

webMethods MSMQ Adapter Installation and User's Guide

Version 6.0

July 2012

This document applies to webMethods MSMQ Adapter 6.0 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 © 2001-2022 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-MMQ-IUG-60-20221202

Table of Contents

About this Guide	5
Document Conventions.....	6
Online Information and Support.....	7
Data Protection.....	8
1 Overview of MSMQ Adapter	9
About MSMQ Adapter.....	10
Overview of Queuing Concepts.....	10
Architecture and Components.....	12
Package Management.....	14
Adapter Connections.....	15
Adapter Services.....	17
Adapter Listeners and Listener Notifications.....	22
Using Version Control Systems to Manage Adapter Elements.....	27
Infrastructure Data Collector Support for the MSMQ Adapter.....	28
Viewing the Adapter's Update Level.....	28
Controlling Pagination.....	28
2 Installing, Upgrading, and Uninstalling the MSMQ Adapter	29
Overview.....	30
Requirements.....	30
Installing MSMQ Adapter 6.0.....	30
Upgrading to MSMQ Adapter 6.0.....	31
Uninstalling MSMQ Adapter 6.0.....	31
3 Package Management	33
Overview.....	34
MSMQ Adapter Package Management.....	34
Group Access Control.....	37
MSMQ Adapter in a Clustered Environment.....	38
4 Adapter Connections	43
Overview.....	44
Before Configuring or Managing Adapter Connections.....	44
Configuring Adapter Connections.....	44
Dynamically Changing a Service's Connection at Run Time.....	48
Viewing Adapter Connection Parameters.....	49
Editing Adapter Connections.....	50
Copying Adapter Connections.....	51
Deleting Adapter Connections.....	51
Enabling Adapter Connections.....	52
Disabling Adapter Connections.....	52

5 Adapter Services	53
Overview.....	54
Before Configuring or Managing Adapter Services.....	54
Using Adapter Services.....	54
Configuring Put Services.....	55
Configuring Get Services.....	59
Configuring Peek Services.....	64
Using the Pipeline Editor to View and Set the Body.....	69
Testing Adapter Services.....	69
Viewing Adapter Services.....	70
Editing Adapter Services.....	70
Deleting Adapter Services.....	71
Enabling Automatic Data Validation.....	71
Reloading Adapter Values.....	72
6 Adapter Notifications	75
Overview.....	76
Listeners.....	76
Listener Notifications.....	84
7 Predefined Health Indicator	103
Predefined Health Indicator.....	104
8 Administrator APIs	105
Administrator APIs.....	106
9 Configuration Variables Templates for Adapter Assets in Microservices Runtime	107
Configuration Variables Templates for Adapter Assets in Microservices Runtime.....	108
10 Logging and Exception Handling	109
Overview.....	110
MSMQ Adapter Logging Levels.....	110
MSMQ Adapter Message Logging.....	111
MSMQ Adapter Exception Handling.....	112
MSMQ Adapter Error Codes.....	112
A Built-In Transaction Management Services	125
Transaction Management Overview.....	126
Built-In Transaction Management Services.....	128
Changing the Integration Server Transaction Timeout Interval.....	130
Transaction Error Situations.....	131

About this Guide

- Document Conventions 6
- Online Information and Support 7
- Data Protection 8

This guide describes how to configure and use webMethods MSMQ Adapter 6.0. It contains information for administrators who configure and manage a webMethods system and for application developers who want to create webMethods Integration Server services that exchange data with Microsoft Message Queue (MSMQ) Server systems.

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

- General concepts of messaging and queuing
- The MSMQ Server
- Terminology and basic operations of your operating system
- The setup and operation of webMethods Integration Server.
- How to perform basic tasks with 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

Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

In addition, you can also access the cloud product documentation via <https://www.softwareag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at <https://knowledge.softwareag.com>.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://techcommunity.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/u/softwareag> and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

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 Overview of MSMQ Adapter

■ About MSMQ Adapter	10
■ Overview of Queuing Concepts	10
■ Architecture and Components	12
■ Package Management	14
■ Adapter Connections	15
■ Adapter Services	17
■ Adapter Listeners and Listener Notifications	22
■ Using Version Control Systems to Manage Adapter Elements	27
■ Infrastructure Data Collector Support for the MSMQ Adapter	28
■ Viewing the Adapter's Update Level	28
■ Controlling Pagination	28

About MSMQ Adapter

webMethods MSMQ Adapter is an add-on to webMethods Integration Server that allows you to exchange information with other systems through a Microsoft Message Queue (MSMQ) system. This capability provides seamless and real-time communication to and from the MSMQ system.

Using MSMQ Adapter, you can route documents, or any piece of information, from webMethods components to systems that use Microsoft Message Queuing. For example, you might use MSMQ Adapter to extract information from an XML-based purchase order, repackage it as an order record, and deliver it to a queue for processing by a back-end enterprise system. You also might use MSMQ Adapter to monitor a queue and issue updates to various systems when a message appears in that queue.

Overview of Queuing Concepts

This section discusses the following topics:

- Queuing
- Message queuing computer
- Messages

Queuing

Queuing is a programming technique in which two programs exchange data asynchronously, without calling one another directly. Using queuing, two programs communicate by passing messages to one another via a queue. Conceptually, a queue is like an "inbox." When one program wants another program to process a piece of information, it simply puts that information into the other program's queue. On the other end, the receiving program monitors its designated queue and processes messages that other programs deposit in it.

There are two types of queues, public and private. Public queues are registered with a directory service and can be located by any message queuing application. Private queues are registered solely on the local system and usually cannot be located by other applications.

Systems that interface with one another through queuing never interact directly; they only interact with their queues. In fact, a program might not process its queue until hours, days, or weeks after another program deposits a message in it.

Message Queuing Computer

A message queuing computer, such as an MSMQ message queuing computer, acts as an intermediary between the programs that want to exchange information. It functions as a server, controlling the flow of information in and out of a set of queues. Queues reside on the message queuing computer and you can associate multiple queues with one message queuing computer.

The programs that want to communicate with one another (whether they are the suppliers of information or the consumers of information) are the clients of message queuing. Because these

programs do not interact directly, they are considered peers of one another. Under this architecture, one program is not subordinate to, or a client of, the other.

Note:

The message queuing computer does not actively process or distribute the messages it receives. Its role is to simply accept messages and hold them in the appropriate queue until a client explicitly requests them.

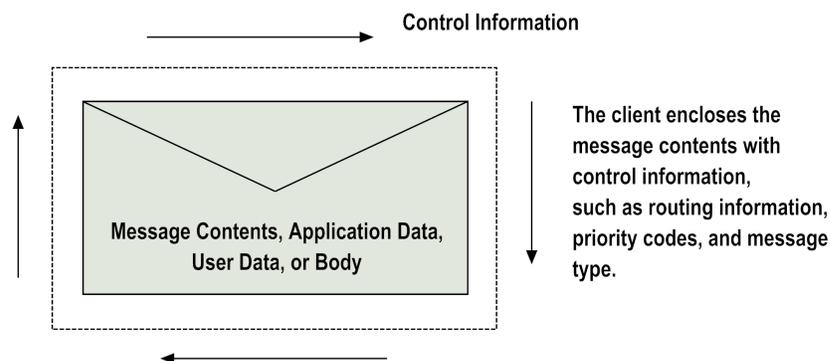
Messages

A message is a discrete unit of information that one program wants to pass to another. The following examples are possible message types:

- A string containing a purchase order
- A binary data structure containing an employee record

The message queuing computer does not impose any constraints on the content or format of messages exchanged via a queue, and neither does MSMQ Adapter.

When a program puts a message into a queue, the client envelops or wraps the message with control information such as routing information, priority codes, and message type. The term "message" also can refer to the entire message structure; that is, the control information in addition to the unit of information to be passed. In this context, the unit of information is usually referred to as the message body, the application data, or the user data. When this distinction is necessary, this book uses the term *Body* to refer to the content of the message.

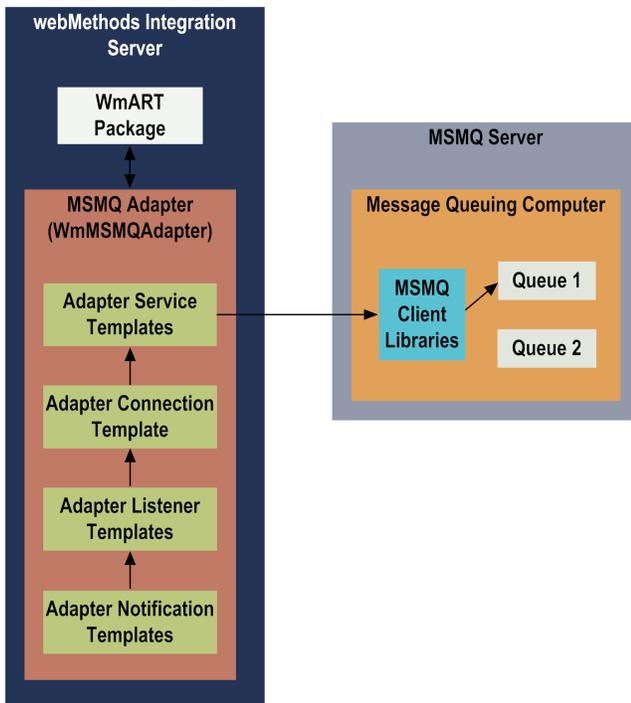


To use queuing effectively, programs that communicate via a queue must agree on a common message content and structure. For example, if Program A is designed to pick up and process cXML-formatted purchase orders from a queue named "Orders", programs that interact with Program A must deliver properly-formatted cXML purchase orders to the Orders queue. Identifying the content and structure of the Body is a key activity that you must perform when designing a system that exchanges data through MSMQ Adapter.

Architecture and Components

This diagram illustrates how MSMQ Adapter interfaces with an MSMQ system using the MSMQ libraries. 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 MSMQ Adapter services, see [“Adapter Services” on page 17](#).



- **webMethods Integration Server** . MSMQ Adapter is installed and runs on Integration Server.
- **WmART Package**. The WmART package provides a common framework for webMethods 6.0 and later adapters to use Integration Server functionality, making Integration Server the run-time environment for MSMQ Adapter. The WmART package is installed with Integration Server.
- **MSMQ Adapter**. MSMQ Adapter is delivered as a single package called WmMSMQAdapter. MSMQ Adapter provides Integration Server Administrator user interfaces that enable you to configure and manage adapter connections and listeners, and Software AG Designer user interfaces that enable you to configure and manage adapter services and notifications.
- **Adapter Connection Templates**. Adapter connections enable MSMQ Adapter to connect to MSMQ systems. You must configure an adapter connection before you can configure adapter services. The adapter provides a template for adapter connections in Integration Server Administrator. For a detailed description of adapter connections, see [“Adapter Connections” on page 15](#).
- **Adapter Service Templates**. Adapter services enable MSMQ Adapter to implement the MSMQ COM libraries to perform operations on MSMQ queues. MSMQ Adapter provides adapter

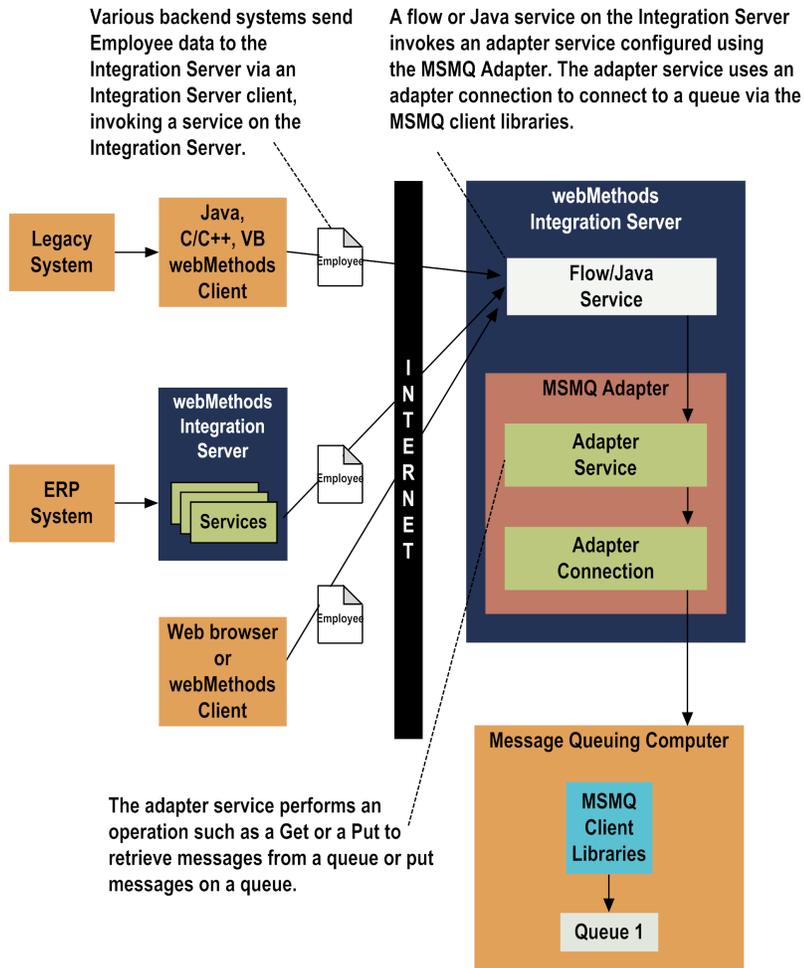
service templates that enable you to configure services to put, get, and peek messages. For example, a Get service enables you to retrieve messages from a MSMQ queue that are then sent to another application such as a purchase order entry system. MSMQ Adapter's adapter service templates are available in Designer. For more information about adapter services and using the adapter service templates, see [“Adapter Services” on page 17](#).

- **Adapter Listener Template.** An adapter listener monitors the queue for messages and passes the messages to a listener notification. You must configure adapter listeners before you can configure adapter notifications. MSMQ Adapter provides adapter listener templates that enable you to configure listeners, which monitor a queue for messages. For more information about listeners and how to use them, see [“Adapter Listeners and Listener Notifications” on page 22](#).
- **Adapter Notification Template.** A notification waits for an adapter listener to pass a message to a listener notification. A listener notification then passes the messages to Integration Server. For example, an adapter listener could listen for a specific priority on a queue, such as a priority of 4. The adapter listener monitors the queue for all messages that contain this priority, retrieves the messages from the queue, and then passes them to the adapter listener notification for further processing.

MSMQ Adapter provides adapter notification templates that enable you to configure notifications, which process the message either synchronously or asynchronously. For more information about listener notifications and using the templates, see [“Adapter Listeners and Listener Notifications” on page 22](#).

- **MSMQ Libraries.** MSMQ Adapter uses the MSMQ client libraries to communicate with the MSMQ server.
- **Message Queuing Computer.** The message queuing computer acts as an intermediary between the programs that want to exchange information. It also functions as a server, controlling the flow of information in and out of a set of queues. MSMQ Adapter requires that you specify the queue path name when you configure transactional or non-transactional connections. For detailed instructions on how to configure connections, see [“Configuring Adapter Connections” on page 44](#). For more information about message queuing computer, see [“Message Queuing Computer” on page 10](#).
- **Queue.** The queue acts as an "inbox" to receive messages from the adapter services and hold messages until other adapter services retrieve them. MSMQ Adapter requires that you specify the queue name when you configure transactional or non-transactional connections. For detailed instructions on how to configure connections, see [“Configuring Adapter Connections” on page 44](#). For more information about queues, see [“Queuing” on page 10](#).

The following diagram illustrates the use of MSMQ Adapter and Integration Server in a business-process integration.



Package Management

MSMQ Adapter is provided as a package called WmMSMQAdapter, which you manage like any package on Integration Server.

There are several considerations regarding how you set up and effectively manage your packages on Integration Server, such as those described in the following list.

- Configure user-defined packages for your connections, adapter services, and listener notifications. See [“Package Management” on page 33](#) for details.
- Understand how package dependencies work so that you make the best decisions about how to manage your adapter services and notifications. See [“Package Dependency Requirements and Guidelines” on page 35](#) for details.
- Control which development groups have access to which adapter services and notifications. See [“Group Access Control” on page 37](#) for details.
- Understand how clustering, an advanced feature of Integration Server, works to effectively manage your adapter services and notifications. See [“Clustering Considerations and Requirements” on page 40](#) for details.

- Enable and disable packages. See [“Enabling Packages” on page 35](#) and [“Disabling Packages” on page 36](#) for details.
- Load, reload, and unload packages. See [“Loading, Reloading, and Unloading Packages” on page 36](#).

Adapter Connections

MSMQ Adapter connects to one queue at run time; that queue can be either public or private. You can configure one or more connections at design time to use in integrations. The number of connections you configure, and the types of those connections, depend on the number of queues that you are connecting to and your integration needs.

MSMQ Adapter connections contain parameters that Integration Server uses to manage connections to the queue. MSMQ Adapter uses these connections to provide services. You configure connections using Integration Server Administrator. You must have Integration Server administrator privileges to access MSMQ Adapter administrative screens.

For instructions to configure and manage MSMQ Adapter connections, see [“Adapter Connections” on page 43](#). For more information about setting user privileges, see the *webMethods Integration Server Administrator’s Guide* for your release.

For a list of tasks that you must complete before you can create your connections, see [“Before Configuring or Managing Adapter Connections” on page 44](#).

Transaction Management of MSMQ Adapter Connections

When you define a connection, the transaction type you choose determines the type of transaction management that the connection's operations will use. MSMQ Adapter connections support the transaction types listed in the following table. For a detailed discussion of transaction management and the MSMQ Adapter, see [“Transaction Management Overview” on page 126](#).

Transaction Type	Description
Transactional	<p>Transactional connections can be applied only to transactional queues. A transactional connection allows you to group one or more requests into a single Logical Unit of Work (LUW). You can control these requests manually (explicit transactions), or you can allow the Integration Server transaction manager to control them for you (implicit transactions).</p> <p>To control these requests explicitly within a given LUW, you use the built-in services described in “Built-In Transaction Management Services” on page 125. If you do not use these built-in services, Integration Server will manage the requests automatically (implicitly).</p> <p>If a transaction uses only a single connection, or multiple connections that are all from the same connection pool, you do not need to manage the transaction explicitly. However, if a transaction uses multiple LOCAL_TRANSACTION transaction connections from different connection pools, you must manage the transaction explicitly.</p>

Transaction Type	Description
	<p>Note: Implicit transactions complete when the flow service that contains the LUW finishes executing. If you create a looping operation within your LUW that could potentially involve a large number of requests, you might want to manage the transactions explicitly to reduce the possibility that you will need to roll back a large number of requests in the event a single request fails.</p>
<p>Non-Transactional</p>	<p>A non-transactional connection is stateless and can be applied only to non-transactional queues. All services associated with a non-transactional connection perform a single request as an autonomous unit of work.</p> <p>For example, when a flow invokes a Get service associated with a non-transactional connection, these services remove each message from the MSMQ queue immediately after they process the message.</p>

Changing the Connection Associated With an Adapter Service at Design Time

Integration Server provides a built-in service that you can use at design time to change the connection associated with an adapter service. This built-in service, `setAdapterServiceNodeConnection`, is provided in the WmART package's `pub.art.service` folder. Using this service, you can change the specific connection associated with an adapter service at design time so that you do not need to create and maintain multiple adapter services.

Note:

You should run this built-in service at design time only; do not use it 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, see the *webMethods Integration Server Built-In Services Reference* for your release.

Changing the Connection Associated with 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 example, a service can be defined to use a default connection that interacts with customer number 1's order queue. However, at runtime you can override the default connection and instead use another connection to interact with customer number 2's order queue.

For more information about overriding a service's default connection at runtime, see [“Dynamically Changing a Service's Connection at Run Time” on page 48](#).

Connection Pools

Integration Server includes a connection management service that dynamically manages connections and connection pools based on configuration settings that you specify for the connection in Integration Server Administrator. All adapter services and adapter listeners 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 configures 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**.

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 43](#).

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 details, see [“Changing the Connection Associated With an Adapter Service at Design Time” on page 16](#).

For information about the `setAdapterServiceNodeConnection` service, see the *webMethods Integration Server Built-In Services Reference* for your release.

Adapter Services

Adapter services enable you to connect to the MSMQ system and initiate an operation on the MSMQ system from Integration Server. For example, you can create a Put service and put a message on a MSMQ queue.

You call adapter services from flow services or Java services to interact with queues. Integration Server uses the adapter connections to execute the adapter services, which perform MSMQ messaging services.

Adapter services are based on templates provided with the MSMQ Adapter. Each template represents a specific technique for doing work on the MSMQ system, such as using the Put template to deliver a message to a specified queue. An adapter service template contains all the code necessary for interacting with the MSMQ system but without the data specifications. You provide these specifications when you configure a new adapter service.

Configuring a new service from an adapter service template is straightforward. Using Designer, you assign the service a default adapter connection.

Important:

If you are using the adapter with Integration Server 8.0 SP1 or earlier, adapter services cannot use connections that are also used for adapter listeners.

After you select the connection for the adapter service, you use Designer to configure the adapter service by selecting the adapter service template and the data specifications, such as the message property fields. For information, see the *webMethods Service Development Help* for your release.

The MSMQ Adapter provides the following adapter service templates:

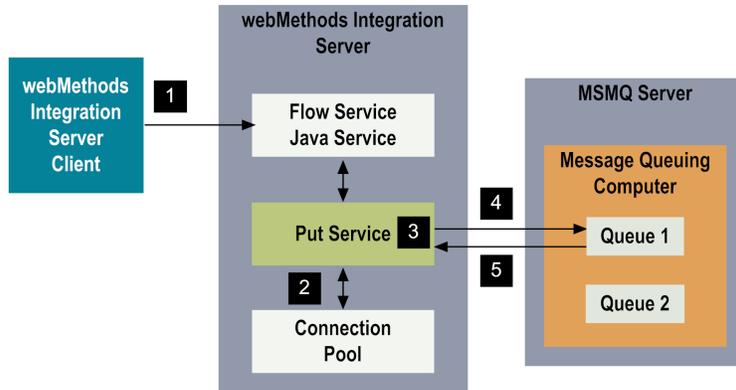
Adapter Service Template	Description
Put Service	Opens a session to an MSMQ Server and delivers a message to a specified queue. For more information, see “Configuring Put Services” on page 55 .
Get Service	Opens a session to an MSMQ Server and retrieves and removes a message from a specified queue. For more information, see “Configuring Get Services” on page 59 .
Peek Service	Opens a session to an MSMQ Server and retrieves a copy of a message without removing the original message from a specified queue. “Configuring Peek Services” on page 64 .

Put Service

A Put service creates an MSMQMessage object from the input signature properties, which becomes the input document. You specify input signature properties when you configure a Put service. A Put service places the message into an MSMQ queue.

Run-Time Processing for Put Services

The following diagram illustrates how a Put service delivers a message to a queue on an MSMQ Server.



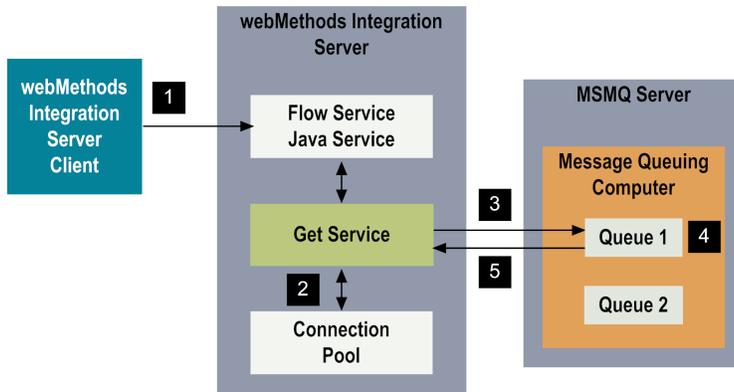
Step	Description
1	<p>A flow or Java service, typically invoked by an Integration Server client, initiates the Put service on Integration Server.</p> <p>You configure the Put service, and the wrapping flow or Java service, using Designer.</p>
2	<p>The Put service retrieves a connection from the service's associated connection pool.</p> <p>You configure and enable the adapter connection using Integration Server Administrator. For more information about connection pooling, including the run-time behavior of connection pools, see “Connection Pools” on page 17.</p>
3	<p>The MSMQ Adapter uses the Put service's input document, or the message properties from the input signature of the Put service, to create an MSMQMessage object.</p> <p>You define the Put service's input signature when you configure the service. Using Designer, the fields that you select on the Input Message Property tab in the adapter service become the service's input signature.</p>
4	<p>The MSMQ Adapter puts the MSMQMessage object into the MSMQ queue.</p>
5	<p>The Put service returns a document that contains the output message fields. You define the output signature of the Put service, or output document, when you configure the service. The fields that you select on the Output Message Property tab become the service's output signature.</p>

Get Service

A Get service retrieves and removes messages from an MSMQ queue. A Get service also creates an output document that contains message properties and the Body. When you configure a Get service, you specify the output signature properties, which become the output document.

Run-Time Processing for Get Services

The following diagram illustrates how a Get adapter service retrieves a message from a queue on a Messaging Queuing computer. When a Get service executes, it checks the queue, retrieves the message, and converts the message into a document.



Step	Description
1	A flow or Java service, typically invoked by an Integration Server client, initiates the Get service on Integration Server. You configure the Get service, and the wrapping flow or Java service, using Designer.
2	The Get service retrieves a connection from the service's associated connection pool. You configure and enable the adapter connection using Integration Server Administrator. For more information about connection pooling, including the run-time behavior of connection pools, see "Connection Pools" on page 17.
3	The Get service uses the values from the input signature fields as filter criteria. You define the Get service's input signature when you configure the service. The fields that you select on the Filter tab in the adapter service become the input signature of the service.
4	The Get service retrieves the message from the MSMQ queue. The document, or the properties from the output signature of the Get service, contain the retrieved message properties and the Body.
5	The Get service returns a message to the Integration Server that contains output signature properties and the Body. You define the output signature of the Get service when you configure the service. The fields that you select on the Output Message Property tab become the output signature of the service.

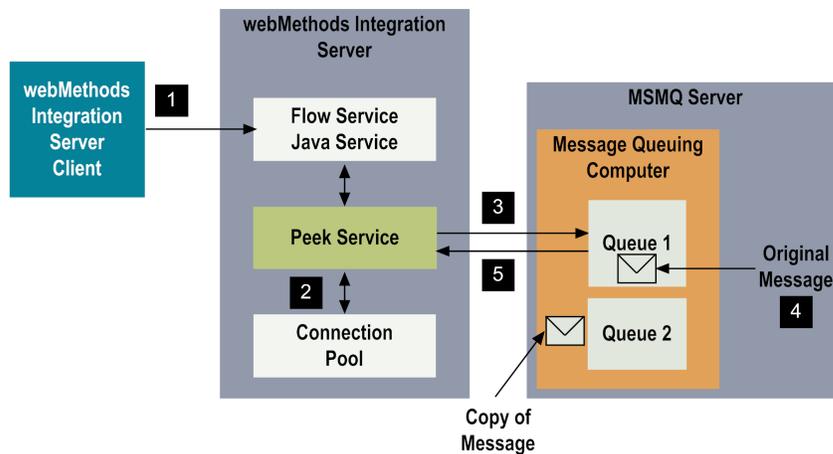
Peek Service

A Peek service retrieves a copy of the first message that matches the filter criteria specified for the Peek service from the MSMQ queue. A Peek service creates an output document that contains

message properties and the Body. When you configure a Peek service, you specify the output signature properties, which become the output document.

Run-Time Processing for Peek Services

The following diagram shows how a Peek adapter service retrieves a copy of a message from a queue on a Messaging Queuing computer without removing the original message. When the Peek service executes, it checks the queue, retrieves a copy of the message, and converts the message data into a document.



Step	Description
1	A flow or Java service, typically invoked by an Integration Server client, initiates the Peek service on Integration Server. You configure the Peek service, and the wrapping flow or Java service, using Designer.
2	The Peek service retrieves a connection from the connection pool associated with the service. You configure and enable the adapter connection using Integration Server Administrator. For more information about connection pooling, including the run-time behavior of connection pools, see “Connection Pools” on page 17.
3	The Peek service uses the values from the input signature fields as filter criteria. You define the input signature of the Peek service when you configure the service. The fields that you select on the Filter tab in the adapter service become the service's input signature.
4	The Peek service retrieves a copy of the message from the MSMQ queue and leaves the original message on the queue. The MSMQ message, or the properties from the output signature of the Peek service, contain the retrieved message properties and the Body.
5	The Peek service returns a message to the Integration Server that contains output signature properties. You define the output signature of the Peek service when you configure the

Step	Description
	service. The fields that you select on the Output Message Property tab become the output signature of the service.

Adapter Listeners and Listener Notifications

The MSMQ Adapter provides listeners and listener notifications to perform the following functions:

- Monitor messages
- Filter messages
- Process messages

Listeners

A listener continually monitors a queue for messages. When a message appears on a queue, the listener fetches the message. The listener passes the message to the listener notification.

Unlike the Put and Get adapter services, you never invoke a listener directly from a service or client. Instead, a listener is a real-time process that you configure, enable, and disable using Integration Server Administrator. Refer to [“Configuring New Listeners” on page 76](#) for detailed instructions on how to configure listeners.

Listener Notifications

A listener notification works in conjunction with a listener to filter and process messages in the MSMQ Adapter. When a listener detects a message on a queue, the listener performs a Get operation to retrieve the message and passes the message to the listener notification. The listener notification receives the message and performs a filter on the messages based on the filter criteria that you selected when you configured the listener notification. The listener notification processes the message either synchronously or asynchronously.

The MSMQ Adapter includes the following listener notifications:

- Asynchronous Listener Notifications
- Synchronous Listener Notifications

Asynchronous Listener Notifications

When you create an asynchronous notification, the MSMQ Adapter creates a publishable document type. The publishable document type defines the structure of the document that includes the *Body* and message properties. At run time, after the listener retrieves a message from the queue, the listener invokes the asynchronous notification. The asynchronous notification publishes the document that has the structure defined by the document type that was created along with the asynchronous notification.

An asynchronous listener notification can publish a document in either of the following ways:

- Publish to a Broker when Integration Server is connected to a Broker (default).
- Publish locally to Integration Server when Integration Server is not connected to a Broker.

To process the message from the queue, create an Integration Server trigger that subscribes to the document type that the MSMQ Adapter created along with the asynchronous notification. For more information about using triggers with services, see the *Publish-Subscribe Developer's Guide* for your release.

Beginning with Integration Server 8.0, you can choose the destination to which asynchronous notifications should publish messages. Specifically, you can choose whether the asynchronous notification templates use JMS APIs to publish messages to Integration Server or webMethods Broker APIs to publish notification messages to webMethods Broker.

Note:

To use the JMS protocol with asynchronous notifications, you must first configure a JMS connection alias on Integration Server. For more information, see the *webMethods Integration Server Administrator's Guide* for your release.

For steps for selecting a publish destination for asynchronous notification messages, see [“Listener Notifications” on page 84](#).

Synchronous Listener Notifications

When you create a synchronous listener notification, the MSMQ Adapter creates the following document types:

- **Request Document.** The synchronous Request document contains the Body and message properties.
- **Reply Document.** At run time the listener notification invokes a specified service and passes it to the specified Request document. The service returns the Reply document to the MSMQ Adapter. The listener notification waits until the service has finished processing the message before it initiates the next service. A synchronous listener notification does *not* publish a document.

Note:

WmART generates both document types by default, however, the MSMQ Adapter does not use the Reply Document.

Synchronous adapter notifications execute flow services and return values from those services. Integration Server 8.0 supports the following two execution modes to determine how the notification executes a flow service and returns output:

- **Service Invoke mode:** When this mode is selected, the adapter notification invokes a flow service directly and returns values as output from that service.
- **Publish and Wait mode:** When this mode is selected, the adapter notification publishes the received data either to the local Integration Server or to the Broker connected to that Integration Server by converting request documents into publishable documents, and then waits for a

reply. This mode is needed when a synchronous adapter notification is represented as a receive step in a process model.

For steps for specifying execution modes for synchronous adapter notifications, see [“Listener Notifications” on page 84](#).

Listener Notification Templates

The MSMQ Adapter provides the following adapter listener notification templates:

Listener Notification Template	Description
Asynchronous Listener Notification	Publishes a document type that Integration Server processes asynchronously. That is, the Integration Server does not wait until it has finished processing a document before processing another document.
Synchronous Listener Notification	Invokes a service to process the message before receiving the next message.

For more information about configuring asynchronous and synchronous notifications, see [“Configuring Listener Notifications” on page 86](#).

Listener Transactionality and Behavior

Listeners can be either transactional or non-transactional, and exhibit different behaviors when they are associated with a non-transactional connection than with a transactional connection.

Transactional Listeners

A listener is transactional when it is associated with a transactional connection, and exhibits the following behavior:

- The Listener monitors the queue.
- The Listener receives a message and passes it to the notifications. (The MSMQ Adapter removes this message from the queue.)
- If the message matches all of the selection criteria for any notification, the notification processes the message.
- If the notification processes the message, the Listener commits the message. (The MSMQ Adapter removes the message from the queue.)
- If no notification processes the message, the message is lost.

Important:

If you enable a transactional listener, but have not enabled any notifications to process the message, the message may be lost. To prevent the loss of messages, Software AG recommends

a default notification that is always enabled. For information about notification configuration, see [“Listener Notifications” on page 84](#).

Non-Transactional Listeners

A listener is non-transactional when it is associated with a non-transactional connection, and exhibits the following behavior:

- The Listener monitors the queue.
- The Listener receives a message. (The MSMQ Adapter removes this message from the queue.)
- If the message matches all of the selection criteria for any notification, the notification processes the message.

However, if the message does not match the selection criteria for the notification, the notification discards the message.

Important:

For a non-transactional listener, there should be an associated listener notification to filter each type of message that arrives in the queue. If you do not have a listener notification that matches the criteria of a message coming through the queue, the first message will be lost. To prevent messages from being lost, Software AG recommends that you configure and enable a default listener notification with no filter criteria.

Important:

If you enable a non-transactional listener, but have not enabled any notifications to process the message, the message is lost. To prevent the loss of messages, Software AG recommends a default notification that is always enabled. For information about notification configuration, see [“Listener Notifications” on page 84](#).

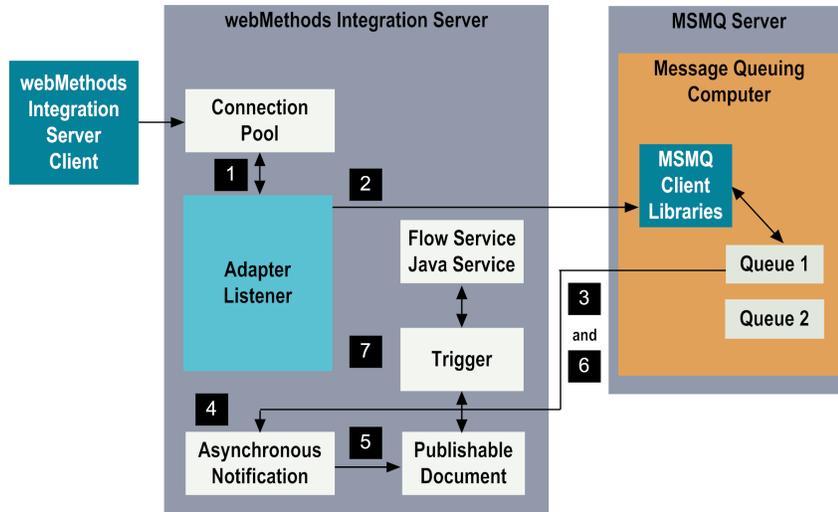
Run-Time Processing of Listeners and Notifications

Run-time processing occurs in listeners and notifications in the following scenarios:

- Listeners and asynchronous listener notifications
- Listeners and synchronous listener notifications

Run-Time Processing of Listeners and Asynchronous Listener Notifications

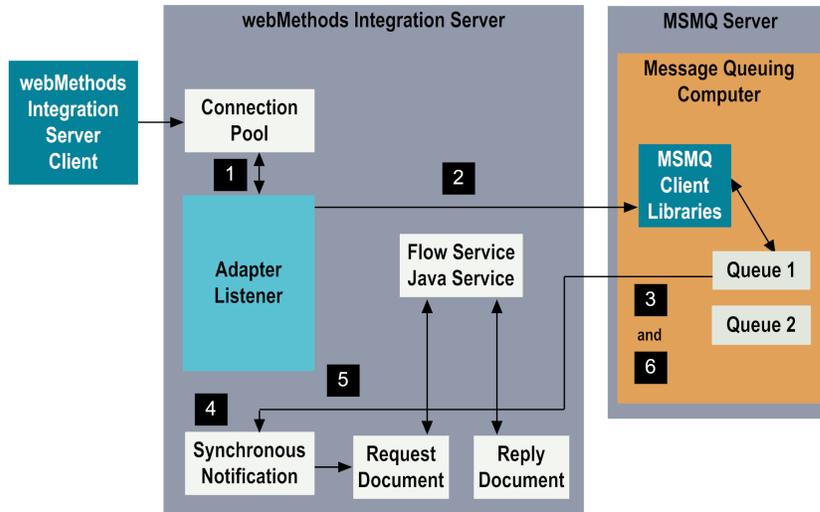
The following diagram and steps illustrate what happens when a listener retrieves a message from a queue and passes the message to an asynchronous listener notification.



Step	Description
1	The listener retrieves a connection from the connection pool associated with the service.
2	The listener uses the MSMQ libraries to monitor and listen for messages on a queue.
3	When a message appears on the queue, the listener retrieves the message from the queue and passes it to the asynchronous notification.
4	The asynchronous notification compares the message property fields against the Filter fields specified for the notification. If all specified criteria matches between the two sets of fields, the asynchronous notification will be invoked to publish the document. If the specified criteria do not match between the message property fields and the Filter fields, the asynchronous notification is not invoked to publish the document.
5	The asynchronous notification creates a publishable document, which contains the <i>Body</i> , message properties, and envelope information. The asynchronous notification publishes the document that the service, which you created, processes at a later time.
6	The listener gets the next message from the queue, and the MSMQ Adapter repeats steps 1 through 5.
7	Using an Integration Server trigger that you configured to use the publishable document created with the notification, a flow or Java service that you configured on Integration Server is invoked to process the message data in the publishable document.

Run-Time Processing of Listeners and Synchronous Listener Notifications

The following diagram and steps illustrate what happens when a listener retrieves a message from the queue and invokes a synchronous listener notification.



Step	Description
1	The listener retrieves a connection from the connection pool associated with the service.
2	The listener uses the MSMQ libraries to monitor and listen for messages on a queue.
3	When a message appears, the listener retrieves the message from the queue. The listener passes the message to the synchronous notification.
4	The synchronous notification compares the message properties fields against the Filter fields specified for the notification. If all specified criteria matches between the two sets of fields, the listener invokes the synchronous notification to process the message. If the specified criteria does not match between the message property fields and the Filter fields, the synchronous notification is not invoked to process the message.
5	The synchronous notification creates the following documents types: <ul style="list-style-type: none"> ■ Request document. The Request document contains the <i>Body</i> and message properties. The synchronous notification invokes the service that you selected when you configured the synchronous notification and passes in the Request document. ■ Reply document. The Reply document is generated by default. You can disregard this document type because the MSMQ Adapter does not use it.
6	After the service has processed the data in the Request document, the listener retrieves another message from the queue.

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 *Configuring 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*.

Infrastructure Data Collector Support for the MSMQ Adapter

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 the Optimize documentation for your release.

Viewing the Adapter's Update Level

You can use Integration Server to 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

When using the adapter on Integration Server 8.0 and later, you can control the number of items that are displayed on the adapter Connections screen and 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 MSMQ Adapter

■ Overview	30
■ Requirements	30
■ Installing MSMQ Adapter 6.0	30
■ Upgrading to MSMQ Adapter 6.0	31
■ Uninstalling MSMQ Adapter 6.0	31

Overview

This chapter explains how to install, upgrade, and uninstall webMethods MSMQ Adapter 6.0. 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, MSMQ products, and webMethods products supported by MSMQ Adapter 6.0, see the *webMethods Adapters System Requirements*.

MSMQ Adapter 6.0 has no hardware requirements beyond those of its host Integration Server.

Installing MSMQ Adapter 6.0

Note:

If you are installing MSMQ Adapter in a clustered environment, you must install the adapter 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 “[MSMQ Adapter in a Clustered Environment](#)” on page 38.

➤ **To install MSMQ Adapter 6.0**

1. Download Installer from the [Empower Product Support Web site](#).
2. If you are installing the adapter on an existing Integration Server, shut down Integration Server.
3. Start the Installer wizard.
 - a. 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 7.1, choose the 7.1 release.
 - b. 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. Installer will install the adapter in the *Integration Server_directory \packages* directory.
 - c. In the product selection list, select **Adapters > webMethods MSMQ Adapter 6.0**. You can also choose to install documentation.
4. After installation is complete, start the host Integration Server.

Upgrading to MSMQ Adapter 6.0

You cannot upgrade to MSMQ Adapter 6.0 from earlier versions of the adapter.

Uninstalling MSMQ Adapter 6.0

➤ To uninstall MSMQ Adapter

1. Shut down the Integration Server that hosts MSMQ Adapter.
2. Software AG Uninstaller will not delete any user-defined MSMQ Adapter 6.0 components such as connections and adapter services. Because these components will not work without the adapter, delete them manually from either at the file system level or using Software AG Designer. For instructions, see the *webMethods Service Development Help* for your release.
3. Start Uninstaller, selecting the webMethods installation directory that contains the host Integration Server. In the product selection list, select **Adapters > webMethods MSMQ Adapter 6.0**. You can also choose to uninstall documentation.
4. Restart the host Integration Server.
5. Uninstaller removes all MSMQ Adapter 6.0 related files that were installed into the *Integration Server_directory* \packages\WmMSMQAdapter 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 delete the WmMSMQAdapter directory.

3 Package Management

- Overview 34
- MSMQ Adapter Package Management 34
- Group Access Control 37
- MSMQ Adapter in a Clustered Environment 38

Overview

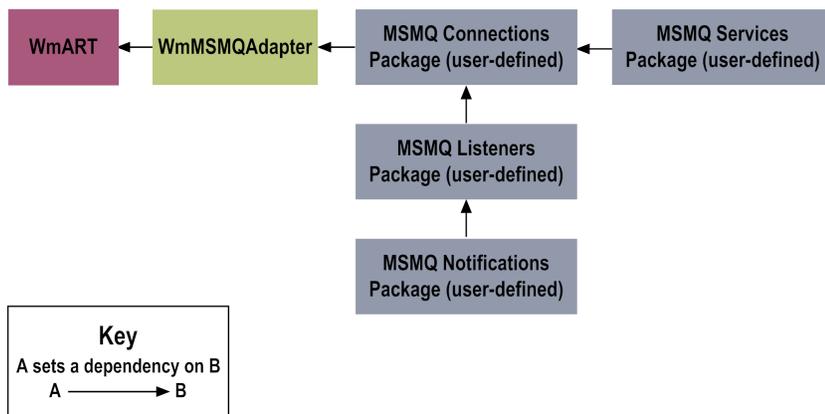
The following sections describe how to set up and manage your webMethods MSMQ Adapter packages, set up access control lists (ACL), and use the adapter in a clustered environment.

MSMQ Adapter Package Management

MSMQ Adapter is provided as a package called WmMSMQAdapter. You manage the WmMSMQAdapter package as you would manage any package on Integration Server.

When you create connections, adapter services, adapter notifications, and adapter listeners, define them in user-defined packages rather than in the WmMSMQAdapter package. Doing this allows you to manage the packages more easily, especially when you need to upgrade a deployed adapter. You can create all the connections and adapter services of an adapter in one package, or you can distribute them among multiple packages. For more information about creating packages, see the *webMethods Service Development Help* for your release.

As you create user-defined packages, use the package management functionality provided in Designer and set the user-defined packages to have a dependency on the WmMSMQAdapter package. That way, when the WmMSMQAdapter package loads or reloads, the user-defined packages load automatically. The following diagram illustrates package dependencies for MSMQ Adapter.



Package management tasks include:

- [Setting package dependencies \(see “Package Dependency Requirements and Guidelines” on page 35\).](#)
- [“Enabling Packages” on page 35.](#)
- [“Disabling Packages” on page 36.](#)
- [“Group Access Control” on page 37.](#)

Package Dependency Requirements and Guidelines

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

- By default, the WmMSMQAdapter package has a dependency on the WmART package. *Do not* change this dependency.
- A user-defined package must have a dependency on the WmMSMQAdapter package. Without this dependency, connections and listeners become disabled when the WmMSMQAdapter package is reloaded.

These dependencies also 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 packages last. The WmART package is installed automatically 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:
 - A package that contains the connections must have a dependency on the adapter package.
 - Packages that contain adapter services, adapter notifications, or adapter listeners must depend 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 35](#).
- 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 36](#).
- You can name connections and adapter services with the same name provided that they are in different packages and folders.

Enabling Packages

All packages are enabled automatically by default. If you want to manually enable a user-defined package, you must first enable its associated adapter package (WmMSMQAdapter).

> 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 for the package that you want to enable.
4. The server issues a prompt to verify that you want to enable the package.

When the package is enabled, the server displays a ✓ and **Yes** in the **Enabled** column.

Note:

Enabling an adapter package will not cause its associated user-defined package(s) to be reloaded. For information about reloading packages, see [“Loading, Reloading, and Unloading Packages”](#) on page 36.

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 package will remain disabled until you explicitly enable it using Integration Server Administrator, and will not be listed in 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.
4. 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.

Loading, Reloading, and Unloading Packages

Recall that if user-defined packages are properly configured with a dependency on the adapter package (as described in [“Package Dependency Requirements and Guidelines”](#) on page 35), at startup Integration Server automatically loads or reloads all packages in the proper order: the

WmART package first, the adapter package next, and the node packages last. You should not need to manually reload the WmART package.

Reloading Packages Manually

Reloading a user-defined package will not cause its associated adapter package to be reloaded. You can reload adapter packages and user-defined packages from either Integration Server Administrator (by clicking the **Reload** icon on the Management window) or from Designer (by right-clicking the package and selecting the **Reload Package** option from the menu).

Unloading Packages

At shutdown, Integration Server unloads packages in the reverse order in which it loaded them: it unloads the node packages first, the adapter package next, and the WmART package last (assuming the dependencies are correct).

Setting Package Dependencies

You set package dependencies if a given package needs services in another package to load before it can load. For example, any packages you create for MSMQ Adapter services should identify the webMethods MSMQ Adapter package (WmMSMQAdapter) as a package dependency because they require services in the WmMSMQAdapter to load first. Use the following guidelines:

- Set package dependencies from the adapter service package to the package containing the connection if you configure a connection in one package and the adapter services in another package. That is, the package that contains the connection should load before the adapter service package.

When you set this package dependency, it ensures that if someone disables the connection package and then re-enables it, the adapter services will reload correctly.

- If both the connection and adapter services are in the same package, then no dependencies need to be set.
- In general, packages containing connections should have a dependency set to the adapter package itself. That is, the adapter service package should depend on the adapter connection package, which should depend on the adapter package. Similarly, if the adapter services are in the same package as the connections, the only dependency that you need to set is between the adapter connection package and the adapter package.

For more information about setting package dependencies, see the *webMethods Service Development Help* for your release.

Group Access Control

To control which development group has access to which adapter services and adapter notifications, 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.

MSMQ Adapter in a Clustered Environment

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

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

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 MSMQ Adapter 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 redirected automatically to another server in the cluster.

Note: 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 Integration Server to the cluster.

Adapter Component Support in Clusters

The following MSMQ Adapter components are supported in a clustering environment; however, there may be some restrictions, as discussed below:

- **Put services.** Can be used in clusters regardless of the transaction type used by the associated adapter connection.
- **Get services.** Can be used in clusters regardless of the transaction type used by the associated adapter connection.
- **Peek services.** Can be used in clusters regardless of the transaction type used by the associated adapter connection. However, the machine on which Integration Server is located must be connected to the machine on which the queue is located.

- **Listeners.** When used with a transactional connection, listeners cannot be used in clusters. This is because the listener must reside on the same machine as the queue, and MSMQ does not support remote transactional read from remote queues.

When used with a non-transactional connection, listeners can be used in a cluster. Ensure that the machine on which Integration Server and MSMQ Adapter are installed and the machine on which the queue is located are on the same network.

Important:

Because MSMQ does not guarantee message ordering in non-transactional messaging, Software AG recommends that you do not deploy a listener that uses a non-transactional connection across a cluster. However, if ordering is not critical for your intended application, duplicate messages can be detected with the exactly-once processing facility in Integration Server. For additional information about duplicate detection and exactly-once processing, see the *Publish-Subscribe Developer's Guide* for your release.

- **Listener notifications.** When synchronous or asynchronous listener notifications are used in a clustered environment, messages may be duplicated and/or received out of order. If necessary, you can use the exactly-once processing facility in Integration Server to detect duplicate messages. If you need to guarantee message order, the application processing the notification documents must handle any reordering of messages that is required. For additional information about duplicate detection and exactly-once processing, see the *Publish-Subscribe Developer's Guide* for your release.

MSMQ Adapter Notification Support in Clusters

MSMQ Adapter supports non-transactional listeners with asynchronous notifications in a clustered environment.

Duplicate Documents

Integration Server provides MSMQ Adapter support for duplicate document detection in a cluster using its publish and subscribe facility.

Ordering of Documents

Neither MSMQ Adapter nor MSMQ guarantee the order of messages received from MSMQ. The application processing the notification documents must handle any reordering of messages that is required.

Replicating Packages to Integration Servers

Every Integration Server in the cluster should contain an identical set of packages that you define using MSMQ Adapter; that is, you should replicate MSMQ Adapter adapter connections, adapter services, flow services, listeners, triggers, and notifications.

To ensure consistency, 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.

Clustering Considerations and Requirements

Note:

The following sections assume that you already have configured the 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 MSMQ Adapter in a clustered environment.

Requirements for Each Integration Server in a Cluster

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

All Integration Servers For Example... in a given cluster must have identical...	
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 MSMQ Adapters in the cluster must be the same version, with the same service packs and fixes applied.
Adapter connections	If you configure a connection to the MSMQ Server, 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>Note: If you plan to use connection pools in a clustered environment, see “Considerations When Configuring Connections with Connection Pooling Enabled” on page 41.</p>
Adapter services	<p>If you configure a specific 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.</p> <p>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.</p>

See [“Replicating Packages to Integration Servers” on page 39](#) for information about replicating adapter packages, connections, and adapter services across multiple Integration Servers in a cluster.

Considerations When Installing MSMQ Adapter Packages

For each Integration Server in the cluster, use the standard MSMQ Adapter installation procedures for each machine, as described in [“Installing, Upgrading, and Uninstalling the MSMQ Adapter” on page 29](#).

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 queue.

For example, if you have a cluster of two Integration Servers with a connection configured to a queue that supports a maximum of 100 connections, 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 then replicate the connection to both servers because a total of 200 connections could then be opened to this queue, exceeding the queue limit.

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 queue connections that can be open at one time.

For information about configuring connections for MSMQ Adapter, see [“Configuring Adapter Connections” on page 44](#).

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

Disabling the Redirection of Administrative Services

As mentioned in [“MSMQ Adapter in a Clustered Environment” on page 38](#), a server that cannot handle a client’s service request can automatically redirect the request to another server in the cluster. However, the WmART package 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 create two MSMQ Adapter 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 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:

Integration Server_directory \config\redir.cnf

3. Add the following line to the file:

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

4. Save the file and restart Integration Server.

4 Adapter Connections

■ Overview	44
■ Before Configuring or Managing Adapter Connections	44
■ Configuring Adapter Connections	44
■ Dynamically Changing a Service's Connection at Run Time	48
■ Viewing Adapter Connection Parameters	49
■ Editing Adapter Connections	50
■ Copying Adapter Connections	51
■ Deleting Adapter Connections	51
■ Enabling Adapter Connections	52
■ Disabling Adapter Connections	52

Overview

The following sections provide instructions for configuring and managing MSMQ Adapter connections. For more information about how connections work, see [“Adapter Connections” on page 15](#).

Before Configuring or Managing Adapter Connections

The following procedure describes what you need to do before you create and configure your MSMQ Adapter connection.

➤ To prepare to configure or manage adapter connections

1. Make sure that the location of webMethods Integration Server allows access to the MSMQ system (that they are located within the same network domain).
2. Make sure you have Integration Server administrator privileges so that you can access MSMQ Adapter administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator’s Guide* for your release.
3. Start Integration Server and Integration Server Administrator, if they are not already running.
4. Using Integration Server Administrator, make sure that the WmMSMQAdapter package is enabled. To verify the status of the WmMSMQAdapter package, see [“Enabling Packages” on page 35](#).
5. Using Software AG Designer, create a user-defined package for each connection you plan to configure. For more information about managing packages, see [“Package Management” on page 33](#).

Configuring Adapter Connections

When you configure MSMQ Adapter connections, you specify information that Integration Server uses to connect to a queue and a message queuing computer. Each connection is associated with one particular queue that has specific access and share modes. You configure MSMQ Adapter connections using Integration Server Administrator.

You can configure the following connection types with MSMQ Adapter:

- MSMQ Non Transactional Connection
- MSMQ Transactional Connection

At design time, you must decide what type of connections to associate with adapter services and adapter listeners to determine how you want them to process messages. For example, a Get service requires a transactional connection to allow MSMQ Adapter to roll back messages. The connection type that is associated with an adapter listener also influences a listener notification because notifications are associated with listeners. To review the dependancies between WmART, MSMQ

Adapter, adapter connections, adapter services, adapter listeners, and listener notifications, see the figure in “[MSMQ Adapter Package Management](#)” on page 34.

Important:

If you are using the adapter with Integration Server 8.0 SP1 or earlier, listeners cannot share connections with adapter services. However, listeners can share connections with other listeners.

For more information on how each service processes a message, see appropriate run-time processing section for each service in “[About MSMQ Adapter](#)” on page 10. For more information on how listeners perform when they are associated with a non-transactional or transactional connection, see “[Listener Transactionality and Behavior](#)” on page 24.

Note:Software AG recommends that you create separate adapter connections for listeners and adapter services while working with non-transactional queues. That is, create one connection that is exclusive to listeners and another connection that is exclusive to adapter services.

➤ **To configure a connection**

1. In Integration Server Administrator in the **Adapters** menu, click **MSMQ Adapter**.
2. On the Connections screen, click **Configure New Connection**.
3. On the Connection Types screen, complete one of the following steps:
 - To make a non-transactional connection, click **MSMQ Non Transactional Connection** to display the Configure Connection Type screen.
 - To make a transactional connection, click **MSMQ Transactional Connection** to display the Configure Connection Type screen.
4. In the **MSMQ Adapter** section, provide values for the following fields:

Parameter	Description
Package	<p>The package in which to configure the connection.</p> <p>You must create the package using Designer before you can specify it using this parameter. For general information about creating packages, see the <i>webMethods Service Development Help</i> for your release.</p> <p>Configure the connection in a user-defined package rather than in the adapter's package. For other important considerations when creating packages for MSMQ Adapter, see “Package Management” on page 33.</p>
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

Parameter	Description
	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
MSMQ Version	Version of the MSMQ Server.
Transaction Type	Type of connection.
Queue Path Name	The queue path name. For example, <i>hostname\queuename</i> . Note: For remote non-transactional private queues, leave this parameter blank and complete the Queue Format Name parameter.
Queue Format Name	Optional. Unique queue name for the queue that is generated by Message Queuing. Note: If you do not specify a Queue Path Name , you must enter a Queue Format Name . An example of a queue format name for a remote private queue is <i>DIRECT=OS:hostname\private\$</i> Note: <i>\queuename</i> .
Queue Access Mode	The direction of the connection. Select one of the following: <ul style="list-style-type: none"> ■ OUTBOUND. webMethods-to-MSMQ. (Put adapter service) ■ INBOUND. MSMQ-to-webMethods. (Get or Peek adapter services or listener)
Queue Share Mode	Access to messages on the same queue. Select one of the following: <ul style="list-style-type: none"> ■ MQ_DENY_NONE. User will have access when other users access a message from the same queue. (Get, Peek, or Put adapter services or listener.) ■ MQ_DENY_RECEIVE_SHARE. User receives an error when other users get a message from the same queue. (Get adapter service or listener.)
Queue Type	The type of queue based on the transactional connection. Select one of the following: <ul style="list-style-type: none"> ■ LOCAL_QUEUE. The transactional queue on the host machine.

Parameter	Description
	<ul style="list-style-type: none"> ■ REMOTE_QUEUE. The transactional queue on a remote machine. <p>To enable your host Integration Server to read messages from remote queues, ensure the following:</p> <ul style="list-style-type: none"> ■ The message queuing service is running on the host machine. ■ MSDTC is configured and running for the host machine to receive messages from remote MSMQ servers. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p>Note: For information about configuring MSTDC for transactional remote receive on your host Integration Server, refer to the Microsoft documentation.</p> </div>

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

Parameter	Description/Action
Enable Connection Pooling	<p>Enables the connection to use connection pooling.</p> <p>For more information about connection pooling, see “Adapter Connections” on page 15.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <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 41.</p> </div>
Minimum Pool Size	<p>The minimum number of connection objects that remain in the connection pool at all times. When the adapter creates the pool, it creates this number of connections. Default: 1.</p>
Maximum Pool Size	<p>The maximum number of connection objects that can exist in the connection pool. When the connection pool has reached its maximum number of connections, the adapter will reuse any inactive connections in the pool or, if all connections are active, it will wait for a connection to become available. Default: 10.</p>
Pool Size Increment	<p>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.</p>
Block Timeout (msec)	<p>If connection pooling is enabled, this field specifies the number of milliseconds that Integration Server will wait to obtain a connection with the MSMQ Server before it times out and returns an error. Default: 1000.</p>

Parameter	Description/Action
Expire Timeout (msec)	<p>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.</p> <p>Note: Note that the adapter will never violate the Minimum Connections parameter. These connections remain in the pool regardless of how long they are inactive.</p>
Startup Retry Count	If connection pooling is enabled, this field specifies the number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails, before issuing an AdapterConnectionException.
Startup Backoff Timeout	If connection pooling is enabled, this field specifies the number of seconds to wait between each attempt to initialize the connection pool.

7. Click **Save Connection**.

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

By default, when you configure a connection, it is not enabled. For more information about enabling connections, see [“Enabling Adapter Connections” on page 52](#).

Dynamically Changing a Service's Connection at Run Time

Using Integration Server, you can 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.

For example, you have a flow whose purpose is to take input from one queue and direct it to one of three output queues, each representing a department within your company—billing, shipping, and personnel. Each document in the input queue contains a field called Recipient. One MSMQ service could act as a router, using the value in the input document's Recipient field to set the value for \$connectionName. Then the flow would use \$connectionName to dynamically pick a connection (output queue) to use to send the input to the appropriate corporate department.

Keep in mind these restrictions when using dynamic connections:

- The connection with which you override the default (that is, the value provided for \$connectionName) must be configured to use the same connection type as the default connection.
- The \$connectionName field is present only in services created with Designer.

For more information, see [“Changing the Connection Associated with an Adapter Service at Run Time”](#) on page 16.

Viewing Adapter Connection Parameters

You can view a connection's parameters from Integration Server Administrator or from 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 navigation area of Integration Server Administrator, click **MSMQ Adapter**.

When using the adapter with Integration Server 8.0 and later, 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 at the top of the column you want to sort.
- 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 28.

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 Connections”](#) on page 44.

3. Click **Return to MSMQ Adapter 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 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 [“Configuring Adapter Connections” on page 44](#).

Editing Adapter Connections

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

Note:

The Transaction Type (transactional and non transactional) cannot be edited. To change the connection type, you must create a new connection. For instructions, see [“Configuring Adapter Connections” on page 44](#).

> To edit a connection

1. In the **Adapters** menu in the navigation area of Integration Server Administrator, click **MSMQ Adapter**.
2. Make sure that the connection is disabled before editing. To disable the connection, see [“Disabling Adapter Connections” on page 52](#) for details.
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 for the connection parameters, see [“Configuring Adapter Connections” on page 44](#).

4. Click **Save Changes** to save the connection and return to the Connections screen.

Copying Adapter Connections

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

> To copy a connection

1. In the **Adapters** menu in the navigation area of Integration Server Administrator, click **MSMQ Adapter**.
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 and edit any connection parameters as needed by typing or selecting the values you want to specify.

Important:

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 Connections” on page 44](#).

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 MSMQ Adapter connection, you can delete it by following the instructions in this section. You delete adapter connections using Integration Server Administrator.

If you delete an MSMQ Adapter connection, the adapter services and listeners that are defined to use the connection will no longer work unless you change the connection an adapter service uses.

You can assign a different connection to an adapter service and re-use the service. To do this, you use the built-in webMethods service `pub.art.service:setAdapterServiceNodeConnection`. For more information, see [“Changing the Connection Associated With an Adapter Service at Design Time” on page 16](#).

> To delete a connection

1. In the **Adapters** menu in the navigation area of Integration Server Administrator, click **MSMQ Adapter**.

2. Make sure that the connection is disabled before deleting. To disable the connection, see [“Disabling Adapter Connections” on page 52](#) for details.
3. On the Connections screen, click  for the connection you want to delete.

Integration Server deletes the adapter connection.

Enabling Adapter Connections

An MSMQ Adapter connection must be enabled before you can configure any adapter service using the connection or before an adapter service 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 be enabled automatically 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 navigation area of Integration Server Administrator, click **MSMQ Adapter**.
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  and **Yes** in the **Enabled** column.

Disabling Adapter Connections

MSMQ Adapter connections must be disabled before you can edit or delete the connections. You disable the adapter connections using Integration Server Administrator.

➤ **To disable a connection**

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

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

5 Adapter Services

■ Overview	54
■ Before Configuring or Managing Adapter Services	54
■ Using Adapter Services	54
■ Configuring Put Services	55
■ Configuring Get Services	59
■ Configuring Peek Services	64
■ Using the Pipeline Editor to View and Set the Body	69
■ Testing Adapter Services	69
■ Viewing Adapter Services	70
■ Editing Adapter Services	70
■ Deleting Adapter Services	71
■ Enabling Automatic Data Validation	71
■ Reloading Adapter Values	72

Overview

This chapter describes how to configure and manage MSMQ Adapter services. For detailed descriptions of the available MSMQ Adapter services, see [“Adapter Services” on page 17](#).

Before Configuring or Managing Adapter Services

Perform the following tasks before configuring or managing adapter services.

➤ To prepare to configure or manage an adapter service

1. Start webMethods Integration Server and Integration Server Administrator, if they are not already running.
2. Make sure you have Integration Server administrator privileges so that you can access MSMQ Adapter administrative screens. For more information about setting user privileges, see the *webMethods Integration Server Administrator’s Guide* for your release.
3. Using Integration Server Administrator, make sure that the WmMSMQAdapter package is enabled. For instructions, see [“Enabling Packages” on page 35](#).
4. Using Integration Server Administrator, configure an adapter connection to use with the adapter service. For instructions, see [“Configuring Adapter Connections” on page 44](#).
5. Start Software AG Designer if it is not already running.

Note:

If you are using Designer, use the Service Development perspective. For more information, see the *webMethods Service Development Help* for your release.

6. Make sure that you enable automatic data validation. For detailed instructions on how to enable automatic data validation, see [“Enabling Automatic Data Validation” on page 71](#).
7. Using Designer, create a user-defined package to contain the service, if you have not already done so. When you configure adapter services, you always should define them in user-defined packages instead of the WmMSMQAdapter package. For more information about managing packages for the adapter, see the *webMethods Service Development Help* for your release.

Using Adapter Services

The following table lists the tasks required to use adapter services.

Task	Description	Use this tool...
1	Configure an adapter connection. See “Adapter Connections” on page 43 for details.	Integration Server Administrator

Task	Description	Use this tool...
2	Select the appropriate adapter service template and configure the adapter service. Refer to the appropriate service section in this chapter for more information on configuring adapter services.	Designer
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. For information, see the <i>webMethods Service Development Help</i> for your release. For instructions on how to invoke adapter services from flow services or java services, see “Using the Pipeline Editor to View and Set the Body” on page 69.	Designer
4	Manage the adapter service. See “Package Management” on page 33 and Logging and Exception Handling .	Designer and Integration Server Administrator

Configuring Put Services

A Put service delivers a message to a specified queue. You configure MSMQ Adapter services using Designer. The input and output signatures for the Put service are located on the **Input/Output** tab of Designer.

When you configure the Put service, the **Input** document is automatically generated in the input signature of the Put service on the **Input/Output** tab. The **Input** document contains the *Body* and the input fields from the **Input Message Property** tab. You can map a variable that is in the pipeline to the *Body* variable in the input signature of the Put service. For more information, see [“Using the Pipeline Editor to View and Set the Body”](#) on page 69.

The **Output** document also is generated automatically in the output signature of the Put service on the **Input/Output** tab. MSMQ Adapter passes the output fields from the **Output Message Properties** tab in Designer to the output signature.

For more information about Put services, see [“Put Service”](#) on page 18.

➤ To configure a Put service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 54.
2. Start Designer.
3. Using Designer, perform the following:
 - a. Right-click the package in which the service should be contained and select **New > Adapter Service**.

- b. Select the parent namespace and type a name for the adapter service.
 - c. Click **Next**.
4. Select **MSMQ Adapter** as the adapter type and click **Next**.
 5. For **Adapter Connection Name**, select a connection with an **outbound** queue access mode, and then click **Next**.

Important:

If you are using the adapter with Integration Server 8.0 SP1 or earlier, adapter services cannot use connections that are also used for adapter listeners.

6. From the list of available templates, select the **Put** template and using Designer, click **Finish**.

The adapter service editor for the Put service appears. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template**, as necessary.

7. Select the **Put Service** tab to identify optional message processing features to use with this adapter service. Set the fields as follows:

Field	Description/Action
Acknowledgement	<p>Specifies whether MSMQ generates an acknowledgment message. Select one of the following values:</p> <ul style="list-style-type: none">■ NONE indicates that no acknowledgment messages are posted.■ FULL_REACH_QUEUE posts a positive or negative acknowledgment, depending on whether the message reaches the queue.■ FULL_RECEIVE posts a positive or negative acknowledgment, depending on whether the message is retrieved from the queue before its time-to-be-received timer expires.■ NACK_REACH_QUEUE posts a negative acknowledgment when the message cannot reach the queue.■ NACK_RECEIVE posts a negative acknowledgment when the message cannot be retrieved from the queue before the message's time-to-be-received timer expires.

Note:
For all values except **NONE**, you must also specify an **Admin Queue Path Name**.

Field	Description/Action
Admin Queue Path Name	Specify an existing administration queue path name if Acknowledgement is not set to NONE . Note: Enter a valid path name for an existing queue. MSMQ Adapter does not validate the path name during run time.
Authentication	Signs the sending message based on the source computer digital signature registry entry.
Journal	A copy of the message is stored in the computer journal if the message is successfully delivered to the next computer.
Deadletter	Sends a message to an applicable dead-letter queue on failure.
Trace	Specifies whether to trace the route of a message.

8. Select the **Input Message Property** tab to identify message properties for the fields in the pipeline before the Put service is executed. Use the following icons to manage the rows in the **Property** column:

- To define new property fields, select the  icon (or the  icon).
- To ensure that the parameters are parsed in the correct order, use the  or  icons to change the order of the property fields or the output field names.
- To delete any property fields or output field names, use the  icon.

You can add the following property fields to the **Property** column:

Property	Property Type	Description
AppSpecific	java.lang.Integer	Specifies application-generated single integer values.
CorrelationId	java.lang.String	Identifies the message using a hex string correlation identifier. For example, {F944A0B1-1DA0-4A93-A266-E54ACDD55342}\30775.
CorrelationIdByteArray	byte array	Identifies the message using a 20-byte correlation identifier.
Delivery	java.lang.Integer	Specifies how Message Queuing delivers the message. Valid values are: <ul style="list-style-type: none"> ■ 1 - indicates that along its route, the message is stored locally on disk until it is forwarded to the next computer.

Property	Property Type	Description
		<ul style="list-style-type: none"> 0 - indicates that along its route, the message stays in volatile memory until it is received.
Label	java.lang.String	Provides a description of the message.
MaxTimeToReachQueue	java.lang.Integer	Specifies the time limit (in seconds) for a message to reach the queue.
Priority	java.lang.Integer	Specifies message priority. Value is between 0 - 7. The lower number indicates lower priority. Default: 3.
		<p>Note: Message Queuing automatically sets the priority level of the transactional messages to 0, which indicates that the priority is ignored by the transaction.</p>
MaxTimeToReceive	java.lang.Integer	Specifies the time limit (in seconds) for a message to be retrieved from the target queue.

9. Select the **Output Message Property** tab to identify message properties returned into the pipeline after the service is executed. You can add the following property fields to the **Property** column:

Property	Property Type	Description
MessageId	java.lang.Integer	Identifies the message using a hex string.
MessageIdByteArray	byte array	Identifies the message using an MSMQ-generated 20-byte message identifier.
CorrelationId	java.lang.String	Identifies the message using a hex string correlation identifier. For example, {F944A0B1-1DA0-4A93-A266-E54ACDD55342}\30775.
CorrelationIdByteArray	byte array	Identifies the message using a 20-byte correlation identifier.
Delivery	java.lang.Integer	Specifies how Message Queuing delivers the message. Valid values are: <ul style="list-style-type: none"> 1 - indicates that along its route, the message is stored locally on disk until it is forwarded to the next computer. 0 - indicates that along its route, the message stays in volatile memory until it is received.

Property	Property Type	Description
Label	java.lang.String	Provides a description of the message.
Priority	java.lang.Integer	Specifies message priority. The value is between 0 - 7. The lower number indicates lower priority. Default: 3.
		Note: Message Queuing automatically sets the priority level of the transactional messages to 0, which indicates that the priority is ignored by the transaction.
MaxTimeToReceive	java.lang.Integer	Specifies the time limit (in seconds) for a message to be retrieved from the target queue.

- When using Designer, Designer contains only the **Adapter Settings** and **Input/Output** tabs. The information from the **Audit** and **Permissions** tabs appears in the **Properties** panel, and the information from the **Results** tab appears in the **Service Results** panel (in Designer). For more information, see the *webMethods Service Development Help* for your release.
- From the **File** menu, select **Save**.

Configuring Get Services

A Get service retrieves and removes a message from a specified queue. You configure MSMQ Adapter services using Designer. The input and output signatures for a Get service are located on the **Input/Output** tab of Designer.

When you configure a Get service, the **Input** document is generated automatically in the input signature of the Get service on the **Input/Output** tab. MSMQ Adapter passes the input fields from the **Filter** tab in Designer to the input signature.

The **Output** document also is generated automatically in the output signature of the Get service on the **Input/Output** tab. The document contains the *Body* and the output fields from the **Output Message Properties** tab in Designer to the output signature.

For more information about Get services, see [“Get Service” on page 19](#).

➤ To configure a Get service

- Review the steps in [“Before Configuring or Managing Adapter Services” on page 54](#).
- Start Designer.
- Using Designer, perform the following:

- a. Right-click the package in which the service should be contained and select **New > Adapter Service**.
 - b. Select the parent namespace and type a name for the adapter service.
 - c. Click **Next**.
4. Select **MSMQ Adapter** as the adapter type and click **Next**.
 5. For **Adapter Connection Name**, select a connection with an **inbound** queue access mode, and then click **Next**.

Important:

If you are using the adapter with Integration Server 8.0 SP1 or earlier, adapter services cannot use connections that are also used for adapter listeners.

6. From the list of available templates, select the **Get** template and using Designer, click **Finish**.

The adapter service editor for the Get service appears. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template**, as necessary.

7. Select the **Get Service** tab to identify optional message processing features to use with this adapter service. Set the fields as follows:

Field	Description/Action
Timeout	Time period that the message queuing waits for a message to arrive. Enter this value in milliseconds.
Reset	Check to move the cursor to the front of the queue.

8. Select the **Filter** tab to specify how messages will be identified. The selected fields refer to the specific properties that must be present in the message for the adapter to retrieve it from the MSMQ queue.
 - a. From the drop-down menu, select one of the following **Message Filtering Modes**:
 - **MATCH_ONE** - Any one of the selected properties can be used to identify the message.
 - **MATCH_ALL** - All selected properties will be used to identify the message.

Note:

If none of the properties are selected, all messages will pass the filter criteria.

- b. Use the following icons to manage the rows in the **Property** column:
 - To define new property fields, select the  icon (or the  icon).

- To ensure that the parameters are parsed in the correct order, use the  or  icons to change the order of the property fields or the output field names.
- To delete any property fields or output field names, click the  icon.

You can add the following property fields:

Property	Property Type	Description
MessageId	java.lang.String	Identifies the message using hex string.
MessageIdByteArray	byte array	Identifies the message using an MSMQ-generated 20-byte message identifier.
CorrelationId	java.lang.String	Identifies the message using a hex string correlation identifier. For example, {F944A0B1-1DA0-4A93-A266-E54ACDD55342}\30775.
CorrelationIdByteArray	byte array	Identifies the message using a 20-bytes correlation identifier.
Delivery	java.lang.Integer	Specifies how Message Queuing delivers the message. Valid values are: <ul style="list-style-type: none"> ■ 1 - indicates that along its route, the message is stored locally on disk until it is forwarded to the next computer. ■ 0 - indicates that along its route, the message stays in volatile memory until it is received.
Priority	java.lang.Integer	Specifies the message priority. The value is between 0 - 7. The lower number indicates lower priority. Default: 3. <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: Message Queuing automatically sets the priority level of the transactional messages to 0, which indicates that the priority is ignored by the transaction.</p> </div>
AppSpecific	java.lang.Integer	Specifies application-generated single integer values.

9. From the **Output Message Property** tab, select which property fields the service will return after the service is executed.

You can add the following fields to the **Property** column:

Property	Property Type	Description
Ack	java.lang.Integer	Specifies the type of acknowledgment messages that Message Queuing will post in the administration queue when acknowledgments are requested.
AdminQueuePathName	java.lang.String	Specifies the queue path name used for Message Queuing-generated acknowledgment messages.
AdminQueueFormatName	java.lang.String	Specifies the queue format name used for Message Queuing-generated acknowledgment messages.
AppSpecific	java.lang.Integer	Specifies application-generated single integer values.
ArrivedTime	java.lang.String	Specifies the time when the message arrives in the queue.
CorrelationId	java.lang.String	Identifies the message using a hex string correlation identifier. For example, {F944A0B1-1DA0-4A93-A266-E54ACDD55342}\30775.
CorrelationIdByteArray	byte array	Identifies the message using a 20-byte correlation identifier.
Delivery	java.lang.Integer	Specifies how Message Queuing delivers the message. Valid values are: <ul style="list-style-type: none"> ■ 1 - indicates that along its route, the message is stored locally on disk until it is forwarded to the next computer. ■ 0 - indicates that along its route, the message stays in volatile memory until it is received.
DestinationQueuePathName	java.lang.String	Specifies the destination queue path name of the message
DestinationQueueFormatName	java.lang.String	Specifies the destination queue format name of the message.
MessageId	java.lang.String	Identifies the message using a hex string.
MessageIdByteArray	byte array	Identifies the message using an MSMQ-generated 20-byte message identifier.

Property	Property Type	Description
IsAuthenticated	java.lang.Integer	Indicates whether the message was authenticated at the request of the sending application. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is authenticated ■ 0 - the message is not authenticated
IsFirstInTransaction	java.lang.Integer	Indicates whether the message was the first message sent in its transaction. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is first ■ 0 - the message is not first
IsLastInTransaction	java.lang.Integer	Indicates whether the message was the last message sent in its transaction. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is last ■ 0 - the message is not last
Label	java.lang.String	Provides a description of the message.
Journal	java.lang.Integer	Specifies whether Message Queuing stores copies of the message as it is routed to the destination queue. Valid values are: <ul style="list-style-type: none"> ■ 0 - the journal is disabled ■ 1 - the message is stored in the applicable dead-letter queue on failure ■ 2 - a copy of the message is stored in the computer journal if the message was successfully delivered to the next computer
MaxTimeToReachQueue	java.lang.Integer	Specifies the time limit (in seconds) for a message to reach the queue.
MaxTimeToReceive	java.lang.Integer	Specifies the time limit (in seconds) for a message to be retrieved from the target queue.
MsgClass	java.lang.Integer	Indicates the message type.
Priority	java.lang.Integer	Specifies the message priority. The value is between 0 - 7. The lower number indicates lower priority. Default: 3.

Property	Property Type	Description
		Note: Message Queuing automatically sets the priority level of the transactional messages to 0, which indicates that the priority is ignored by the transaction.
SendTime	java.lang.String	Indicates when the message is sent.
SourceMachineGuid	java.lang.String	Indicates the GUID of the computer that sent the message.

- When using Designer, Designer contains only the **Adapter Settings** and **Input/Output** tabs. The information from the **Audit** and **Permissions** tabs appears in the **Properties** panel, and the information from the **Results** tab appears in the **Service Results** panel (in Designer). For more information, see the *webMethods Service Development Help* for your release.
- From the **File** menu, select **Save**.

Configuring Peek Services

A Peek service retrieves a copy of a message without removing the original message from a specified queue. You configure MSMQ Adapter services using Designer. The input and output signatures for a Peek service are located on the **Input/Output** tab of Designer.

When you configure a Peek service, the **Input** document is generated automatically in the input signature of the Peek service on the **Input/Output** tab. MSMQ Adapter passes the input fields from the **Filter** tab in Designer to the input signature.

The **Output** document also is generated automatically in the output signature of the Peek service on the **Input/Output** tab. The document contains the *Body* and the output fields from the **Output Message Properties** tab in Designer to the output signature.

When configuring a Peek service, you may want to consider using a non-transactional connection. Peek operations do not need to be transactional because they do not change information in the queue. Also, a Peek service internally does not use MSMQ transaction object.

For more information about Peek services, see [“Peek Service” on page 20](#).

➤ To configure a Peek service

- Review the steps in [“Before Configuring or Managing Adapter Services” on page 54](#).
- Start Designer.
- Using Designer, perform the following:

- a. Right-click the package in which the service should be contained and select **New > Adapter Service**.
 - b. Select the parent namespace and type a name for the adapter service.
 - c. Click **Next**.
4. Select **MSMQ Adapter** as the adapter type and click **Next**.
 5. For **Adapter Connection Name**, select a connection with an **inbound** queue access mode, and then click **Next**.

Important:

If you are using the adapter with Integration Server 8.0 SP1 or earlier, adapter services cannot use connections that are also used for adapter listeners.

6. From the list of available templates, select the **Peek** template and using Designer, click **Finish**.

The adapter service editor for the Peek service appears. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template**, as necessary.

7. Select the **Peek Service** tab to identify message processing features to use with this adapter service.

Specify values for the following optional fields:

Field	Description/Action
Timeout	Time period that the message queuing waits for a message to arrive. Enter this value in milliseconds.
Reset	Check to move the cursor to the front of the queue.

8. Select the **Filter** tab to specify how messages will be identified. The selected fields refer to the specific properties that must be present in the message for the adapter to retrieve it from the MSMQ queue.
 - a. From the drop-down menu, select one of the following **Message Filtering Modes**:
 - **MATCH_ONE** - Any one of the selected properties can be used to identify the message.
 - **MATCH_ALL** - All selected properties will be used to identify the message.

Note:

If none of the properties are selected, all messages will pass the filter criteria.

- b. Use the following icons to manage the rows in the **Property** column:

- To define new property fields, select the  icon (or the  icon).
- To ensure that the parameters are parsed in the correct order, use the  or  icons to change the order of the property fields or the output field names.
- To delete any property fields or output field names, click the  icon.

You can add the following property fields:

Property	Property Type	Description
MessageId	java.lang.String	Identifies the message using a hex string.
MessageIdByteArray	byte array	Identifies the message using an MSMQ-generated 20-byte message identifier.
CorrelationId	java.lang.String	Identifies the message using a hex string correlation identifier. For example, {F944A0B1-1DA0-4A93-A266-E54ACDD55342}\30775.
CorrelationIdByteArray	byte array	Identifies the message using a 20-byte correlation identifier.
Delivery	java.lang.Integer	Specifies how Message Queuing delivers the message. Valid values are: <ul style="list-style-type: none"> ■ 1 - indicates that along its route, the message is stored locally on disk until it is forwarded to the next computer. ■ 0 - indicates that along its route, the message stays in volatile memory until it is received.
Priority	java.lang.Integer	Specifies the message priority. The value is between 0 - 7. The lower number indicates lower priority. Default: 3. <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: Message Queuing automatically sets the priority level of the transactional messages to 0, which indicates that the priority is ignored by the transaction.</p> </div>
AppSpecific	java.lang.Integer	Specifies application-generated single integer values.

9. From the **Output Message Property** tab, select which property fields the service will return after the service is executed.

You can add the following fields to the **Property** column:

Property	Property Type	Description
Ack	java.lang.Integer	Specifies the type of acknowledgment messages that Message Queuing will post in the administration queue when acknowledgments are requested.
AdminQueuePathName	java.lang.String	Specifies the queue path name used for Message Queuing-generated acknowledgment messages.
AdminQueueFormatName	java.lang.String	Specifies the queue format name used for Message Queuing-generated acknowledgment messages.
AppSpecific	java.lang.Integer	Specifies application-generated single integer values.
ArrivedTime	java.lang.String	Specifies the time when the message arrives in the queue.
CorrelationId	java.lang.String	Identifies the message using a hex string correlation identifier. For example, {F944A0B1-1DA0-4A93-A266-E54ACDD55342}\30775.
CorrelationIdByteArray	byte array	Identifies the message using a 20-byte correlation identifier.
Delivery	java.lang.Integer	Specifies how Message Queuing delivers the message. Valid values are: <ul style="list-style-type: none"> ■ 1 - indicates that along its route, the message is stored locally on disk until it is forwarded to the next computer. ■ 0 - indicates that along its route, the message stays in volatile memory until it is received.
DestinationQueuePathName	java.lang.String	Specifies the destination queue path name of the message.
DestinationQueueFormatName	java.lang.String	Specifies the destination queue format name of the message.
MessageId	java.lang.String	Identifies the message using a hex string.
MessageIdByteArray	byte array	Identifies the message using an MSMQ-generated 20-byte message identifier.

Property	Property Type	Description
IsAuthenticated	java.lang.Integer	Indicates whether the message was authenticated at the request of the sending application. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is authenticated ■ 0 - the message is not authenticated
IsFirstInTransaction	java.lang.Integer	Indicates whether the message was the first message sent in its transaction. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is first ■ 0 - the message is not first
IsLastInTransaction	java.lang.Integer	Indicates whether the message was the last message sent in its transaction. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is last ■ 0 - the message is not last
Label	java.lang.String	Provides a description of the message.
Journal	java.lang.Integer	Specifies whether Message Queuing stores copies of the message as it is routed to the destination queue. Valid values are: <ul style="list-style-type: none"> ■ 0 - the journal is disabled ■ 1 - the message is stored in the applicable dead-letter queue on failure ■ 2 - a copy of the message is stored in the computer journal if the message was successfully delivered to the next computer.
MaxTimeToReachQueue	java.lang.Integer	Specifies the time limit (in seconds) for a message to reach the queue.
MaxTimeToReceive	java.lang.Integer	Specifies the time limit (in seconds) for the message to be retrieved from the target queue.
MsgClass	java.lang.Integer	Indicates the message type.
Priority	java.lang.Integer	Specifies the message priority. The value is between 0 - 7. The lower number indicates lower priority. Default: 3.

Property	Property Type	Description
		Note: Message Queuing automatically sets the priority level of the transactional messages to 0, which indicates that the priority is ignored by the transaction.
SendTime	java.lang.String	Indicates when the message is sent.
SourceMachineGuid	java.lang.String	Indicates the GUID of the computer that sent the message.

- When using Designer, Designer contains only the **Adapter Settings** and **Input/Output** tabs. The information from the **Audit** and **Permissions** tabs appears in the **Properties** panel, and the information from the **Results** tab appears in the **Service Results** panel (in Designer). For more information, see the *webMethods Service Development Help* for your release.
- From the **File** menu, select **Save**.

Using the Pipeline Editor to View and Set the Body

Depending on the service, you can use the Get, Put, or Peek services to map variables to either the Input signature or the Output signature.

> To use the Pipeline Editor to view and set the Body

- Configure a new flow service or open an existing one using Designer.
- Insert an invoke step  that calls the adapter service.
- In each of the following services:
 - Put - use the Pipeline Editor to map a variable that is in the pipeline to the *Body* variable in the Input Signature of the Put service.
 - Get, Peek - use the Pipeline Editor to map the *Body* variable in the Output Signature of the Get or Peek service to a variable that is in the Pipeline Editor.

Testing Adapter Services

You use Designer to test adapter services.

For more information about testing and debugging services, see the *webMethods Service Development Help* for your release.

> To test adapter services

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 54.
2. In Designer, expand the package and folder that contain the service you want to test.
3. Double-click the service you want to test.
Designer displays the configured service in the service template's Adapter Service Editor.
4. Using Designer, select **Run > Run As > Run Service**.
5. For every service input field, you will be prompted to enter an input value. Enter a value for each input field and then click **OK**.
6. Click the **Service Result** tab (in Designer) to view the output from this service.

Viewing Adapter Services

You use Designer to view adapter services.

> To view a service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 54.
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 a service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 54.
2. In Designer, expand the package and folder that contain the service you want to delete.
3. Right-click the service you want to delete and then select **Delete**.

Enabling Automatic Data Validation

Designer enables MSMQ Adapter to validate user-defined data for adapter services at design time. You can validate the values for a single adapter service or you can configure Designer to always validate the values for adapter services. Both options could potentially slow your design-time operations.

For more information about the **Adapter Service/Notification Editor** and other Designer menu options and toolbar icons, see the *webMethods Service Development Help* for your release.

Enabling Data Validation for a Single Adapter Service

When you enable data validation for a single adapter service, Designer compares the service values against the resource data that has already been fetched from the selected adapter.

> To enable automatic data validation for a single adapter service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 54.
2. In Designer, expand the package and folder that contain the service for which you want to enable automatic validation.
3. Double-click the service for which you want to validate the data.

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

4. Click the  icon.

Enabling Validation for All Adapter Services

If you select the option to always validate values for adapter services, it will do so for all webMethods WmART-based adapters installed on Integration Server.

> To always validate the values for all adapter services

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 54.
2. Start Designer.
3. Using Designer, select the **Window > Preferences > Software AG > Service Development > Adapter Service/Notification Editor** item.
4. Enable the **Automatic data validation** option.
5. Click **OK**.

Reloading Adapter Values

You can enable MSMQ Adapter to reload and validate user-defined data for adapter services at design time in Designer. You can reload values for a single adapter service or you can configure Designer so it automatically reloads the values for adapter services. Both options could potentially slow your design-time operations.

For more information about the **Adapter Service/Notification Editor** and other menu options and toolbar icons, see the *webMethods Service Development Help* for your release.

Reloading Adapter Values for a Single Adapter Service

When you reload adapter values for a single adapter service, Designer compares the service values against the resource data that has already been fetched from the selected adapter.

> To reload the adapter values for a single adapter service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 54.
2. In Designer, expand the package and folder that contain the service for which you want to enable automatic validation.
3. Double-click the service for which you want to validate the data.

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

4. Click the  icon.

Reloading Adapter Values for All Adapter Services

If you select the option to always reload values for adapter services, it will do so for all webMethods WmART-based adapters installed on Integration Server.

> To reload the adapter values for all adapter services

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 54.
2. Start Designer.
3. Using Designer, select the **Window > Preferences > Software AG > Service Development > Adapter Service/Notification Editor** item.
4. Enable the **Automatic polling of adapter metadata** option.
5. Click **OK**.

6 Adapter Notifications

■ Overview	76
■ Listeners	76
■ Listener Notifications	84

Overview

This chapter describes how to configure and manage listeners and listener notifications. For a detailed description of MSMQ Adapter listeners, see [“Listeners” on page 22](#). For a detailed description of the available MSMQ Adapter listener notifications, see [“Listener Notifications” on page 22](#).

Listeners

Before you Configure New Listeners

Perform the following tasks before configuring new listeners.

➤ **To prepare to configure a new listener**

1. Make sure that you have webMethods Integration Server administrator privileges so that you can access MSMQ Adapter administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator’s Guide* for your release.
2. Start Integration Server and Integration Server Administrator, if they are not already running.
3. Using Integration Server Administrator, make sure that the WmMSMQAdapter package is enabled. To verify the status of the WmMSMQAdapter package, see [“Enabling Packages” on page 35](#).
4. Using Software AG Designer, create a user-defined package to contain the listener, if you have not already done so. For more information about managing packages, see [“Package Management” on page 33](#) for details.

Note:

If you are using Designer, use the Service Development perspective. For more information, see the *webMethods Service Development Help* for your release.

Configuring New Listeners

When you configure MSMQ Adapter listeners, you specify information that Integration Server uses to enable listeners to listen for, and get messages on, a specified queue.

You configure MSMQ Adapter listeners using Integration Server Administrator.

➤ **To configure a new listener**

1. In the **Adapters** menu in Integration Server Administrator navigation area, click **MSMQ Adapter**.

2. In the **MSMQ Adapter** menu, click **Listeners**.
3. On the Listeners screen, click **Configure New Listener**.
4. On the Listeners Type screen, click **MSMQ Adapter**.
5. On the Configure Listener Type screen, in the **MSMQ Adapter** section, complete the following fields:

Parameter	Description/Action
Package	<p>The package in which to create the listener.</p> <p>Create the listener in a user-defined package rather than in the adapter's package. For other important considerations when creating packages for MSMQ Adapter, see “Package Management” on page 33.</p> <p>You must create the package using Designer before you can specify it using this parameter. For general information about creating packages, see the <i>webMethods Service Development Help</i> for your release.</p>
Folder Name	The folder in which to create the listener.
Listener Name	The name of the listener you are configuring.
Connection Name	<p>The connection to associate with the listener.</p> <p>From the list of connections, select a connection with a Queue Access Mode set to Inbound.</p> <p>Listeners exhibit different behaviors when they are associated with a transactional connection than with non-transactional connection. For more information about the differences, see “Listener Transactionality and Behavior” on page 24.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Important: If you are using the adapter with Integration Server 8.0 SP1 or earlier, listeners cannot share connections with adapter services. However, listeners can share connections with other listeners.</p> </div>
Retry Limit	The number of times the adapter tries to reconnect if the adapter fails to connect or loses connection with the message queuing computer. Default: 5.
Retry Backoff Timeout	The number of seconds that elapse between each of the retries specified in Retry Limit . Default: 10.

6. In the **Listener Properties** section, complete the following fields:

Parameter	Description/Action
Timeout	Time that the message queuing will wait for a message to arrive. Enter this value in milliseconds. Default: 20000
Reset Before Operation	Select one of the following: <ul style="list-style-type: none">■ true - Move the cursor to the front of the queue.■ false - Do not move the cursor.

7. Click **Save Listener**.

Testing Listeners

In general, you can test listeners to ensure that you have configured them correctly.

> To test a listener

1. Configure a listener using Integration Server Administrator. For instructions on how to configure a listener, see [“Configuring New Listeners” on page 76](#).
2. Configure a listener notification using Designer. For instructions on how to configure a notification, see [“Configuring Listener Notifications” on page 86](#).
3. Enable the listener notification using Integration Server Administrator. For instructions on how to enable a listener notification, see [“Enabling Listener Notifications” on page 101](#).
4. Enable the listener using Integration Server Administrator. For instructions on how to enable a listener, see [“Enabling Listeners” on page 83](#).
5. Make a change in your back-end system to activate the listener. For example, put a message on a queue.

The listener searches for, filters, and returns messages that match the filter criteria that you specified when you configured the listener on the Configure Listener Type screen. Filtered messages are then sent to the listener notification for further processing.

For more information on how listeners and listener notifications work at run time, refer to the following sections: [“Run-Time Processing of Listeners and Notifications” on page 25](#) and [“Run-Time Processing of Listeners and Synchronous Listener Notifications” on page 26](#).

Viewing Listener Parameters

You can view a listener's parameters from Integration Server Administrator or from Designer. You also can view the notification order of a listener.

Viewing Listener Parameters Using Integration Server Administrator

Perform the following steps to view listener parameters using Integration Server Administrator.

➤ To view a listener's parameters using Integration Server Administrator

1. In the **Adapters** menu in the navigation area of Integration Server Administrator, click **MSMQ Adapter**.
2. In the **MSMQ Adapter** menu, click **Listeners**.

When using the adapter with Integration Server 8.0 and later, you can sort and filter the list of listeners.

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

You can control the number of listeners that are displayed on this screen. For more information, see [“Controlling Pagination” on page 28](#).

3. On the **Listeners** page, click the  icon for the listener that you want to see.

The View Listener screen displays the parameters for the listener. For descriptions of the listener parameters, see [“Configuring New Listeners” on page 76](#).

4. Click **Return to MSMQ Adapter Listeners** to return to the main listeners screen.

Viewing Listener Parameters Using Designer

Perform the following steps to view listener parameters using Designer.

> To view a listener's parameters using Designer

1. Review the steps in [“Before you Configure New Listeners” on page 76](#).
2. From Designer, expand the package and folder in which the listener is located.
3. Double-click the listener you want to view.

The parameters for the listener appear on the **Listener Information** tab. For descriptions of the listener properties, see [“Configuring New Listeners” on page 76](#).

Viewing the Notification Order of a Listener

Perform the following steps to view the notification order of a listener.

> To view the notification order of a listener

1. In the **Adapters** menu in the navigation area of the administrator, click **MSMQ Adapter**.
2. In the **MSMQ Adapter** menu, click **Listeners**.
3. On the Listeners screen, click the  icon for the listener that you want to view.

The View Notification Order screen displays the order of the notifications for the listener. To change the notification order for the listener, see [“Editing the Notification Order” on page 81](#).

4. Click **Return to wMMQAdapter** to return to the Edit Listener screen.

Editing Listeners

You use Integration Server Administrator to edit the listener in the following situations:

- If you need to select a newly configured connection, or if you need to change any listener properties, such as filter values, you can update the listener parameters.
- If you need to change the order of the notifications that are associated with the listener, see [“Editing the Notification Order” on page 81](#).

> To edit a listener

1. In the Adapters menu in the navigation area of Integration Server Administrator, click **MSMQ Adapter**.
2. In the **MSMQ Adapter** menu, click **Listeners**.

3. On the Listeners screen, make sure that the listener is disabled before editing. To disable the listener, see [“Disabling Listeners” on page 84](#).

4. On the Listeners screen, click the  icon for the listener that you want to edit.

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

For descriptions of the listener parameters, see [“Configuring New Listeners” on page 76](#).

5. Click **Save Changes** to save the listener and return to the Listeners screen.

Editing the Notification Order

➤ To edit the notification order of a listener

1. In the **Adapters** menu in the navigation area of the administrator, click **MSMQ Adapter**.
2. In the **MSMQ Adapter** menu, click **Listeners**.
3. On the Listeners screen, make sure that the listener is disabled before editing. To disable the listener, see [“Disabling Listeners” on page 84](#) for details.
4. On the Listeners screen, click the  icon for the listener that you want to edit.
5. On the Edit Listener screen, click **Edit Notification Order**.
6. On the Edit Notification Order screen, use the **Up** and **Down** buttons to determine the processing order in which MSMQ Adapter invokes the notifications.

Note:

For better processing results, arrange your notifications from ascending to descending order starting with the most detailed notifications to the least detailed notifications. For more information on notifications and their filter criteria, see [“Consideration for Listener Notifications” on page 85](#).

7. Click **Save Changes** to save the notification order of the listener.
8. Click **Return to wMMQAdapter** to return to the Edit Listener screen.

Copying Listeners

You can copy an existing listener to create a new listener with the same or similar properties without having to type or specify all properties for the listener. You copy adapter listeners using Integration Server Administrator.

> To copy a listener

1. In the **Adapters** menu in the navigation area of Integration Server Administrator, click **MSMQ Adapter**.
2. In the **MSMQ Adapter** menu, click **Listeners**.
3. On the Listeners screen, click the  icon for the listener that you want to copy.

The Copy Listener screen displays the current parameters for the listener that you want to copy. Name the new listener and edit any listener parameters as needed by typing or selecting the values you want to specify.

For descriptions of the listener parameters, see [“Configuring New Listeners” on page 76](#).

4. Click **Save Changes** to save the listener and return to the Listeners screen.

Deleting Listeners

If you no longer want to use a listener, use the following instructions to delete the listener. You use Integration Server Administrator to delete listeners.

Important:

If you delete an MSMQ Adapter listener, any notifications that are defined to use the listener will no longer work. You cannot change which listener a notification uses after the notification is configured. However, you can change the parameters for an existing listener. For instructions, see [“Editing Listeners” on page 80](#).

> To delete a listener

1. In the **Adapters** menu in the navigation area of Integration Server Administrator, click **MSMQ Adapter**.
2. In the **MSMQ Adapter** menu, click **Listeners**.
3. On the Listeners screen, make sure that the listener is disabled before deleting it. To disable the listener, see [“Disabling Listeners” on page 84](#) for details.
4. On the Listeners screen, click  for the listener you want to delete.

Integration Server deletes the listener.

Enabling Listeners

Before you enable a listener, you need to configure one or more notifications to associate with the listener. If no notifications have been configured when you enable the listener, Integration Server Administrator displays the following warning message: "No notifications for listener."

After you have configured notifications, you must enable the listener so that the associated notifications will communicate appropriately with the listener at run time. You enable the listeners using Integration Server Administrator.

The **Status** column indicates the readiness of the listener. When the status is **Succeeded**, the listener is ready to be enabled. When the status is **Failed**, check the following:

- The queue access mode of the connection. It should be set to inbound.
- The order in which you disabled the listener and notification. You must disable the listener before disabling the notification.

For more information on configuring listeners and notifications, see the sections "[Configuring New Listeners](#)" on page 76 and "[Configuring Listener Notifications](#)" on page 86.

Important:

Before you enable a listener, you must enable the listener notification. For instructions on how to enable a listener notification, see "[Enabling Listener Notifications](#)" on page 101.

Note:

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

> To enable a listener

1. In the **Adapters** menu in the navigation area of Integration Server Administrator, click **MSMQ Adapter**.
2. In the **MSMQ Adapter** menu, click **Listeners**.
3. The Listeners screen appears. Select **Enabled** from the drop-down list in the **State** field. Integration Server Administrator enables the listener.

The **Enable all suspended** link helps you change the state quickly for multiple listeners.

Suspending Listeners

You can use Integration Server to suspend listeners.

When you suspend a listener, the action may not take effect right away. You may have to wait as long as the time specified in the Timeout parameter for the listener. If one or more messages appear on the queue within that time interval, the adapter may receive and process the first message.

Note:

In MSMQ Adapter, a suspended listener behaves identically to a disabled listener.

> To suspend a listener

1. In the **Adapters** menu in the navigation area of Integration Server Administrator, click **MSMQ Adapter**.
2. In the **MSMQ Adapter** menu, click **Listeners**.
3. On the Listeners screen, select **Suspended** from the drop-down list in the **State** field. Integration Server Administrator suspends the listener.

The **Suspend all enabled** link helps you change the state quickly for multiple listeners.

Disabling Listeners

Listeners must be disabled before you can edit or delete them. You disable listeners using Integration Server Administrator.

When you disable a listener, the action may not take effect right away. You may have to wait as long as the time specified in the Timeout parameter for the listener. If one or more messages appear on the queue within that time interval, the adapter may receive and process the first message.

> To disable a listener

1. In the **Adapters** menu in the navigation area of Integration Server Administrator, click **MSMQ Adapter**.
2. In the **MSMQ Adapter** menu, click **Listeners**.
3. The Listeners screen appears. Select **Disabled** from the drop-down list in the **State** field. Integration Server Administrator disables the listener.

Listener Notifications

The following sections provide instructions for configuring and managing MSMQ Adapter listener notifications. MSMQ Adapter has two types of listener notifications that you can configure:

- Asynchronous Listener Notifications
- Synchronous Listener Notifications

For more information on how listener notifications work, see [“Listener Notifications” on page 22](#).

Before You Configure Listener Notifications

Perform the following tasks before configuring listener notifications.

➤ To prepare to configure a listener notification

1. Install Integration Server and MSMQ Adapter on the same machine. For details, see [“Installing, Upgrading, and Uninstalling the MSMQ Adapter”](#) on page 29.
2. Make sure that you have Integration Server administrator privileges so that you can access MSMQ Adapter administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator’s Guide* for your release.
3. Start Integration Server and Integration Server Administrator, if they are not already running.
4. Using Integration Server Administrator, make sure that the WmMSMQAdapter package is enabled. To verify the status of the WmMSMQAdapter package, see [“Enabling Packages”](#) on page 35.
5. Configure a listener using Integration Server Administrator. For more information on how to configure a new listener, see [“Configuring New Listeners”](#) on page 76.
6. Using Designer, create a user-defined package to contain the listener, if you have not already done so. For more information about managing packages, see [“Package Management”](#) on page 33.

Consideration for Listener Notifications

For each listener, configure a default notification that processes all messages that do not pass any notification filter criteria. From Integration Server Administrator, edit the listener's notification order list and place the default notification last. For instructions on how to configure a notification, see [“Configuring Listener Notifications”](#) on page 86. For instructions on how to edit the notification order list, see [“Editing Listeners”](#) on page 80.

Using Asynchronous and Synchronous Listener Notifications

The following table lists the tasks required to use these types of notifications:

Task	Use this tool...	
1	Configure an adapter connection. For details, see “Configuring Adapter Connections” on page 44.	Integration Server Administrator
2	Select the appropriate notification template and configure the notification.	Designer

Task	Use this tool...
<p>For instructions to configure notifications, see “Configuring Listener Notifications” on page 86.</p>	
<p>3 If you plan to use an Integration Server flow or Java Designer service, design it to react to the data changes contained in the notification's publishable document. Configure an Integration Server trigger to use the notification's publishable document.</p>	Designer
<p>For instructions on configuring notifications, see the <i>webMethods Service Development Help</i> for your release.</p>	
<p>Note: To process the message from the queue, you should create an Integration Server trigger that subscribes to the Integration Server document type that MSMQ Adapter created with the asynchronous notification. For more information about using triggers, see the <i>Publish-Subscribe Developer's Guide</i> for your release.</p>	
<p>4 Enable the adapter notifications. For instructions to enable listener notifications, see “Enabling Listener Notifications” on page 101.</p>	Integration Server Administrator

Configuring Listener Notifications

MSMQ Adapter has two types of listener notifications that you can configure:

- Asynchronous Listener Notifications
- Synchronous Listener Notifications

For a description of the asynchronous and synchronous listener notifications, see [“Listener Notifications” on page 22.](#) You configure listener notifications using Designer.

Configuring Asynchronous Listener Notifications

Perform the following tasks to configure asynchronous listener notifications.

➤ To configure an MSMQ asynchronous listener notification

1. Review the steps in [“Before You Configure Listener Notifications” on page 85.](#)
2. Start Designer.

3. Using Designer, perform the following:
 - a. Right-click the package in which the notification should be contained and select **New > Adapter Notification**.
 - b. Select the parent namespace and type a name for the adapter notification.
 - c. Click **Next**.
4. Select **MSMQ Adapter** as the adapter type and click **Next**.
5. Select **Asynchronous Listener Notification** from the template and click **Next**.
6. Select the appropriate **Notification Listener Name** and click **Next**.
7. Select a package and folder to contain the notification, type a unique name for the notification, and click **Next**.
8. Click **Finish**.
9. Designer creates an asynchronous listener notification and an asynchronous publishable document type, and the editor for the adapter notification appears.
 - a. You can select the **Adapter Settings** tab at any time to confirm listener notification properties such as the **Adapter Name**, **Adapter Listener Name**, and **Adapter Notification Template** as necessary.
 - b. In the **Publish Document** section, you can specify how you want the notification document to be published:
 - To publish documents to Broker, select **webMethods Broker /Local**. This is the default option.
 - To publish documents to a JMS provider, select **JMS Provider**, and provide values for the following input fields:

Field	Description/Action
Connection alias name	The name of the JMS connection alias configured on Integration Server.
Destination name	The name of the destination from which you want the JMS trigger to receive messages.
Destination type	Whether the destination is a Queue (default) or a Topic .

10. Select the **Filter** tab to specify which property values should match the arriving message to run the notification.

a. From the drop-down menu, select one of the following **Message Filtering Modes**:

- **MATCH_ONE** - Any one of the selected properties can be used to identify the message.
- **MATCH_ALL** - All selected properties will be used to identify the message.

Note:

If none of the properties are selected, all messages will pass the filter criteria.

b. Select the property fields and enter the values for the properties. Use the following icons to manage the rows in the **Property** column:

- To define new property fields, select the  icon (or the  icon).
- To change the order of the property fields to ensure that the parameters are parsed in the correct order, use the  or  icons.
- To delete a property field, use the  icon.

You can add the following property fields to the **Property** column.

Property	Description
MessageId	Identifies the message using a hex string.
CorrelationId	Identifies the message using a hex string correlation identifier. For example, {F944A0B1-1DA0-4A93-A266-E54ACDD55342} \ 30775.
Delivery	Specifies how Message Queuing delivers the message. Valid values are: <ul style="list-style-type: none"> ■ 1 - indicates that along its route, the message is stored locally on disk until it is forwarded to the next computer. ■ 0 - indicates that along its route, the message stays in volatile memory until it is received.
Priority	Specifies the message priority. The value is between 0 - 7. The lower number indicates lower priority. Default: 3.

Note:

Message Queuing automatically sets the priority level of the transactional messages to 0, which indicates that the priority is ignored by the transaction.

Property	Description
AppSpecific	Specifies application-generated single integer values.

11. Select the **Output Message Property** tab to identify the message property fields that will be generated in the notification publishable document.

You can add the following property fields:

Property	Property Type	Description
Ack	java.lang.Integer	Specifies the type of acknowledgment messages that Message Queuing will post in the administration queue when acknowledgments are requested.
AdminQueuePathName	java.lang.String	Specifies the queue path name used for Message Queuing-generated acknowledgment messages.
AdminQueueFormatName	java.lang.String	Specifies the queue format name used for Message Queuing-generated acknowledgment messages.
AppSpecific	java.lang.Integer	Specifies application-generated single integer values.
ArrivedTime	java.lang.String	Specifies the time when the message arrives in the queue.
CorrelationId	java.lang.String	Identifies the message using a hex string correlation identifier. For example, {F944A0B1-1DA0-4A93-A266-E54ACDD55342}\30775.
CorrelationIdByteArray	byte array	Identifies the message using a 20-byte correlation identifier.
Delivery	java.lang.Integer	Specifies how Message Queuing delivers the message. Valid values are: <ul style="list-style-type: none"> ■ 1 - indicates that along its route, the message is stored locally on disk until it is forwarded to the next computer. ■ 0 - indicates that along its route, the message stays in volatile memory until it is received.
DestinationQueuePathName	java.lang.String	Specifies the destination queue path name of the message

Property	Property Type	Description
DestinationQueueFormatName	java.lang.String	Specifies the destination queue format name of the message.
MessageId	java.lang.String	Identifies the message using a hex string.
MessageIdByteArray	byte array	Identifies the message using an MSMQ-generated 20-byte message identifier.
IsAuthenticated	java.lang.Integer	Indicates whether the message was authenticated at the request of the sending application. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is authenticated. ■ 0 - the message is not authenticated.
IsFirstInTransaction	java.lang.Integer	Indicates whether the message was the first message sent in its transaction. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is first. ■ 0 - the message is not first.
IsLastInTransaction	java.lang.Integer	Indicates whether the message was the last message sent in its transaction. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is last. ■ 0 - the message is not last.
Label	java.lang.String	Provides a description of the message.
Journal	java.lang.Integer	Specifies whether Message Queuing stores copies of the message as it is routed to the destination queue. Valid values are: <ul style="list-style-type: none"> ■ 0 - the journal is disabled. ■ 1 - the message is stored in the applicable dead-letter queue on failure. ■ 2 - a copy of the message is stored in the computer journal if the message was successfully delivered to the next computer.
MaxTimeToReachQueue	java.lang.Integer	Specifies the time limit (in seconds) for a message to reach the queue.

Property	Property Type	Description
MaxTimeToReceive	java.lang.Integer	Specifies the time limit (in seconds) for the message to be retrieved from the target queue.
MsgClass	java.lang.Integer	Indicates the message type.
Priority	java.lang.Integer	Specifies the message priority. The value is between 0 - 7. The lower number indicates lower priority. Default: 3.
		<p>Note: Message Queuing automatically sets the priority level of the transactional messages to 0, which indicates that the priority is ignored by the transaction.</p>
SendTime	java.lang.String	Indicates when the message is sent.
SourceMachineGuid	java.lang.String	Indicates the GUID of the computer that sent the message.

12. Select the **Permissions** tab to manage the access control list (ACL) information. Use the drop-down menu to select each of the ACL types. For general information about assigning and managing ACLs, see the *webMethods Service Development Help* for your release.
13. From the **File** menu, select **Save**.

Configuring Synchronous Listener Notifications

Perform the following tasks to configure synchronous listener notifications.

➤ To configure an MSMQ synchronous listener notification

1. Review the steps in [“Before You Configure Listener Notifications” on page 85](#).
2. Start Designer.
3. Using Designer, perform the following:
 - a. Right-click the package in which the notification should be contained and select **New > Adapter Notification**.
 - b. Select the parent namespace and type a name for the adapter notification.
 - c. Click **Next**.

4. Select **MSMQ Adapter** as the adapter type and click **Next**.
5. Select **Synchronous Listener Notification** from the template and click **Next**.
6. Select the appropriate **Notification Listener Name** and click **Next**.
7. Select the appropriate **Service Node** and click **Next**.
8. Select a package and folder to contain the notification, type a unique name for the notification, and click **Next**.
9. Click **Finish**.

The Adapter Notification template creates the following items:

- A Synchronous Listener Notification
- Two Synchronous Document Types:
 - Synchronous Reply Document Type
 - Synchronous Request Document Type

Note:

You cannot edit any fields or properties on the **Publications Properties** tab for the Synchronous Request and Reply Document Types. Integration Server does not publish these document types.

10. Select the **Adapter Settings** tab, to view the adapter properties and to select the appropriate service name for the synchronous notification.
 - a. The following fields in the **Adapter Properties** section are read-only:

Adapter Properties	Description
Adapter Name	The name of the adapter.
Adapter Listener Name	The name of the adapter listener that you selected when you configured the synchronous notification.
Adapter Notification Template	The name of the adapter notification template that you selected when you configured the synchronous notification.

- b. Specify one of the following modes in the **Execution Mode** section:
 - To invoke a flow service directly, select **Service Invoke** and specify the following field:

Property	Description
Service Name	<p>The name of the service, which you selected when you initially configured the adapter listener notification, appears in the Service Name field.</p> <p>Select a different service by clicking the  icon to the right of the Service Name field.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Important: Before you select a different service, make sure that you have disabled the notification. When the notification is enabled, then the new service is utilized. For more information on enabling and disabling listeners, see “Enabling Listeners” on page 83 and “Disabling Listeners” on page 84.</p> </div>

- To publish documents locally or to a Broker and wait for a reply, select **Publish and Wait** and specify the following fields:

Properties	Description
Local	<p>Select true to publish documents locally (to this Integration Server only) or false to publish to the Broker connected to this Integration Server.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: If no Broker is configured for Integration Server, documents will be published locally.</p> </div>
Wait Time	<p>Specify the number of milliseconds to wait for a reply. The default is -1, which means to wait indefinitely.</p>

11. Select the **Filter** tab to specify which property values should match the arriving message to run the notification.
 - a. From the drop-down menu, select one of the following **Message Filtering Modes**:
 - **MATCH_ONE** - Any one of the selected properties can be used to identify the message.
 - **MATCH_ALL** - All selected properties will be used to identify the message.

Note:
If none of the properties are selected, all messages will pass the filter criteria.
 - b. Select the property fields and enter the values for the properties. Use the following icons to manage the rows in the **Property** column:

- To define a new property, select the  icon (or the  icon).
- To change the order of the properties to ensure that the parameters are parsed in the correct order, use the  or  icons.
- To delete a property, use the  icon.

You can add the following property fields:

Property	Description
MessageId	Identifies the message using a hex string.
CorrelationId	Identifies the message using a hex string correlation identifier. For example, {F944A0B1-1DA0-4A93-A266-E54ACDD55342} \ 30775.
Delivery	Specifies how Message Queuing delivers the message. Valid values are: <ul style="list-style-type: none"> ■ 1 - indicates that along its route, the message is stored locally on disk until it is forwarded to the next computer. ■ 0 - indicates that along its route, the message stays in volatile memory until it is received.
Priority	Specifies the message priority. The value is between 0 - 7. The lower number indicates lower priority. Default: 3. <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: Message Queuing automatically sets the priority level of the transactional messages to 0, which indicates that the priority is ignored by the transaction.</p> </div>
AppSpecific	Specifies application-generated single integer values.

12. Select the **Output Message Property** tab to identify the message property fields that will be generated in the notification request document.

You can add the following property fields:

Property	Property Type	Description
Ack	java.lang.Integer	Specifies the type of acknowledgment messages that Message Queuing will post in administration queue when acknowledgments are requested.

Property	Property Type	Description
AdminQueuePathName	java.lang.String	Specifies the queue path name used for Message Queuing-generated acknowledgment messages.
AdminQueueFormatName	java.lang.String	Specifies the queue format name used for Message Queuing-generated acknowledgment messages.
AppSpecific	java.lang.Integer	Specifies application-generated single integer values.
ArrivedTime	java.lang.String	Specifies the time when the message arrives in the queue.
CorrelationId	java.lang.String	Identifies the message using a hex string correlation identifier. For example, {F944A0B1-1DA0-4A93-A266-E54ACDD55342} \ 30775.
CorrelationIdByteArray	byte array	Identifies the message using a 20-bytes correlation identifier.
Delivery	java.lang.Integer	Specifies how Message Queuing delivers the message. Valid values are: <ul style="list-style-type: none"> ■ 1 - indicates that along its route, the message is stored locally on disk until it is forwarded to the next computer. ■ 0 - indicates that along its route, the message stays in volatile memory until it is received.
DestinationQueuePathName	java.lang.String	Specifies the destination queue path name of the message
DestinationQueueFormatName	java.lang.String	Specifies the destination queue format name of the message.
MessageId	java.lang.String	Identifies the message using hex string.
MessageIdByteArray	byte array	Identifies the message using an MSMQ-generated 20-byte message identifier.
IsAuthenticated	java.lang.Integer	Indicates whether the message was authenticated at the request of the sending application. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is authenticated. ■ 0 - the message is not authenticated.

Property	Property Type	Description
IsFirstInTransaction	java.lang.Integer	Indicates whether the message was the first message sent in its transaction. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is first. ■ 0 - the message is not first.
IsLastInTransaction	java.lang.Integer	Indicates whether the message was the last message sent in its transaction. Valid values are: <ul style="list-style-type: none"> ■ 1 - the message is last. ■ 0 - the message is not last.
Label	java.lang.String	Provides a description of the message.
Journal	java.lang.Integer	Specifies whether Message Queuing stores copies of the message as it is routed to the destination queue. Valid values are: <ul style="list-style-type: none"> ■ 0 - the journal is disabled. ■ 1 - the message is stored in the applicable dead-letter queue on failure. ■ 2 - a copy of the message is stored in the computer journal if the message was successfully delivered to the next computer.
MaxTimeToReachQueue	java.lang.Integer	Specifies the time limit (in seconds) for a message to reach the queue.
MaxTimeToReceive	java.lang.Integer	Specifies the time limit (in seconds) for the message to be retrieved from the target queue.
MsgClass	java.lang.Integer	Indicates the message type.
Priority	java.lang.Integer	Specifies the message priority. The value is between 0 - 7. The lower number indicates lower priority. Default: 3.
		Note: Message Queuing automatically sets the priority level of the transactional messages to 0, which indicates that the priority is ignored by the transaction.
SendTime	java.lang.String	Indicates when the message is sent.

Property	Property Type	Description
SourceMachineGuid	java.lang.String	The property provides the GUID of the computer that sent the message.

13. Select the **Permissions** tab to manage the access control list (ACL) information. Use the drop-down menu to select each of the ACL types. For general information about assigning and managing ACLs, see the *webMethods Service Development Help* for your release.
14. From the **File** menu, select **Save**.

Editing Synchronous Listener Notification Services

The listener notification service is specified during the initial configuration of the synchronous listener notification. Afterwards you may need to change the service that is invoked. To change the notification service, complete the following steps.

Important:

Before you select a different service, make sure that you have disabled the notification. When the notification is enabled, the new service is utilized. To disable the notification, see [“Disabling Listener Notifications” on page 101](#).

➤ To edit the notification service

1. From Designer, in the adapter notification service editor, select the **Adapter Settings** tab.
2. Click the **Browse** button next to **Service Name**. The **Select Menu** displays.
3. Navigate and select the new service.
4. Click **OK**.
5. From the **File** menu, select **Save**.

Behavior of Listener Notifications

MSMQ Adapter handles messages depending on how you have configured listeners and listener notifications in your system. For example, messages behave differently when the following combinations are set up in your MSMQ Adapter:

- Transactional or non-transactional listeners

The transactionality of a listener depends on the transactionality of the connection. For example, when you associate a non-transactional connection with a listener, the listener is non-transactional.

- Transactional or non-transactional asynchronous notifications

The transactionality of an asynchronous notification depends on the transactionality of a listener. For example, when you associate a non-transactional listener with an asynchronous notification, the asynchronous notification is non-transactional.

- Transactional or non-transactional synchronous notifications

The transactionality of a synchronous notification depends on the transactionality of a listener. For example, when you associate a non-transactional listener with a synchronous notification, the synchronous notification is non-transactional.

Testing Listener Notifications

You can test listener notifications to ensure that you have configured them correctly.

➤ To test listener notifications

1. Configure a listener using Integration Server Administrator. For instructions to configure a listener, see [“Configuring New Listeners” on page 76](#).
2. Configure a listener notification using Designer. For instructions to configure a notification, see [“Configuring Listener Notifications” on page 86](#).
3. Enable the listener notification using Integration Server Administrator. For instructions to enable a listener notification, see [“Enabling Listener Notifications” on page 101](#).
4. Enable the listener using Integration Server Administrator. For instructions to enable a listener, see [“Enabling Listeners” on page 83](#). Make a change in your back-end system to activate the listener. For example, put a message on a queue.

The listener searches for, filters, and returns messages that match the filter criteria that you specified when you configured the listener on the Configure Listener Type screen. Filtered messages then are sent to the listener notification for further processing.

For more information on how listeners and listener notifications work at run time, refer to the following sections: [“Run-Time Processing of Listeners and Notifications” on page 25](#) and [“Run-Time Processing of Listeners and Synchronous Listener Notifications” on page 26](#).

Testing Publishable Document Types

You can test a publishable document type that is associated with an asynchronous notification running it in Designer. When you test a publishable document type, you provide input values that the Designer uses to create an instance of the publishable document type. You also specify a publishing method (such as publish, publish and wait, deliver, or deliver and wait). Designer then publishes a document and displays the results of the publish in the **Results** dialog box. Testing a publishable document type provides a way for you to publish a document without building a service that does the actual publishing. If you select a publication action where you wait for a reply document, you can verify whether reply documents are received.

Note:

Prior to running the **PublishDocument**, you need to make sure that you uncheck the **Field must exist at run-time** field on the **msgBody** property. To access this field, right-click **msgBody**, select **Properties**, and then select the **Constraints** tab. When you test a publishable document type, Integration Server actually publishes the document locally or to the Broker (whichever is specified).

For instructions to test a publishable document type, see the *Publish-Subscribe Developer's Guide* for your release. Also, for a complete description of the envelope parameters located in the `WmPublic` folder, see the *webMethods Integration Server Built-In Services Reference* for your release. The envelope parameters define the sender's address, the time the document was sent, password and certificate information, and other useful information for routing and control.

Viewing Listener Notifications

You use the Designer to view listener notifications.

➤ To view a listener notification

1. Review the steps in [“Before You Configure Listener Notifications” on page 85](#).
2. In Designer, expand the package and folder that contain the listener notification you want to view.
3. Select the listener notification that you want to view.

Designer displays the configured listener notification in the adapter's Adapter Notification Editor.

Editing Listener Notifications

You use Designer to edit both synchronous and asynchronous listener notifications. You also use Designer to edit the publishable document type associated with the asynchronous listener notifications. However, you cannot edit the request and reply document types that are associated with synchronous listener notifications.

Editing a Listener Notification

Perform the following tasks to edit a listener notification.

➤ To edit a listener notification

1. In Designer, expand the package and folder that contain the listener notification you want to edit.
2. Select the listener notification you want to edit.

Designer displays the configured listener notification in the adapter's Adapter Notification Editor.

3. 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.
4. Modify the values for the listener notification's parameters as needed. For detailed descriptions of the notification's parameters, see [“Configuring Listener Notifications” on page 86](#) for the specific type of listener notification that you want to edit.

Note:

Because listener notifications inherently depend on listeners, you cannot change a listener for a listener notification after you configure it.

5. After you have completed 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.

Editing an Asynchronous Publishable Document Type

Perform the following tasks to edit an asynchronous publishable document type.

➤ To edit an asynchronous publishable document type

1. In Designer, expand the package and folder that contain the publishable document type that you want to edit.
2. Select the asynchronous publishable document type that you want to edit.

Designer displays the configured asynchronous publishable document type in the adapter's Adapter Listener Editor.

3. Select the **Publication Properties** tab, and modify the available values for the document type's parameters as needed. For detailed descriptions of the asynchronous document type's parameters, see [“Configuring Asynchronous Listener Notifications” on page 86](#).

Deleting Listener Notifications

If you no longer want to use a particular MSMQ Adapter listener notification, you can delete it by following the instructions in this section. You delete listener notifications, both synchronous and asynchronous, using Designer.

Important:

If you delete a synchronous listener notification, the associated request and reply document types are deleted automatically. If you delete an asynchronous listener notification, the associated publishable document type is deleted automatically. You cannot solely delete the document types associated with the synchronous and asynchronous listener notifications.

➤ To delete a listener notification

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

Enabling Listener Notifications

After you configure a listener notification, you need to enable it using Integration Server Administrator.

➤ To enable a listener notification

1. In the **Adapters** menu in Integration Server Administrator navigation area, click **MSMQ Adapter**.
2. In the **MSMQ Adapter** menu, select **Listener Notifications**.
3. On the Listener Notifications screen, click **No** in the **Enabled** column for the listener notification you want to enable.

Integration Server Administrator enables the listener notification and displays a ✓ and **Yes** in the **Enabled** column.

Disabling Listener Notifications

You disable listener notifications using Integration Server Administrator.

Important:

Before you disable a listener notification, you must disable the associated listener. For instructions on how to disable a listener, see [“Disabling Listeners”](#) on page 84.

➤ **To disable a listener notification**

1. In the **Adapters** menu in Integration Server Administrator navigation area, click **MSMQ Adapter**.
2. In the **MSMQ Adapter** menu, select **Listener Notifications**.
3. On the Listener Notifications screen, click **Yes** in the **Enabled** column for the listener notification you want to disable.

The listener notification becomes disabled and **No** displays in the **Enabled** column.

7 Predefined Health Indicator

■ Predefined Health Indicator	104
-------------------------------------	-----

Predefined Health Indicator

Microservices Runtime includes predefined health indicators for some of its basic components. The health indicator captures the connection details for all the WmART based adapters at runtime. For more information, see *webMethods Adapter Runtime User's Guide*.

8 Administrator APIs

■ Administrator APIs	106
----------------------------	-----

Administrator APIs

The Administrator APIs are available for MSMQ Adapter. For more information about Administrator APIs and samples, see *webMethods Adapter Runtime User's Guide*.

9 Configuration Variables Templates for Adapter Assets in Microservices Runtime

- [Configuration Variables Templates for Adapter Assets in Microservices Runtime](#) 108

Configuration Variables Templates for Adapter Assets in Microservices Runtime

The webMethods Adapter Runtime (ART) asset properties that can be configured from Integration Server Administrator are available in the configuration variables template (`application.properties` file) generated by Microservices Runtime. For more information, see *webMethods Adapter Runtime User's Guide* and *Developing Microservices with webMethods Microservices Runtime*.

10 Logging and Exception Handling

■ Overview	110
■ MSMQ Adapter Logging Levels	110
■ MSMQ Adapter Message Logging	111
■ MSMQ Adapter Exception Handling	112
■ MSMQ Adapter Error Codes	112

Overview

This chapter describes message logging, exception handling, and tracing for MSMQ Adapter.

To view the list of error codes and supporting information, refer to “[MSMQ Adapter Error Codes](#)” on page 112.

MSMQ Adapter Logging Levels

MSMQ Adapter uses Integration Server logging mechanism to log messages. You can configure and view Integration Server logs to monitor and troubleshoot MSMQ Adapter. For detailed information about logging in Integration Server, including instructions for configuring and viewing the different kinds of logs supported by the server, see the *webMethods Integration Server Administrator's Guide* for your release.

Accessing the Adapter Logging Levels

Beginning with Integration Server 7.1, you can configure different logging levels for MSMQ Adapter.

> To access the adapter's logging information

1. From Integration Server Administrator screen, select **Settings > Logging**.

The **Logging Settings** screen appears. The **Loggers** section has Adapters included in the **Facility** section.

2. Expand the **Adapters** tree to see a list of all installed adapters with their code number and adapter description, along with the logging level.

Changing the Logging Settings

Perform the following procedure to change the adapter's logging settings. For complete information about specifying the amount and type of information to include in the log, see the *webMethods Audit Logging Guide* for your release.

> To change logging settings for the adapter

1. Click **Edit Logging Settings**. Select the required **Level of Logging** for MSMQ Adapter.
2. After making your changes, click **Save Changes**.

MSMQ Adapter Message Logging

Integration Server maintains several types of logs; however, MSMQ Adapter logs messages only to the audit, error, and server logs. Because MSMQ Adapter works in conjunction with the WmART package, the adapter's messages and exceptions typically appear within log messages for the WmART package, although that is not necessarily true for verbose-level messages.

The logging levels are different depending on which version of Integration Server you are running the adapter on, as shown in the following table.

Integration Server	Log	Description
Integration Server 6.5	Audit Log	You can monitor individual adapter services using the audit log as you would audit any service in Integration Server. The audit properties for an adapter service are available in each MSMQ Adapter service template on the Audit tab.
	Error Log	MSMQ Adapter automatically posts critical-level and error-level log messages to the server's error log. These log messages will appear as adapter run-time messages.
	Server Log	MSMQ Adapter posts messages to the Server log, depending on how the server log is configured. Critical-level through debug-level log messages appear as Adapter run-time log messages. Verbose Logging Levels log messages appear as MSMQ Adapter log messages. The Verbose Logging Levels range from Level 5 (V1-Verbose1) through Level 8 (V4-Verbose4).
Integration Server 7.1 and higher	Audit Log	You can monitor individual adapter services using the audit log as you would audit any service in Integration Server. The audit properties for an adapter service are available in each MSMQ Adapter service template on the Audit tab.
	Error Log	MSMQ Adapter automatically posts fatal-level and error-level log messages to the server's error log. These log messages will appear as adapter run-time messages.
	Server Log	MSMQ Adapter posts messages to the server log, depending on how the server log is configured. Fatal-level through debug-level log messages appear as adapter run-time log

Integration Server	Log	Description
		messages. Trace-level log messages appear as MSMQ Adapter log messages.

MSMQ Adapter log messages appear in the format `ADA.0660.nnnn`, where:

- ADA is the facility code that indicates that the message is from an adapter.
- 0660 (or 660) is the MSMQ Adapter major code, which indicates that MSMQ Adapter generated the error.
- *nnnn* represents the error's minor code. For detailed descriptions of minor codes that MSMQ Adapter uses, see “[MSMQ Adapter Error Codes](#)” on page 112.

To monitor MSMQ Adapter log messages in Integration Server log, ensure that your server log's logging settings are configured to monitor the following facilities:

- 0113 Adapter Runtime (Managed Object)
- 0114 Adapter Runtime
- 0115 Adapter Runtime (Listener)
- 0116 Adapter Runtime (Notification)
- 0117 Adapter Runtime (Adapter Service)
- 0118 Adapter Runtime (Connection)
- 0121 Adapter Runtime (SCC Transaction Manager)
- 0126 Adapter Runtime (SCC Connection Manager)

MSMQ Adapter Exception Handling

If an MSMQ Adapter object encounters an error with the MSMQ system, it will throw an adapter error coupled with the MSMQ error, exactly as it was thrown by the MSMQ system.

For example, if the property values of the `MSMQQueueInfo` object fail to refresh, you will receive an adapter error. This error indicates that the refresh failed, and the adapter error will contain the specific error generated on the MSMQ system indicating why the service failed. In this case, you would receive an MSMQ error specifying that the queue does not exist, or that you do not have sufficient permissions to perform the operation.

MSMQ Adapter Error Codes

The following table lists MSMQ Adapter error codes and provides information on the error message, reason, and possible action for each error.

Error Code	Description
100	Fail to destroy non-transactional connection.
	Explanation: The opened queue associated with the non-transactional connection cannot be closed.
	Action: Refer to the Microsoft MSMQ documentation and error codes. Contact your MSMQ system administrator.
150	Resource connection exception.
	Explanation: Cannot open MSMQ for connection.
	Action: Refer to the Microsoft MSMQ documentation and error codes. Contact your MSMQ system administrator.
151	Fail to create MSMQ local transactional object.
	Explanation: MSMQ internal transaction cannot be started.
	Action: Refer to the Microsoft MSMQ documentation and error codes. Contact your MSMQ system administrator.
152	Fail to destroy transactional connection.
	Explanation: The opened queue with the transaction connection cannot be closed.
	Action: Refer to the Microsoft MSMQ documentation and error codes. Contact your MSMQ system administrator.
354	Adapter cannot open queue for operations.
	Explanation: The queue path name or format name is not correct or is not specified.
	Action: Make sure that you provided a valid queue path name or format name.
450	The monitored queue by listener is unavailable.
	Explanation: The queue associated with the connection is not open.
	Action: Make sure the queue is still accessible. Restart the connection and the listener, if necessary.
451	There is no notification associated with this listener or no notification is enabled on this listener.
	Explanation: The listener does not have a notification associated with it or there is an associated notification but it is not enabled.
	Action: Configure and enable a notification, then enable the listener.
1000	Adapter cannot initialize MSMQ message object.

Error Code	Description
	<p>Explanation: Failed to create the MSMQMessage object to provide access to the properties that define a message and the methods used to send the message to its destination queue.</p>
	<p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes.</p>
1002	<p>Fail to send message to non-transactional queue.</p>
	<p>Explanation: Error occurs when a message is put on a non-transactional queue.</p>
	<p>Action: Refer to the Microsoft MSMQ documentation and error codes.</p>
1004	<p>Fail to send message to transactional queue.</p>
	<p>Explanation: Error occurs when a message is put on a transactional queue.</p>
	<p>Action: Refer to the Microsoft MSMQ documentation and error codes.</p>
1005	<p>Fail to write byte sequence as message body.</p>
	<p>Explanation: Error occurs when the adapter tries to write a byte sequence as a message body.</p>
	<p>Action: Check IOException for details.</p>
1009	<p>Fail to read byte sequence from message body.</p>
	<p>Explanation: Error occurs when the adapter tries to read a byte sequence as a message body.</p>
	<p>Action: Check IOException for details.</p>
1015	<p>Fail to specify the message contents.</p>
	<p>Explanation: Error occurs when setting message object contents.</p>
	<p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1019	<p>Fail to retrieve the message contents.</p>
	<p>Explanation: Error occurs when the adapter tries to read the message contents.</p>
	<p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1046	<p>Adapter cannot initialize MSMQ queue object.</p>
	<p>Explanation: Fail to create an MSMQQueue object that represents an open instance of a queue and provides the methods needed to read and delete the messages in the queue.</p>

Error Code	Description
1047	<p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p> <p>Adapter cannot close MSMQ queue.</p> <p>Explanation: Error occurs when closing this instance of the queue.</p> <p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1048	<p>Fail to return the cursor to the start of the queue.</p> <p>Explanation: Error occurs when the adapter tries to reset the cursor position.</p> <p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1049	<p>Cannot find out whether the queue is open or not.</p> <p>Explanation: Fail to check if the queue is open.</p> <p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1050	<p>Cannot retrieve message from non-transactional queue.</p> <p>Explanation: Fail to retrieve or remove a message from a non-transactional queue.</p> <p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1051	<p>Cannot retrieve message from transactional queue.</p> <p>Explanation: Fail to retrieve or remove a message from a transactional queue.</p> <p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1052	<p>Cannot peek message.</p> <p>Explanation: Fail to peek the first message in the queue or the first message that matches the filter values in the queue.</p> <p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1053	<p>Adapter cannot initialize management object of the specified queue.</p> <p>Explanation: Fail to create an MSMQQueueInfo object that provides queue management for a single queue and provides methods for opening a queue, changing or retrieving the properties of a queue.</p>

Error Code	Description
1054	<p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p> <p>Adapter cannot refresh the property values of the queue management object.</p> <p>Explanation: Fail to refresh the property values of the MSMQQueueInfo object. These values are retrieved from the directory service (public queues) or from the local computer (private queues).</p> <p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1056	<p>Adapter cannot open a queue.</p> <p>Explanation: Fail to open a queue for sending, peeking at, or retrieving messages.</p> <p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1057	<p>Cannot find the path name of the queue.</p> <p>Explanation: Error occurs when the adapter tries to locate the complete Message Queuing path name of the queue.</p> <p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1058	<p>Cannot set path name of the queue.</p> <p>Explanation: Error occurs when the adapter tries to set the queue path name of a MSMQQueueInfo object.</p> <p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error code for details.</p>
1059	<p>Cannot find the format name of the queue.</p> <p>Explanation: Error occurs when the adapter tries to find the format name of the queue.</p> <p>Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1060	<p>Cannot set the format name of the queue.</p> <p>Explanation: Error occurs when the adapter tries to set the queue format name of the MSMQQueueInfo object.</p> <p>Action: Check the AdapterException message. Refer to Microsoft MSMQ documentation and error codes for details.</p>

Error Code	Description
1061	Adapter cannot initialize transaction object.
	Explanation: Fail to create an MSMQTransaction object.
	Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1062	Cannot commit MSMQ transaction.
	Explanation: Fail to commit the transaction.
	Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1063	Cannot rollback MSMQ transaction.
	Explanation: Fail to rollback the transaction.
	Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1064	Adapter cannot initialize the object to start MSMQ internal transaction object.
	Explanation: Fail to create an MSMQTransactionDispenser object.
	Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1065	Fail to begin an MSMQ internal transaction.
	Explanation: Error occurs when the adapter tries to initiate a new internal transaction and returns an MSMQ transaction object that represents the new underlying transaction object.
	Action: Check the AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1067	Fail to get format name of a queue.
	Explanation: MSMQQueueInfo object is NULL.
	Action: Ensure the connection is available and the queue is accessible. Re-enable the connection, if necessary.
1080	Fail to set format name of queue.
	Explanation: MSMQQueueInfo object is NULL.
	Action: Ensure the connection is available and the queue is accessible. Re-enable the connection, if necessary.
1081	Fail to set path name of queue.

Error Code	Description
	Explanation: MSMQQueueInfo object is NULL. Action: Ensure the connection is available and the queue is accessible. Re-enable the connection, if necessary.
1082	Fail to get path name of queue. Explanation: MSMQQueueInfo object is NULL. Action: Make sure the connection is available and the queue is accessible. Re-enable the connection, if necessary.
1084	Cannot peek message contents. Explanation: Fail to peek the contents of the first message in the queue or the content of the first message that matches the filter values in the queue. Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1085	Cannot get admin queue format name. Explanation: Error occurs when the adapter tries to find the admin queue format name of the message. Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1086	Cannot get admin queue path name. Explanation: Error occurs when the adapter tries to find the admin queue path name of the message. Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1087	Cannot get destination queue path name. Explanation: Error occurs when the adapter tries to find the destination queue path name of the message. Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1088	Cannot get destination queue format name. Explanation: Error occurs when the adapter tries to find the destination queue format name of the message. Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.

Error Code	Description
1089	Cannot set admin queue path name.
	Explanation: Error occurs when the adapter tries to set the admin queue path name of the message.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1090	Cannot set admin queue format name.
	Explanation: Error occurs when the adapter tries to set the admin queue format name of the message.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1091	Cannot find if the message is the first one in its transaction.
	Explanation: Error occurs when the adapter tries to determine whether the message is the first one in its transaction.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1092	Cannot find if the message is the last one in its transaction.
	Explanation: Error occurs when the adapter tries to determine whether the message is the last one in its transaction.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1093	Cannot specify the type of acknowledgment message that MSMQ will post in admin queue.
	Explanation: Error occurs when the adapter tries to set the ack property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1094	Cannot specify application-generated information.
	Explanation: Error occurs when the adapter tries to set the AppSpecific property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1095	Cannot specify whether the message should be authenticated and what type of digital signature is used.

Error Code	Description
	<p>Explanation: Error occurs when the adapter tries to set the AuthLevel property of the MSMQMessage object.</p>
	<p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1096	<p>Cannot specify how Message Queuing delivers the message.</p>
	<p>Explanation: Error occurs when the adapter tries to set the Delivery property of the MSMQMessage object.</p>
	<p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1097	<p>Cannot specify whether Message Queuing stores copies of the message as it is routed to the destination queue.</p>
	<p>Explanation: Error occurs when the adapter tries to set the Journal property of the MSMQMessage object.</p>
	<p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1098	<p>Cannot specify the message's priority.</p>
	<p>Explanation: Error occurs when the adapter tries to set the priority property of the MSMQMessage object.</p>
	<p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1099	<p>Cannot specify a time limit (in seconds) for the message to be retrieved from the target queue.</p>
	<p>Explanation: Error occurs when the adapter tries to set the TimeToBeReceived property of the MSMQMessage object.</p>
	<p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1100	<p>Cannot specify a time limit (in seconds) for the message to reach the queue.</p>
	<p>Explanation: Error occurs when the adapter tries to set the TimeToReachQueue property of the MSMQMessage object.</p>
	<p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1101	<p>Cannot specify how Message Queuing traces the route of the message.</p>
	<p>Explanation: Error occurs when the adapter tries to set the Trace property of the MSMQMessage object.</p>

Error Code	Description
1102	<p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p> <p>Cannot find the type of acknowledgment messages that Message Queuing will post (in the administration queue) when acknowledgments are requested.</p> <p>Explanation: Error occurs when the adapter tries to get the Ack property of the MSMQMessage object.</p> <p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1103	<p>Cannot find application-generated information.</p> <p>Explanation: Error occurs when the adapter tries to get the AppSpecific property of the MSMQMessage object.</p> <p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1104	<p>Cannot find whether the message should be authenticated and what type of digital signature is used.</p> <p>Explanation: Error occurs when the adapter tries to get the AuthLevel property of the MSMQMessage object.</p> <p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1105	<p>Cannot find the message type.</p> <p>Explanation: Error occurs when the adapter tries to get the MsgClass property of the MSMQMessage object.</p> <p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1106	<p>Cannot find how Message Queuing delivers the message.</p> <p>Explanation: Error occurs when the adapter tries to get the Delivery property of the MSMQMessage object.</p> <p>Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.</p>
1107	<p>Cannot find whether Message Queuing stores copies of the message as it is routed to the destination queue.</p> <p>Explanation: Explanation: Error occurs when the adapter tries to get the Journal property of the MSMQMessage object.</p>

Error Code	Description
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1108	Cannot find the message's priority.
	Explanation: Error occurs when the adapter tries to get the priority property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1109	Cannot find the time limit (in seconds) for the message to be retrieved from the target queue.
	Explanation: Error occurs when the adapter tries to get the TimeToBeReceived property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1110	Cannot specify a time limit (in seconds) for the message to reach the queue.
	Explanation: Error occurs when the adapter tries to get the TimeToReachQueue property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1111	Cannot find how Message Queuing traces the route of the message.
	Explanation: Error occurs when the adapter tries to get the Trace property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1112	Cannot specify a description of the documentation message.
	Explanation: Error occurs when the adapter tries to set the Label property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1113	Cannot find the description of the message.
	Explanation: Error occurs when the adapter tries to get the Label property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.

Error Code	Description
1114	Cannot find the identifier of the source computer that sent the message.
	Explanation: Error occurs when the adapter tries to get the SrcMachineId property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1115	Cannot find when the message arrived at the queue.
	Explanation: Error occurs when the adapter tries to get the ArriveTime property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1116	Cannot find when the message is sent.
	Explanation: Error occurs when the adapter tries to get the SendTime property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1117	Cannot find whether the message was authenticated at the request of the sending application.
	Explanation: Error occurs when the adapter tries to get the Authenticated property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1118	Cannot find the message's 20-byte, correlation identifier.
	Explanation: Error occurs when the adapter tries to get the CorrelationId property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1119	Cannot find the message ID.
	Explanation: Error occurs when the adapter tries to get the Id property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1120	Cannot specify a message's 20-byte, correlation identifier.

Error Code	Description
	Explanation: Error occurs when the adapter tries to set the CorrelationId property of the MSMQMessage object.
	Action: Check the successive AdapterException message. Refer to the Microsoft MSMQ documentation and error codes for details.
1121	Unsupported message body type.
	Explanation: The message body type you specified is not supported by the adapter.
	Action: Make sure that you are using the supported body types, which are String and byte array.

A Built-In Transaction Management Services

■ Transaction Management Overview	126
■ Built-In Transaction Management Services	128
■ Changing the Integration Server Transaction Timeout Interval	130
■ Transaction Error Situations	131

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 MSMQ Adapter services in the WmART package. For descriptions of each of the specific built-in transaction management services that can be used with the WmART package, see [“Built-In Transaction Management Services” on page 128](#).

For information about other built-in services available with MSMQ Adapter, see the *webMethods Integration Server Built-In Services Reference* for your release.

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 Put services that place messages on a queue and one or more Put services fail, all of the messages are rolled back.

Transaction Types

MSMQ Adapter supports local transactions with the MSMQ Server (LOCAL_TRANSACTION), without other resource participants. MSMQ Adapter does not support XATransactions.

Note:MSMQ Adapter can communicate with only one resource at a time.

For a description of the transaction types supported by MSMQ Adapter, see [“Transaction Management of MSMQ Adapter Connections” on page 15](#).

Implicit and Explicit Transactions

Implicit transactions are handled automatically 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 local 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 a flow service executes an adapter service. The connection required by the adapter service is registered with the newly created context and used by the adapter service. If the flow executes another adapter service, the transaction context is searched to see if the connection is registered already. If the connection is already registered, the adapter service uses this connection. If the connection

is not registered, Integration Server retrieves a new connection instance and registers it 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 completes and either is committed or rolled back, depending on the status (success or failure) of the top level flow.

A single transaction context can contain 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 more 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 MSMQ Adapter Connections” on page 15](#).

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.

A transaction context starts when the `pub.art.transaction.startTransaction()` service executes. The transaction context completes when either the `pub.art.transaction.commitTransaction()` or `pub.art.transaction.rollbackTransaction()` service executes. As with implicit transactions, a single transaction context can contain 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()` service, or you will have dangling transactions, which 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 more 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 MSMQ Adapter Connections” on page 15](#).

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.

For more information about implicit and explicit transactions, see [“Transaction Management Overview” on page 126](#).

Input Parameters

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

<i>transactionName</i>	String. Used to associate a name with an explicit transaction. The <i>transactionName</i> must correspond to the <i>transactionName</i> in any <code>pub.art.transaction:startTransaction</code> or <code>pub.art.transaction:rollbackTransaction</code> services associated with the explicit transaction. This value must be mapped from the most recent <code>pub.art.transaction:startTransaction</code> 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.

For more information about implicit and explicit transactions, see [“Transaction Management Overview” on page 126](#).

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. To change the server's default transaction timeout, see [“Changing the Integration Server Transaction Timeout Interval” on page 130](#).

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 current executions associated with the flow are cancelled and rolled back if necessary.

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

Input Parameters

timeoutSeconds **Integer.** The number of seconds that the implicit or explicit transaction stays open before the transaction manager aborts it.

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.

For more information about implicit and explicit transactions, see [“Transaction Management Overview” on page 126](#).

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. The <i>transactionName</i> must correspond to the <i>transactionName</i> in any <code>pub.art.transaction:rollbackTransaction</code> or <code>pub.art.transaction:commitTransaction</code> services associated with the explicit transaction.
------------------------	---

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. The <i>transactionName</i> must correspond to the <i>transactionName</i> in any <code>pub.art.transaction:rollbackTransaction</code> or <code>pub.art.transaction:commitTransaction</code> services associated with the explicit transaction.
------------------------	--

Changing the Integration Server Transaction Timeout Interval

The 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.

Be sure to 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.

Transaction Error Situations

When Integration Server encounters a situation that could compromise transactional integrity, it throws an error. Such situations include the following:

- A transaction includes a resource that only supports local transactions.

If a transaction accesses multiple resources, and more than one of the resources supports only local transactions, the integrity of the transaction cannot be guaranteed. For example, if the first resource commits successfully, and the second resource fails to commit, the first resource interaction cannot be rolled back; it has been committed already. To help prevent this problem, Integration Server detects this case when connecting to more than one resource that does not support two-phase commits. It throws a run-time exception and the service execution fails.

Note:

Because this situation may be acceptable in some applications, the adapter user can include an input in the `startTransaction` service to cause to allow this situation.

- A resource is used in both a parent transaction and a nested transaction.

This situation is ambiguous, and most likely means that a nested transaction was not closed properly.

- A parent transaction is closed before its nested transaction.

After a service request has invoked all its services, but before returning results to the caller, the service may commit its work. This commit could fail if the resource is unavailable or rejects the commit. This will cause the entire server request to fail and to roll back the transaction.

