

webMethods Adapter for JMS Installation and User's Guide

Version 6.1 SP1

January 2013

This document applies to webMethods Adapter for JMS 6.1 SP1 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2001-2021 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <https://softwareag.com/licenses/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

Document ID: ADAPTER-JMS-IUG-61 SP1-20210326

Table of Contents

About this Guide	5
Document Conventions.....	6
Online Information and Support.....	7
Data Protection.....	7
1 About the Adapter for JMS	9
Overview of the Adapter.....	10
Architecture and Components.....	10
Package Management.....	12
Adapter Connections.....	12
Adapter Services.....	15
Adapter Notifications.....	18
Using Version Control Systems to Manage Adapter Elements.....	20
Optimize Infrastructure Data Collector Support for Adapter for JMS.....	21
Viewing the Adapter's Update Level.....	21
Controlling Pagination.....	21
2 Installing, Upgrading, and Uninstalling the Adapter for JMS	23
Overview.....	24
Requirements.....	24
Installing Adapter for JMS 6.1 SP1.....	24
Upgrading to Adapter for JMS 6.1 SP1.....	26
Uninstalling Adapter for JMS 6.1 SP1.....	26
3 Package Management	29
Overview.....	30
Adapter for JMS Package Management.....	30
Group Access Control.....	33
Adapter for JMS in a Clustered Environment.....	33
4 Adapter Connections	37
Overview.....	38
Before Configuring or Managing Adapter Connections.....	38
Configuring Adapter for JMS Connections.....	38
Dynamically Changing a Service's Connection at Run Time.....	42
Viewing Adapter Connection Parameters.....	42
Editing Adapter Connections.....	44
Copying Adapter Connections.....	44
Deleting Adapter Connections.....	45
Enabling Adapter Connections.....	45
Disabling Adapter Connections.....	46

5 Adapter Services.....	47
Overview.....	48
Before Configuring or Managing Adapter Services.....	48
Configuring MessageProducer Services.....	48
Configuring MessageConsumer Services.....	50
Configuring MessageRequestor Services.....	53
Configuring Document-to-Message Conversion Services.....	55
Configuring Message-to-Document Conversion Services.....	59
Viewing Adapter Services.....	63
Editing Adapter Services.....	64
Deleting Adapter Services.....	64
6 Adapter Notifications.....	67
Overview.....	68
Before Configuring or Managing Notifications.....	68
Configuring MessageListener Notifications.....	69
Managing MessageListener Notifications.....	72
Using the Exactly Once Notification Feature.....	74
Viewing Notifications.....	74
Editing Notifications.....	74
Deleting Notifications.....	75
A Built-In Transaction Management Services.....	77
Transaction Management Overview.....	78
Built-In Transaction Management Services.....	80
pub.art.transaction:commitTransaction.....	80
pub.art.transaction:rollbackTransaction.....	81
pub.art.transaction:setTransactionTimeout.....	81
pub.art.transaction:startTransaction.....	82
Changing the Integration Server Transaction Timeout Interval.....	82
B WebLogic Server Configuration Notes.....	85
WebLogic Server Authentication.....	86

About this Guide

- Document Conventions 6
- Online Information and Support 7
- Data Protection 7

This guide describes how to configure and use the webMethods Adapter for JMS. It contains information for administrators and application developers who want to exchange messages with JMS providers using the JMS API.

To use this guide effectively, you should be familiar with:

- JMS API for Java
- JNDI APIs
- JMS Point to Point and Publish/Subscribe concepts (Queue and Topic)
- The terminology and basic operations of your operating system
- The setup and operation of the webMethods Integration Server
- The basic tasks of Software AG Designer

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at <https://empower.softwareag.com/>.

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.aspx and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 About the Adapter for JMS

■ Overview of the Adapter	10
■ Architecture and Components	10
■ Package Management	12
■ Adapter Connections	12
■ Adapter Services	15
■ Adapter Notifications	18
■ Using Version Control Systems to Manage Adapter Elements	20
■ Optimize Infrastructure Data Collector Support for Adapter for JMS	21
■ Viewing the Adapter's Update Level	21
■ Controlling Pagination	21

Overview of the Adapter

The webMethods Adapter for JMS is an add-on to the webMethods Integration Server that enables you to exchange messages with JMS providers through the use of the JMS APIs. The adapter provides seamless and real-time communication with JMS providers. Using the Adapter for JMS, webMethods clients can create and run services that can exchange messages from queues and topics on the JMS provider.

For a list of the JMS providers that the Adapter for JMS supports, see *webMethods Adapters System Requirements*.

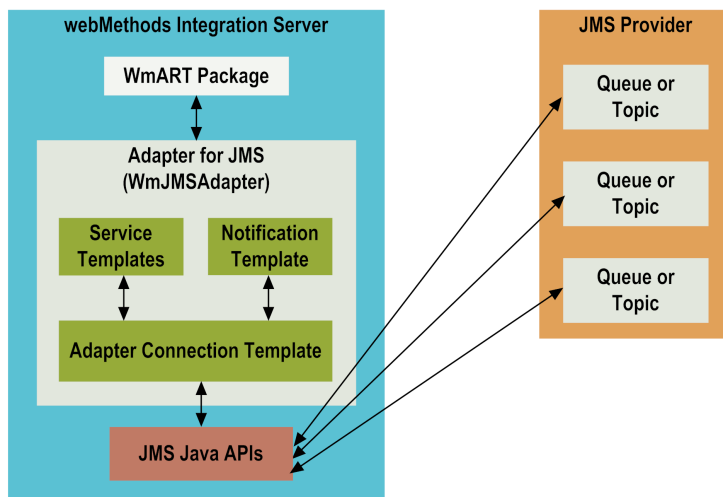
Architecture and Components

The Adapter for JMS enables you to configure the following components:

- **Adapter Connections:** Enable the webMethods product suite to connect to JMS providers at run time.
- **Adapter Services:** Perform operations on JMS providers using the built-in services provided with the Adapter for JMS.
- **MessageListener Notifications:** Provide Asynchronous notification of messages in queues and topics.

The following diagram illustrates how the Adapter for JMS interfaces with a JMS provider using JMS APIs. Following the diagram are descriptions of the different architectural pieces involved in the integration process.

For a more detailed description of the run time data flow for the different types of Adapter for JMS services, see [“Adapter Services” on page 15](#).



- **webMethods Integration Server.** The Adapter for JMS is installed and runs on the Integration Server.

- **WmART Package.** The WmART package provides a common framework for webMethods product suite 6.x and later adapters to utilize the Integration Server functionality, making the Integration Server the run-time environment for the Adapter for JMS. The WmART package is installed with the Integration Server.

- **Adapter for JMS.** The Adapter for JMS is delivered as a single package called WmJMSAdapter. The Adapter for JMS provides Integration Server Administrator and Designer user interfaces that enable you to configure and manage adapter connections, adapter services, and adapter notifications.

- **Adapter Connection Templates.** Adapter connections enable the Adapter for JMS to connect to JMS providers. You must configure an adapter connection before you can configure adapter services or adapter notifications. The Adapter for JMS connections use JNDI to look up the JMS connection factories.

You configure adapter connections using the Integration Server Administrator. For a detailed description of adapter connections, see [“Adapter Connections” on page 12](#).

- **Adapter Service Templates.** Adapter services enable the Adapter for JMS to produce and consume messages from JMS destinations (queues and topics). The Adapter for JMS provides adapter service templates that enable you to configure adapter services. The Adapter for JMS services use JNDI to look up the JMS destinations.

You configure adapter services in Designer using the Adapter for JMS adapter services templates. For a detailed description of adapter services, see [“Adapter Services” on page 15](#).

- **Adapter Notification Template.** Adapter notifications enable the Adapter for JMS to listen to and retrieve messages from JMS destinations (queues and topics). The Adapter for JMS provides an adapter notification template that enables you to configure adapter notifications. The Adapter for JMS notifications use JNDI to look up the JMS destinations.

You configure adapter notifications in Designer using the Adapter for JMS adapter notification templates. For a detailed description of adapter notifications, see [“Adapter Notifications” on page 18](#).

- **JMS APIs.** The Adapter for JMS uses JMS APIs to interface with JMS providers.

For more information about the JMS API libraries that the adapter requires, see [“Installing, Upgrading, and Uninstalling the Adapter for JMS” on page 23](#).

- **JMS Provider.** Adapter for JMS services and notifications are associated with specific JMS destinations (queues or topics) in the JMS provider.

The Adapter for JMS can retrieve configured `ConnectionFactory` and `Destination` class objects from a JNDI-supported naming service. The Adapter for JMS does not dictate how a JMS provider should publish connection factory and message destination entities. However, it supports only those JMS providers that publish these objects in a JNDI-supported naming service. To configure objects of these classes and to register them in a JNDI naming service, you must use the administrative tools provided by your JMS provider.

The `ConnectionFactory` class exposes methods that enable a JMS application to establish connections to the underlying messaging system.

Destination class objects identify:

- **Topics:** An application may publish to, or subscribe to, a topic.
- **Queues:** An application may send messages to, or receive messages from, a queue.

For a complete list of JMS providers that the Adapter for JMS supports, see [“Installing, Upgrading, and Uninstalling the Adapter for JMS” on page 23](#).

Package Management

The Adapter for JMS is provided as a package called WmJMSAdapter. You manage the adapter package the same way you manage any package on the Integration Server. There are several considerations regarding how you set up and manage your packages on the Integration Server, such as those described in the following list.

- Create user-defined packages for your connections and adapter services. For details, see [“Adapter for JMS Package Management” on page 30](#).
- Understand how package dependencies work so you make the best decisions regarding how you manage your connections and adapter services. For details, see [“Package Dependency Requirements and Guidelines” on page 31](#).
- Control which groups have access to which adapter services. For details, see [“Group Access Control” on page 33](#).
- Understand how clustering, an advanced feature of the webMethods Integration Server, works to effectively manage your adapter services. For details, see [“Adapter for JMS in a Clustered Environment” on page 33](#).
- Enable and disable packages. For details, see [“Enabling Packages” on page 31](#) and [“Disabling Packages” on page 32](#).

Adapter Connections

The Adapter for JMS uses adapter connections to define how its adapter services and adapter notifications connect to JMS providers. Each Adapter for JMS connection contains a set of connection parameters, including the logon parameters that the adapter requires to connect to the JMS provider. You must configure an adapter connection before you can configure an adapter service.

You configure connections using the Integration Server Administrator. For instructions for configuring and managing Adapter for JMS connections, see [“Adapter Connections” on page 37](#).

The Adapter for JMS has two types of connections:

- Adapter for JMS QueueConnection:
Select this connection type to connect to a queue.
- Adapter for JMS TopicConnection:
Select this connection type to connect to a topic.

Transaction Management of Adapter for JMS Connections

Adapter for JMS connections support the following transaction types:

Transaction Type	Description
XA_TRANSACTION	<p>This transaction type allows the connection to support two-phase transactions executed across multiple resources. In one transaction boundary, all of the operations on multiple connections will be committed or rolled back together. A transaction boundary means the scope of the transaction, from the beginning to the end of a transaction. It can be in one adapter service, one flow service, one Java service, or several steps in a flow service.</p> <p>The following JMS providers do not support the XA_TRANSACTION transaction type:</p> <ul style="list-style-type: none"> ■ webMethods Broker 8.2 SP2, 9.0 SP1 and 9.5 SP1 when used as a JMS provider ■ BEA WebLogic 10.3.1.0 JMS provider <p>Note: All of the connections involved in a two-phase transaction must support the XA_TRANSACTION transaction type.</p> <p>Note: You cannot use connections that use the XA_TRANSACTION transaction type with notifications.</p>
NO_TRANSACTION	<p>When receiving messages, the connection automatically acknowledges the received message. (Sessions are created with the CLIENT_ACKNOWLEDGE acknowledge mode.) When sending messages, the message is automatically delivered.</p> <p>For instructions on managing transactions manually, see “Built-In Transaction Management Services” on page 77.</p>
LOCAL_TRANSACTION	<p>The connection does not automatically commit transactions. You can manually define the transactions, or the Integration Server transaction manager will manage it for you. For instructions on managing transactions manually, see “Built-In Transaction Management Services” on page 77.</p>

When you define a connection, the transaction type that you choose determines the type of transaction management that the connection's operations use implicitly. Implicit transactions, which include the transactions types in the preceding table, are managed by the Integration Server transaction manager.

Note:

When using a MessageRequestor service, the requesting message that the service sends to the JMS destination cannot be rolled back if an operation fails within the transaction context. That is, the MessageRequestor service automatically commits the requesting message. If it did not, the JMS provider would never deliver the request and would therefore never receive a reply message. See the JMS Specification for more information about transaction management requirements in JMS providers.

You can also explicitly manage transactions using built-in services. For information about explicitly managing transactions, see [“Built-In Transaction Management Services” on page 77](#).

Connection Pooling

Integration Server includes a connection management service that dynamically manages connections and connection pools based on configuration settings that you specify for the connection. All adapter services use connection pooling.

A connection pool is a collection of connections with the same set of attributes. Integration Server maintains connection pools in memory. Connection pools improve performance by enabling adapter services to reuse open connections instead of opening new connections.

Run-Time Behavior of Connection Pools

When you enable a connection, Integration Server initializes the connection pool, creating the number of connection instances you specified in the connection's **Minimum Pool Size** field. Whenever an adapter service needs a connection, Integration Server provides a connection from the pool. If no connections are available in the pool, and the maximum pool size has not been reached, the server creates one or more new connections (according to the number specified in **Pool Increment Size**) and adds them to the connection pool. If the pool is full (as specified in **Maximum Pool Size**), the requesting service will wait for Integration Server to obtain a connection, up to the length of time specified in the **Block Timeout** field, until a connection becomes available. Periodically, Integration Server inspects the pool and removes inactive connections that have exceeded the expiration period that you specified in **Expire Timeout**.

If the connection pool initialization fails because of a network connection failure or some other type of exception, you can enable the system to retry the initialization any number of times, at specified intervals. For information about configuring connections, see [“Adapter Connections” on page 37](#).

Built-In Services For Connections

Integration Server provides built-in services that enable you to programmatically control connections. You can use them to enable and disable a connection, and to return usage statistics and the current state (Enabled or Disabled) and error status for a connection. These services are located in the WmART package, in the pub.art.connection folder.

Another built-in service, pub.art.service:setAdapterServiceNodeConnection, enables you to change the connection associated with an adapter service. For information, see [“Changing the Connection for an Adapter Service or Notification at Design Time” on page 16](#).

For details about the WmART services, see the *webMethods Integration Server Built-In Services Reference* for your release.

Adapter Services

Adapter services enable the Adapter for JMS to initiate operations that interact with JMS providers. You configure adapter services using adapter service templates, which are provided with the Adapter for JMS. Each template represents a specific operation. For example, the MessageProducer template enables you to send (publish) messages to a JMS queue or topic.

Adapter service templates contain all of the code that is necessary to perform an operation on a JMS provider, but without the data specifications. You provide these specifications when you configure an adapter service. Configuring a new service from an adapter service template is straightforward. Using Designer, you assign a default adapter connection to the service, select the adapter service template, and supply the service's data specifications.

After you configure a service, you can incorporate it into a flow or Java service to interact with a JMS provider. For example, you can create a MessageConsumer flow service that retrieves JMS messages from a specific queue, call a MessageToDocument service to convert the JMS message to an Integration Server document, and then call another service that routes the JMS message data to a different backend resource.

The Adapter for JMS provides the following adapter service templates:

Adapter Service Template	Description
MessageProducer	Sends (publishes) JMS messages to a queue or topic.
MessageConsumer	Receives (subscribes to) JMS messages from a queue or topic.
MessageRequestor	Sends JMS messages to a queue or topic and then waits for a reply JMS message.
DocumentToTextMessage	Converts an Integration Server document to a JMS text message.
TextMessageToDocument	Converts a JMS text message to an Integration Server document.
DocumentToObjectMessage	Converts an Integration Server document to a JMS object message.
ObjectMessageToDocument	Converts a JMS object message to an Integration Server document.
DocumentToMapMessage	Converts an Integration Server document to a JMS map message.
MapMessageToDocument	Converts a JMS map message to an Integration Server document.

Adapter Service Template	Description
DocumentToBytesMessage	Converts an Integration Server document to a JMS bytes message.
BytesMessageToDocument	Converts a JMS bytes message to an Integration Server document.
DocumentToStreamMessage	Converts an Integration Server document to a JMS stream message.
StreamMessageToDocument	Converts a JMS stream message to an Integration Server document.

Using MessageProducer, MessageConsumer, and MessageRequestor

The following steps lists the tasks required to use the MessageProducer, MessageConsumer, and MessageRequestor adapter services:

1. Configure an adapter connection using Integration Server Administrator. For more information, see [“Adapter Connections” on page 37](#).
2. Select the appropriate adapter service template and configure the adapter service using Designer. For more information, see [“Adapter Services” on page 47](#).
3. If you plan to use an Integration Server flow or Java service to invoke the adapter service, design the flow or Java service to use this adapter service using Designer. For more information, see the *webMethods Service Development Help* for your release.
4. Manage the adapter service using Designer and Integration Server Administrator. For more information, see [“Package Management” on page 29](#).

Changing the Connection for an Adapter Service or Notification at Design Time

Integration Server provides built-in services that you can use at design time to change the connection associated with an adapter service or notification. These built-in services are named `pub.art.service:setAdapterServiceNodeConnection` and `pub.art.notification:setPollingNotificationNodeConnection`. Using these services, you can change the specific connection associated with an adapter service or an adapter notification at design time so that you do not need to create and maintain multiple adapter services or notifications.

Note:

These built-in services can be run at design time only; do not use them within an Integration Server flow or Java service. You must run this service directly from Designer by selecting the service and running it.

For details about these services, see the *webMethods Integration Server Built-In Services Reference* for your release.

Other built-in services enable you to control connections; for more information, see [“Built-In Services For Connections” on page 14](#).

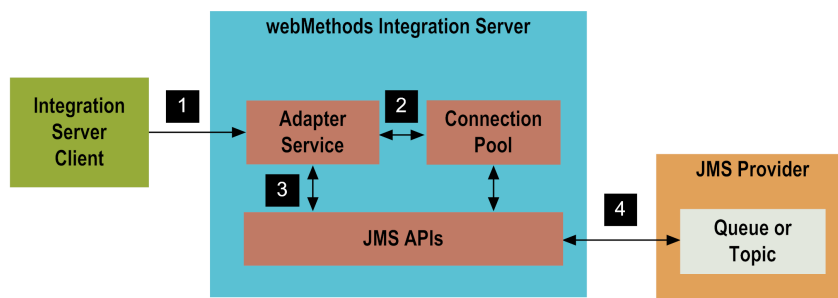
Changing the Connection for an Adapter Service at Run Time

Integration Server enables you to dynamically select the connection a service uses to interact with the adapter's resource. This feature enables one service to interact with multiple, similar backend resources.

For more information about overriding a service's default connection at run time, see [“Dynamically Changing a Service's Connection at Run Time” on page 42](#).

Adapter Service Run-Time Processing

The following diagram illustrates how the Adapter for JMS processes MessageConsumer, MessageProducer, and MessageRequestor services at run time.



Step	Description
1	An Integration Server client, typically using a flow or Java service, invokes a Adapter for JMS service on webMethods Integration Server to perform an operation on a JMS provider. You configured the adapter service earlier using Designer.
2	The adapter service gets a connection from the service's connection pool. Adapter connections contain connection information for the JMS provider.
3	The adapter service uses the JMS APIs to operate on the JMS destination. You created and enabled the adapter connection earlier using Integration Server Administrator.
4	The adapter service retrieves a JMS message from the JMS destination or places a JMS message on the JMS destination as follows, depending on the type of service:

Step	Description
	<ul style="list-style-type: none">■ For MessageConsumer services, the adapter service retrieves a JMS message from a queue or topic.■ For MessageProducer services, the adapter service places a JMS message on a queue or topic.■ For MessageRequestor services, the adapter service sends a JMS message to a queue or topic and then waits for a reply from an implicit temporary queue or topic.

Adapter Notifications

The Adapter for JMS provides a notification called `MessageListener`, which monitors a specified JMS destination (queue or topic) and notifies the adapter when a message arrives at that destination.

You can configure the notification to either publish the notification's associated publishable document or you can configure the notification to invoke a flow or Java service using the data from the notification.

If the notification publishes the document, you can use an Integration Server trigger to monitor for a given publishable document associated with a notification. When Integration Server receives this publishable document, it invokes a flow or Java service that is registered with the trigger. The flow or Java service then reacts to the data changes contained in the notification's publishable document. For more details about Integration Server publishable documents, see the *Publish-Subscribe Developer's Guide* for your release.

Note:

If you need to use a Client ID for the notifications, it is better to specify it along with the other notification parameters using Designer, and not while creating the connection factory or the Adapter for JMS connections. When the Client ID is specified in the connection factory or JMS connections, only one connection can be created through this connection factory or connection. This limits the Adapter for JMS from leveraging the connection pool.

For instructions for configuring `MessageListener` notifications, see [“Configuring MessageListener Notifications” on page 69](#).

Considerations When Using MessageListener Notifications

- If you are using the `MessageListener` notification on an Integration Server that does not have a Broker configured (and does not have a service associated with the notification), the notification publishes the document in local mode.
- When an Integration Server hosts a `MessageListener` notification that is configured to publish a document, only the Integration Server that hosts the notification should subscribe to the notification's publishable document type. This is because a different Integration Server will not be able to understand the document's serialized message unless that Integration Server also has an Adapter for JMS installed.

- When you configure a MessageListener notification, the adapter creates a publishable document type to use with the notification. When you associate a MessageListener notification with a service (rather than publish the document), the service must use the same input signature as the notification's publishable document type would use.

Exactly Once Notifications

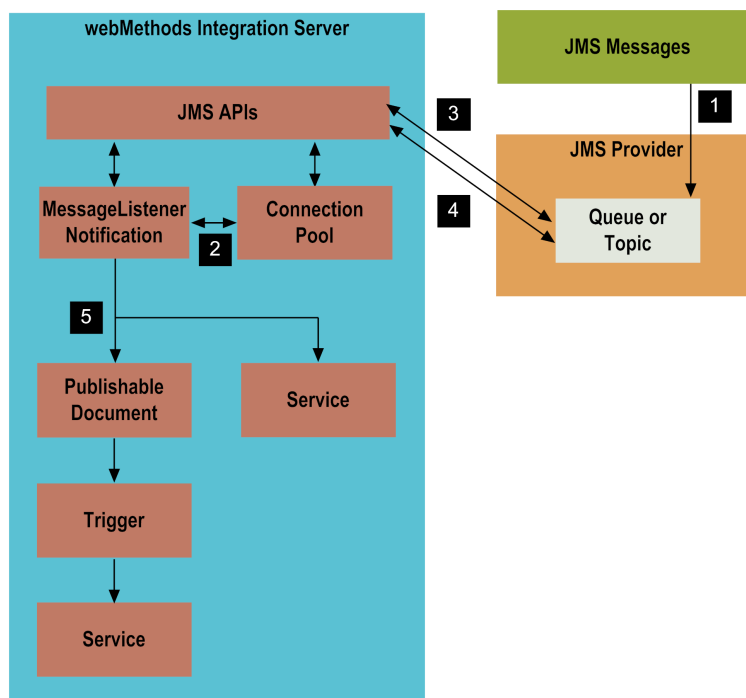
Adapter notifications can use the Exactly Once notification feature. This feature ensures that notification data will not be duplicated even if a failure occurs during processing. This is achieved by assigning unique IDs for each publishable document. After a processing failure, Integration Server checks for duplicate records in storage and ignores any duplicate IDs.

Note:

Exactly Once Notifications are only supported when you use the notification to publish messages, not when you associate the notification data directly with a service.

MessageListener Notification Run-Time Processing

The following diagram and steps illustrate what happens at run time when MessageListener notifications are invoked. Integration Server continues to invoke the notification periodically, as defined when you configured the schedule parameters for the MessageListener notification.



Step	Description
1	JMS messages are placed on a JMS destination (queue or topic).
2	The notification gets a connection from the notification's connection pool.

Step	Description
3	The notification uses the JMS APIs to monitor the JMS destination.
4	The notification retrieves the JMS messages from the queue or topic.
5	<p>The notification does one of the following actions, depending on how you configured the notification:</p> <ul style="list-style-type: none">■ The notification creates a publishable document, which contains the JMS message, and then publishes the document. <p>When you use this option, you can create an integration scenario that uses an Integration Server trigger to invoke a flow or Java service using the data contained in the publishable document.</p> <ul style="list-style-type: none">■ The notification invokes a service in Integration Server. To use this option, you specify the name of the service to be invoked when you configure the notification. <p>For more details about Integration Server publishable documents, see the <i>Publish-Subscribe Developer's Guide</i> for your release.</p>

Using MessageListener Notifications

The following steps lists the tasks required to use these types of notification:

1. Create an adapter connection using Integration Server Administrator. See [“Configuring Adapter for JMS Connections” on page 38](#) for details.
2. Configure the notification using Designer. See [“Configuring MessageListener Notifications” on page 69](#) for instructions for configuring notifications.
3. If you plan to use an Integration Server flow or Java service, design it to react to the data changes contained in the notification's publishable document. Create the Integration Server trigger to use the notification's publishable document using Designer. For details, see the *webMethods Service Development Help* for your release.
4. Schedule and enable the adapter notification using Integration Server Administrator. See [“Managing MessageListener Notifications” on page 72](#) for instructions.
5. Manage the notification using Designer and Integration Server Administrator. See [“Package Management” on page 29](#) and [“Adapter Notifications” on page 67](#) for details.

Using Version Control Systems to Manage Adapter Elements

The adapter supports the Version Control System (VCS) Integration feature provided by Designer. When you enable the feature in Integration Server, you can check adapter packages or elements into and out of your version control system from Designer. For more information about the VCS Integration feature, see the *Administering the VCS Integration Feature*.

Beginning with Integration Server 8.2 SP3, the adapter supports the local service development feature in Designer. This feature extends the functionality of the VCS Integration feature to check package elements and their supporting files into and out of a VCS directly from Designer. For more information about local service development and how it compares to the VCS Integration feature, see the *webMethods Service Development Help*.

Optimize Infrastructure Data Collector Support for Adapter for JMS

Optimize Infrastructure Data Collector monitors the system and operational data associated with webMethods run-time components such as Integration Servers, Broker Servers, Brokers, and adapters, and reports the status of these components on Optimize for Infrastructure or other external tools. When you start monitoring an Integration Server, Infrastructure Data Collector automatically starts monitoring all ART-based adapters that are installed on Integration Server.

For information about monitored key performance indicators (KPIs) collected for the monitored adapter components, see *Administering webMethods Optimize* for your release.

Viewing the Adapter's Update Level

You can view the list of updates that have been applied to the adapter. The list of updates appears in the **Updates** field on the adapter's About page in Integration Server Administrator.

Controlling Pagination

You can control the number of items that are displayed on the adapter Connections screen, Listeners screen, and Listener Notifications screen. By default, 10 items are displayed per page. Click **Next** and **Previous** to move through the pages, or click a page number to go directly to a page.

To change the number of items displayed per page, set the `watt.art.page.size` property and specify a different number of items.

➤ To set the number of items per page

1. From Integration Server Administrator, click **Settings > Extended**.
2. Click **Edit Extended Settings**. In the Extended Settings editor, add or update the `watt.art.page.size` property to specify the preferred number of items to display per page. For example, to display 50 items per page, specify:

```
watt.art.page.size=50
```

3. Click **Save Changes**. The property appears in the Extended Settings list.

For more information about working with extended configuration settings, see the *webMethods Integration Server Administrator's Guide* for your release.

2 Installing, Upgrading, and Uninstalling the Adapter for JMS

■ Overview	24
■ Requirements	24
■ Installing Adapter for JMS 6.1 SP1	24
■ Upgrading to Adapter for JMS 6.1 SP1	26
■ Uninstalling Adapter for JMS 6.1 SP1	26

Overview

This chapter explains how to install, upgrade, and uninstall webMethods Adapter for JMS 6.1 SP1. The instructions use the Software AG Installer and the Software AG Uninstaller wizards. For complete information about the wizards or other installation methods, or to install other webMethods products, see *Installing webMethods Products On Premises* for your release.

Requirements

For a list of the operating systems, JMS products, and webMethods products supported by Adapter for JMS 6.1 SP1, see *webMethods Adapters System Requirements*.

Adapter for JMS 6.1 SP1 has no hardware requirements beyond those of its host Integration Server.

Installing Adapter for JMS 6.1 SP1

If you are installing the adapter in a clustered environment, you must install it on each Integration Server in the cluster, and each installation must be identical. For more information about working with the adapter in a clustered environment, see [“Adapter for JMS in a Clustered Environment” on page 33](#).

➤ To install Adapter for JMS 6.1 SP1

1. Download Software AG Installer from the [Empower Product Support Web site](#).
2. If you are installing the adapter on an existing Integration Server, shut down the Integration Server.
3. Start the Software AG Installer wizard.
4. Choose the webMethods release that includes the Integration Server on which to install the adapter. For example, if you want to install the adapter on Integration Server 8.2, choose the 8.2 release.
5. Specify the installation directory as follows:
 - If you are installing on an existing Integration Server, specify the webMethods installation directory that contains the host Integration Server.
 - If you are installing both the host Integration Server and the adapter, specify the installation directory to use.
6. In the product selection list, select **Adapters > webMethods Adapter for JMS 6.1 SP1**. You can also choose to install documentation and these items:

If you want to...	Choose to install this...
Use Adapter for JMS with webMethods Broker as a JMS provider	webMethods Broker client libraries for JMS
Use Adapter for JMS with a file-based JNDI implementation	Oracle JNDI libraries

If you do not want to install these libraries now, you can do so later manually. For instructions, see [“Adding the JMS Provider Client Libraries to the Integration Server Classpath” on page 25](#).

Software AG Installer installs the adapter in the *Integration Server_directory/packages* directory.

When installing the module on Integration Server 9.6 and above, you can choose to install the package in the default instance. In this case, the package will be installed both in the package repository and the default instance packages directory located in *Integration Server_directory\instances\default\packages* directory.

You can move the package to an Integration Server instance of your choice by using the *is_instance* Script. For more information about using *is_instance* script, see the *webMethods Integration Server Administrator's Guide*.

7. After installation is complete, start the host Integration Server.

Adding the JMS Provider Client Libraries to the Integration Server Classpath

Integration Server requires access to some of the JMS provider's Java client libraries. Place these libraries in the server's classpath by copying them into the *Integration Server_directory\instances\instance_name\packages\WmJMSAdapter\code\jars\static* directory, as described in the following sections. The server will automatically add the .jar files to its classpath.

➤ To add the JMS provider's client libraries to the Integration Server classpath

1. If you are going to use Adapter for JMS with a file-based JNDI implementation, download the *fscontext-1_2_beta3.zip* file from the Oracle website.

From the **Download JNDI 1.2.1 & More** menu, select **File System Service Provider, 1.2 Beta 3** to download the *fscontext-1_2_beta3.zip* file.

2. Extract the *fscontext.jar* and *providerutil.jar* files from the *fscontext-1_2_beta3.zip* file and copy the files to the *Integration Server_directory\instances\instance_name\packages\WmJMSAdapter\code\jars\static* directory.
3. Copy the Java API libraries from WebLogic 10.3.1.0 server to your Integration Server.
 - a. Create *wlfullclient.jar* from the WebLogic installation using the following command: `java -jar wljarbuilder.jar`

- b. Place the `wlfullclient.jar` in the *Integration Server_directory* `\instances\instance_name\packages\WmJMSAdapter\code\jars\static` directory.
4. Restart the host Integration Server.

Upgrading to Adapter for JMS 6.1 SP1

➤ To upgrade to Adapter for JMS 6.1 SP1

1. Back up the `WmJMSAdapter` package and any existing custom adapter packages.
2. Uninstall the older version of the adapter. For information about how to uninstall Adapter for JMS, see *Installing webMethods Products On Premises* for your release.
3. Delete the `WmJMSAdapter` package from the *Integration Server_directory* `\instances\instance_name\packages` directory.

You should not delete any other packages, such as packages that contain connections and services, even when the packages have a dependency on the `WmJMSAdapter` package.

4. Install Adapter for JMS as described in “[Installing Adapter for JMS 6.1 SP1](#)” on page 24.
5. Copy the JMS provider client libraries from the old adapter installation to the *Integration Server_directory* `\instances\instance_name\packages\WmJMSAdapter\code\jars\static` directory.
6. Restart Integration Server.

Uninstalling Adapter for JMS 6.1 SP1

➤ To uninstall Adapter for JMS 6.1 SP1

1. Shut down the Integration Server that hosts Adapter for JMS.
2. Start Software AG Uninstaller, selecting the *Software AG_directory* that contains the host Integration Server. In the product selection list, select **Adapters > webMethods Adapter for JMS 6.1 SP1**. You can also choose to uninstall documentation and these items, if you installed them:
 - webMethods Broker client libraries for JMS
 - Oracle JNDI libraries
3. Restart the host Integration Server.

4. Uninstaller removes all files related to Adapter for JMS that were installed into the *Integration Server_directory* \packages\WmJMSAdapter directory. However, Uninstaller does not delete files created after you installed the adapter (for example, user-created or configuration files), nor does it delete the adapter directory structure. You can go to the *Integration Server_directory* \packages directory and *Integration Server_directory* \instances\default\packages. Delete the WmJMSAdapter directory.

3 Package Management

- Overview 30
- Adapter for JMS Package Management 30
- Group Access Control 33
- Adapter for JMS in a Clustered Environment 33

Overview

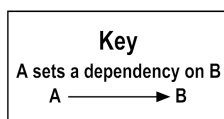
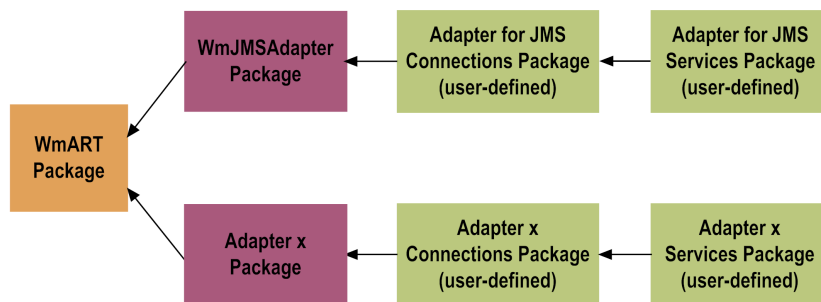
The following sections describe how to set up and manage your Adapter for JMS packages, set up Access Control Lists (ACL), and use the adapter in a clustered environment.

Adapter for JMS Package Management

The Adapter for JMS is provided as a package called WmJMSAdapter. You manage the WmJMSAdapter package as you would manage any package on Integration Server.

When you create connections, adapter services, and adapter notifications, define them in user-defined packages rather than in the WmJMSAdapter package. Doing so will allow you to manage the package more easily.

As you create user-defined packages in which to store connections, adapter services, and adapter notifications, use the package management functionality provided in Designer and set the user-defined packages to have a dependency on the WmJMSAdapter package. That way, when the WmJMSAdapter package loads or reloads, the user-defined packages load automatically. See the following diagram:



Package management tasks include:

- [Setting package dependencies \(see “Package Dependency Requirements and Guidelines” on page 31\)](#)
- [“Enabling Packages” on page 31](#)
- [“Disabling Packages” on page 32](#)
- [“Importing and Exporting Packages” on page 32](#)

Package Dependency Requirements and Guidelines

This section contains a list of dependency requirements and guidelines for user-defined packages. For instructions for setting package dependencies, see the *webMethods Service Development Help* for your release.

- A user-defined package must have a dependency on its associated adapter package, WmJMSAdapter. (The WmJMSAdapter package has a dependency on the WmART package.)
- Package dependencies ensure that at startup Integration Server automatically loads or reloads all packages in the proper order: the WmART package first, the adapter package next, and the user-defined package(s) last. The WmART package is automatically installed when you install Integration Server. You should not need to manually reload the WmART package.
- If the connections and adapter services of an adapter are defined in different packages, then:
 - A package that contains the connection(s) must have a dependency on the adapter package.
 - Packages that contain adapter services must have a dependency on their associated connection package.
- Keep connections for different adapters in separate packages so that you do not create interdependencies between adapters. If a package contains connections for two different adapters, and you reload one of the adapter packages, the connections for both adapters will reload automatically.
- Integration Server will not allow you to enable a package if it has a dependency on another package that is disabled. That is, before you can enable your package, you must enable all packages on which your package depends. For information about enabling packages, see [“Enabling Packages” on page 31](#).
- Integration Server will allow you to disable a package even if another package that is enabled has a dependency on it. Therefore, you must manually disable any user-defined packages that have a dependency on the adapter package before you disable the adapter package. For information about disabling packages, see [“Disabling Packages” on page 32](#).
- You can name connections, adapter services, and notifications the same name provided that they are in different folders and packages.

Enabling Packages

All packages are automatically enabled by default. To enable a package manually, use the following procedure.

Note:

Enabling an adapter package will not cause its associated user-defined packages to be reloaded. For information about reloading packages, see the *webMethods Service Development Help* for your release.

Important:

Before you manually enable a user-defined package, you must first enable its associated adapter package (WmJMSAdapter).

> To enable a package

1. Open Integration Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Management**.
3. Click **No** in the **Enabled** column. The server displays a ✓ and **Yes** in the **Enabled** column.

Disabling Packages

When you want to temporarily prohibit access to the elements in a package, disable the package. When you disable a package, the server unloads all of its elements from memory. Disabling a package prevents Integration Server from loading that package at startup.

A disabled adapter will:

- Remain disabled until you explicitly enable it using Integration Server Administrator.
- Not be listed in Software AG Designer.

Note:

If your adapter has multiple user-defined packages, and you want to disable some of them, disable the adapter package first. Otherwise, errors will be issued when you try to access the remaining enabled user-defined packages.

> To disable a package

1. Open Integration Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Management**.
3. Click **Yes** in the **Enabled** column for the package that you want to disable. The server issues a prompt to verify that you want to disable the package. Click **OK** to disable the package. When the package is disabled, the server displays **No** in the **Enabled** column.

Importing and Exporting Packages

You import and export packages using Designer. Exporting allows you to export the package to a .zip file and save it to your hard drive. The .zip file can then be imported for use by another package.

Important:

Do not rename packages you export; the rename function is comparable to moving a package, and when you import the renamed package, you lose any triggers, connections, and notifications associated with this package.

For details about importing and exporting packages, see the *webMethods Service Development Help* for your release.

Group Access Control

To control which development group has access to which adapter services, use access control lists (ACLs). You can use ACLs to prevent one development group from inadvertently updating the work of another group, or to allow or deny access to services that are restricted to one group but not to others.

For general information about assigning and managing ACLs, see the *webMethods Service Development Help* for your release.

Adapter for JMS in a Clustered Environment

What is webMethods Integration Server Clustering?

Clustering is an advanced feature of webMethods Integration Server that substantially extends the reliability, availability, and scalability of Integration Server.

Clustering accomplishes this by providing the infrastructure and tools to deploy multiple Integration Servers as if they were a single virtual server and to deliver applications that leverage that architecture. Because this activity is transparent to the client, clustering makes multiple servers look and behave as one.

Integration Server 8.2 SP2 and higher supports the caching and clustering functionality provided by Terracotta. Caching and clustering are configured at the Integration Server level and Adapter for JMS uses the caching mechanism that is enabled on Integration Server.

With clustering, you get the following benefits:

- **Load balancing.** This feature, provided automatically when you set up a clustered environment, allows you to spread the workload over several servers, thus improving performance and scalability.
- **Failover support.** Clustering enables you to avoid a single point of failure. If a server cannot handle a request, or becomes unavailable, the request is automatically redirected to another server in the cluster.

Note:

webMethods Integration Server clustering redirects HTTP and HTTPS requests, but does not redirect FTP or SMTP requests.

- **Scalability.** You can increase your capacity even further by adding new machines running webMethods Integration Server to the cluster.

Note:

The Adapter for JMS MessageListener notifications are not supported in a clustered environment.

For details on webMethods Integration Server clustering, see the *webMethods Integration Server Clustering Guide* for your release.

Configuring the Adapter for JMS

When you configure the Adapter for JMS to configure adapter services, you must:

- Ensure that each webMethods Integration Server in the cluster contains an identical set of packages.
- Disable the redirection capability for certain predefined administrative services.

Replicating Packages to webMethods Integration Servers

Every webMethods Integration Server in the cluster should contain an identical set of packages that you define using the Adapter for JMS; that is, you should replicate the Adapter for JMS services and the connections they use. You do not replicate the polling notifications.

To ensure consistency, we recommend that you create all packages on one server, and replicate them to the other servers. If you allow different servers to contain different services, you might not derive the full benefits of clustering. For example, if a client requests a service that resides in only one server, and that server is unavailable, the request cannot be successfully redirected to another server.

For information about replicating packages, see the *webMethods Integration Server Administrator's Guide* for your release.

Disabling the Redirection of Administrative Services

As mentioned in “[What is webMethods Integration Server Clustering?](#)” on page 33, a server that cannot handle a client's service request can automatically redirect the request to another server in the cluster. However, the Adapter for JMS uses certain predefined administrative services that you should not allow to be redirected. These services are used internally when you configure the adapter. If you allow these services to be redirected, your configuration specifications might be saved on multiple servers, which is an undesirable result. For example, if you configure two Adapter for JMS services, one might be stored on one server, while the other one might be stored on another server. Remember that all adapter services must reside on all webMethods Integration Servers in the cluster.

➤ To disable the redirection of administrative services

1. Shut down Integration Server Administrator. For the procedure to do this, see the *webMethods Integration Server Administrator's Guide* for your release.
2. Edit the following file:

IntegrationServer_directory\config\redir.cnf

3. Add the following line to the file:

```
<value name="wm.art">false</value>
```

4. Save the file and restart webMethods Integration Server.

Clustering Considerations and Requirements

Note:

The following sections assume that you have already configured the webMethods Integration Server cluster. For details about webMethods clustering, see the *webMethods Integration Server Clustering Guide* for your release.

The following considerations and requirements apply to the Adapter for JMS in a clustered environment.

Requirements of Each Integration Server in a Cluster

The following table describes the requirements of each Integration Server in a given cluster:

All Integration Servers in a given cluster must have identical...	For Example...
Integration Server versions	All Integration Servers in the cluster must be the same version, with the same service packs and fixes applied.
Adapter packages	All adapter packages on one Integration Server should be replicated to all other Integration Servers in the cluster.
Adapter versions	All adapters in the cluster must be the same version.
Adapter connections	<p>If you configure a connection to the database, this connection must appear on all servers in the cluster so that any Integration Server in the cluster can handle a given request identically.</p> <p>If you plan to use connection pools in a clustered environment, see “Considerations When Configuring Connections with Connection Pooling Enabled” on page 36.</p>
Adapter services	If you configure a specific InsertSQL Adapter Service, this same adapter service must appear on all servers in the cluster so that any Integration Server in the cluster can handle the request identically.

All Integration Servers in a given cluster For Example... must have identical...

If you allow different Integration Servers to contain different services, you might not derive the full benefits of clustering. For example, if a client requests a service that resides on only one server, and that server is unavailable, the request cannot be successfully redirected to another server.

See [“Replicating Packages to webMethods Integration Servers” on page 34](#) for information about replicating adapter packages, connections, and adapter services across multiple Integration Servers in a cluster.

Considerations When Installing Adapter for JMS Packages

For each Integration Server in the cluster, use the standard Adapter for JMS installation procedures for each machine, as described in [“Installing, Upgrading, and Uninstalling the Adapter for JMS” on page 23](#).

Considerations When Configuring Connections with Connection Pooling Enabled

When you configure a connection that uses connection pools in a clustered environment, be sure that you do not exceed the total number of connections that can be opened simultaneously for that database.

For example, if you have a cluster of two Integration Servers with a connection configured to a database that supports a maximum of 100 connections opened simultaneously, the total number of connections possible at one time must not exceed 100. This means that you cannot configure a connection with an initial pool size of 100 and replicate the connection to both servers, because there could be possibly a total of 200 connections opened simultaneously to this database.

In another example, consider a connection configured with an initial pool size of 10 and a maximum pool size of 100. If you replicate this connection across a cluster with two Integration Servers, it is possible for the connection pool size on both servers to exceed the maximum number of database connections that can be open at one time.

For information about configuring connections for the Adapter for JMS, see [“Configuring Adapter for JMS Connections” on page 38](#).

For more general information about connection pools, see the *webMethods Integration Server Administrator’s Guide* for your release.

4 Adapter Connections

■ Overview	38
■ Before Configuring or Managing Adapter Connections	38
■ Configuring Adapter for JMS Connections	38
■ Dynamically Changing a Service's Connection at Run Time	42
■ Viewing Adapter Connection Parameters	42
■ Editing Adapter Connections	44
■ Copying Adapter Connections	44
■ Deleting Adapter Connections	45
■ Enabling Adapter Connections	45
■ Disabling Adapter Connections	46

Overview

This chapter describes how to configure and manage Adapter for JMS connections. For more information about how adapter connections work, see [“Adapter Connections” on page 12](#).

Before Configuring or Managing Adapter Connections

Perform the following tasks before configuring or managing adapter connections.

➤ To prepare to configure or manage adapter connections

1. Install webMethods Integration Server and the Adapter for JMS on the same machine, and add the JMS client libraries to the Integration Server classpath. See [“Installing, Upgrading, and Uninstalling the Adapter for JMS” on page 23](#) for details.
2. Make sure you have webMethods Integration Server administrator privileges so that you can access the Adapter for JMS administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator’s Guide* for your release.
3. Start your Integration Server and Integration Server Administrator, if they are not already running.
4. Using Integration Server Administrator, make sure the WmJMSAdapter package is enabled. See [“Enabling Packages” on page 31](#) for instructions.
5. Using Designer, create a user-defined package to contain the connection, if you have not already done so. For more information about managing packages for the adapter, see [“Adapter for JMS Package Management” on page 30](#).
6. Create your connections, as described in [“Configuring Adapter for JMS Connections” on page 38](#).

Configuring Adapter for JMS Connections

When you configure Adapter for JMS connections, you specify information that Integration Server uses to connect to a JMS provider. You configure Adapter for JMS connections using Integration Server Administrator.

➤ To configure a connection

1. In the **Adapters** menu in the Integration Server Administrator navigation area, click **Adapter for JMS**.
2. On the Connections screen, click **Configure New Connection**.

3. On the Connection Types screen, click the connection type with which you want the adapter to connect to the JMS provider.
 - Select **Adapter for JMS QueueConnection** as the connection type to connect to a queue.
 - Select **Adapter for JMS TopicConnection** as the connection type to connect to a topic.
4. In the **Adapter for JMS Connection** section, provide values for the following fields.

Parameter	Description/Action
Package	<p>The package in which to configure the connection. You must create the package using Designer before you can specify it for this parameter. For general information about creating packages, see the <i>webMethods Service Development Help</i> for your release.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: Configure the connection in a user-defined package rather than in the adapter's package. See “Package Management” on page 29 for other important considerations when creating packages for use with the Adapter for JMS.</p> </div>
Folder Name	The folder in which to configure the connection.
Connection Name	The name you want to give the connection. Connection names cannot have spaces or use special characters reserved by Integration Server or Designer. For more information about the use of special characters in package, folder, and element names, see the <i>webMethods Service Development Help</i> for your release.

5. In the **Connection Properties** section, provide values for the following fields:

Parameter	Description
Transaction Type	<p>The type of transaction support that the connection provides. Select one of the following transaction types:</p> <ul style="list-style-type: none"> ■ XA_TRANSACTION: The connection uses XA transactions. ■ NO_TRANSACTION: When receiving messages, the connection automatically acknowledges the received message. (Sessions are created with the CLIENT_ACKNOWLEDGE acknowledge mode.) When sending messages, the message is automatically delivered. ■ LOCAL_TRANSACTION: The connection does not automatically commit transactions. You can manually define the transactions, or the Integration Server transaction manager will manage it for you. For instructions on managing transactions manually, see “Built-In Transaction Management Services” on page 77.

Parameter	Description
JNDI Initial Context Factory	The fully qualified class name for the Initial Context Factory used by your JNDI provider. For example: <code>weblogic.jndi.WLInitialContextFactory</code> .
JNDI Provider URL	The URL to the starting context within the JNDI directory service. For example: <code>t3://MyHost.MyCompany.com:7001</code> .
JNDI Security Principal	(Optional) Your JNDI login name or security principal, for JNDI lookup purposes.
JNDI Security Credentials	(Optional) Your JNDI login password or security credentials, for JNDI lookup purposes.
JNDI Other Properties	<p>(Optional) Enables you to provide additional JNDI properties. The properties you specify in this field will differ depending on the specific JNDI Naming Context you are using. Specify properties in the following format, separating multiple properties using a semicolon (;): <code>propertyName1=value1; propertyName2=value2</code>. For JBoss connections, to define the security protocol at the server side to the default authentication module, specify the <code>java.naming.security.protocol</code> property with a value of "other", as follows:</p> <pre>java.naming.security.protocol=other</pre> <p>Alternatively, if you need to assign the security protocol to use a custom module defined in the <code>client-login conf/login-config.xml</code> entry in the JBoss server, specify the name of the custom module as the value of the <code>java.naming.security.protocol</code> property:</p> <pre>java.naming.security.protocol=CustomModule</pre>
Topic/Queue ConnectionFactory JNDI Name	The full path name identifying the JMS Queue or Topic ConnectionFactory bound with the JNDI service. For Topic connections provide Topic ConnectionFactory Name and for Queue connections provide Queue ConnectionFactory Name.
JMS User	(Optional) Your JMS login name to connect to the JMS provider.
JMS Password	(Optional) Your JMS login password for connection purposes.
Connection Client ID	<p>(Adapter for JMS TopicConnection only) A unique client identifier for this connection. The purpose of the client identifier is to associate a connection and its objects with a state maintained on behalf of the client by the provider. The only such state identified by the JMS API is that required to support durable subscriptions.</p> <p>Note: You can only use this property if there is no default Client ID stored with the topic connection factory.</p>

Parameter	Description
	<p>Note: If you need to use a Client ID for the notifications, it is better to specify it along with the other notification parameters using Designer, and not while creating the connection factory or the Adapter for JMS connections. When the Client ID is specified in the connection factory or JMS connections, only one connection can be created through this connection factory or connection. This limits the Adapter for JMS from leveraging the connection pool.</p>

6. In the **Connection Management Properties** section, provide values for the following fields:

Parameter	Description
Enable Connection Pooling	<p>Enables the adapter to use connection pooling. Default: true. See “Adapter Connections” on page 12 for more information about connection pooling.</p> <p>Note: If you plan to enable connection pooling in a clustered environment, consider the connection pool size. For details, see “Considerations When Configuring Connections with Connection Pooling Enabled” on page 36.</p>
Minimum Pool Size	If connection pooling is enabled, this field specifies the minimum number of connection objects that remain in the connection pool at all times. When the adapter creates the pool, it create this number of connections. Default: 1.
Maximum Pool Size	The maximum number of connection objects that can exist in the connection pool. The adapter will reuse any inactive connections in the pool or, if all connections are active and the connection pool has reached its maximum size, the adapter will wait for a connection to become available. Default: 10.
Pool Increment Size	If connection pooling is enabled, this field specifies the number of connections by which the pool will be incremented if connections are needed, up to the maximum pool size. Default: 1.
Block Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that Integration Server will wait to obtain a connection before it times out and returns an error. Default: 1000.
Expire Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that an inactive connection can remain in the pool before it is closed and removed from the pool. For example, to specify 10 seconds, specify 10000. Enter 0 to specify no timeout. Default: 1000.

Parameter	Description
	Note: Note that the adapter will never violate the Minimum Pool Size parameter. These connections remain in the pool regardless of how long they are inactive.
Startup Retry Count	The number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails. Default: 0.
Startup Backoff Timeout	The number of seconds that the system should wait between attempts to initialize the connection pool.

7. Click **Save Connection**.

The connection you configured appears on the adapter's Connections screen and in Designer's Package Navigator.

By default, when you configure a connection, it is not enabled. For information about enabling connections, see [“Enabling Adapter Connections” on page 45](#).

Dynamically Changing a Service's Connection at Run Time

You may run a service using a connection other than the default connection that was associated with the service when the service was created. To override the default, you must code your flow to pass a value through the pipeline into a service's \$connectionName field.

If you have multiple instances of a JMS provider running, you can use the dynamic connections feature to override the default and switch between the instances.

Both connections—the default and override—must use the same JMS destination type (topic or queue).

For more information, see [“Changing the Connection for an Adapter Service at Run Time” on page 17](#).

Viewing Adapter Connection Parameters

You can view a connection's parameters from Integration Server Administrator or Designer.

Using Integration Server Administrator to View Adapter Connection Parameters

Use the following procedure to view adapter connection parameters using Integration Server Administrator.


➤ **To view the parameters for a connection using Integration Server Administrator**

1. In the **Adapters** menu in the Integration Server Administrator's navigation area, click **Adapter for JMS**.

You can sort and filter the list of connections that appears on the Connections screen.

- To sort information on the Connections screen, click the **Up** and **Down** arrows.
- To filter the list of connections:
 1. On the Connections screen, click **Filter Connections**.
 2. Type the criterion by which you want to filter into the **Filter criteria** box. Filtering is based on the node name, not the connection alias. To locate all connections containing specific alphanumeric characters, use asterisks (*) as wildcards. For example, if you want to display all connections containing the string "abc", type *abc* in the **Filter criteria** box.
 3. Click **Submit**. The Connections screen displays the connections that match the filter criteria.
 4. To re-display all connections, click **Show All Connections**.

The Connections screen appears, listing all the current connections. You can control the number of connections that are displayed on this screen. For more information, see ["Controlling Pagination" on page 21](#).

2. On the Connections screen, click the  icon for the connection you want to see.

The View Connection screen displays the parameters for the connection. For descriptions of the connection parameters, see ["Configuring Adapter for JMS Connections" on page 38](#).

3. Click **Return to Adapter for JMS Connections** to return to the main connections screen.

Using Designer to View Adapter Connection Parameters

Use the following procedure to view adapter connection parameters using Designer.

➤ To view the parameters for a connection using Designer


1. From the Designer navigation area, open the package and folder in which the connection is located.
2. Double-click the connection you want to view.

The parameters for the connection appear in the **Connection Information** tab. For descriptions of the connection parameters, see the table of parameters in ["Configuring Adapter for JMS Connections" on page 38](#).

Editing Adapter Connections

If the login information for a server changes, or if you want to redefine parameters that a connection uses when connecting to a server, you can update a connection's parameters. You edit adapter connections using Integration Server Administrator.

> To edit a connection

1. Make sure that the connection is disabled before editing it. See [“Disabling Adapter Connections” on page 46](#) for instructions.
2. In the **Adapters** menu in the Integration Server Administrator navigation area, click **Adapter for JMS**.
3. On the Connections screen, click the  icon for the connection you want to edit.

The Edit Connection screen displays the current parameters for the connection. Update the connection's parameters by typing or selecting the values you want to specify.


For descriptions of the connection parameters, see [“Configuring Adapter for JMS Connections” on page 38](#).

4. Click **Save Changes** to save the connection and return to the Connections screen.
5. Enable the connection you edited. For instructions, see [“Enabling Adapter Connections” on page 45](#).

Copying Adapter Connections

You can copy an existing Adapter for JMS connection to configure a new connection with the same or similar connection properties without having to re-type all of the properties for the connection. You copy adapter connections using Integration Server Administrator.

> To copy a connection

1. In the **Adapters** menu in the Integration Server Administrator navigation area, click **Adapter for JMS**.
2. On the Connections screen, click the  icon for the connection you want to copy.

The Copy Connection screen displays the current parameters for the connection you want to copy. Name the new connection, specify a Package Name and Folder Name, and edit any connection parameters as needed by typing or selecting the values you want to specify.

Note:

When you copy a connection, the new connection does not save the password of the original connection. You must enter and then retype the password before you can save the new connection.

For descriptions of the connection parameters, see [“Configuring Adapter for JMS Connections” on page 38](#).

3. Click **Save Connection** to save the connection and return to the Connections screen.

Deleting Adapter Connections

If you no longer want to use a particular Adapter for JMS connection, you can delete it by following the instructions in this section. You delete adapter connections using Integration Server Administrator.

If you delete an Adapter for JMS connection, the adapter services that are defined to use the connection will no longer work. However, because you can change which connection an adapter service uses, if you delete an Adapter for JMS connection, you can assign a different connection to an adapter service and re-use the service. To do this, use the built-in Integration Server function `setAdapterServiceNodeConnection`. For more information, see [“Changing the Connection for an Adapter Service or Notification at Design Time” on page 16](#).

➤ To delete a connection

1. Make sure that the connection is disabled before deleting. To disable the connection, click **Yes** in the **Enabled** column and click **OK** to confirm. The **Enabled** column now shows **No (Disabled)** for the connection.
2. In the **Adapters** menu in the Integration Server Administrator navigation area, click **Adapter for JMS**.
3. On the Connections screen, click the **X** icon for the connection you want to delete.

Integration Server deletes the adapter connection.

Enabling Adapter Connections

An Adapter for JMS connection must be enabled before you can configure any adapter service or notification using the connection, or before an adapter service or notification can use the connection at run time. You enable adapter connections using Integration Server Administrator.

Note:

When you reload a package that contains enabled connections, the connections will automatically be enabled when the package reloads. If the package contains connections that are disabled, they will remain disabled when the package reloads.

➤ To enable a connection

1. In the **Adapters** menu in the Integration Server Administrator navigation area, click **Adapter for JMS**.
2. On the Connections screen, click **No** in the **Enabled** column for the connection you want to enable.

Integration Server Administrator enables the adapter connection and displays a ✓ and **Yes** in the **Enabled** column.

Disabling Adapter Connections

Adapter for JMS connections must be disabled before you can edit or delete them. You disable adapter connections using Integration Server Administrator.

Note:

When you disable an adapter connection, it stops (but does not disable) any notifications associated with the connection. However, if you disable a connection that is used by a notification that has “Run as True Listener” enabled (see [“Configuring MessageListener Notifications” on page 69](#)), the notification continues to function as usual, and runs whenever a message arrives in the JMS destination.

> To disable a connection

1. In the **Adapters** menu in the Integration Server Administrator navigation area, click **Adapter for JMS**.
2. On the Connections screen, click **Yes** in the **Enabled** column for the connection you want to disable.

The adapter connection becomes disabled and you see a **No** in the **Enabled** column.

3. On the Connections screen, click **No** in the **Enabled** column for the connection you want to enable. Integration Server Administrator enables the adapter connection displays a ✓ and **Yes** in the **Enabled** column.

5 Adapter Services

■ Overview	48
■ Before Configuring or Managing Adapter Services	48
■ Configuring MessageProducer Services	48
■ Configuring MessageConsumer Services	50
■ Configuring MessageRequestor Services	53
■ Configuring Document-to-Message Conversion Services	55
■ Configuring Message-to-Document Conversion Services	59
■ Viewing Adapter Services	63
■ Editing Adapter Services	64
■ Deleting Adapter Services	64

Overview

This chapter describes how to configure and manage Adapter for JMS services. Adapter for JMS services allow clients to interact with JMS providers using a configured adapter connection. For detailed descriptions of the available Adapter for JMS services, see [“Adapter Services” on page 15](#).

Before Configuring or Managing Adapter Services

Perform the following tasks before configuring or managing adapter services.

➤ To prepare to configure or manage Adapter for JMS services

1. Start your Integration Server and Integration Server Administrator, if they are not already running.
2. Make sure you have webMethods Integration Server administrator privileges so that you can access Integration Server Administrator's administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* for your release.
3. Using Integration Server Administrator, make sure the WmJMSAdapter package is enabled. See [“Enabling Packages” on page 31](#) for instructions.
4. Using Integration Server Administrator, configure the adapter connection you plan to use with the adapter service. For instructions, see [“Configuring Adapter for JMS Connections” on page 38](#).
5. Start Software AG Designer if it is not already running.
6. Using Designer, create a user-defined package to contain the service, if you have not already done so. When you configure adapter services, you should always define them in user-defined packages rather than in the WmJMSAdapter package. For more information about managing packages for the adapter, see [“Package Management” on page 29](#).

Configuring MessageProducer Services

The MessageProducer service enables you to send messages to a JMS Destination (queue/topic). You configure MessageProducer services using a specific adapter connection. After you follow the steps in this section to configure the MessageProducer service, you can invoke it from a flow or Java service. You use the MessageProducer service in combination with other Adapter for JMS services that enable you to create JMS messages to add to the service's input signature.

To use MessageProducer services, you provide values for the service's input signature in the pipeline. For more information about using MessageProducer services, see [“Using MessageProducer Services” on page 49](#).

You can use the same configured MessageProducer service to send messages to different JMS Destinations, provided that the JMS Destinations are all available using the same adapter connection.

➤ To configure a MessageProducer service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 48.
2. Start Designer.
3. Right-click the package in which the service should be contained and select **New > Adapter Service**.
4. Select the parent namespace and type a name for the adapter service. Click **Next**.
5. Select **Adapter for JMS** as the adapter type and click **Next**.
6. Select the appropriate **Adapter Connection Name** and click **Next**.
7. From the list of available templates, select the **MessageProducer** template and click **Finish**.

The service is created and its parameters and controls are displayed in the adapter service editor. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as the **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template**, as necessary.

8. On the **Send Message** tab, the **Destination Type** field will be populated with either Topic or Queue. This value is determined by the service's associated adapter connection.
9. The **Input/Output** tab shows the service's input and output signatures. To use the MessageProducer service, you provide values for the signature input within the service pipeline in Designer. See [“Using MessageProducer Services”](#) on page 49 for instructions on using the MessageProducer service.

For information about configuring the **Input/Output**, **Settings**, **Audit**, **Permissions**, and **Results** tabs, see the *webMethods Service Development Help* for your release. These tabs apply to all services that you configure using Designer.

10. Select **File > Save**.

Using MessageProducer Services

To enable the MessageProducer service to send a JMS Message to a Queue or a Topic, provide values for the parameters listed below in the service's input signature when configuring the adapter service within a flow service in Designer. You can either map in values from the pipeline, or you can set constant values using the pipeline editor.

Parameter	Description/Action
JMSMessage	The JMS message object that you want to send to the JMS Destination.

Parameter	Description/Action
	<p>Create the JMS Message object using one of the following Adapter for JMS services:</p> <ul style="list-style-type: none"> ■ DocumentToTextMessage ■ DocumentToObjectMessage ■ DocumentToMapMessage ■ DocumentToBytesMessage ■ DocumentToStreamMessage <p>See “Configuring Document-to-Message Conversion Services” on page 55 for more information.</p>
JMSDestination	<p>The JMS Destination object or the JMS Destination JNDI Name (String).</p> <p>If the adapter connection is configured for a queue, the JMS destination must be a queue.</p> <p>If the adapter connection is configured for a topic, the JMS Destination must be a topic.</p>
JMSDeliveryMode	<p>(Optional) The Delivery Mode to use (PERSISTENT or NON_PERSISTENT). You can provide String literals 'PERSISTENT' or 'NON_PERSISTENT' or an Integer value. Default: 'PERSISTENT'</p>
JMSPriority	<p>(Optional) The message's priority 0-9. Default: 4</p>
JMSTimeToLive	<p>(Optional) The message's lifetime (in milliseconds). Default: 0 (forever)</p>

Configuring MessageConsumer Services

The MessageConsumer service enables you to synchronously receive messages from a JMS Destination (Queue/Topic). You configure MessageConsumer services based on specific adapter connections. After you follow the steps in this section to configure the MessageConsumer service, you can invoke it from a flow or Java service. You use the MessageConsumer service in combination with other Adapter for JMS services that enable you to convert JMS messages into documents that you can use within other services in Designer.

To use MessageConsumer services, you provide values for the service's input signature in the pipeline. For more information about using MessageConsumer services, see [“Using MessageConsumer Services” on page 51](#).

You can use the same configured MessageConsumer service to retrieve messages from multiple JMS Destinations, provided that the JMS Destinations are all available using the same adapter connection.

➤ [To configure a MessageConsumer service](#)

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 48.
2. Start Designer.
3. Right-click the package in which the service should be contained and select **New > Adapter Service**.
4. Select the parent namespace and type a name for the adapter service. Click **Next**.
5. Select **Adapter for JMS** as the adapter type and click **Next**.
6. Select the appropriate **Adapter Connection Name** and click **Next**.
7. From the list of available templates, select the **MessageConsumer** template and click **Finish**.

The service is created and its parameters and controls are displayed in the adapter service editor. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as the **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template**, as necessary.

8. On the **Receive Message** tab, the **Destination Type** field will be populated with either Topic or Queue. This value is determined by the service's associated adapter connection.
9. The **Input/Output** tab shows the service's input and output signatures. To use the MessageConsumer service, you provide values for the signature input within the service pipeline in Designer. See [“Using MessageConsumer Services”](#) on page 51 for instructions on using the MessageConsumer service.
10. For information about configuring the **Input/Output**, **Settings**, **Audit**, **Permissions**, and **Results** tabs, see the *webMethods Service Development Help* for your release.
11. Select **File > Save**.

Using MessageConsumer Services

To enable the MessageConsumer service to retrieve a JMS Message from a Queue or Topic, provide values for the parameters listed below in the service's input signature when configuring the adapter service within a flow service in Designer. You can either map in values from the pipeline, or you can set constant values using the pipeline editor.

Parameter	Description/Action
JMSDestination	The JMS Destination object or the JMS Destination JNDI Name (String).

Parameter	Description/Action
	<p>If the adapter connection is configured for a queue, the JMS destination must be a queue.</p> <p>If the adapter connection is configured for a topic, the JMS Destination must be a topic.</p>
JMSMessageSelector	<p>(Optional) A String contain a SQL92 conditional expression.</p> <p>Only messages with JMS Properties or JMS Headers that match the expression will be delivered.</p> <p>A value of null or an empty string indicates that there is no message selector for the service.</p>
JMSTimeout	<p>The maximum length of time the service waits on the JMS Destination if a message does not arrive.</p>
Topic.isDurableSubscriber	<p>(Optional) This parameter is for use with topics only.</p> <p>Set this parameter to true if the subscriber to the topic is a durable subscriber. Otherwise, set the field to false. Default: false</p>
Topic.JMSSubscriptionName	<p>(Optional) This parameter is for use with topics only.</p> <p>If the subscriber to the topic is a durable subscriber, set the field to the subscription name to use.</p> <div style="background-color: #f0f0f0; padding: 5px;"><p>Note: Do not change this field after you have saved the service because doing so could potentially cause orphaned durable subscriptions on the JMS destination.</p></div>

The MessageConsumer service contains the following output signature:

Parameter	Description/Action
JMSMessage	<p>The JMS message object that is retrieved from the JMS Destination.</p> <p>Convert the document to a JMS message object using one of the following Adapter for JMS services:</p> <ul style="list-style-type: none">■ TextMessageToDocument■ ObjectMessageToDocument■ MapMessageToDocument■ BytesMessageToDocument■ StreamMessageToDocument

Parameter	Description/Action
	See “Configuring Message-to-Document Conversion Services” on page 59 for more information.
JMSMessageClass	A String containing the name of the Java class for the JMS message. You can use the value from this parameter to determine which message-to-document service to use to convert the JMS message to a document.

Configuring MessageRequestor Services

MessageRequestor service will allow you to Synchronously do Request/Reply from a JMS Destination (Queue/Topic). You configure MessageRequestor services based on specific adapter connections. After you follow the steps in this section to configure the MessageRequestor service, you can invoke it from a flow or Java service. You use the MessageRequestor service in combination with other Adapter for JMS services that enable you to create JMS messages to add to the service's input signature and to convert JMS messages into documents that you can use within other services in Designer.

To use MessageRequestor services, you provide values for the service's input signature in the pipeline. For more information about using MessageRequestor services, see [“Using MessageRequestor Services” on page 54](#).

You can use the same configured MessageRequestor service to retrieve messages from multiple JMS Destinations, provided that the JMS Destinations are all available using the same adapter connection.

➤ To configure a MessageRequestor service

1. Review the steps in [“Before Configuring or Managing Adapter Services” on page 48](#).
2. Start Designer.
3. Right-click the package in which the service should be contained and select **New > Adapter Service**.
4. Select the parent namespace and type a name for the adapter service. Click **Next**.
5. Select **Adapter for JMS** as the adapter type and click **Next**.
6. Select the appropriate **Adapter Connection Name** and click **Next**.
7. From the list of available templates, select the **MessageRequestor** template and click **Finish**.

The service is created and its parameters and controls are displayed in the adapter service editor. You can select the **Adapter Settings** tab at any time to confirm adapter service properties

such as the **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template**, as necessary.

8. On the **Send Message** tab, the **Destination Type** field will be populated with either Topic or Queue. This value is determined by the service's associated adapter connection.
9. The **Input/Output** tab shows the service's input and output signatures. To use the MessageRequestor service, you provide values for the signature input within the service pipeline in Designer. For instructions on using the MessageRequestor service, see [“Using MessageRequestor Services” on page 54](#).
10. For information about configuring the **Input/Output**, **Settings**, **Audit**, **Permissions**, and **Results** tabs, see the *webMethods Service Development Help* for your release.
11. Select **File > Save**.

Using MessageRequestor Services

To enable the MessageRequestor service to send a JMS Message to a queue or topic and then wait on a temporary destination for a reply, provide values for the parameters listed below in the service's input signature when configuring the adapter service within a flow service in Designer. You can either map in values from the pipeline, or you can set constant values using the pipeline editor.

Parameter	Description/Action
JMSMessage	<p>The JMS message object that you want to send to the JMS Destination.</p> <p>Create the JMS Message object using one of the following Adapter for JMS services:</p> <ul style="list-style-type: none">■ DocumentToTextMessage■ DocumentToObjectMessage■ DocumentToMapMessage■ DocumentToBytesMessage■ DocumentToStreamMessage <p>See “Configuring Document-to-Message Conversion Services” on page 55 for more information.</p>
JMSDestination	<p>The JMS Destination object or the JMS Destination JNDI Name (String).</p> <p>If the adapter connection is configured for a queue, the JMS destination must be a queue.</p>

Parameter	Description/Action
	If the adapter connection is configured for a topic, the JMS Destination must be a topic.

The MessageRequestor service contains the following output signature:

Parameter	Description/Action
JMSMessage	<p>The JMS message object that is retrieved from the JMS Destination.</p> <p>Convert the document to a JMS message object using one of the following Adapter for JMS services:</p> <ul style="list-style-type: none"> ■ TextMessageToDocument ■ ObjectMessageToDocument ■ MapMessageToDocument ■ BytesMessageToDocument ■ StreamMessageToDocument <p>See “Configuring Message-to-Document Conversion Services” on page 59 for more information.</p>
JMSMessageClass	A String containing the name of the Java class for the JMS message. You can use the value from this parameter to determine which message-to-document service to use to convert the JMS message to a document.

Configuring Document-to-Message Conversion Services

The Adapter for JMS provides services that enable you to convert Integration Server documents into JMS Messages. You use these services in combination with the MessageProducer and MessageRequestor adapter services.

➤ To configure a document-to-message conversion service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 48.
2. Start Designer.





Note:

Make sure the server with which you want to use Designer is running.

3. Right-click the package in which the service should be contained and select **New > Adapter Service**.

4. Select the parent namespace and type a name for the adapter service. Click **Next**.
5. Select **Adapter for JMS** as the adapter type and click **Next**.
6. Select the appropriate **Adapter Connection Name** and click **Next**.
7. Select one of the following templates, depending on the type of message to which you want to convert:
 - DocumentToTextMessage
 - DocumentToObjectMessage
 - DocumentToMapMessage
 - DocumentToBytesMessage
 - DocumentToStreamMessage
8. Click **Finish**.





The service is created and its parameters and controls are displayed in the adapter service editor. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as the **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template**, as necessary.

9. Select the service's message tab to confirm or define the message body name, as follows:
 - For the DocumentToTextMessage service template, select the **Text Message** tab to confirm that the **Body Field Name** is called TextMessageBody. Do not change the name of this field. This field appears in the service's input signature. You map the message data to this parameter.
 - For the DocumentToObjectMessage service template, select the **Object Message** tab to confirm that the **Body Field Name** is called ObjectMessageBody. Do not change the name of this field. This field appears in the service's input signature. You map the message object to this parameter.
 - For the DocumentToMapMessage service template, select the **Map Message** tab to define the message body.
 - To define new field names, select the  icon.
 - To change the order of the fields, use the  or  icons.
 - To delete any field names, use the  icon.

Provide values for the rows as follows:

Parameter	Description/Action
Body Field Name	The name of the JMS message body field.
Body Field Type	The data type of the JMS Message body field. Select the data type from the list of available data types.
InputFieldType	The Java data type of Body Field Name . Select the data type from the list of available data types.
InputField Name	The name of the input field.





10. Select the **JMS Header** tab to configure the JMS Message Header fields that you want the JMS message to contain. Refer to the JMS specification for the JMS Header fields that can be set by a producer. If you are not using JMS Header fields, skip this step.

- To define new Message Header parameters, select the  icon.
- To change the order of the fields, use the  or  icons.
- To delete any field names, use the  icon.

Provide values for the rows as follows:

Parameter	Description/Action
Header Field Name	The name of the JMS message header field.
Header Field Type	The data type of the JMS Message header field. Select the data type from the list of available data types.
Input Field Type	The Java type of the Input Field Name . Select the data type from the list of available data types.
Input Field Name	The name of the input field.

11. Select the **JMS Properties** tab to configure the JMS message property fields that you want the JMS message to contain. If you are not using JMS property fields, skip this step.

- To define new JMS message property fields, select the  icon.
- To change the order of the fields, use the  or  icons.
- To delete any field names, use the  icon.

Provide values for the rows as follows:

Parameter	Description/Action
Property Field Name	The name of the JMS message property field.

Parameter	Description/Action
Property Field Type	The data type of the JMS message property field. Select the data type from the list of available data types.
Input Field Type	The Java data type of the Input Field Name . Select the data type from the list of available data types.
Input Field Name	The name of the input property field.

12. For information about configuring the **Input/Output, Settings, Audit, Permissions, and Results** tabs, see the *webMethods Service Development Help* for your release.

13. Select **File > Save**.

Using Document-to-Message Conversion Services

The document-to-message conversion services contain the following input signatures.

Provide values for the parameters listed below in the service's input signature when configuring the adapter service within a flow service in Designer. You can either map in values from the pipeline, or you can set constant values using the pipeline editor.

Parameter	Description/Action
Message Body	<ul style="list-style-type: none">■ For DocumentToTextMessage, the message body will be a String called TextMessageBody. Map the message data to this field.■ For DocumentToObjectMessage, the message body will be an Object called ObjectMessageBody. Map the message data object to this field.■ For DocumentToMapMessage, the message body will be a document called MapMessageBody that contains a set of parameters corresponding to the message body fields you added in step 8 in “Configuring Document-to-Message Conversion Services” on page 55.■ For DocumentToBytesMessage, the message body will be an Object called BytesMessageBody. Map the message data object to this field.■ For DocumentToStreamMessage, the message body will be a document called ObjectMessageBody that contains a set of parameters corresponding to the message body fields you added in step 8 in “Configuring Document-to-Message Conversion Services” on page 55.
JMSHeader	This document will contain any JMS Header fields you added in step 9 in “Configuring Document-to-Message Conversion Services” on page 55 .

Parameter	Description/Action
Properties	This document will contain any JMS property fields you added in step 10 in “Configuring Document-to-Message Conversion Services” on page 55.

The document-to-message conversion services contain the following output signature:

Parameter	Description/Action
JMSMessageClass	A String containing the name of the Java class for the JMS message.
JMSMessage	The JMS message object to be sent to a JMS Destination. This object can be passed to a MessageProducer or MessageRequestor service.

Configuring Message-to-Document Conversion Services

The Adapter for JMS provides services that enable you to convert JMS messages into Integration Server documents. You use these services in combination with the MessageConsumer and MessageRequestor adapter services.

» To configure a message-to-document conversion service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 48.
2. Start Designer.

Note:

Make sure the server with which you want to use Designer is running.





3. Right-click the package in which the service should be contained and select **New > Adapter Service**.
4. Select the parent namespace and type a name for the adapter service. Click **Next**.
5. Select **Adapter for JMS** as the adapter type and click **Next**.
6. Select the appropriate **Adapter Connection Name** and click **Next**.
7. Select one of the following templates, depending on the type of message to which you want to convert:
 - TextMessageToDocument
 - ObjectMessageToDocument
 - MapMessageToDocument

- BytesMessageToDocument
- StreamMessageToDocument

8. Click **Finish**.





The service is created and its parameters and controls are displayed in the adapter service editor. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as the adapter name, adapter connection name, and adapter service template, as necessary.

9. Select the service's message tab to confirm or define the message body field name, as follows:

- For the TextMessageToDocument service template, select the **Text Message** tab to confirm that the **Body Field Name** is called TextMessageBody. Do not change the name of this field. This field contains the message data. It appears in the service's output signature.
- For the ObjectMessageToDocument service template, select the **Object Message** tab to confirm that the **Body Field Name** is called ObjectMessageBody. Do not change the name of this field. This field contains the message object. It appears in the service's output signature.
- For the MapMessageToDocument service template, select the **Map Message** tab to define the message body.
 - To define new field names, select the  icon.
 - To change the order of the fields, use the  or  icons.
 - To delete any field names, use the  icon.





Provide values for the rows as follows:

Parameter	Description/Action
Body Field Name	The name of the JMS message body field.
Body Field Type	The data type of the JMS message body field. Select the data type from the list of available data types.
Output Field Type	The Java data type of Output Field Name . Select the data type from the list of available data types.
Output Field Name	The name of the output field.

- For the DocumentToBytesMessage service template, select the **Bytes Message** tab to define the message body.
 - To define new field names, select the  icon.
 - To change the order of the fields, use the  or  icons.
 - To delete any field names, use the  icon.

Provide values for the rows as follows:





Parameter	Description/Action
Body Field Name	The name of the JMS message body field.
Body Field Type	The data type of the JMS message body field. Select the data type from the list of available data types.
Output Field Type	The Java data type of Output Field Name . Select the data type from the list of available data types.
Output Field Name	The name of the input field.

- For the DocumentToStreamMessage service template, select the **Stream Message** tab to define the message body.
 - To define new field names, select the  icon.
 - To change the order of the fields, use the  or  icons.
 - To delete any field names, use the  icon.

Provide values for the rows as follows:

Parameter	Description/Action
Body Field Name	The name of the JMS message body field.
Body Field Type	The data type of the JMS message body field. Select the data type from the list of available data types.
Output Field Type	The Java data type of Body Field Name . Select the data type from the list of available data types.
Output Field Name	The name of the output field.

10. Select the **JMS Header** tab to configure the JMS Message Header fields that you want the service to expose. If you are not using JMS Header fields, skip this step.





- To define new Message Header parameters, select the  icon.
- To change the order of the fields, use the  or  icons.
- To delete any field names, use the  icon.

Provide values for the rows as follows:

Parameter	Description/Action
Header Field Name	The name of the JMS Message header field.

Parameter	Description/Action
Header Field Type	The data type of the JMS Message header field. Select the data type from the list of available data types.
Output Field Type	The Java type of the Output Field Name . Select the data type from the list of available data types.
Output Field Name	The name of the output field.

11. Select the **JMS Properties** tab to configure the JMS message property fields that you want the service to expose. If you are not using JMS property fields, skip this step.

- To define new JMS message property fields, select the  icon.
- To change the order of the fields, use the  or  icons.
- To delete any field names, use the  icon.

Provide values for the rows as follows:

Parameter	Description/Action
Property Field Name	The name of the JMS message property field.
Property Field Type	The data type of the JMS message property field. Select the data type from the list of available data types.
Output Field Type	The Java data type of the Output Field Name . Select the data type from the list of available data types.
Output Field Name	The name of the output property field.

12. For information about configuring the **Input/Output, Settings, Audit, Permissions, and Results** tabs, see the *webMethods Service Development Help* for your release. These tabs apply to all services that you configure using Designer.

13. Select **File > Save**.

Using Message-to-Document Conversion Services

The message-to-document conversion services contain the following input signatures.

Provide values for the parameters listed below in the service's input signature when configuring the adapter service within a flow service in Designer. You can either map in values from the pipeline, or you can set constant values using the pipeline editor.

Parameter	Description/Action
JMSMessage	The JMS message object to convert to a document.

The message-to-document conversion services contain the following output signature:

Parameter	Description/Action
JMSMessageClass	A String containing the name of the Java class for the JMS message.
Message Body	<ul style="list-style-type: none"> ■ For <code>TextMessageToDocument</code>, the message body will be a String called TextMessageBody. ■ For <code>ObjectMessageToDocument</code>, the message body will be an Object called ObjectMessageBody. ■ For <code>MapMessageToDocument</code>, the message body will be a document called MapMessageBody that contains a set of parameters corresponding to the message body fields you added in step 8. ■ For <code>BytesMessageToDocument</code>, the message body will be an Object called BytesMessageBody. ■ For <code>StreamMessageToDocument</code>, the message body will be a document called ObjectMessageBody that contains a set of parameters corresponding to the message body fields you added in step 8 in “Configuring Message-to-Document Conversion Services” on page 59.
JMSHeader	This document will contain any JMS Header fields you added in step 9 in “Configuring Message-to-Document Conversion Services” on page 59 .
Properties	This document will contain any JMS property fields you added in step 10 in “Configuring Message-to-Document Conversion Services” on page 59 .

Viewing Adapter Services

You use Designer to view adapter services.

➤ To view an adapter service

1. Review the steps in [“Before Configuring or Managing Adapter Services” on page 48](#).
2. In Designer, expand the package and folder that contain the service you want to view.
3. Double click the service you want to view.

Designer displays the configured service in the service template's Adapter Service Editor.

Editing Adapter Services

You use Designer to edit adapter services.

➤ To edit an adapter service

1. In Designer, browse to and open the adapter service that you want to edit.
2. Double-click the service that you want to edit.

Designer displays the adapter service in the service template's Adapter Service Editor.

3. Do one of the following:
 - If you have the VCS Integration feature enabled, right-click the service and select **Check Out**.
 - If you do not have the VCS Integration feature enabled, right-click the service and select **Lock for Edit**.
 - If you are using the local service development feature, from the **Team** menu in Designer, select the appropriate option to check out the service. The options available in the **Team** menu depend on the VCS client that you use.
4. Modify the values for the adapter service's parameters as needed. For detailed descriptions of the service's parameters, see the section on configuring a service for the specific type of service you want to edit.
5. After you complete your modifications, save the service and do one of the following:
 - If you have the VCS Integration feature enabled, right-click the service and select **Check In**. Enter a check-in comment and click **OK**.
 - If you do not have the VCS Integration feature enabled, right-click the service and select **Unlock**.
 - If you are using the local service development feature, from the **Team** menu in Designer, select the appropriate option to check in the service. The options available in the **Team** menu depend on the VCS client that you use.
6. Save the service.

Deleting Adapter Services

You use Designer to delete adapter services.

➤ To delete an adapter service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 48.
2. In Designer, expand the package and folder that contain the service you want to delete.
3. Right-click the service and click **Delete**.

6 Adapter Notifications

■ Overview	68
■ Before Configuring or Managing Notifications	68
■ Configuring MessageListener Notifications	69
■ Managing MessageListener Notifications	72
■ Using the Exactly Once Notification Feature	74
■ Viewing Notifications	74
■ Editing Notifications	74
■ Deleting Notifications	75

Overview

MessageListener notifications monitor a JMS destination and notify Integration Server when a message arrives on the JMS destination. This section provides instructions for configuring and managing MessageListener notifications.

Before Configuring or Managing Notifications

Perform the following tasks before configuring or managing adapter notifications.

➤ To prepare to configure or manage MessageListener notifications

1. Start your Integration Server and Integration Server Administrator, if they are not already running.
2. Make sure you have Integration Server Administrator and Software AG Designer privileges so that you can access the Integration Server's administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* for your release.
3. If you plan to use the Only Once notification feature, see [“Using the Exactly Once Notification Feature” on page 74](#) for details.
4. Using Integration Server Administrator, make sure the WmJMSAdapter package is enabled. See [“Enabling Packages” on page 31](#) for instructions.
5. Using Integration Server Administrator, configure an adapter connection to use with the notification. See [“Configuring Adapter for JMS Connections” on page 38](#) for instructions.

Note:

Integration Server solves this limitation by providing a built-in service that you can use at design time to change the connection associated with a polling notification. For more information, see [“Changing the Connection for an Adapter Service or Notification at Design Time” on page 16](#).

6. Start Designer if it is not already running.
7. Using Designer, create a user-defined package to contain the notification, if you have not already done so. When you configure notifications, you should always define them in user-defined packages rather than in the WmJMSAdapter package. For more information about managing packages for the adapter, see [“Package Management” on page 29](#).
8. You must schedule a notification and then enable it before you can use the notification. See [“Managing MessageListener Notifications” on page 72](#) for instructions.

Configuring MessageListener Notifications

When you configure MessageListener notifications, you specify information that Integration Server uses to read data from a JMS Destination and publish a notification document. For more information about notifications, see [“Adapter Notifications” on page 18](#).

➤ To configure a MessageListener notification

1. Review the steps in [“Before Configuring or Managing Notifications” on page 68](#).
2. Start Designer.
3. Right-click the package in which the notification should be contained and select **New > Adapter Notification**.
4. Select the parent namespace and type a name for the adapter notification. Click **Next**.
5. Select **Adapter for JMS** as the adapter type and click **Next**.
6. Select the **MessageNotification** template and click **Next**.
7. Select the appropriate **Adapter Connection Name** and click **Next**.
8. The name of the publishable document associated with this notification displays. Click **Finish**.

For more information about adapter notifications and publishable documents, see [“Adapter Notifications” on page 18](#). For more details about Integration Server publishable documents, see the *Publish-Subscribe Developer’s Guide* for your release.

9. Designer creates the notification, and the editor for the adapter notification appears. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as the **Adapter Name**, **Adapter Connection Name**, and **Adapter Notification Template**, as necessary.
10. Select the **Notification** tab and provide values for the following fields:

Parameter	Description
Destination Type	This property is populated based on the Adapter for JMS connection associated with this notification. For connections associated with queues, the value will be QUEUE. For connections associated with topics, the value will be TOPIC.
Destination JNDI Name	The JNDI name of the queue or topic you want the notification to monitor for messages. Provide a Queue Name if the Destination

Parameter	Description
	Type is QUEUE. Provide a Topic Name if the Destination Type is TOPIC.
Message Selector	The SQL-92 compliant message selector expression used for filtering messages.
Connection Client ID	<p>This parameter is only used if the Run as True Listener option is selected. This field only applies to topics. The unique connection ID to use when connecting to a JMS provider topic. If the connection factory has a default Client ID, the notification will use the default, not the value specified in this parameter.</p> <p>Note: Do not change this field after you have saved the notification; doing so could potentially cause orphaned durable subscriptions on the JMS destination.</p> <p>Note: If you need to use a Client ID for the notifications, it is better to specify it along with the other notification parameters using Designer, and not while creating the connection factory or the Adapter for JMS connections. When the Client ID is specified in the connection factory or JMS connections, only one connection can be created through this connection factory or connection. This limits the Adapter for JMS from leveraging the connection pool.</p>
Durable Subscription Name	<p>The unique subscription name to use when creating durable subscriptions. If this field is left blank, the adapter will use the name of the notification for the value of this field.</p> <p>Note: Do not change this field after you have saved the notification because doing so could potentially cause orphaned durable subscriptions on the JMS destination.</p>
Maximum Messages	<p>This field only applies when the notification does not have the Run as True Listener option enabled. The value of this field specifies the maximum number of messages to retrieve every time the notification polls the JMS destination for messages. By default the value is -1, which indicates that the notification retrieves all available messages from the queue or topic every time it polls the JMS destination.</p> <p>Note: Do not set the maximum messages to zero because the notification will not retrieve any messages.</p>
Receive Timeout	The notification will wait for as many milliseconds as are indicated in this field. Provide a value as follows:

Parameter	Description
	<ul style="list-style-type: none"> ■ 1 or greater: The number of milliseconds the notification will wait. ■ 0: The notification will not wait if there are no messages on the queue or topic. ■ -1: The notification will wait indefinitely to receive messages. This is the default.
Run as True Listener	If this option is enabled, the notification will not poll for messages. Instead, it listens to the JMS destination and publishes the associated notification as soon as a message arrives in the JMS destination. If you disable the associated connection, this notification will continue to run (see “Disabling Adapter Connections” on page 46).
Exactly Once Notification	If this option is enabled, Integration Server ignores duplicate messages from this notification if published. <p>Note: This option applies only if you are publishing a document with the notification.</p>
Use Durable Subscription	If this option is enabled, the notification creates durable subscriptions for the topic.
notificationField Names and notificationField Types	This table shows the structure of the publishable document type associated with the notification.
Notification Service	If you want the notification to directly invoke a service rather than publishing a document to Integration Server (which you could then use with an Integration Server trigger to invoke a service), provide the name of a service in this field. If you provide a service name here, the input signature for the service must be constructed the same way as an input signature for a service that is invoked by a trigger. However, the service cannot use any of the <code>_env</code> fields in the signature. See the <i>Publish-Subscribe Developer’s Guide</i> . <p>Important: This option provides the only way to use notifications for JMS providers whose implementation of the JMS message interface <code>javax.jms.Message</code> does not implement the <code>Serializable</code> interface. This option must be used by the Web Application Server JMS providers.</p>

11. For information about using the **Permissions** tab to assign an access control list (ACL) to an element, see the *webMethods Service Development Help* for your release.

12. Select **File > Save**.
13. By default a notification is disabled when it is configured. You must schedule and enable the notification using Integration Server Administrator before you can use it. See [“Managing MessageListener Notifications” on page 72](#) for details.

Managing MessageListener Notifications

You must schedule a MessageListener notification and then enable it before you can use the notification to monitor JMS destinations for messages. Use Integration Server Administrator along with the following procedures to do so.

Note:


You cannot enable a notification if the adapter connection that the notification is using is disabled.




Note:


You must have webMethods Integration Server privileges to access the Integration Server Administrator's administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* for your release.

➤ To manage MessageListener notifications

1. Review the steps in [“Before Configuring or Managing Notifications” on page 68](#).
2. Start Integration Server Administrator.
3. From the **Adapters** menu in the navigation area of Integration Server Administrator, select **Adapter for JMS**.
4. From the navigation area, select **Polling Notifications**. All configured Adapter for JMS notifications are displayed.
5. From the **Adapter for JMS Polling Notifications** table, use the fields in the following table to manage each adapter notification:

Field	Description/Action
Notification Name	The name of the notification.
Package Name	The name of the package for the notification.
State	<p>You must schedule a polling notification before you can enable it. To schedule a polling notification, use the  icon described in these procedures. After doing so, you can use this option's dropdown list to set the polling notification's state:</p> <ul style="list-style-type: none"> ■ Enabled: The polling notification performs as scheduled.

Field	Description/Action
	<ul style="list-style-type: none"> ■ Suspended: The polling notification is removed from the scheduler. ■ Disabled: The polling notification is removed from the scheduler. <p>Note: Even if you enable the Run as True Listener option when you configure the MessageListener notification, the Suspended state is functionally identical to the Disabled state.</p> <p>The Suspend all enabled and Resume all suspended links help you change states quickly for multiple polling notifications. If there is no polling notification scheduled for a given adapter notification, control for this field is disabled. Use the  icon to create a polling notification as described in step 6.</p>
Edit Schedule	<p>Click the  icon to create or modify a schedule for a polling notification. Continue to step 6.</p> <p>Note: Before you can edit the schedule for a polling notification, you must first disable it.</p>
View Schedule	<p>To view the polling schedule for the selected polling notification, click the  icon. Click Return to Adapter for JMS Notifications to return to the main polling notification page.</p>

- To create or modify schedule parameters for the selected adapter notification, click on the  icon and provide values for the following parameters:

Field	Description/Action
Interval (seconds)	Type the polling interval time in seconds.
Overlap	Do not use this option; otherwise, an exception is generated and you could potentially receive duplicate messages from a topic.
Immediate	Select this option to start polling immediately.

- Click **Save Schedule**.
- After you create a polling notification, you can enable it. Use the **State** or **Enabled** field described in step 5 to enable a polling notification.

Using the Exactly Once Notification Feature

Adapter notifications can use the Exactly Once notification feature. This feature ensures that notification data will not be duplicated even if a failure occurs during processing. This is achieved by assigning unique IDs for each publishable document. After a processing failure, Integration Server checks for duplicate records in storage and ignores any duplicate IDs.

To use the Exactly Once feature, you must enable client-side queuing in Integration Server. For details, see the *webMethods Integration Server Administrator's Guide* for your release.

Note:

You can only use this feature if the notification is used to publish a document, not if it is associated with a service.

Viewing Notifications

You use Designer to view a notification.

➤ **To view a notification**

1. Review the steps in [“Before Configuring or Managing Notifications” on page 68](#).
2. In Designer, expand the package and folder that contain the notification you want to view.
3. Double-click the notification you want to view.

Designer displays the notification in the notification template's Adapter Notification Editor.

Editing Notifications

You use Designer to edit notifications.

You can change the connection associated with an adapter notification with the built-in service `pub.art.notification:setPollingNotificationNodeConnection`. For more information, see [“Changing the Connection for an Adapter Service or Notification at Design Time” on page 16](#).

➤ **To edit a notification**

1. Review the steps in [“Before Configuring or Managing Notifications” on page 68](#).
2. In Integration Server Administrator, enable the connection for the notification you want to edit. Refer to [“Enabling Adapter Connections” on page 45](#) for instructions on enabling connections.
3. In Designer, expand the package and folder that contain the notification you want to edit.

4. Select the notification you want to edit.

Designer displays the notification in the notification template's Adapter Notification Editor.

5. Do one of the following:

- If you have the VCS Integration feature enabled, right-click the notification and select **Check Out**.
- If you do not have the VCS Integration feature enabled, right-click the notification and select **Lock for Edit**.
- If you are using the local service development feature, from the **Team** menu in Designer, select the appropriate option to check out the notification. The options available in the **Team** menu depend on the VCS client that you use.

6. Modify the values for the notification's parameters as needed. For detailed descriptions of the notification's parameters, see the section on configuring a notification for the specific type of notification you want to edit.

Note:

When you edit a notification, all the parameters in the adapter notification are reset.

7. After you complete your modifications, save the notification and do one of the following:

- If you have the VCS Integration feature enabled, right-click the notification and select **Check In**. Enter a check-in comment and click **OK**.
- If you do not have the VCS Integration feature enabled, right-click the notification and select **Unlock**.
- If you are using the local service development feature, from the **Team** menu in Designer, select the appropriate option to check in the notification. The options available in the **Team** menu depend on the VCS client that you use.

Deleting Notifications

You use Designer to delete adapter notifications.

> To delete a notification

1. In Designer, expand the package and folder that contain the notification you want to delete.
2. Right-click the notification and then click **Delete**.

A Built-In Transaction Management Services

■ Transaction Management Overview	78
■ Built-In Transaction Management Services	80
■ pub.art.transaction:commitTransaction	80
■ pub.art.transaction:rollbackTransaction	81
■ pub.art.transaction:setTransactionTimeout	81
■ pub.art.transaction:startTransaction	82
■ Changing the Integration Server Transaction Timeout Interval	82

Transaction Management Overview

This appendix provides an overview and examples of using transactions. It describes how Integration Server supports the built-in services used to manage explicit transactions for your Adapter for JMS services in the WmART package. See “[Built-In Transaction Management Services](#)” on page 80 for descriptions of each of the specific built-in transaction management services that can be used with the WmART package.

Transactions

Integration Server considers a transaction to be one or more interactions with one or more resources that are treated as a single logical unit of work. The interactions within a transaction are either all committed or all rolled back. For example, if a transaction includes multiple database inserts, and one or more inserts fail, all inserts are rolled back.

Transaction Types

Integration Server supports the following kinds of transactions:

- A *local transaction* (LOCAL_TRANSACTION) which is a transaction to a resource's local transaction mechanism
- An *XAResource transaction* (XA_TRANSACTION) which is a transaction to a resource's XAResource transaction mechanism

For information about using Integration Server to manage XA transactions, see the *webMethods Integration Server Administrator's Guide* for your release.

Implicit and Explicit Transactions

Implicit transactions are automatically handled by the Integration Server transaction manager. When you define an explicit transaction, you define the start-on-completion boundaries of the transaction. As such, implicit and explicit transactions need to be created and managed differently.

The following sections describe implicit and explicit transactions and how to manage them.

Implicit Transactions

With implicit transactions, Integration Server automatically manages both local and XAResource transactions without requiring you to explicitly do anything. That is, Integration Server starts and completes an implicit transaction with no additional service calls required by the adapter user.

A transaction context, which the transaction manager uses to define a unit of work, starts when an adapter service is encountered in a flow execution. The connection required by the adapter service is registered with the newly created context and used by the adapter service. If another adapter service is encountered, the transaction context is searched to see if the connection is already registered. If the connection is already registered, the adapter service uses this connection. If the

connection is not registered, a new connection instance is retrieved and registered with the transaction.

Note that if the top level flow invokes another flow, adapter services in the child flow use the same transaction context.

When the top level flow completes, the transaction is completed and is either committed or rolled back, depending on the status (success or failure) of the top level flow.

A single transaction context can contain any number of XA_TRANSACTION connections but no more than one LOCAL_TRANSACTION connection. If your flow contains adapter services that use more than one LOCAL_TRANSACTION connection, you must use explicit transactions, which are described in the next section.

For information about designing and using flows, see the *webMethods Service Development Help* for your release. For more information about transaction types, see [“Transaction Management of Adapter for JMS Connections” on page 13](#).

Explicit Transactions

You use explicit transactions when you need to explicitly control the transactional units of work. To do this, you use additional services, known as built-in services, in your flow.

Note:

When using a MessageRequestor service, the requesting message that the service sends to the JMS destination cannot be rolled back if an operation fails within the transaction context. That is, the MessageRequestor service automatically commits the requesting message. If it did not, the JMS provider would never deliver the request and would therefore never receive a reply message. See the JMS Specification for more information about transaction management requirements in JMS providers.

A transaction context starts when the `pub.art.transaction.startTransaction()` service is executed. The transaction context is completed when either the `pub.art.transaction.commitTransaction()` or `pub.art.transaction.rollbackTransaction()` service is executed. As with implicit transactions, a single transaction context can contain any number of XA_TRANSACTION connections but no more than one LOCAL_TRANSACTION connection.

Note:

With explicit transactions, you must be sure to call either a `commitTransaction()` or `rollbackTransaction()` for each `startTransaction()`; otherwise you will have dangling transactions that will require you to reboot Integration Server.

A new explicit transaction context can be started within a transaction context, provided that you ensure that the transactions within the transaction context are completed in the reverse order they were started—that is, the last transaction to start should be the first transaction to complete, and so forth.

For example, consider the following is a valid construct:

```
pub.art.transaction.startTransaction()  
  pub.art.transaction.startTransaction()  
    pub.art.transaction.startTransaction()
```

```
pub.art.transaction.commitTransaction()  
pub.art.transaction.commitTransaction()  
pub.art.transaction.commitTransaction()
```

The following example shows an invalid construct:

```
pub.art.transaction.startTransaction()  
pub.art.transaction.startTransaction()  
pub.art.transaction.commitTransaction()  
pub.art.transaction.commitTransaction()
```

For information about designing and using flows, see the *webMethods Service Development Help* for your release. For more information about transaction types, see [“Transaction Management of Adapter for JMS Connections” on page 13](#).

Built-In Transaction Management Services

The following sections describe each of the built-in services you can use with the wmART package.

pub.art.transaction:commitTransaction

This service commits an explicit transaction. It must be used in conjunction with the `pub.art.transaction:startTransaction` service. If it does not have a corresponding `pub.art.transaction:startTransaction` service, your flow service will receive a run-time error.

See [“Transaction Management Overview” on page 78](#) for more information about implicit and explicit transactions.

Input Parameters

commitTransactionInput **Document** A document that contains the variable *transactionName*, described below.

transactionName **String** Used to associate a name with an explicit transaction. The *transactionName* must correspond to the *transactionName* in any `pub.art.transaction:startTransaction` or `pub.art.transaction:rollbackTransaction` services associated with the explicit transaction.

This value must be mapped from the most recent `pub.art.transaction:startTransaction` that has not previously been committed or rolled back.

Output Parameters

None.

pub.art.transaction:rollbackTransaction

This service rolls back an explicit transaction. It must be used in conjunction with the `pub.art.transaction:startTransaction` service. If it does not have a corresponding `pub.art.transaction:startTransaction` service, your flow service will receive a run-time error.

See [“Transaction Management Overview” on page 78](#) for more information about implicit and explicit transactions.

Input Parameters

rollbackTransactionInput **Document** A document that contains the variable *transactionName*, described below.

transactionName **String** Used to associate a name with an explicit transaction. The *transactionName* must correspond to the *transactionName* in any `pub.art.transaction:startTransaction` or `pub.art.transaction:commitTransaction` services associated with the explicit transaction.

This value must be mapped from the most recent `pub.art.transaction:startTransaction` that has not previously been committed or rolled back.

Output Parameters

None.

pub.art.transaction:setTransactionTimeout

This service enables you to manually set a transaction timeout interval for implicit and explicit transactions. When you use this service, you are temporarily overriding the Integration Server transaction timeout interval. See [“Changing the Integration Server Transaction Timeout Interval” on page 82](#) to change the server's default transaction timeout.

You must call this service within a flow before the start of any implicit or explicit transactions. Implicit transactions start when you call an adapter service in a flow. Explicit transactions start when you call the `pub.art.transaction:startTransaction` service.

If the execution of a transaction takes longer than the transaction timeout interval, all transacted operations are rolled back.

This service only overrides the transaction timeout interval for the flow service in which you call it.

Input Parameters

<i>timeoutSeconds</i>	Integer The number of seconds that the implicit or explicit transaction stays open before the transaction manager marks it for rollback.
-----------------------	---

Output Parameters

None.

pub.art.transaction:startTransaction

This service starts an explicit transaction. It must be used in conjunction with either a `pub.art.transaction:commitTransaction` service or `pub.art.transaction:rollbackTransaction` service. If it does not have a corresponding `pub.art.transaction:commitTransaction` service or `pub.art.transaction:rollbackTransaction` service, your flow service will receive a run-time error.

See [“Transaction Management Overview” on page 78](#) for more information about implicit and explicit transactions.

Input Parameters

<i>startTransactionInput</i>	Document A document that contains the variable <i>transactionName</i> , described below.
------------------------------	---

<i>transactionName</i>	String Specifies the name of the transaction to be started. This parameter is optional. If you leave this parameter blank, Integration Server will generate a name for you. In most implementations, it is not necessary to provide your own transaction name as input.
------------------------	--

Output Parameters

<i>startTransactionOutput</i>	Document A document that contains the variable <i>transactionName</i> , described below.
-------------------------------	---

<i>transactionName</i>	String The name of the transaction the service just started.
------------------------	---

Changing the Integration Server Transaction Timeout Interval

Integration Server default transaction timeout is no timeout (NO_TIMEOUT). To change the server's transaction timeout interval, use a text editor to modify the `server.cnf` file and add the parameter below. Note that this parameter does not exist by default in the `server.cnf` file; you must add it to the file as described below.

You must shut down Integration Server before you edit this file. After you make changes, restart the server.

Add the following parameter to the server.cnf file:

```
watt.art.tmgr.timeout=TransactionTimeout
```

where *TransactionTimeout* is the number of seconds before transaction timeout.

This transaction timeout parameter does not halt the execution of a flow; it is the maximum number of seconds that a transaction can remain open and still be considered valid. For example, if your current transaction has a timeout value of 60 seconds and your flow takes 120 seconds to complete, the transaction manager will rollback all registered operations regardless of the execution status.

For more information about adding parameters to the server.cnf file, see the *webMethods Integration Server Administrator's Guide* for your release.

B WebLogic Server Configuration Notes

■ WebLogic Server Authentication	86
--	----

WebLogic Server Authentication

To ensure proper authentication when Adapter for JMS is connected to multiple WebLogic servers, you must configure:

1. the `watt.jmsadapter.weblogic` parameter.
2. JAAS authentication.

If you do not configure the parameter and JAAS authentication, the adapter throws an exception when connected to multiple WebLogic servers using different security principles and credentials.

Configuring the `watt.jmsadapter.weblogic` Parameter

To authenticate successfully with multiple WebLogic servers, you must set the `watt.jmsadapter.weblogic` parameter.

> To set the `watt.jmsadapter.weblogic` parameter

1. In Integration Server Administrator, click **Settings > Extended > Edit Extended Settings**.
2. To authenticate with multiple WebLogic servers, in the Extended Settings editor, specify the following:

```
watt.jmsadapter.weblogic = true
```

The default value is `false`.

3. Click **Save Changes**. A key for the new property appears in the Extended Settings list.
4. Click **Show and Hide Keys**. You will see the `watt.jmsadapter.weblogic` parameter listed as a Key and as visible.
5. Restart Integration Server.

Configuring JAAS Authentication

You use JAAS authentication with multiple WebLogic servers.

> To configure JAAS authentication

1. Create a file, named `WmJMSAdapter_jaas.config` with the following content:

```
/** Login Configuration for the JAAS Sample Application */
WmJMSAdapter
{
    weblogic.security.auth.login.UsernamePasswordLoginModule
    required debug=true;
```

```
};
```

2. Save the file to the `${JAVA_DIR}/jre/lib/security` directory, where `${JAVA_DIR}` is set in the `server.bat` file (`server.sh` in Unix) located in the *Integration Server_directory* \bin directory.
3. Append the following line to the `${JAVA_DIR}/jre/lib/security/java.security` package:

```
login.config.url.1=  
file:${java.home}/lib/security/WmJMSAdapter_jaas.config
```

