

# webMethods AS/400 Adapter Installation and User's Guide

Version 6.0.1

July 2012

This document applies to webMethods AS/400 Adapter 6.0.1 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 © 2006-2021 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

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

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

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

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

**Document ID: ADAPTER-WAS-IUG-601-20210426**

# Table of Contents

<b>About this Guide</b> .....	<b>5</b>
Document Conventions.....	6
Online Information and Support.....	7
Data Protection.....	7
<b>1 Overview of webMethods AS/400 Adapter</b> .....	<b>9</b>
About the Adapter.....	10
Architecture and Components.....	10
Controlling Pagination.....	11
<b>2 Installing and Uninstalling the AS/400 Adapter</b> .....	<b>13</b>
Overview.....	14
Requirements.....	14
The Integration Server Home Directory.....	14
Installing AS/400 Adapter 6.0.1.....	14
Uninstalling AS/400 Adapter 6.0.1.....	15
<b>3 Managing AS/400 Adapter Connections</b> .....	<b>17</b>
Overview.....	18
Configuring Adapter Connections.....	18
Viewing Adapter Connections.....	20
Editing Adapter Connections.....	21
Deleting Adapter Connections.....	22
Enabling Adapter Connections.....	22
Disabling Adapter Connections.....	23
<b>4 Managing Data Queue Listeners</b> .....	<b>25</b>
Overview.....	26
Clustering considerations.....	26
Configuring Data Queue Listeners.....	26
Viewing Data Queue Listeners.....	28
Editing Data Queue Listeners.....	28
Deleting Data Queue Listeners.....	29
Enabling Data Queue Listeners.....	29
Disabling Data Queue Listeners.....	30
<b>5 Adapter Logging and Exception Handling</b> .....	<b>31</b>
Overview.....	32
Adapter Message Logging.....	32
AS/400 Adapter Exception Handling.....	32
Debugging or Setting up the Trace.....	33

<b>6 Externalizing Adapter Connection Assets.....</b>	<b>35</b>
Externalizing Adapter Connection Assets.....	36
Externalizing AS400 Adapter.....	36
<b>7 Predefined Health Indicator for WmAS400 Adapters.....</b>	<b>39</b>
<b>8 AS/400 Adapter Administrator APIs.....</b>	<b>41</b>
Overview.....	42
Connection Operations.....	43
Listener Operations.....	46
<b>A Built-in Services.....</b>	<b>49</b>
Overview.....	50
Access Services.....	50

# About this Guide

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

---

This guide describes how to install, configure, and use the webMethods AS/400 Adapter 6.0.1. This guide also describes the built-in services provided by the AS/400 Adapter. It contains information for administrators and application developers who want to exchange data with AS/400 programs and data queues using the AS/400 IBM ToolBox for Java.

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

- The terminology and the basic operations of your operating system, AS/400 IBM ToolBox for Java, AS/400 RPG, AS/400 Control Language (CL), and the data queue concepts.
- The setup and operation of webMethods Integration Server.
- How to perform basic tasks with Software AG Designer.

## Document Conventions

---

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

---

## Online Information and Support

---

### Software AG Documentation Website

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

### Software AG Empower Product Support Website

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

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

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

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

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

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

### Software AG TECHcommunity

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

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

## Data Protection

---

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





# 1 Overview of webMethods AS/400 Adapter

---

■ About the Adapter .....	10
■ Architecture and Components .....	10
■ Controlling Pagination .....	11

## About the Adapter

The webMethods AS/400 Adapter is an add-on to the webMethods Integration Server that enables you to exchange data with an IBM AS/400 server through the use of IBM ToolBox for Java. The adapter provides seamless and real-time communication with the AS/400 server.

Using the AS/400 Adapter, webMethods clients can create and run services that execute commands and programs, and read and write data from the data queues on the AS/400 server.

To use all the IBM ToolBox for Java classes, use the Java 2 Platform, Standard Edition (J2SE). You should also review the OS/400 requirements for running the IBM ToolBox for Java.

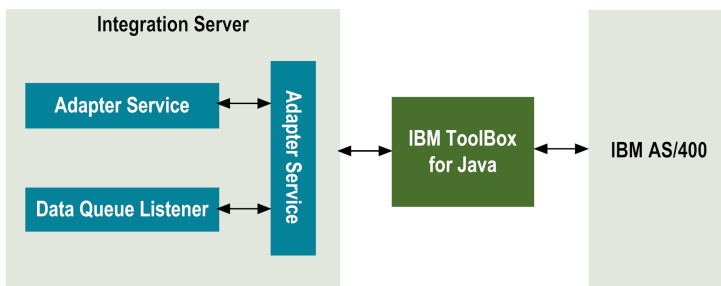
## Architecture and Components

The AS/400 Adapter is installed on the webMethods Integration Server. The AS/400 Adapter enables Integration Server to connect to, and perform operations on the AS/400 server using the IBM ToolBox for Java classes. You must install the supported IBM ToolBox for Java on Integration Server and make sure the classes are loaded when the server starts.

The AS/400 Adapter enables you to configure the following components:

- **Adapter Connections:** Enable Integration Server to connect to the AS/400 systems at run time.
- **Adapter Services:** Perform operations on the AS/400 resources using the built-in services provided with the AS/400 Adapter.
- **Data Queue Listeners:** Monitor a data queue and notify the resources when an entry is written to the queue.

The following diagram illustrates how the Integration Server uses these components to interact with the AS/400 server.



## Adapter Connections

AS/400 Adapter connections enable Integration Server to connect to the AS/400 systems at run time.

You can configure one or more AS/400 Adapter connections for integrations. The number of connections and the type of connections that you create depends on your integration needs. For

example, if you have a test system and a production system, you can create a connection for each system. Additionally, you could configure multiple connections to the same server.

The AS/400 Adapter connections contain parameters that Integration Server uses to manage connections to the AS/400 server. You configure connections using the Integration Server Administrator.

For more information on configuring and managing AS/400 Adapter connections, see [“Managing AS/400 Adapter Connections” on page 17](#).

## Adapter Services

The AS/400 Adapter provides a set of built-in services that enable you to create and submit requests to an AS/400 server. The built-in adapter services require connecting only the DATAQUEUE and COMMAND connection types. For a complete list of built-in adapter services, see [“Built-in Services” on page 49](#).

## Data Queue Listeners

A Data Queue Listener monitors a specified data queue and notifies the application when an entry is written to the data queue.

When you create a Data Queue Listener, you assign it an adapter connection that you configured earlier. For more information on AS/400 Adapter connections, see [“Managing AS/400 Adapter Connections” on page 17](#). After you create a Listener you must enable it. You use the Integration Server Administrator to accomplish these tasks. See [“Managing Data Queue Listeners” on page 25](#), for instructions.

After you enable a Listener, Integration Server invokes the notification node every time an entry is written to the data queue and places the data queue entry in the pipeline of the service.

## 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. 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 and Uninstalling the AS/400 Adapter

---

■ Overview .....	14
■ Requirements .....	14
■ The Integration Server Home Directory .....	14
■ Installing AS/400 Adapter 6.0.1 .....	14
■ Uninstalling AS/400 Adapter 6.0.1 .....	15

## Overview

---

This chapter explains how to install, upgrade, and uninstall webMethods AS/400 Adapter 6.0.1. 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 the *Installing webMethods Products On Premises* for your release.

## Requirements

---

For a list of the operating systems, AS/400 products, and webMethods products supported by the adapter, see the *webMethods Adapters System Requirements*.

AS/400 Adapter 6.0.1 has no hardware requirements beyond those of its host Integration Server.

## The Integration Server Home Directory

---

Beginning with Integration Server 9.6, you can create and run multiple Integration Server instances under a single installation directory. Each Integration Server instance has a home directory under *Integration Server\_directory \instances\instance\_name* that contains the packages, configuration files, log files, and updates for the instance.

For more information about running multiple Integration Server instances, see the *webMethods Integration Server Administrator's Guide* for your release.

If you are using Integration Server 9.5 and lower, the Integration Server home directory is *Integration Server\_directory*. For example, on Integration Server 9.5 the adapter package is installed in the *Integration Server\_directory \packages* directory.

This guide uses the *packages\_directory* as the home directory in Integration Server classpaths. For Integration Server 9.6 and above, the *packages\_directory* is *Integration Server\_directory \instances\instance\_name\packages* directory. For Integration Server 9.5 and lower, the *packages\_directory* is *Integration Server\_directory \packages* directory.

## Installing AS/400 Adapter 6.0.1

---

### **Important:**

AS/400 Adapter 6.0.1 is not supported in a clustered Integration Server environment.

### ➤ To install AS/400 Adapter 6.0.1

1. Download Installer from the [Empower Product Support Web site](#).
2. If you are installing the adapter on an existing Integration Server, shut down the Integration Server.
3. Start the Installer wizard.

4. Choose the webMethods release that includes the Integration Server on which to install the adapter.
5. Specify the installation directory as follows:
  - If you are installing on an existing Integration Server, specify the webMethods installation directory that contains the host Integration Server.
  - If you are installing both the host Integration Server and the adapter, specify the installation directory to use.
6. In the product selection list, select **Adapters >webMethods AS/400 Adapter 6.0.1**.

If you are using Integration Server 9.6 and above, you can choose to install the package in the default instance. In this case, Software AG Installer installs the adapter in both locations, *Integration Server\_directory* \packages and the default instance packages directory located in *Integration Server\_directory* \instances\default\packages.

7. To download the documentation for the adapter, go to [Software AG Documentation website](#).
8. After installation is complete, close Installer.

Integration Server loads the WmAS400 package, but it will not be enabled until you deploy the IBM ToolBox for Java files on Integration Server.

9. To deploy the IBM ToolBox for Java files:
  - a. Go to the IBM ToolBox for Java installation directory.
  - b. Copy the jt400.jar and util400.jar files to the *Integration Server\_directory* \packages\WmAS400\code\jars directory.
10. Start the host Integration Server.

**Note:**

Software AG recommends that you install the latest fix to include updates from previous fixes for this adapter

## Uninstalling AS/400 Adapter 6.0.1

### > To uninstall AS/400 Adapter 6.0.1

1. Shut down the Integration Server that hosts the AS/400 Adapter.
2. Start Software AG Uninstaller, selecting the webMethods installation directory that contains the host Integration Server. In the product selection list, select **Adapters >webMethods AS/400 Adapter 6.0.1**. You can also choose to uninstall documentation.

3. Restart the host Integration Server.
4. Uninstaller removes all AS/400 Adapter 6.0.1 related files that were installed into the *Integration Server\_directory* \packages\WmAS400 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 WmAS400 directory.



# 3 Managing AS/400 Adapter Connections

---

■ Overview .....	18
■ Configuring Adapter Connections .....	18
■ Viewing Adapter Connections .....	20
■ Editing Adapter Connections .....	21
■ Deleting Adapter Connections .....	22
■ Enabling Adapter Connections .....	22
■ Disabling Adapter Connections .....	23

## Overview

This chapter provides instructions for managing AS/400 Adapter connections. You must have webMethods Integration Server administrator privileges to access the AS/400 Adapter's administrative screens.

## Configuring Adapter Connections

When you configure AS/400 Adapter connections, you specify information that the Integration Server uses to connect and to log on to the server.

Before you can configure a connection, you must have the AS/400 Adapter and IBM ToolBox for Java installed and loaded by the Integration Server. For more information, see [“Installing and Uninstalling the AS/400 Adapter” on page 13](#). You create AS/400 Adapter connections using the Integration Server Administrator.

### ➤ To configure an AS/400 Adapter connection

1. Start the Integration Server Administrator if it is not already running.
2. Make sure the WmAS400 package is enabled.

To see the status of the WmAS400 package, in the **Packages** menu in the navigation area, select **Management**. The IS Administrator displays the list of packages that are installed on the Integration Server. If the WmAS400 package is disabled, click **No** in the **Enabled** column to enable it.

3. In the **Adapters** menu in the IS Administrator's navigation area, click **AS/400 Adapter**.
4. On the Connections screen, click **Configure New Connection**.
5. On the Connection Types screen, click the AS/400 service you want to connect to.
  - To execute non-interactive CL commands or RPG programs, you need to create a connection using the COMMAND connection type.
  - To perform operations on the DataQueue, for example, read or write, you need to create a connection using the DATAQUEUE connection type.
6. On the Configure Connection Type screen, provide values for the connection's parameters.

Parameter	Description
<b>Connection Alias</b>	Required. The name you want to give the connection. Connection names cannot have spaces or use special characters reserved by the Integration Server, or Designer. For more information about the use

Parameter	Description
	of special characters in package, folder, and element names, see the <i>webMethods Service Development Help</i> for your release.
<b>ServerName</b>	Required. The name or the IP address of the AS/400 server, for example, AS/400 Adapter.com.
<b>User</b>	Required. The user name that the connection will use to connect to the server.
<b>Password</b>	Required. The password for the user account defined in the <b>User</b> field.
<b>Initialization Service</b>	Optional. The integration service that you want to invoke every time a new connection is created, for example <code>wm.as400.sample.connection:initService</code> . You can use the service to call the AS/400 programs or commands that would set the AS/400 sessions environment (current library, and so on.)
<b>Current Library</b>	Optional. The library that you want to use for accessing the AS/400 resources. Instead of hard coding the library name you can use this parameter in your service.

7. In the Connection Management Properties section, provide values for the following parameters:

Parameter	Description
<b>Enable Connection Pooling</b>	Enables the adapter to use connection pooling. Default: true.
<b>Minimum Pool Size</b>	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.
<b>Maximum Pool Size</b>	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.
<b>Pool Increment Size</b>	If connection pooling is enabled, this parameter specifies the number of connections by which the pool will be incremented if connections are needed, up to the maximum pool size. Default: 1.
<b>Block Timeout</b>	If connection pooling is enabled, this parameter specifies the number of milliseconds that the Integration Server will wait to obtain a connection from the pool before it times out and returns an error. Default: 1000 msec.
<b>Expire Timeout</b>	If connection pooling is enabled, this parameter specifies the number of milliseconds that an inactive connection can remain in the pool

Parameter	Description
	before it is closed and removed from the pool. For example, to specify 10 seconds, specify 10000. Type 0 to specify no timeout. Default: 1000 msec.

8. Click **Save Connection**. If the parameters are valid the connection you created appears on the AS/400 Adapter's Connections screen.
9. Click **Return to AS/400 Adapter Connections Types** to return to the Connection Types screen.

## Viewing Adapter Connections

You can view the AS/400 Adapter connections parameters from the Integration Server Administrator, or Designer.

### Viewing Connections Using the IS Administrator


➤ To view the parameters for an **AS/400 Adapter** connection using the IS Administrator

1. Start the Integration Server Administrator if it is not already running.
2. Make sure the WmAS400 package is enabled.
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **AS/400 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 11](#).

4. On the Connections screen, click the  icon for the connection you want to view.

The View Connection screen displays the parameters for the connection. For descriptions of the connection parameters, see the table of parameters in [“Configuring Adapter Connections” on page 18](#).

5. Click **Return to AS/400 Adapter Connections** to return to the Connections screen.

## Viewing Connections Using Designer

### ➤ To view the parameters for a connection using Designer


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

The parameters for the connection appear on the **Connection Information** tab. For descriptions of the connection parameters, see [“Configuring Adapter Connections” on page 18](#).

## Editing Adapter Connections

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

### ➤ To edit an AS/400 Adapter connection

1. Disable the connection you want to edit. For instructions on disabling a connection, see [“Disabling Adapter Connections” on page 23](#).
2. Start the Integration Server Administrator if it is not already running.
3. Make sure the WmAS400 package is enabled.
4. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **AS/400 Adapter**.
5. On the Connections screen, click the  icon for the connection you want to edit.

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

For descriptions of the connection parameters, see the table of parameters in [“Configuring Adapter Connections” on page 18](#).


6. Click **SaveChanges** to save the connection.
7. Click **Return to AS/400 Adapter Connections** to return to the Connections screen.
8. Enable the connection you edited. See [“Enabling Adapter Connections” on page 22](#) for instructions.

## Deleting Adapter Connections

---

If you no longer want to use an AS/400 Adapter connection, use the following instructions to delete the connection. If you delete an AS/400 Adapter connection, the AS/400 Adapter flow services that are defined to use the connection will no longer work.

### ➤ To delete an AS/400 Adapter connection

1. Start the Integration Server Administrator if it is not already running.
2. Make sure the WmAS400 package is enabled. For instructions, see [“Enabling Adapter Connections” on page 22](#).
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **AS/400 Adapter**.
4. Disable the connection. See [“Disabling Adapter Connections” on page 23](#) for details.
5. On the Connections screen, click the  icon in the **Delete** column for the connection you want to delete.

The Integration Server deletes the adapter connection.

## Enabling Adapter Connections

---

The AS/400 Adapter connections must be enabled before you can use the connection in your services. When you create a connection, it is not automatically enabled.

### Note:

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

### ➤ To enable an AS/400 Adapter connection

1. Start the Integration Server Administrator if it is not already running.

2. Make sure the WmAS400 package is enabled.
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **AS/400 Adapter**.
4. On the Connections screen, click **No** in the **Enabled** column for the connection you want to enable.

The AS/400 Adapter enables the adapter connection and displays ✓ and **Yes** in the **Enabled** column.

## Disabling Adapter Connections

---

AS/400 Adapter connections must be disabled before you can edit or delete the connections.

### ➤ To disable an AS/400 Adapter connection

1. Start the Integration Server Administrator if it is not already running.
2. Make sure the WmAS400 package is enabled.
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **AS/400 Adapter**.
4. On the Connections screen, click **Yes** in the **Enabled** column for the connection you want to disable.

The AS/400 Adapter disables the adapter connection and displays **No** in the **Enabled** column.





# 4 Managing Data Queue Listeners

---

■ Overview .....	26
■ Clustering considerations .....	26
■ Configuring Data Queue Listeners .....	26
■ Viewing Data Queue Listeners .....	28
■ Editing Data Queue Listeners .....	28
■ Deleting Data Queue Listeners .....	29
■ Enabling Data Queue Listeners .....	29
■ Disabling Data Queue Listeners .....	30

## Overview

---

A Data Queue Listener monitors a specified data queue and notifies the application when an entry is written to the data queue. This chapter provides instructions for managing AS/400 Data Queue Listeners. You must have webMethods Integration Server administrator privileges to access the AS/400 Adapter's administrative screens.

When you configure a Data Queue Listener, you must provide information about the:

- The AS/400 Data Queue to monitor.
- The service to invoke when an entry is written in the data queue. The Data Queue Listener puts the `dataQueueEntry`, `dataQueueEntryKey`, and `senderInfo` values in the pipeline before invoking this service. You can further use the built-in adapter service `wm.as400.access:dataQEntryToRecord` to convert the `dataQueueEntry` into a more usable format. See the sample service `wm.as400.sample.simpleDQ:DQLListener` or `wm.as400.sample.keyedDQ:DQLListener` in the `WmAS400Samples` package.

## Clustering considerations

---

The usual recommendation to configure Adapter Runtime based adapters in an Integration Server cluster is to deploy all the adapter packages and services on one Integration Server. Then, replicate them to the other servers in the cluster. Adapter listeners ensure that only one node in an Integration Server cluster receives a notification per event.

This behavior is different for AS/400 Adapter.

When you configure a Data Queue Listener to the same queue in an Integration Server cluster and activate it on every Integration Server, the listener receives the same data from the data queue on all the Integration Server nodes. This causes multiple notifications for the same entry on the data queue.

AS/400 system identifies the listener connection from each Integration Server node in the cluster as a unique client and delivers message to every connection causing duplicate messages.

To ensure that the Data Queue Listener in a cluster receives notifications only once per message in the queue, Software AG recommends enabling the listener only on one node in the cluster.

## Configuring Data Queue Listeners

---

When you configure AS/400 Data Queue Listeners, you specify information that the Integration Server uses to read data from an AS/400 data queue and invoke an IS service.

Before you configure the Data Queue Listeners, you must configure an adapter connection. You create AS/400 Adapter connections using the Integration Server Administrator.

### ➤ To configure an AS/400 Data Queue Listener

1. Start the Integration Server Administrator if it is not already running.

2. Make sure the WmAS400 package is enabled.
3. In the **Adapters** menu in the Integration Server Administrator's navigation area, click **AS/400 Adapter**.
4. In the AS/400 Adapter navigation area, click **Data Queue Listeners**.
5. In the Data Queue Listener screen, click **Configure New Data Queue Listener**.
6. In the Configure New Listener screen, provide values for the following listener parameters:

Parameter	Description
<b>Connection Alias</b>	Required. The name of the AS/400 Adapter connection that the listener uses to connect to the server.
<b>Listener Node Name</b>	The Integration Server service that you want to invoke every time an entry is written to the data queue, for example, <code>wm.as400.sample.keyedDQ:DQListener</code> .
<b>Data Queue Name</b>	The fully qualified integrated file system name of the data queue you want the listener to monitor for data, for example, <code>/QSYS.LIB/PUNEET.LIB/MYKEYQUEUE.DTAQ</code> .
<b>Is Keyed</b>	Set to true if the data queue is a keyed data queue. Default value: false.
<b>Key Type</b>	The AS/400 data type of the key field. Valid values are String, Float8, Float4, SignedBinary8, SignedBinary4, SignedBinary2, UnsignedBinary4, UnsignedBinary2, PackedDecimal, and ZonedDecimal.
<b>Key Length</b>	The length of the key field. The valid values are 0-256. For Sequential Data Queues, use a value of 0.
<b>Key Value</b>	The value used to search the Keyed data queue for an entry. Leave this field blank if you want to read from a sequential data queue.
<b>Key Compare Type</b>	The type of comparison to use to determine if a key is a match. Valid values are EQ (equal), NE (not equal), LT (less than), LE (less than or equal), GT (greater than), and GE (greater than or equal). Leave this field blank if you want to read from a simple data queue.


7. Click **Save Listener**. If the parameters are valid, the Data Queue Listener you created appears on the Data Queue Listeners screen.
8. Click **Return to AS/400 Adapter Listeners** to return to the Data Queue Listener screen.
9. To monitor the data queue, enable the Data Queue Listener. For more information on enabling data queue listeners, see [“Enabling Data Queue Listeners” on page 29](#).

## Viewing Data Queue Listeners

---

You can view the AS/400 Adapter Data Queue Listener from the Integration Server Administrator.

### ➤ To view the parameters for an AS/400 Data Queue Listener

1. Start the Integration Server Administrator if it is not already running.
2. Make sure the WmAS400 package is enabled.
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **AS/400 Adapter**.
4. In the AS/400 Adapter navigation area, click **Data Queue Listeners**.
5. In the Data Queue Listener screen, click the  icon for the listener you want to view.
6. The View Listener screen displays the parameters for the connection.
7. Click **Return to AS/400 Adapter Listeners** to return to the Data Queue Listener screen.

## Editing Data Queue Listeners


---

You can redefine the parameters that a Data Queue Listener uses when connecting to a AS/400 server using the Integration Server Administrator.

### Note:

You can edit a listener only if it is disabled. For instructions on disabling a Data Queue Listener, see [“Disabling Data Queue Listeners” on page 30](#).

### ➤ To edit an AS/400 Data Queue Listener

1. Start the Integration Server Administrator if it is not already running.
2. Make sure the WmAS400 package is enabled.
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **AS/400 Adapter**.
4. In the AS/400 Adapter navigation area, click **Data Queue Listeners**.
5. In the Data Queue Listener screen, click the  icon for the listener you want to edit.
6. The Edit Listener screen displays the parameters for the listener. Update the data queue listener parameters by typing or selecting the values you want to specify.

For descriptions of the listener parameters, see the table of parameters in [“Configuring Data Queue Listeners” on page 26](#).

7. Click **Save Changes** to save the listener parameters.
8. Click **Return to AS/400 Adapter Listeners** to return to the Data Queue Listener screen.


## Deleting Data Queue Listeners

If you no longer want to use an AS/400 Data Queue Listener, delete the listener.

### Note:

You can delete a listener only if the connection the listener is using is disabled. For instructions on disabling a connection, see [“Disabling Data Queue Listeners” on page 30](#).

### ➤ To delete an AS/400 Data Queue Listener

1. Start the Integration Server Administrator if it is not already running.
2. Make sure the WmAS400 package is enabled.
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **AS/400 Adapter**.
4. In the AS/400 Adapter navigation area, click **Data Queue Listeners**.
5. In the Data Queue Listener screen, click the  icon for the listener you want to delete.

The Integration Server deletes the listener.

## Enabling Data Queue Listeners

The Data Queue Listener must be enabled before it starts monitoring a data queue for data. When you create a listener, it is not automatically enabled.

### Note:

You cannot enable a listener if the connection alias the listener is using is disabled.

### ➤ To enable an AS/400 Data Queue Listener

1. Start the Integration Server Administrator if it is not already running.
2. Make sure the WmAS400 package is enabled.
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **AS/400 Adapter** to manage the AS/400 polling notifications.

4. In the AS/400 Adapter navigation area, click **Data Queue Listeners**.
5. On the listener screen, click **No** in the **Enabled** column for the listener you want to enable. The Integration Server Administrator or enables the listener and displays a ✓ and **Yes** in the **Enabled** column.

## Disabling Data Queue Listeners

---

The Data Queue Listeners must be disabled before you can modify or delete them.

**Note:**

When you disable a connection alias it also stops (not disables) any listener associated with the connection. When you enable a connection alias it starts all the associated listeners that are enabled.

➤ **To disable an AS/400 Data Queue Listener**

1. Start the Integration Server Administrator if it is not already running.
2. Make sure the WmAS400 package is enabled.
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **AS/400 Adapter**.
4. In the AS/400 Adapter navigation area, click **Data Queue Listeners**.
5. On the listener screen, click **Yes** in the **Enabled** column for the listener you want to disable. The Integration Server Administrator disables the listener and displays **No** in the **Enabled** column.

# 5 Adapter Logging and Exception Handling

---

■ Overview .....	32
■ Adapter Message Logging .....	32
■ AS/400 Adapter Exception Handling .....	32
■ Debugging or Setting up the Trace .....	33

## Overview

---

This chapter describes the message logging and AS/400 Adapter exception handling.

## Adapter Message Logging

---

The AS/400 Adapter uses the Integration Server's logging mechanism to log messages. You can configure and view the Integration Server's logs to monitor and troubleshoot the AS/400 Adapter. For detailed information about logging in the Integration Server, including instructions for configuring and viewing the different kinds of logs supported by the server, see the *webMethods Audit Logging Guide* for your release.

The Integration Server maintains several types of logs; however, the AS/400 Adapter only logs messages to the Error and Server logs, as described in the table below:

Log	Description
Error Log	The AS/400 Adapter automatically posts error-level log messages to the server's Error log.
Server Log	The AS/400 Adapter posts V1-Verbose1 or V2-Verbose2 level log messages to the Server log.

---

## AS/400 Adapter Exception Handling

---

The AS/400 Adapter throws two kinds of exceptions that you should be aware of as you build integrations using the adapter: `WmAS400Exception` and `WmAS400ConnectionException`. When creating a flow or a Java service that incorporates an AS/400 service, you might want to build logic into the wrapping service to catch and handle these types of exceptions.

### WmAS400Exception

The AS/400 Adapter throws a `WmAS400Exception` to report an error related to the adapter's logic, for example, configuration error, or error thrown when the object on which you are trying to perform an operation does not exist on the server, or when you do not have permission to perform the operation.

### WmAS400ConnectionException

The AS/400 Adapter throws a `WmAS400ConnectionException` to report a non-recoverable error in the connection to the AS/400 backend system. In this case, the integration logic should be written to catch this exception and try to create a new AS/400 connection object. The error is thrown, for example, while attempting a connection when the backend system is down, or when the service to which you have been connected is down.



---

## Debugging or Setting up the Trace

---

To enable Trace for your application, place the `jt400.properties` file in a `'com/ibm/as400/access'` directory pointed by the classpath. For information on the content, structure, and usage of the `'jt400.properties'` files, see the JTOpen toolkit API.

**Note:**

A sample file is provided with the AS/400 Adapter. Copy the *Integration Server\_directory* \packages\WmAS400\pub\as400trace.zip file into the following directory:

- *Integration Server\_directory* \lib\jars, if using Integration Server 9.0 and lower
- *Integration Server\_directory* \lib\jars\custom, if using Integration Server 9.5 and higher

By default, all the debug information is logged into the *Integration Server\_directory* \lib\as400.log file.



# 6 Externalizing Adapter Connection Assets

---

■ Externalizing Adapter Connection Assets .....	36
■ Externalizing AS400 Adapter .....	36

## Externalizing Adapter Connection Assets

---

The Microservices Runtime container allows to supply the runtime connection configurations for webMethods Non-ART based adapters while running the Docker image.

These adapter configurations are considered as the assets that are available in the application properties. Integration Server uses the adapter configurations as dynamic variables for initializing the connections. Microservices Runtime is correspondent with webMethods Integration Server.

The format of the configuration is:

```
PackageName.Connection.Configurations = value
```

For Example:

```
AS400Connections.CENTRAL.connectionFactoryType=CENTRAL
AS400Connections.CENTRAL.currentLibrary=IS712
AS400Connections.CENTRAL.initProgram=
AS400Connections.CENTRAL.password={AES}EE0RXY2uQZqAhuiclFokag\=\=
AS400Connections.CENTRAL.userId=sagcc
AS400Connections.COMMAND.connectionFactoryType=COMMAND
AS400Connections.COMMAND.currentLibrary=IS712
AS400Connections.COMMAND.initProgram=
AS400Connections.COMMAND.password={AES}EE0RXY2uQZqAhuiclFokag\=\=
AS400Connections.COMMAND.userId=sagcc
AS400Connections.Command1.connectionFactoryType=COMMAND
AS400Connections.Command1.currentLibrary=
AS400Connections.Command1.initProgram=
AS400Connections.Command1.password={AES}Ju79a+ad+0nP2a4NDi2Wzw\=\=
AS400Connections.Command1.userId=adapters
AS400Connections.DATABASE.connectionFactoryType=DATABASE
AS400Connections.DATABASE.currentLibrary=IS7123
AS400Connections.DATABASE.initProgram=
AS400Connections.DATABASE.password={AES}EE0RXY2uQZqAhuiclFokag\=\=
AS400Connections.DATABASE.userId=sagcc
```

You can modify the parameters in the template based on the respective adapter connection.

## Externalizing AS400 Adapter

---

As the connections vary from adapter to adapter, Microservices Runtime provides a way to deploy the configurations in the properties file. The properties file reflects the changes in the Docker image while running for the adapters.

Microservices Runtime uses the Password-Based Encryption technology (PBE) that is installed along with Microservices Runtime. PBE helps in handling the passwords and other sensitive datas.

1. Start the Microservices Runtime.
2. Run the Integration Server Administrator. The Integration Server Administrator connects to the Microservices Runtime.
3. Go to **Microservices > Configuration Variables > Generate Variable template**.

This downloads the `application.properties` file.

4. Go to **Adapters > webMethods Adapter for AS/400** in the left panel.
5. Enable the connection that requires externalization.
6. Click the view icon to review the connection details.
7. Edit the downloaded file and save it.
8. Run the Docker image.

### Example

When running a Microservices Runtime image in a Docker container, you can specify the configuration variables template in the `Docker run` command.

The following Docker run command uses the `SAG_IS_CONFIG_PROPERTIES` environment variable to specify the name and the location of the configuration variables template. In this example, the Docker image for the Microservices Runtime is named as `isimage:v10.3` and exposes ports 5555 and 9999. Additionally, the location of the `application.properties` file is accessible by the Docker image.

```
Docker run -d --name ARTFix6 -p 4455:5555 -p 9999 -v /home/EUR/kava/  
sag:/home/EUR/kava/sag  
-e SAG_IS_CONFIG_PROPERTIES=/home/EUR/kavp/sag/application.properties -e  
USER12=jack123 isimage:v10.3
```



# 7 Predefined Health Indicator for WmAS400

## Adapters

---

The following table describes the predefined health indicators for the WmAS400 adapter included with Microservices Runtime.

Indicator Name	Status	Description
Adapters	UP	It captures the connection details for WmAS400 adapter defined at runtime.

The following structure identifies the response schema for the connections:

```
alias
  connectionFactoryType
  connectionState
  currentLibrary
  initProgram
  packageName
  system
  userid
```





# 8 AS/400 Adapter Administrator APIs

---

■ Overview .....	42
■ Connection Operations .....	43
■ Listener Operations .....	46

## Overview

The administrator can manage adapter resources such as connections and listeners using Integration Server Administrator. Alternatively, you can use the REST APIs provided by the AS/400 Adapter Administrator API.

### REST URL Structure

An AS/400 Adapter Administrator API makes resources accessible through a URL path. All AS/400 Adapter Administrator API requests must be issued using HTTP or HTTPS. The AS/400 Adapter Administrator API supports the standard HTTP methods such as GET, POST, PUT, PATCH, and DELETE. Not all resources support all HTTP methods.

An absolute URL for an AS/400 Adapter Administrator API resource has the following structure:

```
http://host:port/admin/adapters/resourceType/{resourceId}
```

Where:

- http is the transport protocol. You can use HTTP or HTTPS.
- host:port is the host and port of the Integration Server on which you want to administer the adapter resources.
- admin/adapters is the directive for the adapter's Administrator APIs.
- resourceType is the type of resource, such as adapter connection, or adapter listener.
- resourceId identifies a specific resource.

For example:

```
HTTP GET http://host:port/admin/adapters/connection/AS400Adapter
```

Retrieves all the adapter connections for AS/400 Adapter.

### Common Request Parameters

The following request parameters are commonly used in REST URL to invoke the AS/400 Adapter Administrator APIs.

Parameter Name	Parameter Description
<i>adapterTypeName</i>	<p>Adapter type name registered with the Integration Server. Value is AS400Adapter for AS/400 Adapter.</p> <p><b>Note:</b> You can get the display name and adapter type name of all the adapters registered with Integration Server by invoking the Administrator API as follows:</p> <pre>{</pre>

Parameter Name	Parameter Description
	<pre>"method" : "get", "httpMethod" : "GET", "urlTemplate" : "/admin/adapters/", "input" : "", "output" : "" }</pre>
<i>enableValue</i>	<p>Possible values:</p> <ul style="list-style-type: none"> <li>■ <code>enable</code>. Enables adapter connection, or adapter listener.</li> <li>■ <code>disable</code>. Disables adapter connection, or adapter listener.</li> </ul>
<i>expandValue</i>	<p>Possible values:</p> <ul style="list-style-type: none"> <li>■ <code>true</code>. Displays the details of each adapter connection, or adapter listener.</li> <li>■ <code>false</code>. Default. Displays the name and URL of each adapter connection, or adapter listener.</li> </ul>
<i>connectionAlias</i>	Name of the adapter connection.
<i>connectionFactoryTypeName</i>	Name of the adapter connection factory type.
	<p><b>Note:</b> You can get the list of adapter connection factory types available for AS/400 Adapter by invoking the Administrator API as follows:</p> <pre>{   "method" : "describeConnectionFactory",   "httpMethod" : "GET",   "urlTemplate" : "/admin/adapters/describe/AS400Adapter/connection/",   "input" : "",   "output" : "" }</pre>
<i>listenerAlias</i>	Name of the adapter listener.
<i>listenerFactoryTypeName</i>	Name of the adapter listener factory type.
	<p><b>Note:</b> AS/400 Adapter does not have explicit listener factory types.</p>
<i>listenerTemplate</i>	Same as <i>listenerFactoryTypeName</i>

## Connection Operations

The following AS/400 Adapter's Administrator APIs are available for connection operations:

HTTP Method	Description	Template
GET	Lists the available adapter connection factory types for AS/400 Adapter.	<pre>{   "method" : "describeConnectionFactory",   "httpMethod" : "GET",   "urlTemplate" : "/admin/adapters/describe/ {adapterTypeName}/connection/",   "input" : "",   "output" : "" }</pre>
GET	Describes all the fields for AS/400 Adapter and the specified adapter connection factory type .	<pre>{   "method" : "describe",   "httpMethod" : "GET",   "urlTemplate" : "/admin/adapters/describe/ {adapterTypeName}/connection/ {connectionFactoryTypeName}/",   "input" : "",   "output" : "" }</pre>
GET	Retrieves the list of all available adapter connections for AS/400 Adapter.	<pre>{   "method" : "list",   "httpMethod" : "GET",   "urlTemplate" : "/admin/adapters/connection/ {adapterTypeName}/?expand={expandValue}"   "input" : "",   "output" : "" }</pre>
GET	Retrieves the fields and their values for the specified adapter connection.	<pre>{   "method" : "get",   "httpMethod" : "GET",   "urlTemplate" : "/admin/adapters/connection/ {adapterTypeName}/{connectionAlias}/",   "input" : "",   "output" : "" }</pre>
POST	Creates a new adapter connection.	<pre>{   "method" : "post",   "httpMethod" : "POST",   "urlTemplate" : "/admin/adapters/connection/",   "input" : "",   "output" : "" }</pre>
		<p><b>Note:</b> You can get the list of fields for an adapter connection by invoking the describe method in the AS/400 Adapter Administrator API.</p> <p>Sample for creating an adapter connection.</p> <pre>{   "adapterTypeName": "AS400Adapter",</pre>

HTTP Method	Description	Template
		<pre> "connectionFactoryType": "DATAQUEUE", "connectionAlias": "Test_Conn_DATAQUEUE", "system": "AS400ServerName", "user": "sagcc", "password": "sagccpass", "initProgram": "", "currentLibrary": "TESTLIB", "poolable": "true", "minimumPoolSize": "1", "maximumPoolSize": "10", "poolIncrementSize": "1", "blockingTimeout": "1000", "expireTimeout": "1000" } </pre>
		<p>For more details on input parameters and sample, see <a href="#">“Configuring Adapter Connections” on page 18.</a></p>
PATCH	Updates the specified adapter connection details.	<pre> { "method" : "update", "httpMethod" : "PATCH", "urlTemplate" : "/admin/adapters/connection/ {connectionAlias}", "input" : "", "output" : "" } </pre>
		<p><b>Note:</b></p> <ul style="list-style-type: none"> <li>■ You can get the list of fields for an adapter connection by invoking the <code>describe</code> method in the AS/400 Adapter Administrator API.</li> <li>■ You can get the existing fields values for the specific adapter connection by invoking the <code>get</code> method in the AS/400 Adapter Administrator API.</li> <li>■ You must specify all the fields for the adapter connection in the request body.</li> </ul>
		<p>For more details on input parameters and sample, see <a href="#">“Configuring Adapter Connections” on page 18.</a></p>
DELETE	Deletes the specified adapter connection.	<pre> { "method" : "delete", "httpMethod" : "DELETE", "urlTemplate" : "/admin/adapters/connection/ {connectionAlias}", "input" : "", "output" : "" } </pre>

HTTP Method	Description	Template
PUT	Enables or disables the specified adapter connections.	<pre>{   "method" : "stateChange",   "httpMethod" : "PUT",   "urlTemplate" : "/admin/adapters/connection/ {connectionAlias}/?action={enableValue}",   "input" : "",   "output" : "" }</pre>

## Listener Operations

The following AS/400 Adapter's Administrator APIs are available for adapter listener operations:

HTTP Method	Description	Template
GET	Lists the available adapter listener factory types for AS/400 Adapter.	<pre>{   "method" : "describeConnectionFactory",   "httpMethod" : "GET",   "urlTemplate" : "/admin/adapters/describe/ {adapterTypeName}/listener/",   "input" : "",   "output" : "" }</pre> <p><b>Note:</b> This AS/400 Adapter's Administrator API returns an empty value as the AS/400 Adapter does not have explicit listener factory types.</p>
GET	Describes all the fields for AS/400 Adapter.	<pre>{   "method" : "describe",   "httpMethod" : "GET",   "urlTemplate" : "/admin/adapters/describe/ {adapterTypeName}/listener/{listenerTemplate}/",   "input" : "",   "output" : "" }</pre> <p><b>Note:</b> The variable <i>listenerTemplate</i> is a dummy value as the AS/400 Adapter does not have explicit listener factory types. You can use this AS/400 Adapter's Administrator API to retrieve the fields and their default values for a listener.</p>

HTTP Method	Description	Template
GET	Retrieves the list of all available adapter listeners for AS/400 Adapter.	<pre>{   "method" : "list",   "httpMethod" : "GET",   "urlTemplate" : "/admin/adapters/listener/{adapterTypeName}/?expand={expandValue}",   "input" : "",   "output" : "" }</pre>
GET	Retrieves all the fields for the specified adapter listener.	<pre>{   "method" : "get",   "httpMethod" : "GET",   "urlTemplate" : "/admin/adapters/listener/{adapterTypeName}/{listenerAlias}/",   "input" : "",   "output" : "" }</pre>
POST	Creates a new adapter listener	<pre>{   "method" : "post",   "httpMethod" : "POST",   "urlTemplate" : "/admin/adapters/listener/",   "input" : "",   "output" : "" }</pre>

**Note:**

You can get the list of fields for an adapter listener by invoking the `describe` method in the AS/400 Adapter Administrator API.

Sample for creating an adapter listener.

```
{
  "adapterTypeName":"AS400Adapter",
  "connectionAlias":"Test_Conn_DATAQUEUE",
  "notificationNodeName":"WAS:listenerService",
  "notificationEnabled":"true",
  "notificationInterval":"-1",
  "dataQueueName":"/QSYS.LIB/TESTLIB.LIB/MYDATAQ.DTAQ",
  "isKeyedDataQueue":"false",
  "keyType":"String",
  "keyLength":"50",
  "keyValue":"NAME",
  "keyCompareType":"EQ"
}
```

For more details on input parameters and sample, see [“Configuring Data Queue Listeners” on page 26.](#)

HTTP Method	Description	Template
PATCH	Updates the specified adapter listener details.	<pre>{   "method" : "update",   "httpMethod" : "PATCH",   "urlTemplate" : "/admin/adapters/listener/ {listenerAlias}",   "input" : "",   "output" : "" }</pre> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>■ You can get the list of fields for an adapter listener by invoking the describe method in the AS/400 Adapter Administrator API.</li> <li>■ You can get the existing field values for the specific adapter listener by invoking the get method in the AS/400 Adapter Administrator API.</li> <li>■ You must specify all the fields for the adapter listener in the request body.</li> </ul> <p>For more details on input parameters and sample, see <a href="#">“Configuring Data Queue Listeners” on page 26.</a></p>
DELETE	Deletes the specified adapter listener.	<pre>{   "method" : "delete",   "httpMethod" : "DELETE",   "urlTemplate" : "/admin/adapters/listener/ {listenerAlias}",   "input" : "",   "output" : "" }</pre>
PUT	Enables or disables the specified adapter listener.	<pre>{   "method" : "stateChange",   "httpMethod" : "PUT",   "urlTemplate" : "/admin/adapters/listener/ {listenerAlias}/?action={enableValue}",   "input" : "",   "output" : "" }</pre>



# A Built-in Services

---

■ Overview .....	50
■ Access Services .....	50

## Overview

This chapter describes the built-in services provided with the webMethods AS/400 Adapter. These services are located in the WmAS400 package. Based on the functionality of these services, they are placed into different folders. The following table lists the folders that contain the built-in services based on their functionality:

Folder	Folder contains
access	Services that access the AS/400 resources.
admin	Services that manage the AS/400 connections.  <b>Important:</b> The services in this folder are used internally for the AS/400 Adapter Administrative screens.
rec	Contains the document types, integration specifications, and the IS schemas used by the WmAS400 package and the WmAS400Samples package.  <b>Important:</b> These document types are only for internal use and are not to be used in your integration solutions.

The AS/400 Adapter also provides sample services in the WmAS400Samples package. These sample services show the usage of the services. During installation, the WmAS400Samples package is installed along with the WmAS400 package.

## Access Services

You can use access services to formulate and submit requests to AS/400 servers. These services are available in the access folder. The following table lists the access services:

Service	Description
<a href="#">wm.as400.access:bytesToStringValue</a>	Converts a Byte Array of an AS/400 data type to String.
<a href="#">wm.as400.access:stringValueToBytes</a>	Converts a String into a Byte Array of AS/400 data type.
<a href="#">wm.as400.access:callCommand</a>	Calls an iSeries batch command. Allows the user to call a non-interactive iSeries command.
<a href="#">wm.as400.access:callProgram</a>	Calls an iSeries program. Allows the user to call an iSeries program.
<a href="#">wm.as400.access:connect</a>	Connects to an AS/400 service on the iSeries server.
<a href="#">wm.as400.access:disconnect</a>	Disconnects from the iSeries server.
<a href="#">wm.as400.access:convertAll</a>	Converts AS/400 data types into equivalent Java objects.

Service	Description
<a href="#">wm.as400.access:createDataQ</a>	Creates a sequential or keyed data queue on the iSeries server.
<a href="#">wm.as400.access:dataQEntryToRecord</a>	Converts Bytes read from a data queue into IData.
<a href="#">wm.as400.access:getConnection</a>	Gets a connection from an AS/400 connection pool.
<a href="#">wm.as400.access:returnConnection</a>	Returns a connection to the AS/400 connection pool.
<a href="#">wm.as400.access:readDataQ</a>	Reads data from a sequential or keyed data queue on the iSeries server.
<a href="#">wm.as400.access:writeDataQ</a>	Writes data to a sequential or keyed data queue on the iSeries server.

## wm.as400.access:bytesToStringValue

Converts a Byte Array (byte []) of AS/400 data type to a String containing the corresponding Java data type.

### Input Parameters

<i>\$AS400Alias</i>	<b>String</b> The connection alias of the AS/400 server.
<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> (Optional). A connected AS/400 object.  <b>Note:</b> Either <i>\$AS400</i> or <i>\$AS400Alias</i> should be provided as an input. The AS/400 object is used to find out the correct encoding to be used while converting byte[] of AS/400 data type String to Java String. If the input is not provided the default encoding is used to convert the string.
<i>AS400DataType</i>	<b>String</b> The AS/400 data type of the Byte Array that you would like to convert.
<i>AS400DataLength</i>	<b>String</b> The length of the AS/400 data type. For decimal types, type x.y, for example, 5.2.
<i>AS400ByteArray</i>	<b>Object</b> Byte Array (byte []) containing the AS/400 data type value.
<i>AS400DataOccurs</i>	<b>String</b> The number of times the datatype occurs in the <i>AS400ByteArray</i> .

### Output Parameters

<i>stringValue</i>	<b>StringList</b> The String value of AS/400 byte array.
--------------------	--

## Example

```
wm.as400.sample.conv.testByteArrayToStrArray
```

## wm.as400.access:stringValueToBytes

Converts a String value of AS/400 data type to a byte Array[] containing the corresponding AS/400 data type.

### Input Parameters

<i>\$AS400Alias</i>	<b>String</b> The connection alias of the AS/400 server.
<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> (Optional). A connected AS/400 object.  <b>Note:</b> Either <i>\$AS400</i> or <i>\$AS400Alias</i> should be provided as an input. The AS/400 object is used to find out the correct encoding to use while converting AS/400 data type of String to a byte[]. If they are not provided the default encoding is used to convert the string.
<i>AS400DataType</i>	<b>String</b> The AS/400 data type of the String value that you would like to convert.
<i>AS400DataLength</i>	<b>String</b> The length of the AS/400 data type. For decimal types, type x.y, for example, 5.2.
<i>stringValue</i>	<b>String</b> String containing the AS/400 data type value that you would like to convert.

### Output Parameters

<i>\$AS400Alias</i>	<b>String</b> The connection alias of the AS/400 server.
---------------------	--

---

## Example

```
wm.as400.sample.conv.testBytesToString
```

## wm.as400.access:callCommand

Calls an iSeries batch command. This service allows the users to call a non-interactive iSeries command.

## Input Parameters

<i>\$AS400Alias</i>	<b>String</b> The connection alias of the AS/400 server.
<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> Optional. A connected AS/400 object.  <b>Note:</b> Either <i>\$AS400</i> or <i>\$AS400Alias</i> should be provided as an input.
<i>commandName</i>	<b>String</b> The command to run on the server. If the command is not library qualified, the Server library list will be used to find the command.  <b>Example:</b>  CRTLIB  <b>Note:</b> If command is not in the Server library list then use the <i>libraryName</i> field to specify the command library else the <i>commandName</i> field should contain the fully qualified integrated file system path name of the command.  <b>Example:</b>  /QSYS.LIB/myLibrary.LIB/myCommand.CMD)
<i>libraryName</i>	<b>String</b> Optional. The library in which the command exists. It must be 1 to 10 characters.  <b>Note:</b> Use this field only if the command is not in the Server library list and the <i>commandName</i> does not contain the fully qualified integrated file system name for the command.  If you specify <i>libraryName</i> as %ALIAS_CURLIB% then the library name specified as the Current library while configuring the connection alias will be used.  <b>Note:</b> You cannot specify %ALIAS_CURLIB% if the field <i>\$AS400Alias</i> is empty.
<i>isThreadSafe</i>	<b>String</b> Set to true if the command should be assumed to be thread-safe on the server, else set to false.
<i>InputParameters</i>	<b>Record</b> Optional. Specifies the input fields (Name/Value Pair) for this command as follows:  <ul style="list-style-type: none"> <li>■ <b>Name.</b> The name of the input field, for example, LIB.</li> <li>■ <b>Value.</b> The value of the input field, for example, MYLIB.</li> </ul>

<i>additional Parameters</i>	<b>Record</b> Optional. Specifies the advanced input fields (Name/Value Pair) for this command as follows: <ul style="list-style-type: none"><li>■ <b>Name.</b> The name of the input field, for example, AUT.</li><li>■ <b>Value.</b> The Value of the input field, for example, *LIBCRTAUT.</li></ul>
------------------------------	---

---

## Output Parameters

<i>Success</i>	<b>String</b> The status of the operation. Contains true if the command call was successful else contains false.
<i>AS400MessageList</i>	<b>Record</b> List of AS/400 messages returned while running the command. It returns an empty list if there are no messages. It returns the following: <ul style="list-style-type: none"><li>■ <b>messageID.</b> The message ID of the AS/400 message.</li><li>■ <b>messageText.</b> The message text of the AS/400 message.</li></ul>

---

## Example

```
wm.as400.sample.CL:CRTLIB
```

## wm.as400.access:callProgram

Calls an iSeries program. This service allows a user to call an iSeries server program, pass the input and the output parameters, and access data returned in the output parameters after the program runs.

## Input Parameters

<i>\$AS400Alias</i>	<b>String</b> The connection alias of the AS/400 server.
<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> Optional. A connected AS/400 object. <b>Note:</b> Either <i>\$AS400</i> or <i>\$AS400Alias</i> should be provided as an input.
<i>programName</i>	<b>String</b> The name of the program to call. The program name must be 1 to 10 characters. <b>Example:</b> FIRSCL <b>Note:</b>

Use the *libraryName* field to specify the program library else the *programName* should contain the fully qualified integrated file system path name of the program.

**Example:**

/QSYS.LIB/myLibrary.LIB/myProgram.PGM

*libraryName*

**String** Optional. The library in which the program exists. It must be 1 to 10 characters.

**Note:**

Use this field only if *programName* does not contain the fully qualified integrated file system path name for the program.

If you specify *libraryName* as %ALIAS\_CURLIB%, then the library name specified as the current library while configuring the connection alias is used.

**Note:**

You cannot specify %ALIAS\_CURLIB% if the \$AS400Aliasfield is empty.

*isThreadSafe*

**String** Set to true if the program should be assumed to be thread-safe on the server, else set to false.

*inquiryMessageReply*

**String** Job attribute representing how a job answers inquiry messages. The possible values are:

- INQUIRY\_MESSAGE\_REPLY\_DEFAULT

A constant indicating that the system uses the default message reply to answer any inquiry messages issued while this job is running.

- INQUIRY\_MESSAGE\_REPLY\_SYSTEM\_REPLY\_LIST

A constant indicating that the system reply list is checked to see if there is an entry for an inquiry message issued while this job is running.

- INQUIRY\_MESSAGE\_REPLY\_REQUIRED

A constant indicating that the job requires an answer for any inquiry messages that occur while this job is running.

*maxByteSize*

**String** Specifies the size of the largest program parameter (either input or output) in bytes.

*RecordMetaData*

**Record** Definition of the program parameters (both input and output in the correct order) as follows:

- **DataName.** The name of the program parameter.

- **DataType.** The AS/400 data type of the program parameter.
- **DataLength.** The length of the program parameter. For decimal types type xx.yy, for example, 5.2.

---

<i>recordData</i>	<p><b>Record</b> Optional. Specifies the program parameter values (Name/Value Pair) as follows:</p> <ul style="list-style-type: none"> <li>■ <b>Name.</b> The name of the input parameter.</li> <li>■ <b>Value.</b> The value of the input parameter.</li> </ul>
-------------------	--

---

## Output Parameters

<i>Success</i>	<p><b>String</b> The status of the program. Contains true if the program call was successful else contains false.</p>
<i>AS400MessageList</i>	<p><b>Record</b> List of AS/400 messages returned while running the program. It returns an empty list if there are no messages. It returns the following fields:</p> <ul style="list-style-type: none"> <li>■ <b>messageID.</b> The message ID of the AS/400 message.</li> <li>■ <b>messageText.</b> The message text of the AS/400 message.</li> </ul>
<i>recordData</i>	<p><b>Record</b> Contains both the input and the output program parameter values (Name/Value Pair) as follows:</p> <ul style="list-style-type: none"> <li>■ <b>Name.</b> The name of the input/output parameter.</li> <li>■ <b>Value.</b> The value of the input/output parameter.</li> </ul>

---

## Example

wm.as400.sample.rpg:firstCL

## Usage Notes

- To invoke an RPG program, first find out the parameter definition of the RPG program and then build the input record RecordMetaData to exactly define the program parameters.

For example, if your program (FIRSTCL) takes one input parameter of type String and length 10 bytes and returns one output parameter of type String and length 2 bytes the RecordMetaData appears as follows:

```
recordMetaData
-----
recordMetaData[0]
-----
DataName      "IN"
```

---



	DataType	"String"
	DataLength	"10"
recordMetaData[1]		
	DataName	"OUT"
	DataType	"String"
	DataLength	"2"

Corresponding to the RecordMetaData, build the recordData structure to pass the input values.

recordData	
	IN

The name of the input variable should match the name in the *DataName* field defined in the RecordMetaData record. If the input field is not found in the recordData record, then the default value of that data type is sent to the program.

The field *maxByteSize* is equal to the absolute maximum value of the *DataLength* values in the input record RecordMetaData. In this case the value is:

maxByteSize	"10"
-------------	------

After execution, the record recordData contains the input and the output fields and the corresponding values returned by the programs.

recordData	
	IN
	OUT

- To invoke an RPG program that takes a Structure datatype as input or returns a Structure datatype as output, you also need to provide the meta-data information for this Structure data type.

For example, if your program (returnStructArray) returns one output parameter of array of Structure datatype, then the RecordMetaData looks like:

recordMetaData		
recordMetaData[0]		
	DataName	"StructField"
	DataType	"Structure"

DataLength	"10"
DataOccurs	5

For defining the StructField you need to define a Record List which looks like:

StructFieldMetaData
DataName
DataType
DataLength
DataOccurs

After execution, the record recordData contains the input fields, the output fields, and the corresponding values returned from the programs. In the above case, the recordData looks like:

recordData
StructField
StructField[0]
<Struct Member0> value
StructField[1]
<Struct Member1> value
StructField[n]
<Struct Membern> value

## wm.as400.access:connect

Allows a user to connect to an iSeries server. A connected com.ibm.as400.access.AS400 object is returned which can then be used to perform other operations on the server resources. Users must explicitly call the com.ibm.as400.access:disconnect service to disconnect from the server.

**Note:**

This service would create a new connection to the server every time. To obtain a connection from the connection pool and to use connection aliases use the service com.ibm.as400.access:getConnection.

## Input Parameters

<i>\$AS400System</i>	<b>String</b> The name of the AS/400 server.
<i>\$AS400UserId</i>	<b>String</b> The user profile name to use to authenticate to the server.
<i>\$AS400Password</i>	<b>String</b> The user profile password to use to authenticate to the server.
<i>\$AS400Service</i>	<b>String</b> The name of the AS/400 service to connect. The valid services are:

Service Name	Description
FILE	IFS file service
PRINT	Print service
COMMAND	Command and program call service
DATAQUEUE	Data queue service
DATABASE	JDBC service
RECORDACCESS	Record level access service
CENTRAL	License management service
SIGNON	Sign-on service

## Output Parameters

<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> A connected AS/400 object.
----------------	--

## Example

```
wm.as400.sample.connection:testNoPool
```

## wm.as400.access:disconnect

Disconnects a `com.ibm.as400.access.AS400` object from the iSeries server.

### Note:

Use this service to disconnect an AS/400 object that was obtained by calling `wm.as400.access:connect`. Do not use this service if the connection was obtained by calling `wm.as400.access:getConnection`.

## Input Parameters

<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> A connected AS/400 object.
----------------	--

## Output Parameters

None.

## Example

```
wm.as400.sample.connection:testNoPool
```

## wm.as400.access:convertAll

Allows a user to convert an array of AS/400 data types (bytes array) into an IData record containing the corresponding Java data types.

## Input Parameters

<i>RecordMetaData</i>	<b>Record</b> Definition of the program parameters (both input and output in the correct order) as follows: <ul style="list-style-type: none"><li>■ <b>洗DataName.</b> The name of the program parameter.</li><li>■ <b>洗DataType.</b> The AS/400 data type of the program parameter.</li><li>■ <b>洗DataLength.</b> The length of the program parameter. For decimal types type xx.yy, for example, 5.2.</li></ul>
<i>ByteValues</i>	<b>Object</b> An array of byte array's (byte[][]). Each element of the array is a byte array containing the values of the program parameters (both input and output).

---

## Output Parameters

<i>RecordData</i>	<b>Record</b> The program parameter values (Name/Value Pair) as follows: <ul style="list-style-type: none"><li>■ <b>洗Name.</b> The name of the input parameter.</li><li>■ <b>洗Value.</b> The value of the input parameter.</li></ul>
-------------------	--

---

## Example

```
wm.as400.sample.conv:convertAll
```

## wm.as400.access:createDataQ

Creates a sequential or keyed data queue on the iSeries server. If the field keyLength is equal to zero then a sequential data queue is created else a keyed data queue is created.

## Input Parameters

<i>\$AS400Alias</i>	<b>String</b> The connection alias of the AS/400 server.
<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> Optional. A connected AS/400 object.  <b>Note:</b> Either <i>\$AS400</i> or <i>\$AS400Alias</i> should be provided as an input.
<i>QName</i>	<b>String</b> The name of the data queue. The queue name must be 1 to 10 characters.  <b>Example:</b>  MYQUEUE  <b>Note:</b> Use the <i>libraryName</i> field to specify the Queue library or use the fully qualified integrated file system path name of the data queue.  <b>Example:</b>  /QSYS.LIB/myLibrary.LIB/MYQUEUE.DTAQ
<i>libraryName</i>	<b>String</b> Optional. The library in which the data queue exists. It must be 1 to 10 characters.  <b>Note:</b> Use this field only if the <i>QName</i> does not contain the fully qualified integrated file system path name for the data queue.  If you specify <i>libraryName</i> as %ALIAS_CURLIB%, then the library name specified as the current library while configuring the connection alias is used.  <b>Note:</b> You cannot specify %ALIAS_CURLIB% if the field <i>\$AS400Alias</i> is empty.
<i>keyLength</i>	<b>String</b> The length of the key field. In case of simple data queue type 0. Valid values are 0 to 256.
<i>maxEntryLength</i>	<b>String</b> The maximum number of bytes per data queue entry. Valid values are 1 to 64512. Default: 1000.
<i>authority</i>	<b>String</b> The public authority for the data queue. Valid values are *ALL, *CHANGE, *EXCLUDE, *USE, *LIBCRTAUT.
<i>saveSenderInformation</i>	<b>String</b> Set to true if the entry origin information are to be saved, else set to false.

<i>FIFO</i>	<b>String</b> Set to true if the queue entries are to be processed in the FIFO order. Set to false if the queue entries are to be processed in the LIFO order.
<i>forceToAuxiliaryStorage</i>	<b>String</b> If this value is true, the writeDataQ operation writes an entry to the data queue before return.
<i>description</i>	<b>String</b> The text description of the data queue. This string must be 50 characters or less.

---

## Output Parameters

<i>success</i>	<b>String</b> The status of the command. Contains true if the data queue is successfully created, else contains false.
----------------	--

---

## Examples

```
wm.as400.sample.simpleDQ:createDQ
```

```
wm.as400.sample.keyedDQ:createKeyDQ
```

## wm.as400.access:dataQEntryToRecord

Converts a data queue entry into an IData record.

### Note:

Use this service in a Data Queue Listener notification node (service invoked by the listener) to convert the data queue entry placed in the pipeline by the listener into an IData record.

## Input Parameters

<i>\$AS400Alias</i>	<b>String</b> The connection alias of the AS/400 server.
<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> Optional. A connected AS/400 object.

### Note:

Either *\$AS400* or *\$AS400Alias* should be provided as an input.

<i>dataQueueEntry</i>	One of the following: <ul style="list-style-type: none"><li>■ <b>com.ibm.as400.access.DataQueueEntry</b> In case the entry is read from a sequential data queue.</li><li>■ <b>com.ibm.as400.access.KeyedDataQueueEntry</b> In case the entry is read from a keyed data queue.</li></ul>
-----------------------	---

- **byte[]** byte array representation of the data queue entry. The Data Queue Listener places this queue entry in the pipeline after it has been read from the data queue.

---

*dataQueueEntryKey*      **byte[]** byte array representation of the key of the keyed data queue entry. The Data Queue Listener places this queue entry key in the pipeline after it has been read from the data queue.

---

*recordMetaData*      **Record** Definition of the record (*dataQueueEntry*) as follows:

- **洗DataName**. The name of the program parameter.
- **洗DataType**. The AS/400 data type of the program parameter.
- **洗DataLength**. The length of the program parameter. For decimal types type xx.yy, for example, 5.2.

---

*recordKey*      **Record** Definition of the key used to search the keyed data queue for an entry. Leave this field blank, if the entry was read from a sequential data queue.

- **洗KeyType**. The AS/400 data type of the key.
- **洗keyLength**. The length of the key.
- **洗KeyValue**. The key value used to search the data queue for an entry.

---

## Output Parameters

*success*      **String** The status of the operation. Contains true if the data queue read was successfully created else contains false.

---

*keyValue*      **String** The key for this data queue entry read from the keyed data queue.

---

*senderInfo*      **String** Sender information for this data queue entry.

---

*recordData*      **Record** The values (Name/Value Pair) corresponding to the *dataQueueEntry* read from the Queue. If no entry was read this record does not exist in the pipeline. The values are as follows:

- **洗Name**. The name of the record field.
- **洗Value**. The value of the record field.

---

## Examples

wm.as400.sample.simpleDQ:DQListener

wm.as400.sample.keyedDQ:DQListener

## wm.as400.access:getConnection

This service allows a user to obtain a connection from AS/400 connection pool. A connected `com.ibm.as400.access.AS400` object is returned which can then be used to invoke other commands on the server. Users must explicitly call `wm.as400.access:returnConnection` service to return the connection to the server pool.

### Input Parameters

<i>\$AS400Alias</i>	<b>String</b> The connection alias of the AS/400 server.
---------------------	--

---

### Output Parameters

<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> A connected AS/400 object.
<i>connKey</i>	<b>String</b> The string representation of the connection object.
<i>\$ALIAS_CURLIB</i>	<b>String</b> The current library configured while configuring the connection alias.

---

### Example

```
wm.as400.sample.connection:testPool
```

## wm.as400.acces:returnConnection

This service disconnects a `com.ibm.as400.access.AS400` object from the iSeries server.

**Note:**

Use this service to disconnect an AS/400 object that was obtained by calling `wm.as400.access:getConnection`. Do not use this service if a connection was obtained by calling `wm.as400.acces:connect`.

### Input Parameters

<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> A connected AS/400 object.
<i>\$AS400Alias</i>	<b>String</b> The connection alias of the AS/400 server.

---

### Output Parameters

<i>connKey</i>	<b>String</b> The string representation of the connection object.
----------------	---

---



## Example

wm.as400.sample.connection.testPool

## wm.as400.access:readDataQ

Reads an entry from the data queue and removes it from the queue. This service allows a user to read from both Sequential and Keyed data queues. If the input parameter byteKey is null, then the queue is assumed to be Sequential else it is assumed to be Keyed.

### Input Parameters

<i>\$AS400Alias</i>	<b>String</b> The connection alias of the AS/400 server.
<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> Optional. A connected AS/400 object.  <b>Note:</b> Either <i>\$AS400</i> or <i>\$AS400Alias</i> should be provided as an input.
<i>QName</i>	<b>String</b> The name of the data queue. The queue name must be 1 to 10 characters.  <b>Example:</b>  MYQUEUE  <b>Note:</b> Use the libraryName field to specify the QUEUE Library else the <i>QName</i> should contain the fully qualified integrated file system path name of the data queue.  <b>Example:</b>  /QSYS.LIB/myLibrary.LIB/MYQUEUE.DTAQ
<i>libraryName</i>	<b>String</b> Optional. The library in which the data queue exists. It must be 1 to 10 characters.  <b>Note:</b> Use this field only if the <i>QName</i> does not contain the fully qualified integrated file system path name for the program.  If you specify libraryName as %ALIAS_CURLIB% then the Library name specified as the current library while configuring the connection alias is used.  <b>Note:</b> You cannot specify %ALIAS_CURLIB% if the field <i>\$AS400Alias</i> is empty.

<i>wait</i>	<b>String</b> The number of seconds to wait if the queue contains no entries. Default: do not wait. If the default is Negative one (-1), wait until an entry is available.
<i>recordMetaData</i>	<b>Record</b> Definition of the record to be read from the queue as follows: <ul style="list-style-type: none"><li>■ <b>洗DataName.</b> The name of the record field.</li><li>■ <b>洗DataType.</b> The AS/400 data type of the record field.</li><li>■ <b>洗DataLength.</b> The length of the record field. For decimal types, type <i>xx.yy</i>, for example, 5.2.</li></ul>
<i>recordKey</i>	<b>Record</b> Definition of the key that is used to search the Keyed data queue for an entry. Leave this field blank if you want to read from a Sequential data queue. The values are as follows: <ul style="list-style-type: none"><li>■ <b>洗KeyType.</b> The AS/400 data type of the key.</li><li>■ <b>洗keyLength.</b> The length of the key.</li><li>■ <b>洗KeyValue.</b> The key value used to search the data queue for an entry.</li></ul>
<i>searchType</i>	<b>String</b> The type of comparison to use to determine if a key is a match. Valid values are EQ (equal), NE (not equal), LT (less than), LE (less than or equal), GT (greater than), and GE (greater than or equal). Leave this field blank if you want to read from a simple data queue.

## Output Parameters

<i>success</i>	<b>String</b> The status of the operation. Contains true if the data queue read was successfully created else contains false.
<i>keyValue</i>	<b>String</b> Key for this data queue entry read from the Keyed Data Queue.
<i>senderInfo</i>	<b>String</b> Sender information for the data queue entry.
<i>RecordData</i>	<b>Record</b> The values (Name/Value Pair) corresponding to the record read from the queue. If no entry was read this record does not exist in the pipeline. The values are as follows: <ul style="list-style-type: none"><li>■ <b>洗Name.</b> The name of the record field.</li><li>■ <b>洗Value.</b> The value of the record field.</li></ul>

## Examples

wm.as400.sample.simpleDQ:readDataQ

wm.as400.sample.keyedDQ:readKeyedData

## wm.as400.access:writeDataQ

Writes an entry to the data queue. This service allows a user to write to both Sequential and Keyed data queues.

### Input Parameters

<i>\$AS400Alias</i>	<b>String</b> The connection alias of the AS/400 server.
<i>\$AS400</i>	<b>com.ibm.as400.access.AS400</b> Optional. A connected AS/400 object.  <b>Note:</b> Either <i>\$AS400</i> or <i>\$AS400Alias</i> should be provided as an input.
<i>QName</i>	<b>String</b> The name of the data queue to write data to. The queue name must be 1 to 10 characters.  <b>Example:</b>  MYQUEUE  <b>Note:</b> Use the <i>libraryName</i> field to specify the QUEUE Library else the <i>Qname</i> should contain the fully qualified integrated file system path name of the data queue.  <b>Example:</b>  /QSYS.LIB/myLibrary.LIB/MYQUEUE.DTAQ
<i>libraryName</i>	<b>String</b> Optional. The library in which the data queue exists. It must be 1 to 10 characters.  <b>Note:</b> Use this field only if the <i>QName</i> does not contain the fully qualified integrated file system path name for the program.  If you specify <i>libraryName</i> as %ALIAS_CURLIB% then the library name specified as the current library while configuring the connection alias is used.  <b>Note:</b> You cannot specify %ALIAS_CURLIB%, if the field <i>\$AS400Alias</i> is empty.
<i>RecordMetaData</i>	<b>Record</b> Definition of the record to be written to the queue as follows: <ul style="list-style-type: none"> <li>■ <b>洗DataName.</b> The name of the record field.</li> <li>■ <b>洗DataType.</b> The AS/400 data type of the record field.</li> </ul>

- **DataLength.** The length of the record field. For decimal types type *xx.yy*, for example, 5.2.

---

*RecordData*

**Record** The values (Name/Value Pair) corresponding to the record to be written to the queue as follows:

- **Name.** The name of the record field.
- **Value.** The value of the record field.

---

*recordKey*

**Record** Definition of the key used to search the keyed data queue for an entry. Leave this field blank if the entry was read from a sequential data queue. The values are as follows:

- **KeyType.** The AS/400 data type of the key.
  - **KeyLength.** The length of the key.
  - **KeyValue.** The key value used to search the data queue for an entry.
- 

## Output Parameters

*Success*

**String** The status of the operation. Contains true if the data queue read was successfully created else contains false.

---

## Examples

wm.as400.sample.simpleDQ:writeDataQ

wm.as400.sample.keyedDQ:writeKeyedData