webMethods Adapter for OPC® Installation and User's Guide

Version 10.3

November 2018

WEBMETHODS

This document applies to webMethods Adapter for OPC 10.3 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 © 2018-2022 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-OPC-IUG-103-20221202

Table of Contents

About this Guide

This guide describes how to configure and use webMethods Adapter for OPC®. It contains information for administrators and application developers who want to exchange data with the OPC Unified Architecture (OPC UA) server.

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

- The basic concepts and tasks for working with OPC UA server.
- Creating flow or Java services
- 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 *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	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 the Adapter

About Adapter for OPC

Adapter for OPC is an add-on to webMethods Integration Server that enables you to exchange data with OPC UA server. The adapter provides seamless and real-time communication with the OPC UA server without requiring changes to your existing application infrastructure.

Using Adapter for OPC, Integration Server can create and run services that execute operations to retrieve node information, update node value attribute, and execute method of a node. Also, subscribe to any data change or event notifications on the OPC UA server.

Architecture Overview

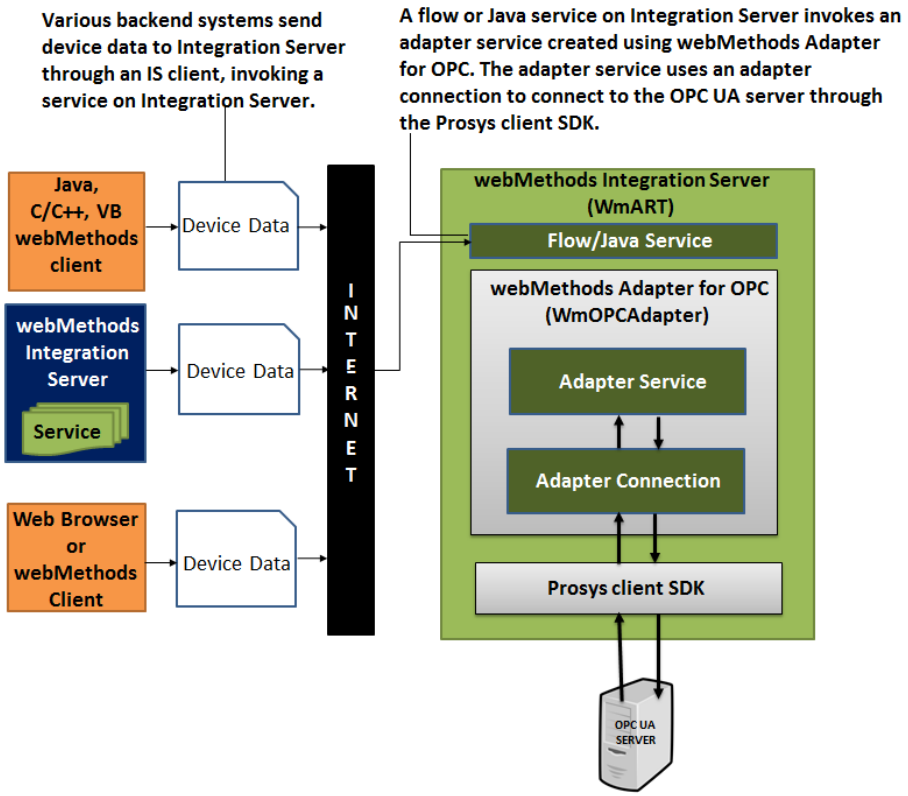
Adapter for OPC provides a set of user interfaces, services, and templates that enable you to create integrations with the OPC UA server. The adapter is provided as a single package that must be installed on Integration Server. For detailed installation instructions, see "Overview of Installing and Uninstalling Adapter for OPC" on page 20. For software requirements, see *webMethods Adapters System Requirements* .

Adapter for OPC enables you to configure the following components:

- **Adapter connections.** Enable Integration Server to connect to OPC UA server at design and runtime. You must configure an adapter connection before you can configure adapter services or adapter notifications. For a detailed description of adapter connections, see "Adapter Connections" on page 13.
- **Adapter services.** Enable Integration Server to initiate and perform operations on OPC UA server.

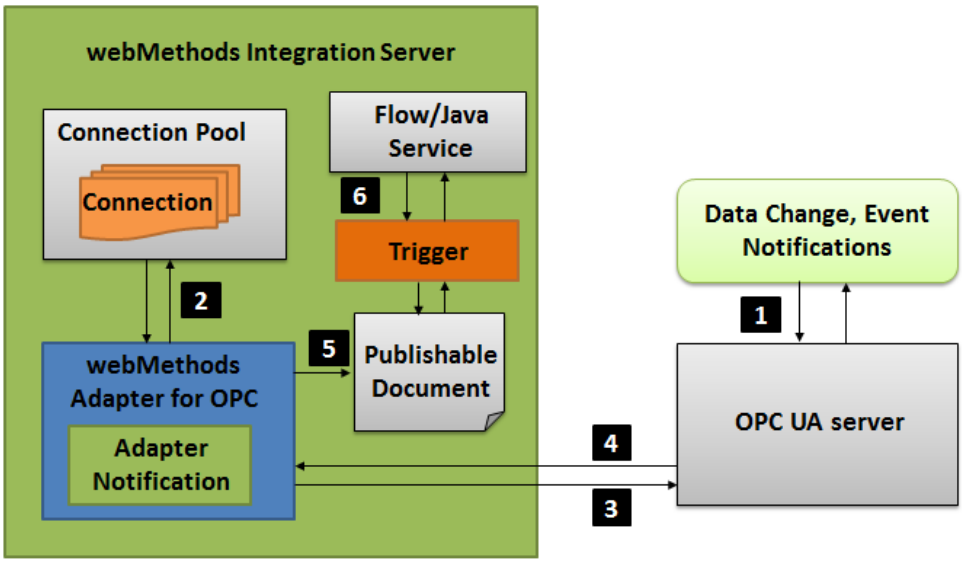
  For example, in a manufacturing unit, an adapter service can be used to read the sensor for current temperature and another service can be used to start and stop a boiler. You can configure adapter services using adapter service templates, which are provided with Adapter for OPC. For more information, see "Adapter Services" on page 14.

  The following diagram illustrates a business integration where an adapter service is used to update the OPC UA server with the device data. The device data can be provided by several different types of external Integration Server (IS) clients.



- **Adapter Listener and Listener notifications.** Register MonitoredItem to subscription and notify Integration Server when a notification message is generated by OPC UA server. For more information, see "Adapter Listeners and Listener Notification" on page 17.

The following diagram illustrates what happens when notifications are invoked.



Step	Description
1	Data Change Notification monitors the data change in variable values. Event Notification monitors the occurrence of an event.
2	The Adapter notification gets a connection from the connection pool of the service. Adapter connections contain the connection information for OPC UA server.
3	You created and enabled the adapter connection earlier using Integration Server Administrator.
4	The Adapter notification retrieves the data from the OPC UA server.
5	Software AG Designer creates the publishable document, which contains the data from the adapter. The notification publishes the publishable document. For more information on Integration Server publishable documents, see *Publish-Subscribe Developer's Guide*.
6	Configure an Integration Server trigger to use the notification's publishable document. Using this Integration Server trigger a flow or Java service on Integration Server is invoked to react to the changes contained in the publishable document.

The primary difference between the types of integrations is that notifications are initiated by events that occur on the OPC UA server, not by actions that occur on Integration Server.

With adapter notifications, you can capture notification data from the OPC UA server and use it to initiate another action within Integration Server. For example, you could create an adapter notification to register MonitoredItem for Sensor temperature and when the temperature changes, a notification message is generated. You can post the temperature data to webMethods Broker. Broker clients can then subscribe to notification's publishable document.

For more information about the architecture for the different types of adapter listener and listener notifications, see "Adapter Listeners and Listener Notification" on page 17.

Package Management

Adapter for OPC is provided as a package called WmOPCAdapter that 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 :

- You must create user-defined packages for your connections, adapter services, and notifications. For details, see " Package Management" on page 12.
- You should understand how package dependencies work so you make the best decisions regarding how you manage your adapter services and notifications. For details, see "Package Dependency Requirements and Guidelines" on page 25.

- You control which development groups have access to which adapter services and notifications. For details, see "Group Access Control" on page 27.

Adapter Connections

Adapter for OPC connects to a OPC UA server at runtime. You create one or more connections at design-time to use in integrations. The number of connections you create, and the types of those connections, depend on the types of security and security level your integration needs. For example, you can create connection without any security policy or security level with **None**.

Adapter for OPC connections contain parameters that Integration Server uses to manage connections to the OPC UA server so that they can be used by the adapter to provide services. You configure connections using Integration Server Administrator. You must have Integration Server Administrator privileges to access Adapter for OPC's administrative screens.

For instructions on configuring, viewing, editing, enabling, and disabling Adapter for OPC connections, see "Overview of Adapter Connections" on page 30. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide*.

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. All adapter services use connection pooling.

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

Run-Time Behavior of Connection Pools

When you enable a connection, Integration Server initializes the connection pool, creating the number of connection instances you specified in the connection's **Minimum Pool Size** field when you configured the connection. Whenever an adapter service needs a connection, Integration Server provides a connection from the pool. If no connections are available in the pool, and the maximum pool size has not been reached, the server creates one or more new connections (according to the number specified in the **Pool Increment Size** field) and adds them to the connection pool. If the pool is full (as specified in **Maximum Pool Size** field), the requesting service waits 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 the **Expire Timeout** field.

If initialization of the connection pool fails because of a network connection failure or some other type of exception, you can enable the system to retry the initialization any number of times, at specified intervals. For information about configuring connections, see "Configuring Adapter for OPC Connections" on page 30.

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
- To return usage statistics
- To return the current state (Enabled or Disabled)
- To return the error status for a connection

These services are located in the WmART package, in the pub.art.connection folder.

The setAdapterServiceNodeConnection and setPollingNotificationNodeConnection built-in services enable you to change the connection associated with an adapter service or notification respectively. For more information, see "Changing the Connection Associated with an Adapter Service or Notification at Design-Time" on page 16.

For details, see the *webMethods Integration Server Built-In Services Reference*.

Adapter Services

Adapter services allow you to connect to the adapter's resource and initiate an operation on the resource from Integration Server. To use Adapter for OPC you create adapter services.

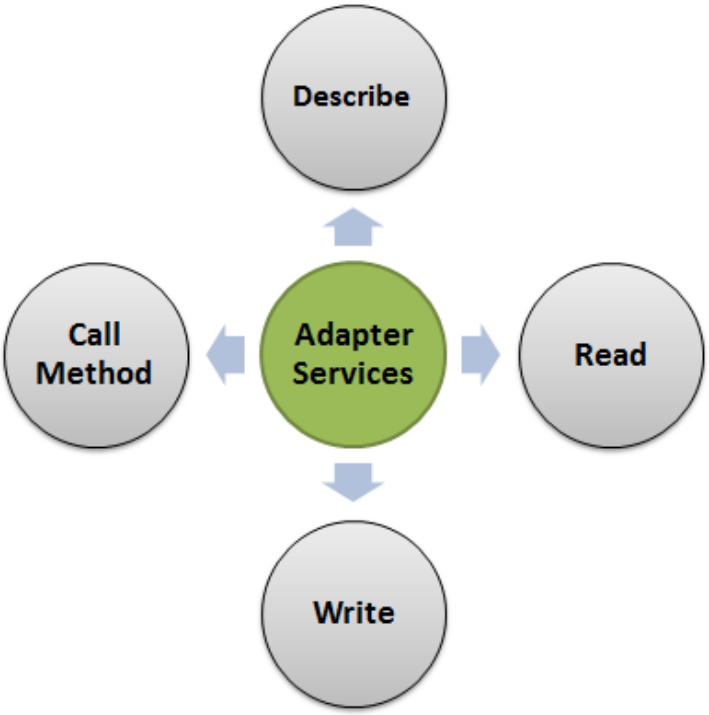
Integration Server then uses adapter connections that you defined earlier to execute the adapter services.

Adapter services are based on templates provided with Adapter for OPC. Each template represents a specific technique for working on a resource, such as using the Describe service to retrieve node references.

An adapter service template contains all the necessary code required to work with the resource. These templates can be customized with specific data sets to create a new service.

Some familiarity with using Software AG Designer is required. For more information, see the *webMethods Service Development Help*.

Adapter for OPC provides the following adapter service templates:



Adapter Service Type	Adapter Template Name	Description
Describe	Describe	Helps you navigate through the OPC UA server address space. Provides selected node details and references. For instructions about configuring the service, see "Configuring Describe Service" on page 40
Read	Read	Reads node attributes. For instructions about configuring the service, see "Configuring Read Service" on page 42
Write	Write	Writes or updates node value. For instructions about configuring the service, see "Configuring Write Service" on page 44

Adapter Service Type	Adapter Template Name	Description
Call Method	Call Method	Executes a method or an operation defined on a node. For instructions about configuring the service, see "Configuring Call Method Service" on page 49

Using Adapter Services

The tasks required to use adapter services are as follows:

1. Create an adapter connection using Integration Server Administrator. For details, see "Overview of Adapter Connections" on page 30

2. Select the appropriate adapter service template and configure the adapter service using Software AG Designer.

   Depending on the type of adapter service, you specify:

   - The adapter connection
   - The node
   - Node attribute to read or update.
   - Node's method to execute.
   - The input fields and types as needed
   - The output fields and types as needed

   For more information about configuring adapter services, see "Overview of Adapter Services" on page 40.

3. Design the Integration Server flow or Java service to invoke the adapter service using Software AG Designer.

   Use Designer and Integration Server Administrator to manage the adapter service. For more information, see "Overview of Package Management" on page 24 and "Overview of Adapter Services" on page 40.

Changing the Connection Associated with an Adapter Service or Notification at Design-Time

Integration Server provides built-in services that you can use at design-time to change the connection associated with an adapter service or notification. The built-in services,

setAdapterServiceNodeConnection and setPollingNotificationNodeConnection, are provided in the WmART package's pub.art.service folder and pub.art.notification folder, respectively. Using this function, you can change a specific connection associated with an adapter service or an adapter notification at design-time. So that you can eliminate maintaining multiple adapter services and notifications.

Note:
The setAdapterServiceNodeConnection and setPollingNotificationNodeConnection services can be run at design-time only. Do not use them within an Integration Server flow or Java service. You must run the services directly from Software AG Designer by selecting a service and running it.

For details, see the *webMethods Integration Server Built-In Services Reference*.

Other built-in services enable you to control connections. For more information, see "Built-In Services for Connections" on page 14.

Changing the Connection Associated with an Adapter Service at Runtime

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 OPC UA server1. However, at runtime you can override the default connection and instead use another connection to interact with OPC UA server2.

For more information about overriding a service's default connection at runtime, see "Dynamically Changing a Service's Connection at Run Time" on page 34.

Adapter Listeners and Listener Notification

Adapter for OPC provides listeners and listener notifications to perform the following functions:

- Monitor Data Changes
- Monitor Alarms or Events

Listeners

The listener in the Adapter for OPC represents a subscription. When an OPC Adapter listener is created, a subscription Object gets created in the OPC UA server address space. To subscribe to a node for data change or get events from any event source you must create an OPC Adapter listener notification and add appropriate nodes.

Using an Adapter for OPC connection, an Adapter for OPC listener waits to receive a message from OPC UA server. When a message or event appears for a node that the listener is subscribed to, OPC UA server pushes the message to the listener. The listener then passes the message to a listener notification.

You must never invoke a listener directly from a service or client. Instead, use the Integration Server Administrator to configure, enable, and disable the services.

All listeners function in the following manner:

- All listeners stop functioning when the package containing the listener node is disabled or when Integration Server shuts down.
- All listeners start functioning when the package containing the listener node is enabled or when Integration Server restarts.

Listener Notification

A listener notification works in conjunction with a listener to process messages in the Adapter for OPC. When a listener receives a message from the OPC UA server, the listener passes the message to an enabled listener notification that you associated with the listener. For more information about enabling listener notification, see "Enabling Listeners" on page 56.

Note:
The message is lost, if you do not configure any notifications with a listener, or if you do not enable any of the already configured notifications. Software AG recommends that you always start a listener notification before a listener. Create separate listeners to receive notifications of different types. you should configure only one notification per listener. If you configured multiple notifications with a listener, enable only the notification that you require.

When you create a listener notification, Adapter for OPC creates a publishable document type. At run time, after the listener receives a message from the server, the listener invokes the notification. The notification publishes the document created with the notification. A listener notification can publish a document in either of the following ways:

- Publish to a JMS queue or topic when Integration Server is connected to a JMS provider.
- To a local instance of Integration Server when it is not connected to a JMS provider.

When you subscribe to a specific listener notification, a publishable document type is created. You must subscribe to an Integration Server trigger to receive these notifications. For more information about using triggers with services, see the *Publish-Subscribe Developer's Guide*.

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

2 Installing and Uninstalling Adapter for OPC

Overview of Installing and Uninstalling Adapter for OPC

This chapter explains how to install, upgrade, and uninstall webMethods Adapter 10.3 for OPC. 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*.

Requirements

For a list of operating systems, webMethods products supported by Adapter for OPC, see *webMethods Adapters System Requirements* .

Adapter for OPC has no hardware requirements beyond those of its host Integration Server.

The Integration Server Home Directory

Beginning with Integration Server 10.3, 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*.

This guide uses the packages_directory as the home directory in Integration Server classpaths. For Integration Server 10.3 and above, the packages_directory is *Integration Server_directory* \instances*instance_name*\packages directory.

Installing Adapter for OPC

Install the latest Adapter Runtime and Designer fix to use this adapter.

To install Adapter for OPC

1. Download Installer from the Empower Product Support website.

2. If you are installing the adapter on an existing Integration Server, shut down the Integration Server.

3. Start the webMethods wizard.

4. Choose the webMethods release that includes the Integration Server on which you want to install the adapter. For example, if you want to install the adapter on Integration Server 10.3, choose the 10.3 release.

5. Specify the installation directory as follows:

- If you are installing on an existing Integration Server, specify the webMethods installation directory that contains the host Integration Server.
- If you are installing both the host Integration Server and the adapter, specify the installation directory to use.

6. In the product selection list, select **Adapters > webMethods Adapter for OPC**

   If you are using Integration Server 10.3 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 the installation completes, close the Installer.

9. Adapter for OPC uses ProSys Java SDK client library to communicate with OPC UA Sever. Copy the ProSys Java SDK client .jar files to *Integration Server_directory* \instances\ instance_name \packages\ WmOPCAdapter\code\jars directory.

   Note:
   To enable the Adapter connection, it is necessary to add the jar files.

10. Start the host Integration Server.

Installing Adapter for OPC using Microservices Container

To install Adapter for OPC using Microservices container

1. Download Installer from the Empower Product Support website.

2. If you are installing the adapter on an existing Integration Server, shut down the Integration Server.

3. Start the Installer wizard.

4. 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.

5. In the product selection list, select **Adapters > webMethods Adapter 10.3 for OPC**.

   From the Software AG Installer dialogue box, select the **Microservices Container 10.1**.

6. Expand Infrastructure and then Libraries.

   In the expanded list of options in libraries, select the **Database Driver Libraries 10.1** check box.

7. To download the documentation for the adapter, go to Software AG Documentation website.

8. After the installation completes, close the Installer and start the host Integration Server.

9. For more information on Microservices Container, see *Developing Microservices with webMethods Microservices Runtime*.

Uninstalling Adapter for OPC

Perform the following steps to uninstall Adapter for OPC.

To uninstall Adapter for OPC

1. Shut down the host Integration Server. You do not need to shut down any other webMethods products or applications that are running on your machine.

2. Start Software AG Uninstaller, selecting the webMethods installation directory that contains the host Integration Server.

3. In the product selection list, select **Adapters > webMethdods Adapter 10.3 for OPC**. You can also choose to uninstall documentation.

4. After Uninstaller completes, restart the host Integration Server.

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

3 Package Management

Overview of Package Management

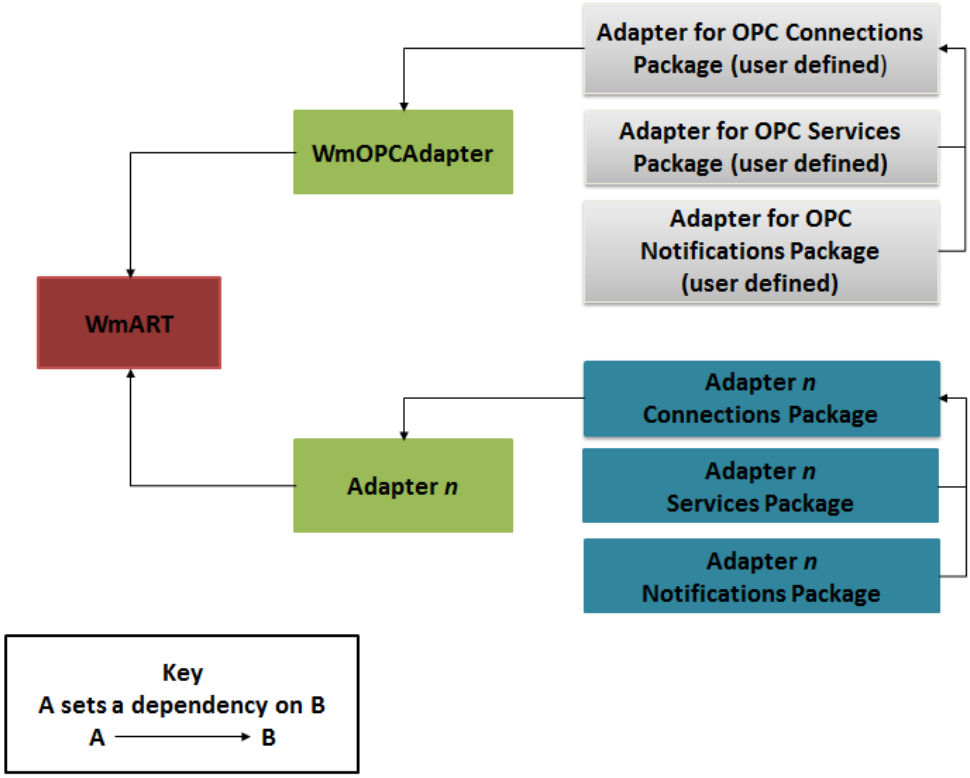
The following sections describe how to set up and manage your Adapter for OPC packages, set up Access Control Lists (ACLs).

Adapter for OPC Package Management

Adapter for OPC is provided as a package called WmOPCAdapter. You manage the WmOPCAdapter package as you would manage any package on webMethods Integration Server.

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

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



Package management tasks include:

- Setting package dependencies (see "Package Dependency Requirements and Guidelines" on page 25)
- "Enabling Packages" on page 25

- "Importing and Exporting Packages" on page 26
- "Group Access Control" on page 27

Package Dependency Requirements and Guidelines

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

- A user-defined package must have a dependency on its associated adapter package, WmOPCAdapter. (The WmOPCAdapter package has a dependency on the WmART package.)
- Package dependencies ensure that at startup the 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 automatically installed when you install Integration Server. You should not need to manually reload the WmART package.
- If the connections and adapter services of an adapter are defined in different packages, then:
  - A package that contains the connections must have a dependency on the adapter package.
  - Packages that contain adapter services must have a dependency on their associated connection package.
- Keep connections for different adapters in separate packages so that you do not create interdependencies between adapters. If a package contains connections for two different adapters, and you reload one of the adapter packages, the connections for both adapters will reload automatically.
- Integration Server will not allow you to enable a package if it has a dependency on another package that is disabled. That is, before you can enable your package, you must enable all packages on which your package depends. For information about enabling packages, see "Enabling Packages" on page 25.
- 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 26.
- You can name connections, adapter services, and notifications the same name provided that they are in different folders and packages.

Enabling Packages

All packages are automatically enabled by default. Use the following procedure when you want to enable a package that was previously disabled.

To enable a package

1. Open Integration Server Administrator if it is not already open.

2. In the **Packages** menu of the navigation area, click **Management**.

3. Click **No** in the **Enabled** column. The server displays a ✔ and **Yes** in the **Enabled** column.

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

> **Important:**
> Before you manually enable a user-defined package, you must first enable its associated adapter package (WmOPCAdapter).

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.

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

To disable a package

1. Open Integration Server Administrator if it is not already open.

2. In the **Packages** menu of the navigation area, click **Management**.

3. Click **Yes** in the **Enabled** column for the package that you want to disable. The server issues a prompt to verify that you want to disable the package. Click **OK** to disable the package. When the package is disabled, the server displays **No** in the **Enabled** column.

A disabled adapter has the following behavior:

- Remains disabled until you explicitly enable it using Integration Server Administrator.
- Is not listed in Software AG Designer.

Importing and Exporting Packages

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

> **Important:**

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

For details about importing and exporting packages, see *webMethods Service Development Help*.

Group Access Control

To control which groups have access to which adapter services, use access control lists (ACLs). For example, 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 information about assigning and managing ACLs, see the *webMethods Service Development Help*.

4 Adapter for OPC Connections

Overview of Adapter Connections

This chapter describes how to configure and manage Adapter for OPC connections. For more information about how adapter connections work, see "Adapter Connections" on page 13.

Preparing to Configure or Manage Adapter Connections

Perform the following steps before configuring or managing adapter connections.

To prepare to configure or manage adapter connections

1. Install webMethods Integration Server and Adapter for OPC on the same machine. For details, see "Overview of Installing and Uninstalling Adapter for OPC" on page 20.

2. Make sure you have Integration Server administrator privileges so that you can access Adapter for OPC 's administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide*.

3. Start your Integration Server and Integration Server Administrator, if they are not already running.

4. Using Integration Server Administrator, make sure the WmOPCAdapter package is enabled. For instructions, see "Enabling Packages" on page 25.

5. Using Software AG Designer, create a user-defined package to contain the connection, if you have not already done so. For more information about managing packages for the adapter, see "Adapter for OPC Package Management" on page 24.

Configuring Adapter for OPC Connections

When you configure Adapter for OPC connections, you specify information that Integration Server uses to connect to a OPC UA server. You can configure Adapter for OPC connections either manually using the Integration Server Administrator screen or programmatically using the service.

To configure an adapter connection

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

2. On the Connections page, click **webMethods Adapter for OPCConfigure New Connection**.

3. On the Connection Types page, click **webMethods Adapter for OPCConnection** to display the Configure Connection Type screen.

4. In the **webMethods Adapter for OPC** section, use the following fields:

Field	Description/Action
Package	The package in which to create the connection. You must create the package using Software AG Designer before you can specify the package using this parameter. For general information about creating packages, see the *webMethods Service Development Help*. **Note:** Configure the connection in a user-defined package rather than in the adapter's package. For other important considerations when creating packages for Adapter for OPC, see " Package Management" on page 12.
Folder Name	The folder in which to create 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 and Software AG Designer. For more information about the use of special characters in package, folder, and element names, see the *webMethods Service Development Help*.

5. In the **Connection Properties** section, use the following fields:

Field	Description/Action
Server URI	The server Uniform Resource Identifier (URI) to get endpoints. **Note:** The **Endpoint URI**, **Security Mode**, **Security Policy** are empty for a new connection. Click **LookUp** to fill the fields. This field is editable.
Endpoint URI	The Endpoint URI to be connected.
Security Mode	The different types of security levels used in OPC UA TCP communication are: ■ **None.** No encryption selected. ■ **Sign.** The binding of message source to the message by message signing. ■ **SignAndEncrypt.** Provides both binding of message source to the message by message signing and confidentiality by encryption. **Note:** For a secure Adapter for OPC connection, the adapter accepts server certificate permanently, which is stored in the folder `Integration Server_directory\instances\instance_name\PKI\CA\certs`.
Security Policy	The security algorithms used to encrypt the message are as follows:

Field	Description/Action
	■ **None.** No security algorithm is used to encrypt the message. ■ **Basic256.** The 256-bit encryption suite of algorithms. ■ **Basic256Sha256.** The encryption is used for configurations with high security needs. ■ **Basic128Rsa15.** The encryption is used for configurations with medium security needs. For more information on **Security Policy** see, OPC UA documentation.
Authentication Mode	The type of user authentication modes that the connection provides. Select one of the following authentication modes: **Certificate** ■ **User Identity Keystore.** For certification based user authentication, user requires **User Identity Keystore**. For more information on Keystore, see *webMethods Integration Server Administrator's Guide*. ■ **User Identity Keys.** For certification based user authentication, user requires **User Identity Key**. For more information on Keys, see *webMethods Integration Server Administrator's Guide*. **UserName** ■ **Username** The user name that the connection uses to connect to the OPC UA server. ■ **Password** The password for the user name. ■ **Retype Password** Retype the password you just entered. **Anonymous** Anonymously connect to OPC UA server. You can use it to check the connection.
Application Identity Keystore	For certification based application authentication, user requires **Application Identity Keystore**. For more information on Keystore, see *webMethods Integration Server Administrator's Guide*.
Application Identity Key	For certification based application authentication, user requires **Application Identity Key**. For more information on Keystore, see *webMethods Integration Server Administrator's Guide*.

The common properties are:

Field	Description/Action
Service Timeout(ms)	Defines the number of seconds that Adapter for OPC waits for a response from OPC UA server before reporting a timeout. Default: 120000
Session Timeout (ms)	Maximum timeout for the session created by the connection.
Auto Reconnect	If the connection is broken and **Auto Reconnect** is True, then the connection attempts to reconnect to the server.

6. In the **Connection Management Properties** section, use the following fields:

Field	Description/Action
Enable Connection Pooling	Enables the connection to use connection pooling. For more information about connection pooling, see "Adapter Connections" on page 13. **Note:** If you plan to enable connection pooling in a clustered environment, consider the connection pool size.
Minimum Pool Size	If connection pooling is enabled, this field specifies the number of connections to create when the connection is enabled. The adapter will keep open the number of connections you configure here regardless of whether these connections become idle.
Maximum Pool Size	If connection pooling is enabled, this field specifies the maximum number of connections that can exist at one time in the connection pool.
Pool Increment Size	If connection pooling is enabled, this field specifies the number of connections by which the pool will be incremented if connections are needed, up to the maximum pool size.
Block Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that Integration Server will wait to obtain a connection with the OPC UA server before it times out and returns an error. For example, you have a pool with **Maximum Pool Size** of 20. If you receive 30 simultaneous requests for a connection, 10 requests will be waiting for a connection from the pool. If you set the **Block Timeout** to 5000, the 10 requests will wait for a connection for 5 seconds before they time out and return an error. If the services using the connections require 10 seconds to complete and return connections to the pool, the pending requests will fail and return

Field	Description/Action
	an error message stating that no connections are available. If you set the **Block Timeout** value too high, you may encounter problems during error conditions. If a request contains errors that delay the response, other requests will not be sent. This setting should be tuned in conjunction with the **Maximum Pool Size** to accommodate such bursts in processing.
Expire Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that an inactive connection can remain in the pool before it is closed and removed from the pool. The connection pool will remove inactive connections until the number of connections in the pool is equal to the **Minimum Pool Size**. The inactivity timer for a connection is reset when the connection is used by the adapter. If you set the **Expire Timeout** value too high, you may have a number of unused inactive connections in the pool. This consumes local memory and a connection on your backend resource. This could have an adverse effect if your resource has a limited number of connections. If you set the **Expire Timeout** value too low, performance could degrade because of the increased activity of creating and closing connections. This setting should be tuned in conjunction with the **Minimum Pool Size** to avoid excessive opening/closing of connections during normal processing.
Startup Retry Count	The number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails. The default is `0`.
Startup Backoff Timeout	The number of seconds that the system should wait between attempts to initialize the connection pool.

7. Click **Save Connection**.

   The connection you created appears on the adapter's Connections screen and in Software AG Designer.

   You can enable a connection only if the parameters for the connection are valid.

Dynamically Changing a Service's Connection at Run Time

You can run a service using a connection other than the default connection that was associated with the service when the service was created.

For more information, see "Changing the Connection Associated with an Adapter Service at Runtime" on page 17.

Viewing Adapter Connection Parameters

You can view a connection's parameters from Integration Server Administrator and Software AG Designer.

Using Integration Server Administrator to View Adapter Connection Parameters

Perform the following steps to view adapter connection parameters in Integration Server Administrator.

To view the parameters for a connection using Integration Server Administrator

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

   When using the adapter with Integration Server 10.3 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**.

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

   The View Connection screen displays the parameters for the connection. For descriptions of the connection parameters, see "Configuring Adapter for OPC Connections" on page 30.

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

Using Software AG Designer to View Adapter Connection Parameters

Perform the following steps to view adapter connection parameters in 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 for OPC Connections" on page 30.

Editing Adapter Connections

If you want to redefine parameters that a connection uses when connecting to the OPC UA server, you can update a connection's parameters using Integration Server Administrator.

To edit a connection

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

2. Make sure that the connection is disabled before editing it. For instructions, see "Disabling Adapter Connections" on page 38.

3. On the Connections screen, click the icon for the connection you want to edit.

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

   For descriptions of the connection parameters, see "Configuring Adapter for OPC Connections" on page 30.

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

Copying Adapter Connections

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

To copy a connection

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

2. On the Connections page, click the icon for the connection you want to copy.

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

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

   For descriptions of the connection parameters, see "Configuring Adapter for OPC Connections" on page 30.

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

Deleting Adapter Connections

If you no longer want to use a particular Adapter for OPC connection, you can delete it. You delete adapter connections using Integration Server Administrator.

If you delete a Adapter for OPC connection, the adapter services or notifications that are defined to use the connection will no longer work. However, you can assign a different connection to an adapter service and re-use the service. To do this, use the setAdapterServiceNodeConnection built-in service. For more information, see "Changing the Connection Associated with an Adapter Service or Notification at Design-Time" on page 16.

To delete a connection

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

2. Make sure that the connection is disabled before deleting. To disable the connection, click **Yes** in the **Enabled** column and click **OK** to confirm. The **Enabled** column now shows **No** (Disabled) for the connection.

3. On the Connections screen, click for the connection you want to delete.

   Integration Server deletes the adapter connection.

Enabling Adapter Connections

Adapter for OPC 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 automatically be enabled when the package reloads. If the package contains connections that are disabled, they will remain disabled when the package reloads.

To enable a connection

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

2. On the Connections page, click **No** in the **Enabled** column for the connection you want to enable.

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

Disabling Adapter Connections

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

To disable a connection

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

2. On the Connections screen, click **Yes** in the **Enabled** column for the connection you want to disable.

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

5 Adapter Services

Overview of Adapter Services

This chapter describes how to configure and manage Adapter for OPC services. For detailed descriptions of the available Adapter for OPC services, see "Adapter Services" on page 14.

Preparing to Configure and Manage Adapter Services

Perform the following steps before configuring or managing adapter services.

To prepare to configure or manage Adapter for OPC services

1. Ensure that Integration Server and Integration Server Administrator is up and running.

2. Make sure you have Integration Server Administrator privileges so that you can access Adapter for OPC's administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide*.

3. Using Integration Server Administrator, make sure the WmOPCAdapter package is enabled.

4. Using Integration Server Administrator, configure an adapter connection to use with the adapter service.

   Note:
   Adapter for OPC provides a built-in service you can use at design time to change the connection associated with an adapter service. For more information, see "Changing the Connection Associated with an Adapter Service or Notification at Design-Time" on page 16.

5. Start Software AG Designer if it is not already running.

6. Using Software AG Designer, create a user-defined package to contain the service, if you have not already done so. When you configure adapter services, you should always define them in user-defined packages rather than in the WmOPCAdapter package. For more information about managing packages for the adapter, see "Overview of Package Management" on page 24.

Configuring Describe Service

Prerequisite:

Before you configure adapter services, ensure that you have completed the steps mentioned in "Preparing to Configure and Manage Adapter Services" on page 40.

A Describe service retrieves the node references from the OPC UA server. You must configure Adapter for OPC services using Software AG Designer. For more information about adapter services, see "Using Adapter Services" on page 16.

To configure a Describe service

1. In Software AG Designer, right click the package in which the service should be contained and select **File > New > Adapter Service.**

2. Select the service from the list of elements. Click **Next.**

3. Select the Adapter for OPC as the adapter type. Click **Next.**

4. Select the appropriate **Adapter Connection Name**. Click **Next.**

5. In the list of available templates, select the **Describe** template. Click **Finish**.

   In the adapter service editor, select the Adapter Settings tab to confirm adapter service properties such as **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template**, as necessary.

6. In **Node Attributes** tab, click **Browse** to navigate and find the node. The following fields are available in the **Node Selector** tab :

Field	Description
Search Node	Searches the node. **Note:** This is mainly for the nodes that are available but not visible due to **Tree View Level** control in **Adapter Service Preference.** For more information on **Adapter Service Preference**, see *webMethods Service Development Help*.

After selecting the node from **Node Selector**, the selected node appears in the **Extended Node Id**. The following fields are available in **Node Attributes** tab:

Field	Description/Action
Extended Node Id	Displays the extended node id of the selected node. The node id is represented in the following format: `nsu=<namespace uri>; Id type=Id value`

In Adapter Tree Viewer, right click on a node to view the following menu options:

Menu Option	Description
Set As Root	Use to set a node in the Tree-View as the root node. To view the hidden nodes in the tree view, set a node as a root node.
Copy Node Id	Use to copy the node's id from the tree view.
Copy Node Details	Use to copy the node's details from the tree view.

7. The selected node has the following fields:

Field	Description/Action
Attribute Name	Lists all the attributes supported by the selected node.
Attribute Value	Show the corresponding attribute value.

8. Select **Configuration tab** to specify the filter for output information:

Field	Description
Browse Direction	■ **Forward.** Output result lists only forward references. ■ **Inverse.** Output result lists only inverse references. ■ **Both.** Output result lists both **Forward** and **Inverse** references.
Reference Type	■ **Hierarchical.** Output result lists all references that have type hierarchical or subtype hierarchical . ■ **NonHierarchical.** Output result lists all references that have type NonHierarchical or subtype NonHierarchical. ■ **Both.** Output result lists both **Hierarchial** and **NonHierarchical.** references.
Include SubType	If selected, the output result includes references of **SubType** of selected **Reference Type**. If not selected, the output result lists only references of the **Reference Type** selected.
Node Class	To filter the references according to the target node's class type. If **Unspecified**, the filter is not be applicable.
Max Reference Returned	Specify the number of references you want to receive when you execute a service. Specifying zero returns all the results.

9. From the **File** menu, select **Save (or Save All)**.

Configuring Read Service

Prerequisite:

Before you configure adapter services, ensure that you have completed the steps mentioned in "Preparing to Configure and Manage Adapter Services" on page 40.

A Read service retrieves a node value from the OPC UA server. You can configure Adapter for OPC services using Software AG Designer. For more information about adapter services, see "Using Adapter Services" on page 16.

To configure a Read service

1. In Software AG Designer, right click the package in which the service is contained and select **File > New > Adapter Service.**

2. Select the service from the list, provide the **Element name**. Click **Next.**

3. Select Adapter for OPC as the adapter type. Click **Next.**

4. Select the appropriate **Adapter Connection Name**. Click **Next.**

5. Select the template for the unique service name. Click **Finish.**

   In the adapter service editor, select the Adapter Settings tab to confirm adapter service properties such as **Adapter Name**, **Adapter Connection Name**, and**Adapter Service Template**, as necessary.

6. In **Node Attributes** tab, click **Browse** to navigate and find the node. The following fields are available in the **Node Selector** tab :

Field	Description
Search Node	Searches the node. **Note:** This is mainly for the nodes that are available but not visible due to **Tree View Level** control in **Adapter Service Preference.** For more information on **Adapter Service Preference**, see *webMethods Service Development Help*.

After selecting the node from **Node Selector**, the selected node appears in the **Extended Node Id**. The following fields are available in **Node Attributes** tab:

Field	Description/Action
Extended Node Id	Displays the extended node id of the selected node. The node id is represented in the following format: `nsu=<namespace uri>; Id type=Id value`
Node Id	Displays the node id of the selected node.
Use Dynamic Node	Provides node id at run-time. This option displays all the attributes supported by OPC UA server.

Field	Description/Action
	Note: This option takes priority over the selected node.

In Adapter Tree Viewer, right click on a node to view the following menu options:

Menu Option	Description
Set As Root	Use to set a node in the Tree-View as the root node. To view the hidden nodes in the tree view, set a node as a root node.
Copy Node Id	Use to copy the node's id from the tree view.
Copy Node Details	Use to copy the node's details from the tree view.

7. To select attributes that retrieve value from the OPC UA server.

Field	Description
Field Name	Lists all the attributes supported by the selected node.
Use Field	To select the field you want to read.

The `value` field is applicable only if the attribute is of `value` type.

Field	Description/Action
Value Field Native Type	Displays the built-in type and complex data types of the value attribute that the OPC UA server supports.
Value Field Output Type	Displays all the supported Java data types for built-in type. Displays java.lang.Object, Document or DocumentList for complex data types.

8. To verify input or output information for this service, use the **Input/Output** tab as needed.

   If the **Value Field Output Type** is of type Document or DocumentList, then the output value displays the structure for corresponding the Document or DocumentList.

9. From the **File** menu, select **Save**.

Configuring Write Service

Prerequisite:

Before you configure adapter services, ensure that you have completed the steps mentioned in "Preparing to Configure and Manage Adapter Services" on page 40.

A write service writes into a node value from the OPC UA server. You can configure Adapter for OPC services using Software AG Designer. For more information about adapter services, see "Using Adapter Services" on page 16.

To configure Write service

1. In Software AG Designer, right click the package in which the service is contained and select **File > New > Adapter Service.**

2. Select the service from the list, provide the **Element name**. Click **Next.**

3. Select Adapter for OPC as the adapter type. Click **Next.**

4. Select the appropriate **Adapter Connection Name**. Click **Next.**

5. Select the template for the unique service name. Click **Finish.**

   In the adapter service editor, select the Adapter Settings tab to confirm adapter service properties such as **Adapter Name**, **Adapter Connection Name**, and**Adapter Service Template**, as necessary.

6. In **Node Attribute** tab, click **Browse** to navigate and find the node. The following fields are available in the **Node Selector** tab :

Field	Description
Search Node	Searches the node. **Note:** This is mainly for the nodes that are available but not visible due to **Tree View Level** control in **Adapter Service Preference.** For more information on **Adapter Service Preference**, see *webMethods Service Development Help*.

After selecting the node from **Node Selector**, the selected node appears in the **Extended Node Id**. The following fields are available in **Node Attribute** tab:

Field	Description/Action
Extended Node Id	Displays the extended node id of the selected node. The node id is represented in the following format: `nsu=<namespace uri>; Id type=Id value`
Node Id	Displays the node id of the selected node.

In Adapter Tree Viewer, right click on a node to view the following menu options:

Menu Option	Description
Set As Root	Use to set a node in the Tree-View as the root node. To view the hidden nodes in the tree view, set a node as a root node.
Copy Node Id	Use to copy the node's id from the tree view.
Copy Node Details	Use to copy the node's details from the tree view.

7. To select attributes that write value to the OPC UA server. The following fields are auto-populated:

Field	Description/Action
Field Name	Displays the value attribute name.
Native Type	Displays the built-in type and complex data types of the value attribute that the OPC UA server supports.
Input Field Type	Displays all the supported Java data types for built-in type. Displays java.lang.Object, Document or DocumentList for complex data types. For more information, see "webMethods Adapter for OPC Data Type Mapping" on page 72. **Note:** If the **Input Field Type** is an array, then you must provide the value at runtime. This field is non- editable.
Input Value	Specify the input value either at runtime or design-time. **Note:** If you input "?", then you must provide the value at runtime.

8. To verify input or output information for this service, use the **Input/Output** tab as needed.

   If the **Input Field Type** is of type Document or DocumentList, then the output value displays the structure for corresponding the Document or DocumentList.

9. From the **File** menu, select **Save**.

Configuring BatchRead Service

Prerequisites:

Ensure that you complete the steps mentioned in "Preparing to Configure and Manage Adapter Services".

BatchRead service retrieves node values from the OPC UA server. You can configure Adapter for OPC services using Software AG Designer.

To configure a BatchRead service

1. In Software AG Designer, right click the package in which the service is contained and select **File > New > Adapter Service**.

2. Select the service from the list, provide the **Element name**. Click **Next**.

3. Select Adapter for OPC as the adapter type. Click **Next**.

4. Select the appropriate **Adapter Connection Name**. Click **Next**.

5. Select the template for the unique service name. Click **Finish**.

   In the adapter service editor, select the **Adapter Settings** tab to confirm adapter service properties such as **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template**, as necessary

6. In the **Nodes** tab, click for node selection. You can select multiple nodes at a time from **Node Selector**. The following fields perform the operation:

Field	Description
Search Node	Searches the node.
Select All	Selects all check boxes for the variable nodes.
Clear All	Deselects all check boxes for the variable nodes.

After selecting the nodes from **Node Selector**, the selected nodes appear in the **Extended Node Id**. The nodes available in the **Nodes** tab appear as disabled in the **Node Selector**. The following fields perform the operation:

Field	Description/Action
Extended Node Id	Displays the extended node id of the selected node. The node id is represented in the following format: `nsu=<namespace uri>; Id type=Id value`
Node Id	Displays the node id of the selected node.
Node Name	Displays the name of the selected node id from the OPC UA server.

The following table gets auto-filled.

Menu Option	Description
Parameter	Displays the sequence number.
Node Id	Displays the selected node Id.
Native Type	Displays the native data types. Displays the built-in type of the value attribute that the OPC UA server supports.
Output Type	The data type of the output field. You can change this type if needed.
Output Field	The output field name. You can change this name if needed.

Configuring BatchWrite Service

Prerequisites:

Before you configure adapter services, ensure that you complete the steps mentioned in "Preparing to Configure and Manage Adapter Services".

BatchWrite service retrieves node values from the OPC UA server. You can configure Adapter for OPC services using Software AG Designer.

To configure a BatchWrite service

1. In Software AG Designer, right click the package in which the service is contained and select **File > New > Adapter Service**.

2. Select the service from the list, provide the **Element name**. Click **Next**.

3. Select Adapter for OPC as the adapter type. Click **Next**.

4. Select the appropriate **Adapter Connection Name**. Click **Next**.

5. Select the template for the unique service name. Click **Finish**.

   In the adapter service editor, select the **Adapter Settings** tab to confirm adapter service properties such as **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template**, as necessary

6. In the **Nodes** tab, click for node selection. You can select multiple nodes at a time from **Node Selector**. The following fields perform the operation:

Field	Description
Search Node	Searches the node.
Select All	Selects all check boxes for the variable nodes.
Clear All	Deselects all check boxes for the variable nodes.

After selecting the nodes from **Node Selector**, the selected nodes appear in the **Extended Node Id** tab. The nodes available in the **Nodes** tab appear as disabled in the **Node Selector**. The following fields perform the operation:

Field	Description/Action
Extended Node Id	Displays the extended node id of the selected node. The node id is represented in the following format: `nsu=<namespace uri>; Id type=Id value`
Node Id	Displays the node id of the selected node.
Node Name	Displays name of the selected node id from the OPC UA server.

The following table gets auto-filled.

Menu Option	Description
Parameter	Displays the sequence number.
Node Id	Displays the selected node id.
Native Type	Displays the native data types. Displays the built-in type of the value attribute that the OPC UA server supports.
Output Type	The data type of the output field. You can change this type if needed.
Output Field	The output field name. You can change this name if needed.

Configuring Call Method Service

Prerequisite:

Before you configure adapter services, ensure that you have completed the steps mentioned in "Preparing to Configure and Manage Adapter Services" on page 40.

A call method service executes a node value from the OPC UA server. You can configure Adapter for OPC services using Software AG Designer. For more information about adapter services, see "Using Adapter Services" on page 16.

To configure Call Method service

1. In Software AG Designer, right click the package in which the service is contained and select **File > New > Adapter Service.**

2. Select the service from the list, provide the **Element name**. Click **Next.**

3. Select Adapter for OPC as the adapter type. Click **Next.**

4. Select the appropriate **Adapter Connection Name**. Click **Next.**

5. Select the template for the unique service name. Click **Finish.**

   In the adapter service editor, select the Adapter Settings tab to confirm adapter service properties such as **Adapter Name**, **Adapter Connection Name**, and**Adapter Service Template**, as necessary.

6. Select **Method** tab, to select the attributes that retrieve value from the OPC UA server.

7. In **Extended Node Id**, click **Browse** to navigate and find the node. The node should have a child node of type `method`. The operation is performed by using the following fields:

Field	Description/Action
Search Node	This field shows the node which is searched and selected from the OPC UA server. **Note:** This is mainly for the nodes that are available but not visible due to **Tree View Level** control in **Adapter Service Preference.** For more information on **Adapter Service Preference**, see *webMethods Service Development Help*.

In Adapter Tree Viewer, right click on a node to view the following menu options:

Menu Option	Description
Set As Root	Use to set a node in the Tree-View as the root node. To view the hidden nodes in the tree view, set a node as a root node.
Copy Node Id	Use to copy the node's id from the tree view.

Menu Option	Description
Copy Node Details	Use to copy the node's details from the tree view.

8. The following fields are auto-populated:

Field	Description/Action
Node Id	Displays the selected node Id.
Method Name	Lists all the methods supported by the selected node.
Argument Name	Lists all argument names required by the selected method.
Argument Description	Displays the corresponding argument description.
Argument Native Types	Displays the built-in type and complex data types of the argument that the OPC UA server supports.
Argument Data Type	Displays all the supported Java data types for built-in type. Displays java.lang.Object, Document or DocumentList for complex data types. For more information, see "webMethods Adapter for OPC Data Type Mapping" on page 72. **Note:** If the **Argument Field Type** is an array, then you must provide the value at runtime. This field is non editable.
Argument Input	Specify the argument input value either at runtime or design-time. **Note:** If you input "?", then you must provide the value at runtime.

Field	Description/Action
Return Value Name	Displays the return value name.
Return Value Native Type	Displays the built-in type and complex data types of the return value that the OPC UA server supports.
Return Value Output Type	Displays all the supported Java data types for built-in type. Displays java.lang.Object, Document or DocumentList for complex data types.

9. To verify input or output information for this service, use the **Input/Output** tab as needed.

If the **Return Value Output Type** is of type Document or DocumentList, then the output value displays the structure for corresponding the Document or DocumentList.

10. From the **File** menu, select **Save**.

6 Adapter Listeners and Listener Notification

Overview of Adapter Listeners and Listener Notifications

This chapter describes how to configure and manage Adapter for OPC Listeners and Listener Notifications. For detailed descriptions of the available Adapter for OPC notifications, see "Adapter Listeners and Listener Notification" on page 17 .

Preparing to Configure New Listeners

To prepare to configure a new listener

1. Ensure that Integration Server and Integration Server Administrator are started.

   Note:
   Make sure that you have webMethods administrator privileges so that you can access the adapter's administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide*.

2. Using Integration Server Administrator, make sure that the WmOPCAdapter package is enabled. To verify the status of the WmOPCAdapter package, see " Package Management" on page 12.

3. Using Software AG Designer, create a user-defined package to contain the listener. For more information about managing packages, see "Overview of Package Management" on page 24.

Configuring Listener Notification

You can configure listener notification using Software AG Designer. For more information, see "Adapter Listeners and Listener Notification" on page 17.

Configuring an Adapter Listener

1. In Integration Server Administrator, select **Adapters** > **Adapter for OPC**.

2. In the **Adapter for OPC** menu, select **Listeners**.

3. On the Listeners screen, select **Configure New Listener**.

4. On the Listener Types screen, select **Subscription Listener**.

5. On the Configure Listener Type screen, configure the following fields:

Parameter	Description/Action
Package	The package in which to create the listener.

Parameter	Description/Action
	Create the listener in a user-defined package rather than in the adapter's package. For other considerations when creating packages for the adapter, see "Overview of Package Management" on page 24. You must create the package using Software AG Designer before you can specify the package by using this parameter. For information about creating packages, see *webMethods Service Development Help*.
Folder Name	The folder in which to create the listener.
Listener Name	The name of the new listener.
Connection Name	The name of the connection.
Retry Limit	The number of times that the adapter tries to reconnect, if the adapter fails to connect to or loses connection with the OPC UA server.
Retry Backoff Timeout	The number of seconds that elapse between each of the retries specified in the retry limit.

Important:
The listener name is prefixed by the folder name and is separated by a colon. For example, if the folder name is "Folder1" and the listener name is "Listener1", the listener name in the Listeners screen will be "Folder1:Listener1".

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

Parameter	Description/Action
Publish Interval(ms)	Defines how often the OPC UA server checks for notification for a Subscription which is then sent back to Adapter for OPC Default: 1000 (ms).
Keep Alive Count	If there is no data to publish after the next **PublishingInterval**, this parameter defines how many intervals the server skips before an empty notification is sent to Adapter for OPC. It indicates that the subscription is still alive in the server and there is no data to send.
Lifetime Count	It defines the number of **PublishingIntervals** to wait for a new `PublishRequest`, before realizing that the Adapter for OPCListener is no longer active. The Subscription is then removed from the server.
Max Notification Per Publish	The limit of notification messages per publish sent from the OPC UA server to the adapter.
Priority	It defines the priority of a subscription relative to the other subscription created by the Adapter for OPC.

Parameter	Description/Action
Monitoring Mode	The monitoring mode defines the behavior of the notification on the OPC UA server. The types of monitoring modes are: ■ **Reporting.** The datasource is sampled and the notification is sent to the Adapter for OPC. ■ **Sampling.** The datasource is only sampled and no notification is sent to the Adapter for OPC. ■ **Disabled.** The datasource is disabled.

7. Click **Save.**

Enabling Listeners

Before you enable a listener, you need to configure one or more notifications to associate with the listener. If no notifications are configured when you enable the listener, Integration Server Administrator displays a warning message.

After you configure your notifications, you must enable the listener so that the associated notifications 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. If the status is **Succeeded**, the listener is ready to be enabled. If the status is **Failed**, an error occurred during startup. If an error occurs during startup, the state will not change to **Enabled** when refreshing the page. Errors at this stage typically indicate a problem with either the listener configuration or the network. Review the listener settings and check the network.

For more information, see "Configuring an Adapter Listener" on page 54 and " Configuring Listener Notification" on page 54.

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

To enable a listener

1. In Integration Server Administrator, select **Adapters** > **webMethods Adapter for OPC**.

2. In the **Adapter for OPC** menu, select **Listeners**. The Listeners screen appears.

3. 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.

Data Change Notification

Notifies the data changes in variable values. For more information, see "Adapter Listeners and Listener Notification" on page 17.

Configure Data Change Notification

To configure data change notification

1. Start Software AG Designer.
2. Right-click the package that contains the listener notification. Select **New > Adapter Notification**.
3. Select the parent namespace, type a name for the adapter listener notification. Click **Next**.
4. Select **webMethods Adapter for OPC** as the adapter type. Click **Next**.
5. Select **Data Change Notification** as the template. Click **Next**.
6. Select the appropriate **Notification Listener Name**. Click **Next**.
7. Click **Finish**.

   Software AG Designer creates a data change listener notification and a publishable document type.

Using Data Change Notification

To use data change notification

1. In the Adapter Notification Editor, Click the **Add Items** tab

   In the **Add Items** tab, click for node selection. You can select multiple nodes at a time from **Node Selector**. The following fields perform the operation:

Field	Description
Search Node	Searches the node.
Select All	Selects all check boxes for the variable nodes.
Clear All	Deselects all check boxes for the variable nodes.

After selecting the nodes from **Node Selector**, the selected nodes appear in the **Add Items** tab. The nodes available in the **Add Items** tab appear as disabled ☑ in the **Node Selector**.

Parameter	Description/Action
Extended Node Id	Displays the extended node id of the selected event source node id. The extended node id is represented in the following format: `nsu=<namespace uri>; Id type=Id value`
Node Id	The unique id of the selected event source node id to be monitored.
Sampling Interval (ms)	The fastest rate at which the OPC UA server should sample its underlying source for data changes. Default : 0
Message Queue Size	The maximum number of messages stored in the queue for delivery. Default : 0
Discard Oldest Message	■ **True.** The oldest message from the queue gets discarded. ■ **False.** No messages are discarded. If the queue size is full, any new messages are ignored.
Monitoring Mode	■ **Default.** The monitoring mode specified during configuration of the listener. **Note:** All the other options override the listener's monitoring mode at each item level. ■ **Reporting.** The datasource is sampled and the notification is sent to the Adapter for OPC. ■ **Sampling.** The datasource is only sampled and no notification is sent to the Adapter for OPC. ■ **Disabled.** The datasource is disabled.
Filter	Defines the conditions on which you want to receive data change notifications. When the filter settings are set. a drop-down appears with the specified filter names.

2. The filter settings are as follows:

Parameter	Description/Action
Filter Name	The name of the filter.

Parameter	Description/Action
Trigger Condition	The types of trigger conditions that enable the notification: ■ **Status / Value.** Notifies if either the status or the value changes. ■ **Status.** Notifies if only the status changes. **Note:** Since there is no value in this type of trigger condition, **Deadband Type** and **Deadband Value** are not applicable. ■ **Status / Value / Timestamp.** Notifies if either the status, the value or the timestamp changes.
Deadband Type	It defines the type of **Deadband Value** applied. The following are the types of deadband available: ■ **Absolute.** If the absolute change in the last data value and new data value is greater than the value specified, it triggers the notification. ■ **Percent.** If the changed data value is greater than the percentage of the configured value, it triggers the notification. ■ **None.** No deadband evaluation is done.
Deadband Value	The given value based on the **Deadband Type**.

Enabling Data Change Notification

Note:
Every notification creates its own publishable document. The name of the publishable document is the name of the notification prefixed with **PublishDocument**. Example: If the name of the notification is *datachange* then the name of publishable document is *datachangePublishDocument*.

To enable data change notification

1. Click on ***Notification_name*PublishDocument**.

Parameter	Description/Action
Subscription	■ **id.** Id of the Subscription object created in the OPC UA Server. ■ **Name.** The given name of the Subscription or Listener. This attribute helps to identify from which listener you get the notification.
data	■ **nodeName.** The name of the node from which you get the notification.

Parameter	Description/Action
	■ **value.** The actual value of the node. ■ **serverTimestamp.** The server timestamp for the value. ■ **sourceTimestamp.** The source timestamp for the value. ■ **serverPicoseconds.** The picoseconds that are added to the serverTimestamp. ■ **sourcePicoseconds.** The picoseconds that are added to the sourceTimestamp.
statusCode	It defines the sever's ability to access the value. The following parameters are described: ■ **name.** It defines the name of the status code. ■ **value.** It defines the value of the status code. ■ **description.** It gives the description of the status code.

Event Notification

Notifies about the occurrence of an event. It may or may not be associated with a condition. For more information, see "Adapter Listeners and Listener Notification" on page 17

Configure Event Notification

To configure event notification

1. Start Software AG Designer.
2. Right-click the package in which the listener notification should be contained. Select **New > Adapter Notification**.
3. Select the parent namespace, type a name for the adapter listener notification. Click **Next**.
4. Select **webMethods Adapter for OPC** as the adapter type. Click **Next**.
5. Select **Event Notification** as the template. Click **Next**.
6. Select the appropriate **Notification Listener Name**. Click **Next**.
7. Click **Finish**.

   Software AG Designer creates an event listener notification and publishable document type.

Using Event Notification

To use event notification

1. In Adapter Notification Editor **Select Event Source** tab, click **Browse** to navigate and select the event source node. The following fields are available in the **Node Selector** tab :

Field	Description
Search Node	Searches the node. **Note:** This is mainly for the nodes that are available but not visible due to **Tree View Level** control in **Adapter Service Preference.** For more information on **Adapter Service Preference**, see *webMethods Service Development Help*.

After selecting the node from **Node Selector**, the selected node appears in the **Extended Node Id**.

Field	Description/Action
Extended Node Id	Displays the extended node id of the selected node. The node id is represented in the following format: `nsu=<namespace uri>; Id type=Id value`

In Adapter Tree Viewer, right click on a node to view the following menu options:

Menu Option	Description
Set As Root	Use to set a node in the Tree-View as the root node. To view the hidden nodes in the tree view, set a node as a root node.
Copy Node Id	Use to copy the node's id from the tree view.
Copy Node Details	Use to copy the node's details from the tree view.

2. In Adapter Notification Editor **Select Event Source** tab, the **Event Source** displays the selected event source node id.

Note:
Only one event source is supported per one event notification.

3. Set the following parameters as required:

Parameter	Description/Action
Monitoring Mode	■ **Default.** The monitoring mode specified during configuration of the listener. **Note:** All the other options override the listener's monitoring mode at each item level. ■ **Reporting.** The datasource is sampled and the notification is sent to the Adapter for OPC. ■ **Sampling.** The datasource is only sampled and no notification is sent to the Adapter for OPC. ■ **Disabled.** The datasource is disabled.
Queue Size	The maximum number of messages stored in the queue for delivery. Default : 0
Discard Oldest Message	■ **True.** The oldest message from the queue gets discarded. ■ **False.** No messages are discarded. If the queue size is full, any new messages are ignored.

4. In the **Configure Filter** tab, **Filter Query**: write the filter condition for the notification to be generated. For more information on the rules for query writing, see "QueryParserLanguage" on page 62.

5. If the condition is met, select the fields which you want to include in the notification.

QueryParserLanguage

QueryParserLanguage overcomes the complexity of providing filter condition for Event Notifications. The objective is to help in writing and running filter conditions for Event Notifications.

It provides a more convenient way to write and manipulate filter condition. By using QueryParser, user can specify filter condition in a query format for Event Notification. It supports most of FilterOperators specified as part of OPC UA Part 4 - Services 1.03 Specification.

Operator	Symbol	Usage Example
equals	==	Operand0 == Operand1
IsNull	IsNull	IsNull(operand)
GreaterThan	>	Operand0 > Operand1
LessThan	<	Operand0 < Operand1

Operator	Symbol	Usage Example
GreaterThanOrEqual	>=	Operand0 >= Operand1
LessThanOrEqual	<=	Operand0 <= Operand1
Like	Like	Like(Operand0 , Operand1)
Not	Not	Not(Operand)
Between	Between	Between(operand0 , operand1, operand2)
InList	InList	InList(Operand [2..n])
And	And	Operand0 && Operand1
Or	Or	Operand0 || Operand1
Cast	Cast	Cast(operand0 , operand1)
BitwiseAnd	BitwiseAnd	BitwiseAnd(operand0 , operand1)
BitwiseOr	BitwiseOr	BitwiseAnd(operand0 , operand1)
InView	InView	InView(Operand)
OfType	OfType	OfType(Operand)
RelatedTo	RelatedTo	RelatedTo(Operand0 , Operand1 ,Operand2 ,Operand3, Operand4 ,Operand5)

QueryParserLanguage supports two ways to provide operand values in the filter query.

- **Literal Operand** : User can directly provide known values as operand that is treated as Literal Operand. Example : 800 , 75 , -3 , 'MyLevel'
- **SimpleAttributeOperand** : User can specify names of condition parameter as operand in the query.

For Example,

```
Severity==800
```

```
HighLimit >= 300
```

In this case, the **value** attribute is considered to be evaluated for **Severity**(Value attribute is compared with 800) and **HighLimit**(Value attribute is compared with 300).

To verify any specific attribute and indexRanges, certain format is used:

```
OperandName{attributeName, {IndexRange}}
```

Few more query examples:

```
Severity == 800
 Between(Severity , 100, 200)
```

```
SuppressedState.Id == 8
(SuppressedState.Id == 8) && (Severity == 700)
(SuppressedState.Id == 8) && (Severity == 700) && (HighLimit  == 200)
Severity > 700
Severity >= 700
Severity < 700
Severity <= 700
Not((SuppressedState.Id == 8) && (Severity == 700))
Not(Severity <= 700)
OfType(i=2048)
OfType(ns=5;s=OPCSampleNode)
(SuppressedState.Id == 200) && (Between(Severity , 100, 200))
Not(Severity == 800) && (SourceName=='MyLevel') && (HighLimit >= 300)
Not(EnabledState.Id == false) && ((Severity == 500) || (Severity == 700))
HighLimit >= LowLimit
ActiveState.Id && EnabledState.Id
SourceName == 'Severity{NodeId} == 500'
ActiveState.Id || EnabledState.Id
InList(Severity , 500 , 600 , 700)
SourceName{DisplayName} == 'SourceName'
((Severity == 500) || (Severity == 700)) && (SourceName{DisplayName} == 'SourceName')
```

Enabling Event Notification

To enable event notification

1. Click on ***Notification_name*PublishDocument**.

Parameter	Description/Action
Subscription	■ **id.** Id of the Subscription object created in the OPC UA Server. ■ **Name.** The given name of the Subscription or Listener. This attribute helps to identify from which listener you get the notification.
eventData	It is dynamic and the structure of it depends on the selection of fields in the second step in "Using Event Notification" on page 61.

7 Predefined Health Indicator

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

The Administrator APIs are available for Adapter for OPC. 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

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*.

A Data Type Mapping

webMethods Adapter for OPC Data Type Mapping

The supported data types are:

OPC Data Type	Data Type
Boolean	java.lang.Boolean/java.lang.String
SByte	java.lang.Byte/java.lang.String
Byte	java.lang.Integer/java.lang.String
Int16	java.lang.Short/java.lang.String
UInt16	java.lang.Integer/java.lang.String
Int32	java.lang.Integer/java.lang.String
UInt32	java.lang.Long/java.lang.String
Int64	java.lang.Long/java.lang.String
UInt64	java.math.BigInteger/java.lang.String
Float	java.lang.Float/java.lang.String
Double	java.lang.Double/java.lang.String
String	java.lang.String
DateTime	java.lang.String
Guid	java.lang.String
ByteString	java.lang.Byte[]
XmlElement	java.lang.String
NodeId	java.lang.String
ExpandedNodeId	java.lang.String
StatusCode	java.lang.Integer/java.lang.String
QualifiedName	java.lang.String
LocalizedText	java.lang.String
ExtensionObject	java.lang.Object
DataValue	java.lang.Object
Variant	java.lang.Object or depends on variant value type.

OPC Data Type	Data Type
DiagnosticInfo	java.lang.Object

Complex Data Types for Adapter for OPC

Adapter for OPC supports following list of Complex data types:

Complex Data Type
Argument
UserIdentityToken
AddNodesItem
AddReferencesItem
DeleteNodesItem
DeleteReferencesItem
ApplicationDescription
BuildInfo
RedundantServerDataType
SamplingIntervalDiagnosticsDataType
ServerDiagnosticsSummaryDataType
ServerStatusDataType
SessionDiagnosticsDataType
SessionSecurityDiagnosticsDataType
ServiceCounterDataType
StatusResult
SubscriptionDiagnosticsDataType
ModelChangeStructureDataType
SemanticChangeStructureDataType
SignedSoftwareCertificate
TimeZoneDataType
EnumValueType

Complex Data Type
OptionSet
Union

Custom Data Types for Adapter for OPC

In Adapter for OPC, when any new custom data types are defined in the OPC server side, then Adapter for OPC is unable to understand these data types as custom data type class definition is not available in Adapter for OPC class path. To avoid this kind of circumstance, additional changes are made so that Adapter for OPC can easily access the custom data types.

To access the custom data type value of the nodes, the custom data type class definition needs to be available in Adapter for OPC class path. You have to generate classes from information model and place into `InformationModeler.jar` file. The `InformationModeler.jar` file needs to be available in `WmOPCAdapter/Code/jars` folder because the custom data types cannot work without the `InformationModeler jar` file.

To understand the data type, `InformationModeler.jar` file must have the following information.

- InformationModeler XML file
- Packages consisting of classes generated from information model file

The server has to map the information into the InformationModeler XML file. For example:

```
<informationModel>
 <modeler namespaceUri="http://ua.prosysopc.com/STypes">
  <modelerInfo uaTypeInstance="example.packagename.client.BuildInfoVarTypeTestImpl"
   serializer="example.packagename.BuildInfoTypeTestSerializer"/>
  <dataTypeDictionary>
   <nodeInfo id="i=449" name="BuildInfoTypeTest"
    class="example.packagename.BuildInfoTypeTest" />
  </dataTypeDictionary>
 </modeler>
</informationModel>
```

The following table explains how the adapter understands based on the information present in the InformationModeler XML file.

Attributes	Description
namespaceUri	Namespace URI of custom data type node
uaTypeInstance	Class name of the custom data type that extends `com.prosysopc.ua.types.opcua.client.BaseDataVariableTypeImpl` and implements `com.prosysopc.ua.types.opcua.BaseDataVariableType`.
serializer	Name of the class for encoding and decoding of data
node Id	Type of Node

Attributes	Description
name	Name of the custom data type node
class	Name of the class that extends `org.opcfoundation.ua.utils.AbstractStructure`

B Built-In Services

Overview

This appendix provides information on the built-in services provided by webMethods Adapter for OPC. These services are located in the WmOPCAdapter package.

pub.opcAdapter:upgrade

The pub.opcAdapter:upgrade service allows you to upgrade Adapter for OPC service or notification nodes from lower version to higher version. A field Extended Node Id is introduced. The format for **Extended Node Id** is *nsu=<namespace URI>;<id type>=<node value>*. The new services use the **Extended Node Id** as the key field associated with the tree view control whereas the old services continue to use **Selected Node** field.

Note:

- It is recommended to use the upgrade service to update Adapter for OPC service or notification nodes from a lower version of the Adapter for OPC template to a higher version. Using the upgrade service to update Adapter for OPC service or notification nodes from a higher version of Adapter for OPC template to a lower version is not supported
- After completing the upgrade services, refresh the package in the Designer

The upgrade service may take a long time to complete if an input package contains large number of connections. This is because packages are reloaded at the end of the upgrade service, and a package with many connections takes more time to reload. By default, the upgrade service runs under simulation mode, that is, no actual changes are made to the corresponding Adapter for OPC service or notification nodes, and only a check is performed if the Adapter for OPC service or notification nodes can be updated successfully.

Input Parameters

Parameter Name	Parameter Name	Description
input		**Document List**. Required. Represents an array of input records to the upgrade service. At least one input record is required. Each input record has the following parameters:
	packageName	**String**. Required. Name of the package containing the Adapter for OPC service or notification nodes to be updated.
	connectionName	**String**. Required. Name of the connection alias. All Adapter for OPC service or notification nodes matched by the namespace filter and having this connection alias will be picked for an update. The upgrade service will use the connection to communicate with OPC Server to fetch namespace table, required to generate extended node id.
	nsFilter	**Document**. Optional. A namespace filter in the form of text expression(s), which can be applied to selectively filter Adapter

Parameter Name	Parameter Name	Description
		for OPC service or notification nodes within the package specified by the *packageName* parameter. The namespace filter has the following parameters: ■ *patternStyle*. **String**. Optional. Allowed values are glob or regex. The style of include or exclude text expression(s). glob allows for UNIX glob like expressions that are used by UNIX shell commands, whereas regex allows regular expressions to be used (as used by Java and Perl languages). The default style is glob. ■ *include*. **String List**. Optional. An array of text to include Adapter for OPC service or notification nodes which are to be updated. The default is to include all nodes. ■ *exclude*. **String List**. Optional. An array of text expressions to exclude Adapter for OPC service or notification nodes from being updated. The default is to exclude none.
	detailedStatusFilter	**Document**. Optional. Filter the output field *detailedStatus* based on the success or failure of Adapter for OPC service or notification nodes being updated . This includes any combination of success, failure or warning. Default is set to true for all the following options: ■ *success*. **String**. Optional. Display the *detailedStatus* of Adapter for OPC service or notification node that successfully updated. Default is **true**. ■ *failed*. **String**. Optional. Display the *detailedStatus* of Adapter for OPC service or notification node that failed to update. Default is **true**. ■ *warning*. **String**. Optional. Display the *detailedStatus* of Adapter for OPC service or notification node that updated successfully but with warnings. Default is **true**.
simulate		**String**. Optional. Allowed values are true or false. Instructs the upgrade service to run in simulation mode during which no changes are made to Adapter for OPC service or notification nodes, and only a check is performed if a service or notification node can be updated successfully or not. Default value is true. Set simulate to false to persist the changes made to an Adapter for OPC service or notification node.
backup		**String**. Optional. Allowed values are true or false. Instructs the upgrade service to continue the backup of each package (specified via input records) before starting the update process. The backup

Parameter Name	Parameter Name	Description
		is created under folder *Integration Server_directory* \instances*instance_name*\replicate\outbound. The backup zip file name format is v*name_yyyy-MM-dd_HHmmss_z*. For example, package *ExtSample1* will be backed up as *ExtSample1_2020-07-08_155656_IST.zip*. Default value of this parameter is false. Please note that the package backup (applicable if this parameter is true) is taken irrespective of whether the update service is running in simulate mode or persistence mode.

Output Parameters

Parameter Name	Parameter Name	Description
output		**Document List**. Required. Represents an array of output records of the upgrade service corresponding to each input record. Each record has the following parameters:
	packageName	**String**. Required. Name of the package containing Adapter for OPC service or notification nodes provided as an input to the upgrade service.
	status	**String**. Required. The overall status of the upgrade service. Allowed values are success, error or warning.
	message	**String**. Required. A summary message in case the Adapter for OPC service or notification nodes are updated successfully, else an error message displaying the reasons for failure to update.
	summary	**Document**. Optional. Status summary of the Adapter for OPC service or notification nodes available in the given package after upgrade. Each record has the following parameters: ■ *total*. **String Array**. Optional. An array of fully qualified NSNames representing the individual Adapter for OPC service or notification node available in the given package. ■ *selected*. **String Array**. Optional. An array of fully qualified NSNames representing the individual Adapter for OPC service or notification nodes filtered based on the input fields *nsFilter* and *connectionName* selected for the upgrade. ■ *success*. **String Array**. Optional. An array of fully qualified NSNames representing the Adapter for OPC service or notification nodes successfully updated by the upgrade service.

Parameter Name	Parameter Name	Description
		■ *failed*. **String Array**. Optional. An array of fully qualified NSNames representing the Adapter for OPC service or notification nodes failed to update by the upgrade service. ■ *warning*. **String Array**. Optional. An array of fully qualified NSNames representing the Adapter for OPC service or notification nodes successfully updated by the upgrade service with a warning.
	detailedStatus	**String Array**. Optional. An array of the Adapter for OPC service or notification node records with detailed information of success, error or warning for each individual Adapter for OPC service or notification node, which was acted upon by the upgrade service. Each record has the following parameters: ■ *name*. **String**. Required. The fully qualified NSName of the Adapter for OPC service or notification node. ■ *status*. **String**. Required. Status of the updated Adapter for OPC service or notification node. Allowed values are success, error or warning. ■ *message*. **String**. Required. A message if the Adapter for OPC service or notification node was updated successfully, else an error message explaining why the node could not be updated. ■ *data*. **Document List**. Optional. List of records where each record represents the updated field information of the Adapter for OPC service or notification node.