ya da çözüm ortağınızdan öğrenebilirsiniz. Bir IBM ürün, program ya da hizmetine gönderme yapılması, açık ya da örtük olarak yalnızca o IBM ürünü, programı ya da hizmetinin kullanılabilirliğini göstermez. Aynı işlevi gören ve IBM'in fikri mülkiyet haklarına zarar vermeyen herhangi bir ürün, program ya da hizmet de kullanılabilir. Ancak, IBM dışı ürün, program ya da hizmetlerle gerçekleştirilen işlemlerin değerlendirilmesi ve doğrulanması kullanıcının sorumluluğundadır.

IBM'in, bu belgedeki konularla ilgili patentleri ya da patent başvuruları olabilir. Bu belgenin size verilmiş olması, patentlerin izinsiz kullanım hakkının da verildiği anlamına gelmez. Lisansla ilgili sorularınızı aşağıdaki adrese yazabilirsiniz:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Çift byte (DBCS) bilgilerle ilgili lisans soruları için, ülkenizdeki IBM'in Fikri Haklar (Intellectual Property) bölümüyle bağlantı kurun ya da sorularınızı aşağıda adrese yazın:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japonya

**Aşağıdaki paragraf, İngiltere ya da bu tür hükümlerin yerel yasalarla uyuşmadığı diğer ülkelerde geçerli değildir:** INTERNATIONAL BUSINESS MACHINES CORPORATION BU YAYINI, HAK İHLALİ YAPILMAYACAĞINA DAİR GARANTİLERLE TİCARİLİK VEYA BELİRLİ BİR AMACA UYGUNLUK İÇİN ZİMNİ GARANTİLER DE DAHİL OLMAK VE FAKS BUNLARLA SINIRLI OLMAMAK ÜZERE AÇIK YA DA ZİMNİ HİÇBİR GARANTİ VERMEKSİZİN "OLDUĞU GİBİ" ESASIYLA SAĞLAMAKTADIR. Bazı ülkeler bazı işlemlerde garantinin açık ya da örtük olarak reddedilmesine izin vermez; dolayısıyla, bu bildirim sizin için geçerli olmayabilir.

Bu yayın teknik yanlışlar ya da yazım hataları içerebilir. Buradaki bilgiler üzerinde düzenli olarak değişiklik yapılmaktadır; söz konusu değişiklikler sonraki basımlara yansıtılacaktır. IBM, önceden bildirimde bulunmaksızın, bu yayında açıklanan ürünler ve/ya da programlar üzerinde iyileştirmeler ve/ya da değişiklikler yapabilir.

Bu belgede IBM dışı Web sitelerine yapılan göndermeler kullanıcıya kolaylık sağlamak içindir ve bu Web sitelerinin onaylanması anlamına gelmez. Bu Web sitelerinin içerdiği malzeme, bu IBM ürününe ilişkin malzemenin bir parçası değildir ve bu tür Web sitelerinin kullanılmasının sorumluluğu size aittir.

IBM'e bilgi ilettiğinizde, IBM bu bilgileri size karşı hiçbir yükümlülük almaksızın uygun gördüğü yöntemlerle kullanabilir ya da dağıtabilir.

(i) Bağımsız olarak yaratılan programlarla, bu program da içinde olmak üzere diğer programlar arasında bilgi değiş tokuşuna ve (ii) değiş tokuş edilen bilginin karşılıklı kullanımına olanak sağlamak amacıyla bu program hakkında bilgi sahibi olmak isteyen lisans sahipleri şu adrese yazabilirler:

IBM Corporation  
Yazılım Birlikte Çalışabilirlik Koordinatörü, Bölüm 49XA  
3605 Highway 52 N

Rochester, MN 55901  
U.S.A.

Bu tür bilgiler, ilgili kayıt ve koşullar altında ve bazı durumlarda bedelli olarak edinilebilir.

Bu belgede açıklanan lisanslı program ve bu programla birlikte kullanılacak tüm lisanslı malzeme, IBM tarafından, IBM Müşteri Sözleşmesi, IBM Uluslararası Program Lisansı Sözleşmesi ya da eşdeğer herhangi bir sözleşmenin kayıt ve koşulları altında sağlanır.

Burada belirtilen performans verileri denetimli bir ortamda elde edilmiştir. Bu nedenle, başka işletim ortamlarında çok farklı sonuçlar alınabilir. Bazı ölçümler geliştirilme düzeyindeki sistemlerde yapılmıştır ve bu ölçümlerin genel kullanıma sunulan sistemlerde de aynı olacağı garanti edilemez. Ayrıca, bazı sonuçlar öngörü yöntemiyle elde edilmiş olabilir. Dolayısıyla, gerçek sonuçlar farklı olabilir. Bu belgenin kullanıcıları, kendi ortamları için geçerli verileri kendileri doğrulamalıdır.

IBM dışı ürünlerle ilgili bilgiler, bu ürünleri sağlayan firmalardan, bu firmaların yayın ve belgelerinden ve genel kullanıma açık diğer kaynaklardan alınmıştır. IBM bu ürünleri sınınamamıştır ve IBM dışı ürünlerle ilgili performans doğruluğu, uyumluluk gibi iddiaları doğrulayamaz. IBM dışı ürünlerin yeteneklerine ilişkin sorular, bu ürünleri sağlayan firmalara yöneltilmelidir.

IBM'in gelecekteki yönelim ve kararlarına ilişkin tüm bildirimler değişebilir ve herhangi bir duyuruda bulunulmadan bunlardan vazgeçilebilir; bu yönelim ve kararlar yalnızca amaç ve hedefleri gösterir.

Bu belge, günlük iş ortamında kullanılan veri ve raporlara ilişkin örnekler içerir. Örneklerin olabildiğince açıklayıcı olması amacıyla kişi, şirket, marka ve ürün adları belirtilmiş olabilir. Bu adların tümü gerçek dışıdır ve gerçek iş ortamında kullanılan ad ve adreslerle olabilecek herhangi bir benzerlik tümüyle rastlantıdır.

#### YAYIN HAKKI LİSANSI:

Bu belge, çeşitli işletim platformlarında programlama tekniklerini gösteren, kaynak dilde yazılmış örnek uygulama programları içerir. Bu örnek programları, IBM'e herhangi bir ödemede bulunmadan, örnek programların yazıldığı işletim altyapısına ilişkin uygulama programlama arabirimiyle uyumlu uygulama programlarının geliştirilmesi, kullanılması, pazarlanması ya da dağıtılması amacıyla herhangi bir biçimde kopyalayabilir, değiştirebilir ve dağıtabilirsiniz. Bu örnekler her koşul altında tüm ayrıntılarıyla sınınamamıştır. Dolayısıyla, IBM bu programların güvenilirliği, bakım yapılabilirliği ya da işlevleri konusunda açık ya da örtük güvence veremez.

Bu bilgileri elektronik kopya olarak görüntülediyseniz, fotoğraflar ve renkli resimler görünmeyebilir.

## Programlama arabirimi bilgileri

Programlama arabirimi bilgileri (sağlandıysa), bu programla birlikte kullanılmak üzere uygulama yazılımları yaratmanıza yardımcı olmak üzere hazırlanmıştır.

Bu kitap, müşterinin WebSphere MQ hizmetlerini edinmek üzere program yazmasına olanak tanıyan, amaçlanan programlama arabirimlerine ilişkin bilgiler içerir.

Ancak, bu bilgiler tanılama, değiştirme ve ayarlama bilgilerini de içerebilir. Tanılama, değiştirme ve ayarlama bilgileri, uygulama yazılımlarınızda hata ayıklamanıza yardımcı olur.

**Önemli:** Bu tanılama, değiştirme ve ayarlama bilgilerini bir programlama arabirimi olarak kullanmayın; bu, değişiklik söz konusu olduğunda kullanılır.

## Ticari Markalar

IBM, IBM logosu, ibm.com, IBM Corporation 'ın dünya çapında birçok farklı hukuk düzeninde kayıtlı bulunan ticari markalarıdır. IBM ticari markalarının güncel bir listesini Web üzerinde "Telif hakkı ve ticari marka bilgileri" [www.ibm.com/legal/copytrade.shtml](http://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: