

# Wonderware Operations Integration - Supervisory Gateway (G-1.2 Series) User Guide



© 2016 Schneider Electric Software, LLC. All rights reserved.

No part of this documentation shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Schneider Electric Software, LLC. No liability is assumed with respect to the use of the information contained herein.

Although precaution has been taken in the preparation of this documentation, Schneider Electric Software, LLC assumes no responsibility for errors or omissions. The information in this documentation is subject to change without notice and does not represent a commitment on the part of Schneider Electric Software, LLC. The software described in this documentation is furnished under a license agreement. This software may be used or copied only in accordance with the terms of such license agreement.

ArchestrA, Avantis, DYNsIM, eDNA, EYESIM, Foxboro, Foxboro Evo, I/A Series, InBatch, InduSoft, InStep, IntelaTrac, InTouch, PIPEPHASE, PRiSM, PRO/II, PROVISION, ROMeo, Schneider Electric, SIM4ME, SimCentral, SimSci, Skelta, SmartGlance, Spiral Software, VISUAL FLARE, WindowMaker, WindowViewer, and Wonderware are trademarks of Schneider Electric SE, its subsidiaries, and affiliated companies. An extensive listing of Schneider Electric Software, LLC trademarks can be found at: <http://software.schneider-electric.com/legal/trademarks/>. All other brands may be trademarks of their respective owners.

Schneider Electric Software, LLC  
26561 Rancho Parkway South  
Lake Forest, CA 92630 U.S.A.  
(949) 727-3200

<http://software.schneider-electric.com/>

Publication date: 7/1/2016

# Contact Us

## Contact Schneider Electric Software Technical Support

Contact our Software Technical Support Group if you have a question or issue with Avantis, SimSci, Skelta, or Wonderware software products.

- Priority email for Customer First Members: [custfirstsupport@wonderware.com](mailto:custfirstsupport@wonderware.com)
- Customers without a support agreement: [wwsupport@wonderware.com](mailto:wwsupport@wonderware.com)

Americas: Lake Forest, California, U.S.

- Tel: + 1 800 966 3371 / + 1 949 639 8500

Asia Pacific: Shanghai, China

- Tel: + 800 4500 3457 (All Asia) / + 400 820 1521 (China) / + 86 400 8201521
- Email: [gcssupport.apac@schneider-electric.com](mailto:gcssupport.apac@schneider-electric.com)

Europe and Middle East, Africa (EMEA): Bratislava, Slovakia

- Tel: + 421 2 32 200 342 / See our website at <http://software.schneider-electric.com/> for a list of toll-free numbers.

## Contact Schneider Electric Software Learning Services

Contact Software Learning Services for assistance regarding classes, schedule, offerings, frequently asked questions, tuition, policies, and more.

- Phone: + 1 866 998 7246 in the U.S. and Canada
- + 1 949 639 8508 direct
- Fax: + 1 949 639 1847
- Email: [software.training@schneider-electric.com](mailto:software.training@schneider-electric.com)



# Contents

Chapter 1 Introduction to the OI Gateway .....	9
About the OI Gateway .....	9
Supported Data Sources and Clients .....	10
Supported Client Protocols .....	10
OPC .....	10
SuiteLink .....	11
DDE/FastDDE .....	11
DDE .....	11
FastDDE .....	11
ArchestrA Message Exchange .....	11
OPC UA .....	11
MQTT .....	12
Chapter 2 Gateway Configuration .....	13
Configuring the Gateway .....	13
Before Configuring the Gateway .....	14
The Gateway Data Source Hierarchy .....	15
Configuring Data Source Redundancy .....	16
Chapter 3 Connecting to Data Sources .....	19
Connecting to an ArchestrA Data Source .....	19
Configuring an ArchestrA Data Source Object .....	19
Configuring an ArchestrA Group Object .....	20
Configuring ArchestrA Device Items .....	21
ArchestrA Item Names .....	21
Example #1 .....	22
Example #2 .....	22
Using Item Prefixes .....	23
ArchestrA Data Conversion .....	23
ArchestrA-DDE/SuiteLink Mappings .....	23
ArchestrA-OPC Mappings .....	25
Connecting to an OPC Data Source .....	28
Configuring an OPC Data Source Object .....	28
Configuring an OPC Group Object .....	30
Configuring OPC Device Items .....	31
OPC Item Names .....	31
Using Item Suffixes .....	31
OPC Data Conversion .....	33
DDE/SuiteLink-OPC Mappings .....	33
Connecting to a DDE/SuiteLink Data Source .....	35
Configuring a DDE/SuiteLink Data Source Object .....	36
Configuring a DDE/SuiteLink Topic Object .....	36

Configuring DDE/SuiteLink Device Items .....	37
DDE/SuiteLink Item Names .....	37
Example #1 .....	37
Example #2 .....	38
DDE/SuiteLink Data Conversion .....	39
OPC–DDE/SuiteLink Mappings .....	39
Connecting to an InTouch Data Source .....	41
Configuring an InTouch Data Source Object .....	41
Configuring an InTouch Group Object .....	42
Configuring InTouch Device Items .....	43
Adjusting for Time Zones .....	43
Handling Time Zones with the Time Property .....	43
Preserving Time Stamps from the Publishing Source .....	43
InTouch Item Names .....	44
Example #1 .....	44
Example #2 .....	45
InTouch Data Conversion .....	45
Connecting to an MQTT Data Source .....	45
Configuring an MQTT Data Source Connection .....	46
Verifying a Self-Signed Certificate from an Untrusted MQTT Data Source .....	47
Configuring an MQTT Group Connection .....	47
Configuring MQTT Device Items .....	48
MQTT Topic Names .....	48
Configuring an MQTT Publisher .....	49
Connecting to an OPC UA Data Source .....	53
Configuring an OPC UA Data Source Object .....	54
Configuring OPC UA Security .....	55
Configuring the OPC UA Namespace .....	56
Configuring an OPC UA Group Object .....	58
Configuring OPC UA Device Items .....	59
OPC UA Item Names and Syntax .....	59
OPC UA Data Conversion .....	61
<b>Chapter 4 Device Groups and Device Items .....</b>	<b>63</b>
Configuring Device Item Definitions .....	63
Device Item Definitions .....	63
Exporting and Importing the Gateway Item Data .....	64
<b>Chapter 5 Managing the Gateway .....</b>	<b>67</b>
Gateway Post-Configuration Steps .....	67
Configuring the Gateway as Service .....	67
Configuring the Gateway as Not a Service .....	68
Archiving Configuration Sets .....	68
Activating/Deactivating the Gateway .....	68
In-Proc/Out-of-Proc .....	69
Hot Configuration .....	69
<b>Chapter 6 Accessing the Data in the Gateway .....</b>	<b>71</b>
About Accessing Data in the Gateway .....	71

Accessing Data Using OPC .....	71
Accessing Data Using DDE/SuiteLink .....	72
Accessing Data Using MQTT .....	72
Accessing Data Using OPC UA .....	72
<b>Chapter 7 The Gateway Features .....</b>	<b>75</b>
Data Source Redundancy .....	75
Runtime Behavior .....	75
Instantiating the Gateway .....	76
Licensing for Gateway Features .....	77
<b>Chapter 8 System Items.....</b>	<b>79</b>
Standard System Items.....	79
Global System Item .....	79
Device-Specific System Items .....	80
Device Group-Specific System Items .....	81
The Gateway-Specific System Items.....	84
The Gateway Redundant Device-Specific System Items .....	85
OPC UA Diagnostic Items.....	86
Diagnostic Item Syntax .....	87
Global Diagnostic Items.....	87
Connection Diagnostic Items .....	89
Subscription Diagnostic Items .....	90
OPC UA Client Diagnostic Items .....	91
Data Quality .....	92
<b>Chapter 9 Troubleshooting.....</b>	<b>95</b>
Using Troubleshooting Tools .....	95
Monitoring Connectivity Status with a Data Source .....	95
Monitoring the Status of Conversations with DDE/SuiteLink Clients .....	96
Using DDEStatus and IOStatus in Excel .....	96
Reading Values from the Gateway into Excel .....	97
Error Messages and Codes .....	97
DDE/SuiteLink Client to Any Data Source – Write Errors .....	97
OPC Client to ArchedrA – Write Errors.....	97
OPC Client to DDE/SuiteLink Data Source – Write Errors.....	98
Runtime Diagnostics and Error Reporting .....	98
Communication Failures .....	99
<b>Chapter 10 Reference.....</b>	<b>101</b>
The Gateway Architecture .....	101
OI Gateway.....	101
Plug-ins.....	102
OI Server Engine .....	102
Data Source Protocol Layer .....	103
Component Environments .....	103
Managing Digital Certificates .....	103

---

Installing a Self-Signed Certificate and Key Pair .....	103
Creating a Self-Signed Certificate and Key Pair.....	104
Using the Self-Signed Certificate and Key Pair .....	106
Adding a Certificate to the Trusted List .....	107
Index .....	111



# CHAPTER 1

## Introduction to the OI Gateway

### In This Chapter

About the OI Gateway .....	9
Supported Data Sources and Clients .....	10
Supported Client Protocols .....	10

### About the OI Gateway

The Wonderware Operations Information Supervisory Gateway (referred to as OI Gateway), acts as a communications protocol converter. OI Gateway can be used to link clients and data sources that communicate using different protocols.

This user assistance publication covers only the information you need to configure and run the OI Gateway component. The documentation that accompanies the related components provide details on their operation.

You can troubleshoot problems with OI Gateway using the ArcestrA Log Viewer, a snap-in to the System Management Console (SMC). See the Log Viewer help file to find information on:

- Viewing error messages.
- Determining which messages are shown.
- Bookmarking error messages.

You may also be able to troubleshoot problems using your client application, such as the Wonderware InTouch HMI software. The client application can use system device items to determine the status of nodes and the values of some parameters.

The basic rules for OI Gateway include:

- One instance of OI Gateway can run per node.
- The Gateway can be configured to run as a service (auto or manual) or not as a service.
- The Gateway can be activated and deactivated using the OI Server Manager snap-in.
- The Gateway can be activated as a COM Server (OPC Server) using standard COM activation mechanisms.
- The Gateway can be run only out-of-proc within OPC clients.
- The Gateway can communicate only with ArcestrA data source components delivered with Wonderware Application Server v2.0 and later. Earlier versions of Wonderware Application Server are not supported.

## Supported Data Sources and Clients

OI Gateway allows Windows application programs access to data from a variety of data sources. The following matrix indicates supported source/client mappings. Clients are listed in the left column, data sources are displayed across the top row, and N/A means not supported.

	OPC v2.05 OI Server	SuiteLink I/O Server	ArchestrA	DDE I/O Server	FastDDE v2 I/O Server	FastDDE v3 I/O Server	InTouch	OPC UA Server
OPC Client	N/A	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SuiteLink Client	Yes	N/A	Yes	Yes	Yes	Yes	Yes	Yes
DDE Client	Yes	Yes	Yes	N/A	N/A	N/A	Yes	Yes
FastDDE v2 Client	N/A	N/A	Yes	N/A	N/A	N/A	N/A	N/A
FastDDE v3 Client	Yes	N/A	Yes	N/A	N/A	N/A	N/A	N/A

**Note:** InTouch v7.11 and greater is supported. FastDDE v2 supports value data only. FastDDE v3 supports VTQ (value, time, quality). All versions of DDE must be local (NetDDE is not supported). OI Gateway must be located on the same node as ArchestrA in order to use that data source.

To access OI Gateway, the chosen client must also have a valid configuration, which is client-specific.

## Supported Client Protocols

The client applications connect to OI Gateway using the following protocols:

- *OPC*
- *SuiteLink*
- *DDE/FastDDE* (on page 11)

## OPC

OPC (originally OLE for Process Control, now Open Platform Communications) is a non-proprietary set of standard interfaces based on Microsoft's OLE/COM technology. This standard makes possible interoperability between automation/control applications, field systems/ devices, and business/office applications.

Avoiding the traditional requirement of software/application developers to write custom drivers to exchange data with field devices, OPC defines a common, high-performance interface that permits this work to be done once, and then easily reused by HMI, SCADA, control and custom applications.

Over a network, OPC uses DCOM (Distributed COM) for remote communications.

## SuiteLink

SuiteLink uses a TCP/IP-based protocol and is designed specifically to meet industrial needs such as data integrity, high throughput, and easier diagnostics. This TCP/IP standard is supported on Windows operating systems.

SuiteLink is not a replacement for DDE or FastDDE. The protocol used between a client and a server depends on your network connections and configurations. SuiteLink provides the following features:

- Value Time Quality (VTQ) places a time stamp and quality indicator on all data values delivered to VTQ-aware clients.
- Extensive diagnostics of the data throughput, server loading, computer resource consumption, and network transport are made accessible through the operating system's performance monitor. This feature is critical for the operation and maintenance of distributed industrial networks.
- Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.
- The network transport protocol is TCP/IP using Microsoft's standard WinSock interface.

## DDE/FastDDE

DDE/FastDDE communication protocols allow communication between a client and a server. DDE protocol is developed by Microsoft whereas FastDDE protocol is proprietary to Wonderware.

---

**Important:** On Windows Vista and later operating systems, Local DDE is supported only when the OI Server is configured as "Not a Service" and activated from its executable file or launched from InTouch. Local DDE is not supported when the OI Server is activated from the System Management Console (SMC).

---

## DDE

DDE is a communications protocol to allow applications in the Windows environment to send/receive data and instructions to/from each other. It implements a Client/Server relationship between two concurrently running applications.

The server application provides the data and accepts requests from any other application interested in its data. Requesting applications are called clients. Some applications such as InTouch and Microsoft Excel can simultaneously be both a client and a server.

## FastDDE

FastDDE provides a means of packing many proprietary Wonderware Dynamic Data Exchange messages into a single Microsoft DDE message. This packing improves efficiency and performance by reducing the total number of DDE transactions required between a client and a server.

Although Wonderware's FastDDE has extended the usefulness of DDE for our industry, this extension is being pushed to its performance constraints in distributed environments.

## ArchestrA Message Exchange

Message Exchange is a proprietary communication protocol used by Invensys's ArchestrA infrastructure. It provides data communication across ArchestrA's object-based system.

## OPC UA

OPC Unified Architecture (OPC UA) is an industrial machine-to-machine communication protocol for interoperability. It provides process control with enhanced security, advanced communication, security, and information models, and cross-platform connectivity.

OPC UA is implemented as a client in OI Gateway.

OPC UA differs significantly from OPC. The following provides the key differences between classic OPC and OPC UA.

<b>Classic OPC</b>	<b>OPC UA</b>
Uses the COM/DCOM technology of Microsoft to communicate. It does not have configurable time-outs. It depends on the DCOM time-out, which is configured in the system.	Uses a services architecture to export data, which improves the ease of communication and connectivity.
Is dependent on Windows operating systems.	Is platform independent and can connect to a wide variety of devices and platforms.
Has limited security.	Has built-in security.
No built-in capabilities to handle problems, such as lost messages.	Has built-in capabilities to handle problems, such as lost messages.

## MQTT

MQTT, formerly called Message Queuing Telemetry Transport, is a publish/subscribe messaging protocol for use over TCP/IP. MQTT is designed to ensure that devices can communicate with each other while minimizing power and bandwidth requirements. It is a simple messaging protocol that is well-suited for use with devices that rely on slow or unreliable networks.

The MQTT protocol is an application layer specification, and has been published as standard ISO/IEC PRF 20922. MQTT uses a Publish-Subscribe mechanism which requires a mediating broker. The publishers send data to the broker, and subscribing clients receive data published to the broker. Only clients that have subscribed to a particular topic receive messages about that topic. The protocol supports bidirectional communication such that a device that is a publisher can also receive updates.

# CHAPTER 2

## Gateway Configuration

### In This Chapter

Configuring the Gateway.....	13
Before Configuring the Gateway .....	14
The Gateway Data Source Hierarchy .....	15
Configuring Data Source Redundancy.....	16

### Configuring the Gateway

Network Communication Bridge/Interface Modules are the communication links between OI Gateway and its supported Data Sources (OPC Server, Application Server, InTouch, DDE/SuiteLink, and OPC UA). You must create these links within the OI Server Manager hierarchy to bridge/route control and information data between different networks to target data sources.

---

**Important:** To run OI Gateway as a service, right-click OI Gateway name (**OI.Gateway.1**) under OI Server Manager and select **Configure As Service** from the shortcut menu. You can configure it as an auto service or manual service. (For more information about configuring as a service see the Activation/Deactivation/ Service Component of the OI Server Manager documentation.) OI Gateway must be run as a service if you are using a DDE/SuiteLink client.

---

OI Gateway is hosted by the OI Server Manager, a Microsoft Management Console (MMC) snap-in, which is part of the ArchestrA System Management Console (SMC) suite of utilities. Many high-level functions and user-interface elements of the OI Server Manager are universal to numerous products created with the ArchestrA DAS Toolkit. Only the documentation for the OI Server Manager contains descriptions of those universal functions/UI elements. Therefore, reading the documentation for both the MMC and the OI Server Manager is critical to understanding this user's guide. To read the documentation about the MMC and OI Server Manager, click the **Help** command on the SMC's **Action** menu. Both the MMC's help and the OI Server Manager's help are displayed. An Adobe Acrobat version of the OI Server Manager documentation (filename: `OIServerManager.pdf`) is also available in the CD-ROM folder `\User Docs\English`.

---

**Note:** The shortcut menu items described in this document typically represent only a subset of any actual shortcut menu. Most items in each shortcut menu are standard Microsoft Windows or MMC-specific commands. For more information about those commands, please see MMC help.

---

#### To prepare OI Gateway

1. Install OI Gateway by double-clicking the **Setup-OI-Gateway.msi** file.
2. Start the System Management Console. by clicking **Start**, pointing to **Programs** and then **Wonderware**, and then clicking **System Management Console**.
3. From the System Management Console, find the Operations Integration Server Manager and then OI Gateway below in the hierarchy tree.

- Click on **Default** Group, then the **Local** node.
- Under the Local node, find **Operations Integration Supervisory Servers**, then **Wonderware - Gateway**, then **OI.Gateway.1**.
- See the OI Server Manager documentation for general information about working in this snap-in environment.

---

**Important:** Selecting the **Configuration** object of the hierarchy tree displays the **Global Parameters** configuration view for OI Gateway. The default Poke Mode settings for OI Gateway is Optimization mode. If you intend to put more than 5,000 items on advise, we recommend that you set Transaction Message Timeout to 120 seconds. Configure all other global parameters as required. For more information about the **Global Parameters** configuration view, including descriptions of the different Poke Modes, see the OI Server Manager documentation. Global parameters that appear dimmed are either not supported or cannot be configured in OI Gateway. Simulation Mode is not supported.

---

4. Before activating OI Gateway for connection, you must first build and configure a hierarchy of one or more data sources to establish communications between sources and clients.

---

**Note:** For step-by-step procedures on how to build and configure this hierarchy, please see *OI Gateway Data Source Hierarchy*.

---

5. You may create desired groups and topics for each data source by:
  - Navigating to a data source object in the OI Server Manager tree view.
  - Right-clicking the object and selecting the group/topic object provided. Each data source has only one type of group or topic object that can be added to the hierarchy.
  - Configure the group or topic.
6. Finally, you may create desired device items for each group or topic by:
  - Selecting the group or topic object.
  - Clicking the **Device Items** tab.
  - Right-clicking anywhere in the **Device Items** configuration view and clicking **Add** from the shortcut menu.

---

**Important:** For step-by-step procedures on configuring Device Items, please see *Configuring Device Item Definitions*.

**Note:** When you are viewing the configuration hierarchy of OI Gateway and someone views the same OI Gateway in another instance of the OI Server Manager, the second instance is displayed in read-only mode. To gain configuration access in this second instance, you must close the first instance of the OI Server Manager (or just remove focus from OI Gateway hierarchy) and then toggle focus away from and then on OI Gateway hierarchy of the second instance.

---

OI Gateway is now ready for use. In order to use it, you must activate it. The following rules apply:

- If you are using an OPC Client, OI Gateway can be auto-started.
- If you are using DDE/SuiteLink, you must start OI Gateway either as a manual or automatic service.
- To activate OI Gateway, right-click **OI.Gateway.1** and click **Activate Server** on the shortcut menu.

## Before Configuring the Gateway

Please note the following important information before you configure the Gateway:

Windows Server operating systems' ICMP(v4) Echo Request may be disabled. You must enable this for an OPC client such as OI Gateway to connect to the server.

To determine the status of ICMP (v4) Echo on the server, open a command prompt on the OI Gateway computer and type:

```
ping SERVER_COMPUTER
```

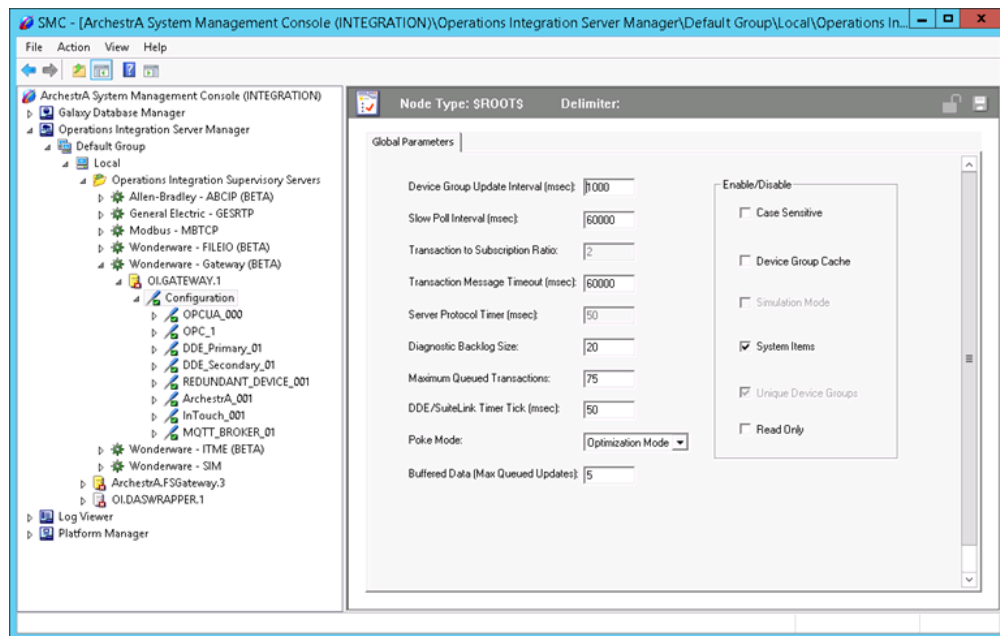
Where "SERVER\_COMPUTER" is the name or IP address of the remote OPC computer.

If the remote computer does not respond to the ping command, enable ICMP(v4) Request on the remote computer according to the procedure for the Microsoft Windows operating system in use on that computer.

## The Gateway Data Source Hierarchy

**Note:** Before attempting to configure OI Gateway, determine the hierarchical structure of the data sources you wish to use.

The data source configuration part of OI Gateway hierarchy begins under the Configuration branch.



**Note:** The default name created from adding a hierarchy object is in the format of **New\_<ObjectName>\_###**, where **<ObjectName>** is the name of the object type and **###** is a numeric value starting from "000" enumerated sequentially per hierarchy object. The hierarchy object name can contain up to 32 characters. The link name for the OPC items is constructed by assembling the respective object names of the nodes along the hierarchy tree in the logical order, starting from the data source root down to the leaf. Therefore, the link name is always unique.

For information about configuring specific data sources, see:

- *Connecting to an ArchestrA Data Source* on page 19
- *Connecting to a DDE/SuiteLink Data Source* on page 35
- *Connecting to an InTouch Data Source* on page 41
- *Connecting to an MQTT Data Source* on page 45
- *Connecting to an OPC Data Source* on page 28

- *Connecting to an OPC UA Data Source* on page 53

## Configuring Data Source Redundancy

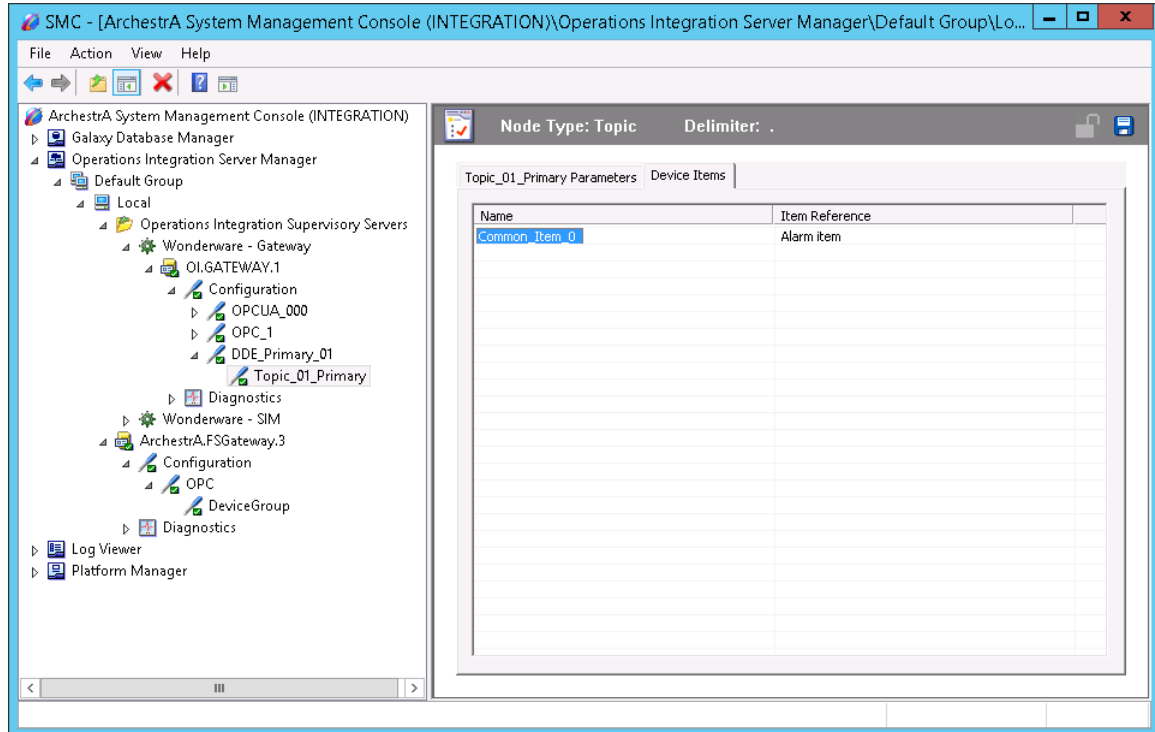
The OI Server Manager provides the ability to assign redundant devices for fail-over protection in the event of data source failure. Two data sources must be configured in the same OI Server having identical item syntax.

Primary and secondary devices will be setup in the REDUNDANT\_DEVICE object in the SMC, along with a common item name (ping item) shared by each device to determine device status.

**Note:** Redundant Device Object does not support the ArchestrA object. There can only be one ArchestrA object per hierarchy.

### To setup up a REDUNDANT\_DEVICE from the configuration branch:

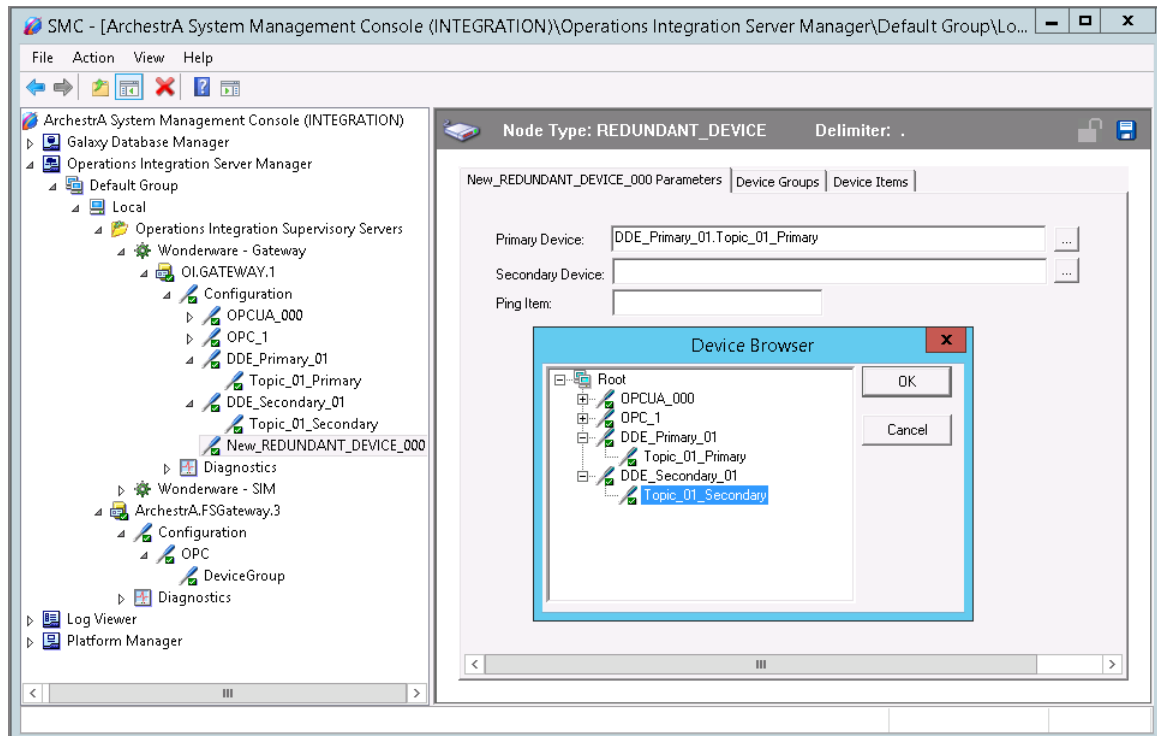
1. Set-up a primary device and hierarchy in the OI Server Manager in the SMC.
2. Create at least one device item that can be shared between the primary and secondary devices to determine device status.



3. Set up a secondary device on the same OI Server. Once again, create an identical device item within the secondary device so that device status can be determined.
4. Select and right-click on the configuration node .
5. Select **Add REDUNDANT\_DEVICE Object**. An object called **New\_REDUNDANT\_DEVICE\_000** is created.
6. Rename the newly created object as appropriate. The **New\_REDUNDANT\_DEVICE\_000** configuration view is displayed in the Configuration branch of the hierarchy.



- Enter or use the device browser to select the primary and secondary devices. Save the hierarchy node configuration by clicking on the save icon.



**Note:** Unsolicited message configuration is not supported from the device redundant hierarchy.

**Important:** A Ping item must be specified and be a valid tag in both the primary and secondary controllers to determine the connection status for \$SYS\$Status. The Ping item can be a static item in the device such as a firmware version or processor type. If the Ping item is invalid or does not exist in the controller, the failover operation may not work correctly as the value of \$SYS\$Status may continue to stay as FALSE in the standby device.



# CHAPTER 3

## Connecting to Data Sources

### In This Chapter

Connecting to an ArcestrA Data Source .....	19
Connecting to an OPC Data Source .....	28
Connecting to a DDE/SuiteLink Data Source .....	35
Connecting to an InTouch Data Source .....	41
Connecting to an MQTT Data Source .....	45
Connecting to an OPC UA Data Source .....	53

## Connecting to an ArcestrA Data Source

To connect to an ArcestrA data source, create and configure its hierarchy (data source and groups), and use the proper item naming conventions in its client(s).

Refer to *Configuration* for a general overview about configuring data sources in OI Gateway.

For more information, refer to one of the following help topics:

- *Configuring an ArcestrA Data Source Object*
- *Configuring an ArcestrA Group Object*
- *Configuring OPC UA Device Items* on page 59
- *ArcestrA Item Names*
- *ArcestrA Data Conversion*

## Configuring an ArcestrA Data Source Object

### To add an ArcestrA data source object to your OI Gateway hierarchy

1. Right-click **Configuration** in the hierarchy, and select **Add ArcestrA Object** from the shortcut menu. The following rules apply:
  - A new object is created in the hierarchy tree and is named **New\_ArcestrA\_000** by default (in "edit mode"). Rename it, if desired. You are allowed to add only one ArcestrA data source.

The **New\_ArcestrA\_000 Parameters** configuration view (right pane) is displayed.

2. Configure the new ArcestrA object according to the following option definitions:
  - **Device Group Name** – Name of the topic to which DDE or SuiteLink clients of FS Gateway connect in order to access items in the ArcestrA data source. Default value is ArcestrA (this cannot be edited).

- **Reconnect Attempts** – Number of times OI Gateway attempts to reconnect to the specified data source if a connection fails. The value (-1) means no limit to the number of attempts. The value Zero (0) means no attempts. Minimum/maximum range is -1 to 1,000,000. Default value is 3. Entry of a value that is excessively out of the allowed range will display an error message about illegal format.
- **Reconnect Period** – Delay (in ms) between reconnection attempts if a connection fails. Minimum/maximum range is 10,000 to 300,000 ms (corresponding to the range of 10 sec to 5 min). Default value is 30000 ms. Entry of a value that is excessively out of the allowed range will display an error message about illegal format.
- **Write Credentials** – User credentials created in ArcestrA for write qualifications.
- **Read Only** – Check this box to make all items connected through the ArcestrA data source read only. This qualification is in addition to any read-only condition that ArcestrA imposes. Unchecking this box only removes OI Gateway-imposed read-only qualifications. In other words, items inherently read-only in the data source remain so. Default value is checked.
- **Domain** – This option, along with **User Name** and **Password**, comprise the credentials used to logon to ArcestrA if the **Read Only** box is unchecked and ArcestrA has security enabled. In such a case, you must enter valid credentials as configured in ArcestrA. Default value is blank.

---

**Note:** The **Domain** option should have a valid domain name when the ArcestrA security authentication mode is "OS Users" or "OS Groups". This option should be left empty when the ArcestrA security authentication mode is "Galaxy".

---

- **User Name** – This option, along with **Domain** and **Password**, comprise the credentials used to logon to ArcestrA if the **Read Only** box is unchecked and ArcestrA has security enabled. In such a case, you must enter valid credentials as configured in ArcestrA. Default value is blank.
- **Password** – This option, along with **Domain** and **User Name** comprise the credentials used to logon to ArcestrA if the **Read Only** box is unchecked and ArcestrA has security enabled. In such a case, you must enter valid credentials as configured in ArcestrA. Default value is blank.

---

**Note:** ArcestrA user login data is not hot-configurable. OI Gateway must be restarted for the new values to take affect.

---

## Configuring an ArcestrA Group Object

Although the ArcestrA namespace is flat, ArcestrA groups provide an artificial grouping hierarchy. Items are added in the same way at both the ArcestrA data source and group levels. In both cases, the same ArcestrA attribute is referenced, the exception being the ArcestrA Item ID Prefix that is provided at the group level.

### To add a group object to your ArcestrA data source hierarchy

1. Select the new data source object, right-click it, and then click **Add ArcestrAGroup Object** on the shortcut menu.
  - A new object is created in the hierarchy tree and is named **New\_ArcestrAGroup\_000** by default (in "edit mode"). Rename it, if desired. You are allowed to add up to 100 new group objects.

---

**Note:** Do not configure an ArcestrA group name to be identical with an item prefix. This name clash could cause unexpected behavior. Identical means the same in a case-insensitive manner. For more information, see *Using Item Prefixes*. Naming an ArcestrA group the same as an item (or the first part of an item name) also causes an ambiguity in OI Gateway namespace. In other words, do not name an ArcestrA group "Float" if a "Float.PV.Value" item exists in the Galaxy.

---

The **New\_ArcestrAGroup\_000 Parameters** configuration view (right pane) is displayed.

2. Configure the new group object according to the following option definitions:
  - **Device Group Name** – Name of the topic that DDE or SuiteLink clients of OI Gateway connect to in order to access items at the ArcestrA group. Default value is the concatenation of the ArcestrA object's name and the group object's name (this cannot be edited).
  - **ArcestrA Item ID Prefix** – A string prefixed to item names added through this ArcestrA group. For instance, a prefix of "Blower\_" would be added to an item such as "001.Temp1" to create an item request of "Blower\_001.Temp1". Default value is blank.
  - **Read Only** – Check this box to make all items connected through the ArcestrA group read only. This qualification is in addition to any read-only condition that ArcestrA imposes. Unchecking this box only removes OI Gateway-imposed read-only qualifications. In other words, items inherently read-only in the data source remain so. Default value is unchecked.

---

**Important:** Each group or topic must be uniquely named for the data source associated with it.

---

## Configuring ArcestrA Device Items

ArcestrA data sources allow you to add items either at the data source branch of the hierarchy or through group objects.

See *Device Item Definitions* for more on Device Items.

You can add items directly to the ArcestrA data source branch or in a group that allows you to group related ArcestrA tagnames together.

To add device items to your group, select the new group object and click the **Device Items** tab. For more information, see *Configuring Device Item Definitions*.

## ArcestrA Item Names

---

**Important:** Writes are not supported from OI Gateway to an ArcestrA item whose security is configured as VerifiedWrite. Writes to items configured as SecuredWrite are supported.

---

This section describes how a connected client requests access to items (or attributes) of a particular ArcestrA data source.

The following are examples of pairs of client/data source connections via OI Gateway, and their associated item name syntax:

- To access an item in ArcestrA via OI Gateway through an OPC client, use the following syntax:

**Establish connection:**

"OI.Gateway.1"

**Reference item:**

"ArcestrA.TIC101.PV"

- To access an item in ArcestrA via OI Gateway through a DDE or SuiteLink client, use the following syntax:

**Establish connection:**

Application = Gateway  
Topic (Device Group) = Archestra  
**Reference item:**  
"TIC101.PV"

## Example #1

Assume the InTouch data source object is named "MyInTouch".

### OPC Client

Access the same TankLevel item through an OPC client as follows:

MyInTouch.TankLevel

### DDE/SuiteLink Client

DDE and SuiteLink clients add items to the Device Group associated with the given InTouch data source object. To access the item in an InTouch data source via OI Gateway through a DDE or SuiteLink client, use the following syntax:

**Application:** Gateway  
**Topic (Device Group):** MyInTouch  
**Item (Tagname):** TankLevel  
**Excel cell reference:** =Gateway|MyInTouch!TankLevel

## Example #2

An InTouch data source object allows you to group related InTouch tagnames together under the InTouch group object. Items can be added to InTouch group objects in the same way as they are added directly to the InTouch data source object. The same InTouch tagname is referenced whether the item is added directly to the InTouch data source object or to an InTouch group object.

Assume a configuration with an InTouch data source object called "MyInTouch" and a single group object called "Cleaner".

### OPC Client

OPC clients can add items to either the InTouch data source object or to the group object. Fully qualified OPC item names are created by concatenating the hierarchy tiers, separated by periods. The following two examples are equivalent:

MyInTouch.TankLevel  
MyInTouch.Cleaner.TankLevel

### DDE/SuiteLink Client

DDE and SuiteLink clients add items to the Device Group associated with either the InTouch data source object or its group object. The topic the DDE/SuiteLink client needs to connect to OI Gateway is provided by this Device Group. The Device Group is created automatically when you create either the InTouch data source object or the group object in the hierarchy.

The item name for a DDE or SuiteLink client would be as follows:

**Application:** Gateway  
**Topic (Device Group):**  
MyInTouch  
or

MyInTouch\_Cleaner

**Item:** TankLevel

**Excel cell reference:**

=Gateway|MyInTouch!TankLevel

or

=Gateway|MyInTouch\_Cleaner!TankLevel

## Using Item Prefixes

In addition, you can configure an item prefix for an ArcestrA group. This prefix, which is added at runtime, can simplify item naming for ArcestrA groups in some situations.

Assume the item prefix for the ArcestrA group "Blower" is "Blower\_". Item names added directly through the data source remain unchanged, but the same items added through the "Blower" group are simplified.

### OPC Client Syntax

ArcestrA.Blower\_001.Temp1 (at the data source level)

ArcestrA.Blower.001.Temp1 (at the group level)

### DDE/SuiteLink Client Syntax

Gateway|ArcestrA!Blower\_001.Temp1 (at the data source level)

Gateway|ArcestrA\_Blower!001.Temp1 (at the group level)

---

**Note:** Do not configure an ArcestrA group name to be identical with an item prefix. This name clash could cause unexpected behavior. Identical means the same in a case-insensitive manner.

---

## ArcestrA Data Conversion

A key part of OI Gateway's protocol conversion capabilities is its data type conversion between DDE, SuiteLink, OPC, and ArcestrA Message Exchange sources and clients.

---

**Note:** Since InTouch communicates through DDE or SuiteLink protocols, its data type conversions are covered in the following sections that address DDE and SuiteLink conversion.

---

Each protocol has a set of supported data types for the values that can be accessed. The following sections describe the data conversion mapping scheme applied by OI Gateway.

---

**Note:** If a client pokes an out-of-range value for any data type, OI Gateway does no clamping on the value. OI Gateway passes the client request to the server.

---

**Important:** All pokes greater than 499 characters return Uncertain quality in the client and SMC. The value is successfully poked to ArcestrA but it is truncated to 499 characters on the read-back. Additionally, all data below +/-1.5e-45 is rounded to 0.0.

---

## ArcestrA-DDE/SuiteLink Mappings

The following sections describe ArcestrA to DDE/SuiteLink and DDE/SuiteLink to ArcestrA data conversions.

## ArchestrA to DDE/SuiteLink Conversions

In the case of the gateway receiving data from an ArchestrA source and sending it to a DDE/SuiteLink client, the gateway converts ArchestrA types to DDE/SuiteLink types as follows:

ArchestrA Type	DDE/SuiteLink Type	Comments
Boolean	Discrete	False = 0, True = 1.
Float	Real	
Integer	Integer	
String	String	If too long, truncated and marked Q=Uncertain.
Double	Real	If overflows, marked Q=Bad and set value = NaN.
Time	String	
ElapsedTime	Real	Pass as float seconds; consistent with InTouch behavior.
CustomEnum	String	If too long, truncated and marked Q=Uncertain.
InternationalString	String	If too long, truncated and marked Q=Uncertain.
BigString	String	If too long, truncated and marked Q=Uncertain.
CustomStruct	Not supported	
MxReference	String	If too long, truncated and marked Q=Uncertain.
Datatype	String	
MxStatus	String	If too long, truncated and marked Q=Uncertain.

## DDE/SuiteLink to ArchestrA Conversions

In the case of the gateway receiving (write) data from a DDE/SuiteLink source and sending it to an ArchestrA client, the gateway converts DDE/SuiteLink types to ArchestrA types as follows:

---

**Note:** Write failures can occur if the target ArchestrA attribute is a non-coercible type. In this case, the gateway returns a failed write status to the client.

---



DDE/SuiteLink Type	ArchestrA Type	Comments
Discrete	Boolean	False = 0, True = 1
Real	Float	
Integer	Integer	OI Gateway does no clamping when writing an integer from a DDE/SuiteLink client to an ArchestrA data source. In the case of a client poking a number greater than 2147483647 or -2147483647, the target link changes the data to a 1 or -1, respectively.
String	String	

## ArchestrA–OPC Mappings

The following sections describe ArchestrA to OPC and OPC to ArchestrA data conversions. The following rules follow the OPC Data Access (DA) Specification v2.05.

### ArchestrA to OPC Conversions

In the case of the gateway receiving data from an ArchestrA source and sending it to an OPC client, the gateway converts ArchestrA types to OPC types as follows:

ArchestrA type	OPC Variant Canonical Mapping	Comments
Boolean	VT_BOOL	Discrete (0/1) translates to OPC VT_BOOL.
Float	VT_R4	
Integer	VT_I4	
String	VT_BSTR	If too long, truncated and marked Q=Uncertain.
Double	VT_R4	If overflows, marked Q=Bad and set value = NaN.
Time	VT_BSTR	
ElapsedTime	VT_R4	Pass as float seconds; consistent with InTouch behavior.

CustomEnum	VT_BSTR	If too long, truncated and marked Q=Uncertain.
InternationalString	VT_BSTR	If too long, truncated and marked Q=Uncertain.
BigString	VT_BSTR	If too long, truncated and marked Q=Uncertain.
CustomStruct	Not supported	
MxReference	VT_BSTR	If too long, truncated and marked Q=Uncertain.
Datatype	VT_BSTR	
MxStatus	VT_BSTR	If too long, truncated and marked Q=Uncertain.

### OPC to ArcestrA Conversions

In the case of the gateway receiving (write) data from an OPC source and sending it to an ArcestrA client, the gateway converts OPC types to ArcestrA types as follows:

---

**Note:** Write failures can occur if the target ArcestrA attribute is a non-coercible type. In this case, the gateway returns a failed write status to the client.

---

OPC Variant Type	ArcestrA Type	Comments
VT_EMPTY	Not supported	Reject write.
VT_NULL	Not supported	Reject write.
VT_I2	Integer	
VT_I4	Integer	
VT_R4	Float	
VT_R8	Float	Reject write if outside of valid float range.
VT_CY	String	
VT_DATE	String	
VT_BSTR	String	Reject write if too large.
VT_DISPATCH	Not supported	Reject write.
VT_ERROR	Integer	
VT_BOOL	Boolean	

VT_VARIANT	Not supported	Reject write.
VT_DECIMAL	Float	
FVT_RECORD	Not supported	Reject write.
VT_UNKNOWN	Not supported	Reject write.
VT_I1	Integer	
VT_UI1	Integer	
VT_UI2	Integer	
VT_UI4	Integer	Reject write if too large.
VT_INT	Integer	
VT_UINT	Integer	Reject write if too large.
VT_VOID	Not supported	Reject write.
VT_HRESULT	Integer	
VT_PTR	Not supported	
VT_SAFEARRAY	Not supported	Reject write.
VT_CARRAY	Not supported	Reject write.
VT_USERDEFINED	Not supported	Reject write.
VT_LPSTR	String	Reject write if too large.
VT_LPWSTR	String	Reject write if too large.
VT_FILETIME	String	
VT_BLOB	Not supported	Reject write.
VT_STREAM	Not supported	Reject write.
VT_STORAGE	Not supported	Reject write.
VT_STREAMED_OBJECT	Not supported	Reject write.

VT_STORED_OBJECT	Not supported	Reject write.
VT_BLOB_OBJECT	Not supported	Reject write.
VT_CF	Not supported	Reject write.
VT_CLSID	String	
VT_VECTOR	Not supported	Reject write.
VT_ARRAY	Not supported	Reject write.
VT_BYREF	Not supported	Reject write.
VT_RESERVED	Not supported	Reject write.

## Connecting to an OPC Data Source

To connect to an OPC data source, create and configure its hierarchy (data source and groups), and use the proper item naming conventions in its client(s).

Refer to *Configuration* for a general overview about configuring data sources in OI Gateway.

For more information, refer to one of the following help topics:

- *Configuring an OPC Data Source Object*
- *Configuring an OPC Group Object*
- *Configuring OPC UA Device Items* on page 59
- *OPC Item Names*
- *OPC Data Conversion*

## Configuring an OPC Data Source Object

**To add an OPC data source object to your OI Gateway hierarchy**

1. Right-click **Configuration** in the hierarchy, and select **Add OPC Object** from the shortcut menu. The following rules apply:
  - A new object is created in the hierarchy tree and is named **New\_OPC\_000** by default (in "edit mode"). Rename it, if desired. You are allowed to add an unlimited number of OPC data sources.

The **New\_OPC\_001 Parameters** configuration view (right pane) is displayed.

The screenshot shows the 'OPC\_001 Parameters' configuration window. The title bar indicates 'Node Type: OPC' and 'Delimiter: .'. The main area contains the following fields and options:

- Server Node:  (with a browse button)
- Server Name:  (with a dropdown arrow and a browse button)
- Reconnect Attempts:
- Reconnect Period:  ms
- Poke Retries:
- Activate Server Out of Proc
- Allow Optional Data Type Suffix In Item Name
- Use Synchronous Reads and Writes

2. Configure the new OPC object according to the following option definitions:

- **Server Node** – The computer node on which the specified data source can be found. Default value is localhost. Use the browse button to select from a list of all nodes on your network.
- **Server Name** – ProgID or ClassID of the OPC server (example of a ProgID: OI.Gateway.1, ClassIDs are GUIDs). Use the browse button to select from a list of OPC server ProgIDs on your network. Default value is blank.

---

**Note:** Use ClassID when referencing a server that does not use OPC enum to enumerate a ProgID.

---

- **Reconnect Attempts** – Number of times OI Gateway attempts to reconnect to the specified data source if a connection fails. The value (-1) means no limit to the number of attempts. The value Zero (0) means no attempts. Minimum/maximum range is -1 to 1,000,000. Default value is 3.
- **Reconnect Period** – Delay (in ms) between reconnect attempts if a connection fails. Minimum/maximum range is 10,000 to 300,000 ms (corresponding to the range of 10 sec to 5 min). Default value is 30000 ms.
- **Poke Retries** – Number of times OI Gateway attempts to retry the write operation if a write operation fails. The value zero (0) means no retry attempts. Minimum/maximum range is 0 to 100. Default value is zero (0).

3. Configure OPC options.

- **Activate Server Out of Proc** – When selected, OI Gateway will attempt to start the external OPC Server Out-of-Proc. When not selected, OI Gateway will attempt to start the external OPC Server In-Proc. For more information, see *In-Proc/Out-of-Proc*.
- **Allow Optional Data Type Suffix in Item Name** – When selected and a VT suffix is appended to the item name, OI Gateway adds the item to the OPC server with the requested type set to the type specified in the VT suffix. FS Gateway removes the VT suffix from the item name string before the item is added to the OPC server.

If selected and no VT suffix is appended to the item name, OI Gateway adds the item to the OPC server with requested type = VT\_EMPTY.

If not selected, the item name is passed to the OPC server with requested type = VT\_EMPTY. OI Gateway does not parse the item names looking for a VT suffix.

- **Use Synchronous Reads and Writes** – When selected, items advising reads and writes will be using synchronous API for communication.

## Configuring an OPC Group Object

### To add a group object to your OPC data source hierarchy

1. Select the new data source object, right-click it, and then click **Add OPCGroup Object** on the shortcut menu.
  - A new object is created in the hierarchy tree and is named **New OPCGroup\_000** by default (in "edit mode"). Rename it, if desired. You are allowed to add up to 100 new group objects.

The **New OPCGroup\_000 Parameters** configuration view (right pane) is displayed.

2. Configure the new group object according to the following option definitions:
  - **Device Group Name** – Name of the topic that DDE or SuiteLink clients of OI Gateway connect to in order to access items at the OPC group. Default value is the concatenation of the OPC data source object's name and the group object's name (this cannot be edited).
  - **Update Rate** – Value (in ms) used by OI Gateway to update the OPC group. Minimum/maximum range is 0 to 2147483646 ms. If the OPC server supports it, zero (0) update rate means the data source sends data changes immediately. If the server does not support zero update rate, it typically returns a message including information about its fastest possible update rate. Default value is 1000 ms.  
  
The update rate can also be adjusted at run time by setting the \$SYS\$OPCUpdateInterval system item.
  - **OPC Item ID Prefix** – String prefixed to all item names added to the OPC group. Default value is blank. Example: Item Prefix=40, Item=001, Item requested from data source=40001.
  - **Use Group Name as Access Path** – Provides control over the OPC Access Path for items added to the OPC group. When checked, the name of the OPC group object is used as the OPC Access Path for all items. When unchecked, the default (blank) OPC Access Path is used. Default value is unchecked.
  - **Read Only** – Check this box to make all items connected through the OPC group read only. This qualification is in addition to any read-only condition that the OPC server imposes. Unchecking this box only removes OI Gateway-imposed read-only qualifications. In other words, items inherently read-only in the data source remain so. Default value is checked.
  - **Demand Read After Poke** – Provides an option to read an item immediately after a write transaction on the item. When checked, a read transaction is performed on the item whenever it has a write transaction irrespective of the subscription group interval. When unchecked, the value of the item is updated at the specified subscription interval after a write operation on the item. Default value is unchecked.
  - **Browse OPC Items button** – Opens the OPC browser, in which you can select items directly from the OPC server.

OPC groups are used to model the behavior of OPC servers. You cannot add items directly to an OPC data source, but must add them at the group level.

---

**Important:** Each group or topic must be uniquely named for the data source associated with it.

---

## Configuring OPC Device Items

To add device items to your group, select the new group object and click the **Device Items** tab. For more information, see *Device Item Definitions*.

---

**Note:** When using a VT suffix, the suffix must be specified in the Item Reference column in the Device Items tab, not in the item name subscribed from a client.

---

## OPC Item Names

This section describes how a connected client requests access to items (or attributes) of a particular OPC data source.

You can also use item suffixes for OPC data sources to specify VT data type information.

## Using Item Suffixes

You can configure OPC data sources to use the /VT item suffix to gain better control of data access in OI Gateway. Using the /VT item suffix enables you to tell the underlying OPC server the type of data you want the underlying OPC server to report to OI Gateway.

### OPC Client Syntax

Apply RequestVT (/VT item suffix) to an item name using the following syntax:

```
/VT_<DataType>
```

Example:

If you want to specify 2-byte signed Integer (/VT\_I2) as the data type for an OPC server path of OPCServer.Blower.Int2, the item name in OI Gateway will be subscribed as: .

```
OPCServer.Blower.Int2 /VT_I2
```

### Supported VT Data Types

The following table describes the supported VT types.

VT Suffix	Description
/VT_BSTR	Automation String
/VT_BOOL	Boolean
/VT_I1	Char
/VT_I2	2-byte Signed Integer
/VT_I4	4-byte Signed Integer
/VT_I8	8-byte Signed Integer
/VT_UI1	Unsigned Char
/VT_UI2	2-byte Unsigned Integer
/VT_UI4	4-byte Unsigned Integer
/VT_UI8	8-byte Unsigned integer
/VT_UINT	Unsigned machine Integer

<b>VT Suffix</b>	<b>Description</b>
/VT_INT	Signed machine Integer
/VT_R4	4-byte Real
/VT_R8	8-byte Real
/VT_DATE	Time stamp
/VT_CY	8-byte Currency
/VT_BSTR[ ]	Automation String array
/VT_BOOL[ ]	Boolean array
/VT_I1[ ]	Char array
/VT_I2[ ]	2-byte Signed Integer array
/VT_I4[ ]	4-byte Signed Integer array
/VT_I8[ ]	8-byte Signed Integer array
/VT_UI1[ ]	Unsigned Char array
/VT_UI2[ ]	2-byte Unsigned Integer array
/VT_UI4[ ]	4-byte Unsigned Integer array
/VT_UI8[ ]	8-byte Unsigned Integer array
/VT_UINT[ ]	Unsigned machine Integer array
/VT_INT[ ]	Signed machine Integer array
/VT_R4[ ]	4-byte Real array
/VT_R8[ ]	8-byte Real array
/VT_DATE[ ]	Time stamp array
/VT_CY[ ]	8-byte Currency array
/VT_EMPTY	Data Type suffix option not enabled; OPC server reports the default data type for the tagname

### OPC Connectivity Examples

Assume a configuration with an OPC data source called "ModbusOverOPC" and a single group called "Group1" and an item reference "R1".

### DDE/SuiteLink Client

To access an item in an OPC server via OI Gateway through a SuiteLink client, use the following syntax:

**Establish connection:**

Application = Gateway



Topic = OPCServer1\_OPGroup1

**Reference item:**

"R1"

DDE and SuiteLink clients add items to a Device Group associated with the OPC group. The topic the DDE/SuiteLink client needs to connect to OI Gateway is provided by this Device Group. The Device Group is created automatically when you create the group in the hierarchy. Its name is generated by concatenating the OPC data source name with the group name, separated by an underscore ("\_"). In the example above, the Device Group would be named "ModbusOverOPC\_Group1".

DDE and SuiteLink clients would access items as follows:

```
Gateway|ModbusOverOPC_Group1!Modbus.QT.R1
```

Using the VT syntax, you can specify the 4-byte Integer (/VT\_I4) from the ModbusOverOPC data source for item "R1" as follows:

```
Gateway|ModbusOverOPC_Group1!Modbus.QT.R1 /VTI4
```

---

**Note:** Use the "Device Group Name" as on the faceplate of the OPC Group Node.

---

## InTouch HMI

Using InTouch as a data access client functions very much the same as the DDE/SuiteLink client. Access an item "R1" in an OPC data source, "ModbusOverOPC", via OI Gateway through InTouch, and specify a 4-byte signed Integer as follows:

**Establish connection:**

Application = Gateway

Topic = OPCServer1\_OPGroup1

**Reference item:**

"R1"

In InTouch, add tagnames in the Tagname Dictionary, configured for the defined Access Name, or for "OPC", the default Access Name. Typically, you will configure Access Names for read or read/write as follows:

```
ModbusOverOPC.R1
```

You can specify the 4-byte Integer (/VT\_I4) from the ModbusOverOPC data source for item "R1" as follows:

```
ModbusOverOPC.R1 /VT_I4
```

## OPC Data Conversion

A key part of OI Gateway's protocol conversion capabilities is its data type conversion between DDE, SuiteLink, and OPC sources and clients.

---

**Note:** Since InTouch communicates through DDE or SuiteLink protocols, its data type conversions are covered in the following sections that address DDE and SuiteLink conversion.

---

Each protocol has a set of supported data types for the values that can be accessed. The following section describes the data conversion mapping scheme applied by OI Gateway.

---

**Note:** If a client pokes an out-of-range value for any data type, OI Gateway does no clamping on the value. OI Gateway passes the client request to the server.

---

## DDE/SuiteLink-OPC Mappings

The following sections describe OPC to DDE/SuiteLink and DDE/SuiteLink to OPC data conversions.

## OPC to DDE/SuiteLink Conversions

In the case of the gateway receiving (write) data from an OPC client and sending it to a DDE/SuiteLink data source, the gateway converts OPC types to DDE/SuiteLink types as follows:

OPC Variant Type	DDE/SuiteLink Type	Comments
VT_EMPTY	Not supported	Reject write.
VT_NULL	Not supported	Reject write.
VT_I2	Integer	
VT_I4	Integer	
VT_R4	Real	
VT_R8	Real	On writes, rejected if out of range.
VT_CY	String	
VT_DATE	String	
VT_BSTR	String	On writes, rejected if out of range.
VT_DISPATCH	Not supported	On writes, rejected.
VT_ERROR	Integer	
VT_BOOL	Discrete	
VT_VARIANT	Not supported	On writes, rejected.
VT_DECIMAL	Float	On writes, rejected if out of range.
VT_RECORD	Not supported	On writes, rejected.
VT_UNKNOWN	Not supported	On writes, rejected.
VT_I1	Integer	
VT_UI1	Integer	
VT_UI2	Integer	
VT_UI4	Integer	On writes, rejected if out of range.
VT_INT	Integer	
VT_UINT	Integer	On writes, rejected if out of range.
VT_VOID	Not supported	On writes, rejected.
VT_HRESULT	Integer	
VT_PTR	Not supported	On writes, rejected.
VT_SAFEARRAY	Not supported	Rejects write. On reads, sets quality to Bad.

VT_CARRAY	Not supported	Rejects write.
VT_USERDEFINED	Not supported	On writes, rejected.
VT_LPSTR	String	On writes, rejects if too long.
VT_LPWSTR	String	On writes, rejects if too long.
VT_FILETIME	String	On writes, rejects if too long.
VT_BLOB	Not supported	On writes, rejected.
VT_STREAM	Not supported	On writes, rejected.
VT_STORAGE	Not supported	On writes, rejected.
VT_STREAMED_OBJECT	Not supported	On writes, rejected.
VT_STORED_OBJECT	Not supported	On writes, rejected.
VT_BLOB_OBJECT	Not supported	On writes, rejected.
VT_CF	Not supported	On writes, rejected.
VT_CLSID	String	
VT_VECTOR	Not supported	On writes, rejected.
VT_ARRAY	Not supported	On writes, rejected.
VT_BYREF	Not supported	On writes, rejected.
VT_RESERVED	Not supported	On writes, rejected.

### DDE/SuiteLink to OPC Conversions

In the case of the gateway receiving data from a DDE/SuiteLink source and sending it to an OPC client, the gateway converts DDE/SuiteLink types to OPC types as follows:

**Note:** Conversion failures can occur. In this case, the gateway returns Bad quality to the OPC client.

DDE/SuiteLink Type	OPC Variant Canonical Mapping
Discrete	VT_BOOL
Float	VT_R4
Integer	VT_I4
String	VT_BSTR

## Connecting to a DDE/SuiteLink Data Source

To connect to a DDE/SuiteLink data source, create and configure its hierarchy (data source and topics), and use the proper item naming conventions in its client(s).

Refer to *Configuration* for a general overview about configuring data sources in OI Gateway.

For more information, refer to one of the following help topics:

- *Configuring a DDE/SuiteLink Data Source Object* on page 36
- *Configuring a DDE/SuiteLink Topic Object* on page 36
- *Configuring OPC UA Device Items* on page 59
- *DDE/SuiteLink Item Names* on page 37
- *DDE/SuiteLink Data Conversion* on page 39

## Configuring a DDE/SuiteLink Data Source Object

### To add a DDE or SuiteLink data source object to your OI Gateway hierarchy

1. Right-click Configuration in the hierarchy, and select either **Add DDE Object** or **Add SuiteLink Object** from the shortcut menu. The following rules apply:
  - A new object is created in the hierarchy tree and is named **New\_DDE\_000** or **New\_SuiteLink\_000** by default. You are allowed to add an unlimited number of DDE and SuiteLink data sources.
  - In this step and succeeding steps, each hierarchy entry is added in "edit mode," providing a convenient place for you to appropriately name components of your specific environment. If you do not rename the object at this time, the numeric sequence system is applied. Any hierarchy entry can be renamed at a later time.

The **New\_DDE\_000 Parameters** or **New\_SuiteLink\_000 Parameters** configuration view (right pane) is displayed.

2. Configure the new DDE or SuiteLink object according to the following option definitions:
  - **Server Name** – Name of the DDE or SuiteLink server you want to use as a data source (for instance, Invensys's DASABTCP). Default value is MyServer. **Server Name** can be from 1 to 32 characters long (cannot be blank), and all printable characters are allowed except a space and >: "/\|, . ; ? ' [ ] { } ` ~ ! @ # \$ % ^ & \* ( ) \_ + - =.
  - **Server Node** – The computer node on which the specified data source can be found. This parameter is displayed for SuiteLink only because DDE servers must be located on the same node as OI Gateway. Default value is localhost. Use the browse button to select from a list of all nodes on your network.
  - **Reconnect Attempts** – Number of times OI Gateway attempts to reconnect to the specified data source if a connection fails. The value minus one (-1) means no limit to the number of attempts. The value zero (0) means no attempts. Minimum/maximum range is -1 to 1,000,000. Default value is 3. Entry of a value that is excessively out of the allowed range will display an error message about illegal format.
  - **Reconnect Period** – Delay (in ms) between reconnection attempts if a connection fails. Minimum/maximum range is 10,000 to 300,000 ms (corresponding to the range of 10 sec to 5 min). Default value is 30000 ms. Entry of a value that is excessively out of the allowed range will display an error message about illegal format.

## Configuring a DDE/SuiteLink Topic Object

### To add a topic to your DDE or SuiteLink object

1. Select the new data source object, right-click it, and then click **Add Topic Object** on the shortcut menu.
  - A new object is created in the hierarchy tree and is named **New\_Topic\_000** by default (in "edit mode"). Rename it to match the Topic name as defined in your DDE or SuiteLink data source to be connected. You are allowed to add up to 100 new topic objects.

The **New\_Topic\_000 Parameters** configuration view (right pane) is displayed.

2. Configure the new Topic object according to the following option definitions:
  - **Device Group Name** – Name of the topic that DDE or SuiteLink clients of FS Gateway connect to in order to access items at this topic in the data source. Default value is the concatenation of the DDE or SuiteLink object's name and the Topic object's name (this cannot be edited).
  - **Read Only** – Check this box to make all items connected through this topic read only. This qualification is in addition to any read-only condition the DDE or SuiteLink data source imposes. Unchecking this box only removes OI Gateway-imposed read-only qualifications. In other words, items inherently read-only in the data source remain so. Default value is unchecked.
  - **Topic Name** – Name of the topic in the DDE/SuiteLink data source. Default value is the name of the topic node in the hierarchy. You can change this name by checking the **Change Topic Name** check box.
  - **Change Topic Name** – Check this box to enable the **Topic Name** box so as to change the topic name. Changing the text in the **Topic Name** box has no effect on the name of the topic node in the hierarchy. Default value is unchecked.

Topic objects, which are identical between DDE and SuiteLink data sources, model the behavior of DDE and SuiteLink servers.

---

**Important:** Each group or topic must be uniquely named for the data source associated with it. That is, the topic object name or its **Topic Name** parameter should exactly match a topic defined in the DDE/SuiteLink server data source in a case-insensitive manner.

---

## Configuring DDE/SuiteLink Device Items

DDE and SuiteLink data sources allow you to add items through topic objects that model the behavior of DDE and SuiteLink servers.

Since items are added to topics in DDE and SuiteLink servers, topic objects are required in the DDE/SuiteLink hierarchy if you want to add items.

To add device items to your topic, select the new topic object and click the **Device Items** tab. For more information, see *Device Item Definitions*.

## DDE/SuiteLink Item Names

This section describes how a connected client requests access to items (or attributes) of a particular DDE/SuiteLink data source.

The following is an example of a client/data source connection via OI Gateway, and its associated item name syntax:

- To access an item in a DDE/SuiteLink server via OI Gateway through an OPC client, use the following syntax:

**Establish connection:**

"OI.Gateway.1"

**Reference item:**

"ABTCPDDE.FastTopic.N7:0"

### Example #1

Assume the InTouch data source object is named "MyInTouch".

### OPC Client

Access the same TankLevel item through an OPC client as follows:

MyInTouch.TankLevel

### DDE/SuiteLink Client

DDE and SuiteLink clients add items to the Device Group associated with the given InTouch data source object. To access the item in an InTouch data source via OI Gateway through a DDE or SuiteLink client, use the following syntax:

**Application:** Gateway

**Topic (Device Group):** MyInTouch

**Item (Tagname):** TankLevel

**Excel cell reference:** =Gateway|MyInTouch!TankLevel

## Example #2

An InTouch data source object allows you to group related InTouch tagnames together under the InTouch group object. Items can be added to InTouch group objects in the same way as they are added directly to the InTouch data source object. The same InTouch tagname is referenced whether the item is added directly to the InTouch data source object or to an InTouch group object.

Assume a configuration with an InTouch data source object called "MyInTouch" and a single group object called "Cleaner".

### OPC Client

OPC clients can add items to either the InTouch data source object or to the group object. Fully qualified OPC item names are created by concatenating the hierarchy tiers, separated by periods. The following two examples are equivalent:

MyInTouch.TankLevel

MyInTouch.Cleaner.TankLevel

### DDE/SuiteLink Client

DDE and SuiteLink clients add items to the Device Group associated with either the InTouch data source object or its group object. The topic the DDE/SuiteLink client needs to connect to OI Gateway is provided by this Device Group. The Device Group is created automatically when you create either the InTouch data source object or the group object in the hierarchy.

The item name for a DDE or SuiteLink client would be as follows:

**Application:** Gateway

**Topic (Device Group):**

MyInTouch

or

MyInTouch\_Cleaner

**Item:** TankLevel

**Excel cell reference:**

=Gateway|MyInTouch!TankLevel

or

=Gateway|MyInTouch\_Cleaner!TankLevel

## DDE/SuiteLink Data Conversion

A key part of OI Gateway's protocol conversion capabilities is its data type conversion between DDE, SuiteLink, and OPC sources and clients.

---

**Note:** Since InTouch communicates through DDE or SuiteLink protocols, its data type conversions are covered in the following sections that address DDE and SuiteLink conversion.

---

Each protocol has a set of supported data types for the values that can be accessed. The following sections describe the data conversion mapping scheme applied by OI Gateway.

---

**Note:** If a client pokes an out-of-range value for any data type, OI Gateway does no clamping on the value. OI Gateway passes the client request to the server.

---

## OPC–DDE/SuiteLink Mappings

The following sections describe OPC to DDE/SuiteLink and DDE/SuiteLink to OPC data conversions.

### DDE/SuiteLink to OPC Conversions

In the case of the gateway receiving data from a DDE/SuiteLink source and sending it to an OPC client, the gateway converts DDE/SuiteLink types to OPC types as follows:

---

**Note:** Conversion failures can occur. In this case, the gateway returns Bad quality to the OPC client.

---

DDE/SuiteLink Type	OPC Variant Canonical Mapping
Discrete	VT_BOOL
Float	VT_R4
Integer	VT_I4
String	VT_BSTR

### OPC to DDE/SuiteLink Conversions

In the case of the gateway receiving (write) data from an OPC client and sending it to a DDE/SuiteLink data source, the gateway converts OPC types to DDE/SuiteLink types as follows:

OPC Variant Type	DDE/SuiteLink Type	Comments
VT_EMPTY	Not supported	Reject write.
VT_NULL	Not supported	Reject write.
VT_I2	Integer	
VT_I4	Integer	
VT_R4	Real	
VT_R8	Real	On writes, rejected if out of range.
VT_CY	String	

VT_DATE	String	
VT_BSTR	String	On writes, rejected if out of range.
VT_DISPATCH	Not supported	On writes, rejected.
VT_ERROR	Integer	
VT_BOOL	Discrete	
VT_VARIANT	Not supported	On writes, rejected.
VT_DECIMAL	Float	On writes, rejected if out of range.
VT_RECORD	Not supported	On writes, rejected.
VT_UNKNOWN	Not supported	On writes, rejected.
VT_I1	Integer	
VT_UI1	Integer	
VT_UI2	Integer	
VT_UI4	Integer	On writes, rejected if out of range.
VT_INT	Integer	
VT_UINT	Integer	On writes, rejected if out of range.
VT_VOID	Not supported	On writes, rejected.
VT_HRESULT	Integer	
VT_PTR	Not supported	On writes, rejected.
VT_SAFEARRAY	Not supported	Rejects write. On reads, sets quality to Bad.
VT_CARRAY	Not supported	Rejects write.
VT_USERDEFINED	Not supported	On writes, rejected.
VT_LPSTR	String	On writes, rejects if too long.
VT_LPWSTR	String	On writes, rejects if too long.
VT_FILETIME	String	On writes, rejects if too long.
VT_BLOB	Not supported	On writes, rejected.
VT_STREAM	Not supported	On writes, rejected.
VT_STORAGE	Not supported	On writes, rejected.
VT_STREAMED_OBJECT	Not supported	On writes, rejected.
VT_STORED_OBJECT	Not supported	On writes, rejected.
VT_BLOB_OBJECT	Not supported	On writes, rejected.



VT_CF	Not supported	On writes, rejected.
VT_CLSID	String	
VT_VECTOR	Not supported	On writes, rejected.
VT_ARRAY	Not supported	On writes, rejected.
VT_BYREF	Not supported	On writes, rejected.
VT_RESERVED	Not supported	On writes, rejected.

## Connecting to an InTouch Data Source

To connect to an InTouch data source, create and configure its hierarchy (data source and topics), and use the proper item naming conventions in its client(s).

Refer to *Configuration* for a general overview about configuring data sources in OI Gateway.

For more information, refer to one of the following help topics:

- *Configuring an InTouch Data Source Object*
- *Configuring an InTouch Group Object*
- *Configuring OPC UA Device Items* on page 59
- *Adjusting for Time Zones* on page 43
- *InTouch Item Names*
- *InTouch Data Conversion*

## Configuring an InTouch Data Source Object

### To add an InTouch data source object to your OI Gateway hierarchy

1. Right-click **Configuration** in the hierarchy, and select **Add InTouch Object** from the shortcut menu. The following rules apply:
  - A new object is created in the hierarchy tree and is named **New\_InTouch\_000** by default (in "edit mode"). Rename it, if desired. You are allowed to add an unlimited number of InTouch data sources.

The **New\_InTouch\_000 Parameters** configuration view (right pane) is displayed.

2. Configure the new InTouch object according to the following option definitions:
  - **Device Group Name** – Name of the topic that DDE or SuiteLink clients of OI Gateway connect to in order to access items at the InTouch data source. Default value is the InTouch data source object's name (this cannot be edited).
  - **Read Only** – Check this box to make all items connected through the InTouch data source read only. This qualification is in addition to any read-only condition that InTouch imposes. Unchecking this box only removes OI Gateway-imposed read-only qualifications. In other words, items inherently read-only in the data source remain so.. Default value is unchecked.
  - **InTouch Runtime Node** – The name of the node (computer) on which the InTouch application runs. If the InTouch data source is local, value is LocalHost. Click the ellipse button to browse nodes.
  - **Item Browse Path** – The full universal naming convention (UNC) directory path that contains the InTouch Tagname Dictionary file, Tagname.X, for the target InTouch application. The format is: \\Node\directory or Drive:\directory (local or mapped drive)  
The InTouch application directory must be a shared directory. Click the ellipse button to browse to the shared directory.

- **Reconnect Attempts** – Number of times OI Gateway attempts to reconnect to the specified data source if a connection fails. The value (-1) means no limit to the number of attempts. The value Zero (0) means no attempts. Minimum/maximum range is -1 to 1,000,000. Default value is 3.
- **Reconnect Period** – Delay (in ms) between reconnect attempts if a connection fails. Minimum/maximum range is 10,000 to 300,000 ms (corresponding to the range of 10 sec to 5 min). Default value is 30000 ms.
- **Connection Protocol** – The protocol OI Gateway should use to connect to InTouch. Default value is SuiteLink.

---

**Note:** If the **InTouch Runtime Node** option is blank, then the InTouch data source would default to LocalHost.

---

- **Tag Browser button** – Click to open the InTouch Tag Browser, in which you can select InTouch tags for inclusion in the items list on the Device Items tab. See InTouch documentation for information about how to use the Tag Browser. While using the Tag Browser, note that you can use typical Windows operations such as **Ctrl-Click** to toggle selections and **Shift-Click** to multi-select tagnames.

---

**Note:** When a DDE connection fails, the InTouch data source object automatically switches to SuiteLink even though DDE has been configured as its **Connection Protocol**. This happens in instances such as connecting to a remote InTouch node in which NetDDE is not supported.

---

## Configuring an InTouch Group Object

### To add a group to your InTouch object

1. Select the new data source object, right-click it, and then click **Add InTouchGroup Object** on the shortcut menu.
  - A new object is created in the hierarchy tree and is named **New\_InTouchGroup\_000** by default (in "edit mode"). Rename it, if desired. You are allowed to add up to 100 new group objects.

The **New\_InTouchGroup\_000 Parameters** configuration view (right pane) is displayed.

2. Configure the new group object according to the following option definitions:
  - **Device Group Name** – Name of the topic that DDE or SuiteLink clients of OI Gateway connect to in order to access items at the InTouch group. Default value is the concatenation of the InTouch data source object's name and the group object's name (this cannot be edited).
  - **Read Only** – Check this box to make all items connected through the InTouch group read only. This qualification is in addition to any read-only condition that InTouch imposes. Unchecking this box only removes OI Gateway-imposed read-only qualifications. In other words, items inherently read-only in the data source remain so. Default value is unchecked.
  - **InTouch Runtime Node** – The name of the node (computer) on which the InTouch application runs. Default value is the same as the InTouch data source object's InTouch Runtime Node setting (this is not editable).
  - **Item Browse Path** – The path to the InTouch file, Tagname.X. It identifies the InTouch application whose tagname database is accessed by this InTouch group. Default value is the same as the InTouch data source object's Item Browse Path setting (this is not editable).

- **Tag Browser button** – Click to open the InTouch Tag Browser, in which you can select InTouch tags for inclusion in the items list on the Device Items tab of this group. See InTouch documentation for information about how to use the Tag Browser. While using the Tag Browser, note that you can use typical Windows operations such as **Ctrl-Click** to toggle selections and **Shift-Click** to multi-select tagnames.

---

**Note:** Since an InTouch group always belongs to a given InTouch data source object, all of its parameters (except the **Read Only** check box and the Tag Browser button) are implicitly inherited and thus for reference only (non-configurable) from the InTouchGroup configuration view.

---

Although the InTouch tagname database is flat, InTouch groups provide an artificial grouping hierarchy.

---

**Important:** Each group or topic must be uniquely named for the data source associated with it.

---

## Configuring InTouch Device Items

You can add items directly to the InTouch data source branch or in a group that allows you to group related InTouch tagnames together.

To add device items to your group, select the new group object and click the Device Items tab. For more information, see *Configuring Device Item Definitions*.

## Adjusting for Time Zones

*Handling Time Zones with the Time Property* on page 43

*Preserving Time Stamps from the Publishing Source* on page 43

## Handling Time Zones with the Time Property

If you need to share time stamp values across different time zones (Platforms), use the Time data type in every time zone location. However, if you need to share it as string, remember that when converting the Time data type to a string (for example, in a script), it is automatically converted to local time, so you lose the ability to adjust it in a different time zone.

For example, to convert the Time property to a string GMT:

```
Dim localDateTime As System.DateTime;
localDateTime = System.DateTime.Parse( obj.attr.Time );
obj.udStringGMTfromLocalTime= localDateTime.ToUniversalTime().ToString();
```

To convert the string GMT to a string of local time:

```
Dim univDateTime As System.DateTime;
univDateTime = System.DateTime.Parse( obj.udStringGMTfromLocalTime );
Obj.udStringLocalTimeFromGMT = univDateTime.ToLocalTime().ToString();
```

## Preserving Time Stamps from the Publishing Source

In the following cases, if you want to pass only the time stamp, the subscriber gets the time stamp as converted to the local time zone of the publisher and not the time zone of the data source.

Examples of configurations that do not preserve the original time zone are as follows.

In this configuration, GalaxyB:Object1.TimeAttr shows the time adjusted to the local time zone of the GalaxyA OI Gateway and not the time zone of the PLC:

```
PLC.Item <= GalaxyA Object1.IntAttr.Time <= Gateway <= GalaxyB OPCClient <=
Object1.TimeAttr
```

In this configuration, GalaxyB:Object1.TimeAttr shows the time adjusted to the local time zone of the InTouch application and not the time zone of the PLC:

```
PLC.Item <= GalaxyA Object1.IntAttr.Time <= InTouch App I/O Message Tag <= GalaxyB
InTouchProxy <= Object1.TimeAttr
```

To avoid these problems, subscribe to the GalaxyA:Object1.IntAttr value property. This way, both the value and time stamp propagate to GalaxyB:Object1.IntAttr. You can then use the GalaxyB:Object1.IntAttr.Time. For example:

```
PLC.Item <= GalaxyA Object1.IntAttr <= Gateway <= GalaxyB OPCClient <= Object1.IntAttr
PLC.Item <= GalaxyA Object1.IntAttr <= InTouch App I/O Integer Tag <= GalaxyB InTouchProxy
<= Object1.IntAttr
```

In this configuration, the time property propagates from InTouch to Object.IntAttr.Time:

```
PLC.Item <= InTouch I/O Integer Tag <= Galaxy InTouchProxy <= Object.IntAttr
```

## InTouch Item Names

This section describes how a connected client requests access to items (or attributes) of a particular InTouch data source.

The following is an example of a client/data source connection via OI Gateway, and its associated item name syntax:

- To access an item in InTouch via OI Gateway through an OPC client, use the following syntax:

**Establish connection:**

```
"OI.Gateway.1"
```

**Reference item:**

```
"InTouch1.Pump1"
```

An InTouch data source is a special case of DDE and SuiteLink data source. OI Gateway always communicates with InTouch using either DDE or SuiteLink.

Items can be added either directly to the InTouch data source object or to its group object.

### Example #1

Assume the InTouch data source object is named "MyInTouch".

**OPC Client**

Access the same TankLevel item through an OPC client as follows:

```
MyInTouch.TankLevel
```

**DDE/SuiteLink Client**

DDE and SuiteLink clients add items to the Device Group associated with the given InTouch data source object. To access the item in an InTouch data source via OI Gateway through a DDE or SuiteLink client, use the following syntax:

**Application:** Gateway

**Topic (Device Group):** MyInTouch

**Item (Tagname):** TankLevel

**Excel cell reference:** =Gateway|MyInTouch!TankLevel

## Example #2

An InTouch data source object allows you to group related InTouch tagnames together under the InTouch group object. Items can be added to InTouch group objects in the same way as they are added directly to the InTouch data source object. The same InTouch tagname is referenced whether the item is added directly to the InTouch data source object or to an InTouch group object.

Assume a configuration with an InTouch data source object called "MyInTouch" and a single group object called "Cleaner".

### OPC Client

OPC clients can add items to either the InTouch data source object or to the group object. Fully qualified OPC item names are created by concatenating the hierarchy tiers, separated by periods. The following two examples are equivalent:

MyInTouch.TankLevel

MyInTouch.Cleaner.TankLevel

### DDE/SuiteLink Client

DDE and SuiteLink clients add items to the Device Group associated with either the InTouch data source object or its group object. The topic the DDE/SuiteLink client needs to connect to OI Gateway is provided by this Device Group. The Device Group is created automatically when you create either the InTouch data source object or the group object in the hierarchy.

The item name for a DDE or SuiteLink client would be as follows:

**Application:** Gateway

**Topic (Device Group):**

MyInTouch

or

MyInTouch\_Cleaner

**Item:** TankLevel

**Excel cell reference:**

=Gateway|MyInTouch!TankLevel

or

=Gateway|MyInTouch\_Cleaner!TankLevel

## InTouch Data Conversion

Since InTouch communicates through DDE or SuiteLink protocols, refer to its data type conversions in *DDE/SuiteLink Data Conversion* on page 39.

## Connecting to an MQTT Data Source

The MQTT data source is a top-level node that can be added to the OI Gateway configuration.

To connect to an MQTT data source, create and configure its hierarchy (data source and groups), and use the proper item naming conventions in its client(s).

Refer to *Configuration* for a general overview about configuring data sources in OI Gateway. For more information, refer to one of the following help topics:

- *Configuring an MQTT Data Source Connection* on page 46

- *Configuring an MQTT Group Connection* on page 47
- *Configuring MQTT Device Items* on page 48
- *MQTT Topic Names* on page 48

## Configuring an MQTT Data Source Connection

### To add an MQTT data source connection to your OI Gateway hierarchy

1. Right-click **Configuration** in the hierarchy, and select **Add MQTT\_BROKER Connection** from the shortcut menu.

A new connection is created in the hierarchy tree, named "New\_MQTT\_BROKER\_000" by default. Rename it, if desired. Only one MQTT Broker connection can be added to each OI Gateway instance. If you need to connect to multiple brokers, you can create multiple instances of the OI Gateway. See *Instantiating the Gateway* on page 76 for additional information.

---

**Note:** An OI Server Professional level license is required to create multiple instances of the OI Gateway.

---

2. Configure the connection to the MQTT Broker.
  - a. In the **Network Address** field, enter the MQTT Broker's IP address or host name. The number of characters cannot be more than 255. The field cannot be blank.
  - b. Edit the **Port Number** if your MQTT broker uses a non-default port. TCP/IP port 1883 is the default port without security enabled. Port 8883 is the default port for a secure connection to the broker.
3. Click the **Validate Address and Port** button to verify that the MQTT Broker can be accessed. The status of the test is displayed in a dialog. The initial status is "Connecting to host..."
  - If the connection to the MQTT Broker is successful, the final status is "Connection to host successful."
  - If the MQTT Broker cannot be accessed, the final status is "Unable to connect to host." Check that the network address and port number are correct.

### To enable a TLS-based secured connection (optional)

A digital certificate is required to establish a secure connection with an MQTT broker. The digital certificate, also called a public key certificate, confirms the identity of the broker and is also used to encrypt communications with the broker. Trusted digital certificates are issued by official, trusted agencies known as certification authorities (CA), and guarantee the identity of the broker. In contrast, self-signed digital certificates are issued by private parties and do not guarantee the identity of the broker.

When you validate the security setting of the broker connection, three results are possible:

- A green security icon is displayed with the text **Connection to the broker is secured and trusted.**  
The green security icon indicates that connection to the broker is encrypted and the broker's certificate is issued by a trusted certification authority (CA).
- A yellow security icon is displayed with the text, **Connection to the broker is secured and untrusted.**  
The yellow security icon indicates that connection to the broker is encrypted, but the broker's certificate is self-signed and is therefore untrusted. See *Verifying a Self-Signed Certificate from an Untrusted MQTT Data Source* on page 47 for additional information.

- A red security icon is displayed with the text, **Connection to the broker is unsecured and untrusted.**  
The red security icon indicates that the identity of the broker cannot be checked and the connection is unencrypted. Since the broker's identity cannot be verified, it is considered unknown and untrusted.

---

**Note:** Enabling a secured connection is separate from connecting to a broker. Once security has been successfully enabled, it is possible to see a green security icon without being connected to the broker. However, you must be connected to the broker to be able to validate security.

---

1. To enable security, select the **Secure Connection with Broker** checkbox.
2. The port number automatically changes to 8883. Edit the port number if necessary.
3. Click the **Validate Security** button to check that an encrypted connection over TLS can be established with the MQTT Broker.
  - If the security validation is successful, the final status is "Security Validation completed." Click OK, and either the green, yellow or red security icon will be displayed, along with the corresponding description of the connection.
  - If the security validation is not successful, the final status is "Security Validation failed!" Click OK, and edit the settings as necessary.

## Verifying a Self-Signed Certificate from an Untrusted MQTT Data Source

If you have enabled MQTT Connection Security and the OI Gateway determines that a broker is using an untrusted (self-signed) digital certificate, it will allow the connection to the broker, but display a yellow security icon with the message, "**Connection to the broker is secured and untrusted.**" To verify the digital certificate:

1. Click the **Download** button.
2. Click the browse (...) button to view the broker's self-signed certificate. Do not connect to a broker if you do not trust its self-signed certificate.

## Configuring an MQTT Group Connection

### To add a group connection to your MQTT Broker data source hierarchy

1. Select the MQTT Broker connection, right-click it, and then click **Add MQTTGroup Connection** on the shortcut menu.

A new MQTT group connection is created in the hierarchy tree and is named **New\_MQTTGroup\_000** by default. Rename it, if desired.

The **New\_MQTTGroup\_000 Parameters** configuration view (right pane) is displayed.

The Device Group Name field is automatically filled in and cannot be edited. The Device Group Name is used for accessing MQTT data from a DDE/SuiteLink client.

2. Select a level of **Quality of Service (QoS)** from the drop down list. The QoS level determines message delivery parameters, with 0 as the lowest level of service, and 2 as the highest level.
  - **0 - At Most Once:** The message will be delivered no more than one time, meaning that it may not be delivered. There is no backup of the message, and if the connection to the client is lost, the message will not be delivered. No delivery acknowledgement is provided.
  - **1 - At Least Once:** The message will delivered at least one time, but it may be delivered more than once if the sender does not receive an acknowledgement of a successful transmission. The sender stores the message until a receipt acknowledgement is received.
  - **2 - Exactly Once:** The message is delivered one time only. The sender stores the message until it receives confirmation of receipt. This is the safest, and slowest, message transfer mode.

---

**Note:** If the MQTT publisher has a different QoS than the configured QoS, the configured QoS is used.

---

### To enter user credentials for the group connection (optional)

1. Under **Identity**, select the **Enable** checkbox to turn on authentication for subscribing to MQTT messages.

---

**Note:** If you enable this option, it is highly recommended that you enable MQTT Connection Security to protect the username and password.

---

OI Gateway uses the user name and password settings that you enter here to connect to the configured MQTT broker. If the MQTT sources encrypt the payloads with different user names and passwords, you must create additional groups to support them.

2. Enter the user credentials (user name and password) to be used for subscribing to MQTT messages. This must match a valid MQTT user name at run time. However, the user name is not validated during configuration. The password would be used by MQTT publisher to validate the subscriber. Prior to saving user credentials, you can click **Show Password** to briefly display the password in clear text for verification.
3. Click the **Validate Identity** button to verify that the MQTT Broker can be accessed on the configured MQTT channel.

## Configuring MQTT Device Items

Device items provide alternative names for specifying MQTT items. To add device items to your group, select the new group object and click the **Device Items** tab.

---

**Note:** The MQTT syntax is case sensitive.

---

For more information, see *Configuring Device Item Definitions*.

## MQTT Topic Names

Items subscribed to an MQTT broker are known as topics. MQTT data sources use topic names for sending and receiving messages. A topic name can be divided into multiple topic levels. Each level is separated by a forward slash (/). Wild cards are not supported for MQTT topic names in this release.

The following rules apply to topic names:

- Must be at least once character in length
- Names are case sensitive
- Space character is valid and can be included
- Names can include a leading or trailing forward slash, but the forward slash counts as an identifier (/a, a/ and a are all different topic names)
- '/' is a valid topic name
- Topic names cannot include the null character (U+0000)
- Topic names are UTF-8 encoded strings. The maximum encoded length is 65535 bytes

---

**Note:** In general, MQTT supports the usage of "+" as a single level wild card and "#" as a multi-level wild card. However, the Wonderware MQTT driver does NOT currently support usage of wild cards.

---



## Syntax Example

site/area1/mixer4/valve/input

## Using JSON Strings

Data messages are expected to be in string format. The string can be formatted as a JSON key value pair, which will be detected automatically. JSON messages are parsed by the driver to allow the extraction of each key value pair as attributes of an object.

For example, the field device from a pump station transmits a message that includes location, pump running status, oil pressure, and maintenance data. The MQTT topic for this message is `Field/FS785/Status`. The payload for this message could use a JSON formatted message, such as:

```
{"lat":32.95646, "lon":-96.82275, "Pump_running":1, "Oil_Press":67.23, "Maintenance":"Last Maintenance Dec 14-2015"}
```

An application could subscribe to any of the following topics:

Field/FS785/Status

Field/FS785/Status.lat

Field/FS785/Status.lon

Field/FS785/Status.Pump\_Running

Field/FS785/Status.Oil\_Press

Field/FS785/Status.Maintenance

## Configuring an MQTT Publisher

The MQTT protocol allows the OI Server and associated software to communicate with edge devices that have published MQTT data to an MQTT broker. The publisher functionality allows the OI Server to expose, and publish only the specific references that are configured for publishing purposes. Hence, the user can then select a specific set of references whose data will be published to the MQTT broker.

To configure an MQTT publisher, create and configure the hierarchy of the data source and groups. Use the proper item naming conventions in its client(s).

Ensure the following steps are completed before attempting to configure the MQTT publisher.

1. *Configuring an OPC Data Source Object*
2. *Configuring an OPC Group Object*
3. *Configuring an MQTT Data Source Connection* on page 46
4. *Configuring an MQTT Group Connection* on page 47

---

**Note:** We recommend you to set up an MQTT Broker connection before configuring the MQTT Publisher. Refer *Configuring an MQTT Data Source Connection* on page 46 section for steps to configure the MQTT Broker.

---

### Workflow to configure an MQTT Publisher

Follow the workflow given below to configure an MQTT publisher.

#### Add an OPC Data Source Object to your OI Gateway Hierarchy

1. Right-click **Configuration** in the hierarchy, and select **Add OPC Object** from the shortcut menu. The following rules apply:

- A new object is created in the hierarchy tree and is named **New\_OPC\_000** by default (in "edit mode"). Rename it, if desired. You are allowed to add an unlimited number of OPC data sources. The object is named as OPC\_001 in this example.
2. Configure the new OPC object according to the following option definitions:
    - **Server Node** – The computer node on which the specified data source can be found. Default value is localhost. Use the browse button to select from a list of all nodes on your network.
    - **Server Name** – ProgID or ClassID of the OPC server (example of a ProgID: OI.Gateway.1, ClassIDs are GUIDs). Use the browse button to select from a list of OPC server ProgIDs on your network. Default value is blank. We are using the OI.SIM.1 as the OPC server.
    - **Reconnect Attempts** – Number of times OI Gateway attempts to reconnect to the specified data source if a connection fails. The value (-1) means no limit to the number of attempts. The value Zero (0) means no attempts. Minimum/maximum range is -1 to 1,000,000. Default value is 3.
    - **Reconnect Period** –Delay (in ms) between reconnect attempts if a connection fails. Minimum/maximum range is 10,000 to 300,000 ms (corresponding to the range of 10 sec to 5 min). Default value is 30000 ms.
    - **Poke Retries** – Number of times OI Gateway attempts to retry the write operation if a write operation fails. The value zero (0) means no retry attempts. Minimum/maximum range is 0 to 100. Default value is zero (0).

The screenshot shows a configuration window titled "Node Type: OPC" and "Delimiter: .". The main area is labeled "OPC\_001 Parameters" and contains the following fields and options:

- Server Node: localhost
- Server Name: OI.SIM.1
- Reconnect Attempts: 3
- Reconnect Period: 30000 ms
- Poke Retries: 0
- Activate Server Out of Proc
- Allow Optional Data Type Suffix In Item Name
- Use Synchronous Reads and Writes

For more information, see *Configuring an OPC Data Source Object*.

---

**Note:** Currently, only an OPC Data Source can be configured to publish data to an MQTT Broker.

---

### Add an OPC Data Source Group Object

1. Select the new data source object (OPC\_001), right-click it, and then click **Add OPCGroup Object** on the shortcut menu.

- A new object is created in the hierarchy tree and is named **New\_OPGroup\_000** by default (in "edit mode"). Rename it, if desired. You are allowed to add up to 100 negroup objects. We have renamed the group object to **OPGroup\_1** in this example.
2. Configure the new group object according to the following option definitions:
- **Device Group Name** – Name of the topic that DDE or SuiteLink clients of OI Gateway connect to in order to access items at the OPC group. Default value is the concatenation of the OPC data source object's name and the group object's name (this cannot be edited).
  - **Update Rate** – Value (in ms) used by OI Gateway to update the OPC group. Minimum/maximum range is 0 to 2147483646 ms. If the OPC server supports it, zero (0) update rate means the data source sends data changes immediately. If the server does not support zero update rate, it typically returns a message including information about its fastest possible update rate. Default value is 1000 ms.  
  
The update rate can also be adjusted at run time by setting the \$\$SYS\$OPCUpdateInterval system item.
  - **OPC Item ID Prefix** – String prefixed to all item names added to the OPC group. Default value is blank. Example: Item Prefix=40, Item=001, Item requested from data source=40001. In this instance, using the OI Sim server as the selected server, this value is entered as <PORT.PLC.>.

The screenshot shows a configuration window for an OPCGroup. The title bar indicates 'Node Type: OPCGroup' and 'Delimiter: .'. The window has three tabs: 'OPCGroup\_1 Parameters', 'Device Items', and 'MQTT Publish Items'. The 'Parameters' tab is selected and contains the following fields and options:

- Device Group Name:** A text box containing 'OPC\_001\_OPGroup\_1'.
- Update Rate:** A text box containing '1000' followed by a 'ms' label.
- OPC Item ID Prefix:** A text box containing 'PORT.PLC.'.
- Use Group Name as Access Path
- Read Only
- Demand Read After Poke
- Browse OPC Items:** A button.

3. Save the configuration.  
For more information, see *Configuring an OPC Group Object*.

### Add an MQTT Group Connection

1. Select the MQTT Broker connection (BROKER\_01), right-click it, and then click **Add MQTTGroup Connection** on the shortcut menu.
  - A new MQTT group connection is created in the hierarchy tree and is named **New\_MQTTGroup\_000** by default. Rename it, if desired. The group has been renamed to **Group\_1** in this example.

- The Device Group Name field is automatically filled in and cannot be edited. The Device Group Name is used for accessing MQTT data from a DDE/SuiteLink client.
2. Select a level of Quality of Service (QoS) from the drop down list box. The QoS level determines message delivery parameters, with 0 as the lowest level of service, and 2 as the highest level.
    - **0 - At Most Once:** The message will be delivered no more than one time, meaning that it may not be delivered. There is no backup of the message, and if the connection to the client is lost, the message will not be delivered. No delivery acknowledgement is provided.
    - **1 - At Least Once:** The message will delivered at least one time, but it may be delivered more than once if the sender does not receive an acknowledgement of a successful transmission. The sender stores the message until a receipt acknowledgement is received.
    - **2 - Exactly Once:** The message is delivered one time only. The sender stores the message until it receives confirmation of receipt. This is the safest, and slowest, message transfer mode.
  3. Under **Identity**, select the **Enable** checkbox and enter the user credentials (user name and password) for the group connection (optional)
  4. Click the **Validate Identity** button to verify that the MQTT Broker can be accessed on the configured MQTT channel.

---

**Note:** Multiple groups can be added to a single MQTT broker

---

### Add an MQTT Reference Item

1. In the OPCGroup Object parameter, click the **MQTT Publish Items** tab.
  - **Publish User Group:** From the list of broker groups, select a broker group (BROKER\_01.Group\_1).
  - **Unique Item ID:** A unique item ID (UID) for the selected user group is automatically generated by the system. Rename it, if desired. The UID has been renamed to **MB-1** in this example.

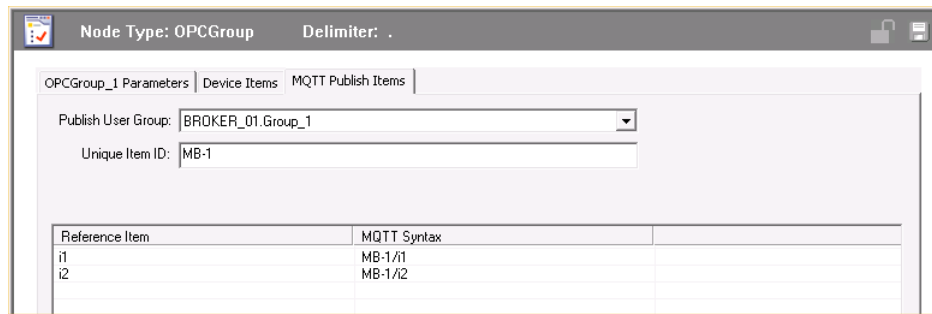
---

**Note:** The characters + and # are considered invalid and cannot be used in the Unique Item ID

---

2. To add a reference item, right click anywhere on the tabular space.
3. Select the **Add** command from the shortcut menu.
  - A device item is created, and it is numerically named by default. For example, Item\_0, Item\_1, and so on.
4. Change the default name by double-clicking on it and entering the new name.
  - Enter a unique name for the new device item (i1 in this example).
5. The MQTT Syntax column is automatically populated with the syntax of the format:  
<Unique Item ID>/<Reference Item>  
In this example, we have MB-1/i1.

6. Multiple reference items can be added for the selected Publish User Group.



7. Save the configuration.

### Publish MQTT Information

- To activate the OI Server, right click OI Gateway.1 and click **Activate Server**.
- The server in this example, OI.SIM.1, activates automatically, as is had been configured as the server we want to use.
- The MQTT information for the reference items is published and can be viewed in the following paths
  - OI.GATEWAY.1 > Diagnostics > Client Groups > MQTT Plugin
  - Wonderware – SIM > OI.SIM.1 > Diagnostics > Client group > OPC\_001.OPCGroup\_1.
- Double click the row to display the Diagnostic info pop-up window showing the following parameters: Name, Client Value, Client Time, Quality, Location, Subscription Message and Device Group.

## Connecting to an OPC UA Data Source

The OPC UA data source is a top-level node that can be added to the OI Gateway configuration.

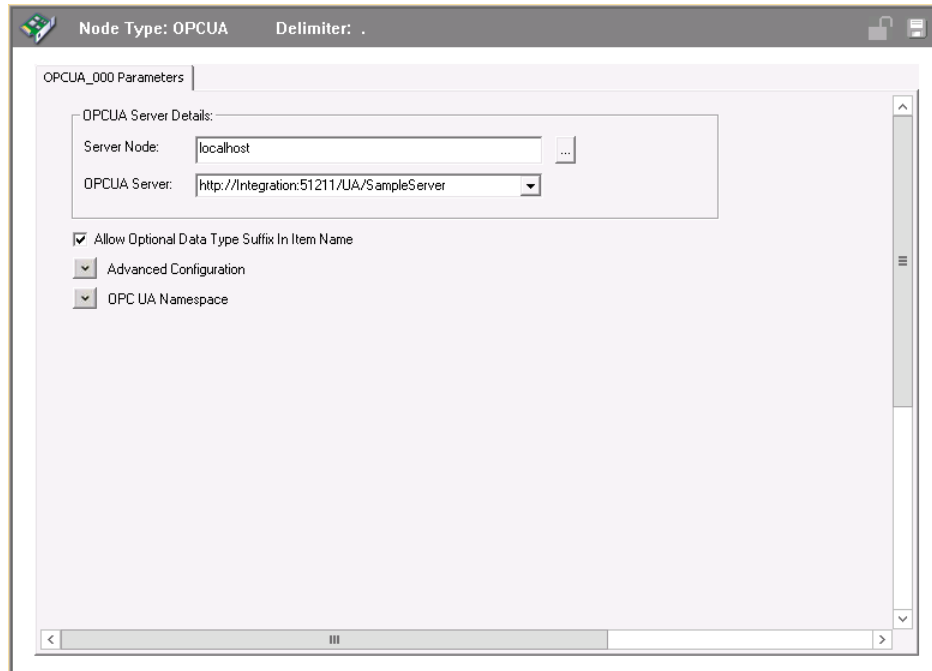
To connect to an OPC UA data source, create and configure its hierarchy (data source and groups), and use the proper item naming conventions in its client(s).

Refer to *Configuration* for a general overview about configuring data sources in OI Gateway.

## Configuring an OPC UA Data Source Object

### To add an OPC UA data source object to your OI Gateway Hierarchy

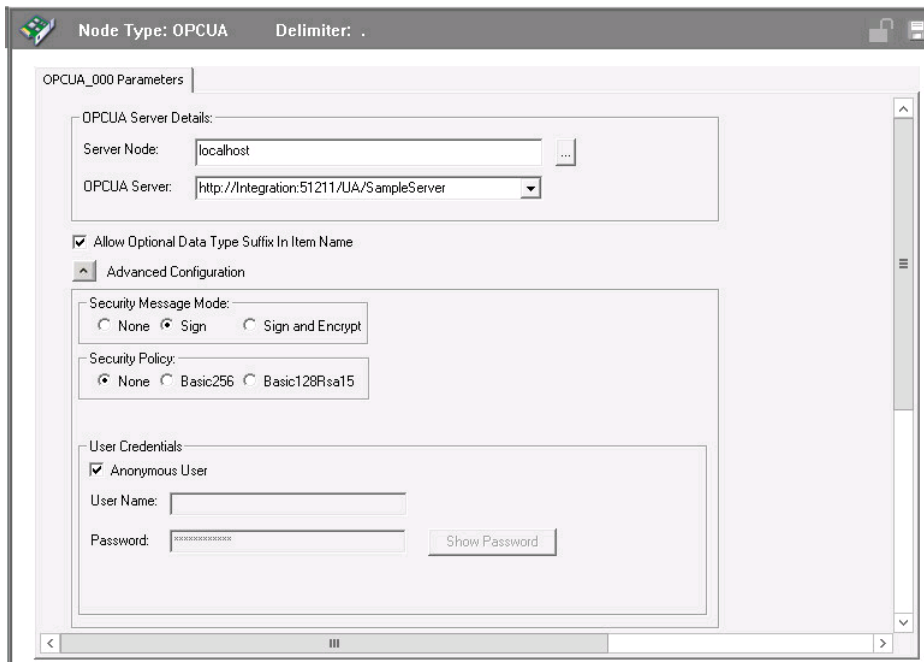
1. Right-click **Configuration** in the hierarchy, and select **Add OPCUA Object** from the shortcut menu. A new object is created in the hierarchy tree, named "New OPCUA\_000" by default. Rename it, if desired. You are allowed to add an unlimited number of OPC UA data sources.



2. Configure the OPC UA object.
  - a. Browse the **Server Node** using the browse button.
  - b. Select an **OPC UA Server** from the server list box.
  - c. Configure **Allow Optional Data Type Suffix in Item Name**. When selected, this parameter adds data type as a suffix to an item name. This parameter is hot configurable. For more information about OPC UA data type suffixes, see *OPC UA Item Names and Syntax* on page 59.
3. Configure OPC UA security using the **Advanced Configuration** items. See *Configuring OPC UA Security* on page 55 for how-to information.
4. Configure the **OPC UA Namespace**. See *Configuring the OPC UA Namespace* on page 56 for how-to information.

## Configuring OPC UA Security

Use the **Advanced Configuration** section of the OPC UA editor to set OPC UA connection security parameters: Security Message Mode, Security Policy, and User Credentials.



### Security Message Mode

The Security Message Mode is set to **None** by default.

Option	Description
None	No security is applied.
Sign	All messages are signed but not encrypted.
Sign and Encrypt	All message are signed and encrypted.

---

**Important:** If the **Security Message Mode** is not set to **None**, the OPC UA Server's digital certificate must be trusted and made available on the OPC UA Client computer. For more information on digital certificates, see *Managing Digital Certificates* on page 103.

---

### Security Policy

Security Policy is set to **None** by default.

Option	Description
None	No security is applied.
Basic256	Indicates that Basic256 security is applied.
Basic128Rsa15	Indicates that Basic128Rsa15 security is applied.

---

**Important:** If the **Security Policy** is not set to **None**, the OPC UA Server's digital certificate must be trusted and made available on the OPC UA Client computer. For more information on digital certificates, see *Managing Digital Certificates* on page 103.

---

### User Credentials

If the **Anonymous User** checkbox is selected, the OPC UA client is allowed to connect to the OPC UA server without credentials. Clear the checkbox to provide a user name and password in their respective text boxes.

You can create a secure connection with the OPC UA server depending on how the server accepts the connection. For example, if your OPC UA client wants to connect to an OPC UA server that does not support anonymous connections, the OPC UA client must provide a valid user name and password.

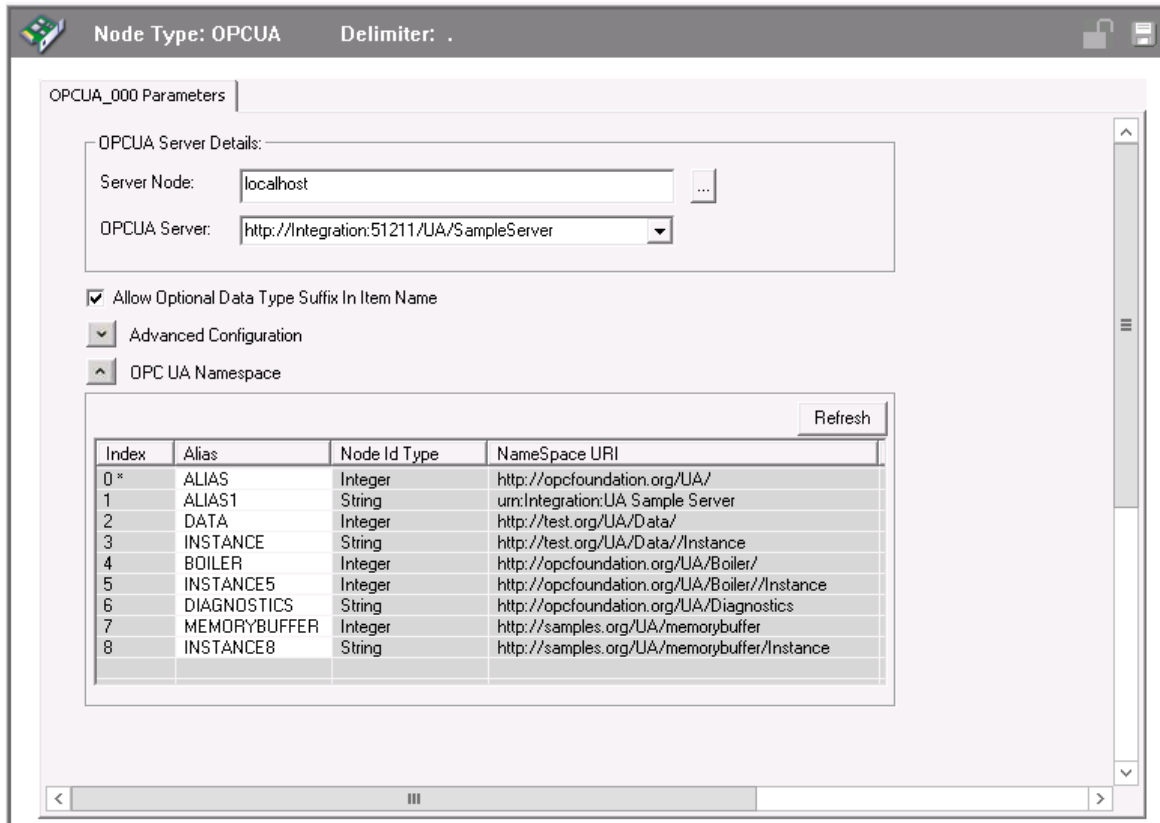
## Configuring the OPC UA Namespace

The OPC UA Namespace is defined and exposed by the OPC UA server. Each namespace is identified by an index ID. You can define an alias name for a namespace according to the following rules:

- The alias name must be defined for the namespace URI exposed by the Gateway.
- The alias name must be in string format.
- There can be no more than one alias per namespace.
- The same alias cannot be given to two OPC UA namespaces.
- The alias must not conflict with any other namespace display name.



The **OPC UA Namespace** section of the OPC UA editor displays the Namespace alias table for the Namespace URI's present in the OPC UA Server to which the OPC UA Client is connected.



The Namespace grid displays the following information:

Column Name	Description
Index	Displays the Namespace index for the Namespace URI present in the OPC UA Server. The default Namespace is appended with * after the Index.  The items configured in the default Namespace need not be subscribed in the syntax using the alias or the Namespace name.
Alias	Displays the alias name for the Namespace URI available in the OPC UA Server.  You can change this name during configuration. The Alias box cannot be blank. You can use the "_" and "#" special characters in the alias name. You cannot create duplicate alias names.
Node ID Type	Displays the Node Id type of the Namespace in the OPC UA Server.  Node Id type can be of String, or Integer data type. This information is obtained from the OPC UA Server.
Namespace URI	Displays the Namespace URI imported from the OPC UA Server.  This information cannot be edited.

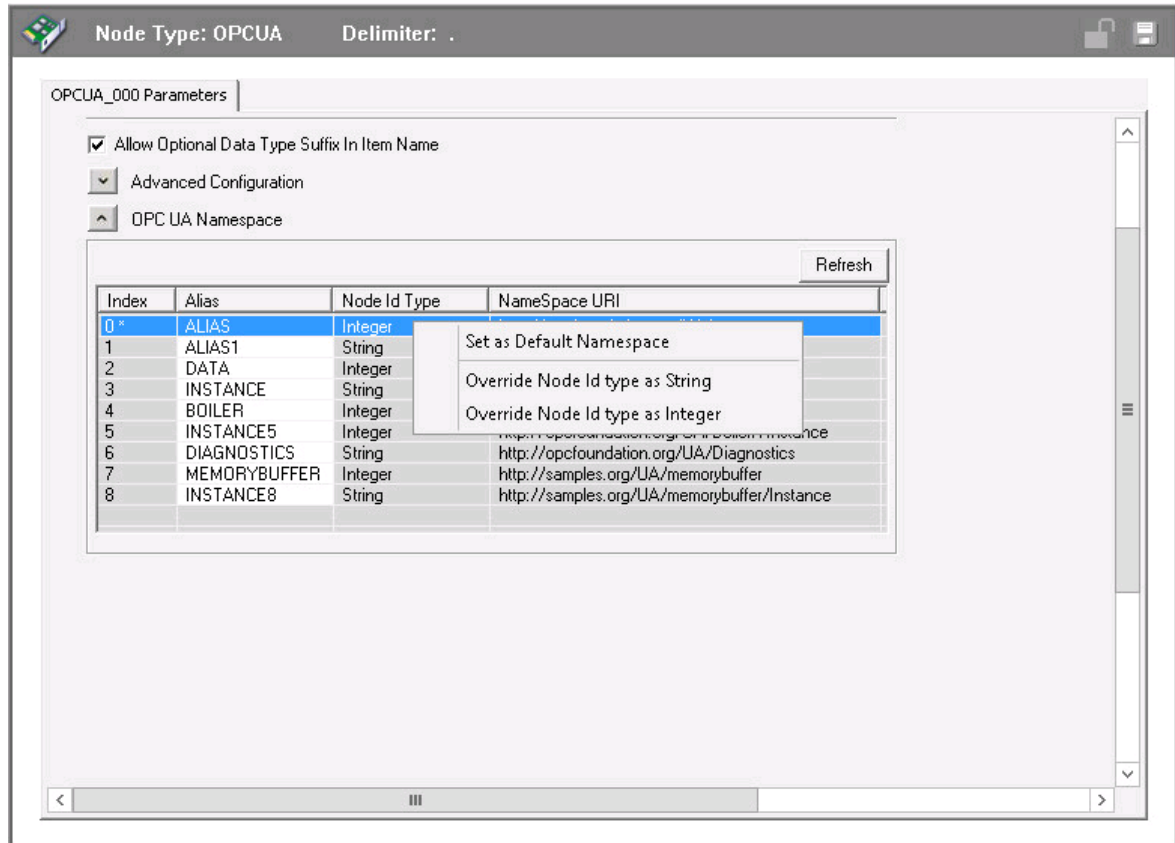
Use the context menu in the Namespace Alias table to edit:

- Default namespace

- Node ID type (override)

### To edit the namespace

1. Right-click the Namespace URI you want to set as the default, and then select **Set as Default Namespace**. The Index will then be appended with an asterisk (\*) to indicate the default Namespace URI.
2. In the **Alias** text box of the selected Namespace, type an alias name for the Namespace URI.
3. Right-click the **Node Id Type** you want to edit, and then select **Override Node Id type as String** or **Override Node Id type as Integer**. The data type of that Node Id is changed accordingly



4. Save the OPC UA configuration.

## Configuring an OPC UA Group Object

### To add a group object to your OPC UA data source hierarchy

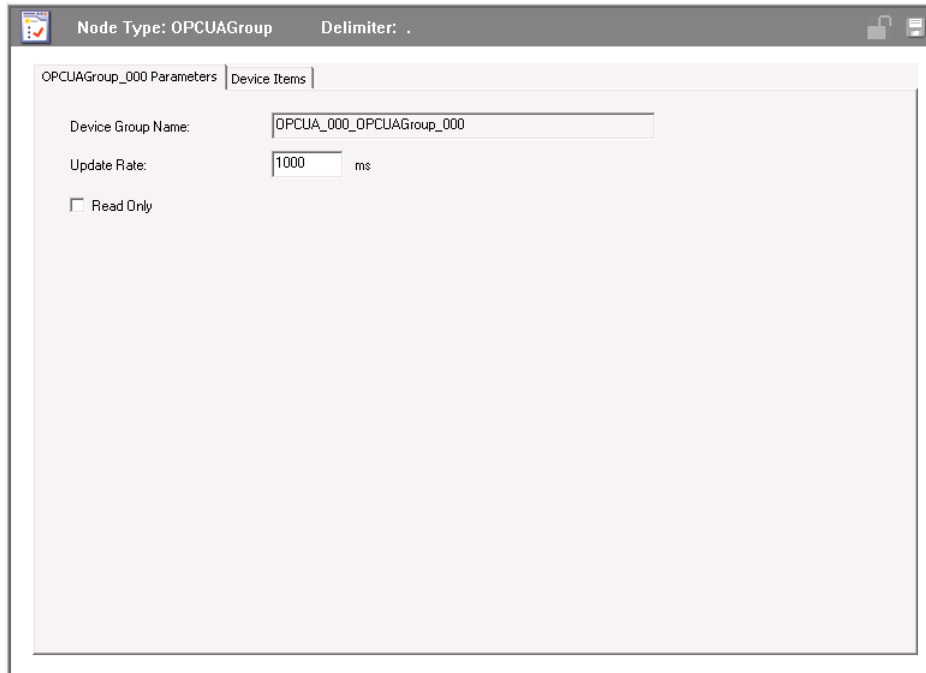
1. Select the new data source object, right-click it, and then click **Add OPC UAGroup Object** on the shortcut menu.

A new object is created in the hierarchy tree and is named **New OPCUAGroup\_000** by default. Rename it, if desired. You are allowed to add up to 100 new group objects.

The **New OPCUAGroup\_000 Parameters** configuration view (right pane) is displayed.

2. Configure the new group object parameters.
  - a. **Device Group Name:** This parameter is configured on the hierarchy tree, and is displayed in a non-editable text box in the parameters configuration pane.

- b. **Update Rate:** Expressed in milliseconds, this parameter controls the fastest rate at which updates are sent from the Gateway to the client. The default value is 1000 ms. The minimum value is 0 ms. The maximum value is 2147483646 ms. This parameter is hot configurable.
- c. **Read Only:** When selected, all items connected through the OPC UA group are read only. This parameter is hot configurable.



3. (Optional) Configure Device Items. Click the **Device Items** tab and add item names and references. For more information, see *Configuring Device Item Definitions*.

## Configuring OPC UA Device Items

You can add items directly to the OPC UA data source branch or in a group that allows you to group related OPC UA tagnames together.

To add device items to your group, select the new group object and click the **Device Items** tab. For more information, see *Configuring Device Item Definitions*.

## OPC UA Item Names and Syntax

### OPC UA Item Syntax

You can access data for OPC UA server items in the default namespace, with a complete namespace URI, and with a namespace alias. Each context requires its own syntax:

Namespace	Item Syntax and Description	Example
Default Namespace	UAItemReference or //UAItemReference <ul style="list-style-type: none"> <li>UAItemReference is the identifier of the item (complete path of the OPC UA Server item, followed by the NodeId) in OPC UA Server.</li> <li>The UA Item Reference is resolved by the OPC UA Server, and the case is preserved. This is case sensitive</li> </ul>	PLCport.PICobject.N40:0

Namespace URI	<p>UANS(UANamespaceURI),UAItemRef(UAItemReference)</p> <ul style="list-style-type: none"> <li>• UANS stands for UANamespace, and is a keyword.</li> <li>• UANamespaceURI is the true namespace URI in OPC UA Server.This is case sensitive.</li> <li>• UAItemRef stands for UAItemReference, and is a keyword.</li> <li>• UAItemReference is the identifier of the item (the complete path of the OPCUA Server item, followed by the NodeId) in the OPC UA Server. UA Item Reference is resolved by the OPC UA Server, and the case is preserved. This is case sensitive.</li> </ul>	UANS(http://test.org/UA/Data)/UAItemRef(10847)
Namespace Alias	<p>/UANamespaceAlias/UAItemReference or UANS(UANamespaceAlias)UAItemRef(UAItemReference)</p> <ul style="list-style-type: none"> <li>• UANS stands for UANamespace, and is a keyword.</li> <li>• UANamespaceAlias is the Namespace alias name used to identify the namespace in the OPC UA Server. This is configured in the OPC UA editor. UANamespaceAlias is replaced by the true namespace URI in UA Client, before it is sent to the server. This is case insensitive</li> <li>• UAItemRef stands for UAItemReference, and is a keyword.</li> <li>• UAItemReference is the identifier of the item (the complete path of the OPC UA Server item, followed by the NodeId) in the OPC UA Server. UA Item Reference is resolved by the OPC UA Server, and the case is preserved. This is case sensitive.</li> </ul>	UANS(Data)UAItemRef(10847)

## Using the VT Item Suffix

You can configure OPC UA data sources to use the /VT item suffix to gain better control of data access in OI Gateway. Using the /VT item suffix enables you to tell the underlying OPC UA server the type of data you want the server to report to OI Gateway.

To avoid delayed item validation, apply the /VT item suffix to an item name using the following syntax: `<item name> /VT_<DataType>`. For example, `item1 /VT_I4`.

### VT Suffix Syntax Example

If you want to specify a 2-byte signed Integer (/VT\_I2) as the data type for an OPC UA server path of `<OPCUAServer>.Blower.Int2`, the item name in OI Gateway would be subscribed as `<OPCUAServer>.Blower.Int2 /VT_I2`.

For information about specific VT data types, see *OPC UA Data Conversion* on page 61.

## OPC UA Data Conversion

The following maps the supported OPC UA data types to OPC data types and to DDE/SuiteLink data types.

OPC UA Type	Suite Link Type	OPC Variant Type	Comments
Boolean	Discrete	VT_BOOL	
Sbyte	Integer	VT_I1	
Byte	Integer	VT_UI1	
Int16	Integer	VT_I2	
UInt16	Integer	VT_UI2	
Int32	Integer	VT_I4/VT_INT	
UInt32	Integer	VT_UI4/VT_UINT	
Int64	Integer	VT_I8	
UInt64	Integer	VT_UI8	
Float	Real	VT_R4	
Double	Real	VT_R8	
String	String	VT_BSTR	
DateTime	Integer	VT_UI8	
GUID	String	VT_BSTR	
ByteString	String	VT_BSTR	
XMLElement	String	VT_BSTR	
NodeId	String	VT_BSTR	
Expanded NodeId	String	VT_BSTR	
Qualified Name	String	VT_BSTR	
Localized Text	String	VT_BSTR	
Status Code	String	VT_BSTR	
Variant	String	Variant	Varies based on the value
Enumeration	String	VT_I4	
Structure	String	VT_BSTR	
Number	String	Numeric variant	Can be any of the valid numeric variants
Integer	String	VT_I1/VT_I2/VT_I4/VT_I8	Type changes on the values
UInteger	String	VT_UI1/VT_UI2/VT_UI4/V T_UI8	Type changes on the values
BooleanArray	Not Supported	VT_ARRAY   VT_BOOL	

SbyteArray	Not Supported	VT_ARRAY   VT_I1	
ByteArray	Not Supported	VT_ARRAY   VT_UI1	
Int16Array	Not Supported	VT_ARRAY   VT_I2	
UInt16Array	Not Supported	VT_ARRAY   VT_UI2	
Int32Array	Not Supported	VT_ARRAY   VT_INT	
UInt32Array	Not Supported	VT_ARRAY   VT_UINT	
Int64Array	Not Supported	VT_ARRAY   VT_I8	
UInt64Array	Not Supported	VT_ARRAY   VT_UI8	
FloatArray	Not Supported	VT_ARRAY   VT_R4	
DoubleArray	Not Supported	VT_ARRAY   VT_R8	
StringArray	Not Supported	VT_ARRAY   VT_BSTR	
DateTimeArray	Not Supported	VT_ARRAY   VT_UI8	
GUIDArray	Not Supported	VT_ARRAY   VT_BSTR	
ByteStringArray	Not Supported	VT_ARRAY   VT_BSTR	
XMLElementArray	Not Supported	Not Supported	
NodeIdArray	Not Supported	Not Supported	
Expanded NodeIdArray	Not Supported	Not Supported	
Qualified NameArray	Not Supported	Not Supported	
Localized TextArray	Not Supported	VT_ARRAY   VT_BSTR	
Status CodeArray	Not Supported	Not Supported	
VariantArray	Not Supported	Not Supported	
EnumerationArray	Not Supported	VT_ARRAY   VT_I4	
StructureArray	Not Supported	Not Supported	
NumberArray	Not Supported	Not Supported	
IntegerArray	Not Supported	Not Supported	
UIntegerArray	Not Supported	Not Supported	

---

**Note:** VT\_CY syntax is not supported for an OPC UA data source.

---

# CHAPTER 4

## Device Groups and Device Items

### In This Chapter

Configuring Device Item Definitions .....	63
Exporting and Importing the Gateway Item Data .....	64

### Configuring Device Item Definitions

The **Device Items** tab in a data source's topic or group (also on the data source tier for InTouch and ArchestrA) is used to define aliases to actual data source items. The **Device Items** configuration view is the place where the following activities are performed:

- Creating new device item definitions for data source items.
- Modifying existing device items.
- Deleting existing device items.
- Exporting the list of device items to a comma-separated values (.csv) file. A .csv file can be opened with Microsoft Excel or any text editor.
- Importing device items from a .csv file into the **Device Items** tab.

Each device item definition should contain a unique name for the data source associated with it.

For detailed information about item naming conventions in OI Gateway, see the Item Names sections for the respective data sources as well as *System Items*.

### Device Item Definitions

The **Device Items** configuration view is used to add, clear all, rename, delete, import and export device items.

The **Device Items** configuration view has the following two columns:

- **Name:** This column defines the alias names to actual data source items.
- **Item Reference:** The actual data source item names defined in this column.

---

**Note:** When you create or add a new device item, a unique name needs to be entered for it.

---

#### To create or add device items

1. Right-click anywhere in the **Device Items** configuration view.
2. Select the **Add** command from the shortcut menu.
  - A device item is created, and it is numerically named by default. For example, Item\_0, Item\_1, and so on.
3. Change the default name by double-clicking on it and entering the new name.
  - Enter a unique name for the new device item.

### To add item references

Item references for each of the device items that have been created can be added as follows:

1. In the **Item Reference** column, double-click on the area in the same horizontal line as the selected device item.
2. Type in the actual data source item name in the frame that appears.
3. Click anywhere in the configuration view or press the **Enter** key to have the change take effect.

### To rename a device item from the list

1. Right-click on the device item to be renamed.
2. Select the **Rename** command from the shortcut menu and enter the new device item name.
3. Click anywhere in the configuration view or press the **Enter** key to apply the change.

### To delete a device item from the list

1. Right-click on the device item to be deleted.
2. Select the **Delete** command from the shortcut menu.
  - The device item and its corresponding data source item name are deleted from the configuration view.

---

**Note:** When you select another part of the OI Gateway tree hierarchy, you are prompted to save the modifications to the configuration set.

---

### To clear all device items

1. Right-click anywhere in the **Device Items** configuration view.
2. Select the **Clear All** command from the shortcut menu.
  - All the device items listed in the configuration view, including their corresponding data source item names, are deleted.

## Exporting and Importing the Gateway Item Data

The **Export** and **Import** commands on the shortcut menu enable you to export and import the OI Gateway item data to and from a CSV file, after the configuration of the Device Items has been completed. These commands will allow you to perform an off-line, large-scale edit on the item data configured for a controller, and import what has been edited back into the controller configuration.

The **Export** and **Import** features on the shortcut menu of the **Device Items** dialog box enable you to export and import the OI Gateway device item data to and from a CSV file, after the configuration of the Device Items has been completed. These features provide you with the following capabilities:

- Archive lists of device items.
- Bring an archived list of device items into the **Device Items** dialog box when you need to utilize or reconfigure any of the device items on the archived list.
- Perform an off-line, large-scale edit on the item data configured for a PLC.
  - Import what has been edited back into the PLC configuration.

### To export device items

When you want to archive a list of device items, use the **Export** feature in the **Device Items** configuration view.

1. Right-click anywhere in the **Device Items** configuration view.
2. Select the **Export** command from the shortcut menu.



3. Select the folder into which the list is to be saved.
4. Name the list to be exported.
5. Click the **Save** button.
  - The whole list is saved as a .csv file.

### **To import device items**

The **Import** feature in the **Device Items** configuration view is used to import an archived list of device items into the configuration view.

1. Right-click anywhere in the **Device Items** configuration view.
2. Select the **Import** command from the shortcut menu.
3. Select the archived list (.csv file) to be imported.
4. Click the **Open** button.
  - The whole list is imported into the Device Items configuration view.

---

**Note:** Duplicate items with the same Item References are ignored during import. Duplicate items with different Item References cause a dialog box to be displayed, in which you must make a selection.

**Important:** OI Gateway resolves item names from its clients at runtime in the following order:

1. System items (those prefixed with \$SYS\$)
  2. Device items (those defined in the **Device Items** configuration view)
  3. All other items (validated directly from the PLC device)
-



# CHAPTER 5

## Managing the Gateway

### In This Chapter

Gateway Post-Configuration Steps .....	67
Configuring the Gateway as Service .....	67
Configuring the Gateway as Not a Service .....	68
Archiving Configuration Sets .....	68
Activating/Deactivating the Gateway .....	68
In-Proc/Out-of-Proc .....	69
Hot Configuration .....	69

## Gateway Post-Configuration Steps

After you configure the Gateway, there are two steps to take before you can access data with your client application.

1. Determine what kind of client applications are to be used with this Gateway.

If any of your client applications use DDE/SuiteLink, you must configure the Gateway as a service. If only OPC client applications will be used, you can configure the Gateway as a service or as not a service.

2. Activate OI Gateway.

Some client applications can programmatically activate the Gateway.

- If you configure OI Gateway as an automatic service, OI Gateway is started and activated when the computer on which the OI Server is installed starts up.
- If you configure the OI Server as a manual service, the OI Server is not started when the computer starts up. Instead, it is started up on the first connection from an OPC client or when activated from the OI Server Manager.

After an OI Server is running as an auto or manual service, it stays running until explicitly stopped in the OI Server Manager or the computer shuts down.

## Configuring the Gateway as Service

To support DDE/SuiteLink clients, the OI Gateway must be configured as a service.

### To configure the OI Gateway as a service

1. In the Operations Integration Server Manager, navigate to the OI Gateway.
  - a. Expand **Operations Integration Server Manager**, expand **Default Group**, and then expand **Local** or the remote computer name.
  - b. Expand **Wonderware - Gateway**.

2. Right-click OI.Gateway.1 and then click **Configure As Service**.
3. Click either **Auto Service** or **Manual Service**.
4. Read the warning message and click **Yes**.

## Configuring the Gateway as Not a Service

The OI Gateway can only be set to run as not a service when the OI Server is in the deactivated state.

### To configure the OI Gateway as not a service

1. In the Operations Integration Server Manager, navigate to the OI Gateway.
  - a. Expand **Operations Integration Server Manager**, expand **Default Group**, and then expand **Local** or the remote computer name.
  - b. Expand **Wonderware - Gateway**.
2. Right-click OI.Gateway.1 and then click **Configure As Service**.
3. Click **Not a Service**.
4. Read the warning message and click **Yes**.

## Archiving Configuration Sets

After your OI Gateway has been configured, you can archive that specific configuration. You can archive more than one configuration set, and subsequently choose different configurations for different purposes.

### To archive configuration sets

1. In the OI Server Manager, right-click on the Configuration node in the hierarchy below your OI Gateway.
2. Select **Archive Configuration Set** from the shortcut menu.
3. In the **Archive Configuration Set** configuration view, provide a Configuration Set Name.
4. Click **Archive**.
  - o All current configuration values are saved to the archived set.

Once you have archived at least one configuration set, you can select it for use.

### To use different configuration sets from the current one

1. Make sure OI Gateway is not running.
2. In the OI Server Manager, right-click the **Configuration** node in the hierarchy below OI Gateway.
3. Select **Use Another Configuration Set** from the shortcut menu and click on a configuration set in the sub-menu.

All parameters in OI Gateway configuration hierarchy change to the chosen configuration set.

## Activating/Deactivating the Gateway

When you activate the OI Gateway, it starts communicating and accepting requests from client applications. If a OI Server is configured as an automatic service, the OI Server is started and activated when the computer starts up. Also, a OI Server can be activated by the an OPC client connection request, but only out-of-proc..

**To activate the OI Gateway**

1. In the Operations Integration Server Manager, navigate to the OI Gateway.
  - a. Expand **Operations Integration Server Manager**, expand **Default Group**, and then expand **Local** or the remote computer name.
  - b. Expand **Wonderware - Gateway**.
2. Right-click OI.Gateway.1 and then click **Activate Server**.

Deactivating your OI Gateway stops it from communicating with client applications.

A OI Server with active OPC clients does not stop until the last OPC client shuts down.

**To deactivate the OI Gateway**

1. In the Operations Integration Server Manager, navigate to the OI Gateway.
  - a. Expand **Operations Integration Server Manager**, expand **Default Group**, and then expand **Local** or the remote computer name.
  - b. Expand **Wonderware - Gateway**.
2. Right-click OI.Gateway.1 and then click **Deactivate Server**.
3. Read the warning message and click **Yes**.

## In-Proc/Out-of-Proc

The OI Server is triggered. Explicitly starting as part of the client process (in-proc) is not currently OI Server supported. Activation using the CLSCTX\_ACTIVATE \_64\_BIT\_SERVER flag is also not supported.

---

Note: The exception to this behavior is OPC UA, which is a service and runs in-proc.

---

When the OI Server is running out-of-proc, it supports requests from both DDE/SuiteLink and OPC client applications.

If the OI Server is running as a service, the icon on the OI Server node in the SMC is yellow. If the OI Server is not running as a service, the icon is white. For more information, see the *Wonderware Operations Integration- Supervisory Server Manager Help*.

## Hot Configuration

OI Gateway is mostly hot-configurable. For instance, you can do the following while the gateway is activated:

- Modify Global Parameters
- Add, delete, or modify data source nodes
- Add, delete, or modify device groups or topics
- Add, delete, or modify device items
- Modify data source and group/topic configuration

ArchestrA user login data is not hot-configurable. OI Gateway must be restarted for the new values to take effect.



# CHAPTER 6

## Accessing the Data in the Gateway

### In This Chapter

About Accessing Data in the Gateway.....	71
Accessing Data Using OPC .....	71
Accessing Data Using DDE/SuiteLink.....	72
Accessing Data Using MQTT .....	72
Accessing Data Using OPC UA .....	72

### About Accessing Data in the Gateway

Client applications read and write to data items that are internal to the OI Server, as well as to the items located in the devices. Client application communication with the OI Server is done using either the OPC or DDE/SuiteLink protocols. The client application may or may not be on the same computer as the OI Server.

You do not need to create device items in the OI Server for your OPC client application.

### Accessing Data Using OPC

In the case of OPC communications, the protocol addresses an element of data in a conversation with six characteristics: node name, program name, group name, device group, link name, and item name.

- The node name (required for remote access) and device group are optional.
- A fully qualified OPC Item name (ItemID) is composed of the link name and item name.
- All other characteristics are specified through separate OI Gateway means.

To access an OPC item, the OPC client needs to connect to OI Gateway (only out-of-process) and create an OPC group defining the data-acquisition properties for the collection of items to be added. Although OPC groups can be either public or private, OI Gateway only supports private groups. Public OPC groups are shared across multiple clients, whereas private OPC groups are local to a single client. Optionally, a device group, which indicates the access path to the items for read/write, can be specified from OI Gateway.

The following briefly describes each characteristic of OPC data access:

- **node name:** Computer (host) name identifying a specific node on the network (for Remote Access ONLY).
- **program name:** The registered OPC server name uniquely identifying a specific server (ProgID). For OI Gateway, the program name is ArcestrA.FSGateway.3.
- **group name:** The OPC group created from the client for organizing a collection of items logically with the same data acquisition properties between the client and the server, such as update rate.

- **device group:** Meaningful names configured in OI Gateway under a specific data source for the common custom attributes between OI Gateway and the source, such as update interval. If not specified from the client, the default device group using the global configuration attribute values from OI Gateway is assumed. Functionally, a device group is equivalent to an access path (optional).
- **link name:** The set of hierarchy node names, representing the specific data source on a communications path link from the hierarchy root to a specific source as configured for OI Gateway under the OI Server Manager, separated by delimiters.
- **item name:** A specific data element, the leaf of the hierarchy tree of OI Gateway, within the specified group.
- **item suffix:** Optionally configure OPC data sources to use the /VT item suffix to tell the underlying OPC server the type of data you want the underlying OPC server to report to OI Gateway. For more information, see *Connecting to an OPC Data Source* on page 28.

## Accessing Data Using DDE/SuiteLink

In the case of DDE/SuiteLink communications, the protocol addresses an element of data in a conversation that uses a four-part naming convention. That convention includes the node name, application name, topic name, and item name. The fully qualified DDE/SuiteLink naming convention includes all four parts, although the node name part (required for remote access only and only for SuiteLink) is optional. The following briefly describes each portion of this naming convention:

- **node name:** Computer (host) name identifying a specific node on the network (for remote access only).
- **application name:** In the case of data going to clients via the DDE/SuiteLink PlugIn of OI Gateway, the application name portion of the address is Gateway.
- **topic name:** Meaningful names are configured in OI Gateway to identify specific data sources. These names are then used as the topic names in all conversations with that source. Topic name maps to a device group defined in OI Gateway.

---

**Note:** You can define multiple device-group (topic) names for the same data source to poll different data at different rates.

---

- **item name:** A specific data element within the specified topic.

## Accessing Data Using MQTT

Once you have the MQTT Data Source object and Group object, you can access data from various MQTT sources that are connected to the internet, for example, connecting to a mosquitto broker.

Payload data is expected to be in string format. If the data is formatted as a JSON key value pair, the values from the JSON string can be extracted as an attribute. See *MQTT Topic Names* on page 48 for additional information.

## Accessing Data Using OPC UA

OPC UA provides a great deal of scalability and flexibility in data access. For example:

1. An OPC client accessing an item in an OPC UA data source:
  - a. Connection: `OI.Gateway.1`
  - b. Item Reference: `OPCUA1.Topic1.Device1.TankLevel`
2. A SuiteLink client accessing an item in an OPC UA data source:
  - a. Connection:



- i. **Application:** OI.Gateway
- ii. **Topic:** OPCUA1\_Topic1
- b. **Item Reference :** Device1.TankLevel



# CHAPTER 7

## The Gateway Features

### In This Chapter

Data Source Redundancy .....	75
Instantiating the Gateway.....	76
Licensing for Gateway Features .....	77

### Data Source Redundancy

The OI Server Manager provides the ability to assign redundant data sources for fail-over protection in the event of data source failure. Two identical data sources are expected to be configured in the OI Server Manager having identical item syntax, connected to the same OI Gateway. Configuring the primary device and secondary device parameters of a Redundant Device Object (RDO) can meet this expectation.

---

**Note:** Items can not be added for updates through the Redundant Device Object if the items do not exist in both controllers.

---

If the Primary data source fails, the OI Gateway will automatically switch over to the Secondary data source. The Secondary data source then becomes the active data source with the failed Primary data source reverting to the backup role. If the failed data source recovers to good status, it will remain in the standby mode.

---

**Note:** Redundant Device Object does not support the ArchestrA object. There can only be one ArchestrA object per hierarchy.

---

### Runtime Behavior

The OI Gateway will start with the active data source. The OI Server Engine will switch to the standby data source when the active data source fails to communicate. The value of the `$$SYS$Status` will determine the communication failure.

---

**Note:** The value of the `$$SYS$Status` of the standby data source must be TRUE in order to switch over to the standby data source. Otherwise, there will not be any failover.

---

When `$$SYS$Status` shows a FALSE value at both active and standby data sources, the OI Server Engine will consider a complete communication failure and mark all the items subscribed to the redundancy data source hierarchy with the current time and the appropriate OPC quality. The OI Server Engine will activate the slow-poll mechanism to retry the communication to both data sources until either one of the Ping Items returns to a good quality and update its `$$SYS$Status` item to TRUE.

When the OI Server Engine switches to the standby data source, the standby data source becomes active and the originally active data source becomes the standby.

When the active data source becomes the standby data source the Ping Item will not be deleted from that the standby data source. This will ensure the standby will be able to recover the communication again.

---

**Note:** The Ping Item must be a valid item from the controller that has not been rejected by the server for the failover to function properly.

---

The OI Gateway will log any failover activities. All other functionality such as diagnostics, enable/disable, and reset will be performed exactly the same as it is performed for any other hierarchy node.

---

**Note:** Unsolicited message configuration is not supported in the Redundant Device Object (RDO) itself. You can still receive unsolicited messages directly from data source groups defined in the regular server hierarchy.

---

This feature allows the OI Gateway to provide fail over support by providing one node which switches between two other nodes. The Redundant Device Object is configured with a redundancy node which directs itself to one of the two nodes and switches to the other based on lack of communications to a common user-configured controller item. In this manner the Redundant Device Object can be used to direct client requests to the redundant node, which switches between data source or communication pathway failure without intervention.

## Instantiating the Gateway

Newer OI Servers that have been optimized for Wonderware Operations Integration support multiple server instances, which means that you create additional server instances and even clone existing server instances in those server groups.

The same behavior applies to the OI Gateway. You can create a new instance of the Wonderware Gateway, and you can clone an existing instance of OI Gateway.

---

**Note:** Creating multiple instances of the Gateway or OI Servers requires a Professional level license. For more information, see *Licensing for Gateway Features* on page 77.

---

### To create a new instance of the Wonderware Gateway

1. In the OI Server Manager, navigate to the Gateway:
  - a. Expand Operations Integration Server Manager, expand Default Group, and then expand Local.
  - b. Expand Operations Integration Supervisory Servers.
2. Right-click the Wonderware - Gateway group, and then click **Create Server Instance** on the shortcut menu. The Create OI Server Instance dialog box appears.
3. In the **Starting Instance** and **Ending Instance** boxes, type the starting and ending numbers for sequentially numbered instances. For example, if Starting Instance is 0001 and Ending Instance is 0005, five instances numbered from 0001 to 0005 will be created.
4. In the **Custom Name** box, type a custom name for the new instances. This is optional, but if you do type something, it will be combined with the instance number to form the actual suffix. For example, if Custom Name is A and the instances are numbered from 0001 to 0005, the actual suffixes will be from A0001 to A0005.
5. Click OK. The new instances are created.

### To clone an existing OI Gateway instance

1. In the OI Server Manager, navigate to the existing OI Gateway:
  - a. Expand the Operations Integration Server Manager, expand Default Group, and then expand Local.
  - b. Expand Operations Integration Supervisory Servers.

- c. Expand Wonderware - Gateway.
2. Right-click the Gateway instance, and then click Clone Instance on the shortcut menu. The **Clone OI Server Instance** dialog box appears.
3. In the **Custom Instance Name** box, type a custom name for the new instance. This is optional.
4. Click OK. The new cloned instance is created.

## Licensing for Gateway Features

Licensing is applicable for the entire Gateway and is not based on individual data sources in the Gateway hierarchy.

1. For standard functionality, OI Gateway requires any one of the following:
  - An InTouch HMI license of any kind, such as InTouch, InTouch for System Platform, and InTouch TSE, should be installed on the node.
  - A standard OI Gateway license (OI Server with count = 1).
2. For professional functionality, OI Gateway requires a professional OI Gateway license (OI Server with count =2).

The system item **\$\$SYS\$Licensed** indicates the status of the license. A return of 1 indicates that it is licensed and 0 indicates that license is not present.

---

**Note:** MQTT is limited to a single connection configuration. You can connect to multiple brokers by leveraging the professional feature which allows multi-instance configuration. Hence, each instance can have its own broker configured.

---

### Demo Mode

You can install the OI Gateway without a license. The OI Gateway runs without a license in Demo mode for 120 minutes. When the 120 minutes expire:

- The OI Server stops updating items.
- All non-system items have a Bad quality status.
- New items are rejected.

While in demo mode the OI Gateway checks for a license every 30 seconds. If a license is not found, the OI Server logs a warning.

Use the **\$\$SYS\$Licensed** system item to check the status of your license. This item returns true if the proper license is found or the OI Server is in demo mode (the 120 minutes). Otherwise, it returns false.

After the OI Server finds a valid license, it logs a message, stops looking for a license, and begins running normally. For more information, see the *License Utility Documentation*.



# CHAPTER 8

## System Items

### In This Chapter

Standard System Items.....	79
OPC UA Diagnostic Items.....	86
Data Quality.....	92

### Standard System Items

System items provide you with easy access to OI Gateway's status and diagnostics information. They are treated just like ordinary items with respect to the client. However, in most cases these items are not directly acquired via the communications layer. System item values are usually generated through internal calculations, measurements, and the tracking of the OI Server Engine.

System items, like ordinary items, are defined by the following properties:

- **Group** (client group/OPC group): The arbitrary collection of items, not correlated.
- **Hierarchical location** (link name/OPC path, the hierarchical node section of the fully qualified OPC item ID): The device the item is attached to.
- **Device group** (OPC access path/topic, or a Scan Group on a hierarchical branch): A collection of items on the same physical location with the same protocol update rate.

---

**Note:** For DDE/SuiteLink clients, `$$SYS$Status` always comes from the leaf level of the gateway hierarchy branch, which is the destination data source. For OPC clients, `$$SYS$Status` can be accessed at all hierarchy levels. `$$SYS$Status` at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. Hence, for practical application, OPC clients should reference `$$SYS$Status` at any hierarchy levels other than the root. In the case of an ArcestrA data source, `$$SYS$Status` is always good, even at the ArcestrA Group level.

---

In the ArcestrA context, the device group plays the most important role of identifying the scope of any item. The device group defines the hierarchical location implicitly when using globally unique device-group names, which is required for DDE/SuiteLink compatibility.

All system items follow the same naming convention:

- All system items start with `$$SYS$`.
- The OI Server Engine scans and parses the name for system items. Parsing of the name is case-insensitive.

All system items can be accessed through subscriptions to a Device Group. However, while some system items return data for that Device Group, others are gateway-wide.

### Global System Item

The following system item refers to specific information regarding a global condition of the gateway.

System Item Name	Type/ Access Rights	Description	Values
\$SYS\$Licensed	Boolean/ Read	<p>Indicates whether a valid license is present for the OI Server.</p> <ul style="list-style-type: none"> <li>If FALSE, this item causes the OI Server to stop updating existing tags, to refuse activation addition of new tags, and to reject write requests in addition to setting quality for all items to BAD.</li> <li>If TRUE, the OI Server functions as configured.</li> </ul> <p>All instances have the same value.</p>	<p>RANGE: 0, 1</p> <p>0: No valid license exists.</p> <p>1: Valid license exists.</p>

## Device-Specific System Items

The following system items refer to specific information regarding the data source(s) OI Gateway is connected to.

System Item Name	Type/ Access Rights	Description	Values
\$SYS\$Status	Boolean/ Read	<p>Binary status indication of the connection state to the device (hierarchy level) the item is attached to. The device group (OPC access path/topic) does not affect the value.</p> <p>The status can be good even if individual items have errors.</p> <p>For DDE/SuiteLink clients, \$SYS\$Status always comes from the leaf level of a gateway hierarchy branch, which is the destination data source.</p> <p>For OPC clients, \$SYS\$Status can be accessed at all hierarchy levels. \$SYS\$Status at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. Hence, for practical application, OPC clients should reference \$SYS\$Status at any hierarchy levels other than the root.</p>	<p>RANGE: 0, 1</p> <p>1: Gateway connection to the data source is intact.</p> <p>0: Error communicating with the data source.</p>



System Item Name	Type/ Access Rights	Description	Values
\$SYS\$ErrorCode	Longint/ Read	Detailed error code of the communications state to the data source.  The device group (OPC access path/topic) does not affect the value.	>= 0: Good status (0 is the default state – connected). >0: is some state like: connecting, initializing, etc. <0: Error status (value indicates the error).
\$SYS\$ErrorText	String/ Read	Detailed error string of the communications state of the data source.  The device group (OPC access path/topic) does not affect the value.	Descriptive text for the communications state corresponding to the error code.
\$SYS\$StoreSettings	Integer/ Read Write	Not used.	

---

**Caution:** For all three device-specific system items, status is always good for an ArchestrA data source.

---

## Device Group-Specific System Items

The following system items refer to specific information regarding device groups that have been configured in OI Gateway.

System Item Name	Type/ Access Rights	Description	Values
\$SYS\$updateInterval		Not used.	
\$SYS\$maxInterval		Not used.	

System Item Name	Type/ Access Rights	Description	Values
\$SYS\$WriteComplete	Integer/ ReadWrite	<p>Accesses the state of pending write activities on the corresponding device group. On device group creation (adding items to an OPC group), the value of this system item is initially 1, indicating all write activities are complete – no pokes are pending.</p> <p>If values are poked into any items of the device group, the value of this item changes to 0, indicating write activity is currently in progress.</p> <p>If the server has completed all write activities, the value of this item changes to 1 if all pokes were successful or to -1 if at least one poke has failed.</p> <p>If the value of this item is not zero, the client can poke 1 or -1 to it (poke a 1 to clear errors, or a -1 to test a client reaction on write errors).</p> <p>If the value of this item is zero, it cannot be poked.</p>	<p>RANGE: -1, 0, 1</p> <p>1: Write complete (no writes are pending – initial state).</p> <p>0: Writes are pending.</p> <p>-1: Writes completed with errors.</p>

System Item Name	Type/ Access Rights	Description	Values
\$SYS\$ReadComplete	Integer/ ReadWrite	<p>Accesses the state of initial reads on all items in the corresponding device group.</p> <p>The value is 1 if all active items in a device group have been read at least once.</p> <p>If at least one item in the device group is activated, this item changes to 0. It changes to 1 if all items have been read successfully, or to -1 if at least one item has a non-good quality.</p> <p>Poking a 0 to this item resets the internal read states of all items in this device group. This resets this item to 0. If all items are read again after this poke, this item changes back to 1 or -1.</p>	<p>RANGE: -1, 0, 1</p> <p>1: Read complete (all values have been read).</p> <p>0: Not all values have been read.</p> <p>-1: All values have been read but some have a non-good quality.</p>
\$SYS\$ItemCount	DWord/ Read	<p>Accesses the number of items in the corresponding device group. This item is read-only.</p>	<p>RANGE: 0...2147483647</p> <p>&gt;=0: Number of active items.</p>
\$SYS\$ActiveItemCount	DWord/ Read	<p>Accesses the number of active items in the corresponding device group. This item is read-only.</p>	<p>RANGE: 0...2147483647</p> <p>&gt;=0: Number of active items.</p>
\$SYS\$ErrorCount	DWord/ Read	<p>Accesses the number of all items (active and inactive) that have errors (non-good OPC quality) in the corresponding topic.</p> <p>If the communications status of a device group is bad, all items have errors. This item is read-only.</p>	<p>RANGE: 0...2147483647</p> <p>&gt;=0: Number of all items (active and inactive) with errors.</p>

System Item Name	Type/ Access Rights	Description	Values
\$SYS\$PollNow	Boolean/ ReadWrite	<p>Poking a 1 to this item forces all items in the corresponding device group to be read immediately.</p> <p>This is useful if you want to force to get the newest values from the device, regardless of its update interval.</p> <p>This also works on device groups with a zero update interval (manual protocol triggering).</p>	RANGE: 0, 1,

## The Gateway-Specific System Items

The following system items refer to specific information regarding OI Gateway.

---

**Important:** OI Gateway-specific systems items are available only at the following hierarchy levels: ArchestrA data source, OPC groups, DDE/SL topics, and InTouch data source.

---

System Item Name	Type/ Access Rights	Description	Values
\$SYS\$Gateway ConnectionStatus	Boolean/ Read-Only	Indicates whether OI Gateway has established a successful connection to the configured data source and topic (if any).	<p>RANGE: True, False</p> <p>True: When connected to the data source. False: When disconnected.</p>
\$SYS\$Gateway ConnectionStatus String	String/ Read-Only	Indicates whether OI Gateway has established a successful connection to the configured data source and topic (if any).	<p>"Connected": When successful connection to data source is achieved. "Disconnected": When otherwise.</p>

System Item Name	Type/ Access Rights	Description	Values
\$SYS\$Reconnect	Boolean/ Read/Write	Triggers a reconnect attempt to the configured data source. If you poke a value of 1 (True), this functionality is exercised even if the maximum number of reconnects is reached. By default, this item reads zero (0, False). Writing False does nothing.	RANGE: True, False  True: Triggers reconnect attempt. If data source is already connected, it is disconnected and then reconnected.  False: Does nothing. Default value.
\$SYS\$OPCUpdate Interval	DWORD/ Read/Write	Indicates the update interval, in milliseconds, of the connected OPC server. This particular system item is available only in the OPC group.	RANGE: 0...2147483647  If 0 is specified, then the fastest update interval supported by the OPC server is used.

## The Gateway Redundant Device-Specific System Items

These system items are specific to the Redundant Device.

System Item Name	Type/ Access Rights	Description	Values
\$SYS\$ForceFailover	Boolean/ ReadWrite	This is required to achieve the failover condition to be forced by client.  Note: By poking a value of "1" (True) into the Force Failover item, a client can conveniently switch to the secondary device.	TRUE, FALSE

System Item Name	Type/ Access Rights	Description	Values
\$SYS\$ActiveDevice	String/Read	This system item will show the current runtime active device.	Node Hierarchy Name
\$SYS\$FailoverTime	Time/Read	This system item will show the time at which the switch occurred.	Time at which the switch occurred
\$SYS\$StandbyDevice	String/Read	This system item will show the current runtime standby device.	Node Hierarchy Name
\$SYS\$Secondary DeviceStatus	Boolean/Read	This system item will show the status of the secondary device. This is the status of the second device defined in the configuration and is not changed with any failover. RANGE: 0, 1	RANGE: 0, 1 (Contains the value of the system item \$SYS\$Status)
\$SYS\$PrimaryDevice Status	Boolean/Read	This system item will show the status of the primary device. This is the status of the first device defined in the configuration and is not changed with any failover. RANGE: 0, 1	RANGE: 0, 1 (Contains the value of the system item \$SYS\$Status)
\$SYS\$FailoverReason	String/Read	This system item will show the reason for the failover.	Descriptive text "ForceFailover" or the value of the system item \$SYS\$errorText.

---

**Important:** The Redundant Hierarchy, including the Device Group, is not hot-configurable, and requires a Reset on the Redundant Hierarchy to effect a configuration change.

---

## OPC UA Diagnostic Items

The following groups of diagnostic system items are available for OPC UA:

- *Global Diagnostic Items* on page 87: diagnostic items for the entire OPC UA hierarchy.
- *Connection Diagnostic Items* on page 89: diagnostic items for a specific connection.
- *Subscription Diagnostic Items* on page 90: diagnostic information for a specific subscription.
- *OPC UA Client Diagnostic Items* on page 91: diagnostic items for the entire OPC UA hierarchy.

The OPC UA diagnostic items follow a specific syntax. See *Diagnostic Item Syntax* on page 87 for syntax information.

## Diagnostic Item Syntax

The diagnostic items are contained in a special namespace named "\$SYS\$DIAG".

The following illustrates the full path to a diagnostic item.

```
$SYS$DIAG/<Object Path>/<Item>
```

Where:

- \$SYS\$DIAG – is a special namespace containing diagnostic items in the service.
- <Object Path> - is substituted with a one or more object qualifiers, each separated by '/'.  
/
- <Item> - is substituted with the name of a specific diagnostic item.

## Global Diagnostic Items

The global object contains diagnostic items for the entire service. The following syntax can be used to access a global diagnostic item:

```
$SYS$DIAG/Global/<Item>
```

Where:

<Item> is one of the diagnostic item names from the following table.

<Item>	Type	R/W	Description
NodeName	String	R	Name of the node this service is running on.
StartDateTime	DateTime	R	Date and Time when the service was started. (Time is in UTC)
UpTime	Int32	R	Elapsed time in seconds since the service was started
ConnectionCount	Int32	R	Number of incoming ASB client connections.
ItemCount	Int32	R	Number of registered items in all connections.
SubscriptionCount	Int32	R	Number of subscriptions in all connections
MonitoredItemCount	Int32	R	Number of monitored items in all subscriptions/connections.

MonitoredItemErrCount	Int32	R	Number of items with bad quality in all subscriptions/connections.
TotalReads	Int32	R	Number of reads operations in all connections since service start or last reset.
TotalReadErrors	Int32	R	Number of read operation errors in all connections since service start or last reset. ( An error is counted if 1 or more item reads failed, or the entire read could not be executed )
TotalItemReads	Int32	R	Number of read items in all connections since service start or last reset. (Does not include subscription updates or items in the diagnostic namespace)
TotalItemReadErrors	Int32	R	Number of read item errors in all connections since service start or last reset. (Does not include subscription updates or items in the diagnostic namespace.)
TotalWrites	Int32	R	Number of writes operations in all connections since service start or last reset
TotalWriteErrors	Int32	R	Number of write operation errors in all connections since service start or last reset. ( An error is counted if 1 or more item writes failed, or the entire write could not be executed )
TotalItemWrites	Int32	R	Number of write items in all connections since service start or last reset. (Does not include items from the diagnostic namespace)
TotalItemWriteErrors	Int32	R	Number of write item errors in all connections since service start or last reset. (Does not include items from the diagnostic namespace)
ConnectionEvents	Int32	R	Number of connection adds or deletes since service start. This counter will rollover to 0 after value reaches Int32_Max.
MaxConnectionIndex	Int32	R	The maximum connection index number within all currently created connections. Indexes are 0 to n. -1 means there currently are no connections. ( i.e. If value is 10, then 0 to 10 inclusive can be used in the system item syntax "conn=x" or "conn=x:y" where x it the connection index)



ResetTotals	Bool	R/W	Reset TotalReads, TotalWrites, TotalItemReads, TotalItemWrites, TotalReadErrors, TotalItemReadErrors, TotalWriteErrors and TotalItemWriteErrors
-------------	------	-----	---

## Connection Diagnostic Items

The connection object contains diagnostic items for a specific connection. The following syntax can be used to access a connection diagnostic item:

```
$SYS$DIAG/Conn=<ConnIndex>/<Item>
```

Where:

- <ConnIndex> is a connection index in the range of 0 to n and always 0 in case of OI Gateway as it supports 1 connection for 1 hierarchy.
- <Item> is one of the diagnostic item names from the following table.

If the <ConnIndex> does not exist, then 0 will be updated for item of type Integer/Float and "" (blank) will be updated for item of type string. Quality of these items will stay as Good.

<Item>	Type	R/W	Description
ItemCount	Int32	R	Number of registered items in this connection.
SubscriptionCount	Int32	R	Number of subscriptions in this connection.
MonitoredItemCount	Int32	R	Number of monitored items in all subscriptions in this connection.
MonitoredItemErrCount	Int32	R	Number of items with bad quality in all subscriptions in this connection.
TotalReads	Int32	R	Number of reads operations in this connection since connection creation or last reset.
TotalReadErrors	Int32	R	Number of read operation errors in this connection since service start or last reset. ( An error is counted if 1 or more item reads failed, or the entire read could not be executed )
TotalItemReads	Int32	R	Number of read items in this connection since connection creation or last reset. (Does not include subscription updates or items in the diagnostic namespace)
TotalItemReadErrors	Int32	R	Number of read item errors in this connection since service start or last reset. (Does not include subscription updates or items in the diagnostic namespace.)
TotalWrites	Int32	R	Number of writes operations in this connection since connection creation or last

			reset
TotalWriteErrors	Int32	R	Number of write operation errors in this connection since service start or last reset. ( An error is counted if 1 or more item writes failed, or the entire write could not be executed )
TotalItemWrites	Int32	R	Number of items written in this connection since connection creation or last reset. (Does not include items from the diagnostic namespace)
TotalItemWriteErrors	Int32	R	Number of write item errors this connection since service start or last reset. (Does not include items from the diagnostic namespace)
SubscriptionEvents	Int32	R	Number of subscription adds or deletes since service start. This counter will rollover to 0 after value reaches Int32_Max.
MaxSubscriptionIndex	Int32	R	The maximum subscription index number within all currently created subscriptions. Indexes are 0 to n. -1 means there currently are no subscriptions. (i.e. If value is 10, then 0 to 10 inclusive can be used in the system item syntax "conn=x:y" where y is the subscription index )
ResetTotals	BOOL	R/W	Reset TotalReads, TotalWrites, TotalItemReads, TotalItemWrites, TotalReadErrors, TotalItemReadErrors , TotalWriteErrors and TotalItemWriteErrors

## Subscription Diagnostic Items

The subscription object contains diagnostic information for a specific subscription. The following syntax can be used to access a subscription diagnostic item

```
$SYS$DIAG/Conn=<ConnIndex>:<SubIndex>/<Item>
```

Where:

- <ConnIndex> is a connection index in the range 0 to n. ConnIndex is always 0 in case of OI Gateway
- <SubIndex> is a subscription index in the range of 0 to n
- <Item> is one of the diagnostic item names from the following table.

If the <ConnIndex> or <SubIndex> does not exist, then 0 will be updated for item of type Integer/Float and "" (blank) will be updated for item of type string. Quality of these items will stay as Good.

<Item>	Type	R/W	Description
Enabled	BOOL	R	Monitored item publishing enabled when true.
PublishInterval	Int64	R	Monitored item publish rate for this subscription in ms.
TotalPublishes	Int32	R	Total number of publishes since subscription creation.
TotalUpdates	Int32	R	Total number of publish values since subscription creation.
ActiveItemCount	Int32	R	Number of active items in this subscription.
LastPublishDuration	Int32	R	Duration in milliseconds of Last Publish in this subscription.  E.g. If MaxNotificationPerPublish is 2000 then it 5 Publish() calls to update 9000 values. In this case LastPublishDuration will be total duration to update 9000 items, after which PublishComplete code is returned to the client.
MaxQueueSize	Int64	R	Max queue size configured in this subscription.
MonitoredItemCount	Int32	R	Number of monitored items this subscription.
MonitoredItemErrCount	Int32	R	Number of monitored items with bad quality in this subscription.
PublishQueueSize	Int32	R	Current number of monitored item values queued and pending to be published to the subscribing client.
PublishQueueOverruns	Int32	R	Number of queue overruns occurrences since subscription creation or last reset. These counter increments by 1 if 1 or more item values are folded at a point in time.
ResetTotals	BOOL	R/W	Reset TotalPublishes, TotalUpdates and PublishQueueOverruns

## OPC UA Client Diagnostic Items

The UA Client object contains diagnostic items for the entire service. The following syntax can be used to access a global diagnostic item:

```
$SYS$DIAG/UAClient/<Item>
```

Where:

<Item> is one of the diagnostic item names from the following table.

<Item>	Type	R/W	Description
IsConnected	BOOL	R	At least one UA Server session established=True, No sessions established=False
TotalConnectionRetries	Int32	R	Number of connection retries since service start. INFO: Item in intended to indicate the number of connection retries between the UA Client and the UA Server.
ServerUri	String	R	UA Server Uri
KeepAliveInterval	Int32	R	Keep Alive interval in ms
SecurityPolicy	Int32	R	0=none, 1=Basic256, 2=Basic128Rsa15
MessagePolicy	Int32	R	0=none, 1=Sign, 2=Sign&Encrypt
NamespaceAliases	String[]	R	Array of namespace aliases.
NamespaceUris	String[]	R	Array of namespace URIs
DefaultNamespace	String	R	Default Namespace configured by the user
IsAnonymousUser	BOOL	R	Anonymous User for UA Server session = True, User credential are provided for UA Server session
UserName	String	R	User Name which is configured for secured session with UA Server (empty if IsAnonymousUser=True)
CertificateExpiry	DateTime	R	Date & Time of application certificate expiry (Time in UTC)
CertificateSubject	String	R	Certificate subject name.
ResetTotals	BOOL	R/W	Reset TotalConnectionRetries

## Data Quality

Data quality is supported in the following protocols:

- ArchestrA Message Exchange
- OPC
- SuiteLink
- FastDDE v3
- OPC UA

Data quality is not supported in the following protocols:

- DDE
- FastDDE v2

For those protocols that support it, quality is consistent with OPC Quality. Therefore, data quality from a source that supports it is passed through OI Gateway unmodified if the client also supports it. In the case of a client that does not support it, the quality is dropped. In the case of a data source that does not support quality, if the client supports it, the quality is fabricated and is always Good (exception: when OI Gateway cannot communicate with the target data source).



# CHAPTER 9

## Troubleshooting

### In This Chapter

Using Troubleshooting Tools .....	95
Monitoring Connectivity Status with a Data Source .....	95
Monitoring the Status of Conversations with DDE/SuiteLink Clients .....	96
Error Messages and Codes .....	97
Communication Failures .....	99

### Using Troubleshooting Tools

This group of help topics describes troubleshooting tools that can be used to deal with Gateway problems you may encounter.

The OI Server Manager provides access to diagnostics and other statistical data, and the Log Viewer provides access to event messages logged during the operation of the Gateway. Also, your client (for example, InTouch) can monitor connectivity with your data source through the \$\$SYS\$Status item. Use these tools together with the information in this section to troubleshoot the Gateway.

---

**Note:** In order to determine the version of your Gateway, perform the following steps. Search for the Gateway DLL (Gateway.dll), right-click on the **File Name**, select **Properties** on the context menu, and select the **Version** tab on the **Properties** dialog box. The version of your Gateway is listed under **File Version**.

---

For more information, see the following help topics.

- *Monitoring Connectivity Status with a Data Source*
- *Monitoring the Status of Conversations with DDE/SuiteLink Clients*
- *Error Messages and Codes*
- *Communication Failures*

### Monitoring Connectivity Status with a Data Source

The built-in discrete item, \$\$SYS\$Status, can be used to monitor the status of communications with your data source. This item is set to the following:

- 0 (zero) when communication with the data source fails.
- 1 (one) when communication is successful.

---

**Note:** For DDE/SuiteLink clients, `$$SYS$$Status` always comes from the leaf level of OI Gateway hierarchy branch, which is the destination data source. For OPC clients, `$$SYS$$Status` can be accessed at all hierarchy levels. `$$SYS$$Status` at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. Hence, for practical application, OPC clients should reference `$$SYS$$Status` at any hierarchy levels other than the root.

---

Enter the following DDE reference formula in the appropriate place in your client:

```
=Gateway|<Device Group>!$$SYS$$Status
```

where:

- Gateway = the name of OI Gateway application.
- <Device Group> = the exact device group defined in OI Gateway for the data source.
- `$$SYS$$Status` = the discrete item used to monitor the status of connectivity with the data source.

**Example:**

```
=Gateway|ModbusOverSL_FastTopic!$$SYS$$Status
```

Enter the following OPC item reference syntax when adding the item in your OPC client:

```
<YourLinkName>.$$SYS$$Status
```

where:

- <YourLinkName> is the assembly of hierarchy node names leading to a specific data source.
- `$$SYS$$Status` is the discrete item used to monitor the status of connectivity with the data source.

**Example:**

```
ModbusOverSL.FastTopic.$$SYS$$Status
```

---

**Note:** In case of a data source disconnection, OI Gateway attempts the number of connection retries as configured for the given data source object, and makes no more attempts afterward. Subsequently, it is up to the client to re-initiate the connection via the system item `$$SYS$$Reconnect`.

---

## Monitoring the Status of Conversations with DDE/SuiteLink Clients

InTouch WindowViewer supports built-in topic names, called **DDEStatus** and **IOStatus**, that can be used to monitor the status of specific OI Server conversations.

For example, assume that WindowViewer (VIEW) is communicating through OI Gateway with a data source with the topic name **Archestra**. The discrete items, **DDEStatus** and **IOStatus**, are set to:

- 0 (zero) when the conversation between OI Gateway and InTouch View fails.
- 1 (one) when the conversation between OI Gateway and InTouch View is successful.

---

**Note:** These items represent the status of communication between the client and OI Gateway.

---

## Using DDEStatus and IOStatus in Excel

The status of communications between OI Gateway and InTouch can be read into Excel by entering the following DDE reference formula in a cell on a spreadsheet:

```
=view|DDEStatus!Archestra
```

or

```
=view|IOStatus!Archestra
```

where:



- **view** = the name of the InTouch application.
- **[DDE][IO] Status** = the built-in topic name used to monitor the status of communications between OI Gateway and InTouch.
- **ArchestrA** = the exact access name defined in OI Gateway for the data source.

## Reading Values from the Gateway into Excel

Values may be read directly into Excel spreadsheets from OI Gateway by entering a DDE formula into a cell using the following format:

```
=applicationname|<devicegroup>!itemname
```

Example formula:

```
=Gateway|ArchestrA!'<tagname>'
```

where:

- Gateway = the name of the OI Gateway application.
- **ArchestrA** = the exact device group name defined in OI Gateway for the data source.
- **<tagname>** = the actual location in the data source that contains the data value. This is the item name.

In this example, each time the value of <tagname> changes in the data source, OI Gateway automatically sends the new value to the cell containing the formula in Excel.

---

**Note:** Refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.

---

## Error Messages and Codes

To troubleshoot OI Gateway problems, use the following error messages together with the OI Server Manager Diagnostics data. Use the Log Flag data to customize the messages logged to the Log Viewer. See the Log Viewer online documentation for more information about using log flags.

OI Gateway processes write requests by receiving them from a client, doing any necessary type conversions, and then forwarding them to the data source. The write request from the gateway to the data source succeeds or fails.

In the case of write success, the gateway informs the client that the write succeeded through write acknowledgement support provided by the client side protocol.

In the case of a write failure, the gateway informs the client that the write failed through the same client side protocol support. In the case of write failure to items on ArchestrA, DDE, SuiteLink and InTouch data sources, OPC\_E\_BADRIGHTS is reported regardless of the failure reason.

## DDE/SuiteLink Client to Any Data Source – Write Errors

In the case of DDE, FastDDE and SuiteLink clients, the write response is a Nak (negative acknowledgement) with no additional failure detail code. When OI Gateway detects a failed write condition, it responds to the client with the Nak.

## OPC Client to ArchestrA – Write Errors

In the case of an OPC Client, the following error code support is used:

Return Code	Description
S_OK	The corresponding item handle was valid. The write will be attempted and the results will be returned on OnWriteComplete.
E_FAIL	The function was unsuccessful.
OPC_E_BADRIGHTS	The item is not writeable.
OPC_E_INVALIDHANDLE	The passed item handle was invalid.
OPC_E_UNKNOWNITEMID	The item is no longer available in the data source's address space.
E_xxx S_xxx	Vendor specific errors may also be returned. Descriptive information for such errors can be obtained from GetLastErrorString.

A failed write to an Archedra data source is handled as follows:

- If Archedra responds with Nak, FS Gateway sends an E\_FAIL error code to the OPC Client.
- If FS Gateway cannot successfully convert the requested OPC data, this maps to a new vendor specific error for OPC indicating "Conversion Error" (OPC\_E\_BADTYPE).
- If the item handle is unknown to FS Gateway or Archedra, the OPC\_E\_INVALIDHANDLE error code is sent.
- If the item name is not valid in FS Gateway or Archedra, the OPC\_E\_UNKNOWNITEMID error code is sent.

## OPC Client to DDE/SuiteLink Data Source – Write Errors

A failed write to a DDE/SuiteLink data source is handled as follows:

- If the data source responds with Nak, OI Gateway sends an E\_FAIL error code to the OPC Client.
- If FS Gateway cannot successfully convert the requested OPC data, this maps to a new vendor specific error for OPC indicating "Conversion Error" (OPC\_E\_BADTYPE).
- If the item handle is unknown to OI Gateway or the data source, the OPC\_E\_INVALIDHANDLE error code is sent.
- If the item name is not valid in OI Gateway or the data source, the OPC\_E\_UNKNOWNITEMID error code is sent.

## Runtime Diagnostics and Error Reporting

For each data source connection, OI Gateway provides a read-only string item to each connected client called:

```
$$SYS$$GatewayConnectionStatusString
```

To each client, this item functions like other items, just under the topic or device group level. It indicates whether the gateway has established a successful connection to the configured data source and topic (if any) as follows:

- "Connected"
- "Disconnected"

Another item, called `$$SYS$GatewayConnectionStatus`, is a Boolean that reads True when connected and False when disconnected. Note in the case of a DDE/SuiteLink data source, the connection is to an application and a topic. Also, in the case of an ArcestrA data source, the connection is to a Platform through Message Exchange. And in the case of an OPC data source, the connection is to an OPC Server through COM/DCOM object creation.

## Communication Failures

OI Gateway behaves in the following manner in the case of failed communication with a data source:

- The gateway attempts to periodically reestablish a connection with the data source up to the maximum number of retry attempts as specified in its Reconnect Attempts parameter.

---

**Note:** The gateway is not responsible for starting the data source server, unless the source protocol supports it. OPC has this capability.

---

- The gateway marks all items being read from the data source with Bad quality. OPC carries a sub-status of Comm Failure.
- Write attempts to the data source are rejected with an appropriate error code.

OI Gateway behaves in the following manner in the case of failed communication with a client:

- The gateway unsubscribes (deactivates) all items on the data source that were previously subscribed to by the failed client. (Exceptions: Those items required by other, still connected, clients remain subscribed. Also, in the case of an OPC client, FS Gateway maintains subscriptions to all items on the data source previously subscribed to by the failed client.)
- The gateway accepts future attempts to reconnect from the client. Reconnection is the responsibility of the client.

---

**Important:** If OI Gateway fails to connect to a remote OPC server through both its ProgID and ClassID, then lower the DCOM **Authentication Level** of the OPC server to None. Do this by opening Control Panel on the remote computer, double-clicking **Administrative Tools**, double-clicking **Component Services**, and then expanding the hierarchy tree under the Console Root as follows: **Component Services, Computers, My Computer and DCOM Config**. Click **DCOM Config**. In the right pane, right-click on the OPC server you cannot connect to, and then click **Properties** on the shortcut menu. On the **General** page of the properties dialog box, select **None** for **Authentication Level**. Click **OK**.

---

Communications failures with an ArcestrA data source behave in the following manner:

Communication Failure	OPC Data Quality
Break between PLC and DIObject	0x1B
DIObject node disconnected	0x00
Node with ArcestrA UserDefined ApplicationObject disconnected	0x04
ApplicationObject undeployed	0x00
WinPlatform undeployed on OI Gateway node	0x00
OI Gateway node disconnected (communicating with local OPC client)	0x04



# CHAPTER 10

## Reference

### In This Chapter

The Gateway Architecture.....	101
Component Environments.....	103
Managing Digital Certificates .....	103

## The Gateway Architecture

OI Gateway is a collection of components that work in concert to provide communications access with a variety of data sources and clients. These components include:

- **OI Server Manager:** This is the Microsoft Management Console (MMC) snap-in, that is part of the ArcestrA System Management Console suite of utilities, supplied with OI Gateway. It provides the necessary user-interface for diagnostics, configuration, and activation.
- **Client Plug-ins:** These are the components that are added to OI Gateway to enable communications with clients. Examples are: OPC, DDE/Suitelink, and so on.
- **OI Server Engine:** This is the library that contains all the common logic to drive data access.
- **Device Protocol:** This is the custom code provided by OI Gateway to define the communications between particular data sources and clients.

---

**Note:** The NetDDE protocol is not supported.

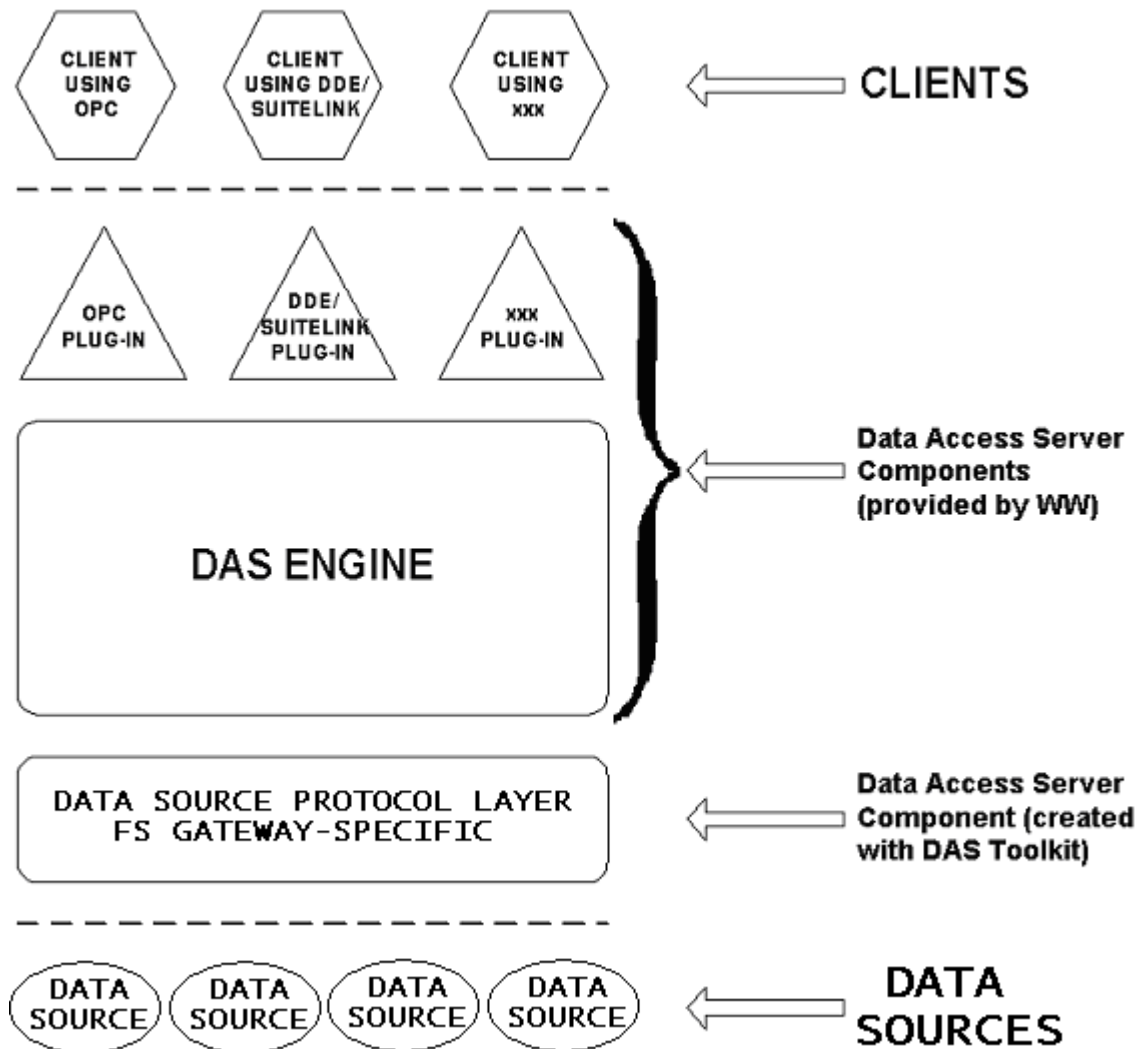
---

## OI Gateway

OI Gateway is comprised of three physical parts (see the following figure). They are the following:

- **Plug-in Component(s):** Responsible for communicating with clients.
- **OI Server Engine:** This common component is used by OI Gateway as well as all Wonderware OI Server.
- **Data Source Protocol Layer, Gateway-specific:** This component is responsible for communicating with the data sources.

The following diagram describes the OI Gateway system architecture.



Each physical part of FS Gateway is comprised of a set of .exe and/or .dll modules. Wonderware provides the Plug-ins and the OI Server Engine. The DAS Toolkit user creates the Data Source Protocol Layer (OI Gateway-specific) modules. All three sets of modules are required for a fully functioning gateway.

## Plug-ins

Plug-ins provide a protocol-translation function for device integration clients. Typical Plug-ins communicate in DDE, SuiteLink, or OPC protocol, and serve as interfaces between their clients and the OI Server Engine.

**Note:** OPC-specific array data type (VT\_ARRAY) is not supported in the DDE/SL plug-in. These arrays are converted to HEX strings, which provide legacy behavior.

## OI Server Engine

The OI Server Engine is a middleware component that exposes two sets of unique interfaces, one for communicating with the Plug-ins and the other one for communicating with the Data Source Protocol Layer components.

## Data Source Protocol Layer

The Data Source Protocol Layer provides a protocol-translation function for specific data sources, such as InTouch, OPC, and ArchestrA; and it serves as an interface between the OI Server Engine and the data sources.

## Component Environments

OI Gateway has the following characteristics:

- The OI Server Engine is dynamically linked to the other OI Gateway components. In other words, a new OI Server Engine (feature enhancement or bug fix) would not require relinking to the other components nor re-QA of those other components. When deployed to the system, the new OI Server Engine would attach to all existing FS Gateway components.
- Newly deployed Plug-ins (feature enhancements or bug fixes) do not require relinking nor re-QA of associated components. Even new Plug-ins (for example, OPC Alarm & Events) would not require any development changes to the other components, and therefore no relinking in a customer-installed base. In fact, it is feasible to implement new functionality in a Plug-in to enhance OI Gateway without any involvement of the code of the other components.
- OI Gateway can be configured in one stand-alone configuration utility (OI Server Manager), and the OI Server Manager is capable of displaying specific configuration views for OI Gateway as well as other Wonderware OI Servers. This utility allows the browsing and editing of Data Access products on different nodes.
- The OI Server Manager diagnostics tool displays generic diagnostic objects common to OI Gateway as well as all OI Server, in addition to OI Gateway-specific/OI Gateway-developer-defined diagnostic data.

OI Gateway's data configuration format is XML. Any XML-enabled program (for example, XML Editor) can read this format.

## Managing Digital Certificates

If you set OPC UA security options to "None", connecting with an OPC UA server will require an OPC UA-compliant digital certificate and key pair.

You can generate and use your own self-signed OPC UA compliant digital certificate and key pair. This digital certificate can be associated to the OPC UA Client service to exchange data with any OPC UA compliant server. You can use a variety of third-party utilities to generate and manage digital certificates, if you are familiar with the OPC UA standard and digital certificate management described in this section.

This section uses a set of sample applications from the OPC Foundation to illustrate how to install, configure and generate OPC UA-compliant digital certificates. A self-signed digital certificate is used in this example. The information and procedures in this section are for illustration purposes, and may vary depending upon your applications and utilities.

---

**Note:** In a secured environment where the digital certificate is required to be signed by a Certificate Authority, please consult your IT department for the procedure.

---

For a complete example and detailed instructions, see the list of related topics.

## Installing a Self-Signed Certificate and Key Pair

### To install self-signed OPC UA compliant certificate/key pair

1. Download the "Sample Applications" file from the OPC UA Foundation.

The download is an archive file: `opc-ua-1.02-.net-sample-applications-setup-336.0-20150630.zip`.

You can download the package from the OPC UA Foundation web site, at:  
<http://www.opcfoundation.org/>

**Note:** You will need to register for an account to download the package.

About ▾
Membership ▾
Products ▾
Certification ▾
Developer Tools ▾
Resources ▾
News & Events ▾

## Unified Architecture

The following Unified Architecture sample code is provided to demonstrate Server and Client implementations, along with many other aspects of UA technology.

Certified Commercial OPC toolkits are the most effective way to maximize productivity while reducing development costs and are available from several OPC members. Search for "toolkits" on the [Products page](#).

[Contact](#) the OPC Foundation for more information or to learn [how to join](#).

Title	Description	Access
<a href="#">Sample Applications</a>	A suite of OPC UA Clients and Servers that demonstrate OPC UA technology and its most popular functionality. <a href="#">All More &gt;</a>	Registered User
<a href="#">.NET Stack and Sample Applications</a>	The .NET Stack contains the core OPC UA stack and many, many reference examples, including: .NET based <a href="#">More &gt;</a>	Corporate Member
<a href="#">Java Stack and Sample Applications</a>	The JAVA Stack contains the core OPC UA stack, and includes: Java based source code which can be opened in <a href="#">More &gt;</a>	Corporate Member
<a href="#">ANSI C Stack Source Code</a>	The ANSI C Stack contains the core OPC UA stack written in portable C code, and includes: C source code and <a href="#">More &gt;</a>	Corporate Member
<a href="#">Local Discovery Server (LDS)</a>	The Local Discovery Server (LDS) provides the necessary infrastructure to publicly expose the OPC UA Servers available <a href="#">More &gt;</a>	Registered User
<a href="#">Local Discovery Server Source Code</a>	The Local Discovery Server (LDS) provides the necessary infrastructure to publicly expose the OPC UA Servers available <a href="#">More &gt;</a>	Corporate Member
<a href="#">Global Discovery Server (GDS) Sample</a>	The Global Discovery Server (GDS) provides the necessary infrastructure to provide enterprise-wide administration of OPC <a href="#">More &gt;</a>	Corporate Member

Showing 1 to 7 of 7 entries

2. Unzip the download archive to extract the "OPC UA Sample Applications 1.02.msi", then right-click the file name, and click **Install** on the context menu.
3. Accept the default settings. The OPC UA sample applications typically install to the following directory:

**C:\Program Files (x86)\OPC Foundation\UA 1.02\Sample Applications**

The sample applications include sample servers, sample clients, and a certificate generator.

## Creating a Self-Signed Certificate and Key Pair

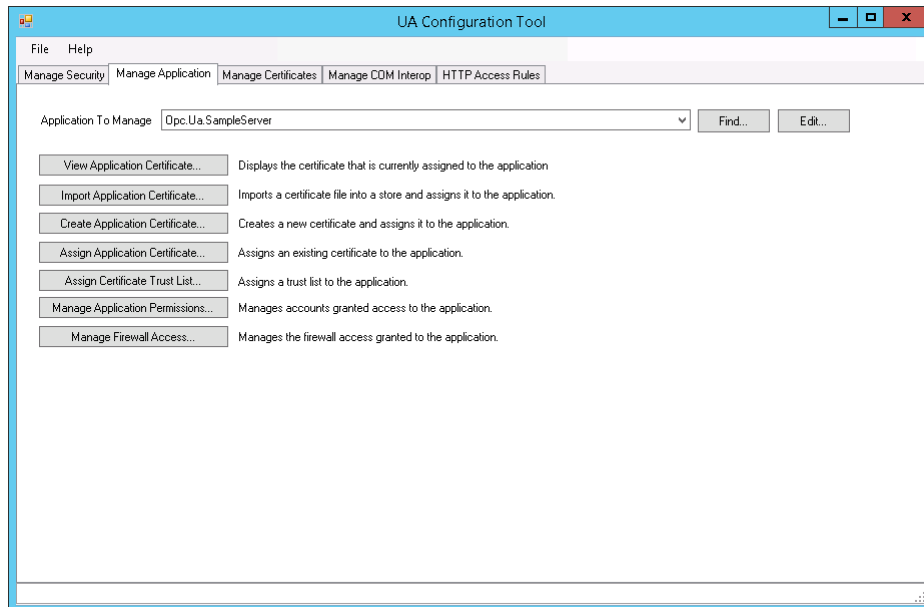
The steps provided here use the OPC UA Sample Server and tools provided by the OPC Foundation for illustration and example purposes. You may use a different OPC UA server and configuration tools, but the main steps remain the same.

### To create a self-signed OPC UA compliant certificate and key pair

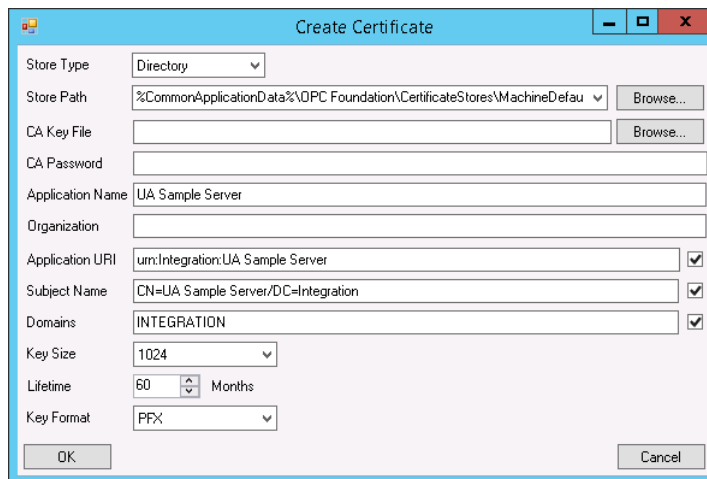
1. Start the UA Configuration Utility by clicking the Start button on the Windows task bar and pointing to Programs. Point to the OPC Foundation folder that contains the sample applications, click UA 1.02, then Sample Applications, and then click UA Configuration Tool (OPC.UA.ConfigurationTool.exe). The **UA Configuration Tool** appears.



2. In the **Application To Manage** list box, enter "Opc.Ua.SampleServer" if not pre-filled.



3. Click the **Manage Application** tab, and then click **Create Application Certificate**. The **Create Certificate** dialog box appears. Enter information in the appropriate text and list boxes.



- Store Type:** From the **Store Type** drop-down list, select "Directory" if your OPC UA client is running on the local computer.
- Store Path:** We recommend that you create a temporary directory to store and copy the certificate and the private key. For example, C:\Tmp\TestUACert. Otherwise, accept the default path or select another on your computer. In this example, we accept the default path.
- CA Key File:** Optional if you are creating a self-signed certificate.
- CA Password:** Optional if you are creating a self-signed certificate.
- Application Name:** The name of your OPC UA server.
- Organization:** Optional.
- Application URI:** The text box will be pre-filled with your OPC UA server. You can specify another using this format:  
*urn:<computer name>:<OPC UA server name>*

- h. **Subject Name:** The text box will be pre-filled with your OPC UA server name. You can specify another using this format:  
*CN=<OPC UA server name>/DC=<computer name>*
  - i. **Domains:** Enter the computer/node name. You can use "localhost" if on the local computer.
  - j. **Key Size:** Select a key size from the list box.
  - k. **Lifetime:** Select a key lifetime in number of months.
  - l. **Key Format:** Select "PFX" from the list box.
4. Click **OK**. The certificate and keys are created in \certs and \private subdirectories under the Store Path you specified in step 3b.

## Using the Self-Signed Certificate and Key Pair

You can rename the certificate and private key files you created with the steps outlined in Creating a Self-Signed Certificate and Key Pair. The certificate and keys are created in \certs and \private subdirectories under the **Store Path** you specified.

- You can rename the certificate file to an easily managed name, but you must maintain the ".der" filename extension. For example, *UAClient.der*.
- You can rename the key file to an easily managed name, but you must maintain the ".pfx" filename extension. For example, *UAClient.pfx*.
- You also can retain the filenames generated by the certificate and pair creation process.

### To use the self-signed certificate and key pair

1. Copy the certificate file, *UAClient.der* in this example, from your **C:\...\certs** directory into the path **%CommonApplicationData%\<Application Name>\CertificateStores\OPCUAClient\certs** on every computer on which you want to deploy the OPC UA Client service instance.

For example, the application name might be *ArchestrA* in the file path.

By doing this, you have replaced the file with same name in that directory.

---

**Note:** The path to store program data is %CommonApplicationData%. By default, this location is C:\ProgramData\.

---

2. Copy the private key file, *UAClient.pfx* in this example, from your **C:\...\private** directory into the path **%CommonApplicationData%\<Application Name>\CertificateStores\OPCUAClient\private** on every computer on which you want to deploy the OPC UA Client service instance.

For example, the application name might be *ArchestrA* in the file path.

By doing this, you have replaced the file with the same name in that directory.

---

**Note:** The path to store program data is %CommonApplicationData%. By default this location is C:\ProgramData\.

---

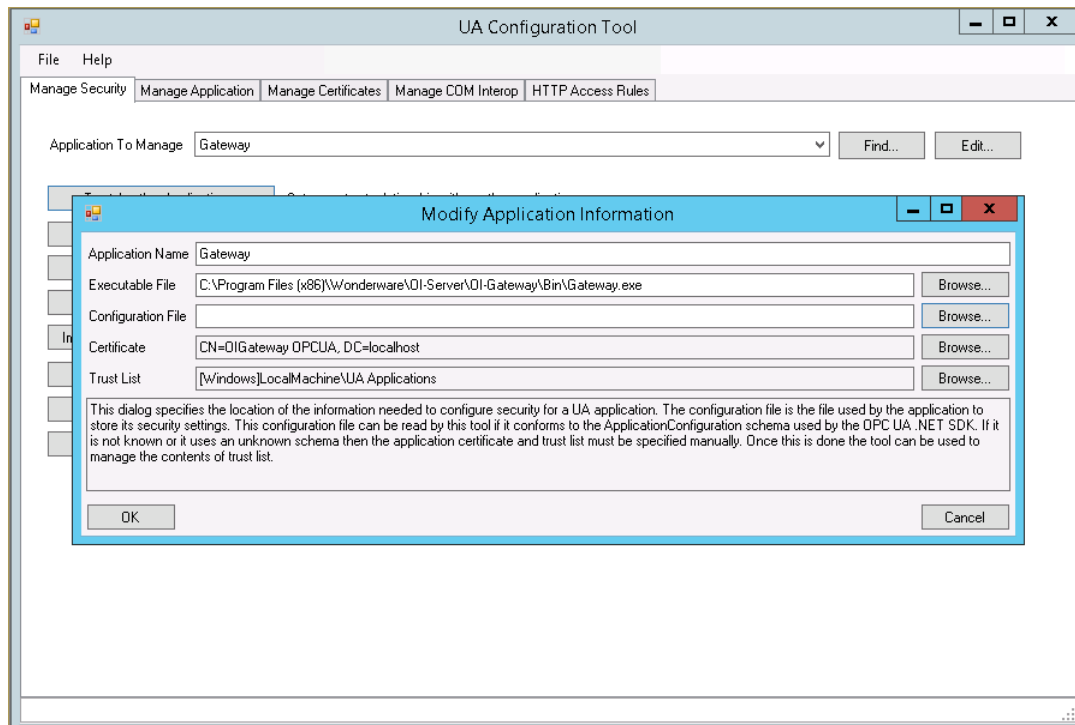
3. On the computer where the targeted OPC UA Server may be running, copy the certificate file *UAClient.der* from your C:\...\certs directory into the directory where you want to add the trusted cert for the OPC UA Server.
4. Deploy the UA Client service instance to the node.

## Adding a Certificate to the Trusted List

You can add a certificate to the list of trusted certificates for the Gateway. You can also add a certificate for ASB (ArchestrA Service Bus) OPC UA client service, if applicable. Use the same procedure and tools to add a certificate to the trusted list of an OPC UA server.

### To add a certificate to the trusted list

1. Open the **UA Configuration Tool**. Use Windows **Start** to navigate to OPC Foundation, then run **UA Configuration Tool**.
2. On the **Manage Security** tab, click **Edit** next to the **Application to Manage** text box. The **Modify Application Information** dialog appears.



3. Point to the Gateway.
  - a. Use the **Browse** button to find and select the **Executable File**. This will populate the **Application Name** text box.
  - b. Use the **Browse** button to find and select the **Configuration File**.

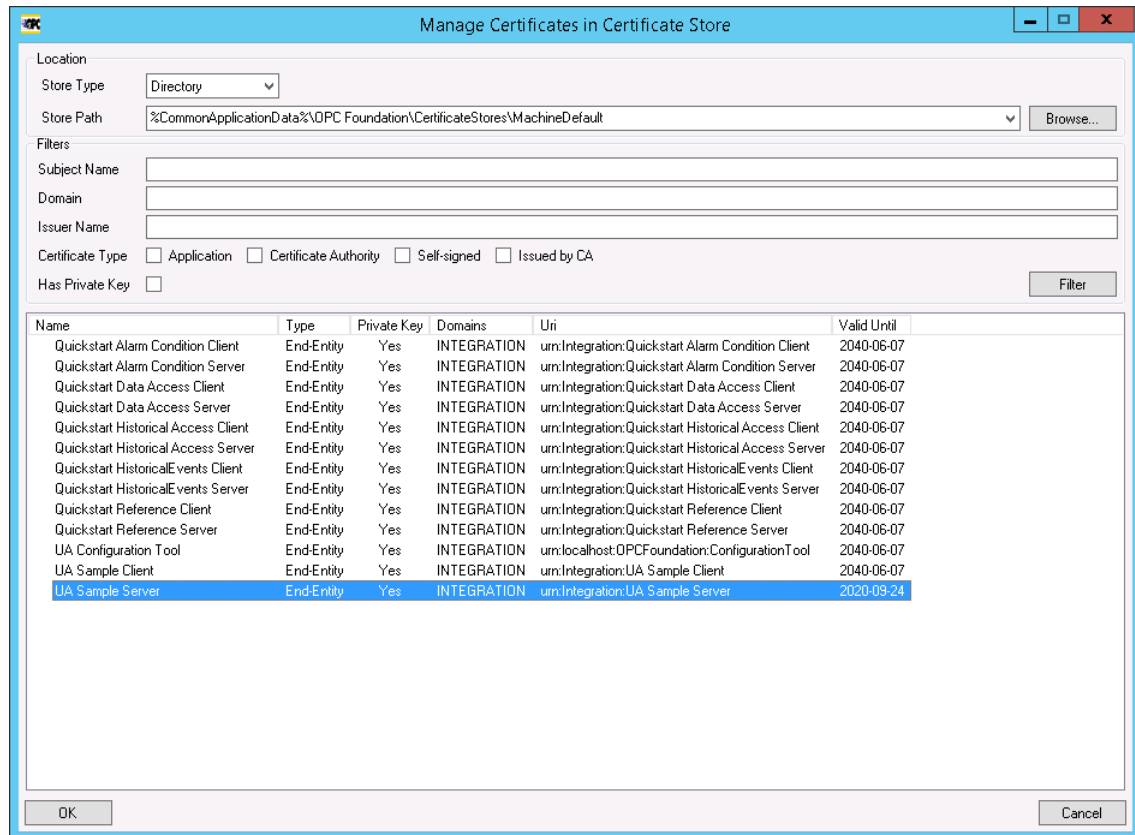
Click **OK** to close the dialog.

---

**Note:** If you are adding a certificate to a UA server trusted list, point to the server executable file and application name.

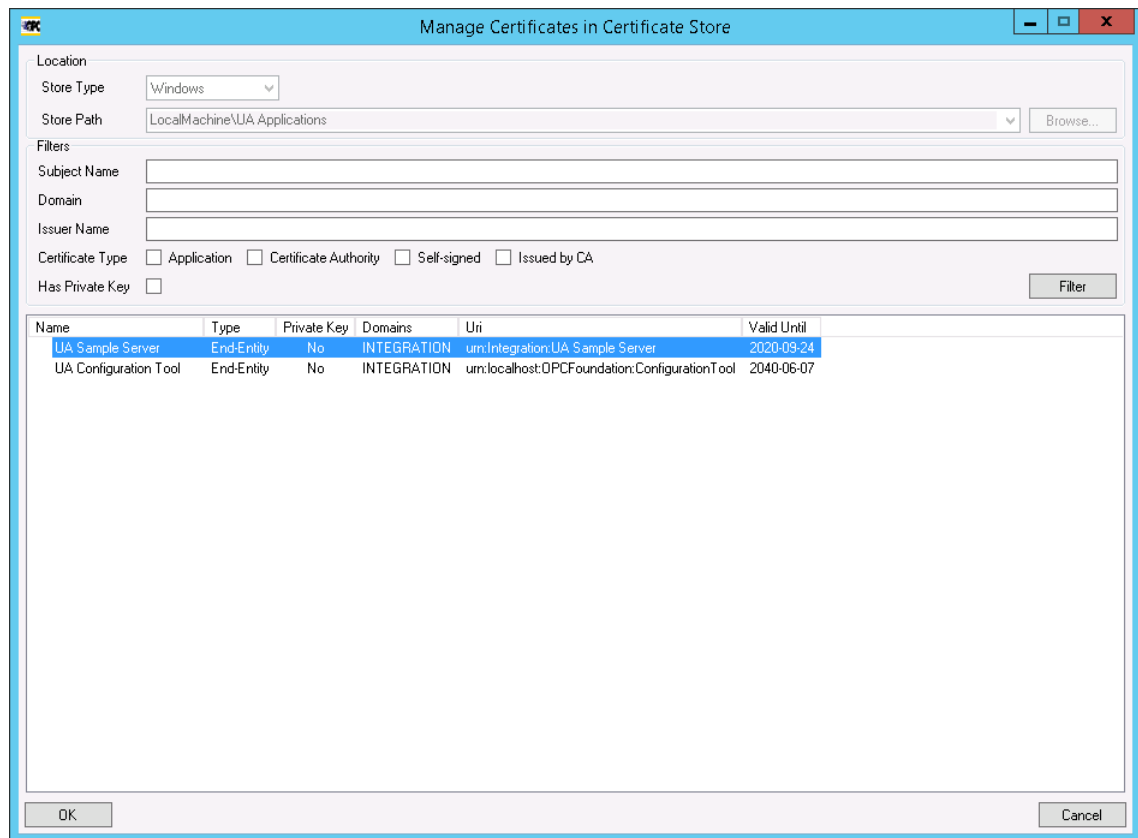
---

4. On the **UA Configuration Tool**, click **Select Certificate to Trust**. The **Manage Certificates in Certificate Store** dialog appears.



5. Select the server certificate you want to trust, then click **OK**.

6. Confirm the certificate has been added to the trust list. On the **UA Configuration Tool** dialog, click **View Trusted Certificates**. The **Manage Certificates in Certificate Store** dialog appears, and displays a list of trusted certificates.





# Index

## \$

\$SYS\$Status • 79, 95

.

.csv file • 63

.csv file in Excel • 64

## A

About Accessing Data in the Gateway • 71

About the OI Gateway • 9

Accessing Data Using DDE/SuiteLink • 72

Accessing Data Using MQTT • 72

Accessing Data Using OPC • 71

Accessing Data Using OPC UA • 72

Accessing the Data in the Gateway • 71

Activate Server • 13

Activating/Deactivating the Gateway • 68

Active Device • 75, 85

actual PLC item names • 63

Add command • 63

Adding a Certificate to the Trusted List • 107

adding DDE/SuiteLink data source object • 36

adding device items • 63

adding group to InTouch object • 42

adding InTouch data source object • 41

adding item references • 63

adding topic to DDE/SuiteLink object • 36

Adjusting for Time Zones • 43

alias names • 63

application name • 72

ArchestrA Data Conversion • 23

ArchestrA Item ID Prefix • 20

ArchestrA Item Names • 21

ArchestrA Message Exchange • 11

ArchestrA System Management Console • 13

ArchestrA to DDE/SuiteLink Conversions • 24

ArchestrA to OPC Conversions • 25

ArchestrA.DASFSGateway.2 • 67, 68

ArchestrA.FSGateway.2 • 13

ArchestrA-DDE/SuiteLink Mappings • 23

ArchestrAGroup object • 20

ArchestrA-OPC Mappings • 25

archiving configuration sets • 68

Archiving Configuration Sets • 68

## B

Before Configuring the Gateway • 14

## C

Clear All command • 63

clearing all device items • 63

comma separated values file • 63

command

Add ArchestrA Object • 19

Add ArchestrAGroup Object • 20

Add DDE Object • 36

Add InTouch Object • 41

Add InTouchGroup Object • 42

Add OPC Object • 28

Add OPCGroup Object • 30

Add SuiteLink Object • 36

Add Topic Object • 36

Communication Failures • 99

Component Environments • 103

Configuration node • 68

configuration set • 63

Configuration Set Name • 68

configuration view

Device Items • 63

Global Parameters • 13

New\_ArchestrA\_000 Parameters • 19

New\_ArchestrAGroup\_000 Parameters •  
20

- New\_DDE\_000 Parameters • 36
  - New\_InTouch\_000 Parameters • 41
  - New\_InTouchGroup\_000 Parameters • 42
  - New\_OPC\_000 Parameters • 28
  - New\_OPCGroup\_000 Parameters • 30
  - New\_SuiteLink\_000 Parameters • 36
  - New\_Topic\_000 Parameters • 36
  - Configuring a DDE/SuiteLink Data Source Object • 36
  - Configuring a DDE/SuiteLink Topic Object • 36
  - Configuring an Arcestra Data Source Object • 19
  - Configuring an Arcestra Group Object • 20
  - Configuring an InTouch Data Source Object • 41
  - Configuring an InTouch Group Object • 42
  - Configuring an MQTT Data Source Connection • 46
  - Configuring an MQTT Group Connection • 47
  - Configuring an MQTT Publisher • 49
  - Configuring an OPC Data Source Object • 28
  - Configuring an OPC Group Object • 30
  - Configuring an OPC UA Data Source Object • 54
  - Configuring an OPC UA Group Object • 58
  - Configuring Arcestra Device Items • 21
  - Configuring Data Source Redundancy • 16
  - Configuring DDE/SuiteLink Device Items • 37
  - Configuring Device Item Definitions • 13, 63
  - Configuring InTouch Device Items • 43
  - Configuring MQTT Device Items • 48
  - Configuring OPC Device Items • 31
  - Configuring OPC UA Device Items • 59
  - Configuring OPC UA Security • 55
  - Configuring the Gateway • 13
  - Configuring the Gateway as Not a Service • 68
  - Configuring the Gateway as Service • 67
  - Configuring the OPC UA Namespace • 56
  - connecting to
    - Arcestra data source • 19
    - DDE/SuiteLink data source • 35
    - InTouch data source • 41
    - OPC data source • 28
  - Connecting to a DDE/SuiteLink Data Source • 35
  - Connecting to an Arcestra Data Source • 19
  - Connecting to an InTouch Data Source • 41
  - Connecting to an MQTT Data Source • 45
  - Connecting to an OPC Data Source • 28
  - Connecting to an OPC UA Data Source • 53
  - Connecting to Data Sources • 19
  - Connection Diagnostic Items • 89
  - Connection Protocol • 41
  - Contact Us • 3
  - Creating a Self-Signed Certificate and Key Pair • 104
  - creating device items • 63
  - credentials • 19
  - CSV file • 64
- ## D
- DAServer Manager • 13, 103
  - DAServer Manager documentation • 13
  - DAServer Manager tree • 13
  - Data Quality • 92
  - Data Source Protocol Layer • 103
  - Data Source Redundancy • 75
  - DCOM • 10
  - DDE • 11
  - DDE communications protocol • 72
  - DDE/FastDDE • 11
  - DDE/SuiteLink • 67, 69
  - DDE/SuiteLink Client to Any Data Source – Write Errors • 97
  - DDE/SuiteLink Data Conversion • 39
  - DDE/SuiteLink Item Names • 37
  - DDE/SuiteLink to Arcestra Conversions • 24
  - DDE/SuiteLink to OPC Conversions • 35, 39
  - DDE/SuiteLink-OPC Mappings • 33
  - DDEStatus • 96
  - Default Group • 13
  - Delete command • 63
  - deleting device items • 63
  - Demand Read After Poke • 30
  - device group • 71, 79
  - Device Groups and Device Items • 63
  - device groups. • 69
  - Device Group-Specific System Items • 81
  - Device Item Definitions • 63
  - device items • 13
  - Device Items configuration view • 63, 64



- Device Items dialog box • 64
- Device Items tab • 63
- Device-Specific System Items • 80
- Diagnostic Item Syntax • 87
- diagnostics • 11
- Distributed COM • 10
- Domain • 19
- duplicate items • 64
- Dynamic Data Exchange • 11

**E**

- Error Messages and Codes • 97
- Example #1 • 22, 37, 44
- Example #2 • 22, 38, 45
- Export command • 64
- Export feature • 64
- Export FSGateway device item data • 64
- Exporting and Importing the Gateway Item Data • 64
- exporting device items • 64

**F**

- FastDDE • 11
- features • 11
- FS Gateway hierarchy • 13
- FS Gateway version • 95

**G**

- Gateway Configuration • 13
- Gateway Post-Configuration Steps • 67
- Global Diagnostic Items • 87
- Global Parameters • 13, 69
- Global System Item • 79
- group • 79
- group name • 71
- groups • 13

**H**

- Handling Time Zones with the Time Property • 43
- hierarchical location • 79
- HMI • 10
- Hot Configuration • 69

**I**

- Import command • 64
- Import feature • 64
- Import FS Gateway device item data • 64
- importing device items • 64
- In-Proc/Out-of-Proc • 69
- Installing a Self-Signed Certificate and Key Pair • 103
- Instantiating the Gateway • 76
- InTouch Data Conversion • 45
- InTouch Item Names • 44
- InTouch Runtime Node • 41
- InTouch WindowViewer • 96
- Introduction to the OI Gateway • 9
- IOStatus • 96
- Item Browse Path • 41
- item name • 71, 72
- Item Reference column • 63
- item suffix • 71
- ItemID • 71

**L**

- Large-scale edit on item data • 64
- Licensing for Gateway Features • 77
- link name • 71
- Local node • 13
- Log Flag data • 97
- Log Viewer • 97

**M**

- Managing Digital Certificates • 103
- Managing the Gateway • 67
- manual or automatic service • 13
- Microsoft Management Console • 13
- MMC • 13
- Monitoring Connectivity Status with a Data Source • 95
- Monitoring the Status of Conversations with DDE/SuiteLink Clients • 96
- MQTT • 12
- MQTT Topic Names • 48

**N**

- Name column • 63

- network transport protocol • 11
- New\_ArchestrA\_000 Parameters configuration view • 19
- New\_ArchestrAGroup\_000 Parameters configuration view • 20
- New\_DDE\_000 • 19, 28, 36, 41
- New\_DDE\_000 Parameters configuration view • 36
- New\_InTouch\_000 Parameters configuration view • 41
- New\_InTouchGroup\_000 Parameters configuration view • 42
- New\_OPC\_000 Parameters configuration view • 28
- New\_OPCGroup\_000 Parameters configuration view • 30
- New\_SuiteLink\_000 • 20, 30, 36, 42
- New\_SuiteLink\_000 Parameters configuration view • 36
- New\_Topic\_000 Parameters configuration view • 36
- node name • 71, 72

## O

- Off-line edit on item data • 64
- OI Gateway • 101
- OI Server Engine • 102
- OLE for Process Control • 10
- OLE/COM technology • 10
- OPC • 10
- OPC Client Syntax • 31
- OPC Client to ArchestrA – Write Errors • 97
- OPC Client to DDE/SuiteLink Data Source – Write Errors • 98
- OPC communications protocol • 71
- OPC Data Conversion • 33
- OPC Item ID Prefix • 30
- OPC Item Names • 31
- OPC protocol • 71
- OPC to ArchestrA Conversions • 26
- OPC to DDE/SuiteLink Conversions • 34, 39
- OPC UA • 11
- OPC UA Client Diagnostic Items • 91
- OPC UA Data Conversion • 61
- OPC UA Diagnostic Items • 86
- OPC UA Item Names and Syntax • 59

- OPC–DDE/SuiteLink Mappings • 39

## P

- Ping item • 16, 75
- PLC configuration • 64
- Plug-ins • 102, 103
- Poke Mode settings • 13
- Poke Retries • 28
- prepare the FS Gateway • 13
- Preserving Time Stamps from the Publishing Source • 43
- Primary Device • 16, 75, 85
- ProgID • 71
- program name • 71

## R

- Read Only • 30
- Reading Values from the Gateway into Excel • 97
- Reconnect Attempts • 28
- Reconnect Period • 28
- Redundant • 16, 75, 85
- Redundant data source • 75
- reference • 101
- Reference • 101
- renaming device items • 63
- Runtime Behavior • 75
- Runtime Diagnostics and Error Reporting • 98

## S

- SCADA • 10
- Secondary Device • 16, 75, 85
- Server Name • 28
- Server Node • 28
- Service • 13
- setup.exe • 13
- SMC • 13
- snap-in environment • 13
- Standard System Items • 79
- Subscription Diagnostic Items • 90
- SuiteLink • 11
- SuiteLink communications protocol • 72
- Supported Client Protocols • 10
- Supported Data Sources and Clients • 10
- Supported VT Data Types • 31

system items • 79

  \$SYS\$ActiveDevice • 85

  \$SYS\$ErrorText • 85

  \$SYS\$FailoverReason • 85

  \$SYS\$FailoverTime • 85

  \$SYS\$ForceFailover • 85

  \$SYS\$PrimaryDeviceStatus • 85

  \$SYS\$SecondaryDeviceStatus • 85

  \$SYS\$StandbyDevice • 85

  \$SYS\$Status • 16, 75, 85

System Items • 79

Write Credentials • 19

## T

The Gateway Architecture • 101

The Gateway Data Source Hierarchy • 15

The Gateway Features • 75

The Gateway Redundant Device-Specific  
  System Items • 85

The Gateway-Specific System Items • 84

topic name • 72

topics • 13

Troubleshooting • 95

## U

Update Rate • 30

Use Another Configuration Set • 68

Use Group Name as Access Path • 30

User Name • 19

Using DDEStatus and IOStatus in Excel • 96

using different configuration sets • 68

Using Item Prefixes • 23

Using Item Suffixes • 31

Using the Self-Signed Certificate and Key Pair  
  • 106

Using Troubleshooting Tools • 95

## V

Value Time Quality (VTQ) • 11

Verifying a Self-Signed Certificate from an  
  Untrusted MQTT Data Source • 47

## W

WinSock • 11